Artificial Intelligence Based Classification for Urban Surface Water Modelling

Submitted by **Mohammed Chachan Younis** to the University of Exeter as a thesis for the degree of Doctor of Philosophy in Computer Science in October 2019

I certify that all material in this thesis/dissertation* which is not my own work has been identified and that any material that has previously been submitted and approved for the award of a degree by this or any other University has been acknowledged.

(Signature) ………………………………………………………………………………

Declaration

I hereby declare that, except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted, in whole or in part, for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains no outcome of work, done in collaboration with others, excluding those parts, as specified in the text and Acknowledgements. This dissertation contains fewer than 100,000 words, including bibliography, footnotes, tables and equations.

Mohammed Chachan Younis 2019

Acknowledgements

Thank you to my supervisors, Prof. Edward Keedwell, Prof. Dragan Savic and Dr Albert Chen for providing guidance and feedback throughout my project. The conversations with them inspired me into thinking outside the box from multiple angles to form objective and comprehensive critiques. I would also like to thank my mentor Dr. Mustafa Aziz, for his support. Another big thanks to Mr. Anthony Raine from South West Water for his guidance during my research, and for providing the data required for my area of study. Also, I would also like to extend my thanks to Iraqi MOHESR, Iraqi cultural attaché in London and University of Mosul for supporting me to undertake my doctoral study in University of Exeter.

ABSTRACT

Estimations and predictions of surface water runoff can provide very useful insights, regarding flood risks in urban areas. To automatically predict the flow behaviour of the rainfall-runoff water, in real-world satellite images, it is important to precisely identify permeable and impermeable areas. This identification indicates and helps to calculate the amount of surface water, by taking into account the amount of water being absorbed in a permeable area and what remains on the impermeable area. In this research, a model of surface water has been established, to predict the behavioural flow of rainfall-runoff water.

This study employs a combination of image processing, artificial intelligence and machine learning techniques, for automatic segmentation and classification of permeable and impermeable areas, in satellite images. These techniques investigate the image classification approaches for classifying three land-use categories (roofs, roads, and pervious areas), commonly found in satellite images of the earth's surface. Three different classification scenarios are investigated, to select the best classification model.

The first scenario involves pixel by pixel classification of images, using Classification Tree and Random Forest classification techniques, in 2 different settings of sequential and parallel execution of algorithms. In the second classification scenario, the image is divided into objects, by using Superpixels (SLIC) segmentation method, while three kinds of feature sets are extracted from the segmented objects. The performance of eight different supervised machine learning classifiers is probed, using 5-fold cross-validation, for multiple SLIC values, while detailed performance comparisons lead to conclusions about the classification into different classes, regarding Object-based and Pixel-based classification schemes. Pareto analysis and Knee point selection are used to select SLIC value and the suitable type of classification, among the aforementioned two. Furthermore, a new diversity and weighted sum-based ensemble classification model, called ParetoEnsemble, is proposed, in this classification scenario. The weights are applied to selected component classifiers of an ensemble, creating a strong classifier, where classification is done based on multiple votes from candidate classifiers of the ensemble, as opposed to individual classifiers, where classification is done based on a single vote, from only one classifier. Unbalanced and balanced data-based classification results are also evaluated, to determine the most suitable mode, for satellite image classifications, in this study. Convolutional Neural Networks, based on semantic segmentation, are also employed in the classification phase, as a third scenario, to evaluate the strength of deep learning model SegNet, in the classification of satellite imaging. The best results, from the three classification scenarios, are compared and the best classification method, among the three scenarios, is used in the next phase of water modelling, with the InfoWorks ICM software, to explore the potential of modelling process, regarding a partially automated surface water network. By using the parameter settings, with a specified amount of simulated rain falling, onto the imaged area, the amount of surface water flow is estimated, to get predictions about runoff situations in urban areas, since runoff, in such a situation, can be high enough to pose a dangerous flood risk.

The area of Feock, in Cornwall, is used as a simulation area of study, in this research, where some promising results have been derived, regarding classification and modelling of runoff. The correlation coefficient estimation, between classification and runoff accuracy, provides useful insight, regarding the dependence of runoff performance on classification performance. The trained system was tested on some unknown area images as well, demonstrating a reasonable performance, considering the training and classification limitations and conditions. Furthermore, in these unknown area images, reasonable estimations were derived, regarding surface water runoff. An analysis of unbalanced and balanced data-based classification and runoff estimations, for multiple parameter configurations, provides aid to the selection of classification and modelling parameter values, to be used in future unknown data predictions. This research is founded on the incorporation of satellite imaging into water modelling, using selective images for analysis and assessment of results.

This system can be further improved, and runoff predictions of high precision can be better achieved, by adding more high-resolution images to the classifiers training. The added variety, to the trained model, can lead to an even better classification of any unknown image, which could eventually provide better modelling and better insights into surface water modelling. Moreover, the modelling phase can be extended, in future research, to deal with real-time parameters, by calibrating the model, after the classification phase, in order to observe the impact of classification on the actual calibration.

Contents

List of Illustrations

List of Tables

CHAPTER ONE

1 INTRODUCTION

This chapter includes the motivation and the main objectives concerning this work. It also includes the scope of this research study in terms of aims and goals to be achieved and methodologies used. A complete structure of thesis organisation is further depicted.

1.1 Motivation & Objectives

Prediction of upcoming events is an essential part of disaster management [1]. It helps government disaster management agencies to implement the correct protective measures to avoid or minimise the damages caused by a disaster. One increasingly prevalent environmental disaster is flooding [2], and within this subject, urban surface water-based flooding is of critical importance, since it is reaching an alarming situation. This is due to paving over green spaces which are the natural mode of drainage and these exert pressure on the sewerage network. The effects of this kind of flooding are immediate in regard to the human population [3]. The most effective indicator for analysing the modelling of surface water in an urban environment lies with penetrable surfaces (i.e., roofs and roads) or impenetrable surfaces (i.e., vegetation areas) [4]. Predictions attained through the stormwater model include the overall runoff that results from the total surface of the subcatchment, and this takes into account both impervious and pervious areas. By classifying such areas of an urban catchment and by using a hydrological model, such as InfoWorks software, the potential for automated modelling of a surface water network can be explored [5]. A well-calibrated model behaves in the same way as the real system, within a range of tested conditions [6]. Remote sensing imaging and other spatial imaging data are a good source of detailed information regarding locations of impervious surfaces during examination of urban catchments. These imaging types also provide more relevant data for hydrological models and urban field studies compared to any surrogate data produced by artificial methods [7].

The main objective of this study is to explore scenarios and methods for designing a well-automated system to predict surface water runoff from a real-world remotely sensed image by providing a useful classification tool. There are multiple processes involved in this research study, which include data preparation, image analysis, classification and hydraulic modelling.

The objectives of the data preparation phase include a collection of all required images and data and preparation of ground truth/labelled data for the training of classification models.

The objective of image analysis phase is to divide image into multiple segments through superpixels based segmentation (SLIC) method, then extract various kinds of features (RGB, HSV and Texture) of an image and pick the best one for the current dilemma. The next objective is to visually analyse the extracted features by using a grouped box plot for the three classes of interest versus feature variables and to test all three kinds of feature, through the use of selective experimental classification algorithms.

The next objective is to apply three different scenarios of classification to pick out some classifiers for experiments and analyse their performance in order to choose the best settings for the classification phase to ensure an effective setup. This classification phase includes three scenarios. In the first scenario, the classification of the image is applied, using two pixel-based classification methods in a non-parallel and a parallel way. The second scenario involves the use of eight different supervised machine learning (ML) algorithms for classification. Another objective of this classification scenario is to design a new ParetoEnsemble classifier by the combination of individual classifiers used in experiments to design an even better classification model in terms of performance. The third scenario of the classification phase aims to perform deep learning classification, using semantic segmentation. The best model to be classified is picked along with optimal parameters by comparing the models from all three scenarios to be used in the modelling phase.

The objective of the final modelling phase is to combine the classification phase results with a hydraulic model, named InfoWorks ICM, in order to obtain an approximate modelling of surface water network performance by simulation of surface runoff flow and by estimating inundation conditions. The main objective in modelling InfoWorks ICM is to minimise the gap between model-simulated results and actual measurements [6], by evaluating the modelling performance which provides the selection of optimal parameters for future unknown data predictions and modelling.

1.2 Research Contribution

This thesis is focused on designing an application that combines a number of existing methods, to form an improved automated flooding estimation system. Regarding the classification phase, a new ensemble model design is put forward, which performs better than the traditional classification algorithms and ensemble models. In terms of a new application design, this research presents a new idea and a meaningful contribution to the literature, since prior studies concern either modelling or classification alone, whereas this work is linking these two totally different areas of research, to present a novel single system to automatically predict the flooding estimations, by taking satellite images as input.

On a top level approach, this research offers many applications, as explained above. In terms of specific novelties and specific contributions of this work, image division into multiple objects, followed by classification, based on objects instead of pixels, is something new to the field. A detailed analysis of object size and number impact on classification is another important contribution, helping researchers select appropriate superpixels parameter values in specific research problems. Another contribution of this study is the proposal of a new ensemble classification model design, utilising scientific concepts such as diversity, Pareto and Knee point, which provide even better classification results, compared to the traditional classification algorithms, as proven by a detailed comparison between the proposed ensemble and traditional classification models. This research also explored deep learning models, to analyse the performance of deep learning compared to the proposed ensemble algorithm. The deep learning approach also offers a possible solution to data scarcity in the research problems. Finally, all the classification models are compared, in order to select the best performing algorithm for generalisation of proposed system and modelling purposes. Finally, another important contribution of this work regards the analysis of the impact of these many attributes on runoff estimations. A comparison of runoff estimation results in many scenarios and conditions, such as correlation coefficient between classification accuracy versus runoff accuracy, is made to analyse the dependency between runoff performance and classification performance.

1.2.1 Publications

Journal papers:

1. Rhouma, M. B. H., Žunić, J. & Younis, M. C. (2017). Moment invariants for multi-component shapes with applications to leaf classification. *Computers and electronics in agriculture, 142*, 326-337. Elsevier.

2. Younis, M. C. & Keedwell, E. (2019). Semantic Segmentation on Small Datasets of Satellite Images Using Convolutional Neural Networks. *Journal of Applied Remote Sensing*, *13*(4), 046510.

Conferences:

3. Younis, M. C., Keedwell, E., Savic, D. & Raine, A. (2017, September). Evaluating Classification Algorithms for Improved Wastewater System Calibration. *CCWI 2017* – Computing and Control for the Water Industry Sheffield 5th-7th September 2017. (*Oral presentation*)

4. Younis, M. C., Keedwell, E. & Savic, D. (2018, April). Evaluating Image Classification Techniques for Improved Urban Wastewater System Model Calibration. In *EGU2018 General Assembly Conference Abstracts* (Vol. 20, p. 16327) Vienna 4th-13th April 2018. (*Oral presentation*)

5. Younis, M. C., Keedwell, E. & Savic, D. (2018, October). An Investigation of Pixel-Based and Object-Based Image Classification in Remote Sensing. In *2018 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 449-454). IEEE.

1.3 Thesis Scope

This study focuses on the semi-automated modelling of a hydraulic surface water model with the aid of fully automated satellite image classification into pervious and impervious segments, as previously mentioned. The scope of this work is elaborated through the diagram illustrated in Figure 1.1. In order to better explain the structure of this diagram, the procedural details will be explained in the following chapters. The block diagram of the proposed approach shown in Figure 1.1 can be broken down into five phases: 1) Data acquisition, 2) Data preparation, 3) Image analysis, 4) Classification, and 5) Surface water modelling.

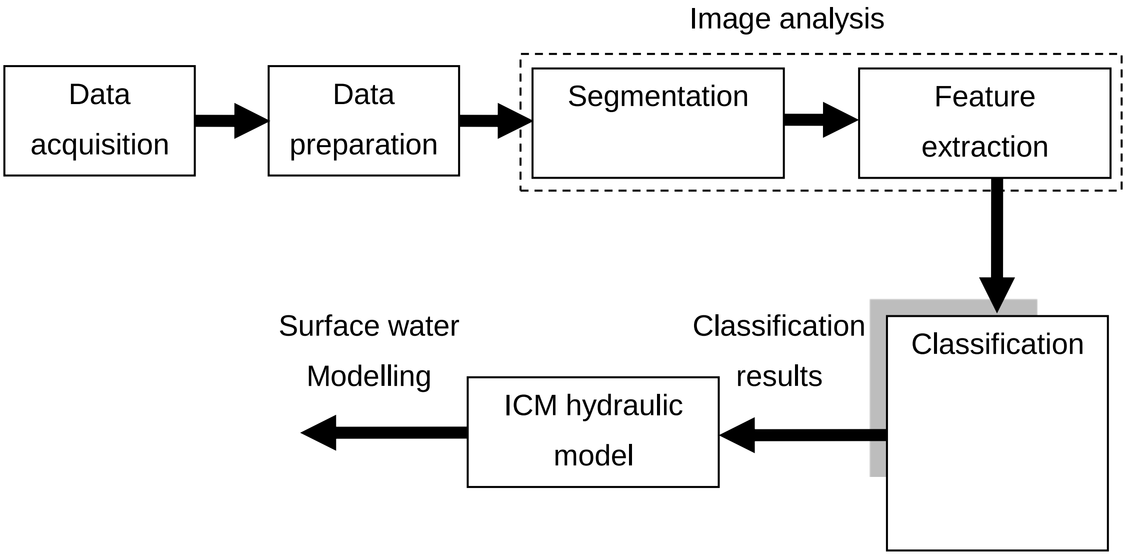


Figure 1.1: Process flow of the proposed approach: from image acquisition to the simulated model.

Image acquisition is the first phase of any computer vision system in the image classification area. This phase focuses on how, where, when, and what is the useful form of such imagery to gain a general idea of the examined area; for example, the quality and general characteristics to select suitable hardware and software components.

The second phase (data preparation) is dependent on the requirements and applications of users and systems under development, so it varies from one situation to another. Data preparation is a common description for operations to input data before any further processing. Detailed explanations and illustrations of the implementation of all such data preparation operations are given in the following chapters.

After data preparation, the image analysis phase covers super-pixels based segmentation (SLIC), followed by feature extraction to be performed on the images for the collection of features required for the next phase.

The fourth but, it would seem, the most important phase in this work is the classification phase, where the final goal of the classification system under development is to enable the allocation of an object whose class membership is unknown to one of several classes based on object features. The scope of this research also includes the design of a strong ensemble classifier to classify the data more accurately. This can be compared to individual classifiers where predictions work based on more than one classifier. This research will also employ some computational techniques for the classification phase, namely Pareto analysis, Knee point, and Diversity-based selection of classifiers and design a Weighted sum-based ensemble classifier called ParetoEnsemble.

In the last phase, the results of the classification phase are to be fed to the InfoWorks ICM hydraulic software to get an approximate model of a surface water network for more accurate runoff flow estimations. This is to be done by classifying the land-cover as impervious and pervious surfaces of rainfall events, and by using the Wallingford Procedure Model. More details related to these five phases reside in their respective chapters.

1.4 Thesis Structure

This thesis has been organised into seven chapters.

Chapter One, Introduction: this describes the objectives, research contribution and scope of the thesis along with an outline of the remaining chapters.

Chapter Two, the Background and Literature Review: this introduces essential useful background information for the methods followed in this study. The literature review includes the research work carried out by other researchers in the areas of the description of data acquisition, data preparation, image analysis, classification and modelling for prediction of runoff for surface water in the urban environment.

Chapter Three, Experimental Setup: this covers the steps of the image processing used for acquisition, data labelling, and image analysis phases. The two image analysis steps (image segmentation and feature extraction) are illustrated, and their implementation is given.

Chapter Four, Urban Land Cover Classification: this provides two scenarios (pixel-based classification and superpixels-based classification) as effective classification tools to classify real-world satellite images into permeable (i.e., vegetation) and impermeable (i.e., roads and buildings) surfaces.

Chapter Five, Convolutional Neural Networks based Segmentation and Classification Results Analysis: this provides the 3rd scenario for classification of satellite images, by utilising deep learning models. It further compares the classification model results from three classification scenarios to select the best performing model. Also, this chapter analyses the quality of generalisation of the trained models by testing two unknown images.

Chapter Six, Surface Water System Modelling: this was carried out using the best classification results from the previous chapter with the InfoWorks ICM hydraulic software to model the runoff in a stormwater network.

Chapter Seven, Conclusions and Further Work: this presents the conclusions of the work followed by suggestions for future works.

CHAPTER TWO

2 BACKGROUND & LITERATURE REVIEW

This chapter includes all required background knowledge about the processes involved in this research study. In addition, there is a detailed literature review, concerning the phase contained in the methodology section. Work done by other researchers, related to each phase, is reviewed in detail, while background information about the specific phases, undertaken in each subsection of this chapter, is also included.

The first phase is about data acquisition, where a brief introduction of respective techniques is given, along with various data acquisition methods, used by other researchers in this specific field. Preparation and adjustment of the data acquired, follows, in the next phase.

Section 2.2 contains background information about data preparation steps, followed in this research, in order to make the data ready for experiments. Various data preparation methods, used to convert data into a useful form, followed by other researchers in the prior art, are also discussed. Following, the modified data, as derived in this phase, can be used in the image analysis phase.

A brief background knowledge about concepts, such as image segmentation and feature extraction, is given in detail further, to better understand the image analysis phase. The SLIC Superpixels-based segmentation method is described in full detail, as it is the segmentation method applied in this study. Also, three different types of feature extraction methods are described, to fully comprehend the various kinds of features, used to distinguish the different class objects in this research. The segmented objects and features, extracted from images, are going to be used in the following classification phase.

Section 2.4 gives an introduction to satellite image classification, followed by a thorough implementation of different classification methodologies, in three different classification scenarios. A literature review of the classification methods, used in this field, is presented and discussed extensively. The classification results of this phase converted into an appropriate format, will be used in the next phase, to simulate the modelling of the stormwater network.

Finally, detailed background information, about the use and function of the modelling environment, is provided, in order to understand the process of simulating surface water runoff modelling and how better predictions and insights, regarding flooding, are possible.

2.1 Data Acquisition Phase

The first phase, as shown in Figure 1.1 (Chapter 1), is image acquisition, the process of getting the required data from the source, to perform analysis of the results. This data is produced by image sensors, providing data associated with a specific location (geospatial), in the form of digital maps. The data might be attributed as colours, symbols or any tabulated form [8]. This phase is focused on capturing the data regarding various land-cover classes of the area under examination, to provide a representative example of urban catchments.

There are various modes of data acquisition, reported in the literature. For instance, the data used by [9] includes high spatial resolution images, captured by ALOS (Advanced Land Observing Satellite), consisting of both visible and near-infrared bands. In [10], a QuickBird image was utilised, captured for assessing a classification technique for land-use/land-cover, in a complex urban-rural environment. Similarly, in [11], a Quickbird image was used, where the area of interest was covering both urban segments and undeveloped regions, providing a diversity of urban land use and land cover classes. IKONOS Quickbird image data was also used in [12], to assess the impact of multiple classification techniques on urban land-cover classification. The main purpose of using the image was to examine whether the proposed classification technique could be effectively applied to an entirely different environmental setting. There are several researchers, in the literature, who have analysed another satellite image format. One such example is the study conducted in [13], for assessing land-cover change, using Landsat Thematic Mapper (TM) images. Likewise, Landsat images were used in [14], for mapping land-use changes, where USGS earth explorer was used, to download the main scenes.

This section provided a detailed literature survey, regarding data acquisition and selection, for specific research problems. The data selected by other researchers provide different results, based on specific research problems and limitations, observed in data type and methodologies used. Literature overview provides an insight into the nature of data, selected for the research problem posed in this study. Finally, appropriate hardware and software components are selected, based on the specific data type, used in this work.

2.2 Data Preparation Phase

Before the image analysis phase, some pre-processing of raw data is carried out. This phase covers all the operations necessary, to bring the input image into a form ready for the next phase, of image analysis. Moreover, this phase is crucial, because the effectiveness of the following segmentation may fail if this phase is not performed correctly. However, applying such a process always depends on the goal of the study. If, for instance, "a check of a specific land-cover or object, using a satellite image", is the purpose, then visual interpretation might be enough, while image enhancement and/or the removal of data errors might not be necessary [15]. In [16], it is clearly illustrated that the aim of the preparation phase is to enhance image data, suppressing unwanted distortions, improving the image, for further processing. According to [17], the pre-processing of data, using methods such as radiometric, atmospheric and geometric corrections, is a preparatory phase to improve the image quality for further analysis. This sort of approach has been considered in [18], where the two sets of images (Landsat TM and Landsat Operational Land Imager (OLI) were geometrically corrected, to remove distortion, caused by Earth rotation or sensor movement. Moreover, geometric rectification, based on a road network map, was utilised, to register ALOS multi-spectral satellite images, in [9]. The method applies the nearest neighbour algorithm, to resample the data. A similar approach has been adopted by [19], for geometrically registering IKONOS and Landsat ETM+ images. Meanwhile, a radiometric correction was used in [20], as a pre-processing step for correcting Landsat images, before the classification stage. Nevertheless, [21] states that the preparation stage of remote sensing data is mainly for the elimination of data registration errors. These errors involve earth rotation, earth curvature, instability of the platform, topographic effects, radiometric correction, noise removal, and georeferencing.

Generally, the data preparation phase includes some important techniques, applied to the input data as a base for further analysis, while simultaneously avoiding unnecessary steps that may introduce additional artefacts, without any additional value.

2.3 Image Analysis Phase

The advancements made in the area of remote sensing have made it possible to acquire high resolution data and allowed the extarction of a wide range of features for analysis, monitoring and evaluation. The extraction of such useful features has continuously increased the demand of automated image segmentation and analysis in the operational field [22]. Image analysis mostly deals with the extraction of image graphical and numerical information, which is further used for defect estimation, image classification and in many cases for the properties estimation of any visual object in the image [23].

Image analysis aims to extract information, useful in solving application-based problems. Image analysis is used to isolate and distinguish the objects of interest, from its surrounding environment, and to extract features, useful in the classification tasks performed, after this phase [24]. Image analysis is a relatively challenging and crucial phase that decreases the complexity of the next working phase, to some extent [25]. On the other hand, any wrong perception, in this phase, will introduce error in the information, transferred to the next phase.

2.3.1 Image Segmentation Step

In computer vision, the segmentation of an image denotes the process of dividing an image into multiple small, non overlapping parts, called segments. Each of these segmented parts consists of multiple pixels, connected together and homogeneous in terms of one or more features, while two segments connected to each other are not considered homogeneous [26]. Generally, image segmentation systems abide by these rules [27]:

- Characteristics such as intensity value, colour or texture of regions, in an image segment, must be uniform and homogenous.
- Region interior needs to be simple and without any tiny holes.
- The values of characteristics/attributes, set as rules for the segmentation of adjacent regions, should be considered appropriately, so as to efficiently differentiate regions of interest.
- Boundaries of each region ought to be simple, regular and spatially accurate.

Image segmentation works on the basis of discriminating features such as texture, colours, grey levels, depth or motion [28], as can be found in abundance, among various studies. Nonetheless, there is no one single procedure available that suits all images. Similarly, not all methods, used for an image, can be considered effective. This shows that image segmentation depends on the variation of object shapes, their type and the discrimination levels of features, as also demonstrated in [29].

A vast number of publications has analysed image segmentation methods and the various designs they constitute. For instance, in [30], image segmentation methods are classified into three schemes, namely features thresholding, region detection and boundary extraction based methods. One more author categorised image segmentation into six schemes, based on techniques, such as single, centroid and hybrid linkage based region growing methods, space measurement guidance based clustering, spatial clustering and merging and splitting based methods [31]. A significant image segmentation classification is presented in [32], which incorporates the thresholding, region, edge and boundary based methodologies, with the possibility of integration of any of these procedures. In [33], a new model for image segmentation is presented, exhibiting high accuracy despite noisy data, in the regions boundaries estimation. The region-based and boundary detection-based methods are further combined successfully in [34], while the properties of threshold-based and region-based methods are jointly optimised in [35]. Another hybrid approach is proposed in [36], regarding range image segmentation, by combining edge and region based segmentation techniques. A new segmentation algorithm, the SLIC superpixels method, was implemented, as it efficiently decomposes an image into visually homogeneous regions and is efficient, in terms of computation and memory. It divides the image into relatively small homogeneous patches, which can then be classified, based on the known features [37].

Based on the present work, image segmentation has proven to be a basic procedure, despite being an exhaustive one, as it provides the input to a higher-level image processing, such as the classification. In this section, some of the most common segmentation algorithms are discussed. Next, in section 4.3.1.11, the SLIC algorithm, used in experiments of this research, is presented.

2.3.1.1 Threshold Based Segmentation Method

Thresholding is one of the simplest methods of image segmentation, where binary images are created from a greyscale image [38]. This technique is useful in distinguishing the foreground from the background of an image [39]. These methods can be mainly divided into three techniques: global thresholding, local thresholding and dynamic thresholding. Image thresholding techniques are employed, when the adjacent pixels follow similar or close criteria, such as grey level and colour, belonging to the same segment type. However, the main drawbacks of these approaches are the abandonment of spatial relationships between the region pixels and high sensitivity to noise [40].

2.3.1.2 Grey Level Thresholding Method

In binary image segmentation, one straightforward approach~~es~~ is the grey level thresholding method, dividing the grey level range of the given image into different regions. Each of these regions is specified by two threshold values, as described in [26]. Several designs were produced, to tackle the issue of threshold limits definition, which proves to be a disadvantage of this method. One of the usual procedures is the histogram method, where the threshold values are obtained from the peak and valleys of the histogram curve [41] [42]. This approach refers to the grey values of any similar pixels region, representing a normal distribution like curve, with a peak occurring (the most frequently occurred pixels) at the mean value, while the two tails determine the minimum and maximum limits of the grey levels of pixels region.

2.3.1.3 Colour Thresholding Method

Three dimensional colour spaces are developed from the colours of the image pixels, using a colour thresholding design. Following, the clustering of homogenous and similar colour characteristics, based on the distance in the given space, is carried out [43] [44]. Even though the spatial distribution of the image pixels is not considered in this procedure, it ought to be distinguished from the colour slicing technique, which utilises the colour as the third dimension to the two-dimensional space of the image domain.

2.3.1.4 Multi-Spectral Image Classification Method

Digital information in images is more accurately assessed applying multispectral image based classification methodologies, used in remote sensing to extract useful information from available satellite imagery [45]. Multispectral image classification is based on feature space measurements [46]. However, the major difference, between this and spatial based segmentation, is that now segmentation is done in the feature space of multispectral bands, taken by remotely sensed platforms, where each band represents a feature.

2.3.1.5 Region-Based Segmentation Method

This method works on the basis of homogeneity of adjacent pixels, where image under consideration is divided into different segments [47]. The region-based process results in the partitioning of the image into different segments, taking place in the image domain, while the partitioning, in the feature space thresholding method, occurs in another space, without providing knowledge of the spatial coordinates of the image pixels [29].

2.3.1.6 Region Growing Method

A pixel, known as the seed, is grown, by linking it to the neighbouring pixels with similar characteristics. This continues until similar neighbouring pixels are no longer present in the image, for segmentation, for growing regions [48]. This process of growing regions is carried out in the range of a 3x3 window, using 4-connected or 8-connected neighbourhood algorithms.

2.3.1.7 Region Splitting and Merging Method

This methodology divides an image into specific parts, while, based on the homogeneity measurement, the similar parts are combined. First, it involves a given region passing the homogeneity test, using one of the image characteristics, such as grey level, colour or texture. Next, the image is separated into regions of a similar size. Following, the homogeneity test is applied and, if the region passes, it continues to merge with neighbours. Finally, the whole 3-step process is repeated in a loop until all regions pass the test [49].

2.3.1.8 Texture Segmentation Method

This type of image segmentation approach is rather complicated, due to the inability to detect the type of textures in an image, the number of different textures present and the regions of specific textures. Actually, in order to carry out this process, the type of textures, present in the image, are negligible, while the only condition, for two different textures to be present in an image, is usually satisfied in adjacent regions. The quality of input features profoundly affects the performance of this image segmentation method [50]. A massive number of studies has been dedicated to discovering texture parameters, adequate for classification, generally involving features, concerned with adequately characterising each region texture. For example, some of the conditions of features, taken into account, are co-occurrence matrices, fractal dimension, Markov random fields, etc. [51].

2.3.1.9 Clustering Based Segmentation Method

In this image segmentation approach, individual elements are positioned into groups, according to some metrics of similarity, among the elements in that group. The most straightforward procedure, in the clustering method, is to split the space into regions desired, by selecting the centre or median, along each dimension and dividing it there. This is done repetitively until space is separated into the specific number of regions required [52]. The aim of clustering, which is an unsupervised learning problem, is to identify clusters that can be considered as classes. Basically, clustering methods are of two types: one is called hard clustering, such as k-means, and the other is called soft clustering, such as fuzzy c-mean clustering [53].

2.3.1.10 Boundary/Border Based Segmentation Method

The detection of image pixels that intermediate two different regions is based on the understanding that pixel values alter instantaneously, at the perimeter between two different regions, in the case of boundary or border-based methods. The edge enhancement and detection methods such as Laplace, Sobel, Canny and Robert operators [54], are applied within the framework of various methods involved in the detection of region boundaries. According to the edge detection technique, borders are first enhanced and detected as line segments. Next, they are linked to form the entire border, which is generally applied to a multi-resolution image, beginning from low to high resolution [55]. On the other hand, the brightness or colour of the border points and even the texture of the region itself are the major components under consideration, in the edge enhancement techniques. This method is handy for simple images, consisting of regular regions, such as engineering drawings.

2.3.1.11 Superpixels Based Segmentation Methods

The major challenge in object-based classification is the art of robust segmentation of objects. The term ‘superpixels’ refers to a set of image pixels that share similar visual features. Generally, it regards clustering according to colour and distance characteristics of image pixels, while specifically superpixels prove to be very helpful for image segmentation, since they are more efficient than traditional techniques [56]. Various algorithms exist that segment superpixels; however, SLIC algorithm is the most state-of-the-art and best performing algorithm, while it needs a minimum of computational power to run efficiently. It is our understanding that the following most necessary properties are desirable [37]: 1) Image boundaries must be adhered to by superpixels. 2) If there is a pre-processing step, requiring computational complexity, it is vital that superpixels are able to meet that requirement, with efficient memory usage and simple overall process. 3) For the purpose of segmentation, there should be an improvement in the results, both in terms of quality and speed, when superpixels based segmentation method is used.

When performing the classification and segmentation processes, superpixels can be a very useful method, particularly in the case of larger images. It helps in the image division into groups of regions, which are more meaningful, in terms of structure. The created regions have boundaries that take into consideration the original image and its existing edge information. Following the division of each image into superpixel sections, it is possible to utilise classification algorithms, in order to classify each region, instead of solving the potential issue of classification, concerning the grid of the full original image. The benefit in performance grows when superpixels approach is used, especially when addressing issues related to image classification, while simultaneously maintaining high-quality segmentation [57].

In this study, the segmentation of the image (as a grid of pixels) is accomplished by the superpixels (SLIC) method, which adopts a k-means clustering method, to group pixels into regions with similar colour space (values), to reduce the complexity of the segmentation. The most important benefits of using SLIC are [35]:

1. Simple to implement and easy to apply: the only parameter required is the desired number of superpixels.
2. Efficient in terms of computation and memory: its advantage in solving the problems increase with the size of the image, as it is the most memory-efficient method, to handle large images, while other methods are, in comparison, very demanding in memory.

The main steps of SLIC superpixel segmentation algorithm are as follows [37]:

```
Initialisation Step:

  • Choosing initial centres of clusters $C_v = [l_v, a_v, b_v, x_v, y_v]^T$ by sampling the
    image pixels into regular steps of grid S.
  • Moving cluster centres at the location having least gradient in 3x3
    neighbourhood window.
  • Set label $l(j) = -1$ for pixel j.
  • Set distance $d(j) = \infty$ for pixel j.

repeating
   Assignment:
    for centre of cluster $C_v$ do
      for pixel j around cluster centre $C_v$ in the nearby region of 2S x 2S do
            Compute distance L between j and $C_v$
                  If $L < d(j)$ then
                        Set $d(j) = L$
                        Set $l(j) = v$
                  end
      end
    end
   Updating:
      Computing centres of new clusters
      Computing error R.
until $R \le$ threshold
```

Where, v represents the total clusters, which are specified based on the desired count of object segments, to be extracted from an image. Coloured images in CIELAB colour space are processed, in the form of clusters, where each cluster is defined by five parameter values, including l_v, a_v, b_v, x_v and y_v, where l represents lab, a and b are colour channel value of each pixel and x and y represent spatial location of each pixel. A grid step *S* is defined at the start, to divide the image into different region windows, which are later updated into segments, based on cluster centre and pixel updates. Initially, all pixels are assigned a fixed label value -1, while a distance matrix of size equal to number of image pixels is created, as ∞ value is assigned to each distance matrix location. Next, each cluster centre is processed, one by one, and the distance, between cluster centre and pixel location, is calculated for *2S x 2S* region, around each cluster centre. Next, if the distance calculated is less than the distance value, already assigned to each pixel, then the distance value of pixel is updated by the newly calculated distance value, while pixel label value is updated by cluster number value. During this step, all the pixels are assigned to their nearest clusters. Once a round of processing all clusters is completed, new cluster centres are calculated, based on updated pixel cluster labels, by calculating mean vector of all pixels, inside a cluster, and the error is calculated, in order to keep track of end condition of loops, specified by an error threshold value. The processing and updating of cluster centres continue until a specific error limit is achieved. Next, a post-processing step follows, where all disjoint pixels are associated with the nearby clusters, to maintain the connectivity of regions. In the end, a matrix similar to the size of the image is obtained, containing cluster label for each pixel, specifying the segmented object count in the image.

The distance values being calculated, during processing, have some issues because of the processing of superpixels in CIELAB colour space, since each pixel vector is composed of three colour channel values and two spatial location values. The range of colour values is well determined, but the range of spatial locations can vary from image to image, because a small image can have fewer pixels, while a big image will have more pixels, which can affect the overall distance value, depending on five parameter values. To deal with this issue, a normalisation step is applied, for colour and spatial distance parameters, where both distances, dc and ds, are divided by maximum colour and maximum spatial distance, Nc and Ns, inside each cluster, while then both distances are combined to create an overall distance formula, as shown in Equation (2.1).

$$
\begin{aligned}
d_c &= \sqrt{\left(l_j - l_i\right)^2 + \left(a_j - a_i\right)^2 + \left(b_j - b_i\right)^2}, \\
d_s &= \sqrt{\left(X_j - X_i\right)^2 + \left(Y_j - Y_i\right)^2}, \\
D' &= \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2}.
\end{aligned}
\tag{2.1}
$$

Defining maximum colour distance value is not obvious, because it can vary from image to image, that is why a fixed value m is determined as maximum colour distance, as shown in Equation (2.2).

$$D' = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2} \qquad (2.2)$$

Equation 2.2 is simplified to the form in Equation (2.3), which is normally used in processing.

$$D = \sqrt{{d_c}^2 + \left(\frac{d_s}{S}\right)^2 m^2} \qquad (2.3)$$

Here, the value of m determines the weight of spatial and colour parameters. If m is very high, then spatial parameters are weighed more, meaning the shape of segments is more critical, while in case of low m value, shape and size of segments are less regular. Thus, a value of m from the range [1,40] is selected, in case of coloured images [37].

2.3.2 Features Extraction Step

The selection for the input data, particularly the definition of relevant features, is an essential setting for the classification process. Actually, there are some relevant and significant features, for each class, that need to be taken into account. However, if insignificant features are included in the classification phase, the results obtained will probably not be as accurate and precise. This is how the unnecessary features affect and discredit the relevant features, leading to erroneous classification. Feature selection plays an imperative role in designating the desired classification phase features. Nevertheless, distinguishing the significance of extracted features is challenging, as they are generally undetectable to the naked eye [58].

From a general perspective, visual features are classified into low and high level features. The low level kind of features represent information like colour, texture, and shape of objects, while the high level ones are usually extracted, based on the type of application. For any given feature, there are several types of information, which can be used to represent the feature from different perspectives [59] [60]. Also, different forms of an image can be considered, when performing a comparison operation, which inherently results in different types of comparison criteria. For instance, one could be interested in images with similar colours, or distribution of colours, or images containing similar objects. The comparison, in this case, is not performed on the image directly, but rather on the features, extracted from the image, represented in vector form [61].

Reducing the number of resources, required to determine a massive set of data, is part of the feature extraction step. The number of variables to be included, when carrying out an analysis of complex data, proves to be one of significant complications. The two drawbacks, when dealing with too many variables, are usually the high demands in memory and computation power usage, as well as the overfitting of classifiers in training data sets and poor generalisation over new samples. Extraction of features usually refers to the approach of developing combinations of variables, to overcome the issue of too much available data, while simultaneously representing this data more accurately [62].

Feature extraction plays a vital role in the domain of object recognition systems. It can be performed by several techniques, in numerous fields, including machine learning, image processing and pattern recognition etc., which has resulted in a recent high volume of studies in this particular area of feature extraction. A method for integrating multiple features extraction methods, for pixel-based texture classification, was proposed in [63]. Also, various analyses, targeted at supporting texture classification and retrieval were presented in [64], using some perceptual features, for perceptive visual texture classification and retrieval. In [65], a technique for classifying rock, using both textural and spectral features, was proposed, while in [66] a method for feature extraction, based on the spectral histogram, was demonstrated. In [67], an approach was proposed, for representing features in a wavelet domain, for automatic texture segmentation. In addition, in [68] an approach to image retrieval was suggested, based on features, derived through the mean and variation of the Gabor filtered image.

2.3.2.1 Features Models

This section will discuss three models of features, since these systems are the most commonly used in visual features [69].

2.3.2.1.1 RGB Colour Space-Based Features

Colour is a widely used important feature in image and scene analysis [70]. The Commission of International de l'Eclairage (CIE), in year 1931, presented a standardisation of primary colours with wavelengths: R (Red)= 700 nm, G (Green)= 346.1 nm, B (Blue)= 435.8 nm, which are considered as the basis of colour monitors. This definition, inherently, makes RGB colour space, the standard for image storage and computer graphics [71] [72]. One of the most interesting properties of colour space representation is that, colour-based features can be extracted from an image, with less complexity and computational cost. Besides, they are usually invariant to rotation, scaling, fuzziness, and photometric transformations [73]. RGB colours are generally considered as primary colours, while they are additive, because new colours can be derived from a different mode of combination of the three bands [74].

2.3.2.1.2 HSV Colour Space-Based Features

Most digital images are encoded in the RGB colour space. However, the spatial structure does not satisfy the human vision, in a subjective definition of colour similarity. Therefore, it is common to convert RGB to HSV (Hue Saturation Value) space, which is the closest to the human eye, based on subjective perception [75]. The conversion expressions, from RGB to HSV, are described in [73]. As a result, HSV can readily be considered as an alternative to the RGB colour space. Rather than assessing the values of the RGB bands separately, a metric representing the amount of hue, each band is composed of, has been suggested [76]. Hue is simply a representation of the colour type, such as red or green, while Saturation describes how colourful a part of an image is, with respect to its brightness; the Value denotes the lightness or luminance of colour [77] [78].

2.3.2.1.3 Texture-Based Features

Texture features are mainly composed of valuable information about surface structures and their relationship with the surrounding environment [74]. More specifically, texture features, embed important information about the arrangement of the structure of a surface and its neighbouring pixels, which regional intensity does not sufficiently describe. Actually, texture can be regarded as homogenous patterns or pixel spatial arrangements that cannot be adequately described by regional intensity or colour alone. Also, texture offers a description of the properties of various real-world images, like fabrics, clouds, bricks and trees [67]. Furthermore, texture analysis has been extensively used to classify images, captured through remote sensing, as well as to classify land use, where homogeneous regions include different types of land (such as wheat, water bodies, urban areas, etc.) [79].

2.3.2.2 Box Plot Graph

Various graphs and plots, such as box-and-whiskers diagrams (or plot boxes), have been designed to visually summarise data and trends. Box plot is a simple method in descriptive statistics that graphically depict numerical data groups, through their quartiles, instead of parametric indices. A box plot, which is also regarded as a box and whisker illustration, can be defined as a visual illustration of the univariate sample's key features [80]. A rectangle, which extends from lower quartile to upper quartile, is drawn, dividing the "box" into equal halves, while lines ("whiskers") are drawn to extreme values, from the ends of the box [81].

The box plot representation is a simple way of comparing many different class samples, in the form of a single plot, which is not easy to do, by using a histogram plot of data. Samples of individual classes can be displayed in the form of boxes, side by side, by using the same scale for the representation of all data samples. This graphic representation makes it feasible to compare the nature of feature values, in different class samples [82]. Figure 2.1 shows a box plot example, representing samples of 4 different classes, where the mid line, in each box, represent median value of samples of that feature, while the boundary of box distribution represents the middle half samples of data. Box plots, shown in Figure 2.1, have similar centre/median value for samples, which exceed the median of Box 4. Box 3 samples have more variability, in sample values, compared to the other 3 Box samples. Box 2, 3 and 4 seem to be symmetric, while Box 1 is skewed upwards. Also, it should be noted that there are no sample outliers, in these box plots. The box plots that do not overlap with each other, in terms of median lines or box area boundary, without too many outliers, are considered as data samples of good quality [83]. Several studies have used the box plot technique, to show a simple summary of the features and demonstrate comparative results [84-86].

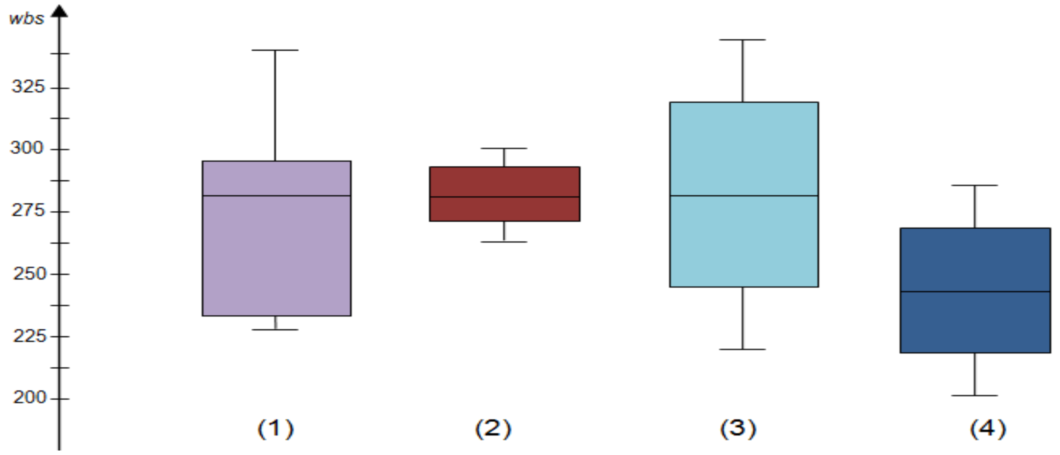


Figure 2.1: An example Box Plot showing samples of 4 individual classes, where *wbs,* which is y-axis label, is a scale of wellbeing at school site [87].

2.4 Remotely Sensed Urban Land-cover Classification

Land-cover image classification is a challenging problem, due to many attributes, like landscape's complexity, remote sensing data and image processing. Therefore, classification methods that deal with these challenges have a major impact on the success of this process. The purpose of classification systems, in remote sensing, is to detect and classify the geographical elements, on the Earth's surface. This is useful in a plethora of real-world applications, such as land use/cover mapping, urban planning, agriculture and geology, etc. [88].

The classification phase implies a process, where the objects are grouped into categories, based on their properties, for some specific purpose. It is about splitting of multi-spectral feature space into multiple categories (classes, region, cluster or entities), based on prior knowledge, concerning the identities and some statistics related attributes of the classes. The selection of a robust and efficient classification method plays a crucial role in obtaining highly accurate results, especially when faced with high and low intra- and inter-class variability. The goal of such a taxonomy is to segregate the image element, whose real class membership is unknown, into one of the expected classes [89].

The output of image segmentation, followed by feature extraction, serves usually as an input to higher-level image processing, such as classification, which is the case of the current work. However, some image classification approaches may be more appropriate than others, in distinguishing human-made categories, particularly when classifying high spatial resolution imagery, for urban environments [90].

Since many approaches are known for implementing data classification, these can be typed into three instances. The most common two types of learning are Supervised classifications and Unsupervised classifications, while the less known type is Hybrid classification [91]. Each type has its requirements, methods and algorithms that comprise the functionality and consistency of the classifier, in terms of addressing user needs. For each instance, the classification aims at assigning, foremost, a suitable class label, serving to remotely sense the images, according to the region or the pixel. There is a corresponding class for each label, with its own properties. The assigning process is implemented via an algorithm, known as the classifier. Regardless of whether or not it is supervised, the classifier is able to extract specific features, from the data, while, in turn, selects the labels of interest [58].

Supervised classification is the process, where multi-spectral feature spaces are grouped into categories (classes or entities), according to prior knowledge [92], based on identities and statistical properties of classes. This type of classification uses the already available data of classes for training, while in the next step, the trained system predicts the labels and classes of unknown samples.

Unsupervised classification of remotely sensed data refers to the division of multi-spectral sets of features, into different clusters, based on a fundamental similarity between pattern vectors [93]. This type of classification proves to be of great significance, especially in conditions where prior knowledge (i.e., ground truth) of class identity and characteristics is not available.

Hybrid classification, employed in the scope of remote sensing data, refers to a scheme that is simultaneously based on using both supervised and unsupervised classifications, in a complementary mode, to produce a unique system of classification [94]. The idea of hybrid classification was adopted, since both types of the aforementioned classification show specific limitations when applied separately.

2.4.1 Pixel and Object based Image Classification

Most machine learning classification algorithms, applied in studies, involving remote sensing, regarding surfaces material and the physical cover on the earth's surface, are along with three main research directions [95]:

1. Per-pixel algorithms, which are employed for different spatial resolution, in order to map impervious kind of surfaces, offering a kind of land cover classification.
2. Subpixel algorithms, mainly applied to medium resolution, for prediction and mapping of impervious kind of surfaces, providing a kind of surface material classification.
3. Object-based feature extraction techniques, which are largely applied to high-resolution airborne imagery, to extract man-made features, like buildings and roads.

Most digital classifications are based on a pixel-based approach (classification is done on a per-pixel level), which considers only single pixels [29]. In many datasets [96-98], even though the semantic unity of the object, under consideration, seems to work well, it is not the general case. Thus, it is also essential to take neighbourhood pixel-based methods into account. As the pixel-based techniques were developed for images of medium resolution (10-100 meters), their use on high-resolution data involves some complications. Furthermore, they are generally time-consuming, when applied to data of a higher resolution. This indicates the need to apply an entirely new method for classification, namely an object-based method [89].

Digital object-based classification (carried out on a localised group of pixels, the segments) involves collecting pixels with similar structural characteristics. These homogeneous regions are categorised, so that they fall under the correct thematic classes, based on several attributes, for analysing and sorting them into objects [99] [100]. There is a growing interest in comparing dissimilar machine learning classifiers, when applied to such objects. Factors that govern highly efficient classification, using ML methods, are proper image segmentation, training data selection, features selection and tuning [101]. These parameters have been well investigated for their impact, in past studies and research [102-104].

2.4.2 Machine Learning for Supervised Classification

Supervised classification is the learning process, during which, the objects are grouped into categories, based on their properties, for some specific purpose. It is division of a multi-spectral set of features space, in multiple categories (classes, regions, clusters or entities), based on prior knowledge about the identity and some statistical attributes of classes. Classification phase involves the mapping from input data domain to target (labels/classes) domain [105]. The present study considers multiple approaches of classification which are elaborated in the next subsections.

2.4.3 Classification Tree (CT) based Classification

Categorical datasets, a notable example of land cover classification, are used in the creation of the CT. A CT, which is a kind of Decision Tree (elaborated further in this section), comprises of a set of tree-structured decision tests, working by means of a divide-and-conquer approach. Accordingly, each leaf node has an associated class label, which is assigned to the test instances falling into this node. A predicted outcome is acquired, when a series of feature tests are conducted, which start from root and end, when a leaf node is reached [106]. CT is a supervised classification algorithm, which is based on the construction of a tree, like a set of decisions, while the testing of a new sample is done, by checking all branches of tree and then reaching on a decision node, for a prediction label [107]. An example of classification tree construction is shown in Figure 2.2. The starting node of a tree is the root node, while the ending node is called the leaf node. The nodes which are not a leaf can have maximum two nodes extended from them. A branch represents a condition for values while a node represents the result of that condition. The range of values in branches determine the characteristics of a node. In other words, a node is a point, where a decision is made (e.g., if x5<0.23154 then go through the left branch). A branch is a range-value condition, such as 0.23145<=x5<0.23154, because, after the branch, another node is reached, with another decision.

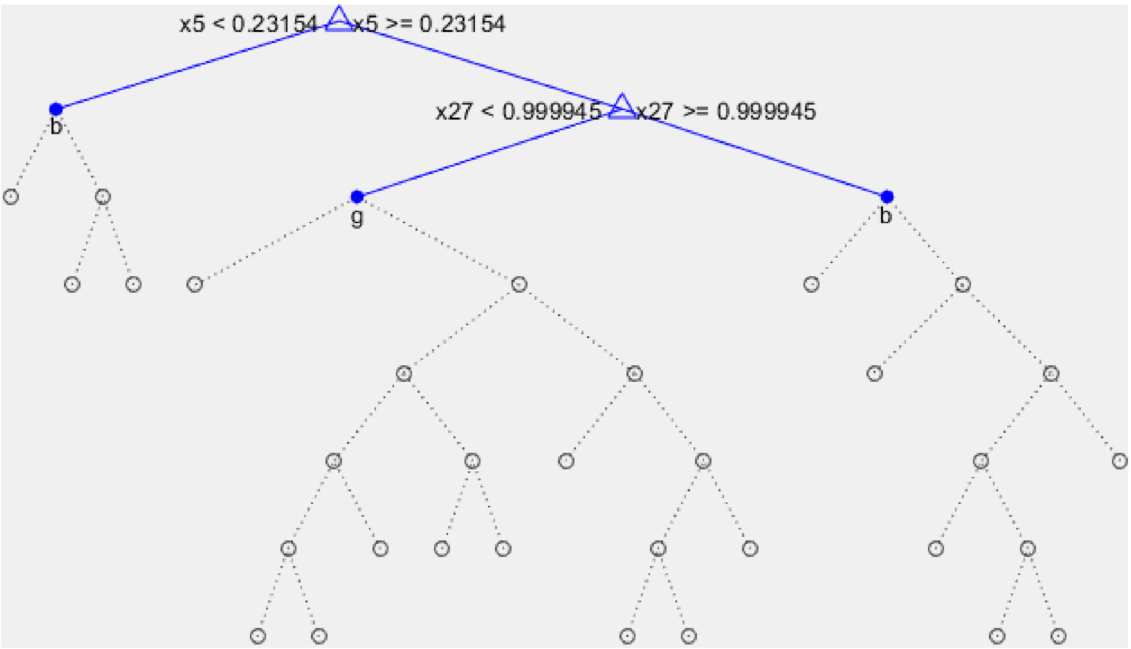


Figure 2.2: Example of creating a classification tree [108].

The feature to be tested at first on the root node is the first question, when constructing a tree. Therefore, each of the attributes is evaluated, using a statistical hypothesis test, based on entropy and information gain values, to select the one, which alone can perform well in the training samples classification. This best attribute is selected to be used as a root node of the tree. If there is another similar image, instead of the image being used, then only the feature values (i.e., the pixel values) may vary, depending on the colour distribution in the image, while the tree remains constant. If the image is so different, like only a plain, single-coloured or grey-scale image, then the whole tree needs to be constructed again, with different structure, because of different number of attributes [108].

2.4.3.1 Random Forest (RF) based Classification

RF is considered as a supervised kind of machine learning method, which is created by combining multiple base classification tree classifiers, in the form of an ensemble. This ensemble algorithm uses majority voting-based decisions, to predict the labels for unseen data samples. The correlation, among base trees and the strength of base individual classification trees, determines the strength of an RF classification algorithm. Once trained, these models can be used to predict labels for unknown instances [109]. Therefore, this classification model can be considered as a trained predictive model, where training is the process of generating the tree. RF is considered to be amongst the most popular, efficient, and respected classification techniques, which stands out amongst the multitude of ensemble approaches, due to its boosting and bagging methods. The technique is based on an ensemble of tree classifiers, where a forest of classifiers is created, based on a number of growing classification trees, while then the input vector is classified by every single tree, contained within the forest. The RF method exhibits many advantages, such as nonparametric nature that is flexible, concerning the parameters that determine the classification predictions; enhanced importance of individual variables, in classification and its good performance in multisource classification problems [110]. For example, a response variable (e.g., percentage tree cover in a land-cover) is computed using the RF method, by creating many (usually hundreds) different decision trees (a forest of trees), modelling down each of the decision trees, with all the objects. The response is then calculated, by evaluating the responses from all the trees in the forest. Regarding classification, the output class label, most predicted by decision trees in the forest, is marked as a predicted class label for the corresponding object. The key to the success of RF is how it creates each of the decision trees, making up the forest [111].

2.4.3.2 Decision Tree (DT) based Image Classification

DT is a scientific model that includes multiple decisions in the form of tree branches, where each decision set gives a predicted outcome label for the data sample. Decision trees are mostly used as decision-making tools, for research analysis and strategy planning. These are also easy to learn and understand [106]. Different kernel values and functions can be considered, to design different kinds of decision trees, during the implementation and classification phase [110]. Decision Trees are effective in decision making, for the following reasons [111]:

- The problem is clearly stated, and all options are explored and tested.
- The analysis of possible consequences, from a decision, is possible.
- The quantification of possible outcome values and the probability of achievement of those outcomes are provided.
- The best decision making, based on the available information and assumptions, is greatly facilitated.

Some of the commonly used decision tree kernels types include a coarse tree, medium tree and fine tree [112], which can be further adjusted for other parameter values, for the refinement of the classification process.

2.4.3.3 KNN-based Image Classification

KNN is performing the classifier training, based on the training samples provided [113]. The training samples are compared to neighbouring samples, in multiple dimensions, during the training phase. The neighbours of a sample are mostly determined, based on Euclidean or some other distance metric. Finally, the predicted label is determined for a sample, by taking votes from neighbouring samples, which are determined by the value of K, usually selected as an odd number, to avoid the tie of votes. A high value of K can cause instability and overfitting in decisions, so an appropriate value of K is selected, based on a specific application [114]. The prediction of a class label, for any unknown sample, is carried out, by collecting class labels of K nearest neighbours of that sample, while the most voted label is selected as predicted label for that sample. As a result, K is considered a vital tuning parameter of KNN classification algorithm [101] [115]. Many different kernel functions can create many kinds of KNN algorithms. Some of these kinds include fine, medium, coarse, cosine, cubic and weighted KNN [112].

2.4.3.4 Ensemble-based Image Classification

One of the most recent methods, suggested for land-cover classification of remotely sensed images, is that of ensemble methods, which is a family of algorithms, used in many data mining applications [5]. Some other terms for ensemble learning, cited in the prior art, are committee-based learning, mixtures of experts, and learning multiple classifier systems [116]. Four fundamental approaches are used for ensembles [117]: combination of multiple strategies; combination of multiple classification models; combining multiple feature subsets; and using a diverse training set. In using multiple classifiers-based ensemble methods, many diverse classification models are combined through majority voting. Every classifier, in an ensemble, gets a single vote for a result, while the output is the most voted [118]. This approach of combining classifiers, based on the majority voting principle, has been widely utilised in studies [119] [120]. However, there are various other versions of the voting principle and combinations of several machine learning classification methods, to predict new observations, that have been studied in literature [121] [122].

Different aspects of learning processing, such as features representation, architecture construction, learning algorithms, or the type of training dataset can influence the behaviour of a classification model. As such, the ensemble classification results of several classifiers usually leads to improved performance, compared to a single sophisticated classification model [123]. However, there are two costs, linked to ensemble methods, which include high memory requirement, to store the contributing classification models and long computational time, required for prediction of unknown data sample [124]. Consequently, the classifier ensemble has been extensively studied over the past few decades [125-127]. There are several well-known ensemble methods, such as Bagging, Boosting and RUS Boosting, which have been applied in diverse real-world applications [113]. However, despite the many methods of ensemble creations, there is still no clear evidence of which ensemble method is best, because the selection of best classification model depends on different parameters, related to the type of problem and data properties, while it is performed by applying multiple classification algorithms on a specific dataset. Therefore, the best performing classification model is selected as best model for that specific scenario and data type. Figure 2.3 shows a typical ensemble architecture, which contains a few model learners (generated from training data) and model combination. Different learned *n* models are created from *x* training data, while a combination of these learned models gives a single ensemble model *y*.

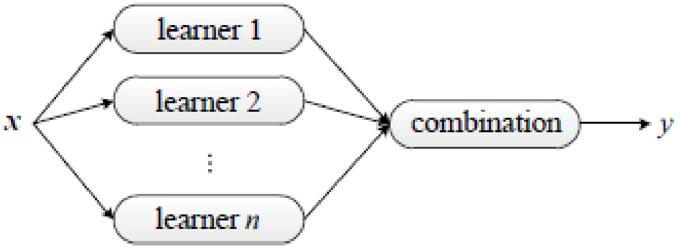


Figure 2.3: A simple ensemble architecture to combine multiple learners into one [128].

In an ensemble system, the generalisation error is decided by calculating the error among individual classification model results and the diversity among them [129]. Diversity is considered an essential characteristic, measuring the suitability of classifier combinations, for successful classifier ensembles and identifying the best classifiers to be included in an ensemble. There is no fixed definition nor method for the calculation of diversity score, in the known literature. There are various kinds of statistics, proposed by researchers, for the assessment of similarity between two classifiers, which mainly belong to one of two classes: 1) Pairwise, and 2) non-pairwise diversity measurements. The first class considers only two classifiers, at a time, while the second one considers more than two classifiers, at a time [130]. Ensembles of diverse classifiers allow higher accuracy achievement, that is often not case of a single model. Nevertheless, the optimisation of an ensemble is highly dependent on the diversity of the classifiers participating in a combination process [123]. However, diversity itself needs to achieve the right balance with the average accuracy term, to reach optimal performance on a dataset, improving the overall ensemble accuracy [131]. For example, removing the variance error from individual learners, through determining the optimal weights of objects, for a combined decision [132] [133].

This section presents many research studies, for classification approaches, used in classification and segmentation of similar area of research (i.e., urban land-cover remotely sensed classification). The comparison between different traditional classification methodologies, for the improvement of the performance of conventional classification algorithms, presented in the prior art, such as diversity, weighting and ensemble ideas, provide many ideas to be utilised in this research, to achieve best possible results. It is noted that the results, presented in the cited studies, show improved performance for ensemble classification algorithms, compared to conventional individual classification algorithms.

In [123], a new ensemble classifier, integrating diversity and weighting with the base classifiers, is proposed to create a more reliable classification algorithm. The work mainly focused on developing an ensemble, which otherwise optimises the weight of the model, to combine several base classifiers. The results, achieved in the experiments, demonstrated that this approach consistently performs better than classic ensemble methods, such as Bagging.

The work in [129] introduced a new weighted ensemble classification technique, which uses quadratic formulation. The technique mainly utilises the ensemble error, to derive the optimal weight vector of the classifier, rather than assessing accuracy and diversity. Besides, the results, attained in the experiments, showed better performance for the ensemble, when compared to other fusion methods.

In [134], it was investigated the influence of confidence (i.e., more accurate predictions), gained through base classifiers' classification on ensemble learning. This issue is approached from two different perspectives: one aspect is that of learning the weights of the base classifier, using the optimisation of the margin distribution, while the other utilised a weighted voting method. The study performed a comparison of the proposed methods to classic algorithms, while experimental results showed that weighted voting is more suited for assessing classifier confidence.

The study in [135] proposed a local learning and diversity-based ensemble feature weighting algorithm. The work used sample complexity assessment, for the evaluation of the proposed methods. Through several experiments, performed on different kinds of real-world data sets, it was discovered that designed ensemble performs better than other ensembles, as well as other stable feature selection strategies (such as sample weighting).

In [136], a method for optimising the ensemble selection task, using ensemble-based measures, was presented. The study practically takes two main characteristics of ensemble measures into consideration, which are: (a) ensemble measures evaluate the quality of an ensemble, using multiple classifiers, as opposed to the quality of a single classifier; (b) the ensemble selection is usually performed, using heuristic search techniques. This is achieved using weighted accuracy, ensemble-based evaluation measures, and diversity measures.

2.4.3.5 Deep Learning-based Classification

Deep learning has received growing interest in the last ten years, due to its unprecedented capability in the processing of images. Due to the availability of higher computational power and the versatility of neural networks, deep learning techniques have been applied in many fields of research, outperforming traditional machine learning methodologies. Deep neural networks are generic models that are able to model any multivariate non-linear relationship, given a sufficient number of neurons and layers. For this reason, they can be employed for classification, regression, clustering and generative processes, and they are able to process complex data, such as digital signals (audio, images, videos) [137] [138].

A popular and promising application of deep learning is semantic segmentation, which means the splitting of an image in multiple smaller homogeneous regions [139], meaning every pixel in a segmented homogeneous area is associated with the same meaning in some sense. For example, an image, representing an indoor scene, could include a chair, table, person and background, while an image, representing an outdoor scene, could include mountains, fields, beaches, roads and buildings. Semantic segmentation is a particular case of classification, in which each image pixel classifies according to the probability of each class.

Convolutional Neural Networks (CNN) are proved to be effective for image segmentation [140] [141]. Indeed, the most popular algorithms for semantic segmentation employ CNN, as this is the most suitable architecture to process images, since it is very efficient and effective [142], while it can even be employed for real-time applications [143].

More generally, CNNs are particularly suited for image processing. The most important feature of CNN is the convolutional layer: this layer convolves the input with a certain number of filters. Each filter is able to capture a specific feature (e.g., edges and corners) [144], while each time that feature is detected in the image, the filter outputs an increased value. The outputs of the filters are aggregated, to form a new representation of the input, while the more convolutional layers there are, the more abstract and complex the representation is [145]. The first layers are able to capture basic geometric features, while higher levels may model features with high-level semantics and complex shapes (e.g., faces, cars and trees). Convolutional layers are usually followed by a pooling layer with the purpose of reducing the dimensionality of the input [146] [147], and non-linear functions (sigmoid, rectified linear unit, hyperbolic tangent) to introduce the ability to model non-linear relationships. The architecture of the deep learning CNN is presented in Figure 2.4. The overall structure of a CNN consists of convolution layers, which are connected with max-pooling layers, fully connected and output layers, as explained in Table 2.1.

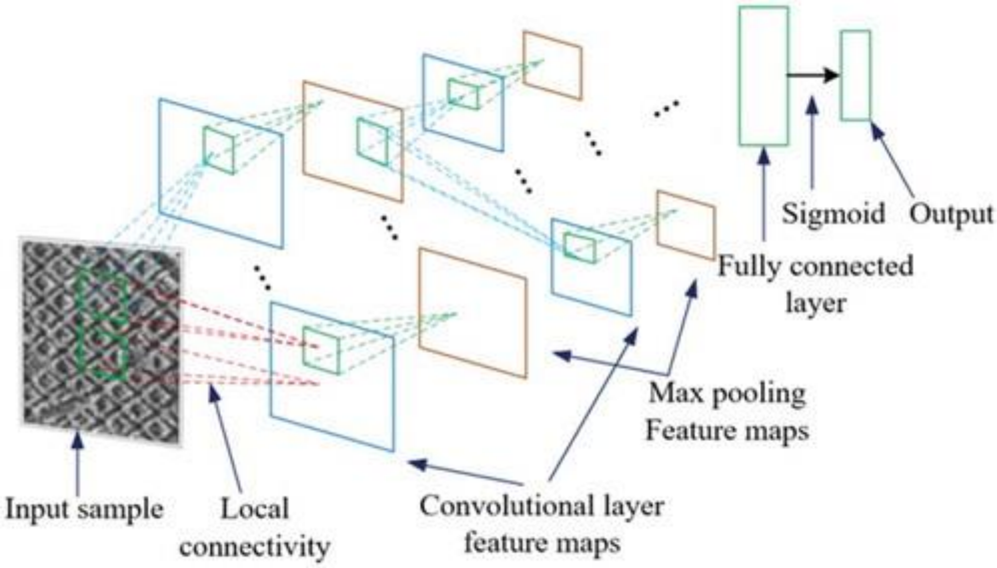


Figure 2.4: CNN architecture [148].

Table 2.1: CNN architecture layers functioning details.

Type	Terms	Explanation
Feature Learning	Input sample	Digital image of size (Width x Height x channels), where channels can be 1 (grayscale image) or 3 (RGB, BGR, etc.). Pixel values can be expressed in different formats (uint8, float16, float32, etc.)
	Convolutional layer feature maps	These layer apply a convolution operation on the input. Each layer has N kernels with specific parameters (kernel size, stride) and weights. Each kernel produces a different output, so the dimension of the output of the layer depends on N.
	Max pooling Feature maps	The max pooling operation is used for the reduction of the spatial dimension of the input layer. For each group of M pixels (where M is a fixed parameter), only the maximum value is kept. The dimension of the output depends on M.

Type	Terms	Explanation
Classification	Fully connected layer	This is typically used at the end of a convolutional neural network. The 3D input layer that is fed to the FC is flattened and its values are multiplied by the weights of the FC layer, producing an input vector whose size depends on the neurons count in this layer.
	Sigmoid	The sigmoid is one of the various non-linear operations that can be applied to the output of a layer. The sigmoid maps the values to the range [0, 1] with a non-linear operation.
	Output	The output is a collection of numbers whose dimensionality depends on the number of outputs of the FC layer.

Semantic segmentation has been applied in different scenarios, such as urban scenes [149], indoor scenes [150], outdoor scenes [151] and autonomous driving [152], whereas several studies focus on satellite images [153]. The purpose of the CNN model, in this application, is to divide the image into elements that characterise a map, such as vegetation, buildings and roads, providing a real-time application range from coverage mapping to urban planning. This task is particularly difficult, since elements of same class may show a large variation, in terms of shape, colour and texture. Moreover, it is not easy to collect a large set of data samples for the training stage. It is known that deep learning requires thousands of images, in order to achieve good performance, but building such a dataset is very time-consuming, thus limiting the application of the technique.

To produce optimal results, the training dataset should have the following characteristics, where possible: 1) **Class balance:** every class should appear in the dataset with approximately the same frequency (same number of samples/observations). For example, images with 95% volume of vegetation class and just one small building class, will result in poor classification performance, regarding the class with the fewer members. Theoretically, if some classes have a low probability, these will have a low accuracy of determination, because CNN net has poor training for this [154]. 2) **Intra-class homogeneity:**

pixels/areas, belonging to the same class, should be similar to each other. For example, if all the areas belonging to vegetation are green, in the RGB image, red trees are unlikely to be correctly classified. Similarly, if the network learns that all buildings are rectangular, a building with another shape may be assigned to another class [155]. 3) **Scale:** the images used should have the same approximate zoom level. Different sized images make it difficult to create a model [156]. 4) **Dataset size:** more images used lead to better results, particularly for CNNs [157].

2.4.3.5.1 Semantic Segmentation

Semantic segmentation was addressed before the advent of deep learning, with popular algorithms, such as watershed segmentation [158] [159], semantic texton (the elements of texture perception) forests [160], and random forest-based classifiers [161]. In satellite image segmentation, several approaches have been tried. In [162], two swarm-intelligence based global optimisation algorithms, for multilevel thresholding, were employed, obtaining good results for satellite image segmentation. In [163], the authors present a more computationally efficient algorithm, in terms of accuracy and computational time, for satellite image segmentation, based on a modified artificial bee colony.

2.4.3.5.2 Convolutional Networks

The advent of the neural network has had a considerable impact on image processing. Convolutional neural networks show excellent performance, with respect to state-of-the-art methods, both for semantic segmentation and other applications. Generally speaking, it can be said that deep learning-based methods outperform the traditional ones [164].

In 2014, fully convolutional networks [140] were shown to be able to produce dense predictions, without any fully connected layers, allowing much faster predictions for large images. Subsequent works on deep learning-based semantic segmentation followed this paradigm.

In 2015, SegNet was introduced [142]. SegNet is a fully deep convolutional network, designed for image segmentation. It is based on an encoder-decoder architecture, with a high number of convolutional layers. There are no fully connected layers, reducing the number of parameters of the network. The final layer produces a probability value, for each pixel, in the original image. An important feature in SegNet is the use of maxpooling indices in the decoder, to perform up-sampling of low-resolution features. More specifically, when maxpooling is performed in the encoder, the locations of the maximum feature value, in each pooling window, are stored and used by the decoder. As a consequence, high-frequency details are retained in the segmented images, preventing blurred boundaries, while the total number of trainable parameters, in the decoder, is reduced. The architecture is trained end-to-end, using Stochastic Gradient Descent (SGD) optimisation technique. The network is tested on several test cases, such as urban scenes and indoor scenes, obtaining impressive results.

The literature on semantic segmentation includes some works that face a problem, similar to the one presented in this study, which are used as a baseline for comparison purposes to our method. A short description of these works, mostly based on traditional methods, is provided. In [165], the authors combine multimodal data, coming from remote sensors, to model the shape of buildings and land cover. Fuzzy c-means clustering algorithms are employed. In [166] and [167], traditional classification methods, based on decision trees, are employed on aerial multispectral images. In [168], features are extracted from high-resolution aerial images and used to train pixel-based (Support Vector Data Description (SVDD), Gaussian Mixture Model (GMM), Nearest-Neighbour) and object-based classifiers (eCognition) of vegetation and urban areas. In [169], segmentation in 9 categories, from remotely sensed images, using Genetic Sequential Image Segmentation, an iterative segmentation algorithm is used, to optimise the local balance between coverage, consistency, and smoothness of each class. In [170], a combination of low-computation algorithms is employed, on aerial orthophotography and Digital Elevation Model (DEM) data, implementing a 7 class’s segmentation task. In [171], a knowledge-based system is used on multimodal data, in order to better discriminate between asphalt road, vegetation and non-vegetation.

2.5 Performance Measures for Classification

Machine learning-based applications benefit from attention given to the importance of performance measurement criteria (involving accuracy, error-rate, precision, sensitivity and specificity, etc.), used in classification [105]. The performance measurement of classification algorithms concentrates on two criteria: 1) the comparison of different classification algorithms, and 2) the ability of algorithms to be applied on a specific domain [172] [173]. One of the most common performance measurement parameters, to assess the quality of classification, is Confusion Matrix (count cases) [174]. Confusion Matrix is a table that represents the resultant data, in a comprehensive manner, in which the columns of the table represent the predicted sample count of each class, after classification, while the rows represent the actual/target sample count of each class. It is usually more intuitive to represent the prediction results data in percentages, due to the high number of sample pixels. A Confusion Matrix table is designed to depict the performance of a classifier (when the true values are known), on a set of test data, where the count of sample cases is equal to the count of the known pixels [5]. The structure of a Confusion Matrix is shown in Table 2.2. The rows of this table represent the actual class label count, for each class, while the columns represent the predicted label count of each class. The four basic terms in the Confusion Matrix table are defined as the following [175]:

- *tp* (true positives): sample cases, where the actual and predicted class labels are positive.
- *tn* (true negatives): sample cases, where the actual and predicted class labels are negative.
- *fp* (false positives): sample cases, where the actual class labels are negative, while the predicted class labels are positive.
- *fn* (false negatives): sample cases, where the actual class labels are positive, while the predicted class labels are negative.

Table 2.2: Confusion Matrix structure and its corresponding array for classification, where pos and neg are two classes, under consideration [176].

Data class	Classification as *pos*	Classification as *neg*
pos	true positive (*tp*)	false negative (*fn*)
neg	false positive (*fp*)	true negative (*tn*)

$$\begin{bmatrix} tp & fn \\ fp & tn \end{bmatrix}$$

Many performance measuring attributes can be computed from the Confusion Matrix compiled, based on the actual and predicted class samples. These attributes provide meaningful information, regarding the quality of classification models, acquired after training, as well as the behaviour of these models, in terms of future predictions. Equations (2.4)-(2.7) show the mathematical formulas, for the calculation of Accuracy, Recall, Precision and Specificity attributes of classification [176]:

Accuracy: the effectiveness measure of a trained model.

$$Accuracy = \frac{tp + tn}{tp + fn + fp + tn} \tag{2.4}$$

Recall (Sensitivity): the strength of a classifier to identify positive class labels.

$$Recall = \frac{tp}{tp + fn} \tag{2.5}$$

Precision: the agreement of class data labels, with the positive class labels, from the classifier.

$$Precision = \frac{tp}{tp + fp} \tag{2.6}$$

Specificity: the percentage of negative class samples, which are correctly classified as negative, by the classifier.

$$Specificity = \frac{tn}{tn + fp} \tag{2.7}$$

2.6 Generalisation Assessment for Unknown Data

Supervised learning methods have been investigated and implemented in several real-world applications. However, most of the existing techniques only perform well, on the basis of the assumption that the training and test data are represented with similar feature sharing characteristics, drawn from the same data distribution. In addition, the performance of these methods is strictly dependent on having well labelled and large enough training datasets, to train the model. However, the case is different in real-world applications, as the well-labelled training set is usually not available, or can only be obtained at a high cost. This challenge has, therefore, become a stumbling block for generalising learning models, applicable in real-world scenarios [177] [178].

Generalisation (also known as out-of-sample [179]), in this context, is the ability for a machine learning model to adapt properly to new, unknown entities, after experiencing learning data [159]. However, the difference in training data and testing data distributions can cause poor learning of machine learning algorithms, which can eventually affect the performance of a trained model, in unknown data testing. For that reason, it is important that both training and testing data have similar characteristics, to create a highly generalised classification model. Sometimes, generalisation performance seems to be poor, which is considered to be a result of classification models receiving bad training, while actually the model is tested in conditions and environments, previously not encountered and, therefore, not learned [180].

A generalised model provides more certainty because the classification model is well trained to deal with unknown data with a similar distribution (i.e., there is enough similar information between the testing and the training set) [181]. The major issue, linked with generalisation performance, is the overfitting in the trained model, due to the limited availability of training data, which also limits the generalisation property and ability of a model to test unknown data [182]. A training model is over fitted, when it provides high training accuracy (i.e., regarding the data used for the training of the model) and limited accuracy, when applied on unknown testing data [183]. Different machine learning studies utilise the power of generalisation and the transferability of data learning, between different image scenes [103] [184] [185].

2.7 Modelling Urban Surface Water in Remote Sensing

Remote sensing provides excellent opportunity to solve urban surface water problems, by improving the classification of land cover images, for urban drainage. The modelling of urban drainage networks is an artificial system, used to carry out surface water to a wastewater plant. Such networks assist in the estimation of runoff to model surface water management, further leading to the flooding and other environmental factors predictions. To cite an example of drainage, a natural one (river) can be found, doing the same work as the network used for artificial drainage. However, recognition of urban areas, in natural-scene images, which is an important factor in runoff modelling, is a tedious task, requiring the consideration of three related components: the selection of remotely sensed data (localisation), feature selection/extraction (image analysis) and the detection system (classification method). The selection of these factors depends on the objective of the study, variance in camera attributes, scene diversity of land cover (permeable and impermeable surfaces), variable range illumination environments, and the capacity of hardware and software [186].

To perform urban flow system simulations, it is necessary to model the flow of water, over the drainage basin, by determining the relevant drainage, rainfall, and elevations (levels) of the landscape. Regarding human-made water flow systems, successful modelling imitates and allows for the prediction of the dynamics of the temporary surface retention (ponds), illustrating the flow across the urban catchment area, along designated water flow highways [187]. This is achieved by adding drainage assets, such as manholes, sewers and other wastewater ancillaries, in combination to the human created and natural water channels that creates a model representing the real time catchment floodplain. It has also got the ability to add pumps, bridges, weirs and sluices which can create even more accurate and complete models [5].

2.7.1 Urban Flooding Phenomena

Over the last few decades, the intensity of urban flooding has increased, throughout the world, as a result of urbanisation and climate change. On a local level, urbanisation has a more significant effect than climate change, on localised urban flooding [188]. Urban flooding is a serious worldwide problem, and one of the most natural catastrophic phenomena, especially in coastal cities, where it can cause severe material and human losses. There is no way to control natural disasters, but it is possible to lower the effects of such occurrences, by flood planning. By taking the appropriate action, losses can be minimised. It is vital to conduct the right reviews, in order to be able to estimate the potential flood extent and hazards, for the various flood conditions. This will enable the correct flood and disaster management procedures, to be set up in advance [189].

The European Standard of EN 752 states 'flooding' as a state in which the surface water, either cannot go into the drains or drain water escapes out of the sewer systems, staying on the ground surface or entering into buildings. Therefore, urban drainage modelling (pipe and drainage networks) may experience flooding, at different points along the process of hydraulic surcharge, according to the style of drainage system (i.e., if separate sewer systems or combined sewers systems exist), overall drainage designing and local factors regarding the area in question. Correspondingly, hydraulic discharge, into the sewer systems, could potentially cause flood in private areas, where water enters through storm drains, where the inlet levels are below the water levels of the storm or combined sewers [190]. Aside from the outlined scenarios, as mentioned above, the possibility and effects of the earth surface flooding are more likely to be affected by localised factors and surface properties, e.g. pavements, street levelling and curb height. These attributes, however, are not easy in practice to consider, because data on specific physical features is not always available [191].

Heavy rainfall can also cause urban flooding, when the canals and city drainage systems do not have enough capacity to drain all the excess water pouring in. Runoff modelling has got three main attributes [192]:

- Volume – the rainfall on the area surface and entering in the sewer system.
- Routing – the attenuations and delays, linked to the runoff.
- Initial Losses (surface depression) – the rainfall landing during the first few millimetres and getting lost before the runoff.

The surface depression is commonly used to store the incident rainfall, which is expected to experience evaporative loss. As the depression storage is exceeded by the rainfall and the depth of the remaining water on the ground surface, at a particular time step, the excess rainfall tends to runoff, based on the volume model used. Also, water begins to percolate downwards, as the soil gets to a certain saturation threshold. A certain amount of the water that seeps through the soil, directly penetrates the sewer network, while the remaining proportion infiltrates deeper, to the groundwater storage reservoir.

Despite many centuries of floods, it is only recently that flood flows, in urban environments, are being investigated [193]. Several studies have described the effect of storage capacities in urban areas. Some researchers investigated the surge sequence, and redistribution on roads, whilst the storm is happening and the implication, in terms of flood modelling [194]. The work in [191] refers to the urban drainage modelling, between diverse surface flow and sewer flow in overloaded sewer systems, describing how it includes: single drainage areas (i.e., roofs, roads, garden areas, etc.); distinct area drainage constituents; surface level flow, which may happen, at the time of surface flooding (i.e., roads surfaces); and blocked under surface sewers, which create the sewer networks.

2.7.2 Imperviousness in the Urban Environment Remotely Sensed Images

In order to predict the behaviour of the urban runoff water, it is important to identify the permeable areas (also known as pervious or porous surfaces) and impermeable areas (also known as impervious or solid surfaces). This identification helps in deriving some indication about the amount of surface water, by differentiating between the water being absorbed in a permeable and an impermeable area.

Several digital remote sensing approaches are examined in [95], to segment and analyse impervious urban surfaces. This study concludes that pixel-based algorithms are employed for a low spatial resolution, to map impervious areas, as one kind of land use classification, while feature computation techniques are mostly used for the high-resolution airborne images, for man-made feature extractions, like buildings and roads.

A multi-resolution approach is presented in [195], for mapping surface imperviousness in urbanised areas. This study has a two-step methodology: 1) Produce details of urban maps, from high-resolution remote sensing imagery, covering all parts of the test samples. 2) Train a neural network-based subpixel classification model, to determine rainfall-runoff modelling, at the catchment level, to fill portions in the medium resolution pixels data. After detailed observation of impervious surface areas, extracted from high-resolution data and subpixel estimates, derived from medium-resolution data, it is concluded that multi-resolution data based methods can be used, instead of the expensive high-resolution mapping of impervious surfaces.

The usage of integrated Landsat Thematic Mapper (TM) and radar data, with a higher spatial resolution, is discussed in [196], for improved performance of impervious surface areas. Data fusion (wavelength) was used in this study to make spatial resolution better, while saving remotely sensed multi-spectral attributes. A high-resolution QuickBird satellite based collected data impervious surface image was utilised, as base data of Altamira city in northern Brazil, to assess the results of the impervious surface, by using TM and fusion imagery.

Mapping of impervious surface areas, by utilising high-resolution QuickBird satellite images, is investigated in [197]. Two techniques for digital classification, object-based and pixel-based, are compared, in order to determine how to derive more accurate information, relating to urban impervious structures mapping, as well as estimation in high-resolution imagery, for the Minnesota State University in the USA. A comparison between object-based and per-pixel-based classification was the first objective of this work. The study focused on noise manipulation (such as shadows in QuickBird images), in the digital classification process, as well as on how QuickBird data can optimally be used to map the impervious surface.

The classification of an impervious area, having high-resolution remote sensing images, utilising principal component analysis and image morphological operations, is performed in [198]. Two features of impervious cover were extracted, on a per-pixel basis: roads and buildings. Trees and connected canopy, in the small study area selected, have caused roads and driveways, partially shadowed, to be completely undetectable. This condition gave rise to a methodology, where multi-spectral bands (blue, green, red, and near-infrared) are mixed with the panchromatic band, using an Intensity-Hue-Saturation (IHS) transformation, in order to create a panchromatic-sharpened four-band, which could distinguish the spectral attributes of impervious area, from the tree canopies and the associated shadows.

The conventional spectral-based image classification method, to generate highly accurate maps of urban landscapes, using remotely sensed high spatial resolution imagery, is demonstrated by [199]. The subject area, Raleigh, North Carolina, USA, was obtained from the Digital Globe's image archives. Six categories of land-cover were selected, for mapping within the selected study area: human-made impervious surfaces, natural and artificial surface waters, unpaved non-vegetated surfaces, trees that have falling leaves before the winter, trees that don't have any falling leaves throughout the winter and urban grasses.

The impact of different methods, for estimating impervious surface cover on estimated peak discharges, is studied in [19]. The upper part of the Woluwe River catchment, in the south-eastern part of Brussels, Belgium is considered. Two remotely sensed data sets, with different dates, were used. The high-resolution land-cover map was obtained from high-resolution sensors, like IKONOS or QuickBird, deriving a detailed high-resolution land-cover map. Also, a medium resolution image was produced from the Landsat ETM+, which was applied for estimating land-cover class proportions, at the subpixel level. The study concluded that both high-resolution and medium-resolution images are valuable data sources, for getting improved distributed runoff prediction, in urbanised catchments, while the use of subpixel classification models, for the prediction of imperviousness from medium-resolution satellite data, maybe a useful alternative to the more costly high-resolution mapping of rainfall-runoff modelling, on a catchment scale, specifically for areas of considerable extent.

2.7.3 Hydraulic Models

Hydraulic models provide an approximate model of stormwater network performance, capturing the large-scale elements of the system; however, these systems require adjustments, according to real-world data, in order to maximise accuracy in measured outcomes. There are numerous established software applications, available for urban flood modelling (e.g., InfoWorks, Stormwater Management Model (SWMM), and MIKE-Urban), to create simulations of flows of underground pipelines and surface runoff; it can even serve to estimate inundation conditions [5]. Such applications have been widely used, successfully, in urban flood planning and management for model automation [200]. A recent modification to these hydrological models is a description of overland flow, which enables the modelling of urban flooding, to provide accurate calculations of the water depth and velocity, during the whole flood period [201].

2.7.3.1 InfoWorks ICM Model

The first fully-integrated modelling platform, including urban and river catchments, is InfoWorks ICM (Integrated Catchment Modelling). InfoWorks ICM combines the hydraulics and hydrology of both natural and human created environments, into one model [192]. ICM software was chosen, primarily based on how, within a single software package, models, for both river and sewer networks, as well as surface water flow routes, can be created. All it requires is the importing of the transportable database, into InfoWorks ICM. The files that are required, for the completion of reruns of the model, are all included [202].

There are up to twelve runoff areas, in the subcatchment table of the ICM model, although it can be assigned at least 99 runoff parameters, in the Land-use table. This means there can be hundreds of differing runoff surfaces, if this is the case. However, modellers usually keep things simple and rarely differentiate between various urban/highway surfaces, etc. The runoff area is defined as ‘absolute’ or ‘percentage’ of the contributing area, determining how much of the subcatchment belongs to the particular runoff surface type [5] [203].

The ICM model can be simulated, based on the classification results by changing the original runoff areas 1, 2, and 3 in the subcatchment table. Each subcatchment can be categorised into as many as 12 different contributing surface types (runoff areas). However, traditionally only the first three are used: Runoff Area 1 uses runoff surface 10 (impermeable (roofs)), Runoff Area 2 uses runoff surface 20 (impermeable (roads)) and Runoff Area 3 uses runoff surface 21 (permeable (for example grass)). The other 12 are called Runoff Area (1-12) Absolute, defining the area in hectares (ha) or acres (depending on which area unit is being used). The Runoff Area fields define how much of the subcatchment belongs to the particular runoff surface type, which can be assigned its own unique runoff characteristics, using various runoff models and coefficients [204].

2.7.3.2 Wallingford Procedure Model

The ICM model has been calibrated on flow survey data and/or SWW SCADA data, from over ten years ago [5]. The Wallingford PR procedure (the standard UK percentage runoff model) is the most commonly used model, while historically has been used to predict runoff from urban catchments, in the UK, although the New PR Equation [203] and even more recently the UKWIR (UK Water Industry Research) runoff models [205] are being applied, as a replacement to the Wallingford PR Model. This is used for predicting runoff from both impermeable (Areas 1 Paved and 2 Roof) and permeable (Area 3) areas, using 'Contributing Area' (the area that drains into the system, being used for modelling) and not the 'Total Area' (the full area of the subcatchment, including those parts that do not drain into the modelled system) of the subcatchment.

The Wallingford PR equation establishes the runoff coefficient, based on factors, including the type of soil, the individual catchment's antecedent wetness and the density of development, through the use of a regression equation. Predictions, attained through the model, include the overall runoff, from the total number of surfaces in the subcatchment, considering those that are impervious as well as pervious. The ongoing loss, experienced by the UK urban catchments, is typically calculated by this model, used alongside the model for initial losses, mentioned earlier. The assumption is that runoff losses generally maintain consistency, throughout the event of rainfall and are thus illustrated, using the following relationship [206] [207]:

$$PR = 0.829PIMP + 25.0SOIL + 0.078UCWI - 20.7 \qquad (2.8)$$

Where, PR shown in Equation (2.8) is the percentage of runoff; the PIMP parameter is the percentage of impermeable (the amount of paved and roofed area). In particular, this aspect represents the catchment's percentage of imperviousness, which is identified through the division of the overall, directly connected, impervious area, by the overall contributing area; UCWI is an Urban Catchment Wetness Index (antecedent precipitation index (mm)); while the SOIL factor (soil depth parameter (mm)) is based on the parameter WRAP (Winter Rain Acceptance Parameter), which is included in the Flood Studies Report (FSR) and can be collected from a revised map of soil. The value of SOIL index represents the infiltration ability (water saving limit) of the land; it depends on different properties, such as the topographic slope of the soil, the permeability of the soil and the probability of soil layers, likely to be impermeable. Five different types of SOIL index values are recognised and presented in Table 2.3.

Table 2.3: Different soil classes [204].

Soil Class Type	WRAP	Runoff	SOIL Value
1	Very high	Very low	0.15
2	High	Low	0.30
3	Moderate	Moderate	0.40
4	Low	High	0.45
5	Very low	Very high	0.50

The observation of the nature of Equation (2.8) depicts how its lower valued parameters, including PIMP, SOIL and UCWIL, might produce a lower or even a negative runoff prediction. However, the minimum and maximum values presented, in the ICM software for PR, are 20% and 100%, respectively.

The Wallingford model utilises all the surfaces, including pervious and impervious, to predict the total runoff in a subcatchment, which is why this model cannot be mixed with any other model in any subcatchment. The estimation of Runoff is comprised of Runoff contributions, from different surface areas, to a respective degree, regulated by weight coefficients. This way, all surfaces are contributing, to various degrees, towards the total runoff estimations, provided that the initial loss factors are satisfied. Weight coefficients, for all other contributing surfaces, are computed as [208]:

$$PR_i = \frac{f_i A_i}{\sum_{n=1,3} f_n A_n} . PR \qquad (2.9)$$

Where, PRi, fi and Ai depict the percentage runoff for surface i, the weighting coefficient for surface i, and the area for surface i, respectively.

The default parameter values, for the weight coefficients that are used in Equation (2.9) above, are outlined in Table 2.4.

Table 2.4: Default value of the weighting coefficients.

Weighting coefficient	Surface	Value
f_1	Paved	1.0
f_2	Roofed	1.0
f_3	Pervious	0.1

All these parameter values are calculated and utilised in Wallingford model equations, for the prediction of percentage runoff.

This chapter includes many essential aspects of this thesis, including a brief introduction to important concepts, used in the implementation, as well as a detailed literature review on all the aspects of this thesis. The work of several other researchers, in the area of remote sensing, is presented in this chapter, where positive and negative aspects of those works are highlighted, in order to determine the most effective techniques and ideas, to be used in this thesis, providing the best possible results and performance. Satellite imagery segmentation and classification, as challenging tasks, produce different results for different sets of data, available under certain limitations and conditions.

Many ideas are derived from a thorough review of the prior art, in this field. First of all, various known data acquisition techniques are studied, to determine the appropriate technique for the data collection of this research, while appropriate hardware and software tools are selected accordingly. Next, the investigation of data transformation and ground truth generation techniques sets the basis for suitable data preparation and actual ground truth data generation, for the specific data of this study. Following, different types of data segmentation and features extraction techniques are studied, in the literature, concluding that Superpixels segmentation technique, along with three types of features set, are the most fit to be used in this research. Further, various types of classification techniques are studied, to select the most suitable ones, to be used in the classification phase of this research, based on their respective positive and negative aspects. Next, a new ensemble classifier design is proposed based on diversity of classifiers and weighted ensemble used by some other researchers. Thus, conventional classification algorithms are used, in this thesis, as a new ensemble classifier called ParetoEnsemble. The results of the best performing classification model will be used as input to the modelling network, in the next phase. Finally, a detailed review and description of surface water modelling tools is included, as well as a presentation of the Wallingford modelling network, to be used in this thesis, for runoff estimations and predictions.

CHAPTER THREE

3 DATA PREPARATION AND ANALYSIS

This chapter illustrates the selection, transformation and analysis of the satellite image under consideration for this research study. The first section in this chapter explains what kind of data is needed, and from where and how this data is acquired. This section has more than one subsection elaborating details on how the image to be used in this research is captured, and how it is transformed into a usable form. Also, it explains how we have created a labelled ground truth image for the collected image. The adjusted image is used in the next section, where analysis is performed to extract useful information from it.

The following section contains a detailed explanation of what kind of image analysis methods are applied to the data image to convert it into useful information. In the first step of the image analysis, the SLIC superpixel segmentation method is applied to segment the test image into different objects. In the second step of this phase, three different kinds of feature extraction methods are applied to extract discriminating attributes from objects of different classes. Visual analysis of the extracted features is performed in this section by incorporating box plot analysis to differentiate/separate the most and least contributing features for classification. The feature datasets are compiled to be used for classification purposes in the next phase.

3.1 Image Acquisition

The study data represents the small village of Feock, Cornwall in southwest England. Figure 3.1 shows an Impermeable Area Survey (IAS), also known as a Contributing Area Survey (CAS), data which represents the map of the area of study provided by the Pell Frischmann Company [209]. This map covers a coastal rural parish area about 5 miles south of Truro city at the head of Carrick Roads on the River Fal. The 2011 National Census records the population of Feock Parish as 3,708, and the parish covers an area of 1,204 hectares [210]. Given the variation of land covers that this map includes, it provides a good test site for the detection of landscape objects for the purpose of classification of urban units (buildings, roads, parking lots, vegetation, soil, etc.), which are essential components for hydrological modelling and urban planning.

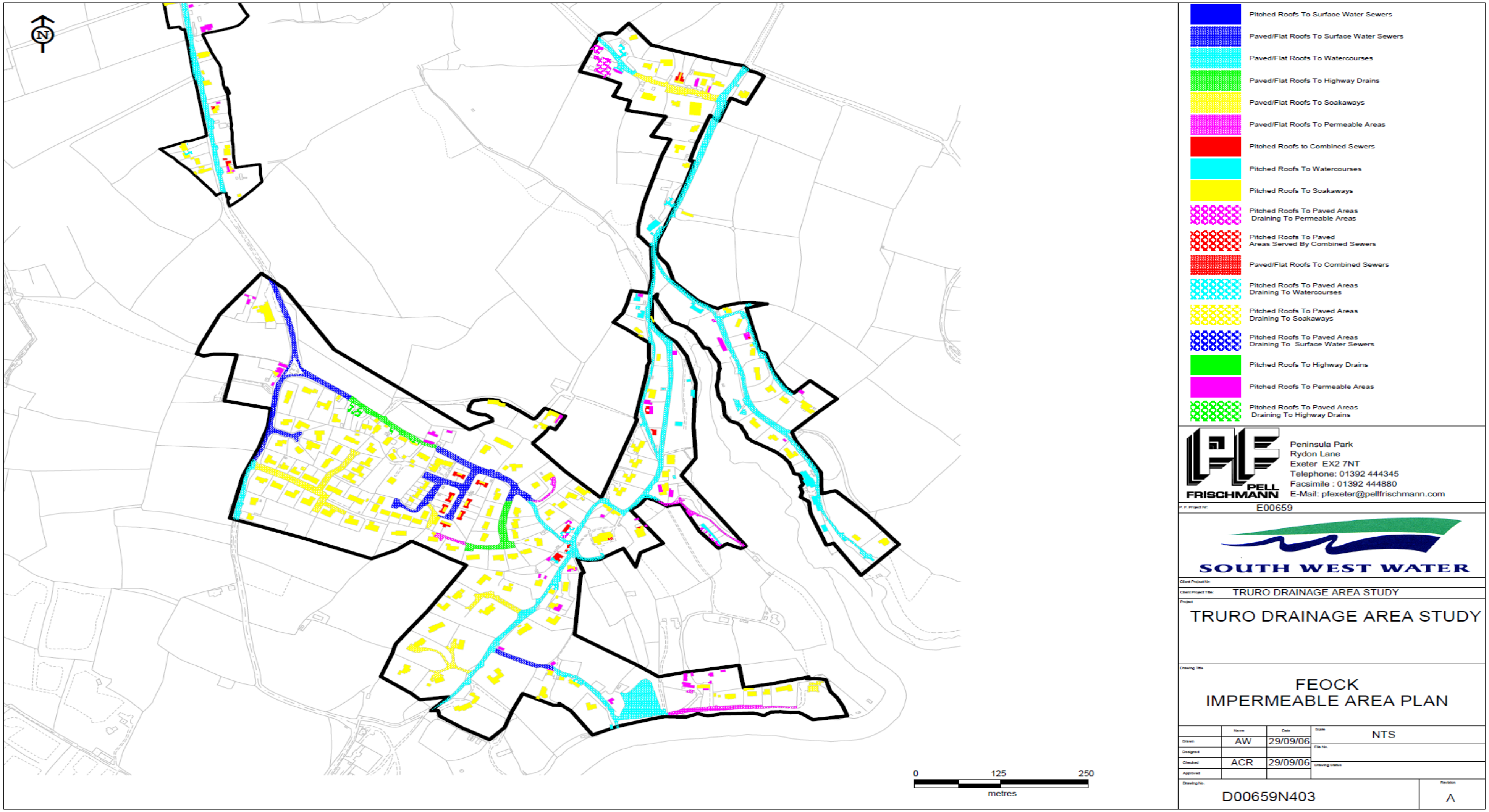

N
0
125
250
metres
Pitched Roofs To Surface Water Sewers
Paved/Flat Roofs To Surface Water Sewers
Paved/Flat Roofs To Watercourses
Paved/Flat Roofs To Highway Drains
Paved/Flat Roofs To Soakaways
Paved/Flat Roofs To Permeable Areas
Pitched Roofs to Combined Sewers
Pitched Roofs To Watercourses
Pitched Roofs To Soakaways
Pitched Roofs To Paved Areas Draining To Permeable Areas
Pitched Roofs To Paved Areas Served By Combined Sewers
Paved/Flat Roofs To Combined Sewers
Pitched Roofs To Paved Areas Draining To Watercourses
Pitched Roofs To Paved Areas Draining To Soakaways
Pitched Roofs To Paved Areas Draining To Surface Water Sewers
Pitched Roofs To Highway Drains
Pitched Roofs To Permeable Areas
Pitched Roofs To Paved Areas Draining To Highway Drains
PELL FRISCHMANN
Peninsula Park
Rydon Lane
Exeter EX2 7NT
Telephone: 01392 444345
Facsimile : 01392 444880
E-Mail: pfexeter@pellfrischmann.com
P. F. Project Nr: E00659
SOUTH WEST WATER
Client Project Nr:
Client Project Title: TRURO DRAINAGE AREA STUDY
Project
TRURO DRAINAGE AREA STUDY
Drawing Title
FEOCK
IMPERMEABLE AREA PLAN
Name
Date
Scale NTS
Drawn AW 29/09/06
Designed
File No.
Checked ACR 29/09/06
Drawing Status
Approved
Drawing No. D00659N403
Revision A

Figure 3.1: IAS data map with an urban catchment of Feock [209].

3.2 Image Preparation

In this study, some essential operations are considered for preparing the data to use in the subsequent phases. The first operation in the proposed system is the mapping of the Feock map with high-resolution satellite imagery of the area. Mapping of satellite images is done by using the online Google Maps Customizer [211] tool, which allows capturing of the satellite images from google maps. This tool provides an excellent way to acquire images of any size from google maps without necessarily assembling them manually. If we intend to capture an image larger than the computer screen, Google Map Customizer’s screen capture tool, such as Fireshot for Firefox, that can capture the whole page, proves a very useful tool. The size of the map that can be captured is mainly limited by the computer’s processing power and the network connection download speed/bandwidth. The output from this tool is a high quality 2D RGB satellite image in PNG format, according to the desired image resolution [212].

The captured satellite image of the Feock map is 7200 x 10400 pixels resolution png format image, its geographical coordinates are 50° 12' 0" North, 5° 3' 0" West. This satellite image is resized and rotated to the proper orientation to obtain a perfect match with the map, where any mismatch is observed by making the study area transparent using Microsoft PowerPoint software, as shown in Figure 3.2. The capturing details of the satellite image being used, including satellite name, north, legend and scale, are shown in Figure 3.3.

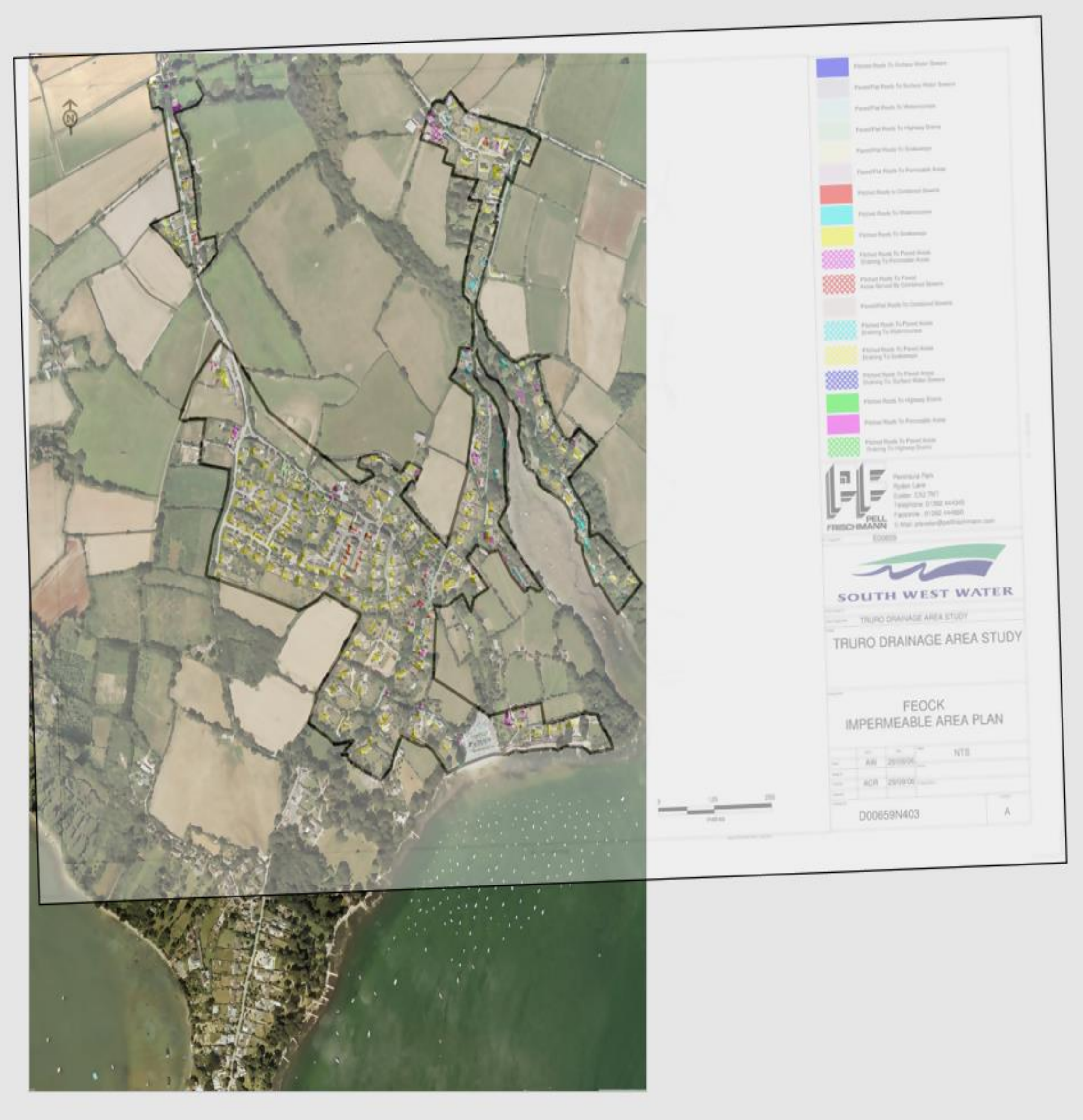


Figure 3.2: The captured real-world satellite image and the data map of Feock by using Google Maps Customizer.

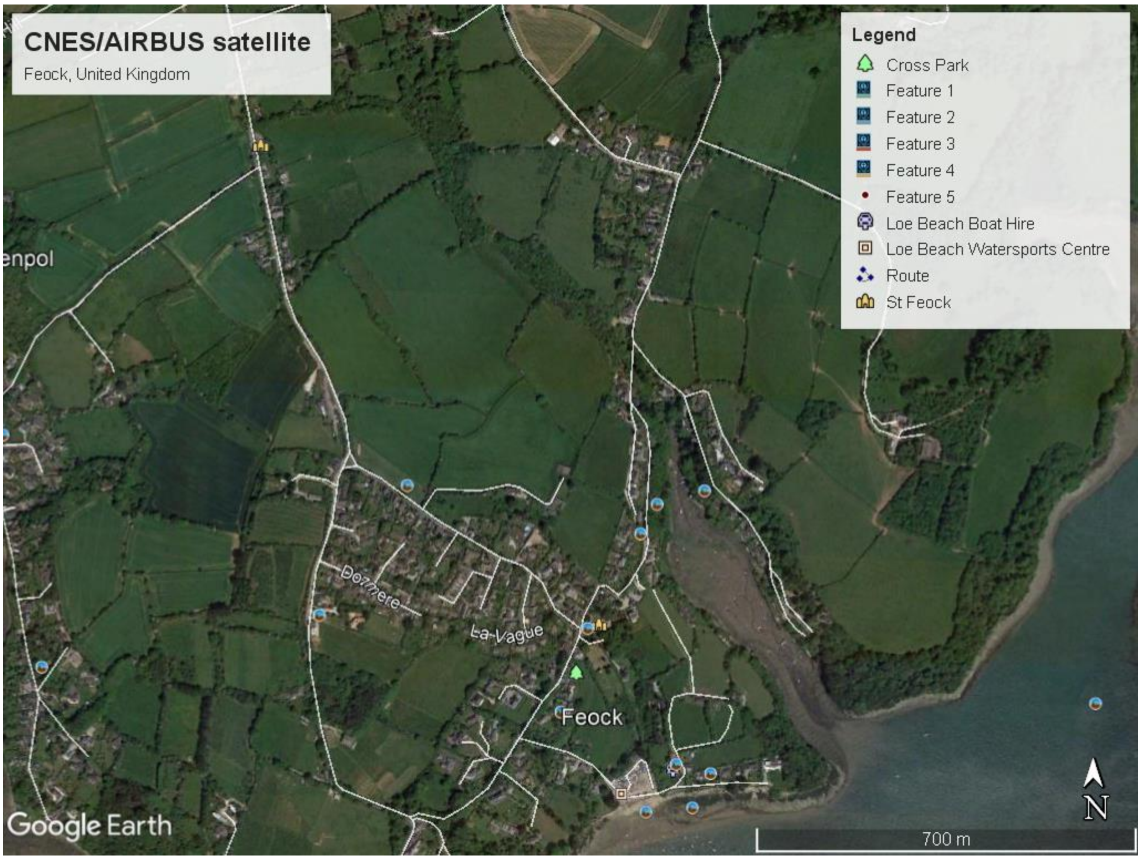
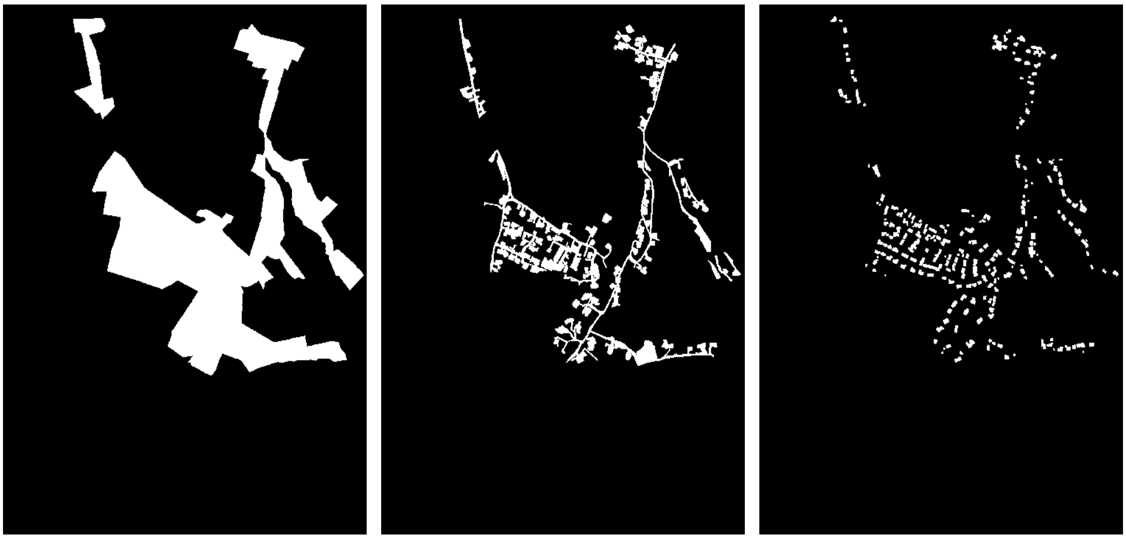


Figure 3.3: Feock satellite image showing Capturing satellite attributes.

Once this step is complete, then demarcation of three masks is performed for the areas of interest (the area of information available), as shown in Figure 3.4, by using a freeform tool for marking using the insert shapes menu in PowerPoint.



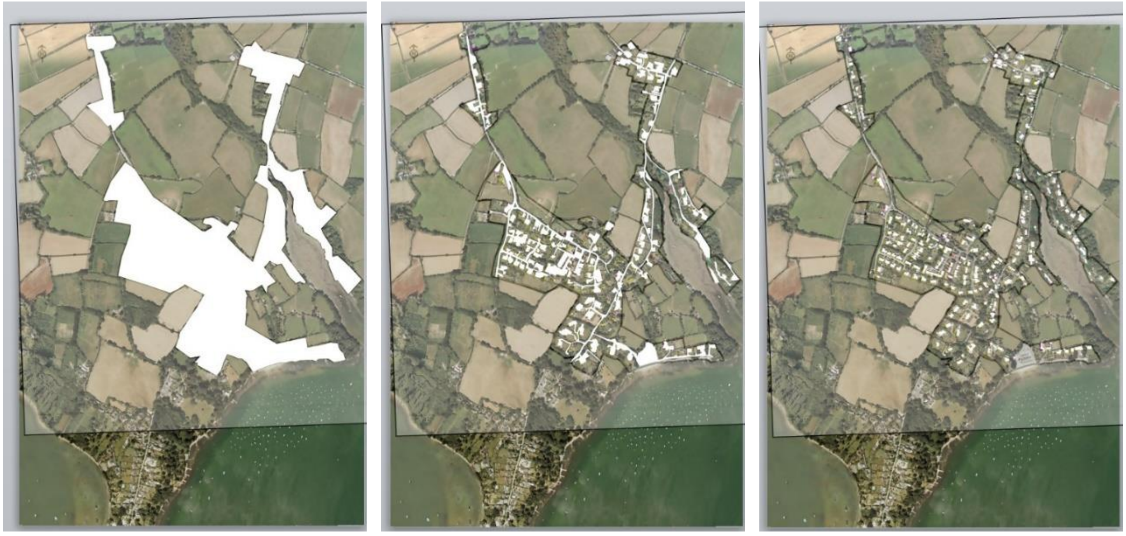


Figure 3.4: Top three images are the masks of the area of interest (black parts are not considered), and the bottom images show the mapping of the three masks on the corresponding satellite image.

However, marking all ground truth labels accurately requires a significant amount of work, especially distinguishing different objects from each other. Even at the highest resolution on google maps, it is challenging to distinguish gravel from roads, and some roofs are also indistinguishable from parking spots, as can be seen in Figure 3.5. In this specific image, a good hint to help mark such challenging ground truth labels better is that buildings have shadows which help in determining the boundary of paths around buildings. It can be noticed in Figure 3.5 that some parts of the map (top right) do not coincide precisely with the corresponding parts of the satellite image (bottom right). This can be attributed to the wide time frame between the period when the study was conducted (2006) and when the map currently being used is collected, which is very recent. Hence some features are unmatched, and some other features are also not marked. Nevertheless, it can be observed that the match is close enough to be used as a guide for the manual marking of features.

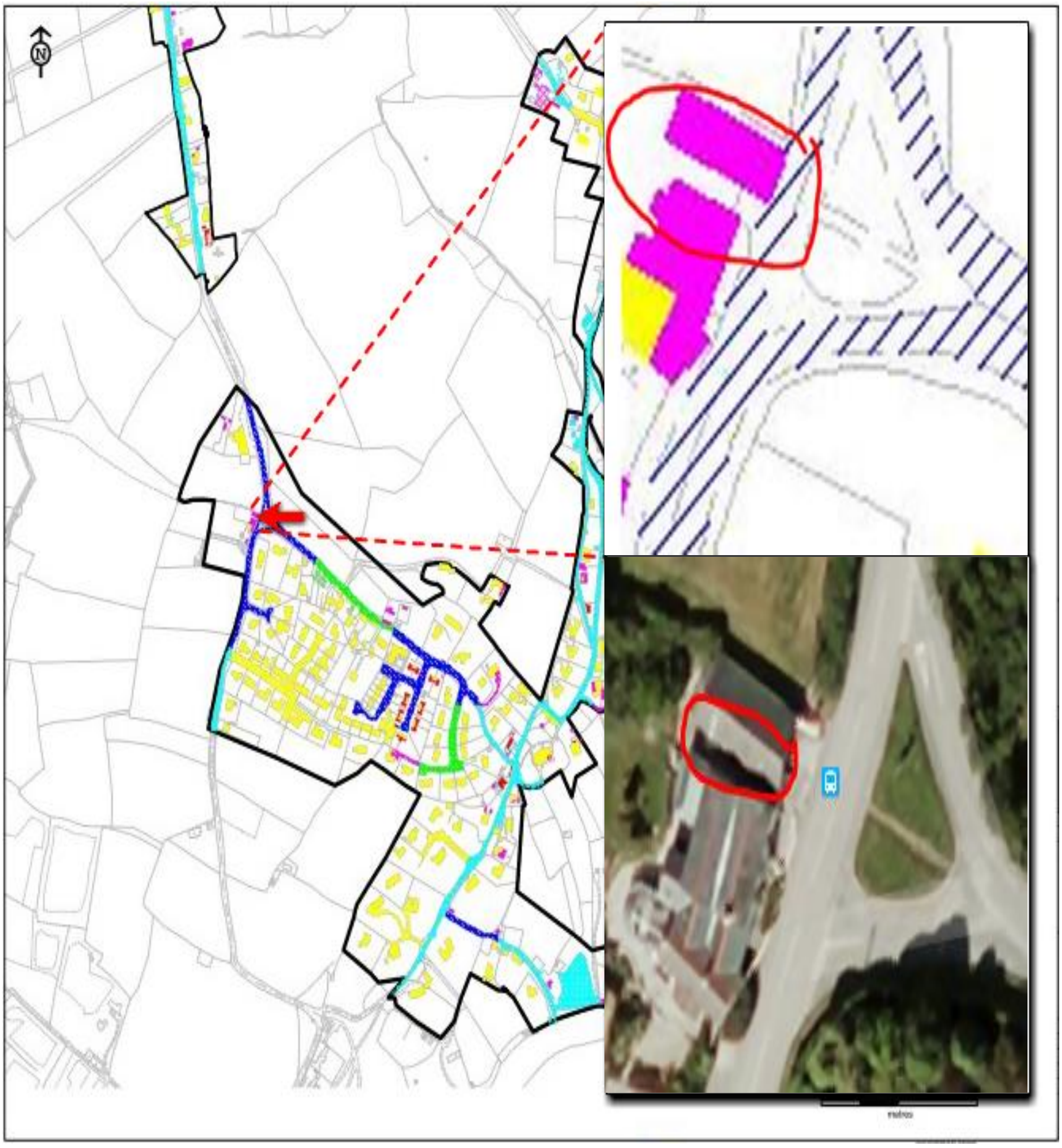


Figure 3.5: Feock map (left), and a zoomed part from the Feock map (top right) and its corresponding satellite image (bottom right).

In the current work, a supervised learning classification mode is adopted. Therefore, the pre-interpretation of the reference data (ground truth) of the given image is necessary. However, the acquisition of ground truth is often a critical issue in remote sensing, this research aims to reduce the dependency of remote sensing classification on ground truth by constructing a fully automated system to classify any satellite image. Hence three masks are combined and labelled manually to generate a single ground truth map with three colours for three classes: buildings with red, roads with blue, and vegetation with green, as depicted in Figure 3.6. Choosing these three colours simplifies the coding task because all three colours are most distant from one another, which makes it easier to distinguish different classes of pixels from a programming perspective. The black parts are ignored since the study map for those areas is not available.

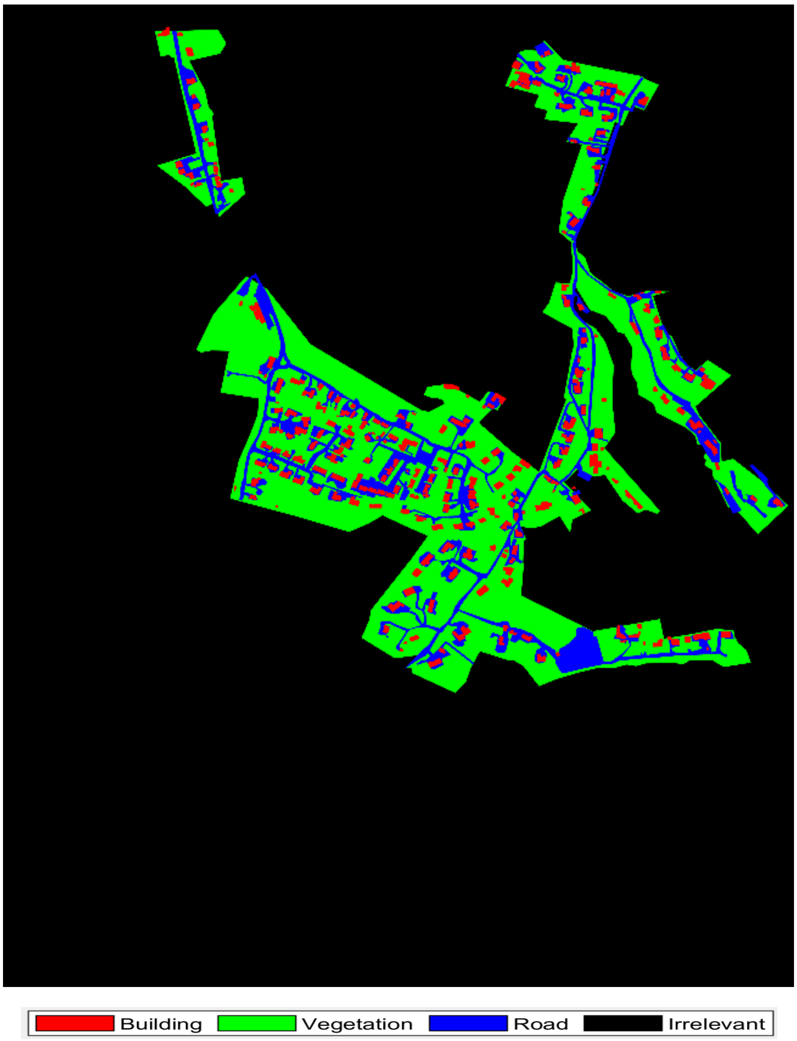


Figure 3.6: Labelled ground truth image for Feock map.

3.3 Image Analysis

As shown in Figure 1.1 (in chapter 1), the image analysis phase includes a segmentation step followed by feature extraction. These two steps are relatively difficult yet crucial since the outputs from these steps are to be used as an input to the higher-level image processing, such as the classification phase in this current work.

3.3.1 Superpixels Based Image Segmentation

This step segregates the objects of interest from its surrounding environment inside the study image. A 64-bit Windows 10 OS, with an Intel(R) Core i7 processor (2.20GHz) with 16GB of RAM has been used for the experimentations.

Image processing functionalities to segment regions and to obtain the segment attributes are applied by using MatLab R2018b. During segmentation, the image is divided into homogeneous regions of unequal size. Various discriminating RGB colour space, HSV colour space and Textural features are then computed from the segmented regions, and the most useful features for classification are selected by analysing the candidate features visually through box plot analysis.

The image analysis task of the current work involves two consecutive steps. In the first step, image segmentation is performed by using the SLIC algorithm [35], which uses superpixels of the image under processing to divide the image into different segments. SLIC segmentation is applied by using a MatLab built-in function which takes the image as input along with the desired number of SLIC objects to divide the image into. This function returns a segmented image as output having different pixel labels for different segments. The segmented image pixels are scanned one by one, and different objects are extracted based on the segmented image labels. For example, if object number 1 contains 210 pixels, that means all the 210 pixels for this object inside segmented image will have the same label (i.e., the pixels having the same label will be combined as a single object). All the objects are extracted based on unique labels in the segmented image, which are used for feature extraction in the next step. The next step extracts feature from every segment/object (where every segment has different feature values) and creates a feature dataset by using total object count in the image as the total number of instances in the feature dataset.

An example of Feock image segmentation output is shown in Figure 3.7 by applying superpixels (SLIC) based image segmentation for three different SLICs. However, the number of instances in the dataset varies based on SLIC value; for example, SLIC 10,000, when applied to Feock image, gives 9,869 object samples, which creates a dataset having 9,869 samples.

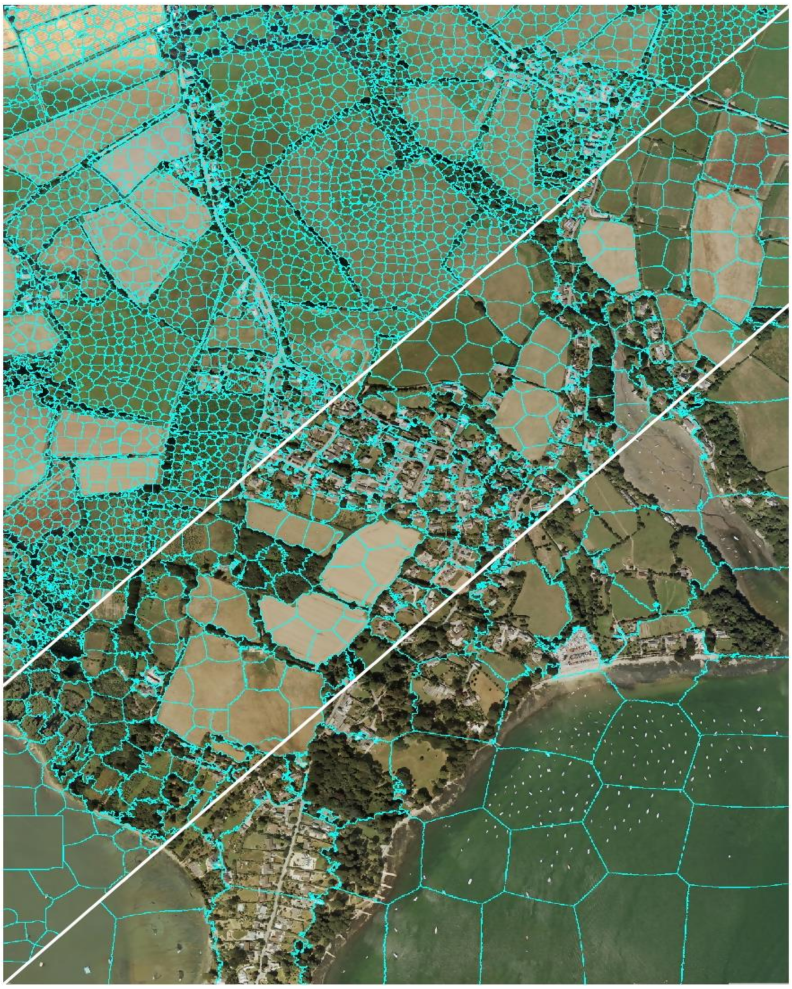


Figure 3.7: Superpixels regions boundaries overlaid on Feock satellite image for multiple SLIC.

3.3.2 Feature Extraction

The next step in the image analysis process is to extract the measurements of the most useful features for each region in the segmented image; where all extracted features are to be used in the classification task performed after this step. The choice of features is closely related to the area of study and the kind of problem being worked with. The analysis of land-cover characteristics to perform classification of land objects into three classes, i.e., buildings, vegetation and roads, is focused upon in this work. Three different kinds of feature sets are extracted in the feature extraction phase of this study, which covers the attributes which have different natures. These attributes are extracted from objects to conduct a comparison between the three types of feature sets to pick the best type for the current problem.

3.3.2.1 RGB Colour Space Based Features

Red, Green and Blue (RGB) Colour Space is the most widely used colour space for image processing and analysis related research problems [76], where most image classification related tasks are performed based on features extracted from RGB colour space. We have extracted 10 RGB colour space based potentially useful features, which can help reduce the amount of resources (memory and computation power) required to describe the test data without compromising the accuracy. The extracted RGB features are illustrated in Table 3.1, which are related to the colours and shapes of SLIC objects to differentiate different objects. This table shows the criteria used and the corresponding terms for automated image analysis [213], where F is the feature extracted from each segment. These features have been widely researched for satellite image objects discrimination, see for example [214-216]. All these feature values, along with each object label, are saved as a dataset for the training purpose of the following classification phase. There are three numeric labels, 1, 2, and 3, assigned to each class of objects, where label 1 denotes building class objects, 2 represents vegetation, and 3 is used to represent road class objects. Thus, the dataset is arranged in the form a matrix in MatLab where the feature values are in the first ten columns of the dataset matrix, and column 11 of the dataset matrix is used to store object labels.

Table 3.1: RGB Colour Space attributes.

Feature No	Features
F1	Horizontal location of the centre pixel of region centroid
F2	Vertical location of the centre pixel of region centroid
F3	Number of pixels in a region
F4	Mean colour intensity over the region ((R+G+B)/3)
F5	Region R-channel average colour measurement
F6	Region B-channel average colour measurement
F7	Region G-channel average colour measurement
F8	Region R-channel excess measurements (2R - (G + B))
F9	Region B-channel excess measurements (2B - (G + R))
F10	Region G-channel excess measurements (2G - (R + B))

The quality of the ten extracted features shown in Table 3.1 is analysed for the selected SLICs 10,000, 25,000 and 50,000 by visually inspecting the values of the features in the form of a graph box plot for each class samples, as shown in Figures 3.8-3.10. There are ten box plots presented in each figure, where each box plot represents one of the ten features. There are three boxes inside each feature plot, where the middle red line inside each box represents the median of all the samples, and each box boundary represents the middle half samples of each class. The outer boundaries of each box represent the minimum and maximum value limit for each feature, and the plus red marks represent outlier sample values for each feature. The presence of a few outliers does not impact the quality of feature and sometimes is good for the determination of decision boundary for the classification algorithm [217]. However, if there are too many outliers present in feature samples then it is not suitable for the classification algorithm, and that specific feature is considered as not a good quality attribute [9] because lots of outliers can affect the estimation of decision boundary for classifier which can eventually affect the learning and training of the classifier. Another important factor to be considered in box plots is the overlapping of red lines in the three boxes of the box plot, which represents medians of data samples for each of the three classes and boundary of boxes [217]. According to the ten feature box plots presented in Figures 3.8-3.10, it is observed that feature F9 is the least contributing features among all ten features due to the presence of the most outlier samples in the plot while other features include very few or no outliers which indicates that these features are most essential and contributing features for the learning of ML algorithms.

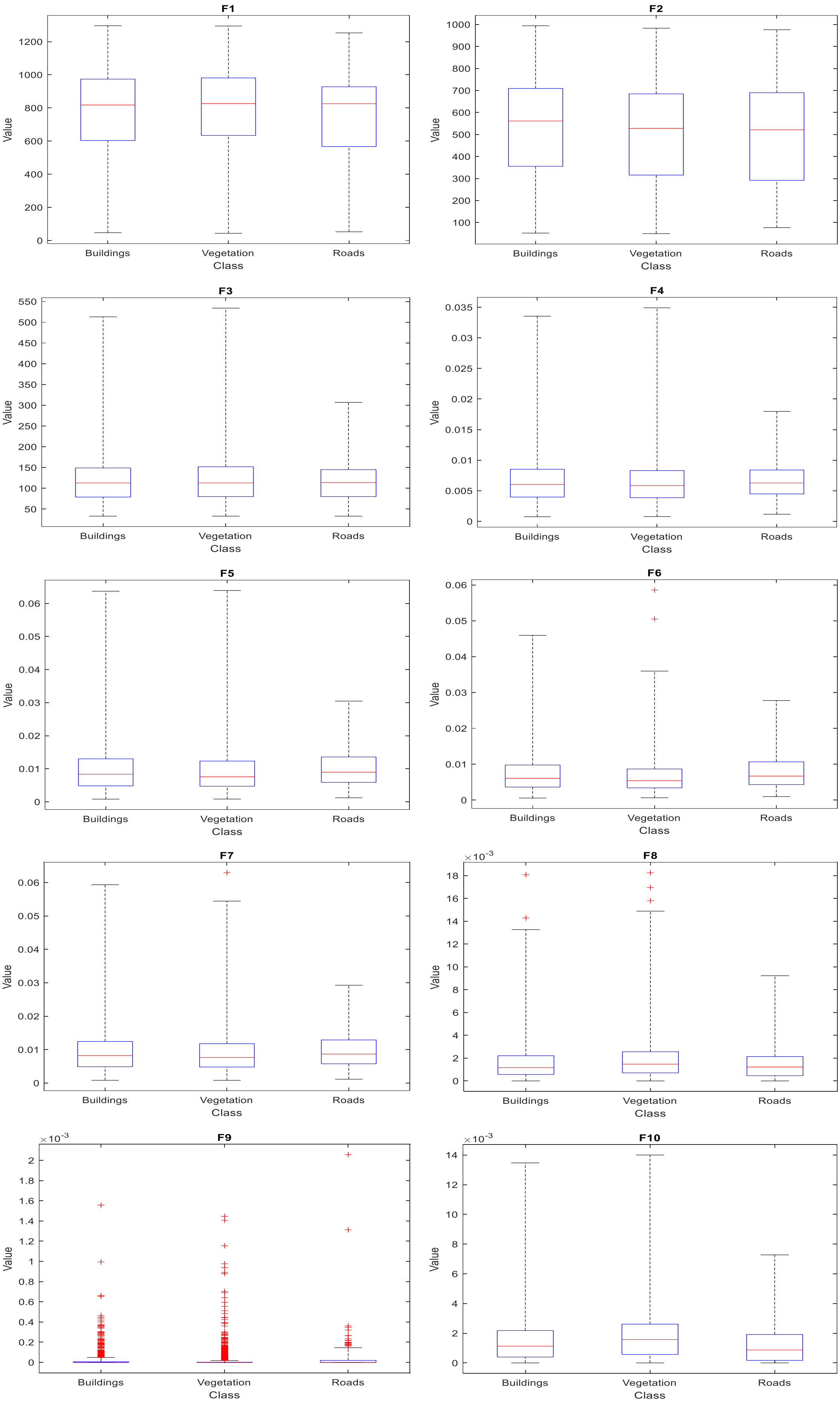


Figure 3.8: Box Plot representation of RGB features of Feock image for SLIC 10,000.

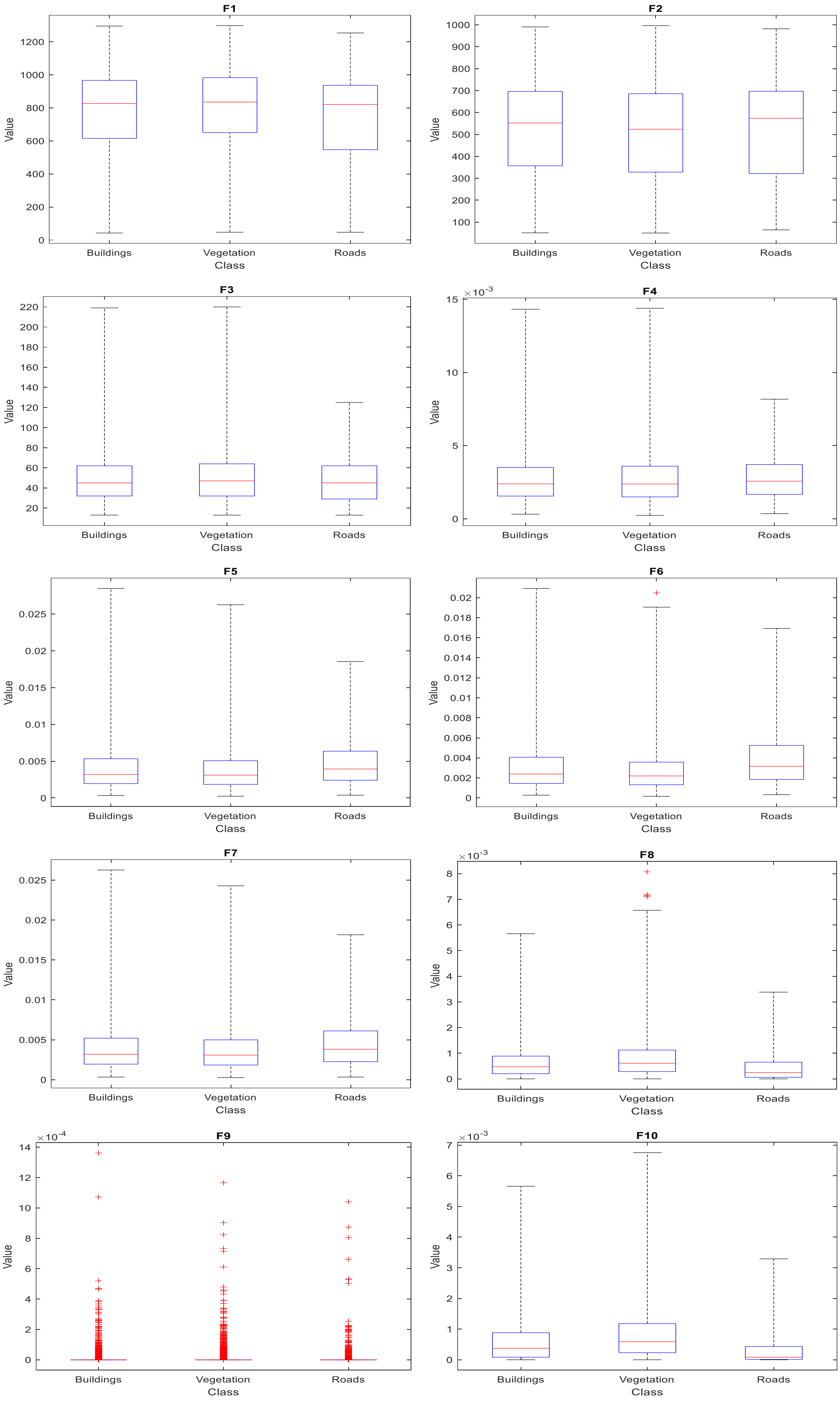


Figure 3.9: Box Plot representation of RGB features of Feock image for SLIC 25,000.

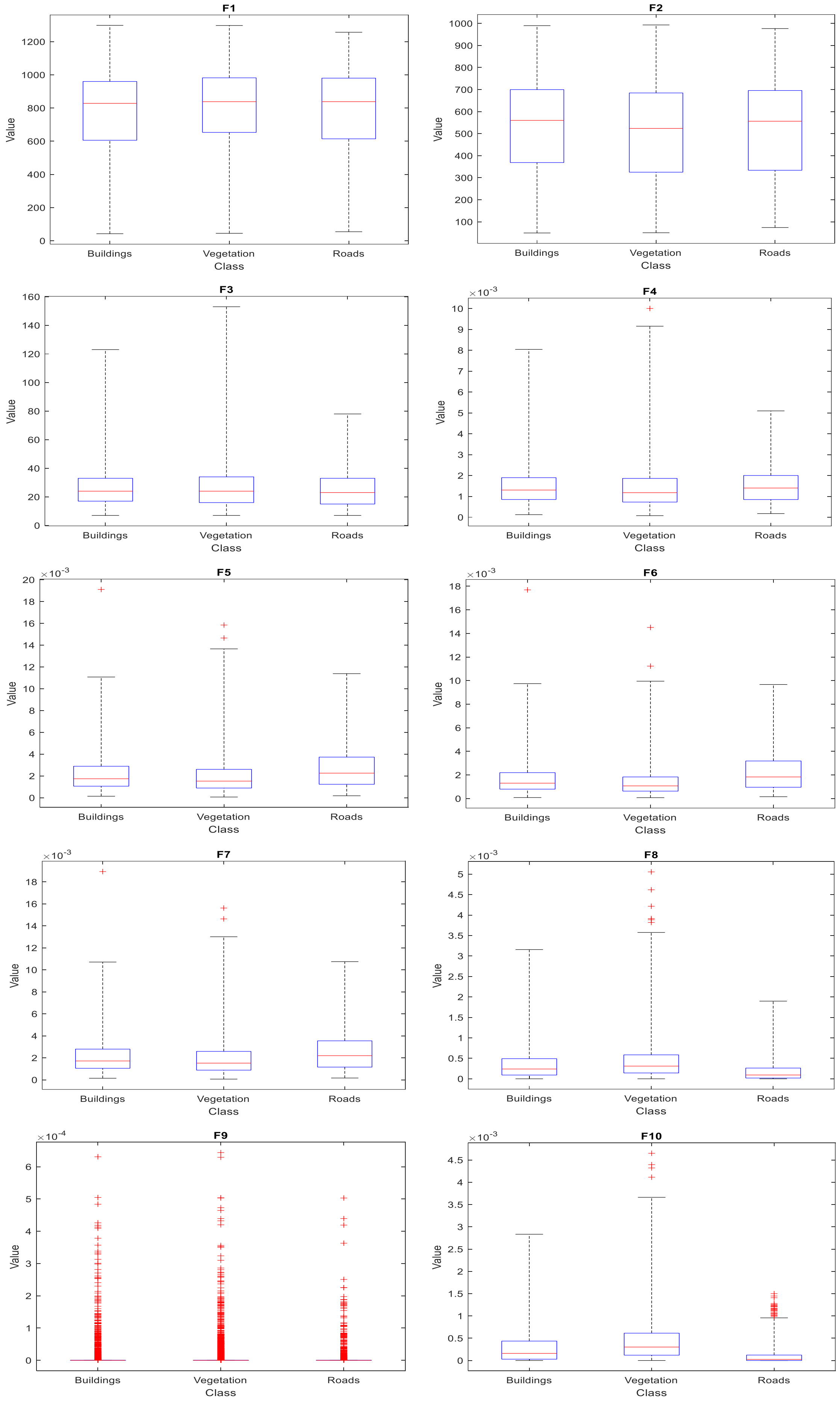


Figure 3.10: Box Plot representation of RGB features of Feock image for SLIC 50,000.

3.3.2.2 HSV Colour Space Based Features

Analysing objects in another colour space can occasionally be helpful for discrimination and classification purposes of different data samples. To analyse sample objects in another colour space, 12 Hue Saturation Value (HSV) colour space-based features (alternative to the RGB colour space [75]) are extracted from the Feock image to differentiate different classes of objects in another colour space, as mentioned in Table 3.2. The dataset created in the form of the MatLab matrix contains feature values in the first 12 columns of the dataset matrix, and column number 13 saves actual ground truth labels of objects.

Table 3.2: HSV colour space attributes.

Feature No	Features
F1	Variance of the region (H channel)
F2	Variance of the region (S channel)
F3	Variance of the region (V channel)
F4	Standard deviation of the region (H channel)
F5	Standard deviation of the region (S channel)
F6	Standard deviation of the region (V channel)
F7	Mean of the region (H channel)
F8	Mean of the region (S channel)
F9	Mean of the region (V channel)
F10	Skewness of region (H channel)
F11	Skewness of region (S channel)
F12	Skewness of region (V channel)

HSV features are also analysed visually through box plots for same SLICs as for RGB features. Figures 3.11-3.13 show box plots for the HSV features, where each box plot represents one of the 12 features. From the 12 feature box plots, it is observed that feature F1 contains too many outliers, which indicates that this feature is not useful for classification. While other features contain very few or no outliers which depict that these features contribute well for classification method decision boundary estimation. Also, there is some overlapping of features data distribution boundaries in these box plots because many of the HSV features are colour based and the visual nature of different class objects is very similar to each other in the test image.

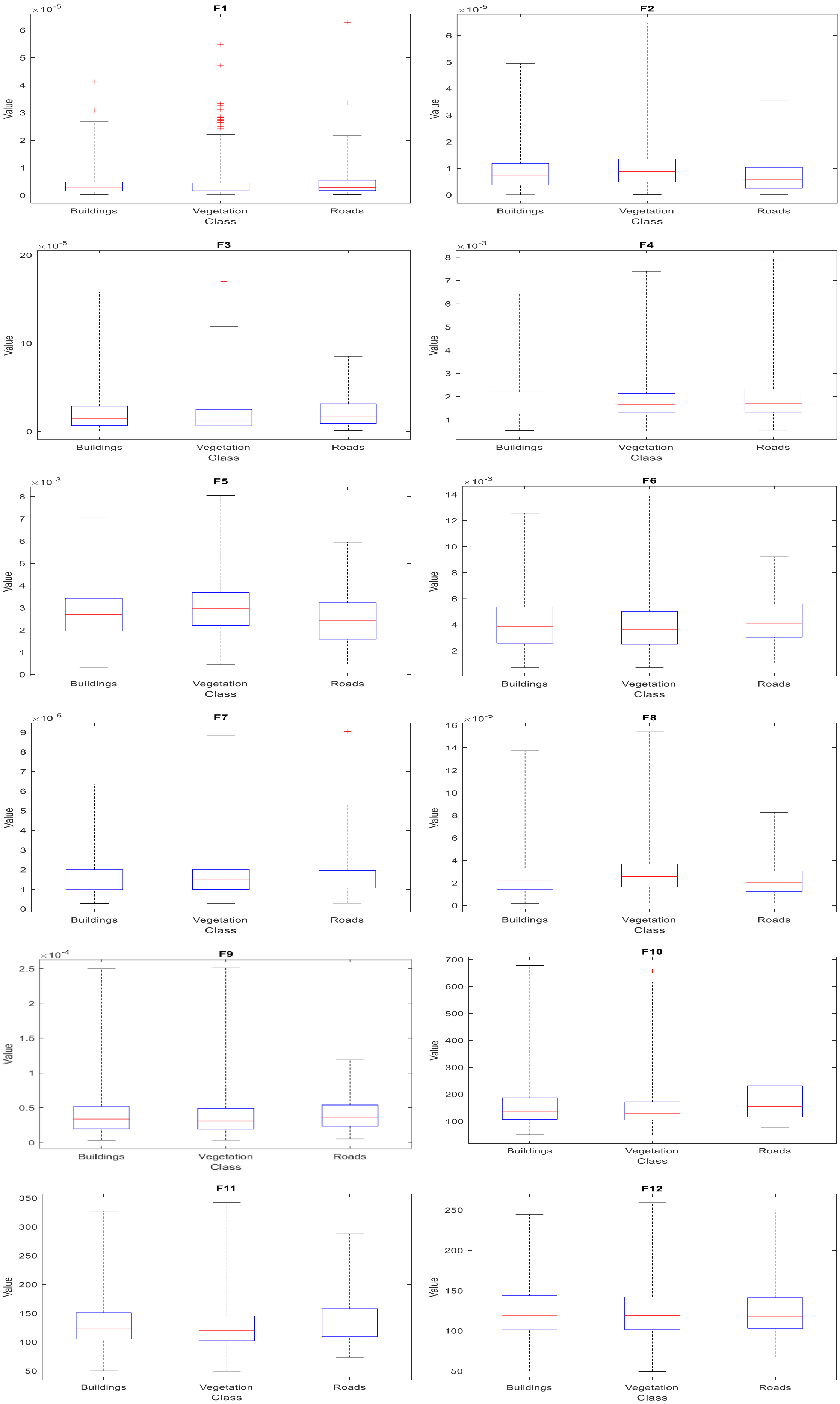


Figure 3.11: Box plot representation of HSV features of Feock image for SLIC 10,000.

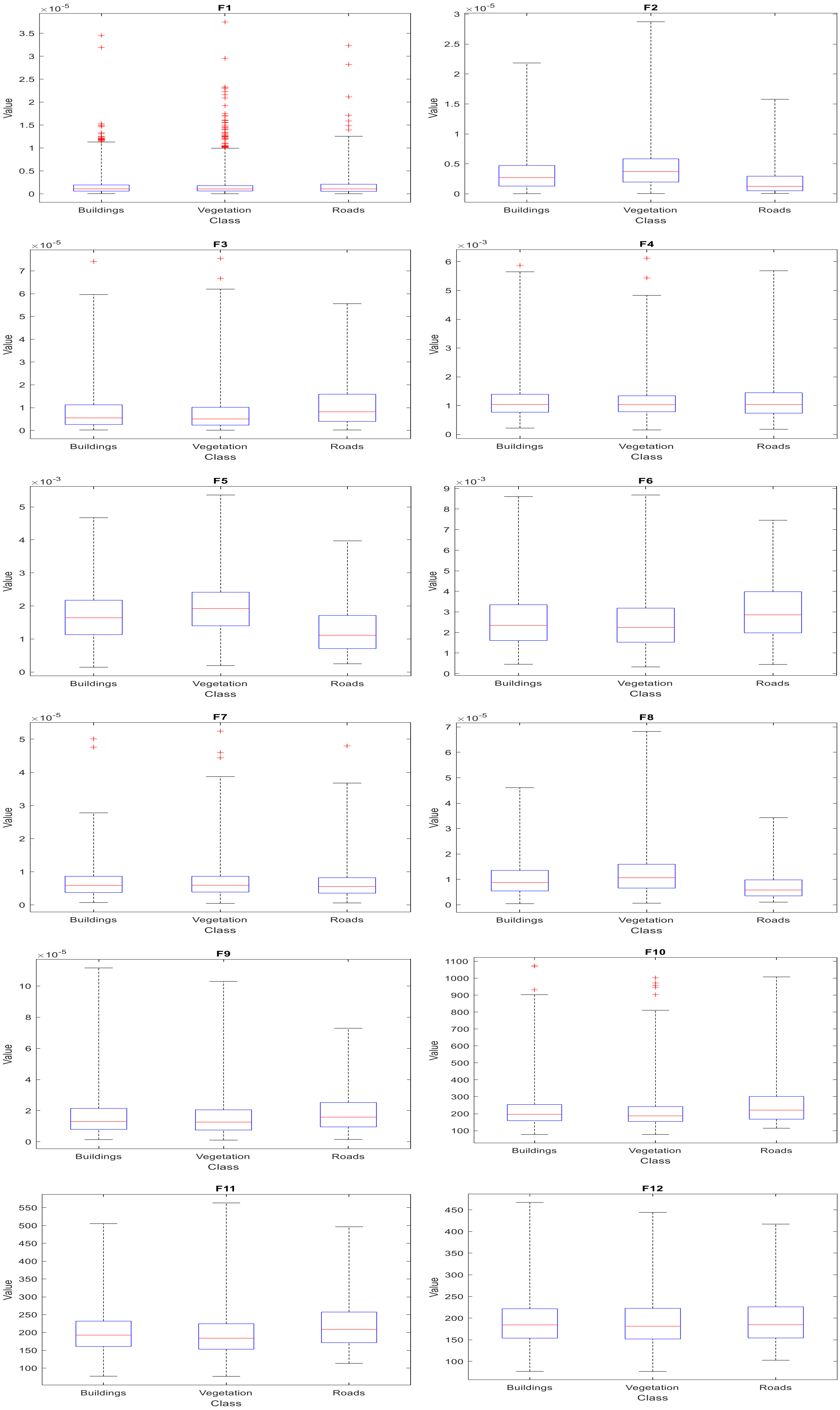


Figure 3.12: Box plot representation of HSV features of Feock image for SLIC 25,000.

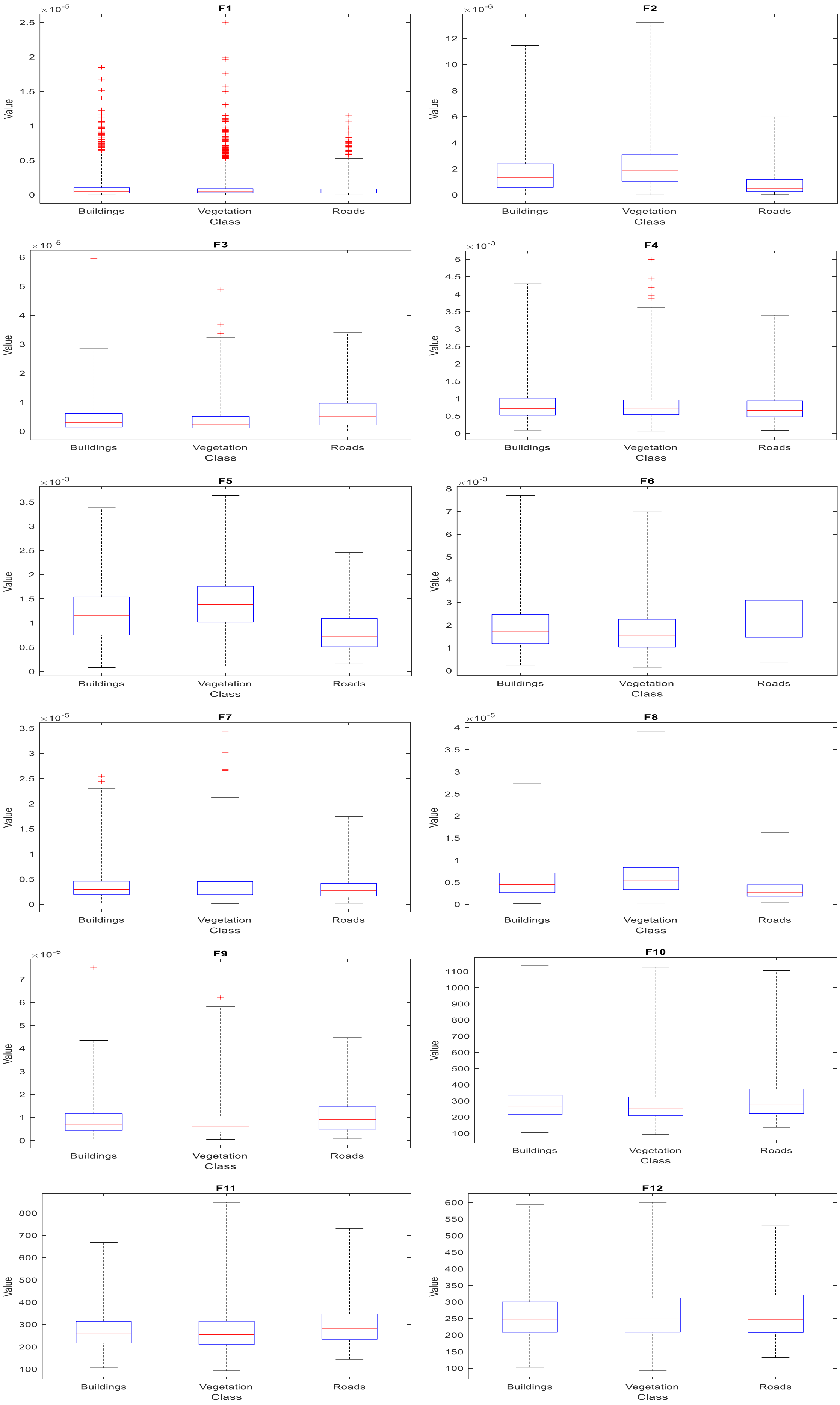


Figure 3.13: Box plot representation of HSV features of Feock image for SLIC 50,000.

3.3.2.3 Texture Based Features

The texture of an object is a discriminating attribute for classification or segmentation of different kinds of objects of interest in image processing research problems involving samples with a different visual external structure which can belong to any aerial image, photomicrograph, or satellite image (as in this case study) [218]. To assess the impact of texture features on the classification of the SLIC segmented objects from Feock, four texture-based features are extracted to differentiate different classes of objects based on their texture. Extracted texture features are presented in Table 3.3, which includes different representational texture properties of the objects that help in discriminating different kinds of objects. To estimate texture features for objects, GrayLevel Co-occurrence Matrix (GLCM) is constructed for each object in MatLab, and subsequently, various properties of the object are calculated by using GLCM matrix [2018]. The Contrast feature provides the contrast of intensity values between a pixel and its neighbouring pixels in an object, where contrast value 0 denotes a constant object. The Correlation feature provides a measure of how correlated a pixel is with its neighbours in an object, where 1 and -1 indicate perfectly correlated and non-correlated objects, respectively. The Energy feature offers the uniformity measurement of an object structure, where energy value 1 denotes a constant object. The Homogeneity feature represents the extent of closeness between the distributions of pixel values in an object [219]. All these texture features are extracted and compiled as a dataset matrix in MatLab having four columns for four feature values, and the last column holding the class label of each object.

Table 3.3: Texture-based attributes of Feock objects.

Feature No	Feature
F1	Contrast of object pixels
F2	Correlation among all object pixels
F3	Energy of object pixels
F4	Homogeneity of object pixels

The box plot analyses of the four texture features is depicted in Figure 3.14-3.16 for same SLICs as used in RGB and HSV features, where each plot represents one of the four texture features. From the four feature box plots, it is evident that none of the features have any outliers, which means that these features can well discriminate the classification of objects. In terms of median and distribution overlapping, texture features also have got different data median while overlapping of distribution because of the visually similar texture of different class objects which is adding the probability of challenging classification in the next phases.

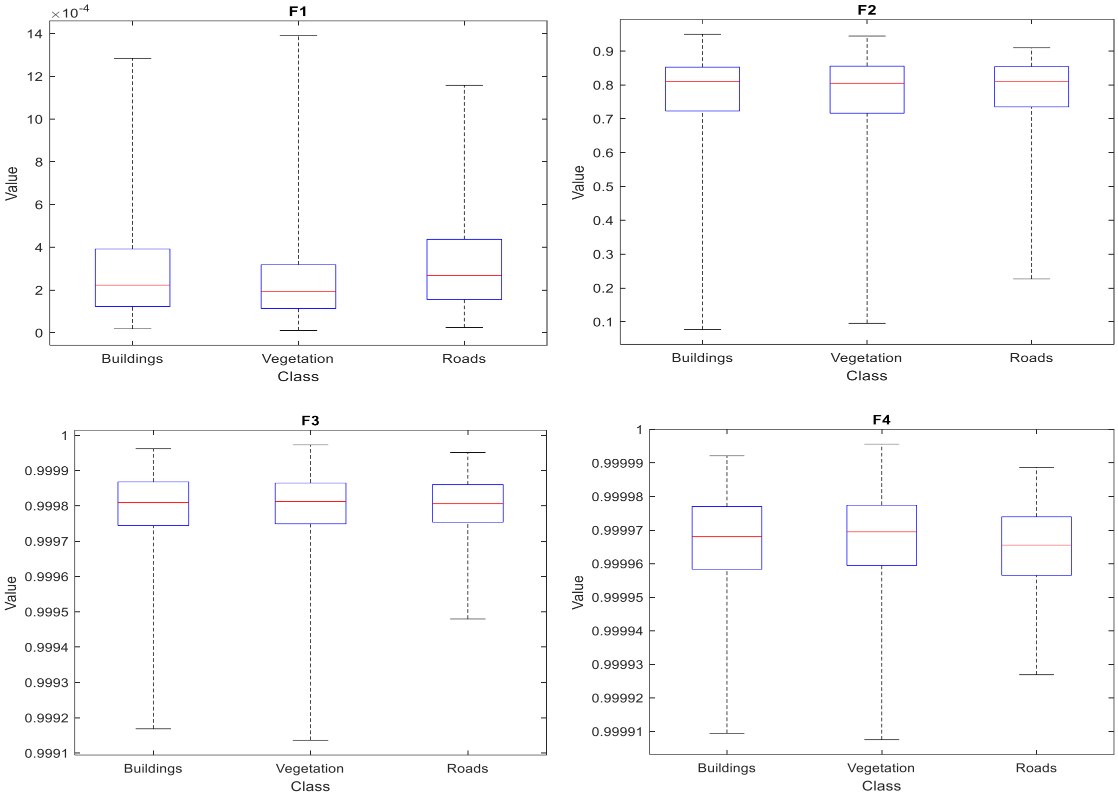


Figure 3.14: Box plot representation of texture features of Feock image for SLIC 10,000.

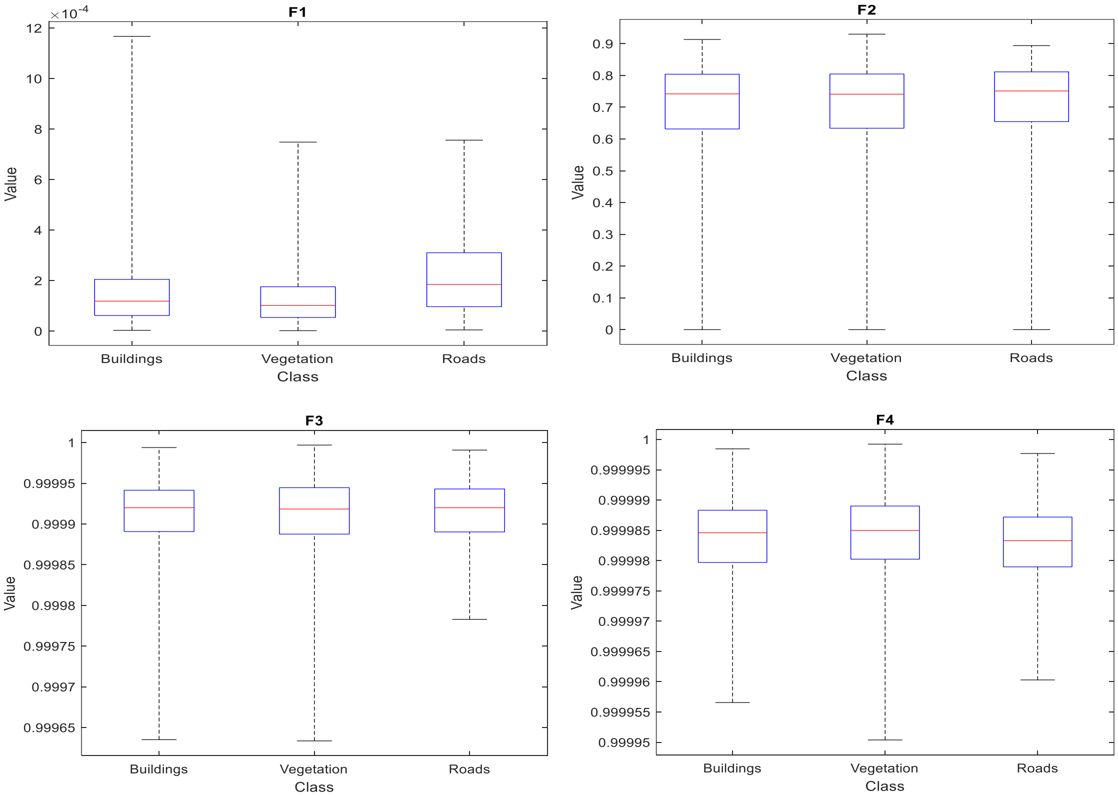


Figure 3.15: Box plot representation of texture features of Feock image for SLIC 25,000.

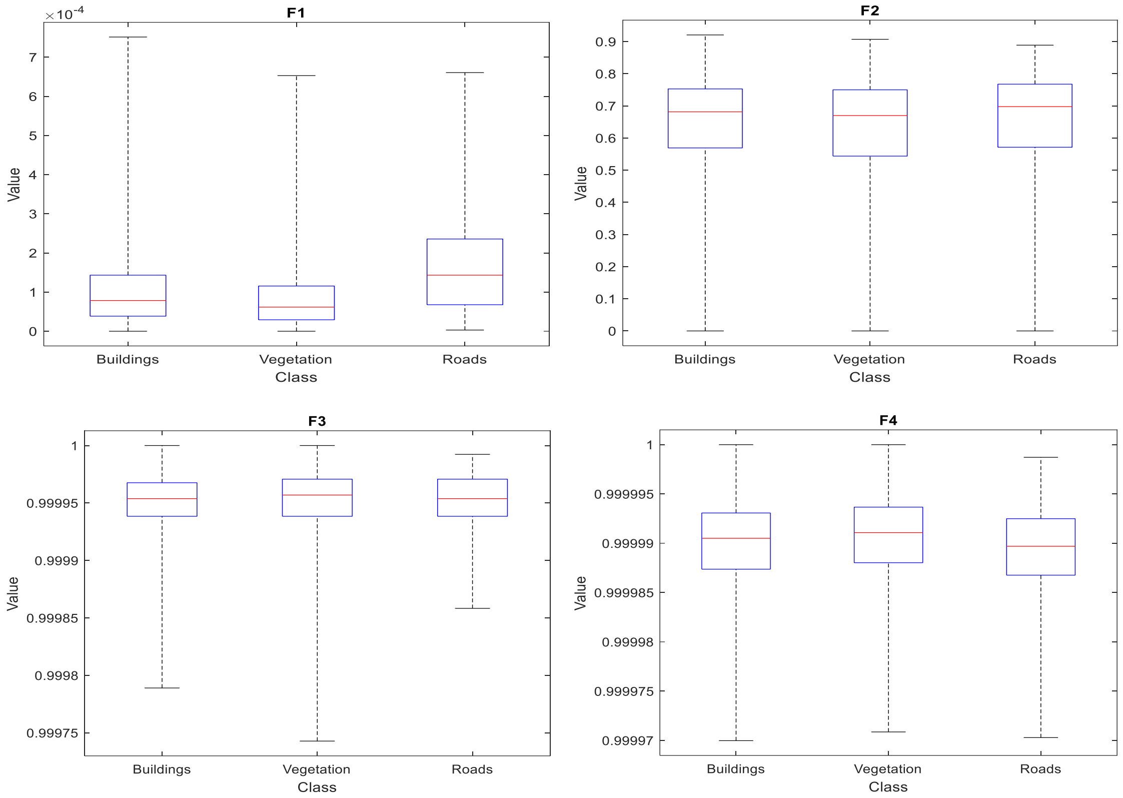


Figure 3.16: Box plot representation of texture features of Feock image for SLIC 50,000.

In this work, three different kinds of features are extracted and analysed for the Feock image based on outlier sample count, median and distribution overlapping in each feature. As noted in box plot figures for many SLICs, there are many useful features in RGB, HSV and Texture type of features since many of the feature plots have very few or no outliers while the plots are also show overlapping of distribution which also adds the boundary estimation challenge for classification. Too many outlier samples affect the learning of classifier in determining the right boundary for classification. However, it is not necessarily the case that the features which have fewer outliers and are non-overlapping in terms of median and boundary distribution are also the best in terms of classification because the number of outliers and overlapping conditions determine the quality of sample feature values. However, the classification task considers the combined effect of features quality and discrimination between features, which in turn can give the best classification results.

Once the features have been extracted for every object and a dataset of all object features has been compiled, all the compiled feature sets are to be tested and compared in the classification phase to pick the most suitable kind of features for this specific research problem. Then the classifiers are to be fed with the selected feature sets and their associated ground truth class labels to create an automated classification system for discrimination of different class objects in a satellite image.

CHAPTER FOUR

4 URBAN LAND COVER CLASSIFICATION

In this chapter, two scenarios are presented as a practical classification tool to classify real-world satellite images as permeable (i.e., vegetation surfaces) or impermeable (i.e., roads and buildings surfaces) by utilising features and information extracted from images data, presented in the previous chapter.

The first section in this chapter contains a step-by-step description of the selection process of the image data used for experiments in the first classification scenario. Two different classification algorithms, Classification Tree and Random Forest, are used with different parameter settings on selected images for Pixel-based classification. The results of these classification algorithms are compared and displayed in the form of Confusion Matrices and error plots in this section.

The next section includes the details about the second classification scenario. In the first part of this section, eight potential machine learning classifiers are selected and briefly described in the form of a table. After the selection of classifiers, a dataset of images was selected for use in this classification scenario. Next, this section includes a comparison of three kinds of features extracted in the previous chapter to aid in the selection of the most suitable kinds of features to focus on this research problem. Following, image segmentation is applied using superpixels-based segmentation to divide the image into multiple small objects and the selected features are extracted from segmented objects. In addition, the classification of objects in the dataset is conducted using the eight classifiers. For pixels-based approach, each image pixel is considered as an object, and same features are extracted from each pixel object, as extracted from superpixels-based objects. This pixels-based approach in scenario 2 is different from pixels-based approach in scenario 1 of classification, where each pixel colour channel values were considered as features.

A comparison of results from the eight selected classifiers for Pixels-based approach and Object-based approach is performed in the form of performance tables of the eight classifiers by incorporating Pareto with the diversity idea. Pareto-selected-classifiers are combined with weighted-sum concepts to create an ensemble classifier called ParetoEnsemble that will provide more reliable and improved classification results. The detailed design and implementation of the ParetoEnsemble classifier, along with a comparison of the results with individual classifiers, is given in this section. To reduce the effect of overfitting, data balancing was conducted to the training data, and the results concerning balanced and unbalanced data types are compared to select the most suitable kind of classification.

4.1 Remotely Sensed Land-cover Classification

The use of remote sensing data is pervasive in image analysis and artificial intelligence studies during the development of computer-supported decision-making, categorising and integrating systems to identify distinct surface materials in a convoluted urban environment via airborne platforms. Essentially, automatic classification, a term coined by the remote sensing community, indubitably acts as a critical catalyst in materialising the benefits of remote sensing data.

4.1.1 The 1st Scenario (Pixel-Based Classification)

In the first scenario, Pixel-based image classification is carried out by using MatLab built-in tools, to apply two classification methods, named Classification Tree (CT) and Random Forest (RF). Pixel-based classification involves the use of each pixel's spectral signature to allocate each individual pixel to the most appropriate class. In this section accuracy, reliability, and training size dependency of CT and RF are analysed using different tree sizes. The results show that, by using properly dimensioned RF, efficient, accurate, and reliable classification results can be achieved, and it is possible to identify permeable and impermeable areas more accurately.

4.1.1.1 Data and Study Area Image

The test image shown on the left of Figure 4.1, is an urban area of Sweden (180x249 pixels) taken from MIKE Powered by DHI urban mapping [220]. This image is used along with the Feock (area of study image) as the main test images for the performance evaluation of the CT and RF algorithms. Because both methods use supervised learning, ground truth data is required in the training phase. For that reason, a rough map with ground truth pixel labels was manually created at the same resolution as the sample image, as shown on the right of Figure 4.1. The marked map is labelled very carefully to make it coincide with the original map pixels to keep the resolution and orientation of both images precisely the same as elaborated in section 3.2, Chapter 3. The capturing details of the Sweden image, including satellite name, legend, north and scale, are shown in Figure 4.2.

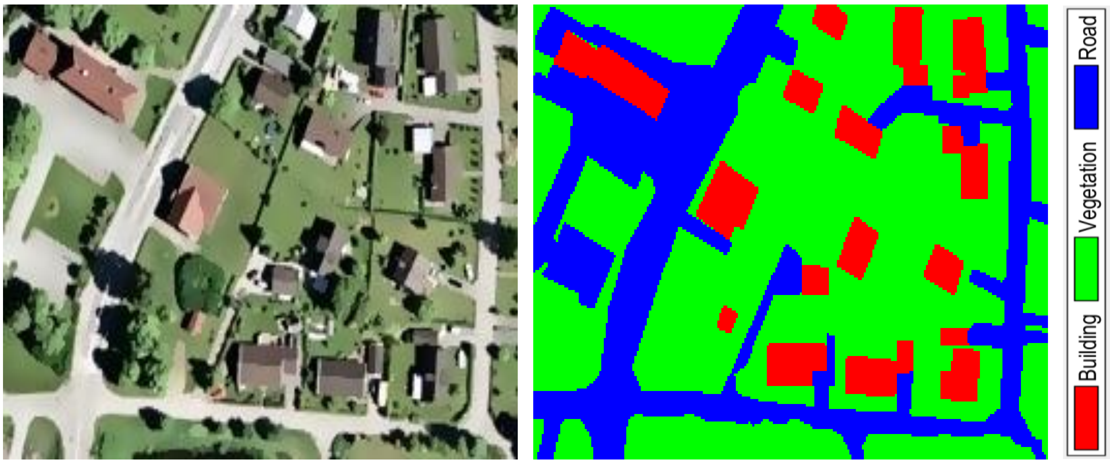


Figure 4.1: Sweden test image on the left and its manually labelled ground truth on the right.

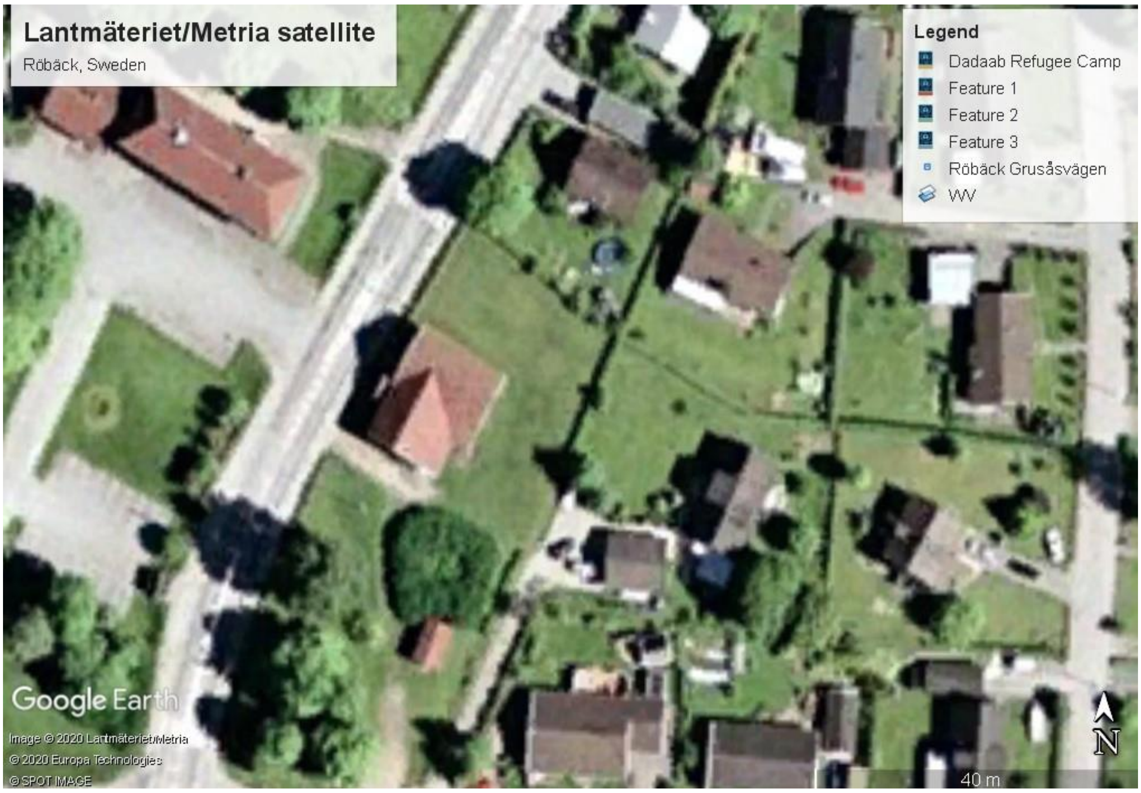


Figure 4.2: Sweden satellite image presenting captured satellite attributes.

4.1.1.2 CT and RF Bases Image Classification

This is the first classification scenario of this study, which is dependent on constructing pixel-based features and was carried out by using the CT and RF classification methods by utilising the tools available in the MatLab Statistics Toolbox. The classification accuracy for all the classification results is calculated by using the ground truth reference points of the validation areas (i.e., the ones excluded from the training area).

4.1.1.2.1 Data Division for Classification

As a first step, the satellite image and its ground truth were loaded into MatLab. Usually, the set of images is divided into training and testing sets. However, because only one image was used with its ground truth at a time, some parts of the image were used for training, and the rest of the image was used for testing.

Secondly, the data is divided between training and testing by creating a binary training data mask, as shown in Figure 4.3. The white dots represent the areas of the image that were used to train the tree, and the black dots show the areas used for testing. For that, an empty matrix of the size of the original image was created in MatLab and a random percentage of pixels was selected and used for training data. The 1s (or white dots) in the mask represent the training data. The remaining 0s (or black dots) in the mask were used later for testing the data. To get random pixel positions, the Rand function in MatLab was used to get a matrix of random pixel locations in an image. The Rand function returns true and false values based on random selections. True is represented by 1 and false by 0. A value of 1 indicates that these pixels were selected for training, and a value of 0 means that the pixels were not selected.

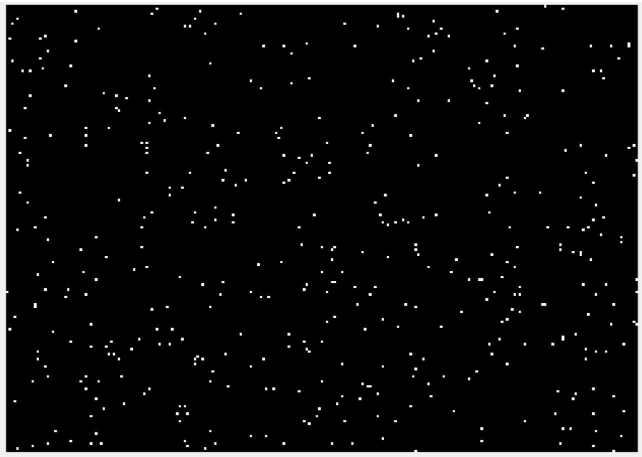


Figure 4.3: Binary mask representation of selected pixels for training data.

4.1.1.2.2 Image Pre-processing

The satellite images used in this section are represented in RGB format during processing. For the processing of each pixel, either training or testing, both the original image and the ground truth were required because the actual pixel label properties of the image come from the ground truth of that image. Regarding ground truth, labels are pre-determined for each class pixels where the buildings are marked as red, roads are marked as blue, vegetation areas are marked as green and irrelevant/unwanted area is marked as black. In other words, the colour intensity values are fixed to differentiate different colours. For example, red has intensity values of 255, 0 and 0. Therefore, to distinguish vegetation, one only needs to see whether the green component of the pixel colour intensity value is 255 and that the red and blue components are 0.

To create a data matrix of vegetation, a matrix having all false is created having the same size as the image, and the pixel values detected as green are then set as true in the data matrix. For building, the red component of pixels is 255 and the green and blue components are 0. Therefore, these points are set to true in the building data matrix. For roads, if the blue is 255, then these blue pixels are marked as true in the road data matrix. All the ground truth pixels are expected to contain three fixed-coloured pixels, red, green and blue, to mark all data pixels with some class labels. However, the ground truth image being used is not just red, green and blue. It also includes some other colour shades because of aliasing effects, as shown on the left side of Figure 4.4. This is possibly due to the use of the freeform tool for the creation of the ground truth image. Therefore, the ground truth data was reworked to apply antialiasing by making it contain only red, green and blue colours with no shades of other colours because the aliasing effect in the ground truth image introduces pixels with non-fixed values for red, green and blue channels. To deal with this issue, all pixel values were checked one-by-one, and when some pixels that did not have fixed values for the three or any of the three channels were identified, the channel with the maximum value was assigned as the fixed colour for that pixel. For example, if a pixel had a red channel value of 205, a green channel value of 225 and a blue channel value of 101, then the channel with the maximum value, the green channel in this case, was assigned as the pixel colour by setting its red channel and blue channel values to 0 and green channel value to 255. This way, all the pixels in the ground truth image were converted to fixed colours in the ground truth image, as shown on the right side of Figure 4.4. All the unwanted or irrelevant areas in the ground truth image were marked with black pixels.

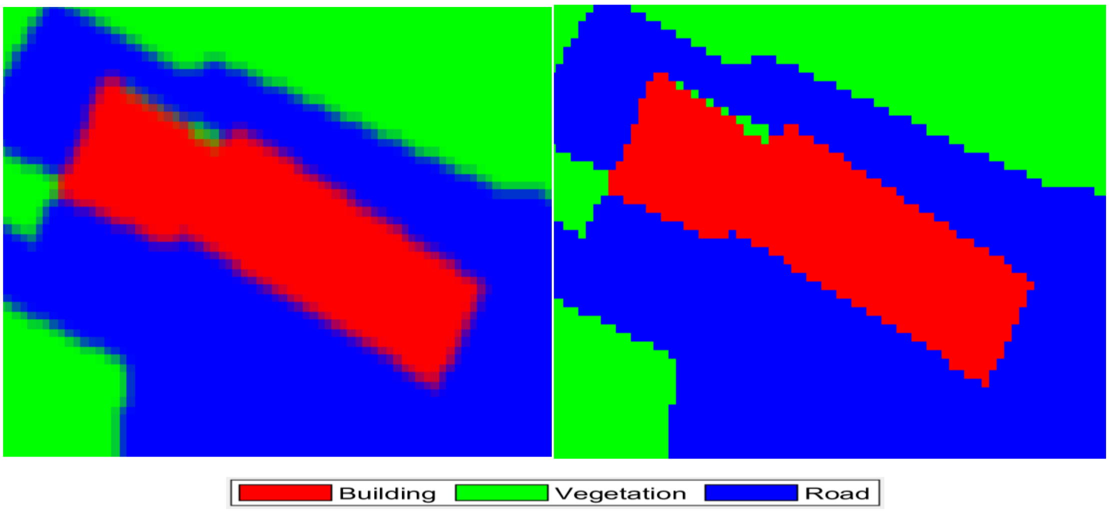


Figure 4.4: Aliasing effect in ground truth image on the left; ground truth after adjusting on the right.

Next, three binary mask images were created based on each class pixels, as shown in Figure 4.5. The left image represents the vegetation-class pixels, which are labelled as white. All the green pixels in the ground truth map were marked as white in the binary vegetation mask image, and all remaining pixels were marked as black. Similarly, binary mask images for buildings and road classes were created and shown on the middle and right side of Figure 4.5, respectively.

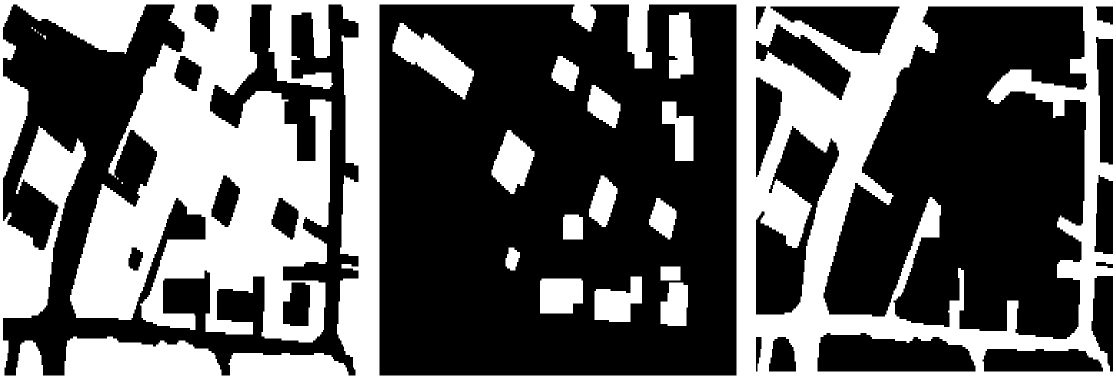


Figure 4.5: Binary masks for vegetation (left), buildings (middle) and roads (right).

After the ground truth masks for data images were created and adjusted, two input data matrices were created to apply cross-validation for the training of trees. The first matrix is the input data features set, and the second matrix is the ground truth labels. The data in the first matrix has three values for each pixel because there are three colour components in RGB images, and this classification is pixel-based. For each pixel, there is one ground truth label in the second matrix (where buildings, vegetation and roads are represented by numeric labels 1, 2 and 3, respectively). Therefore, if there is a dataset with 1,000 training pixels, then there will be two data matrices: the first one having 1,000 rows with three columns containing feature values, and the second one having 1,000 rows with one column containing data sample labels.

4.1.1.2.3 CT and RF Methodology

CTs and RFs use training data to build predictive models, and the trained models are then used to predict unseen instances of new data. Therefore, these classification models can be considered as trained predictive models, where training is the process of generating the tree. The RF classification method is applied by combining multiple CTs to create a forest of trees in which the results of multiple CTs are aggregated to create a final result that can reduce overfitting (i.e., poor performance in classification) in the classification model without increasing the number of classification errors. As an example, consider the top-down-trained binary tree that was constructed through suitable training, shown in Figure 4.5, from Sweden image. If a new test pixel is introduced in the form of x=[x1,x2,x3]=[R,G,B] as x(test)=[150,150,150], where x1 corresponds to R channel, x2 corresponds to G channel, and x3 corresponds to B channel, and x1=150, x2=150, and x3=150, then starting from the root node with x3, the tree is traversed by following the appropriate branches. Because x3=150 (<156.5), the left branch is followed, and x1 is tested. Next, because x1=150 (<152.5), the left branch is tested and x2 feature is tested. Because x2=150 (>132.5), this time, the right branch is followed, terminating at the terminal node of 2, giving the classification result as 2 (vegetation class), which might be any of the three pre-determined class labels like building, road, or vegetation. In this particular example, the tree was constructed by using 75% of randomly selected pixels with known ground truths, the remaining 25% of the pixels were predicted as test samples by using the constructed/trained tree. In the current case, the observation vector x (R, G, B) is the instance, and the attributes are the pixels' R, G and B values, where each instance is in the form of a vector with three components. As shown in the tree diagram of this Figure 4.6, x3 is considered as best attribute and is considered as root node of tree.

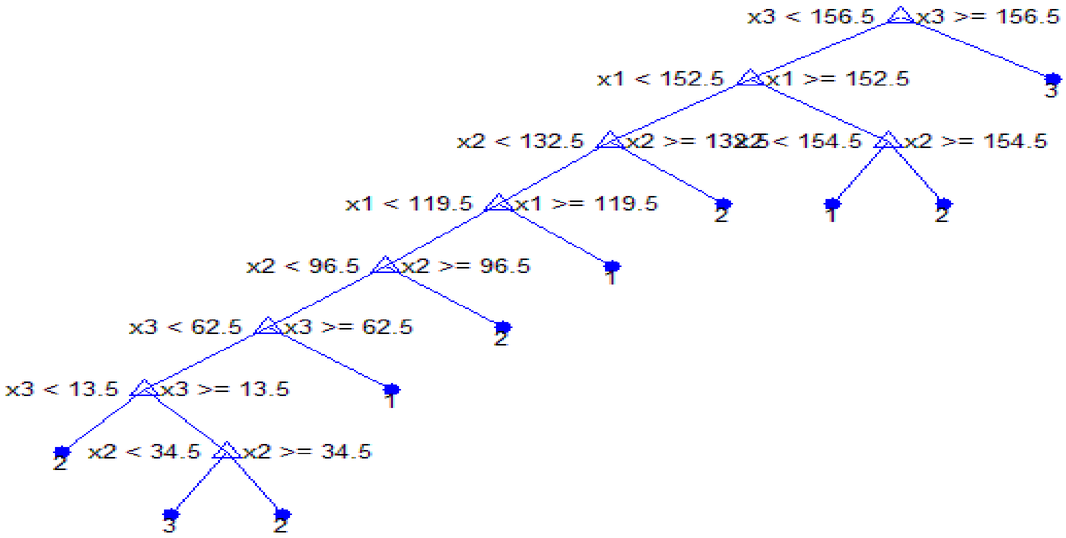


Figure 4.6: Tree structure containing a root, nodes, branches and leaves.

Different tree size experiments are performed by adjusting the maximum splits (points of decision) in the tree to observe the effects of tree size on validation and testing errors by reducing some branches of the tree (i.e., reducing the complexity of the tree). However, too much reduction in branching might not leave enough complexity in a tree to distinguish all features correctly, which means reducing the tree too much will not leave enough decisions to classify samples correctly. However, adding more questions after getting a correct model does not help either because it will only increase computational cost without giving any improvement in performance. In theory, the minimum possible tree count that can give the best possible predictions should be used. Therefore, one of the many training-phase challenges includes finding the minimum number of trees that satisfy the accuracy requirements, which is the optimum tree size concerning efficiency and speeding up algorithm processing. Concerning precision, higher tree size and using smaller image size is the best choice. However, regarding speed, the use of fewer trees and smaller-sized images is best.

4.1.1.2.4 Performance Evaluation

The tree constructed based on the training samples is then used to predict the test pixels of the original image. The result is a vector of single points evaluated as either vegetation, buildings or roads, represented by 1, 2 and 3, respectively. Because the prediction is pixel-based, each pixel gets evaluated by the trained tree based on the pixel's colour. After testing, the percentage of incorrectly evaluated data is calculated and displayed to estimate the percentage error of the predicted data against the ground truth because percentage accuracy and error are among the most common performance evaluation metrics. For that reason, only the test data pixels (all except the training data pixels) are considered in the following Equation (4.1):

$$Percentage\ error = (counter\ unmatched/counter\ total) * 100 \tag{4.1}$$

Where the counter total is the total number of pixels used in the testing or all the black dots in the mask created in Figure 4.3 (the number of test data pixels), and the counter unmatched is the number of incorrectly evaluated test data pixels. Figure 4.7 provides an example plot of the percentage error equation used above. The red dots represent the training data that does not have any share in the evaluation part. The green dots represent the testing data (i.e., the counter total is 5). The testing data in red squares is incorrectly predicted data (i.e., the counter unmatched is 2), which means these predicted labels are not same as ground truth labels. The testing data in green squares represents correctly predicted data (i.e., the counter match is 3). Thus, the percentage error was estimated in Equation (4.1.1) as follows by applying Equation (4.1) above:

$$Percentage\ error = (2/5) * 100 = 40\% \tag{4.1.1}$$

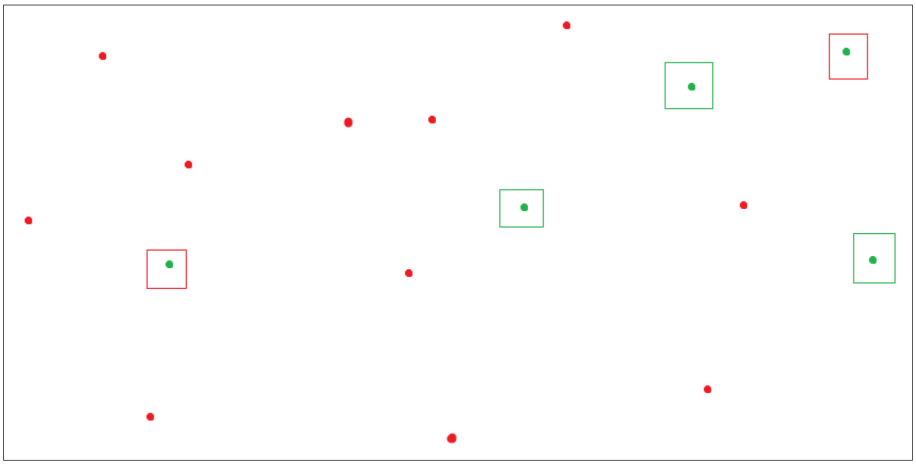


Figure 4.7: Percentage error example plot.

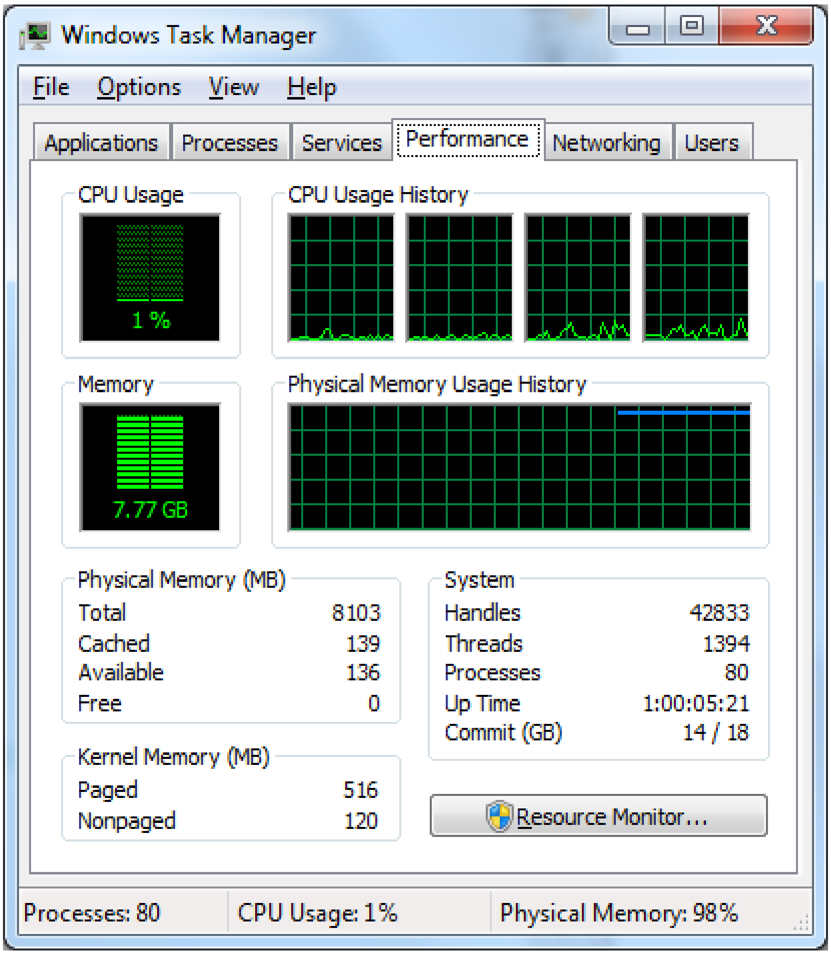
4.1.1.3 Results and Discussion

To compare the effectiveness of various classification methods, a common evaluation system must be used. The same test conditions and calculations are applied for multiple runs of CT and RF algorithms, and the results of all runs are averaged (to reflect the more meaningful statistical averaged values) because a result can fluctuate slightly every time the code runs because randomly-selected pixels are used for training and testing in each run. In RF, four different forest size configurations are explored for the test data, which includes: 10 trees for the first configuration, 20 trees for the second configuration, 50 trees for the third configuration, and 100 trees for the fourth configuration to consider the impact of tree count on the performance of classification. The four different instances are chosen randomly to consider both low and high tree count values in evaluation.

Because tree construction and testing take much time, depending on the number of features and the amount of testing data, an attempt is made to reduce the processing time for CT and RF by utilising parallel processing. For that reason, two different forms of processing are applied in this section, and the results for both are compared, which include normal AxA (with single-core) and parallel AxA (with multi-core using Parfor function in MatLab). AxA means selecting training and testing areas on the same image, either Sweden or Feock. The selected parts in the training image are to be used for training of the algorithms, and the remaining parts of the image are to be used as the input for the testing of the algorithm. In other terms, both normal and parallel structures use the same training and testing conditions and data, differing only in processing modes. The codes of both structures compute average error (incorrectly-predicted pixels), meantime per run (average time per pass/round of calculation) and Confusion Matrix (target data against predicted data) for a single image (where 75% of the image pixels are used for training, and the remaining 25% of pixels are used as the testing data, without any overlap between training and testing data). After running multiple rounds of the same settings to reduce the effect of random training and testing data selection, the results were averaged to get the average prediction result. Multiple rounds reduced the effects of the random selection of data, but there is a memory issue with this idea because by adding more rounds, the memory required to keep all that data increases linearly, which is a highly visible issue in the case of large images.

In an attempt to run RF with a set of 20 trees, for 10 rounds, on an 8GB RAM HDD device, to test the computational power requirement of this algorithm, its execution time was more than a day, then the device became unresponsive. The computer resources were monitored, during the execution of the algorithm, to discover that the issue was caused due to insufficient RAM (Figure 4.8). 8GB should be fine, but it is very close to the maximum. If there are other programmes, taking up space, allowed to swap, it may be delayed due to swapping memory.

For this reason, a higher resources-based system having 16GB RAM with SSD is used instead, to run the algorithms.



Physical Memory (MB)

Total	8103
Cached	139
Available	136
Free	0

Kernel Memory (MB)

Paged	516
Nonpaged	120

System

Handles	42833
Threads	1394
Processes	80
Up Time	1:00:05:21
Commit (GB)	14 / 18

Processes: 80 CPU Usage: 1% Physical Memory: 98%

Figure 4.8: Windows task manager showing insufficient RAM issue.

The processing speed of algorithms is also of great importance when working with high tree count for CT and RF for multiple rounds of processing. To speed up the processing, the code was modified to handle parallel processing by using MatLab's Parfor function, which is functionally the same as the normal structure code, but it has the capability of working on multiple workers in parallel by detecting the physical and logical cores of the operating system. Parallel computation makes the execution of the classification algorithm faster, but it uses a considerable amount of memory as a cost because each worker in parallel processing keeps its copy of processing data. For this reason, when applied in parallel cores, the memory required for processing becomes higher than that required in one core. The RF algorithm needs much more memory than CT because RF is a collection of multiple CTs that create a forest, hence it needs much more memory comparatively. This is more visible with larger images

(Feock). Another important aspect here is that the tree size was reduced to a reasonable number of leaf-node observations in each tree leaf, which can lead to the best possible accuracy because too many leaf-node observations can create overfitting in tree learning and reduce generalisation in learning while too few leaf-node observations can decrease classification accuracy. The code splits branch nodes layer by layer until a proposed split cause the number of observations to be at least one leaf node fewer than the minimum leaf size. The value of the minimum samples per leaf was set to 100 for both input sample images to get a deep tree, so if a leaf has more than 100 data samples, it is truncated. How much data is needed per leaf for the current scenario is specified by using the MatLab function MinLeafSize. However, this function seems to work only for the finished tree and not during its creation. The memory occupied during calculations is the same because the algorithm first computes the tree and makes modifications only once it is complete.

4.1.1.3.1 Performance Measurements

Figures 4.9-4.11 show 3 exported table plots that summarise the overall performance of CT and RF-based classification by generating graphs for mean testing error and mean computational time(s) for all configurations, including CT, RF10, RF20, RF50 and RF100 for both images of Sweden and Feock. The graphical representation of the results is more comprehensible than just tables. The red data points in the graph represent values for the Feock image, and blue colour points represent values for the Sweden image. The y-axis of the mean testing error graphs of Figure 4.9 shows the error percentages for classification. The mean testing error for the two structures (CT and RF) shows that the performance improvement (which is indicated by a reduction in average error) occurs with the increase in step size, which means an increase in forest size, but after some specific increases in tree size, the increase in classification performance is not much. It indicates that the maximum necessary complexity of the classification algorithm has already been reached and further increases to tree count would just make the algorithm slower without much improvement in performance. Also, the performance graphs show that RF100 achieved the best precision, mainly because of the greater capacity of this classification setup to describe data. The error plot, in Figure 4.8, illustrates the classification errors of both parallel and non-parallel modes, with two curve points, for two images

(Sweden and Feock), because both processing modes exhibit exactly the same errors and only processing time varies.

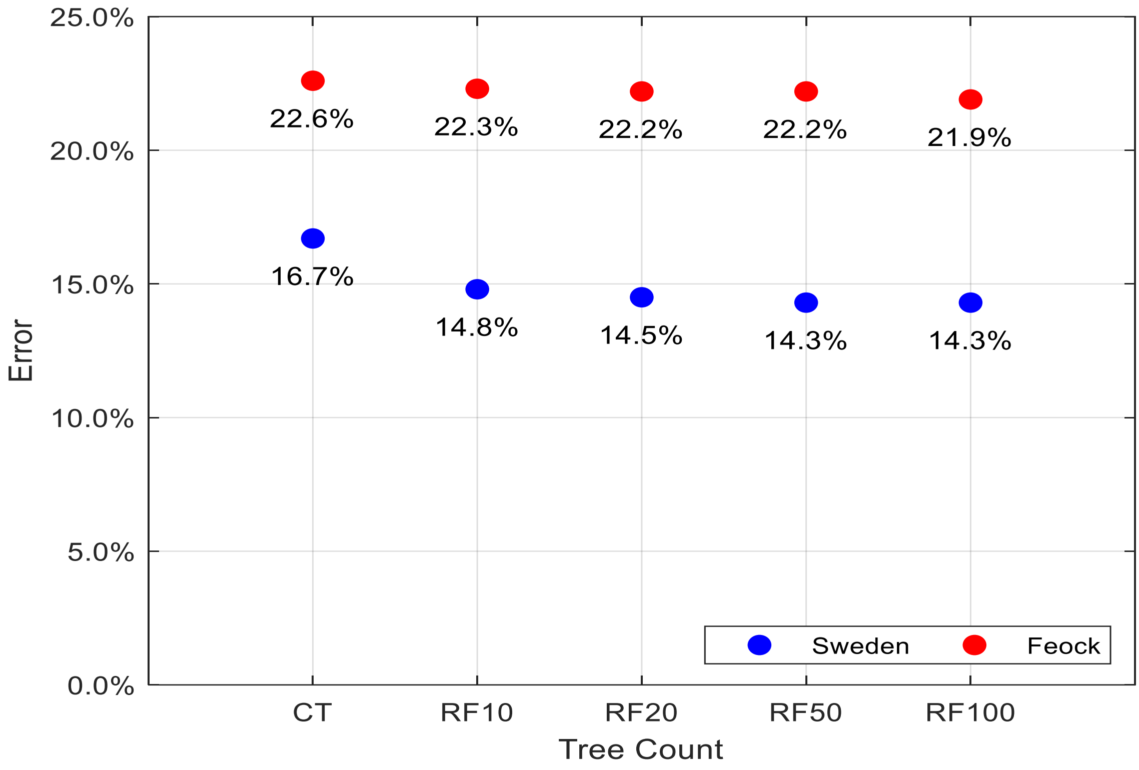


Figure 4.9: Tree count versus mean testing error for normal AxA and parallel structures.

The y-axis of the mean computation time graph in Figures 4.10 and 4.11 shows the time elapsed in seconds during classification, where the mean computational time graph shows that Parallel AxA (which means the same image was used for training and testing) is much faster than normal AxA. The machine used is 64-bit Windows 10 OS, with an Intel(R) Core i7 processor (2.20GHz) with 16GB of RAM. It includes 2 physical cores and 4 logical cores. MatLab is assigned 4 logical cores by the OS, out of which it is using 2 logical workers because hyper-threading is enabled. Each worker received some rounds to process when using the Parfor function during parallel processing, especially for the Feock test image, which is about 100 times larger in size than the Sweden test image (as shown in Figure 4.12).

Meanwhile, the mean computational time for the Feock image was much higher than that for the Sweden image because of the smaller size of the Sweden image. Another observation from result graphs is the better mean computational time performance of the CT method compared to the RF methods for both sample images, mainly due to the simple and less computationally demanding structure of the CT algorithm. Also, it can be observed in the computational time graph that the elapsed time of the parallel processing algorithm (especially for the Feock test image) is much less than that of the normal processing algorithm due to the use of parallel computing, which makes the fast execution of algorithms possible. Another observation from the graphs is that the results about mean computational time increase with increases in step size/forest size.

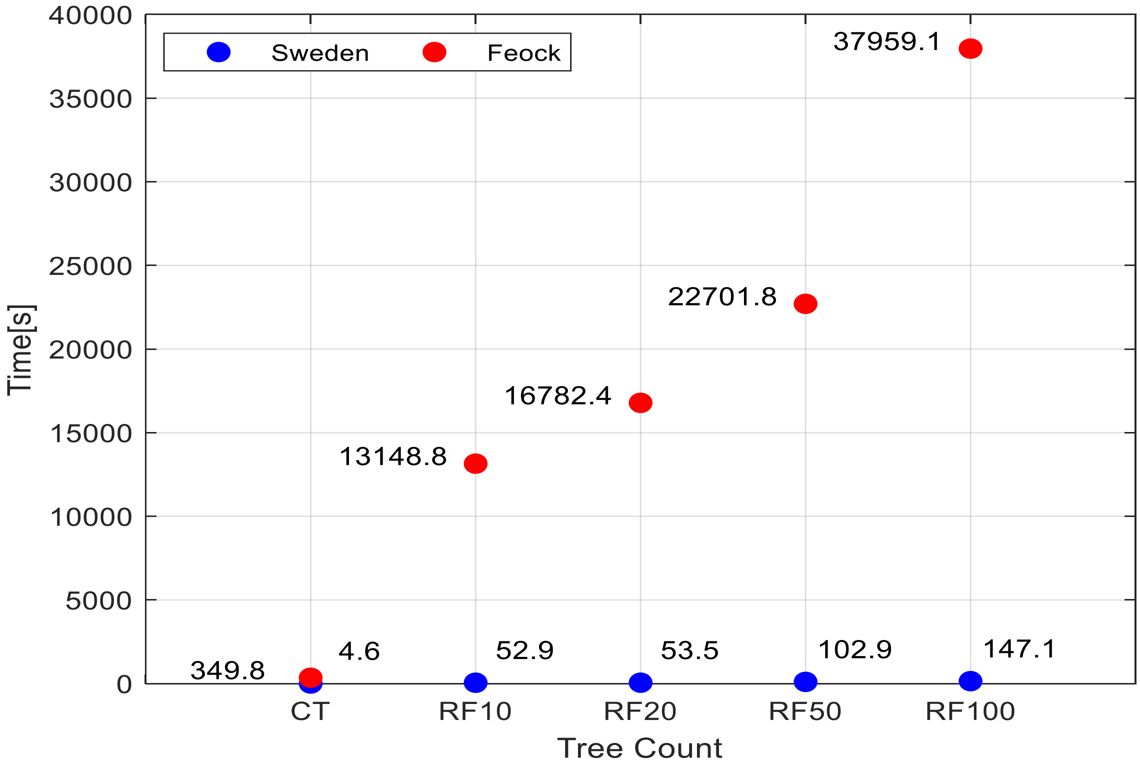


Figure 4.10: Tree count versus mean processing time for normal structure.

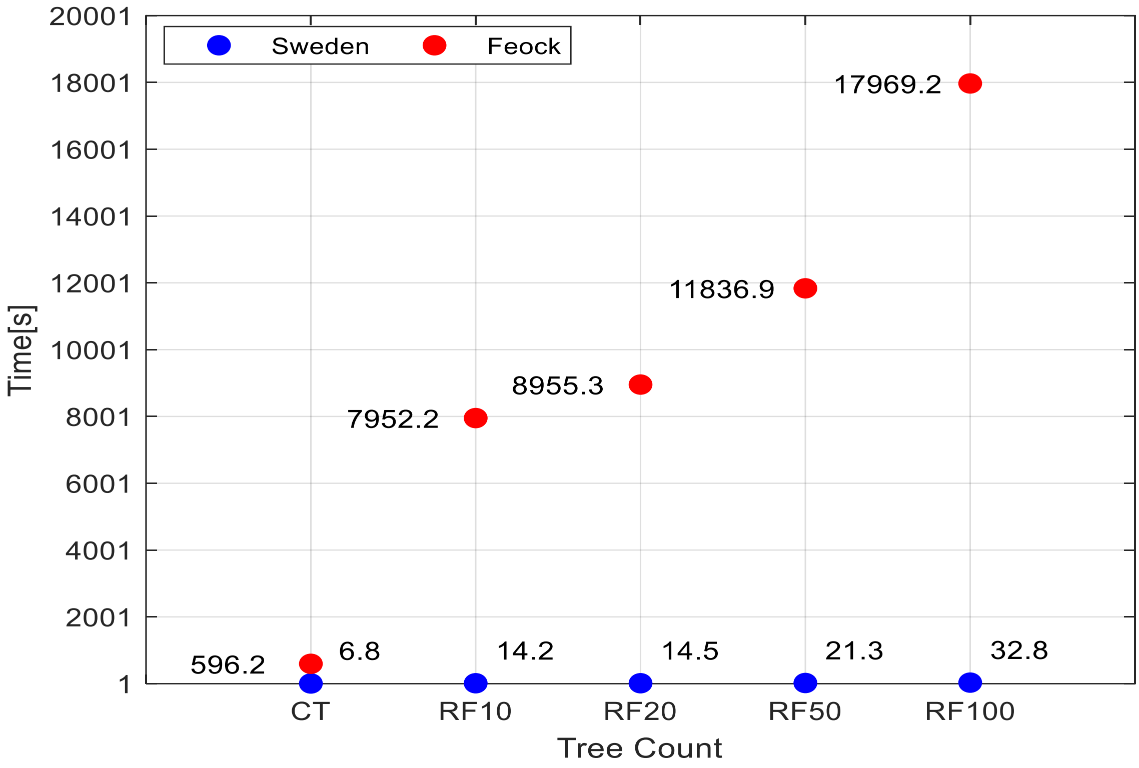


Figure 4.11: Tree count versus mean processing time for parallel structure.

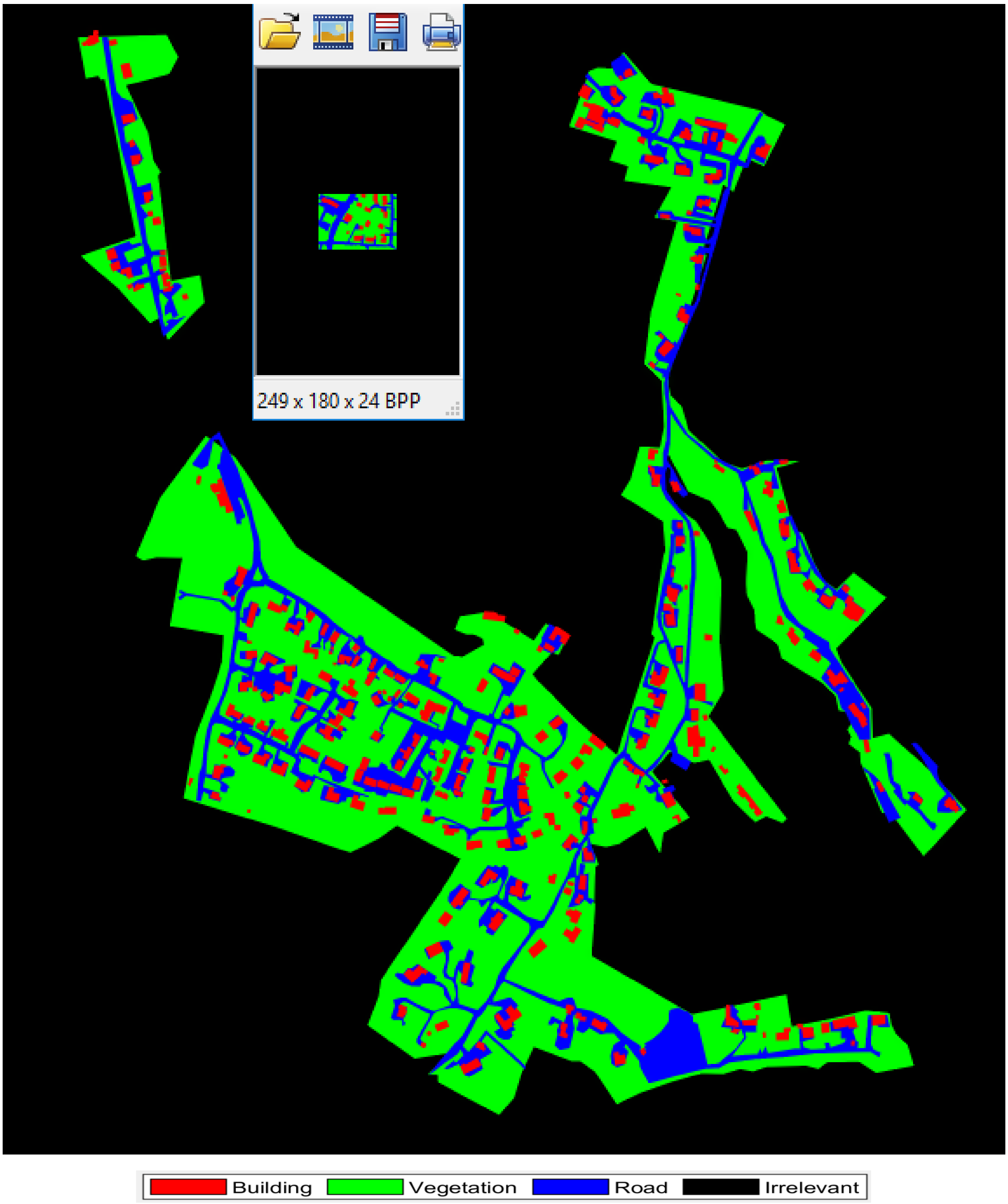


Figure 4.12: Size comparison between the Sweden and Feock maps.

4.1.1.3.2 Confusion Matrix Based Evaluation

The Confusion Matrices for both tested images are presented as a performance evaluation parameter, where the percentage accuracy of individual classes and the overall accuracy percentages are also included. Each case can fall into one of nine categories, BB, BV, BR, VB, VV, VR, RB, RV or RR, considering the three classes in this research scenario. The first letter represents the predicted value, and the second letter represents the ground truth. B, V and R represent buildings, vegetation and roads classes, respectively. The structure of Confusion Matrix for the current scenario is shown in Table 4.1.

Table 4.1: Confusion Matrix structure in this study.

		Actual		
		Buildings	Vegetation	Roads
Predicted	Buildings	BB	BV	BR
	Vegetation	VB	VV	VR
	Roads	RB	RV	RR

Tables 4.2 and 4.3 illustrate the Confusion Matrices of the performance of the classification model of RF100 because this classification parameter value gives the best performance amongst all settings according to Figure 4.9. The results are expectedly better for the Sweden image due to the quality of the test image because of the clear attributes of the high-resolution image. Note that both Confusion Matrices contain pixels from 10 runs/iterations to deal with the issue of random training and test pixel selection in each different run. Therefore, the total number of samples in rows and columns of a Confusion Matrix are 448,200 instead of 44,820 for the Sweden image with a size of 180x249 pixels. Similarly, the Feock image is 3456x4992 pixels (17,252,352 total pixels), but only 2,583,580 are the known ground truth pixels x10 rounds, which equal 25,835,800.

Table 4.2: Confusion Matrix of RF100 for Sweden sample image.

Sweden **RF100**		Actual values				
		Buildings	Vegetation	Roads	Total	Precision
Predicted values	Buildings	38,477 8.58%	7,177 1.60%	7,866 1.76%	53,520 11.94%	71.89% 28.11%
	Vegetation	5,897 1.32%	238,088 53.12%	14,395 3.21%	258,380 57.65%	92.15% 7.85%
	Roads	7,999 1.78%	20,905 4.66%	107,396 23.96%	136,300 30.41%	78.79% 21.21%
	Total	52,373 11.69%	266,170 59.39%	129,657 28.93%	448,200 100.00%	
	Recall	73.47% 26.53%	89.45% 10.55%	82.83% 17.17%		**85.67%** 14.33%

Table 4.3: Confusion Matrix of RF100 for Feock sample image.

Feock **RF100**		Actual values				
		Buildings	Vegetation	Roads	Total	Precision
Predicted values	Buildings	1,020,851 3.95%	329,317 1.27%	1,018,332 3.94%	2,368,500 9.17%	43.10% 56.90%
	Vegetation	262,486 1.02%	16,497,632 63.86%	684,652 2.65%	17,444,770 67.52%	94.57% 5.43%
	Roads	790,543 3.06%	2,589,327 10.02%	2,642,660 10.23%	6,022,530 23.31%	43.88% 56.12%
	Total	2,073,880 8.03%	19,416,276 75.15%	4,345,644 16.82%	25,835,800 100.00%	
	Recall	49.22% 50.78%	84.97% 15.03%	60.81% 39.19%		**78.04%** 21.96%

For all the previous statistical calculations of CT/RF, whole Sweden or Feock image data samples are used for classification. However, Feock image calculations and processing takes lots of time compared with Sweden image processing as can be seen in Figures 4.10 and 4.11. Therefore, some parts of the Feock image were cropped, to make it the same size as Sweden image (180x249) for training and testing, to get quick results for comparisons between Feock and Sweden image. The selected Feock image area for cropping is displayed in Figure 4.13. Another reason for using the cropped Feock image is that the use of too many data samples for the training of trees does not allow for the tuning of trees, which is the case for the full Feock image. However, the cropped Feock tree can be tuned and adjusted to get improved results but with some risk of added overfitting. The satellite image for cropped Feock, along with details like North, scale, legend and satellite name, are shown in Figure 4.14.

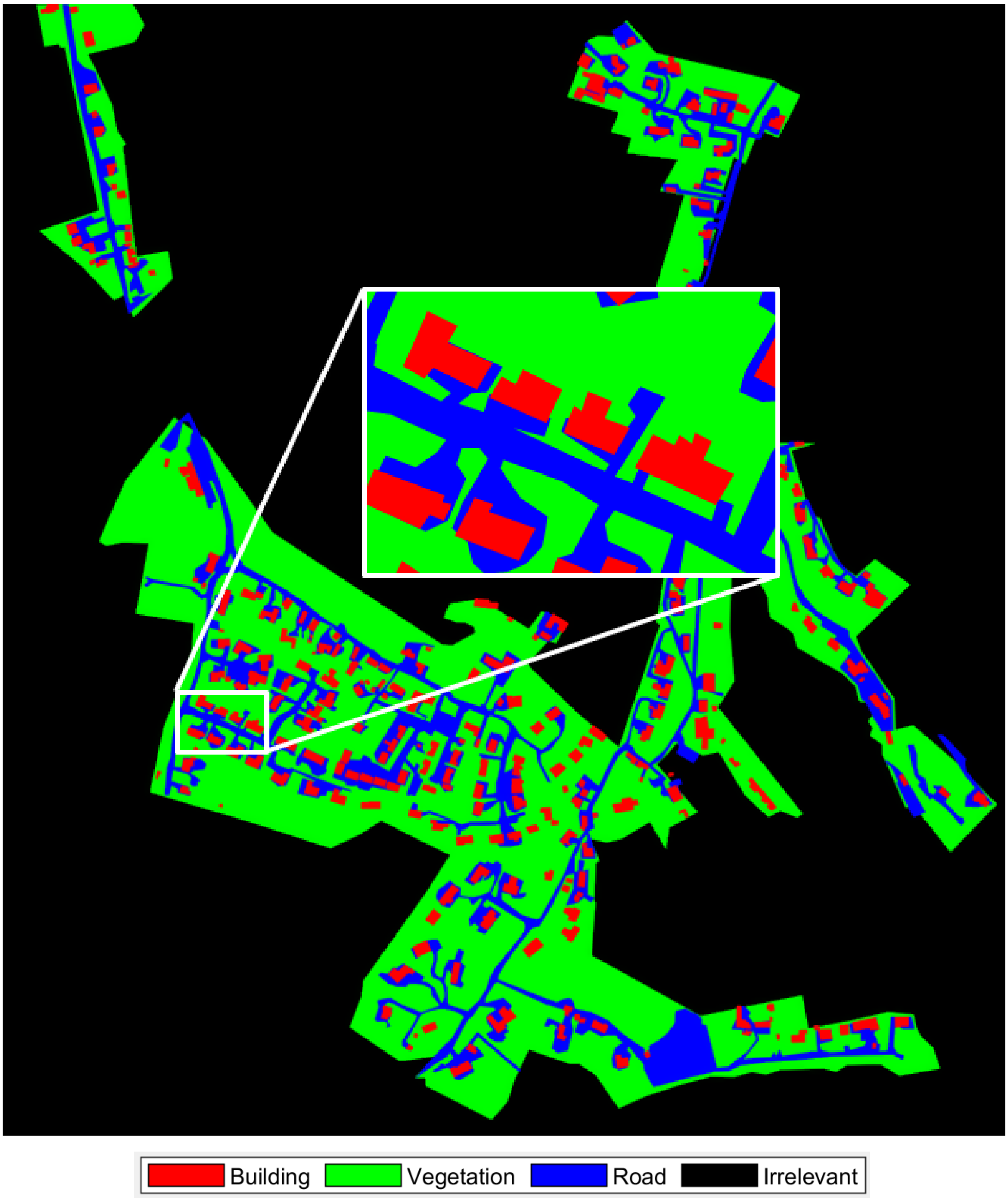


Figure 4.13: Feock ground truth image with some area selection for cropping.

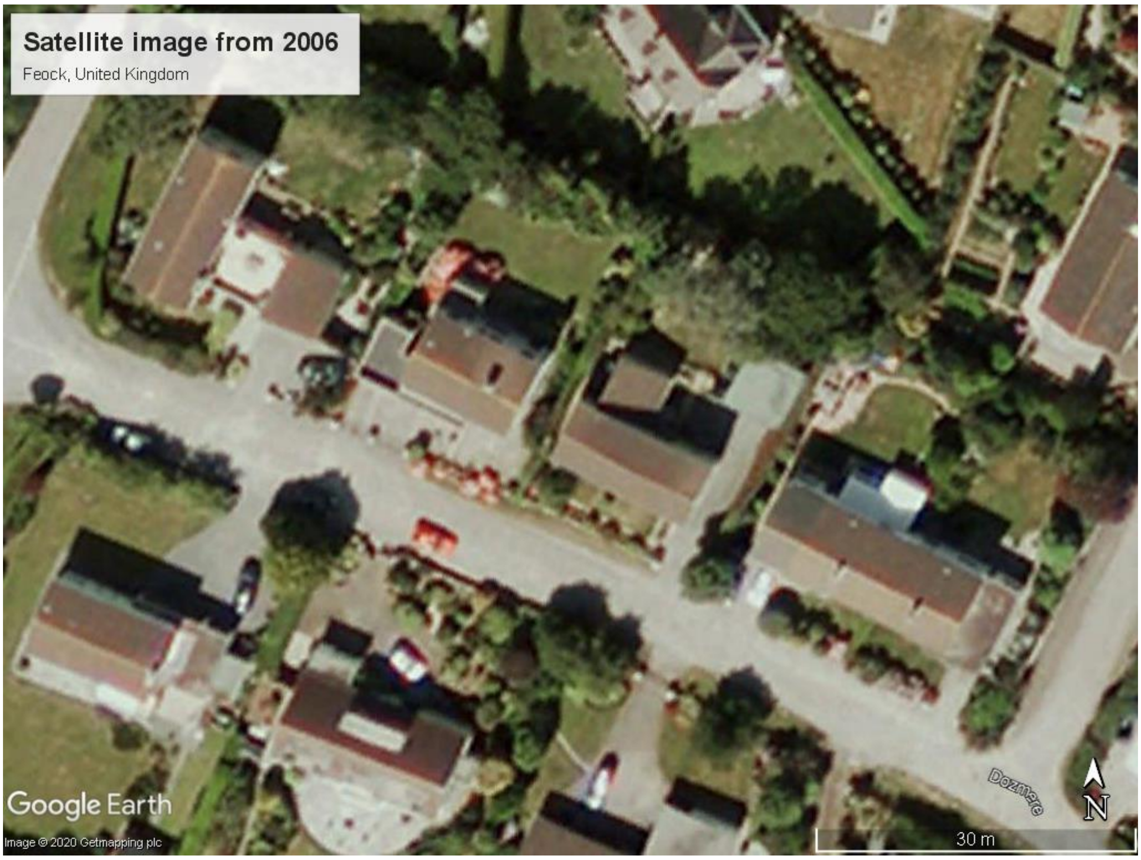


Figure 4.14: Cropped Feock image showing captured parameters.

4.1.1.3.3 Generalisation Assessment

After cropping, the Sweden and Feock images were the same size. These two images were used to add generalisation to the trained system in which one image is used for training and the other image is used for testing. This gives some generalisation to the system because in earlier cases, pixels from the same image were being used for both training and testing. In this case, training was done on one image and testing was done on the other one. One thing to note is that the training and testing pixel data does not overlap (one pixel cannot be used in both sets). The following are the options for selecting training and testing data while using two images, A and B, for processing:

- A is training, B is testing (AvsB).
- A is testing, B is training (BvsA).

The generated statistics in Tables 4.4 and 4.5 show a comparison between Sweden and cropped Feock by applying parallel processing. These tables show that all testing error percentages of CT and RFs for 10, 20, 50 and 100 classifications for AvsB are generally better than the percentages for BvsA. That is because the training was done on Sweden in this case, which is a very high-resolution and clear image compared to the cropped Feock image as can be seen in Figure 4.15. That means the Sweden image is well-defined compared to the cropped Feock image. However, there are some parts of roads in both images that are covered by trees, which also affects the overall predicted testing results.

Table 4.4: Performance table for A versus B (Sweden training versus cropped Feock testing).

	CT	RF10	RF20	RF50	RF100
Time individually(s)	1.62	4.55	7.15	15.18	29.58
Training error	0.13	0.05	0.04	0.04	0.03
Testing error	**0.36**	**0.33**	**0.33**	**0.32**	**0.32**

Table 4.5: Performance table for B versus A (cropped Feock training versus Sweden testing).

	CT	RF10	RF20	RF50	RF100
Time individually(s)	1.36	4.38	6.83	14.53	26.91
Training error	0.09	0.05	0.04	0.04	0.04
Testing error	0.39	0.36	0.35	0.35	0.34

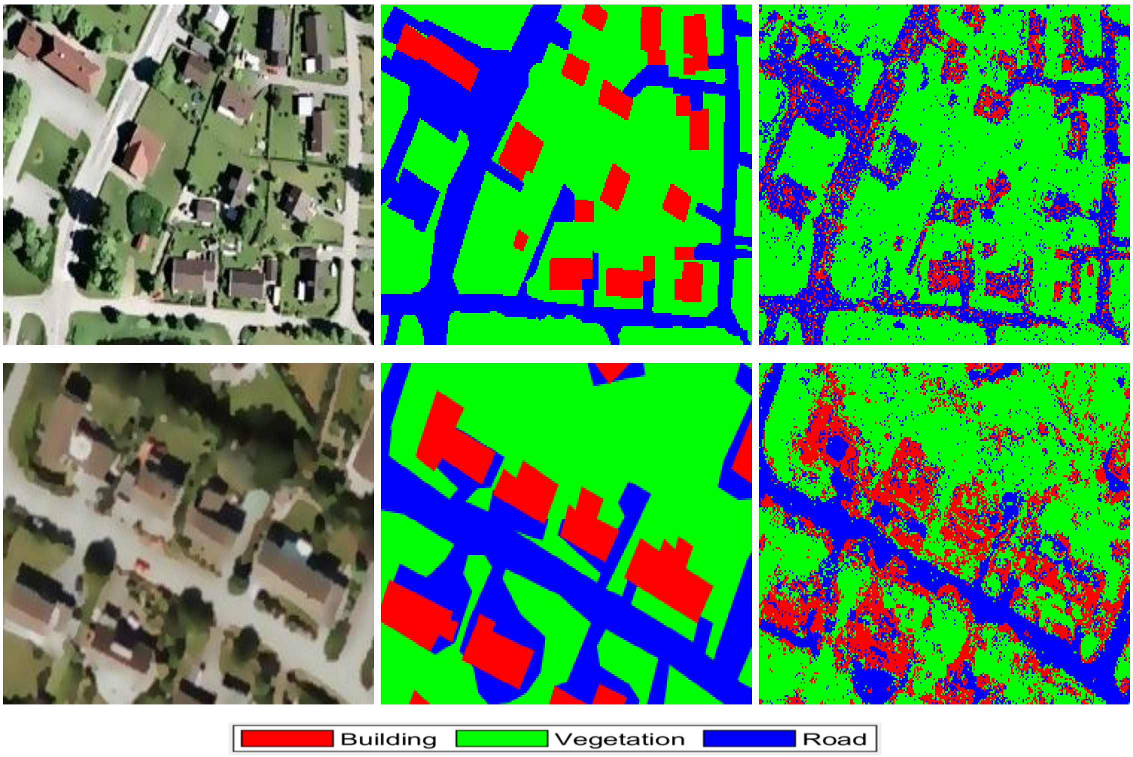


Figure 4.15: Top left is a satellite image of Sweden, top middle is Sweden ground truth image, and top right is predicted Sweden image created based on the training of cropped Feock image. Bottom left is cropped Feock satellite image, bottom middle is Feock ground truth image, and bottom right is predicted cropped Feock image created based on the training of Sweden image.

As observed in Figure 4.15, the predicted images that were created based on predictions of test pixels from the trained tree are not very much like the actual ground truth labels. The following are the possible reasons for which the outputs (predicted images) are not similar to the ground truth:

- The ground truth was created manually, so there is not an exact pixel-by-pixel match between the actual image and ground truth, which means the ground truth data may not correctly describe classes in terms of shape and quantity. For example, some pixels might be the same colour on the roof as grass or due to manual labelling some pixels of one class can be marked as another class when both are too close, for example, pixels on the edge of a roof might be marked as vegetation because vegetation and roof edge are closely linked, or the other way around, which may affect the training of the tree and learning of model.
- Pixel colour values are used as features, where the colour within each feature varies, and two features might have pixels of the same colour, which may add some conflicts in decision conditions during tree construction.
- Based on the simplified rules contained in the tree, the estimation might not be very precise because the tree has its limits.
- There is a big imbalance between buildings, roads and vegetation classes.

Suggestions to solve this problem are:

- The use of more accurate labelling for real images using more correct shapes of classes.
- The use of more classes by dividing vegetation into trees, fields, open ground, and so on.

4.1.1.4 Conclusions and Suggestions of the 1st Scenario

The CT and RF techniques are constructed in this section, first for pixel-based classification by using training and testing data samples from the same image, either Feock or Sweden. These images were then tested for parallel (using Parfor function) and non-parallel mode processing. The parallel mode gives quick results compared to non-parallel mode processing. CT performance was compared to different RF parameters (i.e., RF10, RF20, RF50 and RF100). RF100 was found to deliver the best classification performance.

Comparatively much bigger size of Feock image than Sweden image, it takes too much time for results collection and comparisons. Also, big sample data does not allow for the tuning of trees, like in the case of the Feock image. Also, big-data samples make the construction of the tree more complex, which also affects the performance of the classifiers. To reduce the impact of these issues, the Feock image was cropped to make it of the same size as the Sweden image. The testing error percentages of CT and RFs for 10, 20, 50 and 100 classifications generally show better testing error results when the training was done on Sweden image compared to the training on cropped Feock image.

To add generalisation to the system, training and testing was performed on different images. For that reason, once the cropped Feock image was used for training and the Sweden image was used for testing, then the Sweden image was used for training and the cropped Feock image was used for testing. The results show that Sweden image training and cropped Feock image testing gives better results than cropped Feock training and Sweden testing because the Sweden image is high-resolution than the cropped Feock image and the features are more discriminant in the Sweden image compared to the cropped Feock image.

Another generalisation type can be added here by selecting random pixels from the two sample images (A and B) and then combining them into one training set. This way, some percentage of pixels is selected from each image for training and testing (without overlap). The purpose of training CT/RF with random pixels from more than one image is to train the trees with multiple images to add generalisation to the system to get better predictions for future images. The more variety (different resolution, zooming and capturing lighting conditions) that is present during training, the more variety of images, and the trained system will be able to correctly predict.

Another new option is to generate results, not by using random 75% fixed percentage of pixels as training data and 25% for testing, but by dividing the image into, for example, five partitions to do cross-validation and always using one part for testing and the other four as training. Because the partitions are separate, there is no overlap between the training and testing sets.

4.1.2 The 2nd Scenario (Superpixels Based Classification)

In this section, the classification phase includes the utilisation of 5-fold cross-validation-based process, conducted over the training and validation folds of sample data, where the classifiers are trained on some folds of data and the one separated fold data is used for testing in each fold iteration, leading to the predicted values for the whole data. In this way, testing data is totally unknown to the classifiers, and the predicted and actual labels of data samples are compared, to derive the percentage of accuracy for the specific testing data.

The first step, in this classification phase, is to select various machine learning algorithms, which are going to be used for classification of datasets, in this research. For this reason, eight different classification machine learning algorithms, with varying strong properties, in terms of time, computational cost and complexities, are selected.

The next step is the evaluation of different types of extracted features, as potential features for this system. To this end, the classification of some sample test images is carried out, using three kinds of features: RGB, HSV and Texture features, providing the features set with the best results, in terms of cost and performance. This set is selected as the suitable features type, for the present classification system.

The next objective of this section is to evaluate eight ML classifiers, using the selected features set and comparing the two most common methods used for remote sensing classification: 1) object-based classification method, by applying SLIC superpixels segmentation for four object count instances (100, 300, 1000, and 10,000), and 2) pixel-based classification method, for object count equal to the total number of image pixel, without applying segmentation. This comparison determines whether an object-based analysis of remotely sensed imagery can produce better classification result that is statistically more accurate and less demanding in computational time, than a pixel-based analysis, when applied to the same data. The final step is to decide the optimal classifier and the optimum number of SLIC objects, in terms of execution time and accuracy. These two parameters are optimised through Pareto dominance analysis on the trade-off between classification accuracy and runtime.

The next issue of this section is about improving the produced results, compared to the best performing individual classifier results, for Feock image. This is achieved by implementing classification scores and weighted sum-based ensemble classification method, instead of using simple voting of predicted labels from selected classifiers, in traditional ensemble classifiers. In this case, the class exhibiting the highest weighted sum score value will be selected as a predicted class for all pixels in the object, one by one, providing the predicted labels for all image pixels.

4.1.2.1 The Selection of Machine Learning Classifiers

This section evaluates pixel-based and object-based image classification techniques, for extracting the three land-use categories (buildings, roads, and vegetation areas). Eight selected supervised machine learning algorithms are implemented, using MatLab computer vision toolbox functions. The selection of classification algorithms depends on many factors, such as training data selection, purity, size, composition and resolution of imagery, etc. Also, the sensitivity of classification algorithm changes, based on selection of training data from the available dataset. Therefore, one classification algorithm can give different results, based on different training data, from the same dataset [221]. The selection of eight different classification algorithms is based on the advantages-disadvantages analysis of many different potential algorithms, in terms of performance and associated costs. Most popular classification algorithms, including SVM and ANN, are avoided, because the available datasets consist of huge samples count and they are imbalanced, so they do not work well, under such conditions [114] [225]. Many different classification algorithms are tested, using MatLab Classification Learner toolbox, resulting in the selection of eight best performing algorithms, to be used in this research. The same set of classification methods are applied to the pixel-based approach and the object-based approach. Table 4.6 shows a table of prominent characteristics of classification algorithms used, whereas their detailed description can be found in [112] [114] [115] [224] [225], as these details are beyond the scope of this study.

Table 4.6: General characteristics of the selected classifiers.

Classifier name	Classification model	Learner method	Model flexibility	Interpreta-bility	Prediction speed	Memory usage
Fine Decision Trees	Decision Trees	Group of prediction rules	Maximum number of splits is 100 (high number of leaves)	Easy (simple model)	Fast	Low (1MB)
Medium Decision Trees	Decision Trees	Group of prediction rules	Maximum number of splits is 20 (medium number of leaves)	Easy (simple model)	Fast	Low (1MB)
Fine KNN	Nearest Neighbour Classifiers	Euclidean distance	Number of neighbours is set to 1	Hard (complex model)	Medium	Medium (4MB)
Coarse KNN	Nearest Neighbour Classifiers	Euclidean distance	Number of neighbours is set to 100	Hard (complex model)	Medium	Medium (4MB)
Cubic KNN	Nearest Neighbour Classifiers	Cubic distance metric	Number of neighbours is set to 10	Hard (complex model)	Slow	Medium (4MB)
Bagged Trees	Ensemble Classifiers	Random forest Bag, with Decision Tree learners	High number of splits	Hard (complex model)	Medium	High (100MB)
Boosted Trees	Ensemble Classifiers	Ada Boost, with Decision Tree learners	Medium to high number of splits	Hard (complex model)	Fast	Low (1MB)
RUS Boosted Trees	Ensemble Classifiers	RUS Boost, with Decision Tree learners	Medium number of splits	Hard (complex model)	Fast	Low (1MB)

4.1.2.2 Data Selection and Labelling

Due to the lack of suitable satellite image datasets along with ground truth (for this research specifically), available for the evaluation of different machine learning algorithms, six satellite images are considered for use in this study, which are collected from dissimilar sites (ground truths of these images are created manually), having different resolutions (Figure 4.16 (1)-(6). Varied land-use cases (i.e., buildings, roads and vegetation cover), present in these images, provide good representative examples of urban unit classification, which is important for land use/cover mapping and urban planning. Since the definition and acquisition of reference data, that is labelled ground truth, (direct measurement at ground level, which is used to verify remotely obtained data) is often a critical problem in remote sensing [226], the reference data, for all the images, were fully labelled manually into three classes: red, blue and green, representing buildings, roads and vegetation areas, respectively (Figure 4.16 (a)-(f)). A recent view of the six satellite images above, along with their recorded details, is presented in Figure 4.17. Some places appear different from the old version, because of the changes occurring at those places over time.

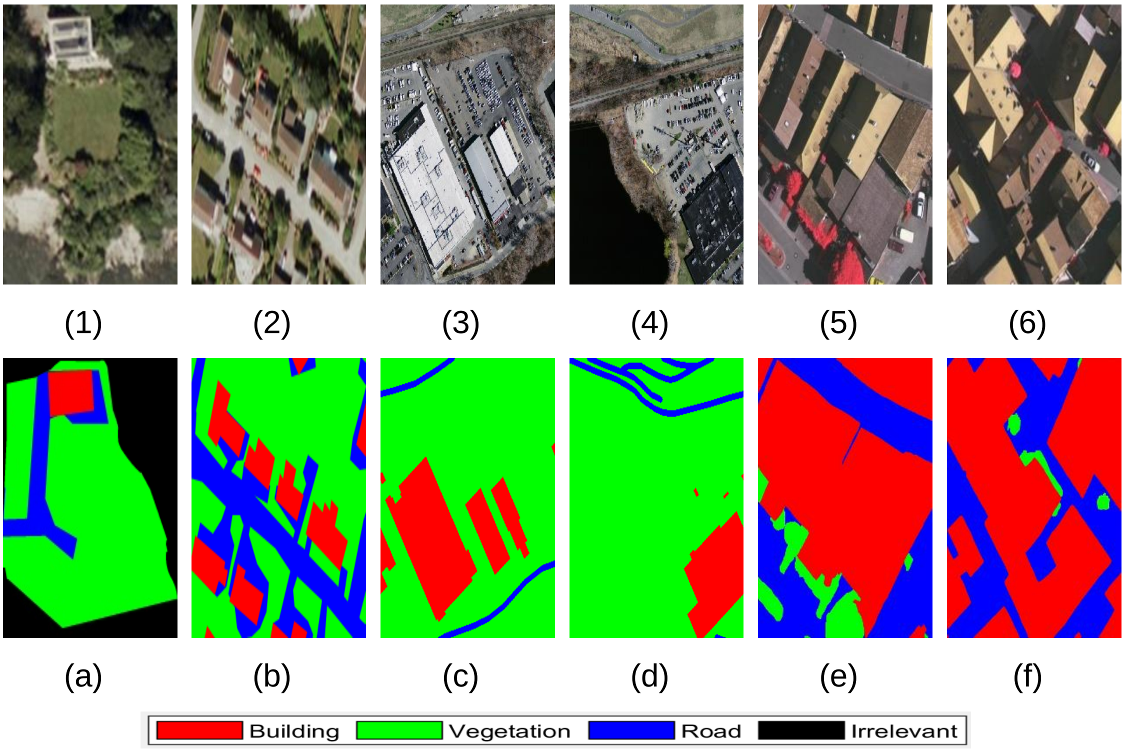


Figure 4.16: (1)-(6): Satellite images; (a)-(f): Respective ground truth images. (1) Cropped part from Feock satellite image (167×195 pixels), (2) Cropped and zoomed part from Feock satellite image (249×180), (3)-(4) Copied images from Toronto Roads and Greater Toronto Area (GTA) Buildings datasets, Canada [227] (300×300), (5) source: ISPRS 2D Semantic Labelling Contest, Vaihingen in Germany [228] (1447×1444), and (6) same source as (5) (1519×1514).

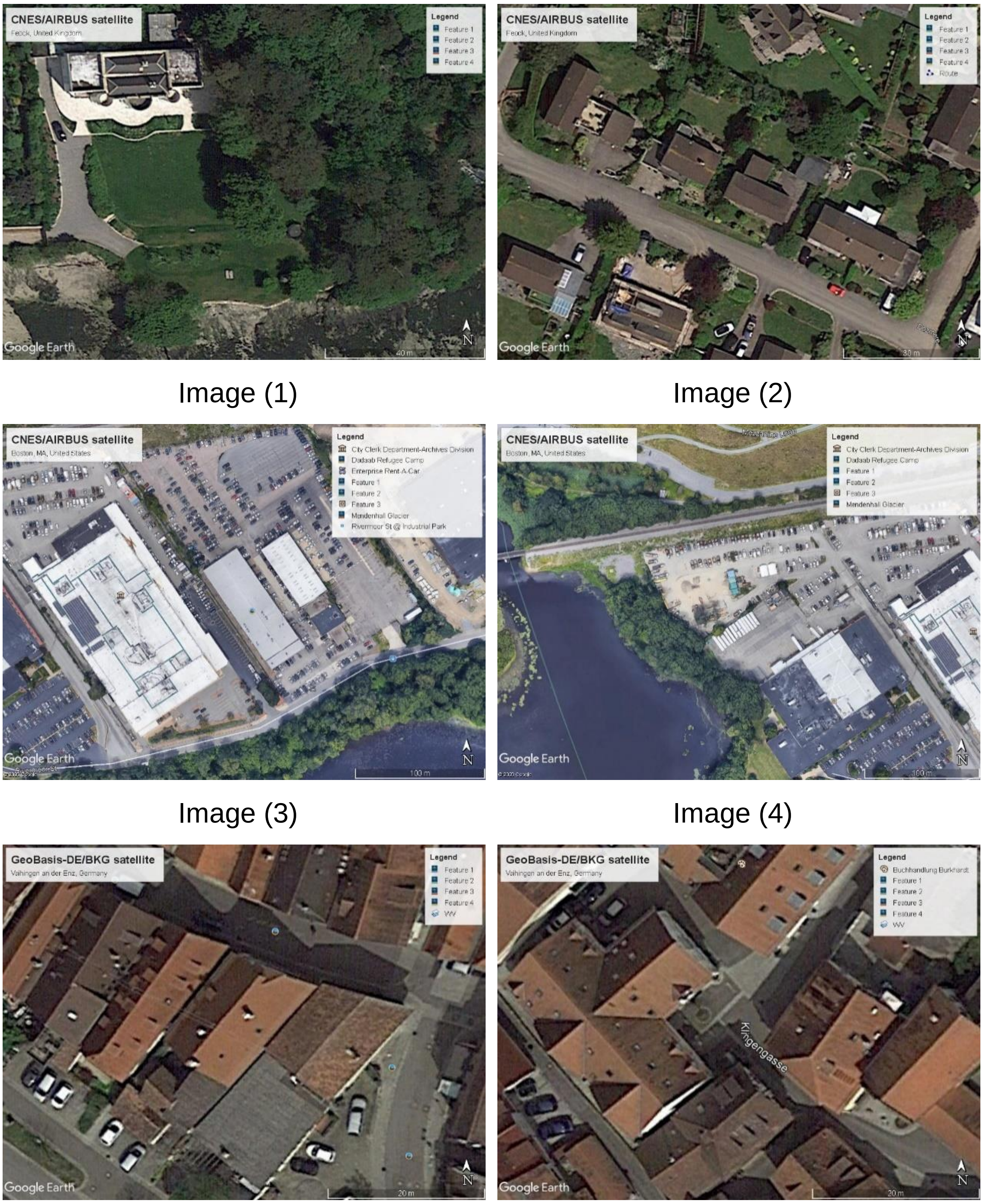


Image (1)

Image (2)

Image (3)

Image (4)

Image (5)

Image (6)

Figure 4.17: Images (1)-(6): Captured details of the six satellite images.

4.1.2.3 Feature Selection

This section performs a comparison between three kinds of features, as specified in Features Extraction section 3.3.2 of Chapter 3, to determine the most suitable kind of features for classification models. Testing of features is performed on the six images, selected in this section, by applying classification, based on eight selected ML classifiers, for three types of feature sets. The SLIC value of 10,000 is used for segmentation of images into objects, and most repeated class pixels, inside an object, is treated as an object label.

The performance comparison of the three kinds of features is shown, for all six images (shown in Figure 4.18), as bar plots. Bar graphs are drawn for the eight classifiers accuracy, where yellow bars represent accuracy values of texture-based features (listed in Table 3.3 of Chapter 3), red bars represent the accuracy of HSV-based features (listed in Table 3.2 of Chapter 3), and blue colour bars represent RGB colour-space-based features (listed in Table 3.1 of Chapter 3) results which are elaborated in section 3.3.2 of Chapter 3. It is clear, from the bar graphs of all images, that the set of 10 RGB colour-space-based features work best for classification of the 6 images, in most of the classifier results, since RGB features are mostly colour based and contain many different colours and shape-based properties of objects. On the other hand, HSV colour space features rely mostly on mere colour values, while texture features, although good in terms of box plot, are not enough alone to get a good discrimination between the objects, in this research area, as this process mostly relies on colour-based features. Hence RGB based features are the most suitable kind of features in this work scenario.

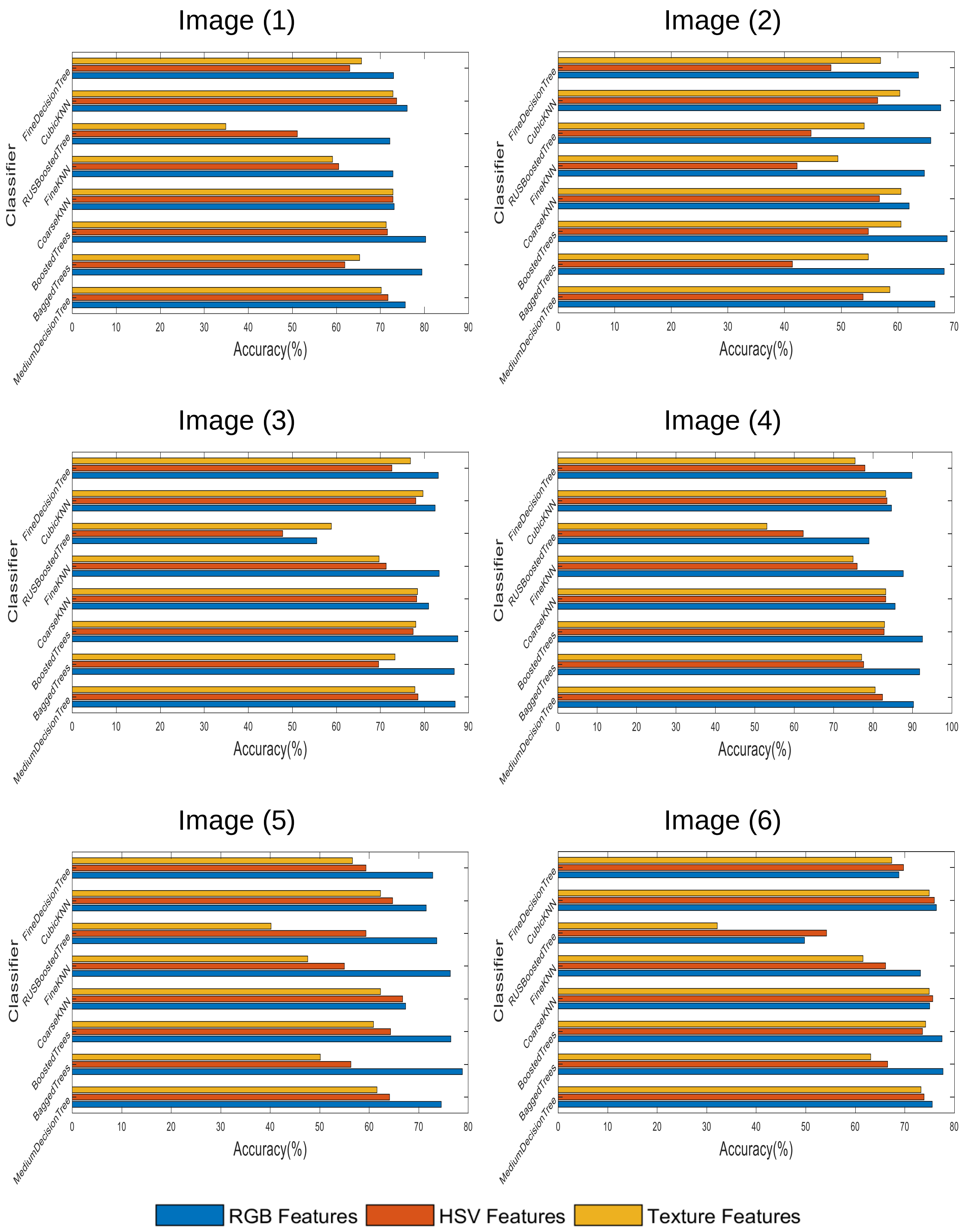


Figure 4.18: Comparison of three kinds of features performance for the six selected images and eight classifiers using SLIC 10,000.

Since the main area of research, for this study, is Feock image, all three kinds of features are also tested on this specific image, to confirm that the selected features work also well in this case. Figure 4.19 shows the performance comparison of the eight ML classifiers, applied on Feock, for all three kinds of features, using SLIC value of 10,000. The illustrated bar graph of Feock (Figure 4.19) verifies that RGB features work best for Feock image as well.

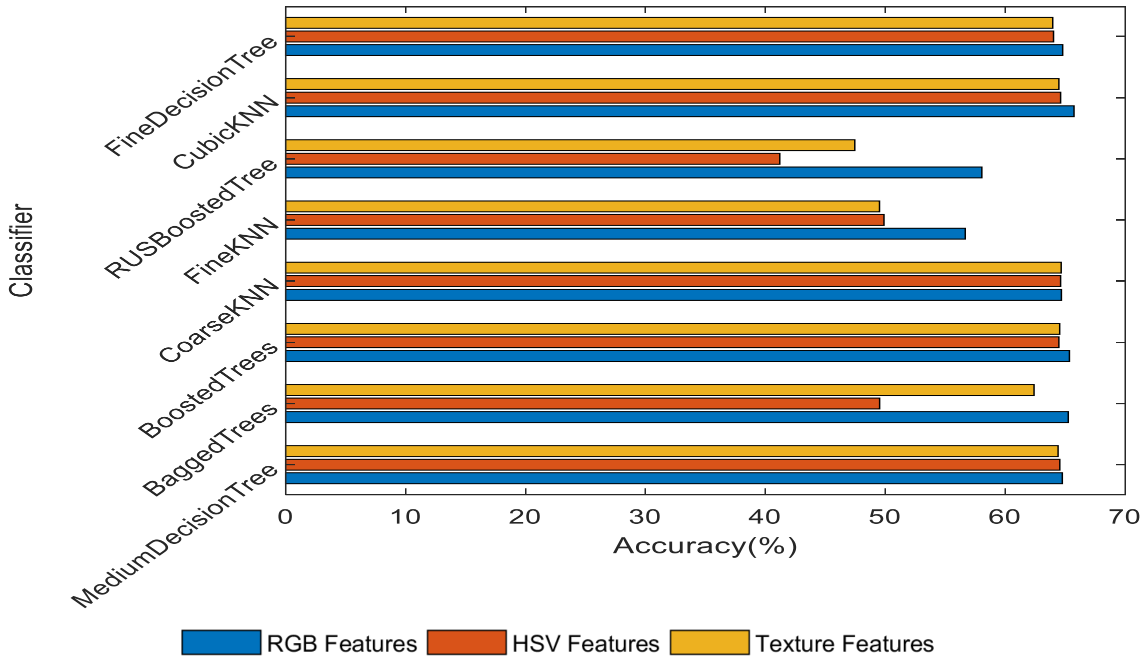


Figure 4.19: Accuracy comparison of the three kinds of features for Feock image for eight classifiers using SLIC 10,000.

4.1.2.3.1 Performance Comparison of the Classifiers

This subsection presents the classification results of the eight selected classification algorithms, on the test images (Figure 4.16), using the proven best suited RGB colour space-based features, to determine the best classification algorithm, for this research area, according to two approaches. The first approach follows the application of classification algorithms to the objects collected from pixel-based approach, where each pixel is considered as an object. In the second approach, the same set of classifiers are applied to the objects collected from object-based approach, where, the area was divided into regions based on different objects counts (SLICs).

As many machine learning algorithms use cross-validation concept for tuning [104] [229] [230], 5-fold cross-validation is also used here, for training and testing of the classifiers, rectifying the situation of limited data availability.

Following are general steps followed during the implementation of K-fold cross-validation process [231]:

1. Shuffling of data randomly.
2. Splitting of data into k parts/folds.
3. Applying for each part/fold:

    1. Keeping one-fold as test data.
    2. Using remaining folds as training data.
    3. Training the classification model with training data and testing on testing data.
    4. Saving the test results and moving to next fold.

4. Evaluation of results based on results from all folds.

All test fold results are combined, after testing and training in all folds, to calculate the overall system performance. The optimal classification algorithm and most suitable SLIC for segmentation, is selected by applying Pareto and Knee point analysis which provide most compromised parameter values in terms of more than one attributes. Pareto's method is used to collect all the candidate points, having minimal distance from at least one of the parameters under consideration, known as Pareto optimal points or trade-off points [232]. The purpose of Pareto Analysis is to highlight the most important (dominant) points, among all the points, considering more than one attribute at a time. The dominant points of the Pareto Front are then used for an automatic selection of a single preferred solution (Knee point). The Knee point, in this study, aims to determine the best compromise among all the Pareto points, in order to indicate the most effective classifier and SLIC value. Several algorithms have been developed in literature, to find the Knee point in the Pareto Optimal Front [233-235]. Figure 4.20 shows a flowchart of feature extraction, classification and Pareto analysis steps, for object-based and pixel-based classification approach.

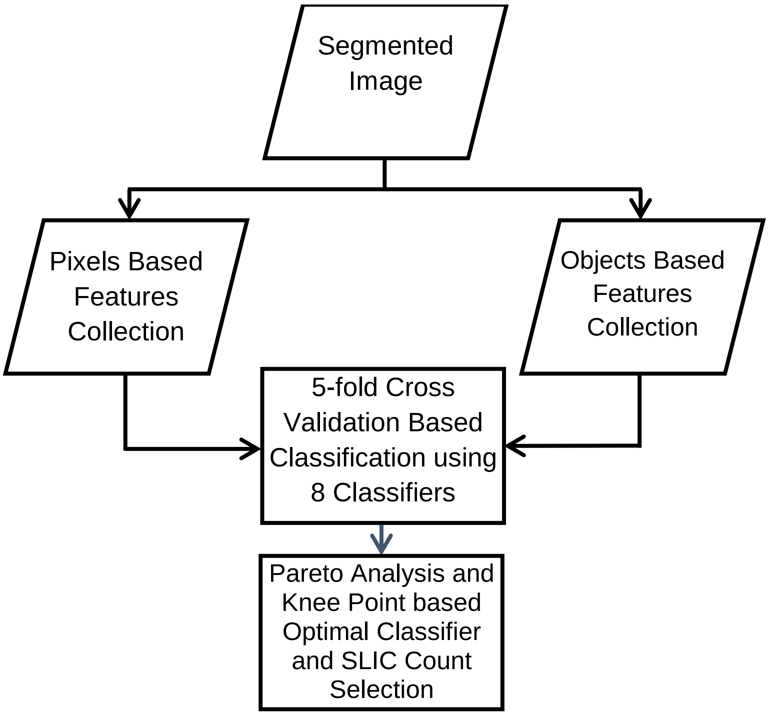

Segmented Image
Pixels Based Features Collection
Objects Based Features Collection
5-fold Cross Validation Based Classification using 8 Classifiers
Pareto Analysis and Knee Point based Optimal Classifier and SLIC Count Selection

Figure 4.20: Flow process for optimal classifier and SLIC count selection.

4.1.2.3.2 Accuracy and Runtime Results Assessment

Table 4.7 shows overall accuracy and processing time comparison of objects-based and pixels-based approach, for all selected classification algorithms, which are tested on six images, being used in this section. This table also shows the best classification algorithm, along with best SLIC count for segmentation, in the object-based and in the pixel-based approach. Table 4.7 demonstrates that, for all test images, object-based classification gives better mean classification accuracy value (93.7% versus 88.5%) and superior mean computational time (869s versus 10,855s), when compared to pixel-based classification. Thus, the classification process requires over 12 times less total runtime, to complete the execution of the algorithms on all the test images, according to the object-based approach, compared to the pixel-based approach, while it also provides notably better accuracy.

Furthermore, the results in Table 4.7 show that 'Bagged Tree' classification algorithm provides maximum accuracy value, among all eight classification algorithms used, when this classifier is applied for 10,000 objects segmented, by using the SLIC superpixels. In spite of the intra-class variability and the considerable number of distinct sources of data acquisition, within the class sample in the utilised images, detection of targeted classes seems to be implemented robustly and with high accuracy. Such accuracies exceed 92% and

86%, in five out of six images, when using object-based and pixel-based paradigms, respectively.

Table 4.7: Overall accuracy and runtime comparison of object-based classifiers and pixel-based classifiers.

	5-fold cross-validation							
Data set	Object-based approach				Pixel-based approach			
	Best classifier	Best object count	Overall Accuracy (%)	Elapsed time[s]	Best classifier	No. of pixels	Overall Accuracy (%)	Elapsed time[s]
Image 1	Bagged Tree	**10,000**	**99.2**	201	Cubic KNN	32,565	86.8	**148**
Image 2	Bagged Tree	**10,000**	**86.4**	**322**	Coarse KNN	44,820	81.3	431
Image 3	Bagged Tree	**10,000**	**93.6**	**263**	Coarse KNN	90,000	90.1	549
Image 4	Bagged Tree	**10,000**	**95.4**	**278**	Coarse KNN	90,000	91.7	547
Image 5	Bagged Tree	**10,000**	**95.2**	**2,066**	Bagged Tree	2,089,468	91.6	33,828
Image 6	Bagged Tree	**10,000**	**92.4**	**2,084**	Bagged Tree	2,299,766	89.8	29,627
Mean		**10,000**	**93.7**	**869**		774,437	88.5	10,855

SLIC method creates a complex tree structure, which leads to good training and minimum error results [35], as shown in Figures 4.21 and 4.22, where a comparison of classification accuracy and the processing time is shown, for the six images, using different SLIC count segmentations. Each curve, in these figures, represents a different image, where there is a prominent rise in accuracy and processing time curves, meaning that these measures are increasing as the SLIC count, on an image, increases. Figure 4.21 illustrates that there is an immediate increase in accuracy, as the object count increases in all the six images. However, the increased number of objects, in an image, requires significantly more runtime, compared to the other lower count instances, used in this research.

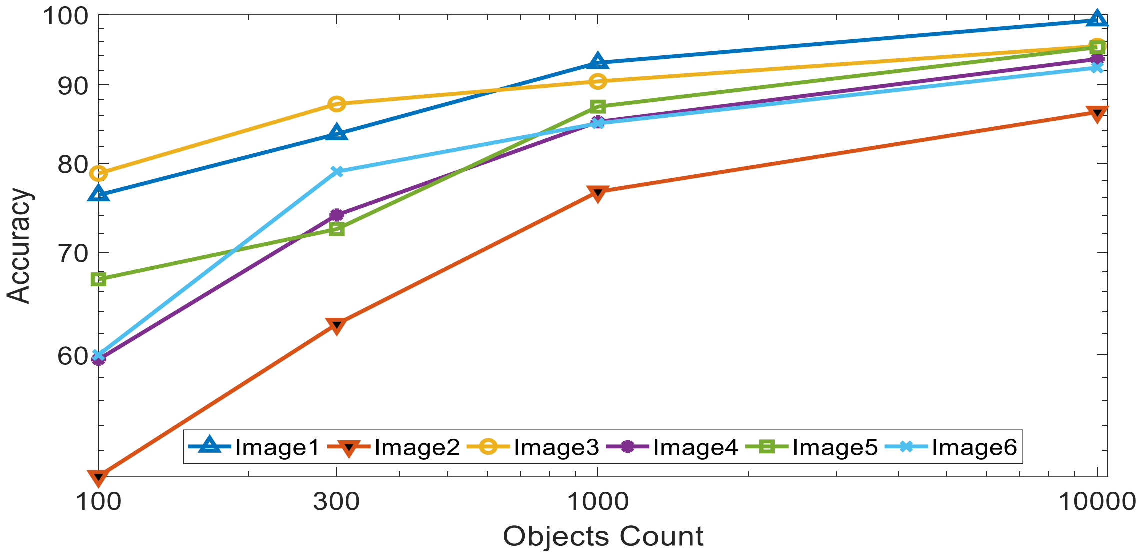


Figure 4.21: Objects count versus classification accuracy.

Figure 4.22 shows that the runtime mostly increases, with the increase in objects count. Therefore, the appropriate number of objects is a parameter set, depending on available computational resources.

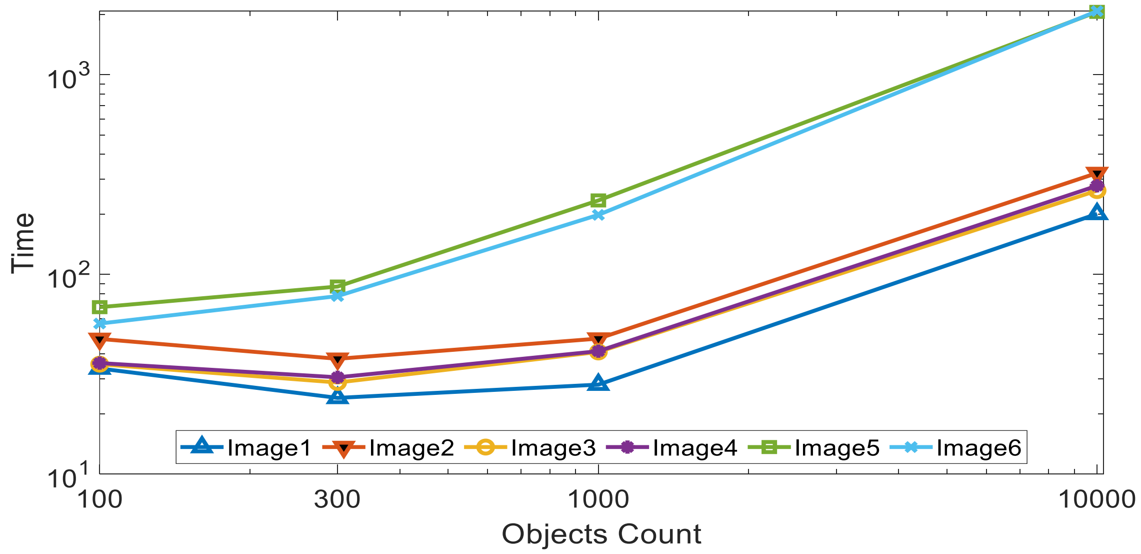


Figure 4.22: Objects count versus computational time.

Figure 4.23 uses a scatterplot to demonstrate the comparative performance of the eight classifiers, regarding different attributes, involving error and time, illustrating a 3D visualisation of the object-based classification results, as error values plot against the number of segmented objects and the runtime, for the six images studied in this section. Since the aim is to achieve less error and lower time values, the points at the bottom and right side of the plot are the preferred ones, producing the optimal SLIC and runtime values. The results from these plots cannot be well determined, because sometimes one SLIC is good, in terms of time, but not good in terms of performance and vice versa. For this reason, Pareto optimal point analysis is performed, in the next steps, to derive the optimal and best compromised attribute, from these results.

Image 1

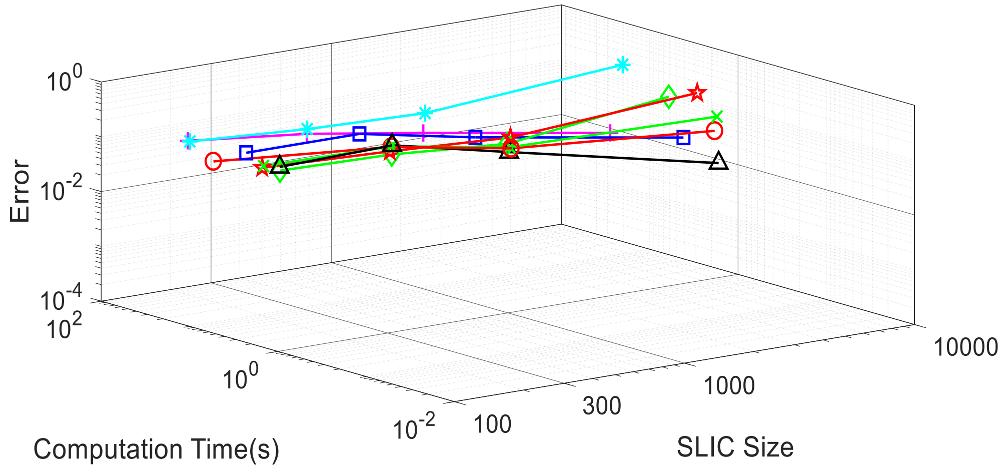


Image 2

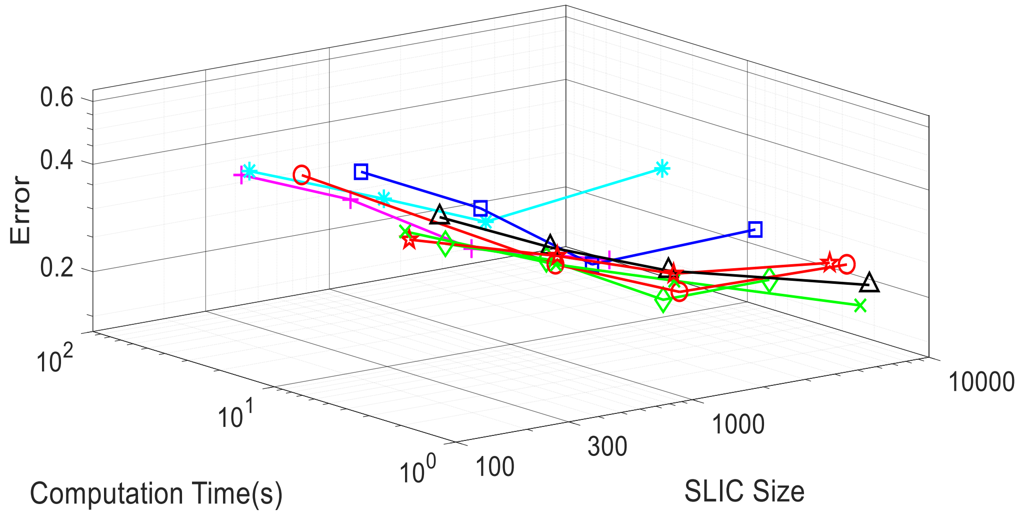
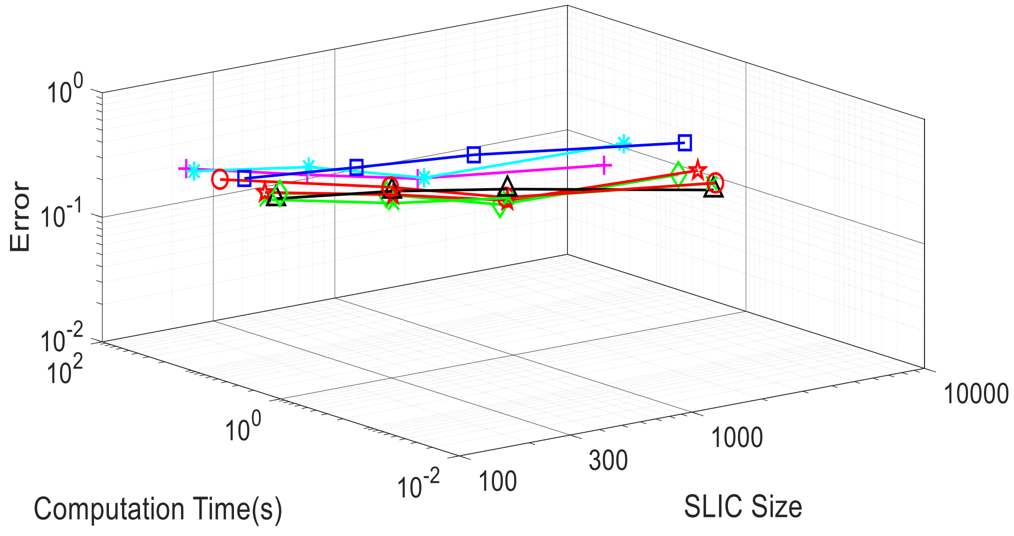


Image 3



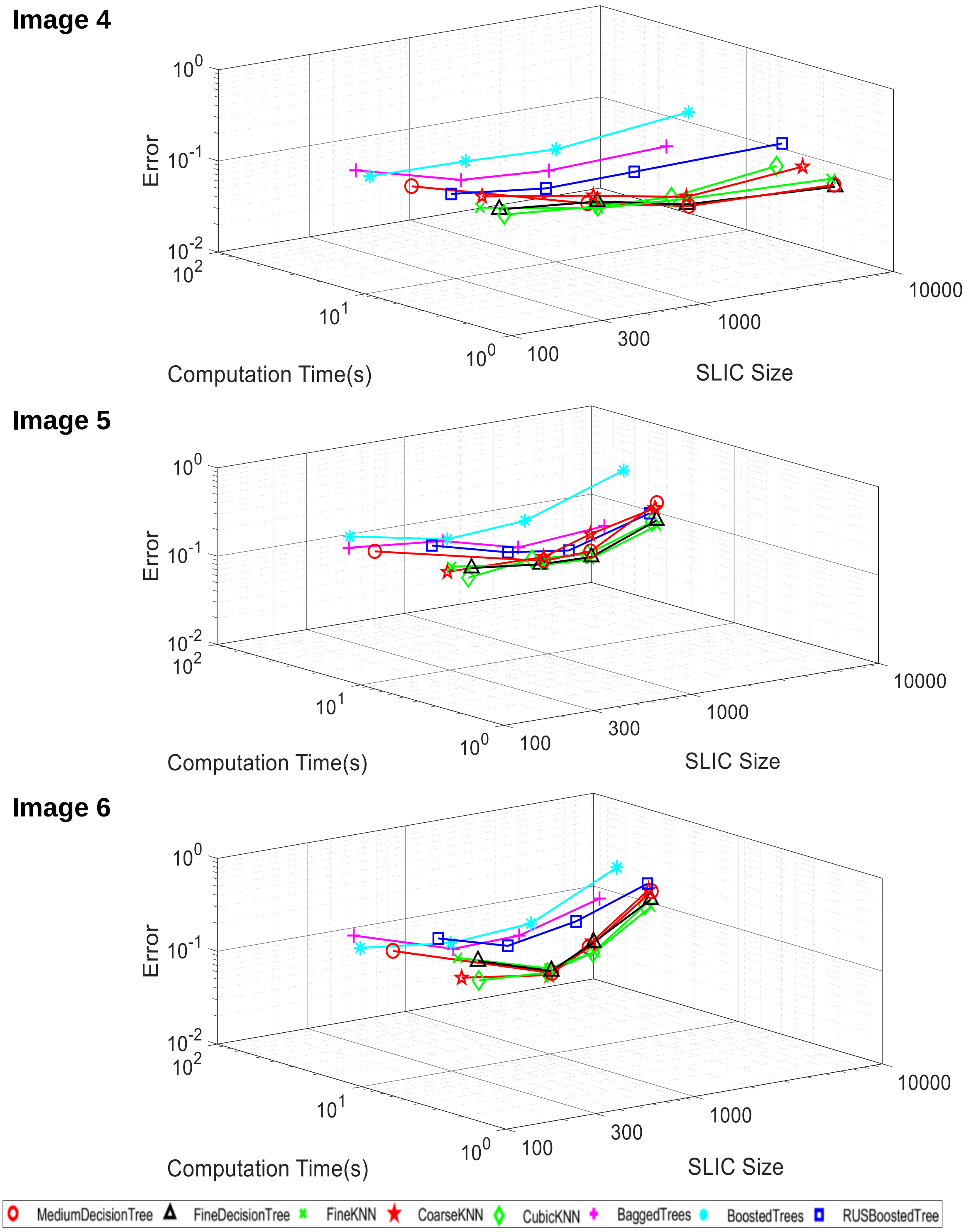


Figure 4.23: 3D scatterplot analysis of classification error versus computation time and SLIC count, for the object-based approach.

Figure 4.24 represents the resultant errors of the pixel-based classification method, drawn against the execution time, for the six test images, using eight classifiers. The plots in (Figure 4.24) show that some points are optimal, in terms of classification time, while others, in terms of classification error. For example, in the scatter plots of Figure 4.24, 'Medium' and 'Fine Decision Trees' classifiers are fast to predict, but they have comparatively low predictive mean accuracy. 'Fine', 'Coarse' and 'Cubic Nearest Neighbour' classifiers are relatively fast predictors and they also have good predictive mean accuracy. 'Bagged Trees' ensemble classifier (a combination of multiple classifiers) has good mean accuracy, but low mean speed, because it often needs many learners to fit the data, which is time-consuming. 'Boosted' and 'RUS Boosted Trees' ensemble classifiers do not have a high accuracy, and have low mean speed, as expected. However, these two classifiers are capable of giving good mean accuracy with the addition of more versatile data for training. Based on these observations, there is no specific classifier, which is optimal in terms of both time and performance, which is why the optimum classifier cannot be determined from these plots, but rather through another method, like Pareto Analysis.

Image 1

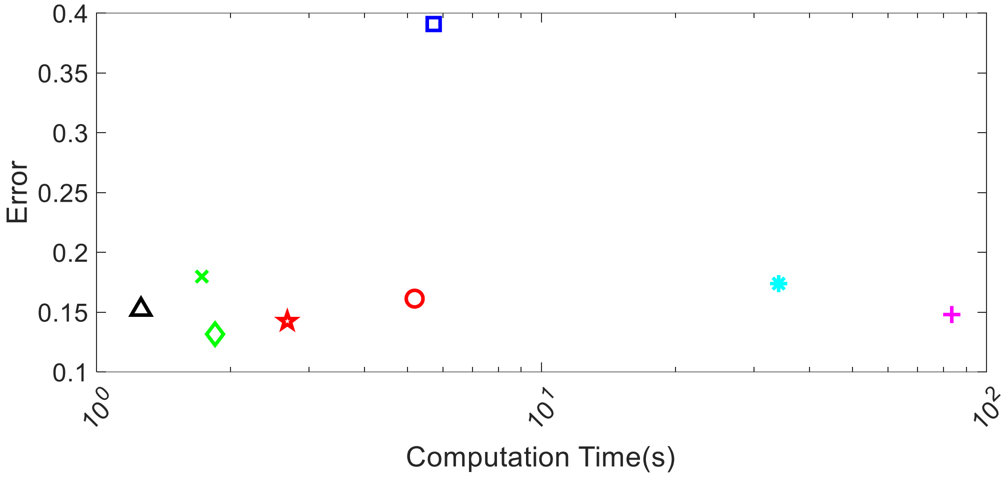


Image 2

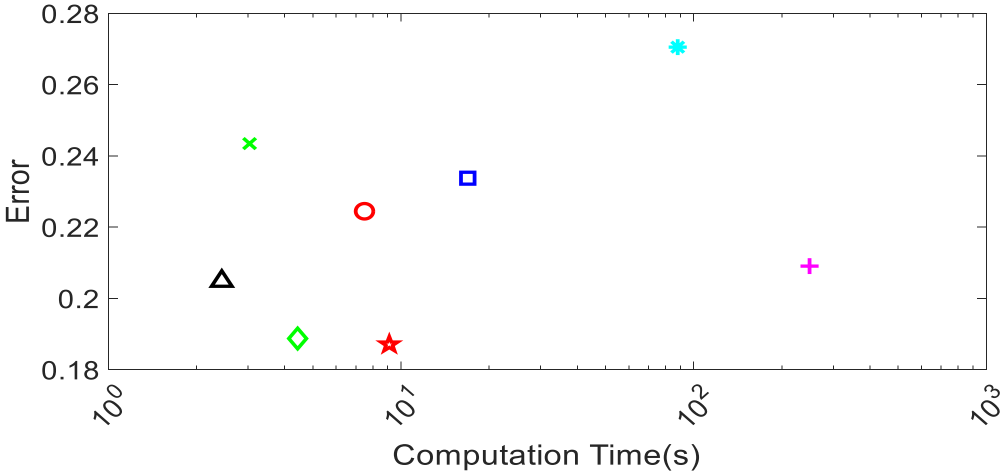


Image 3

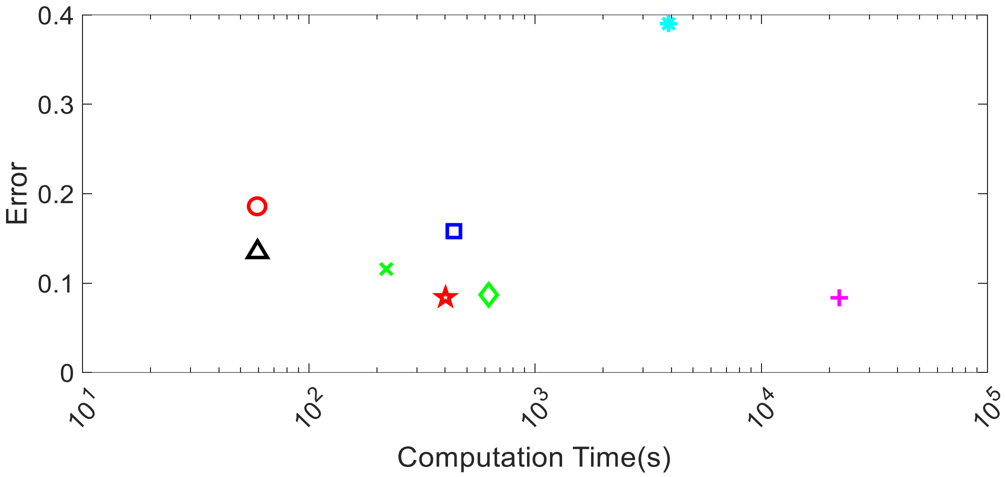


Image 4

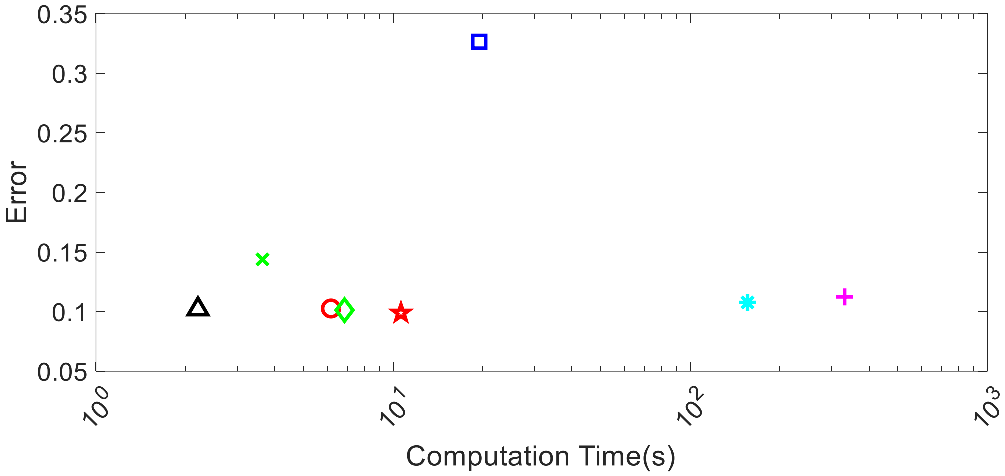
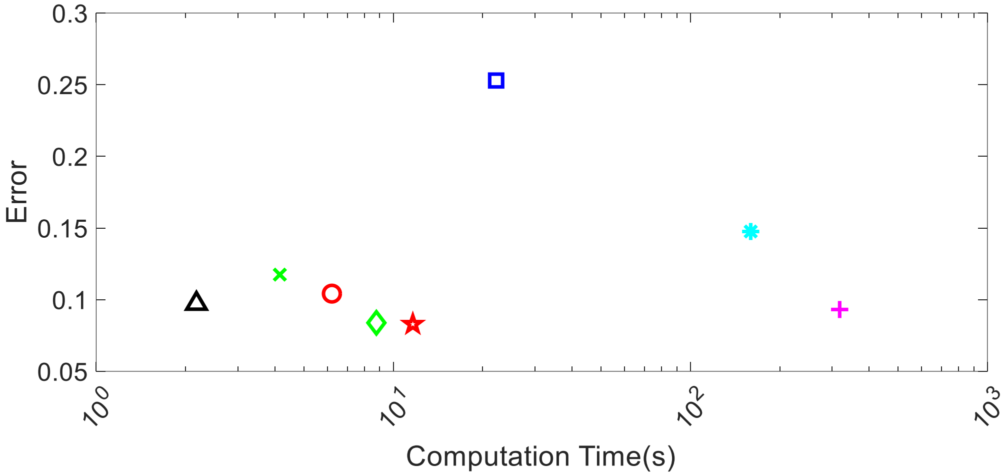


Image 5



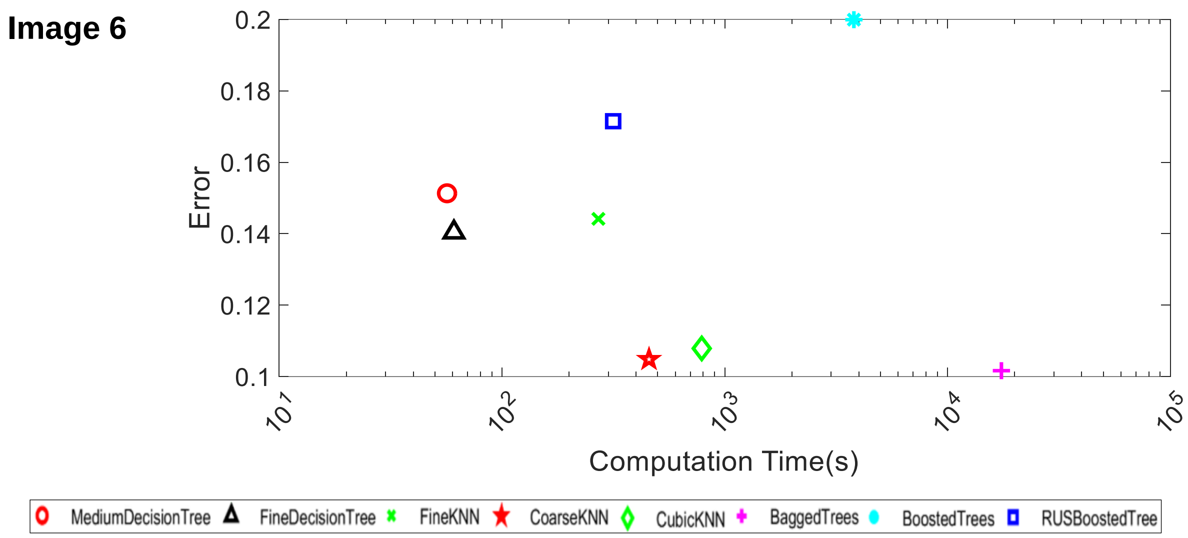


Figure 4.24: Classification error versus computation time plot, for the pixel-based approach, for eight classifiers.

4.1.2.3.3 Pareto Optimality Analysis Based Selection

Analysis of results shown in Figures 4.23 and 4.24 shows that there is no specific unique point (best classifier) for both object-based and pixel-based classification methods, neither is it possible to visually select one classifier, which can give best results in terms of both time and performance. This issue leads to the use of the popular Pareto Optimality, as an analytical optimisation method, to determine the classification approach with the best performance, considering the classification errors and execution time [236-238].

Figure 4.25 presents Pareto Optimality for the six test images, where all the selected Pareto points/classifiers, for object-based and pixel-based classification, are connected by a curve separately, and non-selected points are presented as scattered points in the graph. The red Pareto curve, which is denoting the object-based approach, includes the selected candidate points, out of 32 points (four SLIC object counts for eight classifiers giving total 32 points), while the blue Pareto curve has got the selected points, out of eight points, in the pixel-based approach. Figure 4.25 demonstrates that the Knee point (black arrow) is not necessarily the best point among the eight classifiers, for all six tested images, but is instead the most suitable point, for this research problem, among all Pareto points of the object-based approach (red curve). Also, the selected Knee point belongs to the object-based approach curve, because it has superior performance in accuracy and runtime, compared to the pixel-based approach (blue curve). The optimal parameter error values, for the six images, are 0.0085, 0.1661, 0.07786, 0.05607, 0.1606 and 0.1699, respectively. The optimal runtime parameter values(s), for the six images, are 1.7, 2.4, 2.3, 2.5, 5.1, and 5.0, respectively.

Image 1

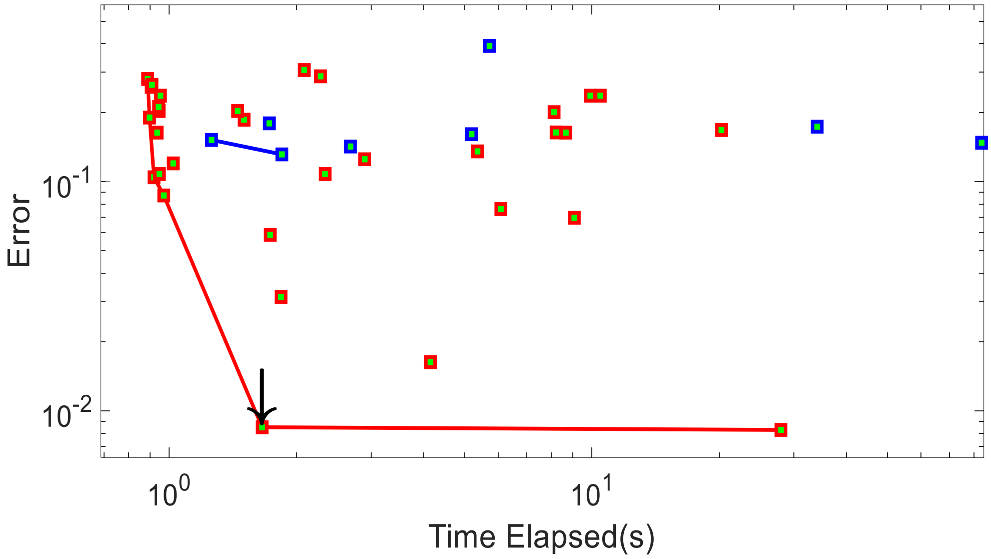


Image 2

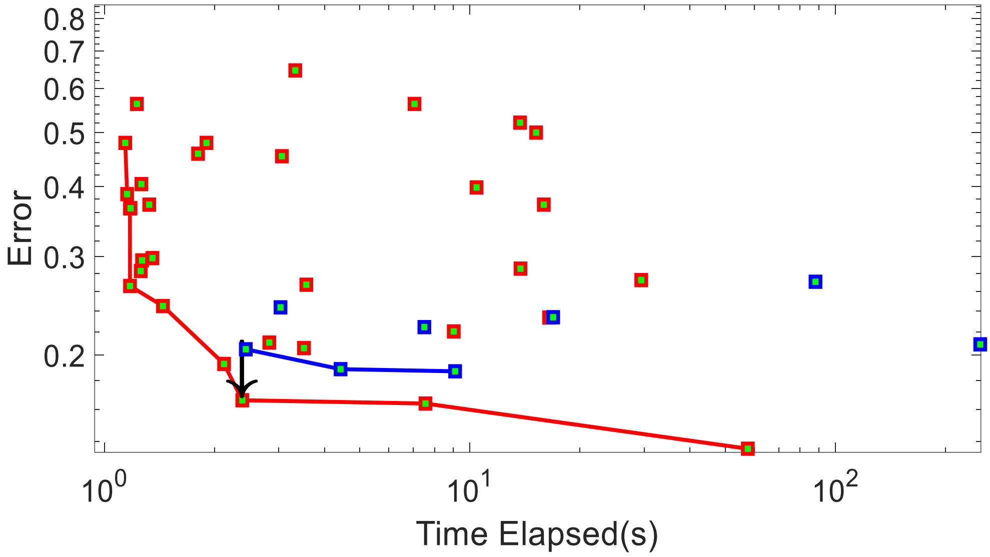
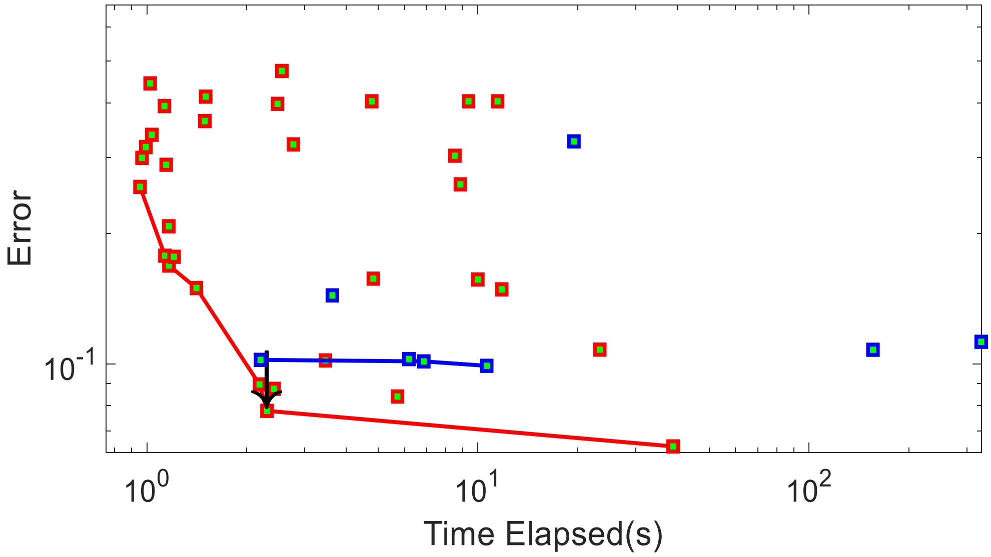


Image 3



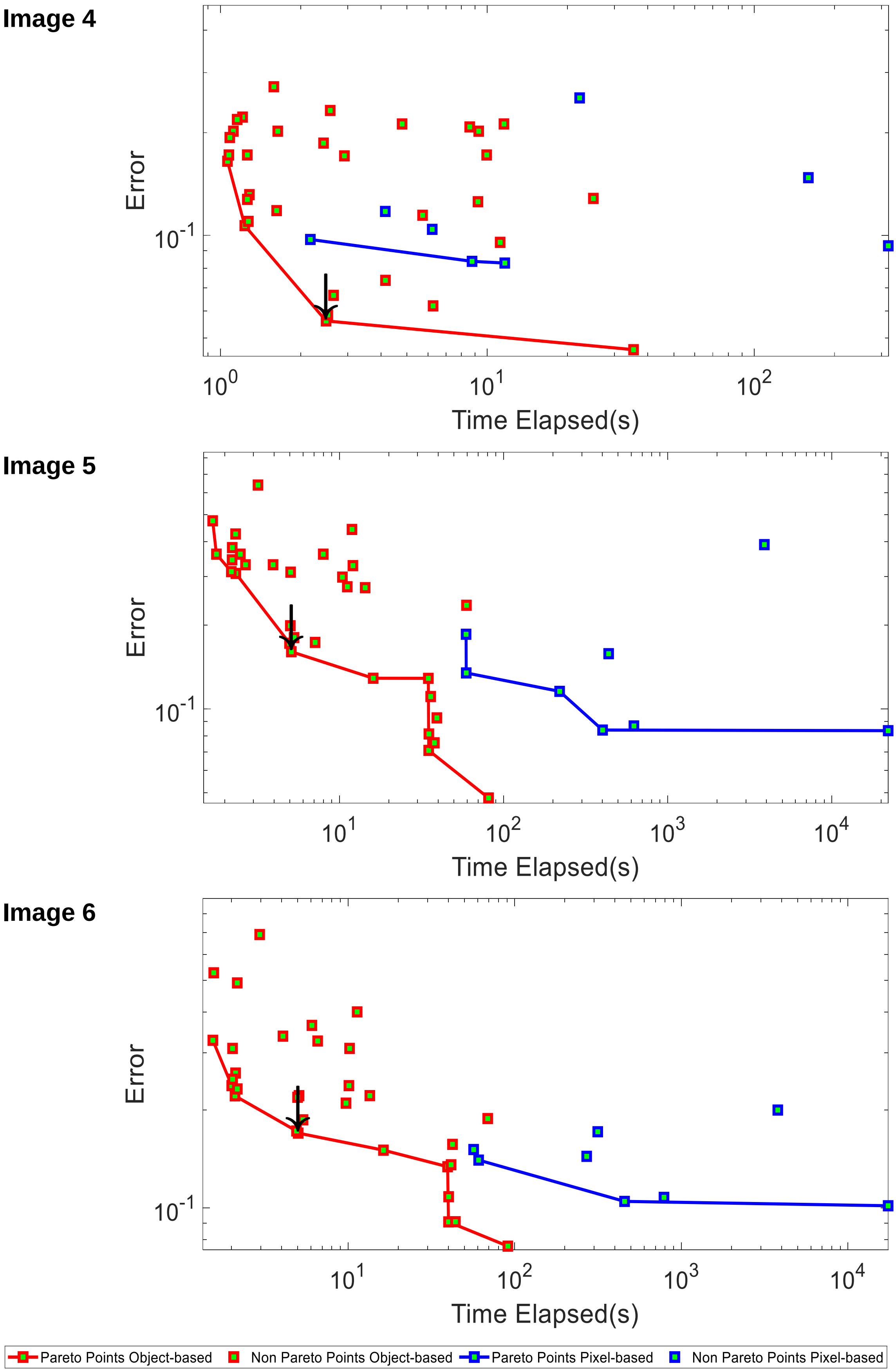


Figure 4.25: Determining the Knee point for Pareto curves of the object and pixel-based approaches.

4.1.2.4 Ensemble: A Combination of Classifiers

There are evident possibilities to improve the results of the above classification. One option is selecting only a few Pareto front classifiers, on the basis of diversity and accuracy, according to the ensemble classifier concept, instead of selecting all the classifiers. For the ensemble, votes are taken from more than one classifier and the most voted classification result is selected as the final classification prediction result. Figure 4.26 shows the step-by-step flow process, followed for the weighted score ensemble, used in this study.

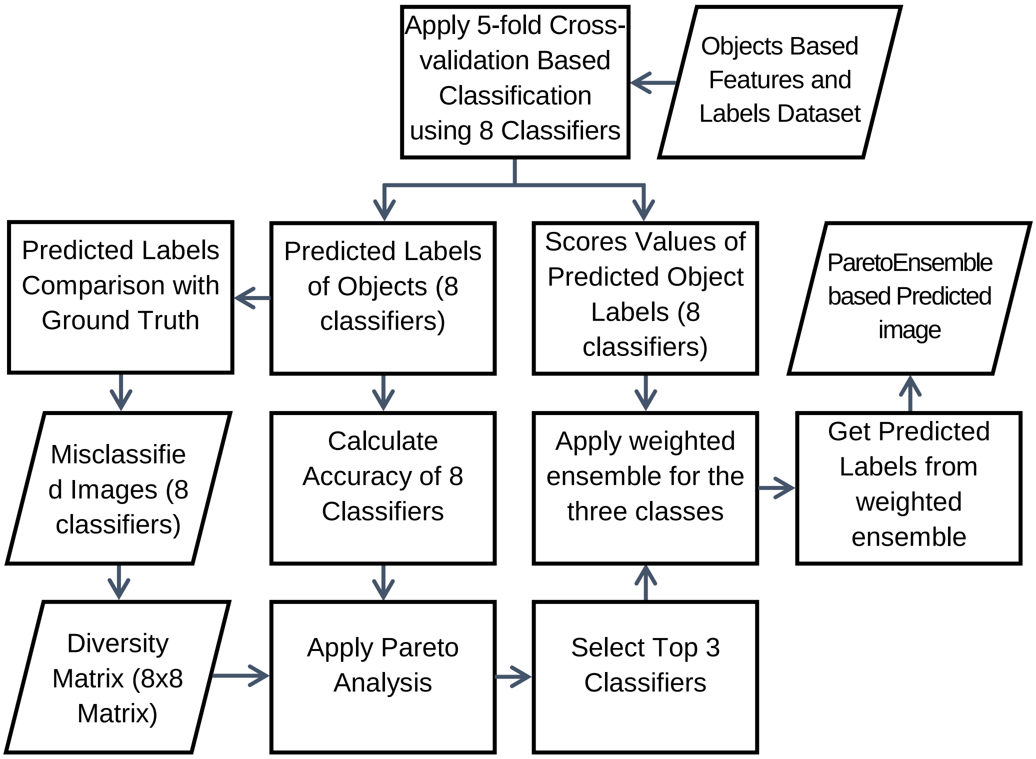

Apply 5-fold Cross-validation Based Classification using 8 Classifiers
Objects Based Features and Labels Dataset
Predicted Labels Comparison with Ground Truth
Predicted Labels of Objects (8 classifiers)
Scores Values of Predicted Object Labels (8 classifiers)
ParetoEnsemble based Predicted image
Misclassifie d Images (8 classifiers)
Calculate Accuracy of 8 Classifiers
Apply weighted ensemble for the three classes
Get Predicted Labels from weighted ensemble
Diversity Matrix (8x8 Matrix)
Apply Pareto Analysis
Select Top 3 Classifiers

Figure 4.26: Flow process for the weighted score ParetoEnsemble classification.

4.1.2.4.1 Diversity-based Selection of Candidate Ensemble Classifiers

To increase the performance of eight individual classifiers, a number of exclusively top accuracy classifiers are combined to create an ensemble classifier, providing even more accurate and reliable predictions for test data, compared to the predictions of any individual classifier. In the ensemble, the prediction results of selected individual classifiers are combined, through a process of voting, instead of using all eight classifiers. This way, a more accurate ensemble classifier is developed, compared to any individual classifier, where the decision is taken from a single one. In the selection of the classifiers, to be part of the ensemble classifier, the idea of diversity has been incorporated, to determine the most diverse ones, in terms of classification, and then Pareto

Analysis is applied on diversity and classification accuracy, to make the final decision on the optimal classifiers selection.

The idea of diversity-based selection is to focus on the classifiers that are much diverse from one another, in classification method, so that a false classification by one classifier can be remedied by another one in the ensemble. Diversity estimation of classifiers is based on misclassified images, as derived from each classifier, by comparing the labelled ground truth and its predicted image, as shown in Figure 4.27 for Feock image. The rightmost image in Figure 4.27 represents an example of misclassified image, where all-black colour pixels depict unwanted/irrelevant area, red, green and blue coloured pixels represent correctly classified pixels, while all yellow coloured pixels represent incorrectly classified pixels, in Feock image, after applying 5-fold cross-validation classification, using SLIC 10,000 and Bagged Trees classifier. The misclassified images, as resulted from the eight classifiers, are then compared to each other, and diversity score value of each classifier is calculated in relation to all other seven classifiers.

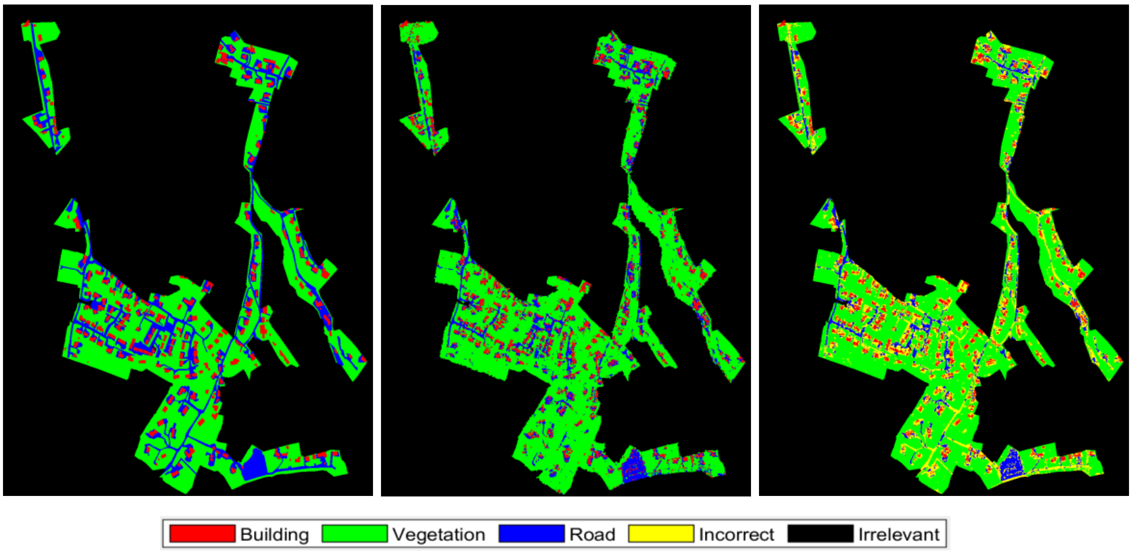


Figure 4.27: Bagged tree results of SLIC 10,000 Feock image in terms of ground truth image (left), predicted image (middle) and coloured misclassified image (right).

In the calculation of diversity score, D_i={D_1, D_2, D_3,..., D_n} are considered as the labelled datasets, from classifiers under consideration. Table 4.8 shows the outputs from a classifier D_i and another classifier D_k, presented as an N-dimensional matrix, where the 4 values (N^{11}, N^{10}, N^{01} and N^{00}) show the correct and incorrect prediction count, from both classifiers. For example, N^{11} represents the number of samples, correctly predicted by both classifiers; N^{10} represents the number of samples, correctly predicted by classifier D_i and falsely predicted by classifier D_k; N^{01} represents the number of samples, falsely predicted by classifier D_i and correctly predicted by classifier D_k; N^{00} represents the samples, falsely predicted by both classifiers. In the diversity score estimation, if two classifiers have both predicted a pixel label right or wrong, then it is ignored (not counted). However, if one provides a correct prediction, while the other a wrong one, then it is taken into account, represented by N^{01} and N^{10}.

Table 4.8: A 2x2 matrix showing relationship between two classifier outputs [130], where i and k represent two classifiers.

	$D_k correct\ (1)$	$D_k\ wrong\ (0)$
$D_i correct\ (1)$	N^{11}	N^{10}
$D_i\ wrong\ (0)$	N^{01}	N^{00}

A modified form of the disagreement measure has been used in the present work, to quantify the diversity measure between two classifiers, as shown in Equation (4.3). In this process, all these pixels, which one classifier has predicted correctly, while the other one has falsely predicted, are counted. Thus, the total count of non-matching pixels, in two misclassified images, is called the diversity score of the two classifiers [130].

$$Diversity_{i,k} = N^{01} + N^{10} \tag{4.3}$$

Since there are eight classifiers, there is an 8x8 matrix that describes diversity for all the classifiers as shown in Table 4.9. The first row of the 8x8 diversity matrix represents the diversity score of the first classifier in relation to all other classifiers. The second row is for the diversity score of the second classifier in relation to all others, and so on. The 0 diversity score values, at the diagonal location of diversity matrix, represent the diversity of each classifier to itself, which is null, as expected, because both misclassified images are the same.

Table 4.9: A sample 8x8 diversity matrix representing the diversity score of each classifier in relation to other classifiers, using SLIC 10,000 of Feock image.

	Medium Decision Tree	Bagged Tree	Boosted Tree	Coarse KNN	Fine KNN	RUS Boosted Tree	Cubic KNN	Fine Decision Tree
Medium Decision Tree	0	223763	126126	189895	453446	383052	317605	165649
Bagged Tree	223763	0	204189	202390	408883	371829	274996	197350
Boosted Tree	126126	204189	0	142997	447602	442854	328871	152195
Coarse KNN	189895	202390	142997	0	437597	459487	317288	181906
Fine KNN	453446	408883	447602	437597	0	523668	480829	444683
RUS Boosted Tree	383052	371829	442854	459487	523668	0	337003	390259
Cubic KNN	317605	274996	328871	317288	480829	337003	0	308032
Fine Decision Tree	165649	197350	152195	181906	444683	390259	308032	0

In the next step, the mean diversity score of each classifier is calculated, by deriving the mean value of each row in the 8x8 matrix (mean of same colour values in 8x8 matrix), producing an 8x1 vector (for every SLIC). Next, Pareto Analysis is applied on the mean diversity score and classification accuracy values of the eight classifiers, to select some classifiers, among all, which are good candidates, in terms of both diversity and accuracy, to be further used in the ensemble classifier. Towards this end, both diversity and accuracy matrix are given to Pareto Analysis function as input, leading to an output consisting of some selective classifiers, out of the eight classifiers. This selection is based on optimum point values, for both diversity and accuracy, meaning that the classifiers/points selected are expected to be towards the upper right side of the plot (Figure 4.28). This figure displays selected and non-selected points, in Pareto plot, where all points, connected with the red line, are the most suitable candidate classifiers, in terms of both accuracy and diversity score. These selected classifiers are used to design a new ParetoEnsemble classifier, as described in the next step.

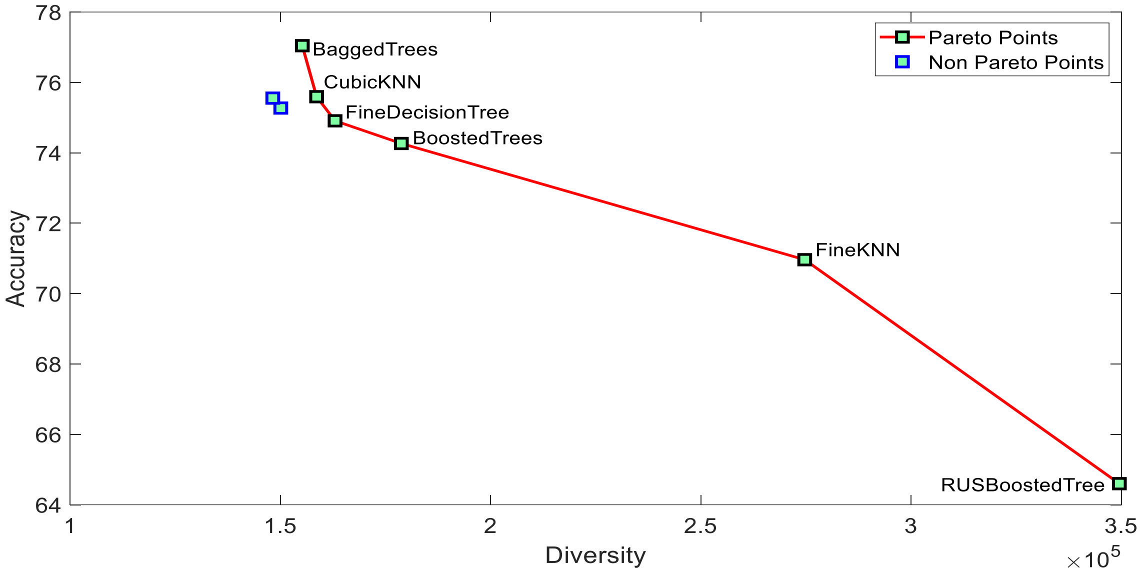


Figure 4.28: Pareto Analysis plot of diversity versus accuracy of the eight classifiers for Feock image.

Following, three different sets of top accuracy classifiers are selected from the Pareto derived classifiers, including 2, 3 and 5 top accuracy classifiers for the six images, to determine the best combination count of classifiers for the ParetoEnsemble, as shown in Table 4.10. Sets 2, 3, or 5 classifiers are selected for the ParetoEnsemble, out of total eight classifiers, because Pareto usually provides less than or equal to 6 classifiers, out of eight classifiers. If there are more than desired number of Pareto points, then the ones having highest accuracy will be selected. For example, in case of top 3 selection and total 6 Pareto points, top 3 accuracy points out of 6 Pareto selected points will be considered. As expected, a set of odd number of classifiers gives a fair decision, during voting for a prediction label, in case of tie. The most voted decision, from these top accuracy classifiers, is considered as the final prediction result for each sample and acts as the comparison measure of different sets of classifiers. Then, the accuracy of each set ParetoEnsemble classifier is calculated, by comparing the respective prediction results to the actual results from the ground truth. Table 4.10 shows the comparison between topmost individual classifier and ParetoEnsemble classifiers, regarding accuracy, in the six images, for SLIC value 10,000, as deducted from Table 4.7 that SLIC 10,000 shows the best results in all images. Table 4.10 illustrates that Top 3 accuracy classifiers-based ParetoEnsemble is the best choice, compared to others. Specifically, in image 1, the accuracies of Top 2, Top 3 and Top 5, are 99.0%, 99.4% and 98.5%, respectively. However, it is evident that the accuracy results of this ParetoEnsemble method are not better than the accuracy values of the best individual classifiers, for some images. For example, the Top 3 accuracy in image 3 is 93.2%, while the best accuracy of individual classifiers is 93.6%. This finding has led to the incorporation of the weighted sum notion, to the ParetoEnsemble classifier, in order to enhance its performance.

Table 4.10: Performance comparison of the best individual classifier versus Top 2, Top 3 and Top 5 ParetoEnsemble classifiers, for SLIC 10,000 in the six images.

Image	Best Individual Accuracy (%)	Top 2 ParetoEnsemble Accuracy (%)	Top 3 ParetoEnsemble Accuracy (%)	Top 5 ParetoEnsemble Accuracy (%)
Image 1	99.2	99.0	**99.4**	98.5
Image 2	**86.4**	84.7	86.3	83.2
Image 3	**93.6**	90.7	93.2	92.7
Image 4	**95.4**	93.4	95.2	94.1
Image 5	**95.2**	86.2	94.5	93.0
Image 6	92.4	90.9	**92.5**	89.2

Since the specific area of study is Feock, it is essential to ensure that the selected parameter values work fine in this case, as well. For this purpose, these parameter selections are also tested on Feock, before proceeding to the weighted sum-based ParetoEnsemble. Feock image is tested with many SLICs values, by applying 5-fold based cross-validation, because it is a bigger image, compared to the six images prior tested. Table 4.11 demonstrates a comparison of the Top 3 accuracy classifiers-based ParetoEnsemble, to all Pareto selected classifiers-based ParetoEnsemble, regarding classification results, for multiple classifiers, using 10 RGB features, as determined at the feature selection process step. This table of the results for best individual classifiers shows evident similarities to the results in the case of the six images. According to the table data, the Top 3 is better than all Pareto selected classifiers-based classification schemes, as in many of the six images earlier. Nonetheless, the results of individual classifiers are more accurate than those of Top 3, for some SLICs, which is the reason for the introduction of the weighted sum approach.

Table 4.11: Feock image: comparison results for the best individual, all Pareto selected, and Top 3 Pareto selected classifiers for ten features-based classifications.

SLIC	Best Individual Accuracy (%)	ParetoEnsemble Accuracy (%)	Top 3 ParetoEnsemble Accuracy (%)
1000	**49.25**	43.29	41.79
10,000	63.94	63.94	**70.17**
15,000	72.43	72.51	**76.25**
20,000	72.51	**72.80**	72.69
25,000	**74.32**	72.43	72.60
30,000	74.83	**74.91**	74.06
35,000	75.36	73.70	**76.92**
40,000	75.97	**76.28**	76.18
45,000	77.01	75.78	**77.92**
50,000	77.04	76.27	**77.36**

According to Table 4.11, the classification accuracy in the case of Feock image is not as high as in the six images. This is due to the considerable size difference between the Feock image and the six images considered for evaluations, affecting the properties of features used, due to SLIC based segmentation, where objects can have different nature of feature values, based on their size due to the inclusion of other class pixels in bigger objects. Solving this issue, three new L*a*b colour space-based features are added to the dataset [57], compiled for Feock image, as shown in Table 4.12. According to the calculation process of the new features, each sample SLIC object is converted into L*a*b colour space, where median colour values of each channel are used as a feature, to add better differentiation to the object's classes, in another colour space.

Table 4.12: New added L*a*b colour space features to improve the RGB features set.

Feature No	L*a*b Features
F11	Median colour of SLIC region in L channel of L*a*b colour space
F12	Median colour of SLIC region in a channel of L*a*b colour space
F13	Median colour of SLIC region in b channel of L*a*b colour space

Figure 4.29 shows the box plot representation of the newly added features. It is evident that there are no outliers in all three plots, which means that all three features serve well the classification. Also, there is not much overlapping of median and distribution boundaries of three class boxes in the box plot, indicating that these features are good candidates for training a classifier with good discriminating properties.

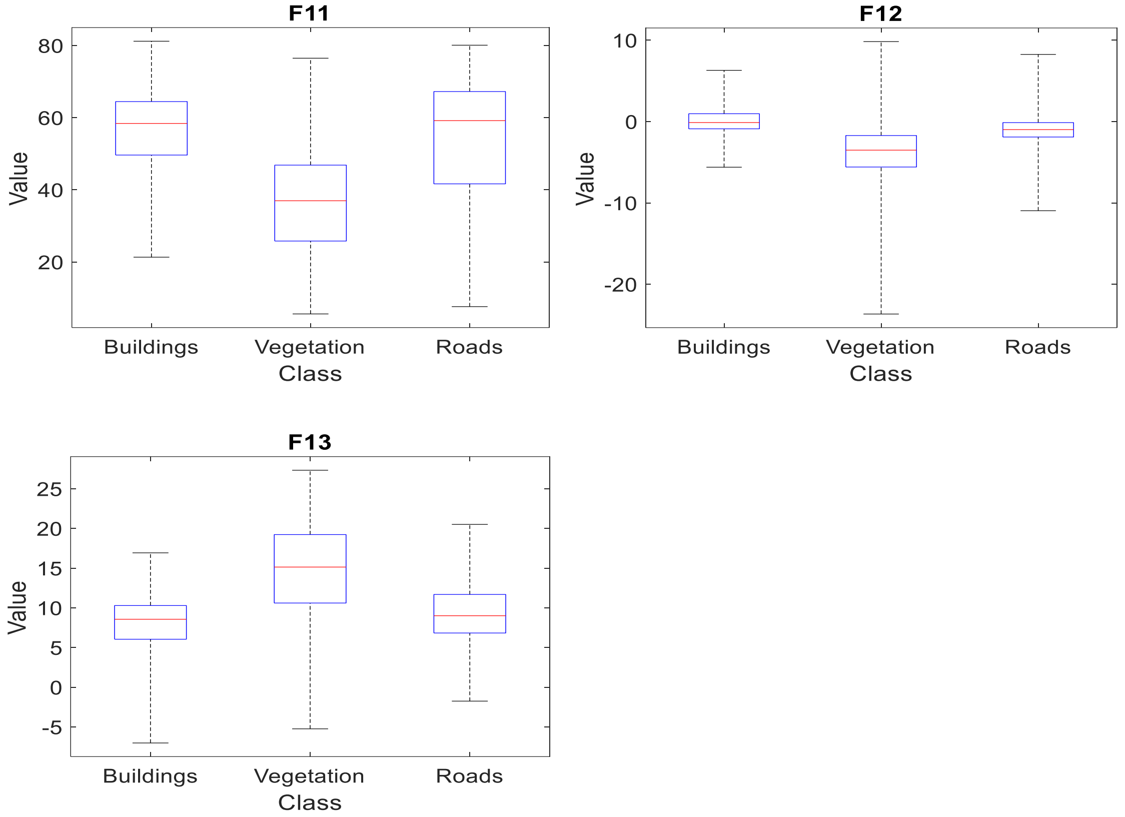


Figure 4.29: Box plot representation of the three added L*a*b features.

Figure 4.30 shows the performance curve in four different cases of classification, where the topmost black curve represents Top 3 accuracy classifiers-based ParetoEnsemble classifier, providing the highest accuracy results for all SLICs, the red curve represents Top 3 classifiers with 10 features, green curve represents all Pareto selected classifiers with 13 features and the blue curve is all Pareto selected classifiers with 10 features. Figure 4.30 gives evidence that, adding three new features increases reasonably the classification accuracy of Feock. According to the plot curves, in the case of more than one SLIC, best classification results are achieved using 13 features with the Top 3 classifiers-based ParetoEnsemble classifier, because combining RGB colour space-based features and new Lab colour space-based features, different class objects are better differentiated, than only RGB colour space-based features.

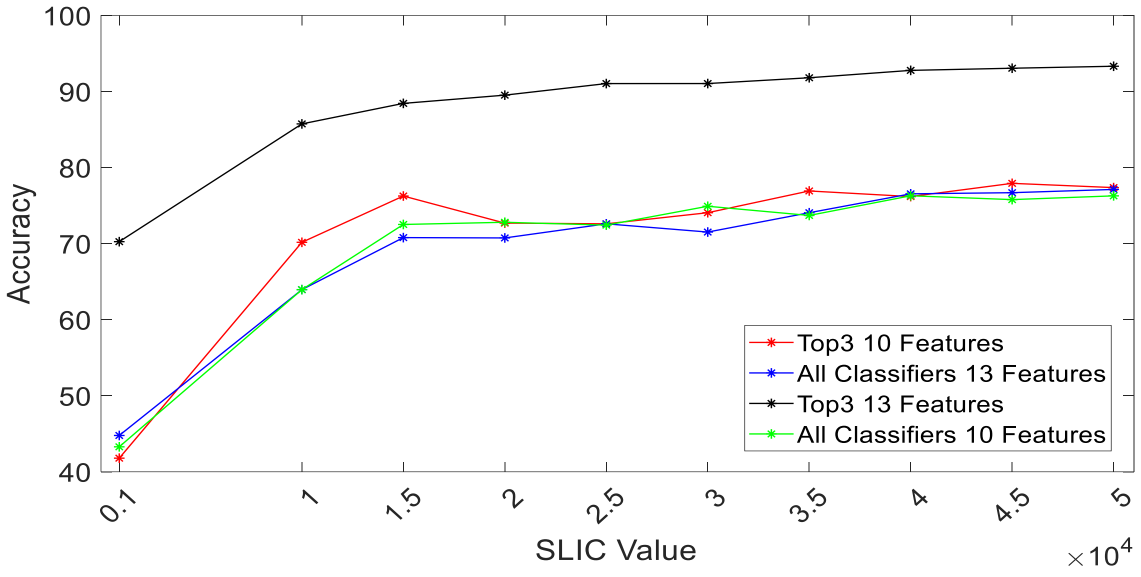


Figure 4.30: Performance comparison for different SLICs, using Top 3 accuracy classifiers-based ParetoEnsemble classifier with 13 features, Top 3 classifiers with 10 features, all Pareto selected classifiers with 13 features, and all Pareto selected classifiers with 10 features.

After improving the features set for classification, the ParetoEnsemble classifier design is improved by incorporating the idea of the weighted sum-based classification. In this modified ParetoEnsemble classification scheme, the prediction score values from top selected classifiers are used, instead of prediction labels. During the prediction phase of each classifier, both predicted labels and predicted scores of each class are collected from each classifier. For example, in this case, when there are three classes, one predicted class label and three predicted score values are collected for each sample prediction. In this modified ParetoEnsemble, instead of using voting of predicted labels from top classifier, predicted scores of classes are used. For this, the sum of predicted scores, for each class, is calculated, by considering the top classifiers, leading to the dominating class, as the one having the highest predicted score sum. In addition, weight values are applied on each classifier, according to the respective accuracy level achieved. Specifically, the classifier, having higher accuracy, thus higher priority, is assigned more weight, keeping the total sum of weights equal to one. This implementation process includes, the prediction score values of the top selected classifiers to be separated in another matrix of MatLab, taking the score values of each sample one by one, and then calculation of the weighted sum of prediction score value for each class, in the form of an ensemble, where the class having the highest predicted score value, is selected as ParetoEnsemble predicted class, for that sample.

Table 4.13 shows sample prediction scores for one object, where the values in the columns represent prediction score for each class, i.e., buildings, roads and vegetation, while the values in each of the three rows represent the prediction score for the top 3 accuracy classifiers. Specifically, the first row includes the predicted score by top classifier 1, for three classes of an object, while the other predicted scores, from top classifier 2 and top classifier 3, are presented in the second and third row, respectively. It is evident that the sum of values of each row is equal to 1, since each cell represents the probability of that sample belonging to one of the three classes. This table includes sample score values for the top selected classifiers, for SLIC 10,000, where each row represents one of the top 3 accuracy classifiers, and each column represents one of the three classes.

Table 4.13: Sample prediction scores for one object, by the top 3 accuracy classifiers.

	Buildings score	Vegetation score	Roads score
Top classifier 1	$5.7063\text{x}10^{-04}$	0.7500	0.2495
Top classifier 2	0	0.9200	0.0800
Top classifier 3	$3.8163\text{x}10^{-16}$	0.9999	$5.6950\text{x}10^{-05}$

After the collection of prediction scores for all sample objects is completed, weights are assigned to each classifier in ParetoEnsemble, based on the priority of each classifier, according to rules described in the next subsection.

4.1.2.4.2 Weight Assignment Rules

The selected weights combination is applied to the sum of scores of each class, for the three selected classifiers, according to the rules below [239], because it so happens that sometimes the number of classifiers, selected by Pareto, is less than 3 (i.e., 2 or 1), so in these cases, three weights cannot be assigned to the scores. On the other hand, in the case, where Pareto selects more than three classifiers, then top 3 accuracy classifiers are picked out of Pareto selected classifiers. The classes 1, 2 and 3 mean buildings, vegetation and roads, respectively.

Rule 1: If there is only one classifier selected from Pareto, then the weight for this single classifier will be 1 (i.e., there is no sum, in case of one classifier), as shown in the Equation (4.4) where *i* can vary from 1 to 3 based on class. The class having the highest score value will be selected as the predicted class. However, this is just a possibility, in the general case of testing some other unknown images. In this specific case of images, used for the assessments, Pareto selection always provides at least two classifiers.

$$Rule\ 1\ Weighted\ sum\ class\ i = (1 * classifier\ score\ class\ i) \tag{4.4}$$

Rule 2: If two classifiers are selected from Pareto, then 0.8, i.e., 80% weight, will be assigned to topmost accuracy classifier, and 0.2, i.e., 20% weight, will be assigned to second top accuracy classifier, so as the sum of weights be equal to 1. The following formulas have been used for weighting the sum of scores for the classifiers. There will be three weighted sum score values, one for each class, while the class having the highest sum value, for a sample, will be selected as the predicted class for that sample, as illustrated in the Equation (4.5) below, where i represents one of the three classes which varies from 1 to 3.

$$\begin{aligned} Rule\ 2\ Weighted\ sum\ class\ i \\ &= (0.8 * top\ 1\ classifier\ score\ of\ class\ i) \\ &+ (0.2 * top\ 2\ classifier\ score\ of\ class\ i) \end{aligned} \tag{4.5}$$

Rule 3: Similarly, if three classifiers are selected from Pareto, then 0.8, i.e., 80% weight, will be assigned to the topmost accuracy classifier, 0.1, i.e., 10% weight, will be assigned to the second top accuracy classifier, and 0.1, i.e., 10% weight, will be assigned to the third top accuracy classifier, so as the sum of weights be equal to 1. The Equation (4.6) below, for weighted scores, provide three weighted score values, one for each class based on the value of i varying from 1 to 3, while the class having the highest weighted score will be selected as the predicted class for the respective sample.

$$\begin{aligned} Rule\ 3\ Weighted\ sum\ class\ i \\ &= (0.8 * top\ 1\ classifier\ score\ of\ class\ i) + (0.1 \\ &* top\ 2\ classifier\ score\ of\ class\ i) + (0.1 \\ &* top\ 3\ classifier\ score\ of\ class\ i) \end{aligned} \tag{4.6}$$

An example of applying weights on predicted scores, for the top 3 diversity-based Pareto selected classifiers, is shown below. For each sample, three score values are collected, by using predicted scores (Table 4.13), in the Equations (4.7)-(4.9) of weighted sum below. The weights are assigned to each classifier, which means that the weight value is multiplied by the score value of each class, one by one.

$$Weighted\ sum\ class\ 1 \\ = (0.8 * 5.7063e^{-04}) + (0.1 * 0) + (0.1 * 3.8163e^{-16}) \\ = 4.5650x10^{-04} \tag{4.7}$$

$$Weighted\ sum\ class\ 2 \\ = (0.8 * 0.7500) + (0.1 * 0.9200) + (0.1 * 0.9999) \\ = 0.7920 \tag{4.8}$$

$$Weighted\ sum\ class\ 3 \\ = (0.8 * 0.2495) + (0.1 * 0.0800) + (0.1 * 5.6950e^{-04}) \\ = 0.2075 \tag{4.9}$$

This process provides three weighted sum values, which represent the probability of the sample, under consideration, to belong to each of the three classes, while the class having the highest probability value is selected as the predicted class for that sample, which is class 2, i.e., vegetation, in this specific example case.

The six satellite images are examined, applying different combinations of weight values, for the top 3 classifiers:

- Combination 1: (0.8, 0.1, 0.1), meaning the classifier, having the highest accuracy/priority, is assigned weight equal to 0.8, while the classifiers, having second and third highest accuracy, are assigned weight of 0.1, producing sum of weights equal to 1; similarly,
- Combination 2: (0.7, 0.15, 0.15);
- Combination 3: (0.7, 0.2, 0.1);
- Combination 4: (0.9, 0.05, 0.05).

Weight values have been selected randomly, to consider different combinations, while the best combination is determined, based on these results. High and low values have been considered as weights and tested their impact on accuracy. The specific aim is to select the best weight values combination, to use with the ParetoEnsemble classifiers, as shown in Table 4.14. Comparing the performance of different weight combinations, for the six satellite images, Combination 1 of weights values proves to be the best option.

Table 4.14: Sample prediction scores for one object, by the top 3 accuracy classifiers.

Image	Combination 1 Accuracy (%)	Combination 2 Accuracy (%)	Combination 3 Accuracy (%)	Combination 4 Accuracy (%)
Image 1	**99.6**	99.6	99.6	99.5
Image 2	**87.0**	86.8	86.9	86.4
Image 3	**94.8**	94.3	93.9	94.2
Image 4	**96.4**	96.2	96.0	95.9
Image 5	**95.9**	95.3	95.3	95.1
Image 6	**93.5**	92.5	93.4	92.4
Mean	**94.5**	94.1	94.2	93.9

After applying score and weighted sum-based ParetoEnsemble classification, instead of simple voting of predicted labels from selected classifiers, the class having the highest score is assigned as predicted class for that specific object, which in the case above is class 2, i.e., vegetation. The same process, of weighting score decision, is applied for all objects of the image, one by one, to derive predicted values for all. These predicted labels, for each object, are converted into a predicted image, by assigning the same class label to all the pixels, inside that object.

After designing and applying ParetoEnsemble classifier on images, the performance accuracy of the ParetoEnsemble method is compared to the accuracy of the best performing, of the eight individual classifiers, already trained, to estimate the efficiency of the ParetoEnsemble classifier. Table 4.15 shows that, after applying weighted sum on score-based ParetoEnsemble classifier, the results are improved, compared to the results of the best performing of the eight individual classifiers, for Feock image. Furthermore, the table data show that the performance accuracy increase, as the SLIC values increase.

Table 4.15: ParetoEnsemble accuracy versus the best performing of the eight individual classifiers accuracy for Feock image.

SLIC size	Best accuracy of individual classifiers (%)	ParetoEnsemble accuracy (%)
1000	63.3	**64.3**
5000	72.3	**72.9**
10,000	75.1	**75.7**
15,000	78.4	**78.7**
20,000	79.3	**80**
25,000	79.8	**80.9**
30,000	80.2	**80.7**
35,000	80.5	**81.7**
40,000	80.5	**81.9**
45,000	80.7	**82.1**
50,000	81.0	**82.4**
75,000	81.2	**82.5**
100,000	81.3	**82.6**
150,000	81.2	**82.6**

The performance of different SLICs is also compared, in regard to overall accuracy, non-vegetation class accuracy (i.e., roads and buildings) and vegetation class accuracy, for seven different SLICs, with gaps in between, to avoid a crowded plot, as shown in Figure 4.31, where the yellow curve shows vegetation class accuracy, the blue curve shows overall accuracy, and the red curve shows non-vegetation class accuracy. The performance curves illustrate that accuracy of classification improves at higher SLIC sizes, for all three curves. To find out the higher limit of SLIC value, for improvement in classification accuracy, Feock image is also tested for very high SLIC values. A significant increase in the performance is evident, compared to individual classifiers, up to a specific limit of SLIC value rise, while, after that top limit, the performance does not vary much, as SLIC value increases. An important matter to consider, for Feock performance estimation, is also computational and time cost, because Feock is a big image, compared to the six testing images. Since the increase in the SLIC value obviously increases computational memory usage and time, significantly, it is better to use a moderate value for SLIC, neither too high nor too low, to make it efficient in terms of both performance and computation costs. As observed in Figure 4.31, a SLIC value of 50,000 seems like a reasonable choice, for Feock image, considering all the factors. The accuracy plot in Figure 4.31 shows that, the vegetation accuracy is the highest for all SLICs, overall accuracy is middle valued, and non-vegetation accuracies are the lowest, in the case where Feock image is validated through cross-validation, by using all sample objects, collected after segmentation. This difference between vegetation and non-vegetation accuracies indicates that the classification system is biased/over fitted for vegetation class samples, because there is much more vegetation area in Feock image, compared to buildings and roads area, as it can be observed in Feock ground truth image (Figure 3.6, Chapter 3). More sample objects for vegetation area can cause the biases/overfitting for vegetation class in classifiers, which in turn gives more accurate predictions for vegetation class, compared to non-vegetation class samples.

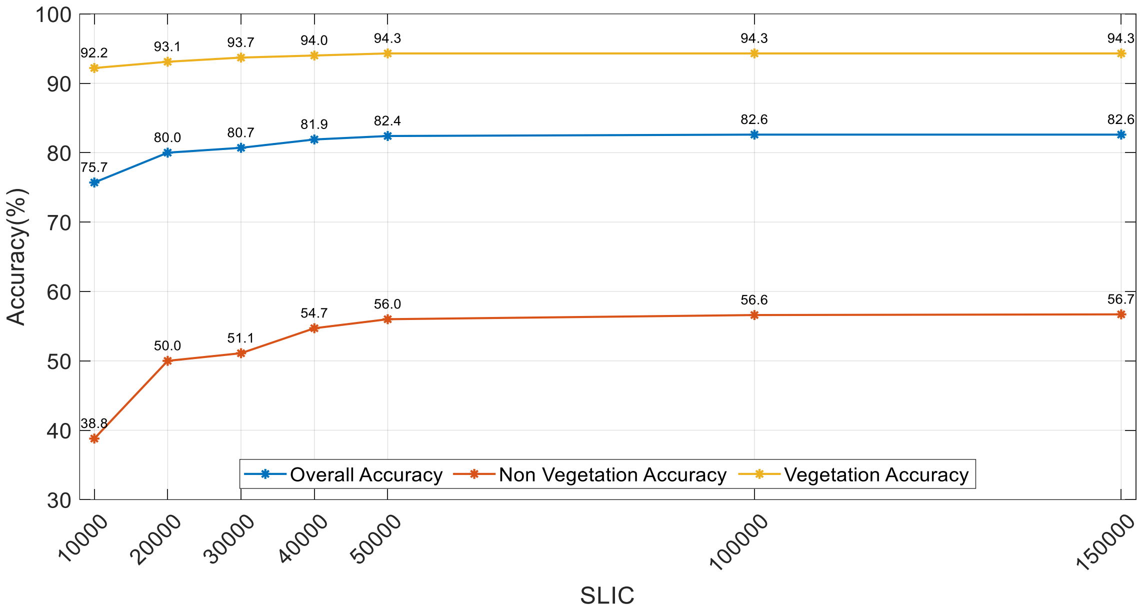


Figure 4.31: Comparison of different SLICS for overall accuracy, non-vegetation accuracy and vegetation accuracy of Feock image.

4.1.2.4.3 Comparison between Unbalanced and Balanced Data-based Classification Results

A possible solution to the issue of overfitting/biasness is to add balancing in training data, which can avoid biasness in vegetation class predictions. Balancing of data samples was done after feature extraction step of classification, where the object count, for each of the three classes, is calculated, selecting the minimum count, as the count to be considered for all the three classes. For example, using SLIC 1000 leads to a count of 560 objects for vegetation class, 250 for buildings class, and 190 objects for roads class. Since 190 is the minimum object count, that many are considered, from each of the three classes, and used during the training of classifiers. The selection of samples from the classes having more samples, was done randomly, i.e., 190 samples were selected randomly, out of 250 objects of buildings class, to be used in balanced data, for training. Since SLIC 50,000 has already been selected as an optimum SLIC count in Figure 4.31, in terms of performance and computational costs, balancing of data samples is applied for this SLIC value and the accuracy results of unbalanced and balanced data were compared in bar graph of Figure 4.32. Based on the bar graph, it is evident that the overfitting is reduced for vegetation class, while there is more balance between vegetation and non-vegetation class accuracies, i.e., 78.7% and 87% which was 94.3% and 56%, in case of unbalanced training, respectively. However, the overall accuracy, in the balanced case, is still less than in the unbalanced case, because balancing is added during training of classifier as well as during validation predictions, which leads to too many false positive predictions for non-vegetation class pixels, since many of vegetation pixels are classified as either roads or buildings, due to balanced classifier training, providing a reduced overall accuracy, compared to unbalanced data validations.

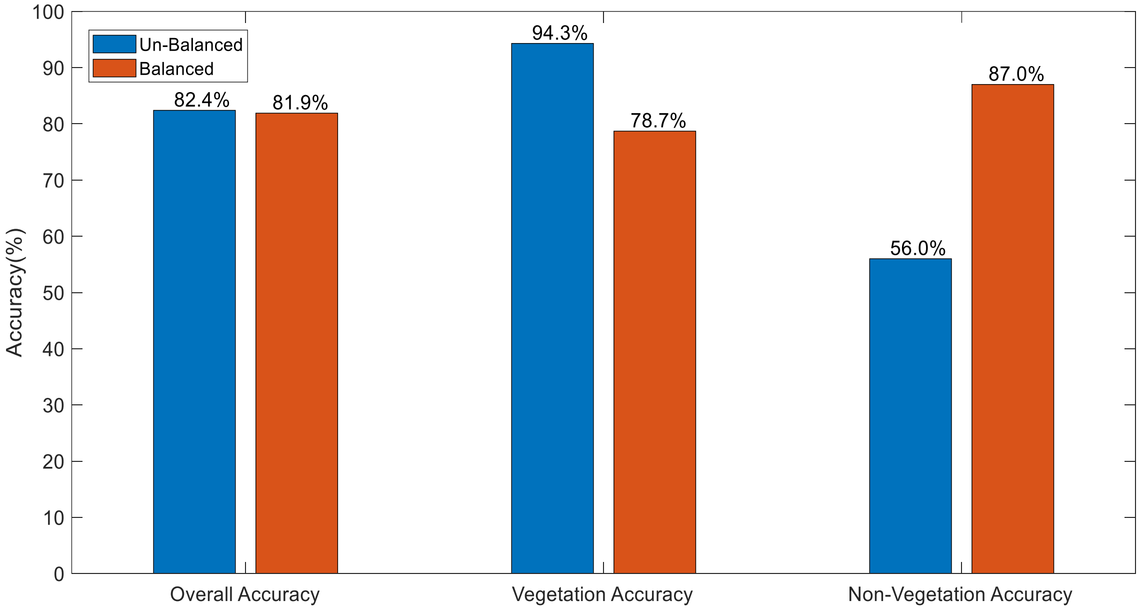


Figure 4.32: Unbalanced and balanced data performance comparison for SLIC 50,000.

Another factor to be considered is the predicted images for both balanced and unbalanced cases, as shown in Figure 4.33, where the left side image represents predicted validation image for the unbalanced case, while the right-side image shows predicted validation image for the balanced case, using SLIC 50,000 of Feock. Both predicted images show that balanced case provides many more false-positive predictions, for buildings and roads classes compared to the unbalanced case. Since more false predictions can lead to more false estimations, for flooding during modelling of runoff (false positive predictions for non-vegetation area is a very sensitive parameter of flooding predictions), it is better to keep unbalanced feature data, during training, which can produce more accurate results for predictions, because most of the images, used in this research, contain more vegetation areas, compared to non-vegetation areas. Thus, it is better to train the models slightly over fitted towards vegetation, to get more accurate runoff modelling and estimations. Both cases are going to be tested with unknown images and runoff estimations, to decide upon the best case.

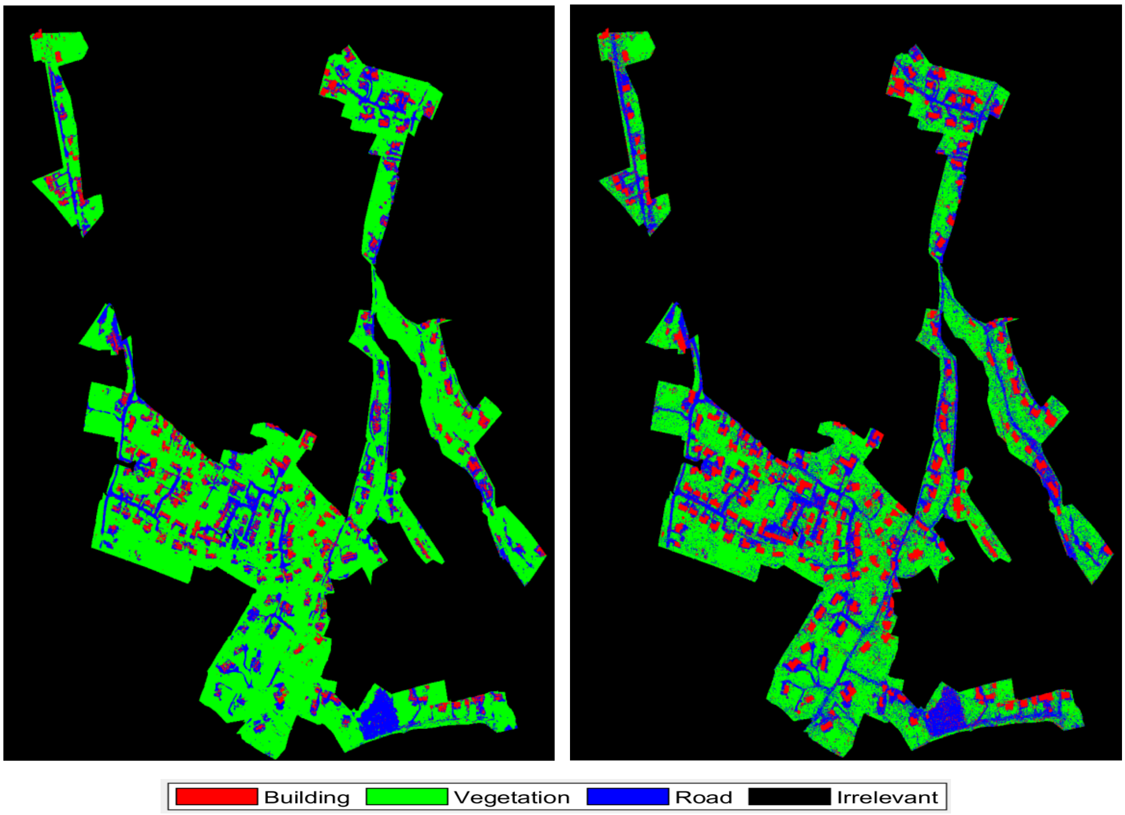


Figure 4.33 Unbalanced data predicted image (left), and balanced data predicted image (right) for Feock using SLIC 50,000.

4.1.2.5 Conclusion and Future Works of the 2nd Scenario

In this scenario, superpixels (SLIC) based segmentation was applied to satellite images, allowing different size objects and multiple features to be extracted from these objects. Next, the objects are categorised using eight different classifiers instead of processing pixel by pixel to save computational resources, including memory and time. The most frequent pixel inside an object is considered the class label for that object in this case.

The results of different SLICs applied to six selected satellite images are compared. Also, the results of the pixels-based approach are compared with the object-based approach by applying Pareto and Knee point analysis. It is noticed that the object-based approach gives better results compared to the pixels-based approach, which also takes less time and saves on memory and processing expenses. Furthermore, it is observed that SLIC 10,000 is the most appropriate SLIC size for the six images in terms of both performance and computational costs.

Next, a modified ParetoEnsemble classifier is designed by selecting a few top performance classifiers (from among the eight individual classifiers used previously) to get more reliable and accurate predictions compared to the predictions of an individual classifier. The diversity of eight classifiers is estimated to pick the most diverse classifiers, so that if some samples are wrongly predicted by one classifier, then these can be corrected by the other classifiers in the ParetoEnsemble. In the modified ParetoEnsemble, predicted scores are used from individual classifiers, rather than taking majority voting from selected classifiers. Also, weights are applied to each selected classifier based on the priority of classifiers, and then the weighted sum of predicted scores is calculated for each class. This allows for the class that has the maximum weighted sum of the scores to be picked as the predicted class for the object. Modified ParetoEnsemble classifier results were compared with the individual classifiers used in this section, and the ParetoEnsemble classifier gives more accurate predictions compared to the individual classifiers.

The results from this classification scenario conclude that, adding concepts of diversity, weighted sum and prediction scores, all significantly enhance the classification performance. A typical example is the Top 3 ParetoEnsemble classifier, using SLIC 50,000 and (0.8, 0.1, 0.1) weights combination for the contributing classifiers, which is a good choice for the case of big images, such as Feock, using unbalanced data for classification.

The classification results from the two scenarios explained in this chapter will be compared with another third scenario of classification (elaborated in next chapter) to select the most efficient classification model to be utilised in unknown data predictions. The assessment of generalisation power of the best classification model will also be done in the next chapter.

CHAPTER FIVE

5 CONVOLUTIONAL NEURAL NETWORKS BASED SEGMENTATION & CLASSIFICATION RESULTS ANALYSIS

The previous chapter elaborates the methodological details on the two classification scenarios applied by using supervised mode of learning. This Chapter provides the details explanation on the deep learning-based classification of urban land cover images. The detailed explanation of Convolutional Neural Networks (CNN) is given in this chapter, which is the 3rd scenario of classification used in this research. This section first gives details on the data selected for the experiments performed with CNN, and then CNN is applied with different parameter values and settings to compare the results of different parameters to select the best setting configuration for classification.

The following section of this chapter compares the results of the three classification scenarios (used in Chapter 4 and Chapter 5) to select the most suitable kind of classifier and its parameter values. The best scenario classifier is further used for the assessment of the generalisability of the classification system by performing the testing of classifier on unknown images in this chapter.

The best-performing classification methodology, as selected in this chapter, is also used in the next chapter with the InfoWorks ICM software to improve the modelling of a surface water network.

5.1 The 3rd Scenario (CNN-Based Classification)

This section introduces a methodology, implemented in MatLab R2018b, for semantic segmentation, initially applied on the six high-resolution satellite images shown in Figure 4.16, Chapter 4. One of the goals of this work is to acquire a high-quality convolutional neural network using a small data set. A convolutional neural network with an encoder-decoder architecture based on SegNet is employed in this study which shows that even with a small number of training images, promising results can be achieved to classify the three classes: buildings, vegetation and roads in satellite images. The data is processed with different augmentation techniques and the best network architecture is searched by running several experiments where the important parameters are tuned. The choice of a convolutional neural network is motivated by the multitude of studies that prove the general superiority of this approach over traditional methods, as explained in the next section.

The purpose of this work is to perform a convolutional neural network based on SegNet (dividing the image in different regions according to the meaning of their content) on satellite images representing urban scenes with different proportions of buildings, vegetation and roads. This task is particularly difficult since elements belonging to the same class may exhibit a large variation in terms of shape, colour and texture. Moreover, it is difficult to collect a large dataset for the training stage.

In the deep learning field, it is commonly known that a large amount of data is required to properly train a network. This concept gets stronger every year, as the trend in the Artificial Intelligence (AI) community is to research always deeper and more complex networks. Unfortunately, accessing a suitable amount of data is not possible for everyone along with data ground truth information, thus making difficult to train a large network for a custom application. The issue of limited dataset is dealt by incorporating data augmentation concept to create a reasonable size dataset where all images created are considered as a different entity and the results are promising with an updated dataset. This research work has many applications in areas in which the amount of captured data is limited or expensive to obtain, such as flood estimation, urban expansion modelling, and agricultural policy modelling.

5.1.1 Methodology

5.1.1.1 Data Preparation

It is possible to see that the six RGB images in Figure 4.16, Chapter 4 can be grouped into three groups in terms of similarity (first and second; third and fourth; fifth and sixth). For that reason, these images are considered separately in the results analysis stage. As mentioned before, the ground truth is manually built by labelling the pixels according to three different classes: Buildings, Vegetation, and Roads. These images are then split into images of size 128x128.

Since this dataset is quite limited for a semantic segmentation task, several augmentation techniques were employed to make it larger. In particular, affine transformations, brightness transformation and the addition of noise were used. These techniques are typically used in deep learning [240] with Affine transformations including horizontal and vertical flipping and rotation with a random angle [241]. Brightness transformation was also randomly applied to each image, while the noise used was is Gaussian [242]. Technically, the augmentation process transforms the training images in such a way that for the neural network they are considered different, increasing the diversity of the data and preventing the network from memorising the exact details of the existing images [243]. For each individual function and for each of the 3 groups of functions, the probability of their occurrence is given, as well as the range of values that the function can accept. Further, a random number of functions that will take part in the transformations is randomly selected, then one is selected from the available range of values in the same way, after which the selected functions process the image in turn, with the results that all received effects overlap each other. This process increases the diversity of the data. Each large image resulted in a number of sub images ranging from 212 to 1164, after augmentation. Examples of such image augmentations are shown in Figures 5.1 and 5.2. These augmented images are created from the original six images even though they do not resemble the original images to a human eye. Then the images were divided as follows: We ran the training using a 6-fold cross-validation strategy. At each training iteration, the sub images and augmented images coming from 5 original images were used for training and validation, while the subimages coming from the remaining one were used for testing. In this way, all the images contributed to the training and testing without overlapping and, at the same time, we perform validation to assess the accuracy of the network and monitor the presence of overfitting.

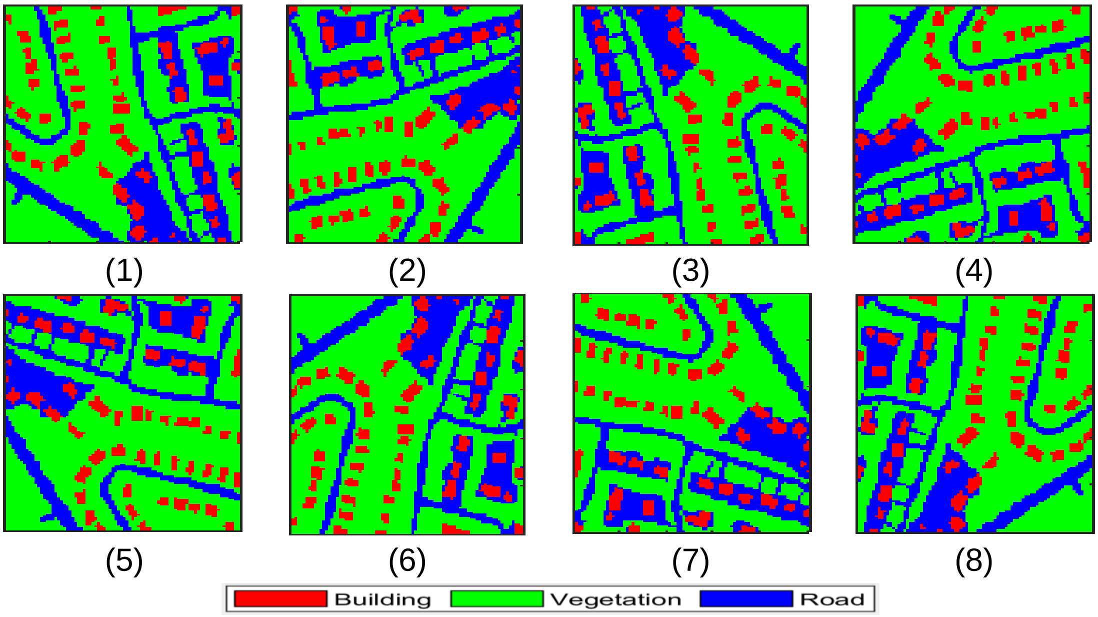


Figure 5.1: Example of image augmentation, (1) A ground truth subimage, (2-4) 90, 180 and 270 degrees rotation, respectively, (5-7) 90, 180 and 270 degrees rotation along with horizontal reflection, respectively, (8) 180 degrees rotation and vertical reflection.

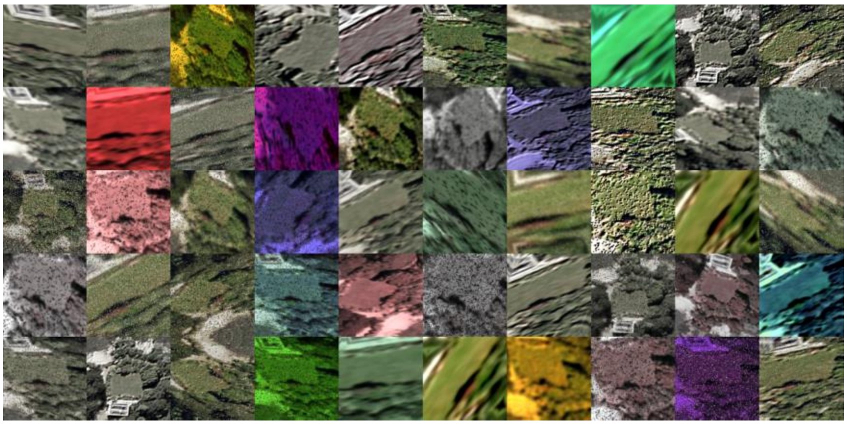


Figure 5.2: Set of augmented images for image 1 in the six images.

Our dataset is summarised in Table 5.1.

Table 5.1: Dataset details.

Original image	# subimages after augmentation
1	212
2	237
3	372
4	372
5	1002
6	1164
Total	3359

An analysis stage was conducted where the occurrence of each class in the six images was checked. If the classes are not balanced in the dataset, some remedial action needs to be taken. Figure 5.3 shows the results of this analysis. It can be seen from this chart that the dataset is not balanced: vegetation has a much higher frequency of occurrence in the first 4 images while for images 5 and 6 the number of pixels related to vegetation was much lower than the other classes. In every image, roads have a lower frequency with respect to the other classes. As we explain in the next section, we take this issue of dominant classes into account by means of class weights.

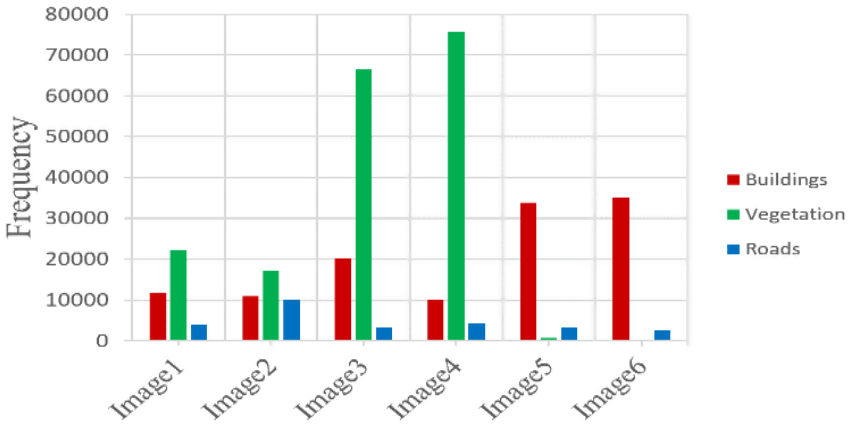


Figure 5.3: Frequency occurrence of each class in the 6 images.

5.1.1.2 Data Pre-processing

The pre-processing step is crucial in neural network training and must be carefully planned in order to make the learning faster and more stable. In particular, it is known that normalising the input data to a fixed range produces better classification results [244].

Each input image is split to a fixed size (128x128x3) in order to have a good trade-off between too large images (long training time) and too small images (bad classification performance). Histogram Equalisation is performed on each RGB channel in order to increase the contrast and improve the network performance [245]. Then the images are normalised to the [0, 1] range. Normalisation is typically performed in neural networks because the non-linear functions that are employed work better in this range. Moreover, if the inputs have different ranges, with normalisation we bring them to the same range so that they are comparable.

5.1.1.3 Network Architecture and Training

For the network implementation, the architecture of SegNet was used as the starting point. The choice was motivated by the fact that this network achieves good results on different datasets and offers a structure that can be modified according to specific needs. SegNet is based on the encoder-decoder architecture. The encoder part takes an image as input and encodes it in a lower dimensional vector which contains the features that best characterise the image. It consists of several convolutional layers, each followed by batch normalisation, Rectified Linear Units (ReLU) non-linearity and a maxpooling. The dimensionality of the data is reduced after each pooling layer.

The decoder takes a vector of features as input and produces an image of the same size as the input. It reflects the same structure as the encoder, with an equal number of de-convolutional blocks followed by batch normalisation, leaky ReLu and upsampling. At the end, the SoftMax layer provides a probability value for each class prediction [246]. Each pixel is assigned to the class with the highest probability, since the purpose is to provide a classification which is as equal as possible to the ground truth.

The number of convolutional layers, the number of filters per layer and the filter size are important parameters and determine the abstraction and modelling ability of a neural network. This number depends on the particular task that is being faced and has to be chosen carefully in order to avoid underfitting and overfitting. Moreover, the computational complexity and the memory requirement of the trained model depends on these parameters. SegNet was conceived to be trained and tested on a large amount of data with many classes. For this reason, as typically found in state-of-the-art deep learning, it employs a very high number of layers and has a particularly large dataset for the training phase. Since we do not have a large number of images at our disposal, as mentioned above, we modified the structure of the network. The number of convolutional layers and the number of parameters per layer was reduced, in order to prevent overfitting. The choice of these numbers was determined after a phase where different configurations were tested. The performance of the network was tested at each phase and the best configuration was chosen. We describe the different configurations in section 5.1.1.21. The architecture of our network is depicted in Figure 5.4.

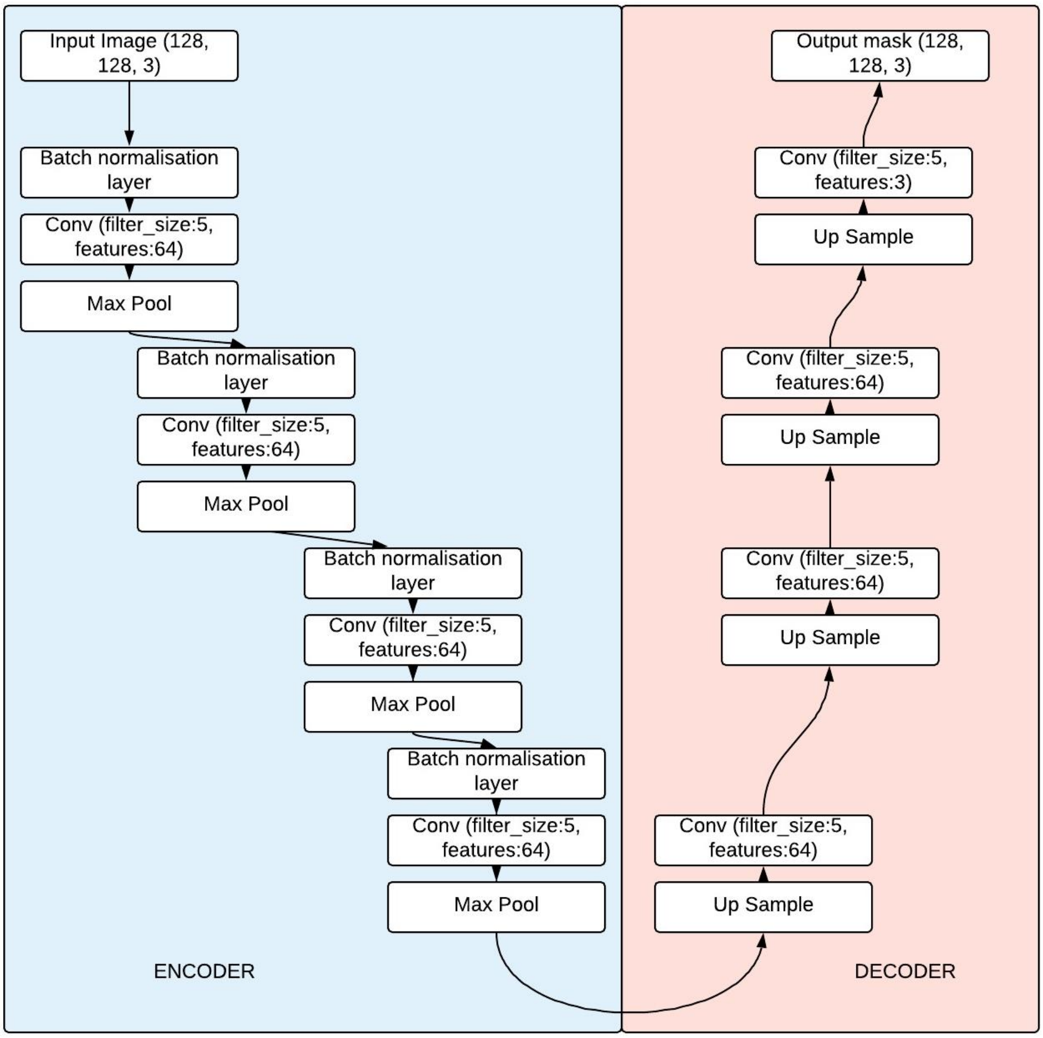

Input Image (128, 128, 3)
Batch normalisation layer
Conv (filter_size:5, features:64)
Max Pool
Batch normalisation layer
Conv (filter_size:5, features:64)
Max Pool
Batch normalisation layer
Conv (filter_size:5, features:64)
Max Pool
Batch normalisation layer
Conv (filter_size:5, features:64)
Max Pool
ENCODER
Up Sample
Conv (filter_size:5, features:64)
Up Sample
Conv (filter_size:5, features:64)
Up Sample
Conv (filter_size:5, features:64)
Up Sample
Conv (filter_size:5, features:3)
Output mask (128, 128, 3)
DECODER

Figure 5.4: The network's architecture.

As the class distribution in our dataset was not balanced (the number of pixels related to Vegetation was higher than the other classes), the SoftMax computation assigns a weight to each class [247], based on the inverse of the frequency of occurrence (i.e., the rarer the class, the higher the weight). Weights are related to the probability of observing a given class. If all classes occur with equal frequency, there is no issue. But if a class is extremely rare, when the network is uncertain whether to predict that class or a more likely one, it will always predict the latter in order to have more probability to guess correctly. However, when using weights, Equation (5.1), this problem is greatly reduced because it prevents the network from classifying every pixel to the most frequent class, which reduces classification error.

The weights are given by

$$w_i = median(p)/p_i \tag{5.1}$$

Where w_i is the weight associated to class i, and p_i is the relative frequency of class i [142].

5.1.1.4 Training

As mentioned above, we conducted the training phase by using 6-fold cross-validation. At each iteration, one of the 6 images was kept out and used for testing once the network was trained. The coefficients of the filters were initialised with a normal distribution. The network was trained using the SGD algorithm with learning rate equal to 0.5, with a drop factor of 0.5 and a drop period of 200 epochs (images are not fed one by one into the network in the training set, but are grouped in batches). The training algorithm uses cross-entropy as the loss function (see [248] for more details), which is commonly used in neural network-based image processing. Filters of various sizes were used (according to the network configuration) and stride 1 for the convolutions. The training accuracy was computed as the percentage of correctly classified pixels in the validation set. When the accuracy reached a stationary level, the training stopped.

5.1.2 Results

In this section, we describe the results that were achieved for the test images, which are the 6 original images. Each test image was considered separately, showing the result of the prediction in terms of a segmented image, which offers an easier interpretation of the results through showing how accurately the image is predicted visually by comparing how well it matches with actual ground truth; and Confusion Matrices, which indicate the correct classification rate for each class by providing the vectors with predicted pixels and true pixels.

5.1.2.1 Network Configurations and Training

As introduced in the previous section, we conducted a comparison of the performance of different network configurations, starting from a simplified version of SegNet. The purpose was to find a good configuration for our limited dataset. Training and validation were performed in an iterative fashion. We considered three parameters: number of layers, number of filters per layer and kernel size. At each iteration, different combinations of these parameters were chosen, and the training was performed. At the end we compared the performance of the different networks that we trained. The comparison is illustrated in Table 5.2. The average train and validation accuracy achieved over the whole dataset was used as the performance metric. The configurations that achieved the worse results have been discarded.

Table 5.2: Impact of different network configurations on results where K is the size of the convolutional filters, and Li is the ith layer. The bold values indicate the best model and the corresponding train and validation accuracy.

No. of Layers	K	No. of Filters					Accuracy (%)	
		L1	L2	L3	L4	L5	Train	Validation
1	5	32					63.39	61.69
1	5	64					63.86	62.35
1	5	96					63.77	62.13
1	5	128					64.06	62.56
2	5	64	32				75.72	74.05
2	5	64	64				77.71	76.78
2	5	96	96				79.61	77.70
2	5	128	128				76.81	75.34
3	5	64	32	16			84.80	83.73
3	5	64	64	64			87.44	86.23
3	3	64	64	64			78.44	77.74
3	7	64	64	64			90.05	88.88
3	5	96	96	96			88.70	87.33
3	5	64	96	128			85.62	84.43
4	**5**	**64**	**64**	**64**	**64**		**90.73**	**89.24**
5	5	64	64	64	64	64	90.56	88.85

From Table 5.2 it can be seen the number of layers most strongly affects the accuracy (as clearly shown in Figure 5.5). In terms of train and validation accuracy, the best model is a model with 4 deep layers and 64 feature maps. However, it is worth noting that the model with the maximum number of layers has a slightly lower accuracy, likely due to the attenuation of the gradient. It was also noted that the number of feature maps does not significantly affect the accuracy. Comparing models 10, 11 and 12, where the number of layers and feature maps is the same, but the filter size is different, we can say that the model with filter size 7 has the highest accuracy.

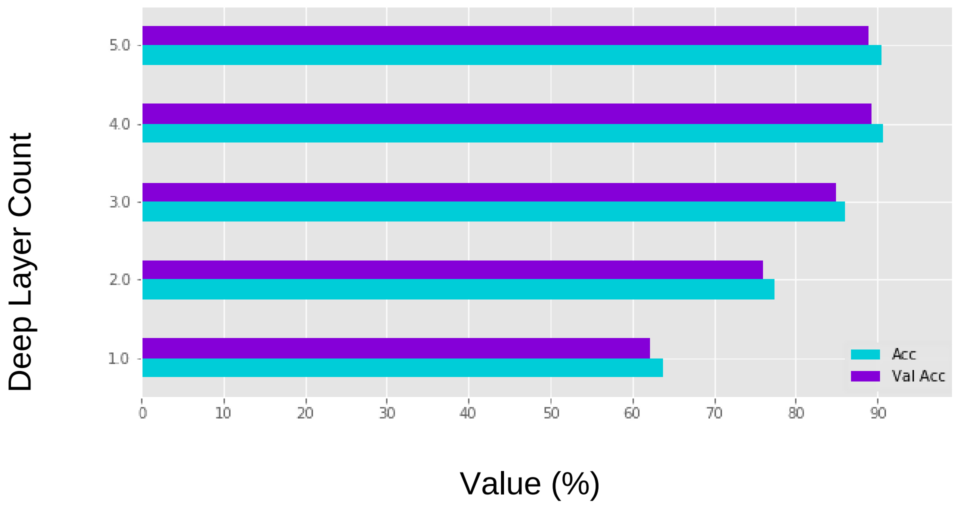


Figure 5.5: Effect of the number of layers on accuracy. Acc – is training Accuracy (%), Val Acc – is Validation Accuracy (%).

If too simple a network is used, the trained model is not able to correctly fit our data. For example, using just one layer, the validation accuracy does not exceed 62%. This is because only simple feature (like edges) have been captured. The highest validation accuracy (89%) is achieved using 4 layers. This appears to be one of the most important parameters as relevant changes were not seen when the filter size or the number of filters per layer was varied. Therefore, our chosen network configuration includes 4 layers with 64 filters per layer and a filter size equal to 5.

The training and validation plots are depicted in Figure 5.6, relatively to the training stage with the dataset including images 1 to 5. For reasons of repetition, the plots related to the other cases are not displayed. However, the plots were similar. In particular, two plots are displayed. The first is related to the accuracy in the training set, i.e., the percentage of correctly classified pixels (Figure 5.6 green curve). An increasing accuracy means that the network is improving its prediction capability. Conversely, the second plot refers to the training loss/error measure (Figure 5.6 red curve). The lower the loss, the higher the performance. The slope of these curves depends on the learning rate and on the state of the network. A higher learning rate means that the weights change faster, and so do the accuracy and the loss. At the beginning, we did not know whether the optimisation algorithm reached a global minimum or a local minimum [249]. In the latter case, we needed a high change in the loss to proceed from the local minimum towards a better minimum. If the curve flattened, we could say that a local or global minimum had been reached, and when such a minimum was reached, each change in the weights did not affect the accuracy and loss significantly. The training was stopped when the accuracy reached a stationary value (about 200 epochs), meaning that further iterations would have produced no significant change in the network's weights. As expected, and mentioned above, the validation accuracy is slightly lower than the training accuracy. The training has been stopped when the performance stopped improving, and the accuracy reached an almost constant value.

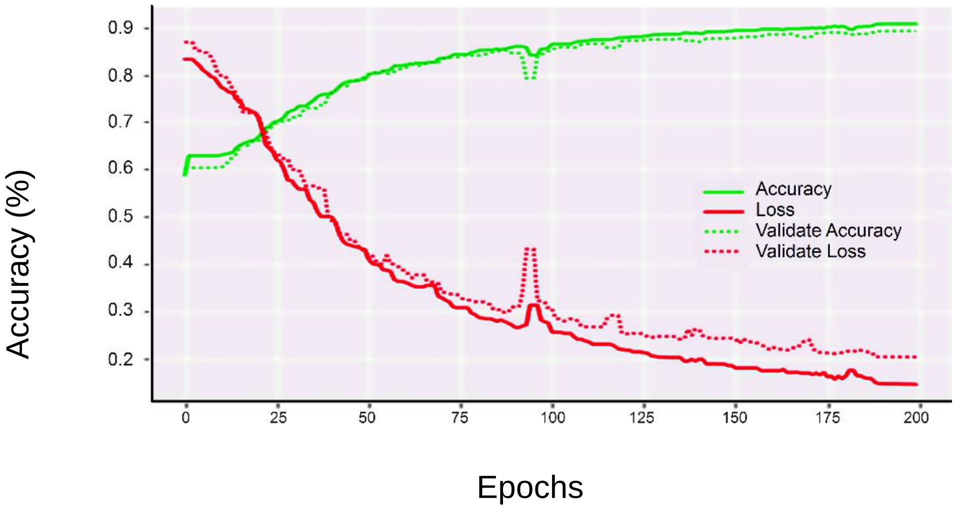


Figure 5.6: Training accuracy and loss with the dataset, for the chosen network architecture.

5.1.2.2 Test Results

In this section, the prediction results are depicted that were achieved in the 6 different test stages. As already mentioned, each time a different image was used as a test case. By looking at the results, it is possible to draw some interesting observations. The vegetation has been well modelled by the network. The network is able to segment the roads, which, however, are not always segmented with straight edges. See, for example, test image 2. Moreover, it is possible to notice some confusion between roads and trees (image 1). The buildings are very well modelled, at different zoom levels.

In Table 5.3 standard metrics, Precision, Recall, F-score, Kappa, and Overall Accuracy (OA) are presented over the different test stages. From this it can be seen that the network is particularly good at predicting buildings with an OA of 93.67%, and vegetation with an OA of 95.83%. As for the roads, the OA is lower (67.71%). This can be explained by the scarcity of pixels related to roads in the datasets, as clearly shown in Figure 5.3. We believe that the same architecture could perform much better even on roads, with a larger dataset. The low precision on roads, together with the accuracy values, indicate that many pixels that the network tend to classify part of the roads as buildings or vegetation.

Precision and Recall are combined in the F-score as shown in Equation (5.2), a measure of test's accuracy which is given by

$$F - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5.2}$$

Based on the F-score, the class that is best modelled by the system is buildings. The relation between the predictions of the various classes is shown in detail in the Confusion Matrices and segmented images presented in Figures 5.7-5.9. B, V and R mean Buildings, Vegetation and Roads, respectively.

Table 5.3: Evaluation metrics.

	Buildings (%)	Vegetation (%)	Roads (%)
Producer Accuracy (Precision)	93.67	95.83	67.71
User Accuracy (Recall)	96.35	92.93	73.56
F-Score	94.99	94.36	70.51
Kappa	86.7%		
Mean OA	92.6%		

When dealing with an imbalanced dataset, it is essential to pay attention not only to the overall evaluation metrics but, also, the corresponding misclassification costs. Thus, Kappa statistics are a good performance measure when facing an imbalanced dataset. [250] proposed a qualitative interpretation of Kappa statistics (Table 5.4) which was assigned to the corresponding agreement measures.

Table 5.4: Strength of agreement for categorical data of Kappa interpretation.

Kappa statistic	Interpretation
< 0.00	Poor agreement
0.00 — 0.20	Slight agreement
0.21 — 0.40	Fair agreement
0.41 — 0.60	Moderate agreement
0.61 — 0.80	Substantial agreement
0.81 — 1.00	Almost perfect agreement

5.1.2.2.1 Image 1 and 2 Results Analysis

Here, the prediction results for images 1 and 2 are shown considering the network with the chosen configuration. The Confusion Matrices and segmented images are shown in Table 5.5 and Figure 5.7, respectively. This concerns the network with 4 layers, so it can be seen that the performance of image 1 (Table 5.5 left and Figure 5.7 top row) is high for classes one and two, while it is very low for class three, suggesting that more abstraction and complexity is needed to model this class. Table 5.5 right and Figure 5.7 bottom row show the Confusion Matrix and the prediction for image 2, respectively. In this case, the prediction accuracy for the third class is much higher. As can be seen in the predicted image, the content related to the third class is much clearer than the previous image, where the roads were covered by trees. This implies the necessity to introduce some more prediction ability to model hidden areas, which can be given by a larger dataset and a more complex network.

Table 5.5: Confusion Matrices for test image 1 (left) and image 2 (right).

Image 1		Actual values			
		B	V	R	Precision
Predicted values	B	11251	1032	295	89.4%
		29.6%	2.7%	0.8%	10.6%
	V	34	19123	1435	92.9%
		0.1%	50.3%	3.8%	7.1%
	R	149	3351	1355	27.9%
		0.4%	8.8%	3.6%	72.1%
	Recall	98.4%	81.4%	43.9%	**83.4%**
		1.6%	18.6%	56.1%	16.6%

Image 2		Actual values			
		B	V	R	Precision
Predicted values	B	10048	412	440	92.2%
		26.4%	1.1%	1.2%	7.8%
	V	622	15072	1483	87.7%
		1.6%	39.6%	3.9%	12.3%
	R	716	1154	8078	81.2%
		1.9%	3.0%	21.2%	18.8%
	Recall	88.2%	90.6%	80.8%	**87.3%**
		11.8%	9.4%	19.2%	12.7%

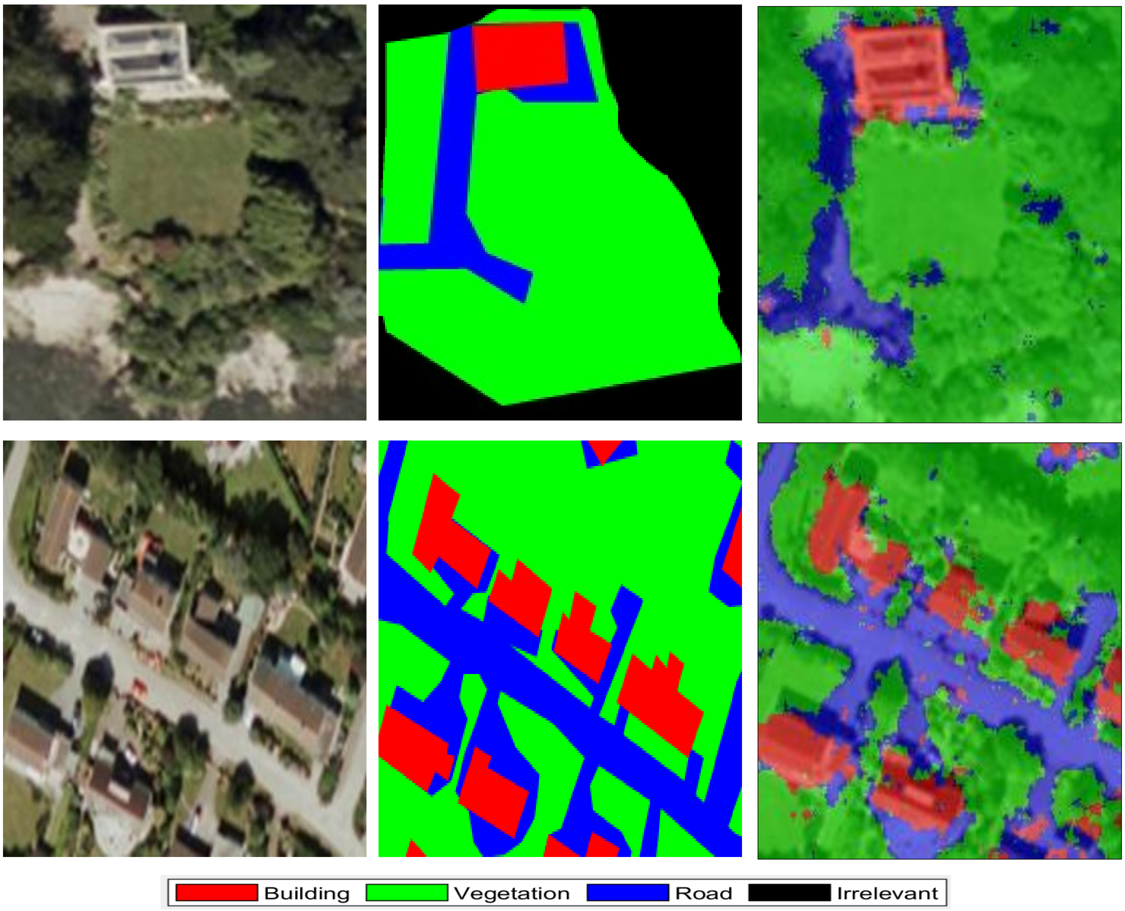


Figure 5.7: Original images (left column), ground truth images (middle column), and predicted images (right column) for image 1 (top) and image 2 (bottom).

5.1.2.2.2 Image 3 and 4 Results Analysis

The third image achieved good performance, although it is very different from images 1 and 2 in terms of content and class distribution. The overall accuracy is 92.6%, as shown in Table 5.6 left. Even better is the accuracy of image 4, which is shown in Table 5.6 right. The overall test value is 95.5%. The roads are more difficult to distinguish with respect to image 3, producing a lower class-specific accuracy. This big influence on the accuracy of roads is owing to the fact that many regions around car parks, which all have the same colour features as roads, are not marked as roads on the ground truth of both images. The images 3 and 4 have large car parks with cars in them, and look like building roofs but are attached to the vegetation class, which raises a segmentation error, Figure 5.8.

Table 5.6: Confusion Matrices for test image 3 (left) and image 4 (right).

Image 3		Actual values			
		B	V	R	Precision
Predicted values	B	17296	2933	4	85.5%
		19.2%	3.3%	0.0%	14.5%
	V	1537	64351	587	96.8%
		1.7%	71.5%	0.7%	3.2%
	R	60	1574	1658	50.4%
		0.1%	1.7%	1.8%	49.6%
	Recall	91.5%	93.5%	73.7%	**92.6%**
		8.5%	6.5%	26.3%	7.4%

Image 4		Actual values			
		B	V	R	Precision
Predicted values	B	8576	1371	66	85.6%
		9.5%	1.5%	0.1%	14.5%
	V	149	74224	1222	98.1%
		0.3%	82.5%	1.4%	1.9%
	R	7	1125	3160	73.6%
		0.0%	1.3%	3.5%	26.4%
	Recall	97.1%	96.7%	71.0%	**95.5%**
		2.9%	3.3%	29.0%	4.5%

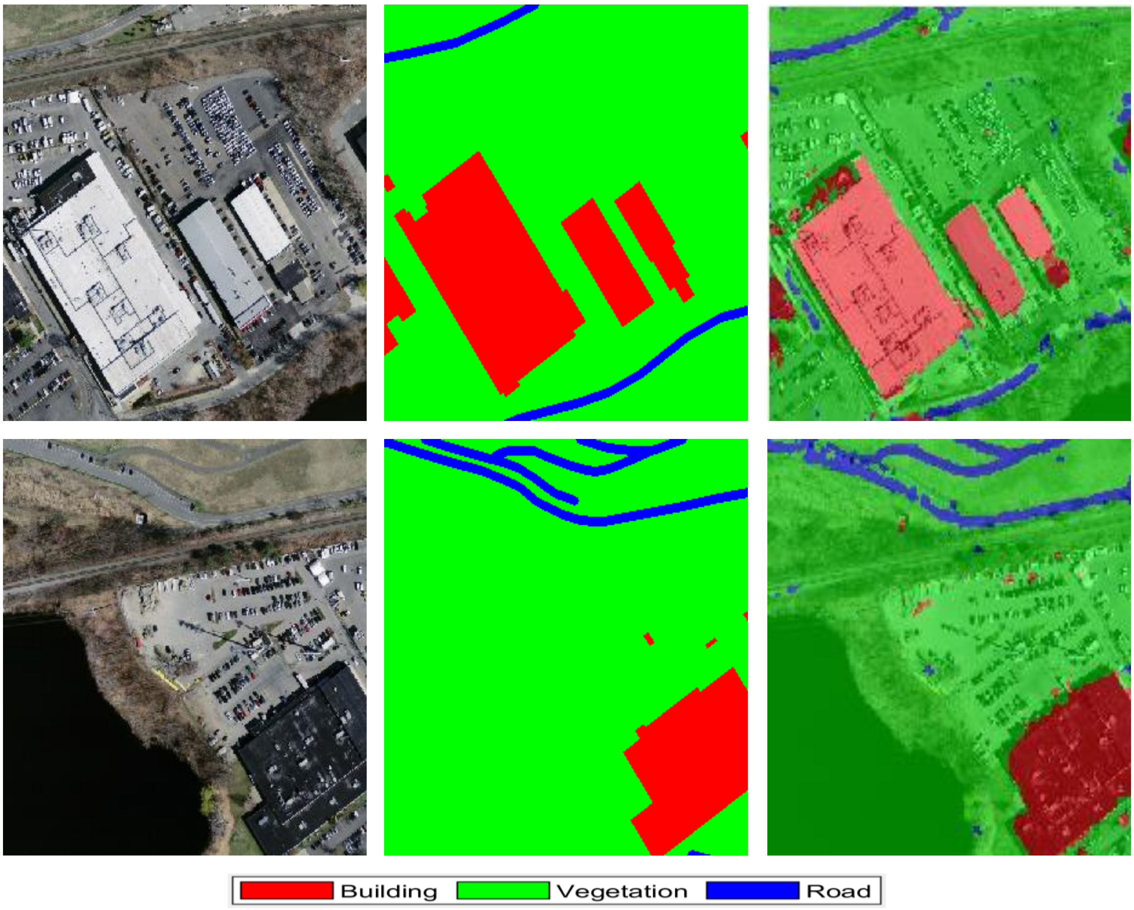


Figure 5.8: Original images (left column), ground truth images (middle column), and predicted images (right column) for image 3 (top) and image 4 (bottom).

5.1.2.2.3 Image 5 and 6 Results Analysis

Images 5 and 6 are the ones that achieved the best performance, especially in terms of buildings and roads. As we can see in Table 5.7 and Figure 5.9, the roofs were clearly discernible, and the network could segment them correctly. The

Confusion Matrices indicate an accuracy of more than 97% for class one, and more than 80% for class three, while the performance for class two is lower. This could be due to the shortage of vegetation in the training set for images 5 and 6.

Table 5.7: Confusion Matrices for test image 5 (left) and image 6 (right).

Image 5		Actual values			
		B	V	R	Precision
Predicted values	B	33496	38	349	98.9%
		88.1%	0.1%	0.9%	1.1%
	V	59	626	120	77.8%
		0.2%	1.6%	0.3%	22.2%
	R	509	152	2676	80.2%
		1.3%	0.4%	7.0%	19.8%
	Recall	98.3%	76.7%	85.1%	**96.8%**
		1.7%	23.3%	14.9%	3.2%

Image 6		Actual values			
		B	V	R	Precision
Predicted values	B	34226	31	789	97.7%
		90.0%	0.1%	2.1%	2.3%
	V	82	143	115	42.1%
		0.2%	0.4%	0.3%	57.9%
	R	330	30	2279	86.4%
		0.9%	0.1%	6.0%	13.6%
	Recall	98.3%	70.1%	71.6%	**96.4%**
		1.2%	29.9%	28.4%	3.6%

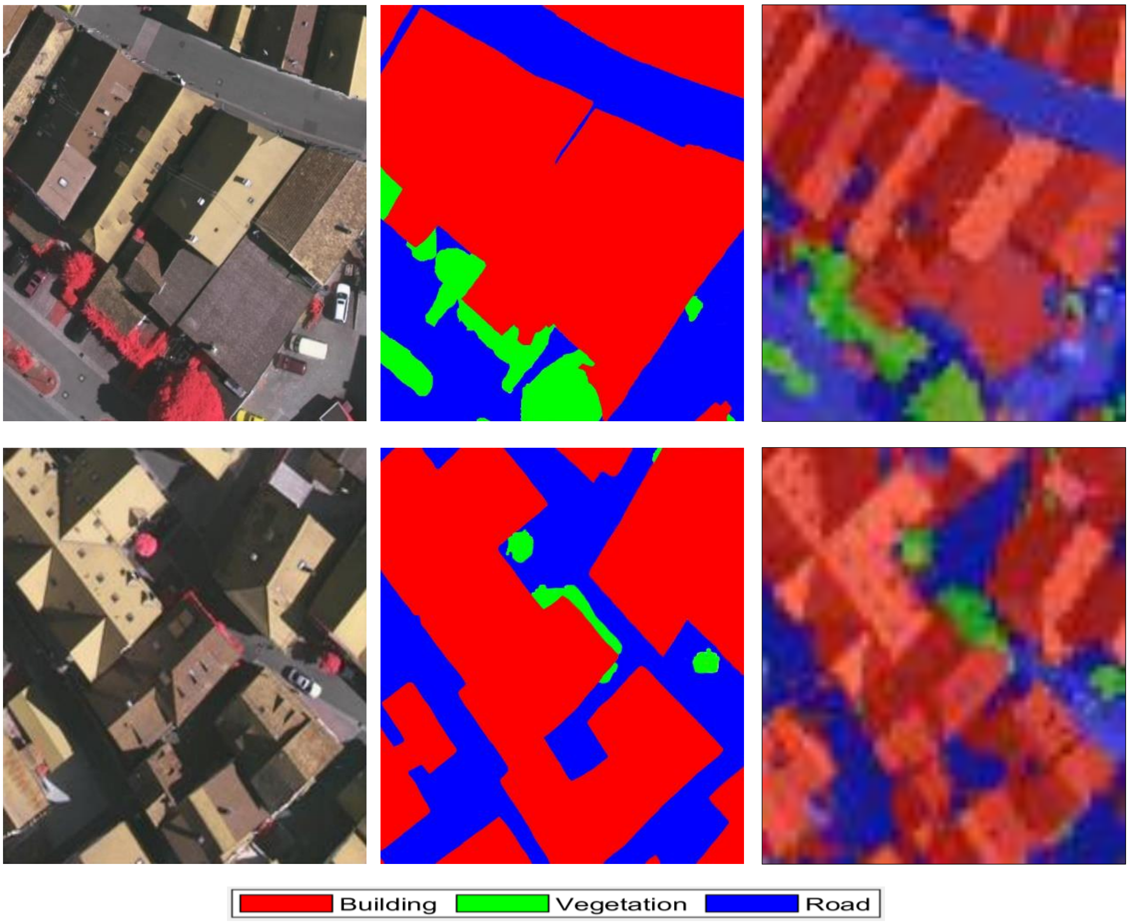


Figure 5.9: Original images (left column), ground truth images (middle column), and predicted images (right column) for image 5 (top) and image 6 (bottom).

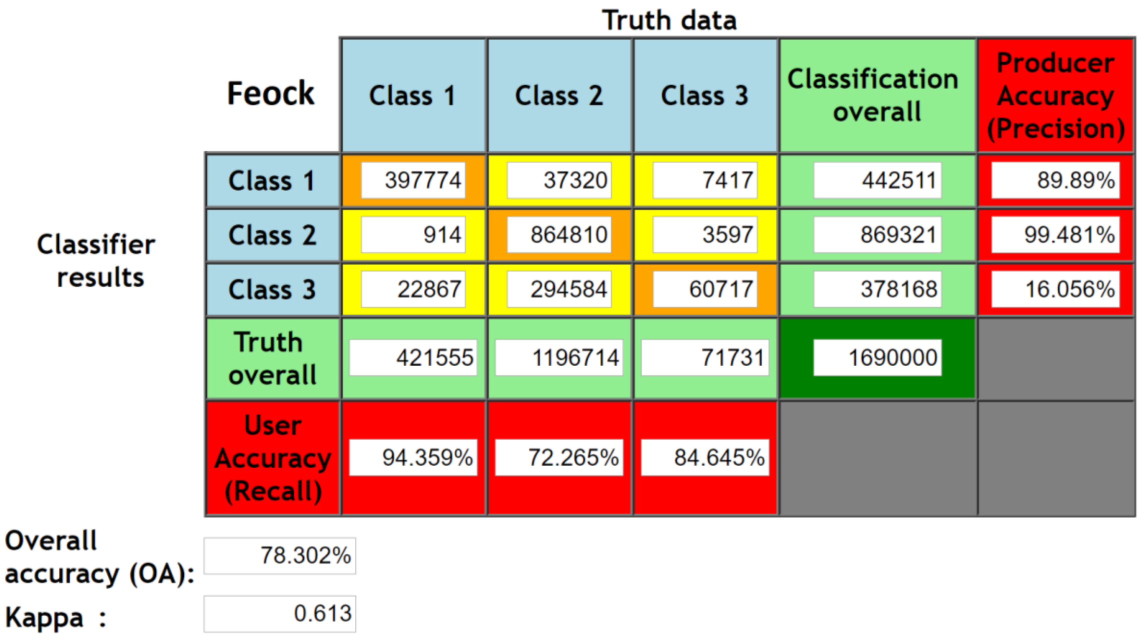
5.1.2.3 Comparison to Prior Art Studies

In this section the results achieved are compared to the prior findings, introduced in section 2.4.3.5.2, Chapter 2. Although the datasets are not of the same type (e.g. some use hyperspectral, some use elevation etc.) and each case uses different tools and software, this comparison provides some indication about the effectiveness of the proposed method. The overall accuracy, obtained by the average of the 6 test cases, is used as the comparison metric. The other required values are taken from the cited study in [226], where the segmentation is performed by training multiple simple neural networks (1 convolution layer, 50 filters) and combining the results. The comparison results are shown in Table 5.8. It is evident that the proposed method outperforms the methods, not based on deep learning, excluding the cases of [170] and [171], which, however, take advantage of a wider dataset, composed by information from more than one source. As for the results obtained using convolutional neural networks in [226], the difference is certainly due to the fact that the presented neural network was trained with much less data. This is an essential aspect in deep learning, while in future studies the plan is to increase the size of the used dataset. The results, however, are very promising, even with the limitations that have been presented. A further comparison to the classification, according to the 2nd scenario, applied in section 4.1.2, Chapter 4, where the same dataset is employed, is also included. The total accuracy achieved in the 2nd scenario is 93.7%, for the object-based approach, before the weighted sum-based ensemble classification enhancement (ParetoEnsemble classifier), which is comparable to the results obtained in this section. As far as the single images are concerned, this method outperforms scenario 2, in many cases, such as in images 2, 4, 5, and 6.

Table 5.8: Comparison between different segmentation methods.

Method	OA (%)	Data	Classes
Fuzzy C means [165]	68.9	Aerial image, laser scanning	4 (vegetation, buildings, roads, and open areas)
Segmentation and classification tree method [166]	70	Multispectral aerial imagery	5 (water, pavement, rooftop, bare ground, and vegetation)
Classification Trees and Test Field Points (TFP) [167]	74.3	Aerial image	4 (buildings, trees, ground, and soil)
Segmentation and classification rules [168]	75	Multispectral aerial imagery	4 (buildings, hard standing, grass, trees, bare soil, and water)
Region-based GeneSIS [169]	89.86	Hyperspectral image	9 (asphalt, meadows, gravel, trees, metal sheets, bare soil, bitumen, bricks, and shadows)
Object-based Imagery Analysis (OBIA) [170]	93.17	Aerial orthophotography and DEM	7 (buildings, roads, water, grass, tree, soil, and cropland)
Knowledge-based method [171]	93.9	Multispectral aerial imagery, laser scanning, Digital Surface Models (DSM)	4 (buildings, trees, roads, and grass)
CNN [226]	94.49	Multispectral orthophotography imagery, DSM	5 (vegetation, ground, roads, buildings, and water)
This CNN work	92.63	Satellite images	3 (buildings, vegetation, and roads)

Feock, the study image in this research, is tested by using the best performing parameters setting combination, as derived from the six images tested earlier. However, Feock is a very big image, compared to the six images used, so it is computationally expensive to train the CNN model, as this also requires a very high-performance computer system. To make the training of Feock possible, by using a moderate level computer system, the image and its ground truth are cropped to smaller sized sub images, i.e., 128x128 resolution, using a cropping function in MatLab. Next, cropped images are divided into 5-fold cross-validation groups, where, in each fold, data augmentation methods are applied on the 4 training folds images, while the validation (testing) is applied on the one remaining fold image. At the end of cross-validation process, testing folds provide results for all sub images of the divided Feock, which are then combined, to generate test prediction for the whole Feock. Predicted Feock is compared to ground truth of Feock, in order to produce the performance matrix for Feock image testing. Figure 5.10 shows Confusion Matrix for Feock image, used to estimate the classification accuracy of the SegNet classifier, for Feock image which is 78.3%. This matrix is used for comparison to other classification schemes results, in order to select the most suitable type of classification algorithm, for Feock image and unknown images testing.



		Truth data				
	Feock	Class 1	Class 2	Class 3	Classification overall	Producer Accuracy (Precision)
Classifier results	Class 1	397774	37320	7417	442511	89.89%
	Class 2	914	864810	3597	869321	99.481%
	Class 3	22867	294584	60717	378168	16.056%
	Truth overall	421555	1196714	71731	1690000	
	User Accuracy (Recall)	94.359%	72.265%	84.645%		

Overall accuracy (OA): 78.302%

Kappa : 0.613

Figure 5.10: Confusion Matrix results for Feock image testing, where Class 1, 2 and 3 represent buildings, vegetation and roads, respectively.

5.1.3 Conclusion and Future Works of the 3rd Scenario

Deep learning is receiving growing interest from the academic community, and the availability of more powerful hardware allows for the development of complex applications. Among these, semantic segmentation is undoubtedly one of the most popular and challenging. It is known that deep learning requires thousands of images in order to achieve good performances. Unfortunately, accessing the required amount of data combined with good quality labelled ground truth for a high-accuracy neural network is not feasible for everyone. In this chapter, we applied semantic segmentation to different satellite images representing urban scenes with different proportions of buildings, vegetation and roads, using a small dataset compared to the ones used in the same field. A convolutional neural network based on SegNet was employed using a limited data set which we expanded with "hard" augmentation.

Different parameter settings (i.e., training and validation, number of layers, number of filters per layer, and the kernel size) were tested for the 6 images shown in Figure 4.16, Chapter 4, and the best performing setting was selected to be used for the training and testing of Feock image. The results show promising performance of the network for the 6 images while a little compromised performance considering the varying conditions (such as different image size, resolution and samples imbalance of classes from the 6 images) which were used to pick parameter settings.

The scarcity of the dataset does not prevent the network from having high test accuracy, especially for some images, as it did not tend to produce overfitting during the training phase. Moreover, our model is very lightweight, resulting in fast inference with respect to more complex neural networks. This work applies state-of-the-art deep learning methods to remotely sensed images and works well even when only a small amount of data is available. The author believes that even better performances can be achieved with more data.

5.2 Comparison of Classifications for the Three Scenarios

After applying three different scenarios for the classification of satellite images (in Chapters 4 and 5), and achieving best possible results from these classification algorithms. The next goal is to pick the most suitable classifier amongst the classifiers used in the three scenarios and use the training of that classifier for the prediction of an unknown image to be classified for the modelling of stormwater in the InfoWorks ICM model. Therefore, all the best results from each of the three classification scenarios are collected and compared in the form of a bar chart to pick the most accurate classifier for the Feock image. For Scenario 1, one setting for CT and four settings for RF (i.e., RF10, RF20, RF50 and RF100) were used. As mentioned earlier, RF100 was found to generate the best results amongst all RF trees used in this study. That is why CT and RF100 were used from Scenario 1 for the comparison. In Scenario 2, many different settings concerning SLIC size were used for classification, and SLIC 50,000 for unbalanced data is considered to be the most suitable SLIC concerning computational costs and performance. Therefore, SLIC 50,000 was used from Scenario 2 for eight individual classifiers and the ParetoEnsemble classifier for comparison from Scenario 2 in the bar graph. In Scenario 3, many combinations of parameter settings were compared for the six selected images, and the best-performing combination setting (which is 4 layers with a filter size of 64, as explained in Table 5.2) was used for the testing of the Feock image. The results of CNN classification for the Feock image by applying the best combination setting were used for comparison with the other two scenario results to select the best classification method for the Feock image and unknown image testing in the future.

After comparing all classification scenarios' performance results in the bar graph shown in Figure 5.11, it was observed that the ParetoEnsemble classifier from Scenario 2 works best regarding classification accuracy amongst all classification algorithms used in this study. Only the trained models that are to be used in the ParetoEnsemble were saved after applying training on Feock image features to be used for the testing of unknown images later because there is no need to apply training again, and the classifiers that were trained only once can be used for any unknown image testing at any time.

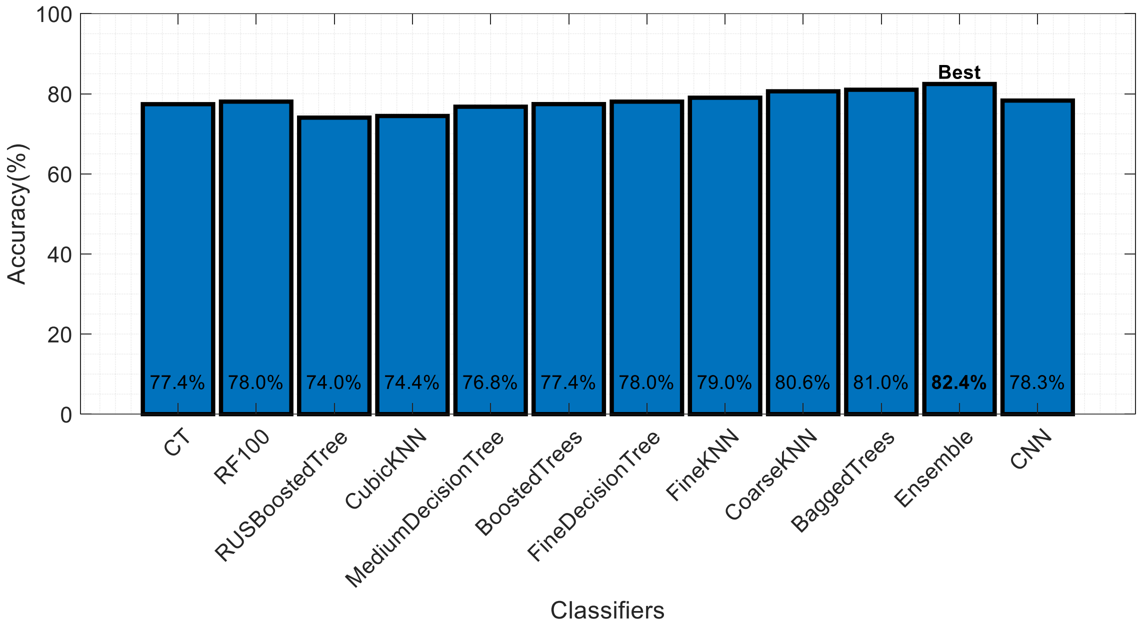


Figure 5.11: Performance comparison of classifiers used in the three scenarios of classification for Feock.

5.3 Unknown Data Testing

The aim of any classification model that uses supervised learning is to train a classification model to be used for the testing of other similar unknown data. Due to the huge amount of effort required for data labelling by humans, it is not possible to get data labelled by a human each time labelling is required. To address this problem, researchers have introduced the concept of data learning, which essentially generalises the knowledge learned on some auxiliary data to boost learning on the target domain task.

To analyse the performance of the best performing classifier (among the three classification scenarios) generalisation on this system, a dataset of two unknown satellite images (Penelewey and Playing Place) was compiled. The Feock image was used to train the system by extracting different superpixel-based (SLIC) objects then extracting features and training the classifiers. The two unknown images, provided by the Pell Frischmann Company, that were used for testing are called Penelewey [251] and Playing Place [252] (Figures 5.12 and 5.13). Both selected maps show the southwest village of city Truro in Cornwall, UK. They have image pixel resolutions of 727x902 and 1,024×1,375 for Penelewey and Playing Place, respectively. Captured conditions and attributes for both images, i.e. Penelewey and Playing Place, are shown in Figure 5.14 and Figure 5.15, respectively.

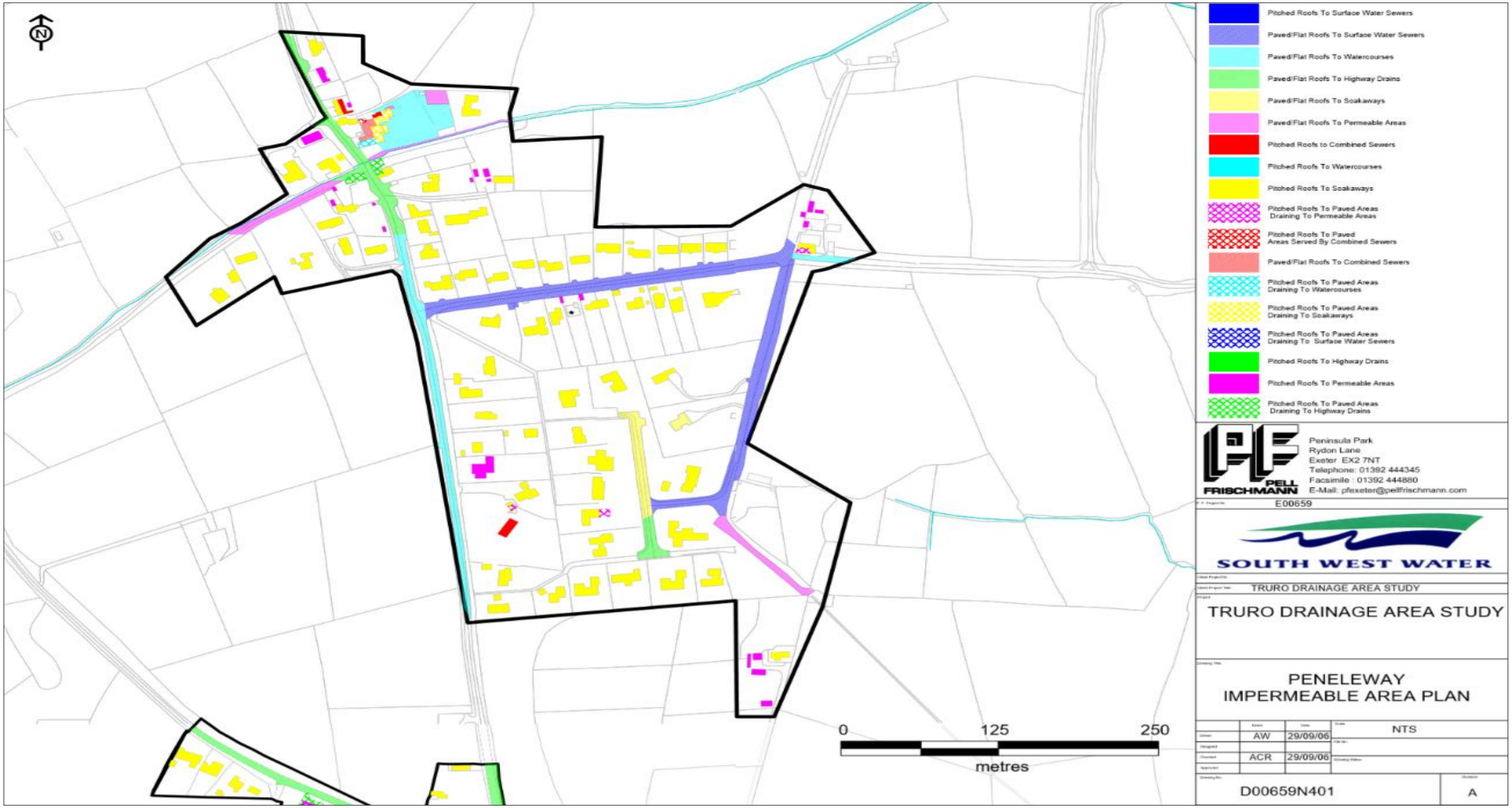

N
0
125
250
metres
Pitched Roofs To Surface Water Sewers
Paved/Flat Roofs To Surface Water Sewers
Paved/Flat Roofs To Watercourses
Paved/Flat Roofs To Highway Drains
Paved/Flat Roofs To Soakaways
Paved/Flat Roofs To Permeable Areas
Pitched Roofs to Combined Sewers
Pitched Roofs To Watercourses
Pitched Roofs To Soakaways
Pitched Roofs To Paved Areas Draining To Permeable Areas
Pitched Roofs To Paved Areas Served By Combined Sewers
Paved/Flat Roofs To Combined Sewers
Pitched Roofs To Paved Areas Draining To Watercourses
Pitched Roofs To Paved Areas Draining To Soakaways
Pitched Roofs To Paved Areas Draining To Surface Water Sewers
Pitched Roofs To Highway Drains
Pitched Roofs To Permeable Areas
Pitched Roofs To Paved Areas Draining To Highway Drains
PELL FRISCHMANN
Peninsula Park
Rydon Lane
Exeter EX2 7NT
Telephone: 01392 444345
Facsimile : 01392 444880
E-Mail: pfexeter@pellfrischmann.com
E00659
SOUTH WEST WATER
TRURO DRAINAGE AREA STUDY
TRURO DRAINAGE AREA STUDY
PENELEWAY
IMPERMEABLE AREA PLAN
AW
29/09/06
NTS
ACR
29/09/06
D00659N401
A

Figure 5.12: IAS data map with an urban catchment of Penelewey [251].

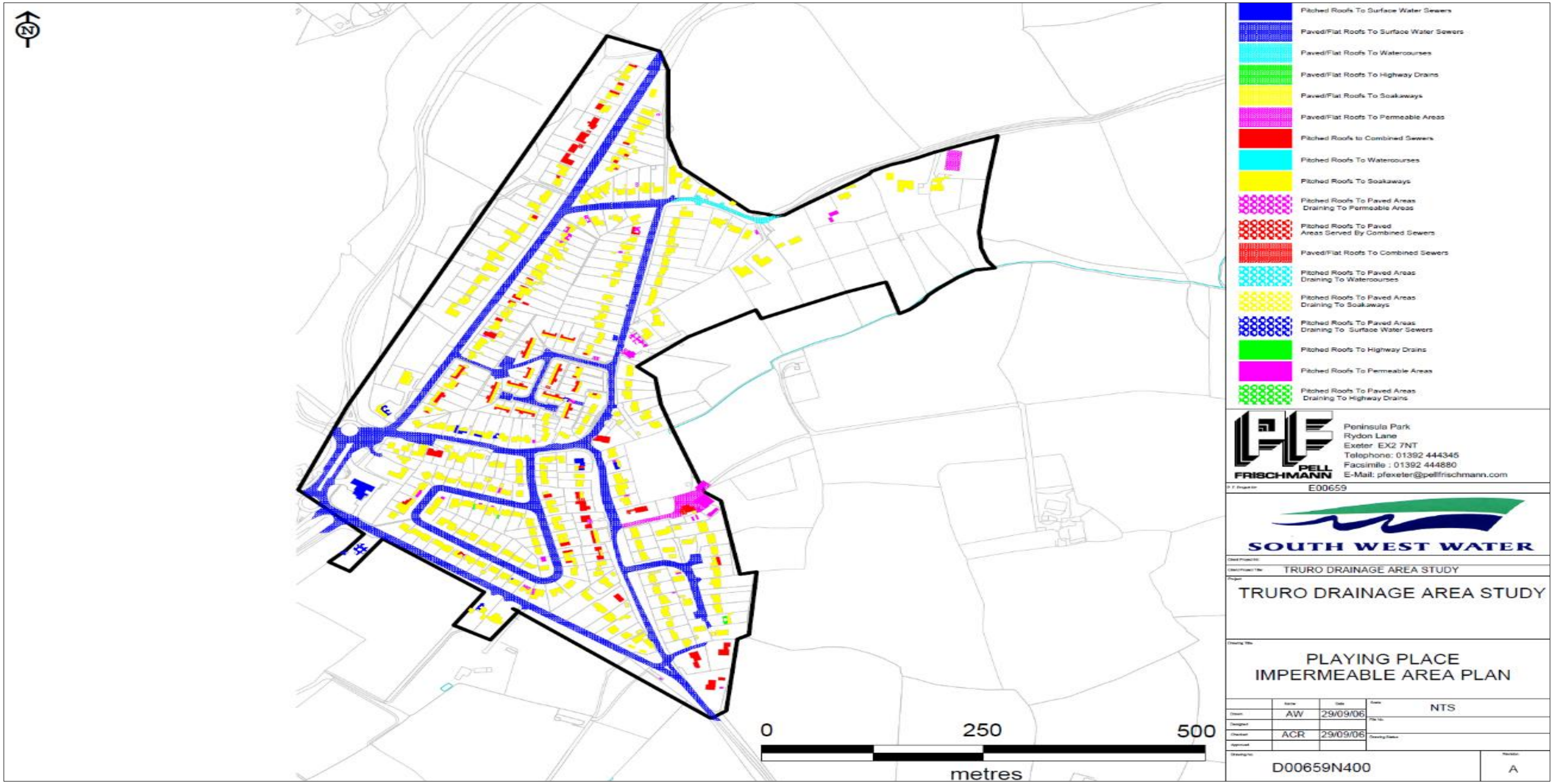

N
0
250
500
metres
Pitched Roofs To Surface Water Sewers
Paved/Flat Roofs To Surface Water Sewers
Paved/Flat Roofs To Watercourses
Paved/Flat Roofs To Highway Drains
Paved/Flat Roofs To Soakaways
Paved/Flat Roofs To Permeable Areas
Pitched Roofs to Combined Sewers
Pitched Roofs To Watercourses
Pitched Roofs To Soakaways
Pitched Roofs To Paved Areas Draining To Permeable Areas
Pitched Roofs To Paved Areas Served By Combined Sewers
Paved/Flat Roofs To Combined Sewers
Pitched Roofs To Paved Areas Draining To Watercourses
Pitched Roofs To Paved Areas Draining To Soakaways
Pitched Roofs To Paved Areas Draining To Surface Water Sewers
Pitched Roofs To Highway Drains
Pitched Roofs To Permeable Areas
Pitched Roofs To Paved Areas Draining To Highway Drains
PELL FRISCHMANN
Peninsula Park
Rydon Lane
Exeter EX2 7NT
Telephone: 01392 444345
Facsimile : 01392 444880
E-Mail: pfexeter@pellfrischmann.com
E00659
SOUTH WEST WATER
TRURO DRAINAGE AREA STUDY
TRURO DRAINAGE AREA STUDY
PLAYING PLACE
IMPERMEABLE AREA PLAN
AW
29/09/06
NTS
ACR
29/09/06
D00659N400
A

Figure 5.13: IAS data map with an urban catchment of Playing Place [252].

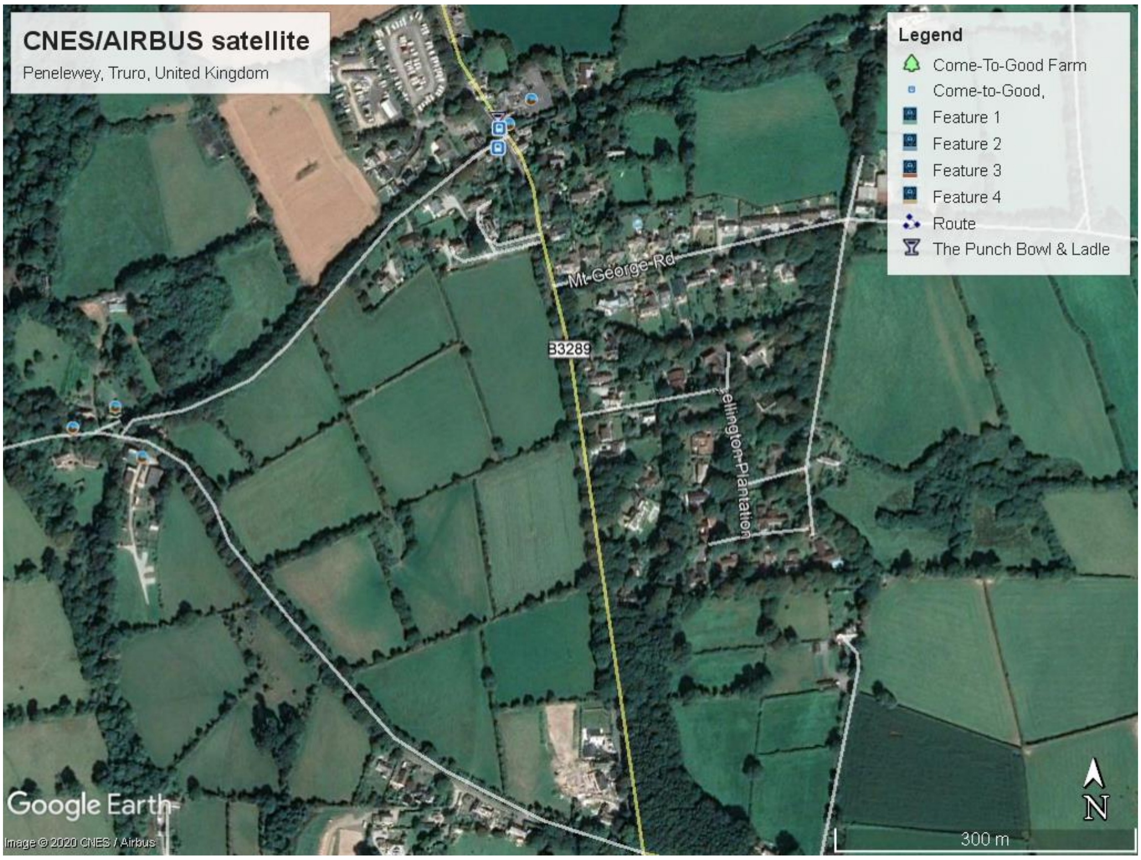


Figure 5.14: Penelewey image captured details.

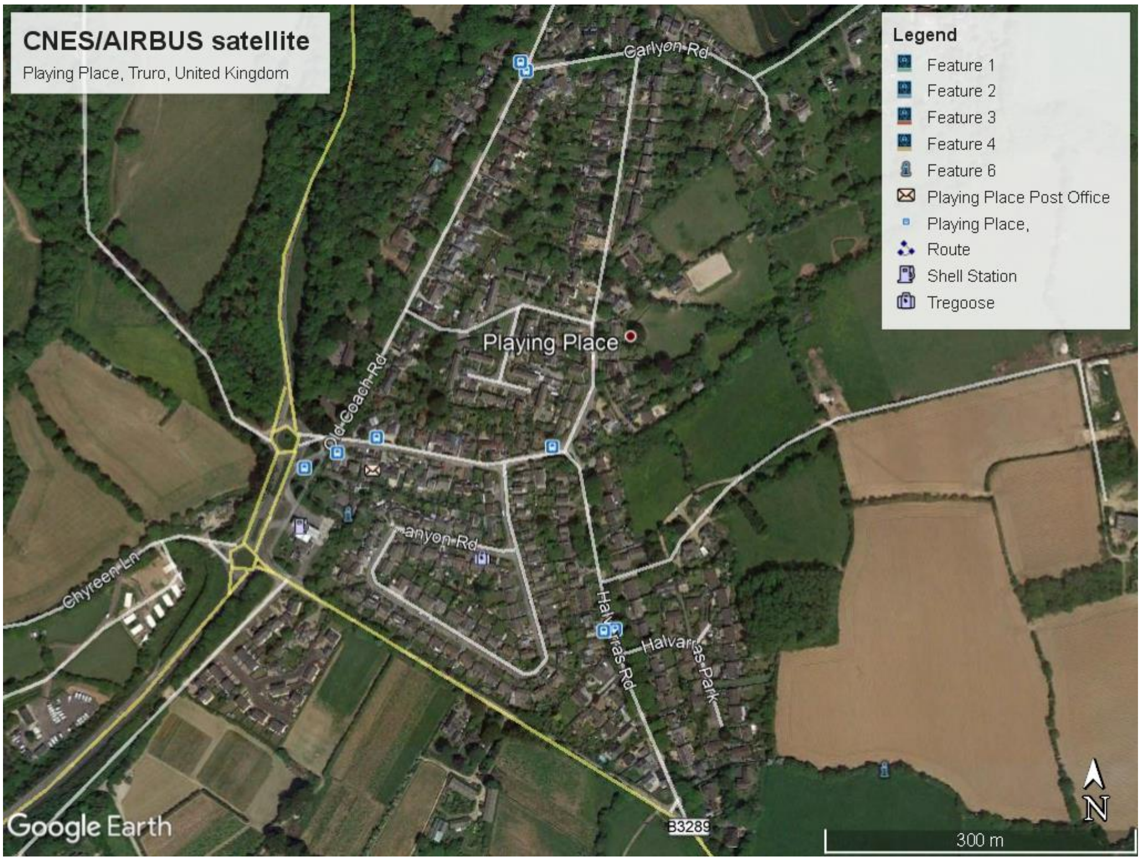


Figure 5.15: Playing Place image captured details.

The essential operations mentioned in section 3.2, Chapter 3 are considered for preparing the above maps for use in the other phases. Figure 5.16, top left and bottom left, shows the captured real-world satellite images of Penelewey and Playing Place, respectively, by using Google Maps Customizer [211]. Ground truth images for Penelewey (Figure 5.16 top right) and Playing Place (Figure 5.16 bottom right) were also compiled to compare the model predictions to actual ground truth to assess the performance of the generalised models.

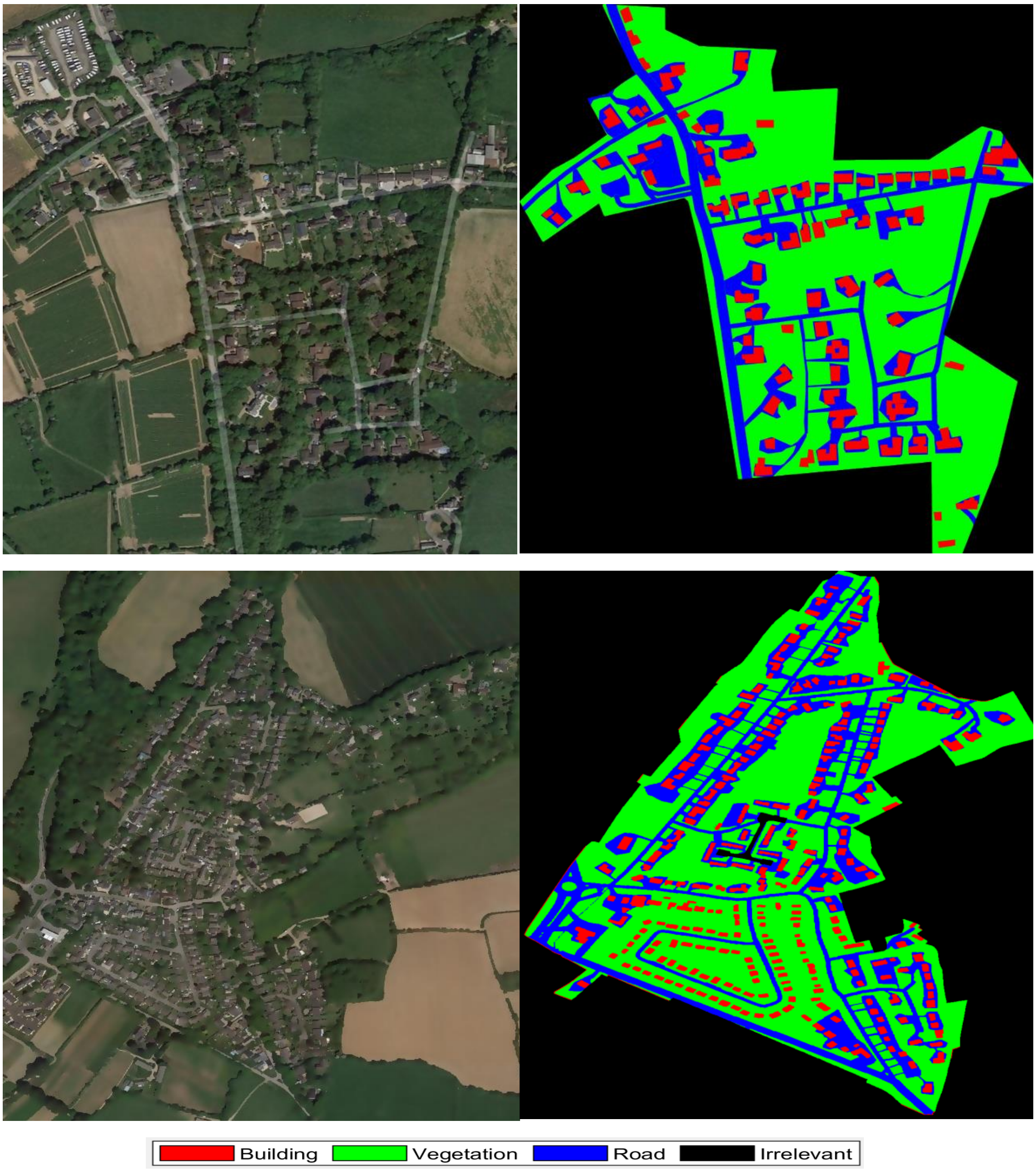


Figure 5.16: Penelewey satellite image on the top left, Penelewey ground truth on the top right, Playing Place satellite image on the bottom left, and Playing Place ground truth on the bottom right.

Next, the unknown image is segmented by applying SLIC segmentation for SLIC count of 50,000 (selected from experiments), and 13 selected features were extracted from each object and compiled in the form of a test dataset. Then predictions were made for these unknown test objects by using ParetoEnsemble of selected trained classifiers from the training phase. After we made predictions from the selected classifiers, we utilised them as ensemble predictions by using weighted-sum equations, which gives predicted labels for all objects for the ParetoEnsemble classifier, which are final predictions for the unknown image. After that, a predicted image was created from the predicted object labels by assigning the same labels to all pixels comprising that object. Next, the actual ground truth labels of the unknown image were compared to those of the ParetoEnsemble predicted labels to estimate the generalisation performance for both unknown images. The same unknown testing steps were applied on both Penelewey and Playing Place images and predicted images for both were obtained. Then the predicted images for both unknown images were compared to the respective ground truth images to create misclassified images, which makes it more feasible to assess the prediction correctness of both images.

The same object prediction method can also be applied on totally unknown images for which there is no actual ground truth available, and a predicted image for the unknown image can be created. The colour of each pixel in the predicted image will determine the class of each pixel (i.e., red, green and blue pixels mean buildings, vegetation and blue classes, respectively).

Because SLIC 50,000 is observed to be the best SLIC from previous experiments, Penelewey and Playing Place images were tested for SLIC 50,000, and misclassified images were created for both images by comparing predicted images and actual ground truth images, which also provided testing accuracy information for the unknown images. Figure 5.17 shows the predicted and misclassified image for Penelewey. The left image shows the predicted image for Penelewey, where red, blue, green and black pixels represent predicted buildings, roads, vegetation and irrelevant areas, respectively. Predicted image pixels were compared to actual ground truth image pixels to create the misclassified image on the right side of Figure 5.17, where yellow pixels represent incorrectly predicted pixels and red, blue and green pixels represent correctly classified pixels and black pixels represent irrelevant pixels. The unknown testing accuracy for the Penelewey image was found to be 66.3%, considering the validation accuracy of 82.4% for the training of the Feock image using SLIC

50,000. The left side of Figure 5.18 shows the predicted image for the Playing Place image, and the right side shows the misclassified image. The yellow pixels show incorrectly classified pixels in the Playing Place image. The testing accuracy for the Playing Place image after comparison to the actual ground truth image was found to be 55.8%, and the validation accuracy for the Feock image was 82.4% for SLIC 50,000.

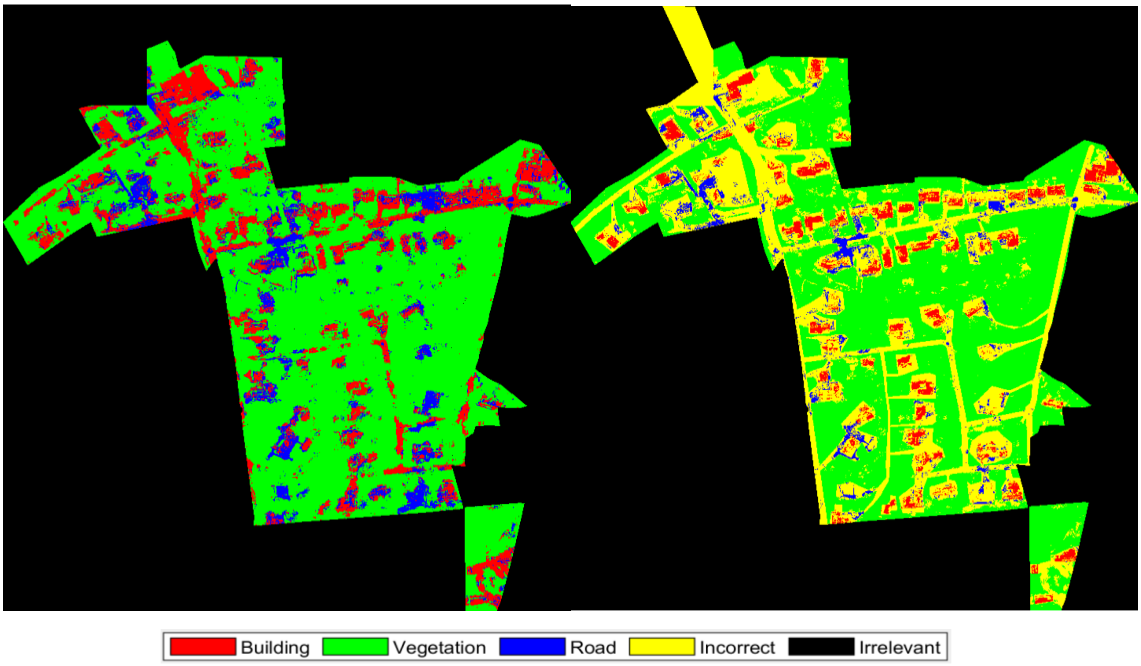


Figure 5.17: The predicted (left) and Misclassified (right) of Penelewey image for unknown testing.

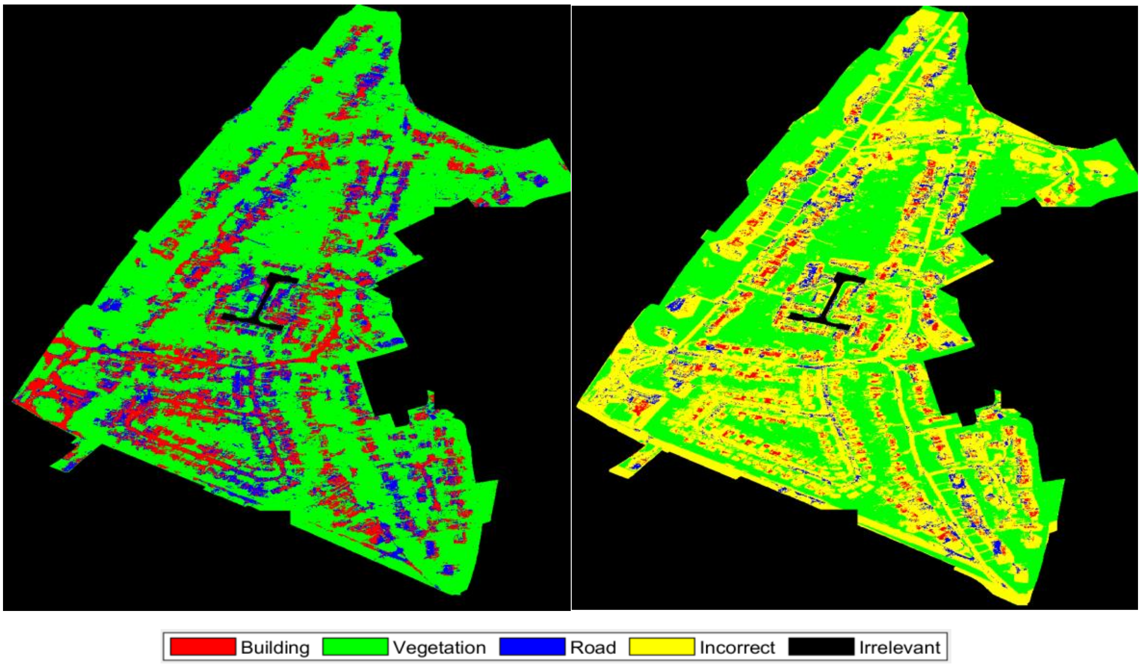


Figure 5.18: The predicted (left) and the misclassified (right) of Playing Place image for unknown testing.

It can be seen from the accuracy values of Penelewey and Playing Place images that the unknown testing accuracy is not good as the validation accuracy for the Feock image. The reason for this is that for a very high-performance generalised model, the system should be trained for a variety of images with different conditions, such as different zoom levels, resolutions and lighting conditions, because all these conditions' including vegetation area, buildings and roads, visual natures can greatly vary from image to image and place to place. However, the implementation of this idea needs compilation of many ground truth images which is a time-consuming task. Another factor to be considered here is the similarity between buildings and road-class objects. Buildings and road-class objects are very similar to each other, as shown in Figure 5.19, where red, green and blue circles are from buildings, vegetation and roads areas, respectively. It is evident that, there is too much similarity between red and blue circles, which makes it challenging for classifiers to differentiate between these classes of objects, compared to vegetation-class objects, marked by green circle (Figure 5.19).

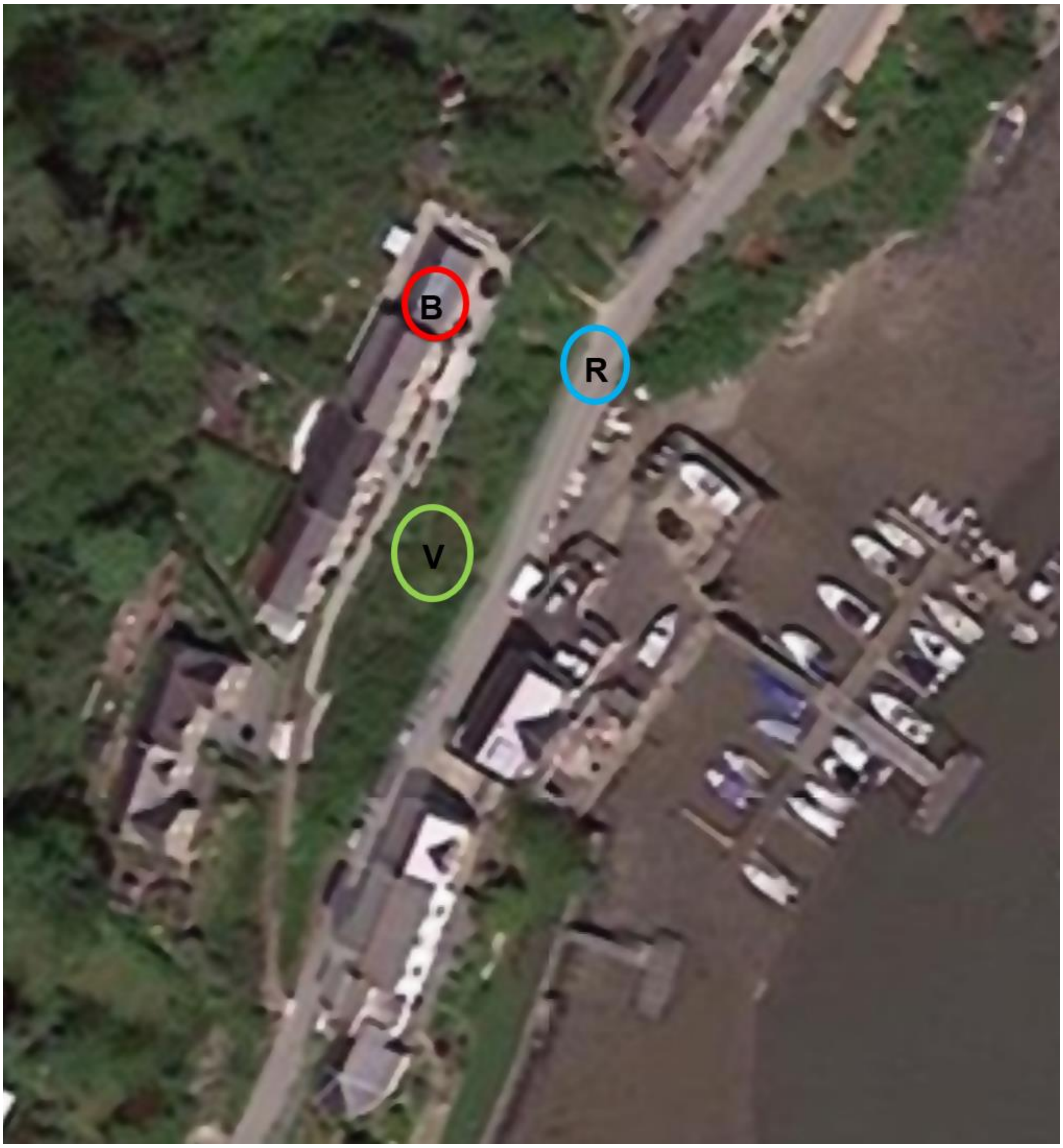


Figure 5.19: Visual similarity comparison of buildings and road-class objects in Feock image, marked with red and blue circles.

Therefore, the results achieved by testing unknown images after training on only one image are satisfactory considering the training conditions. If this trained system is tested on some part of the same training image, as was done in the cross-validation testing, then it works very well. The reason for this is that this system is well-trained for similar attributes while, for unknown images, it is quite tricky for trained models to differentiate between different classes of objects under different conditions. Therefore, to achieve a high-performance generalised trained model, the system needs to be trained with very high-resolution images that were captured at different locations and under differing capturing conditions, such as images with differing zoom levels, resolutions and lighting conditions, including images taken during the day and night. Also, the images containing effective discrimination between buildings and roads class objects is another important factor to be considered. Only after fulfilling all the mentioned conditions, it can be expected that the system will work well on some totally unknown images.

CHAPTER SIX

6 SURFACE WATER SYSTEM MODELLING

A satellite image with correctly classified areas is important to setting up parameters for a more reliable and accurate surface water modelling and runoff estimation, because the percentage area contribution, from different classes, when connected to InfoWorks, affects the modelling process. This chapter includes a detailed description of the connection between classification phase and surface water modelling phase, which is the final goal of this research study. The process of the required data acquisition, followed by the data conversion into a suitable form, for the classification phase, as well as the classification itself, are all discussed in this chapter. Finally, a detailed description of the classification results conversion, into InfoWorks ICM model input, is also included.

The first section of this chapter gives detailed information on the importance of surface water modelling and the role of InfoWorks ICM software in modelling of stormwater. This section also describes the modelling process, in terms of detailed theoretical background, along with an example of manual runoff estimation, by using runoff model equations.

In the next section classification of the satellite map of interest is carried out, into pervious surfaces (i.e., vegetation) and impervious surfaces (i.e., roofs and roads), by using trained classification models, derived from previous phase. The results are converted into InfoWorks model input format, by calculating percentages of the three surface type areas within each subcatchment.

After the conversion of classification results into InfoWorks model input, surface water modelling is applied to simulate runoffs. Two case studies, Penelewey and Playing Place, were selected to test the methodology. The simulated runoff obtained base on the parameters derived from the different unknown satellite images (Penelewey and Playing Place) was compared to the one obtained from the parameters determined by the ground truth image.

Since there are many parameterisations of SLIC considered for the testing of unseen images, a single SLIC is selected, based on the comparison between different SLIC performances in terms of ParetoEnsemble classification accuracy and ParetoEnsemble runoff accuracy.

Next, the comparison of best SLIC performance is carried out for two cases, including balanced and unbalanced case classifications, for Penelewey and Playing Place images, using SLIC 50,000, where it is concluded that the unbalanced case classification performs best, in terms of runoff estimation.

6.1 Modelling Stormwater in Urban Environment

Prediction of upcoming events is an essential part of disaster management. It helps the state disaster management agencies in taking the right protective measures, to avoid or minimise the damages made by an incident. One of the crucial environmental disasters is flooding, specifically, urban surface water flooding, which is very important, because its effects are immediate on the human population. The most critical indicators in the modelling analysis of surface water, in an urban environment, are the pervious and impervious areas. Predominantly, the building and paved areas often contribute to the increased runoffs in urban environment. To simulate flood dynamics, the rainfall-runoff modelling in InfoWorks first divides an area into multiple subcatchments, in which the surface flow will concentrate to a drainage node and enter into sewer system. In model set up, each subcatchment uses a parameter to represent the surface permeability, assuming the runoff within a percentage of area inside the catchment will infiltrate into soil instead of propagating toward the sewer system. Adequate representation of such areas in a hydraulic model is essential to accurate simulations. Figure 6.1 shows a step-by-step flow diagram of the runoff estimation process, as followed in this work, during modelling simulation of the InfoWorks.

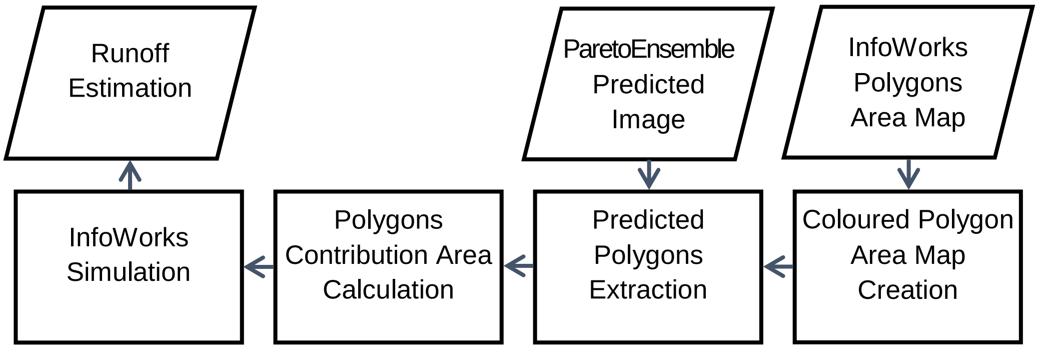

Runoff Estimation
ParetoEnsemble Predicted Image
InfoWorks Polygons Area Map
InfoWorks Simulation
Polygons Contribution Area Calculation
Predicted Polygons Extraction
Coloured Polygon Area Map Creation

Figure 6.1: Flow process of the proposed runoff modelling approach.

6.2 Stormwater System Modelling

Hydraulic models provide an approximate description of rainfall collection (stormwater), network performance, capturing the large-scale element of the system. Nonetheless, such systems require calibration, based on real-world data, to achieve reliable and accurate results. However, many factors, surrounding real-world network and catchment characteristics, are unknown and can influence the hydraulic performance of the network. Consequently, calibrating hydraulic models, to reflect real-world conditions accurately, is a time consuming and complicated process. Therefore, this research adopts the results from a well-calibrated InfoWorks model in Feock as the benchmark to examine the modelling results based on the parameters derived from satellite images.

This study employs a two-stage urban runoff forecasting approach, combining image classification techniques and InfoWorks ICM based modelling. The image classification part consists of automated processing of satellite images, as an aid to modelling surface water in the urban environment, by classifying the land-cover in an urban catchment into three classes: Area 1 (roofs), 2 (roads) and 3 (permeable area). In the next stage, Wallingford PR Equation was utilised to evaluate the potential of a partially automated surface water network model.

In this study, InfoWorks ICM software by Innovyze [204] is adopted, to improve the modelling of a surface water network. InfoWorks ICM is chosen, based primarily on its ability to create models for both sewer networks and surface water flow routes and is also considered and used as an industry standard for this type of modelling. By using the image classification technique described in previous chapters with InfoWorks ICM, engineers may be able to work with one unified model, by incorporating a range of environmental variables in a hydrological system. The software used for the creation of the model is version 7.5.4 of InfoWorks ICM.

6.3 InfoWorks ICM Modelling in Connection with Classification

The following steps are implemented to connect the classification phase to InfoWorks software to accurately model runoff and flooding. The very first step is extraction of Feock area map from the large drainage area of Truro. Figure 6.2 shows the highlighted Feock area in Truro map which is taken as a case study for this research, which is a small typical section of the Truro drainage area study map. The whole drainage model area is divided into multiple polygons/subcatchment areas.

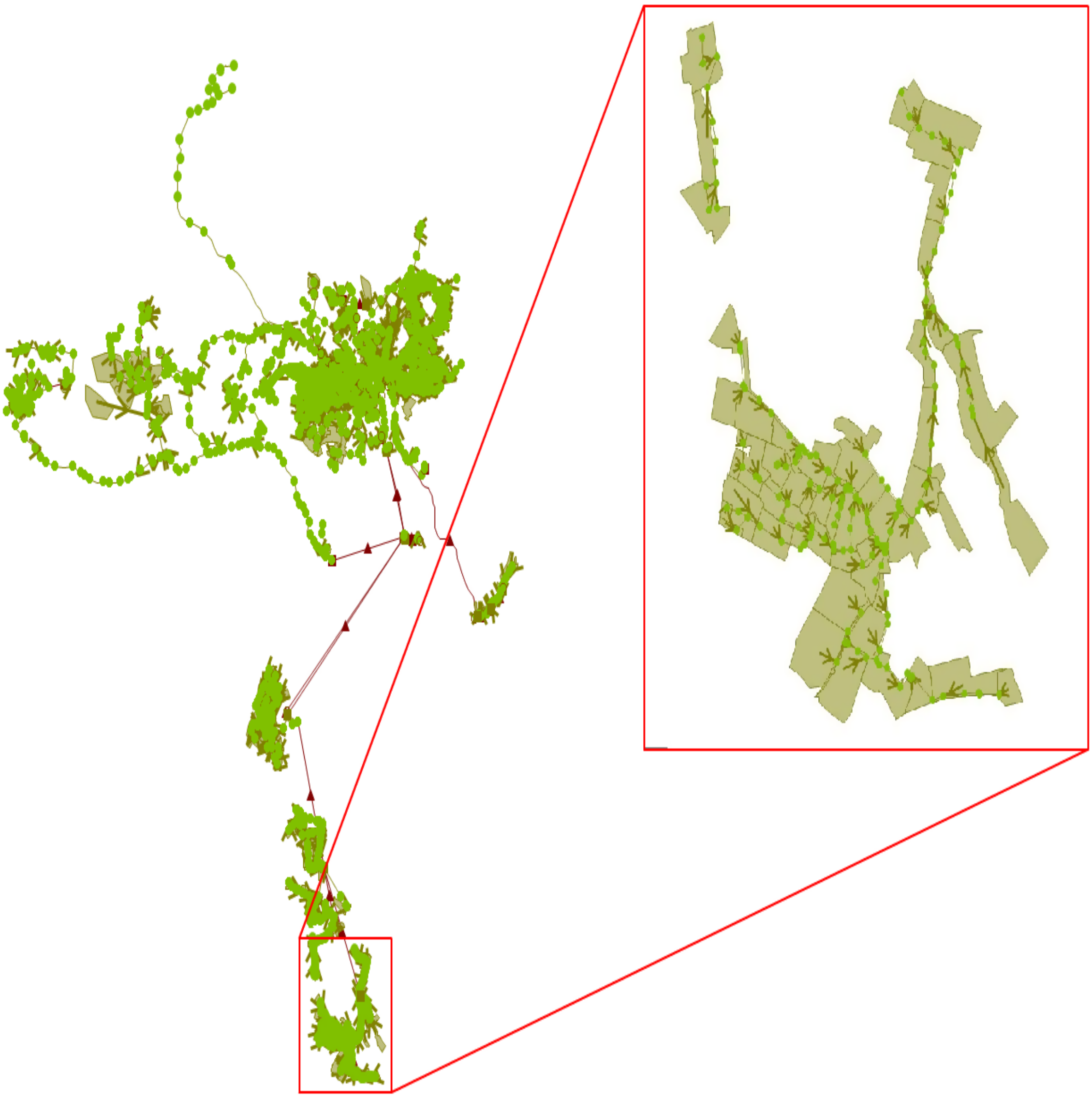


Figure 6.2: Feock area extraction as case study from Truro drainage area map.

In the InfoWorks ICM drainage network, the contribution area percentages of the three classes (i.e., buildings, vegetation and roads) are used as input of subcatchment parameters. To derive the contribution percentage values of polygons, the predicted image for the Feock area is required, which includes the prediction labels for each pixel of Feock, where a satellite view image of Feock is needed for this purpose. First, Feock area map is extracted from InfoWorks, while next the satellite view of that same map area and the ground truth image are compiled as described in section 3.2, Chapter 3. Different colour pixels in the ground truth image represent different class pixels identification (i.e., red pixels for buildings, green pixels for vegetation and blue pixels for roads). Next, 5-fold cross-validation based training and testing are applied to get a predicted image, having the same colour label markings as ground truth image, for different classes.

Many different parameterisations of SLIC are used for the 5-fold cross-validation classification, while SLICs higher than 50,000 are ignored, since no considerable further improvement in the performance is noted. Then, the percentage area contributions of the three classes are calculated, for each polygon. The polygons pixels locations for Feock map are unknown, in satellite Feock image. To match the polygons locations, in both Feock map and Feock satellite image, a polygon image, having different colours of each polygon, of the same size as satellite image, has been manually created, as shown in Figure 6.3 right. The left side image is from the InfoWorks, and the right side is created manually, locating pixels of each polygon inside the satellite image. The creation of a coloured polygon image is an important task to locate each polygon in satellite image. This is to be done manually each time we get a new case study image because there is no suitable way available in the InfoWorks to get the polygon pixel locations automatically in relation to satellite image form of same polygon area, which is acquired from a different source. This is one of the limitations of this work and can be resolved in future research works. Each polygon colour is saved for the ascending order of polygons, one by one. It is important to keep track of polygons order and colour because this order determines the input columns for each polygon in InfoWorks. Pixel locations of all polygons, in ascending order, are saved in a mat file, which is then used to locate each polygon pixel in the satellite and the predicted image to convert classification phase results into InfoWorks input.

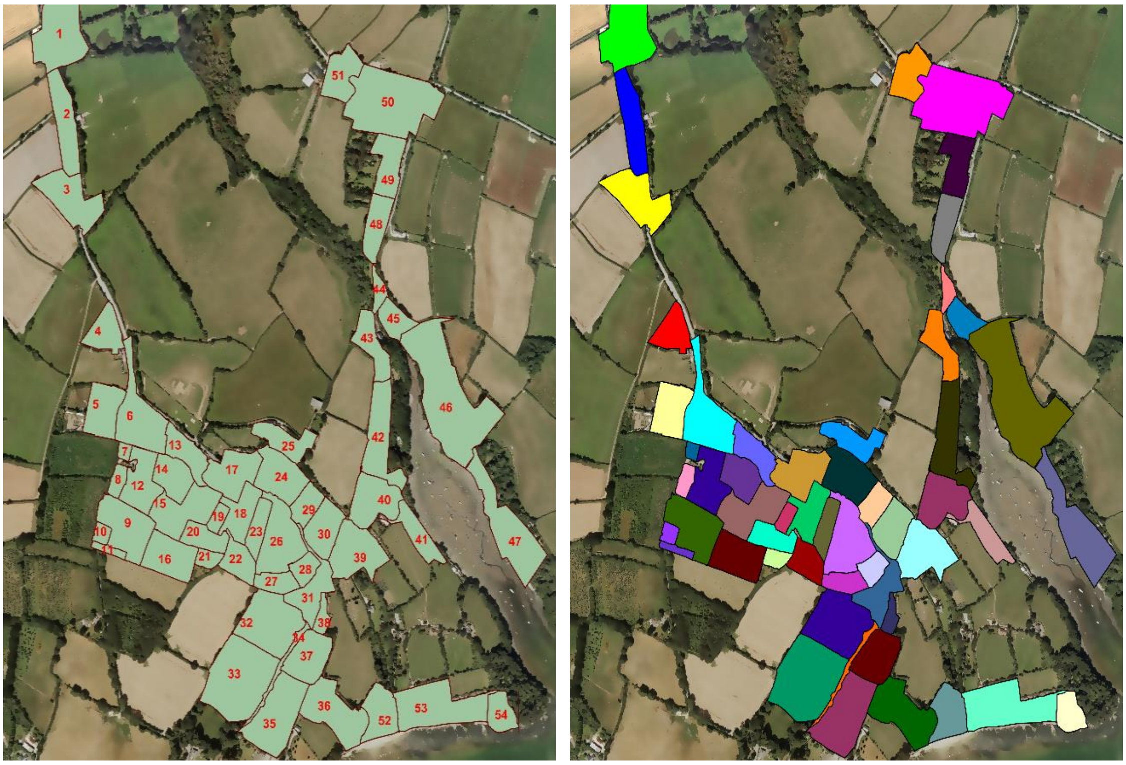


Figure 6.3: Feock map marked with 54 polygons, according to the drainage network in the InfoWorks software, showing overlay of the 54 polygons map on the corresponding satellite image (left) and overlay of the coloured 54 polygons map on the corresponding satellite image (right).

After applying segmentation and classification on Feock image, the predicted image from classification results is divided into 54 polygons (same number of polygon areas as in the InfoWorks), by locating pixels of all polygons, based on pixels locations, extracted from coloured polygon image and saved as mat file. After extracting predictions of different polygons, the contribution area percentage of each class, in all polygons which represents the percentage of permeable and impermeable areas in each polygon, is calculated, according to Equation (6.1), where the value of i varies within the range 1 to 3, based on class 1-3, where the sum of percentage contribution of all 3 classes is equal to 1.

$$\begin{aligned}&Class\ i\ area\ contribution\ percentage\\&= ((number\ of\ pixels\ of\ class\ i\ in\ polygon)\\&/(total\ number\ of\ pixels\ for\ all\ classes\ in\ polygon)) * 100\end{aligned} \tag{6.1}$$

The InfoWorks is fed with these percentages, for the estimation and prediction of runoff, based on these contribution values.

6.4 Runoff Modelling

As an example, we have taken subcatchment 46 of Feock map (Figure 6.4), which is matching to subcatchment ID no. SW82386802 in the runoff table of the InfoWorks model. After applying classification on Feock satellite image, a predicted image is created, which is then used to estimate the contribution area percentages.

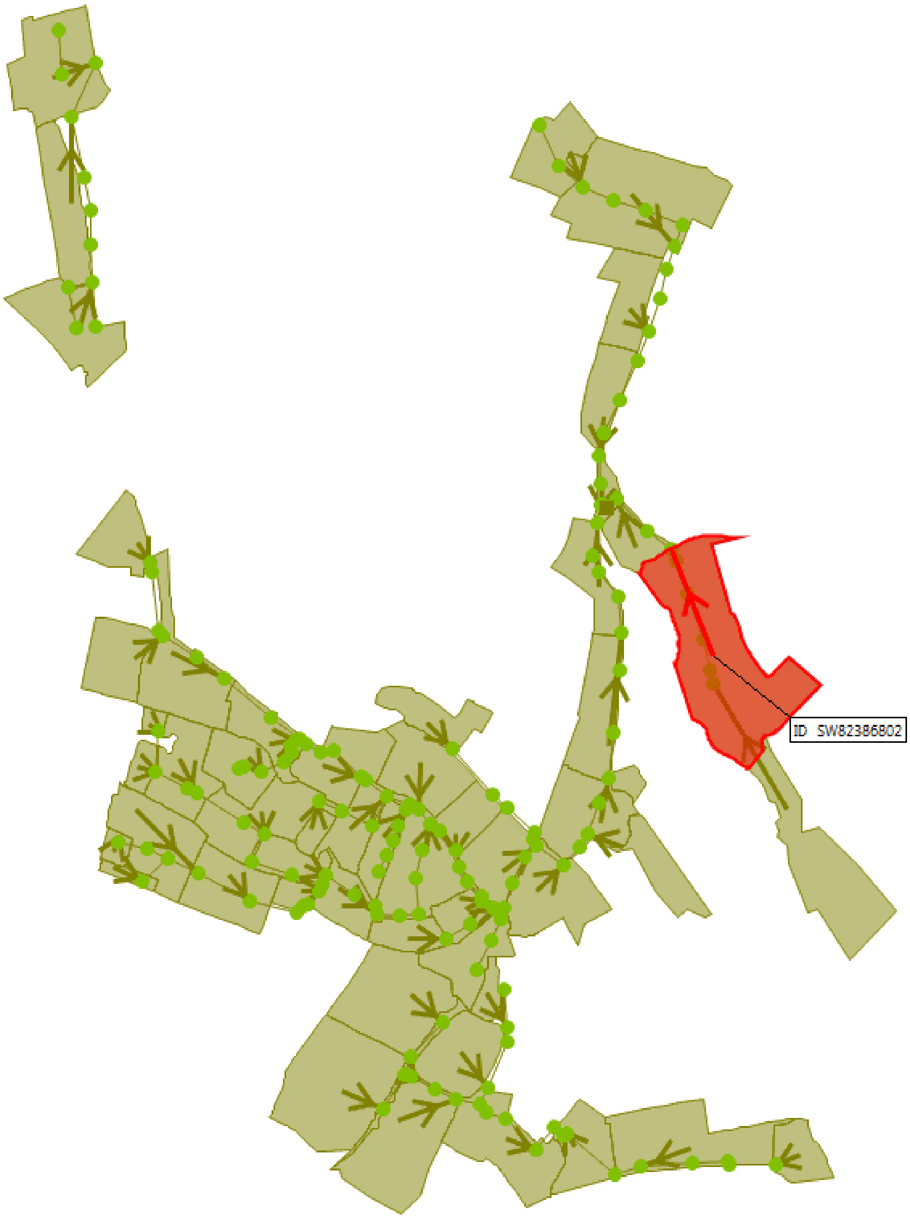
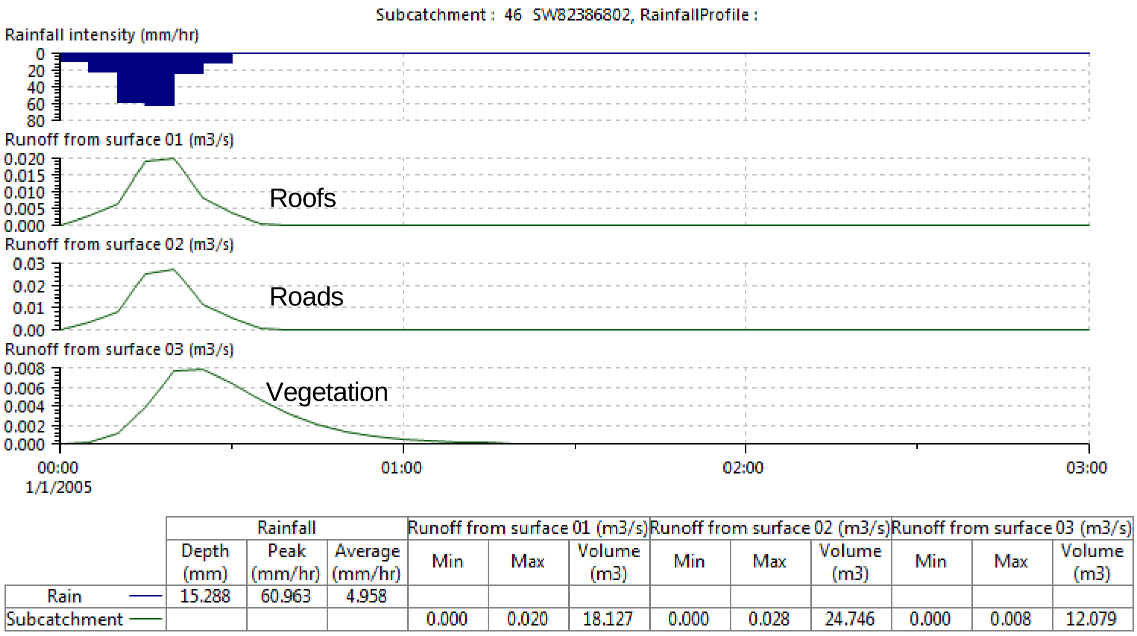


Figure 6.4: Subcatchment 46 marked as a red polygon in Feock area map.

Figure 6.5 shows the simulation results of modelling, applied on the actual labelling ground truth of Feock satellite image, for each of the three classes (i.e., buildings, vegetation and roads). The hydrographs show the runoff estimations, for each of the three surfaces/classes, individually of subcatchment 46. This simulation plot also shows the total rainfall intensity values and duration, along with min, max and volume of estimated runoffs, for each of the surfaces/classes.

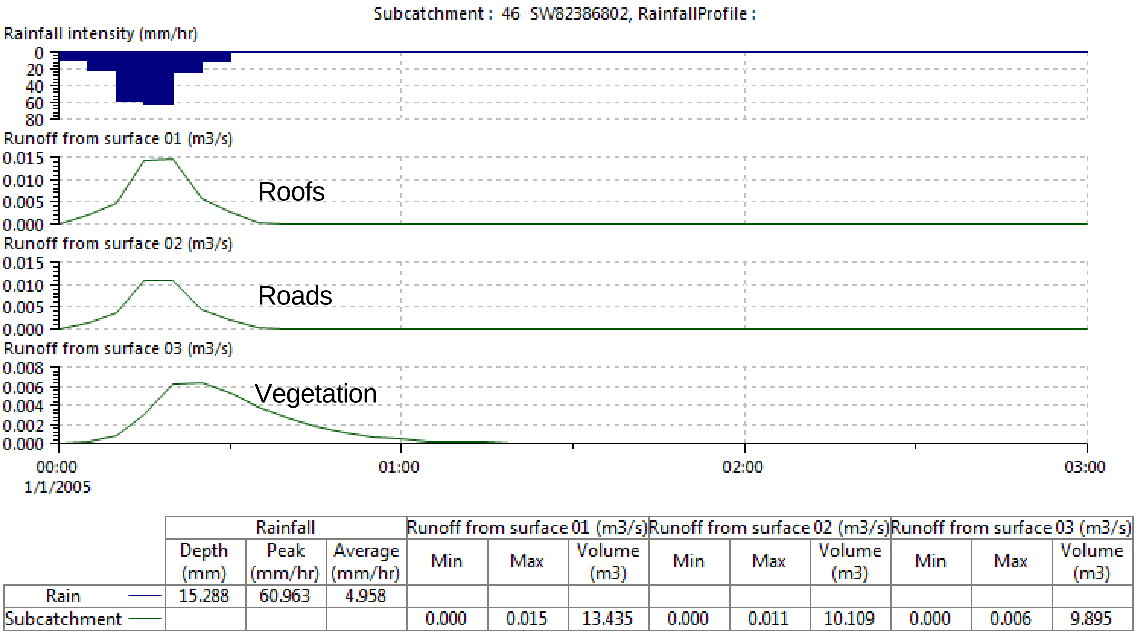


	Rainfall			Runoff from surface 01 (m3/s)			Runoff from surface 02 (m3/s)			Runoff from surface 03 (m3/s)		
	Depth (mm)	Peak (mm/hr)	Average (mm/hr)	Min	Max	Volume (m3)	Min	Max	Volume (m3)	Min	Max	Volume (m3)
Rain ——	15.288	60.963	4.958									
Subcatchment ——				0.000	0.020	18.127	0.000	0.028	24.746	0.000	0.008	12.079

Figure 6.5: Simulation hydrograph results of the actual ground truth data of subcatchment 46 on Feock area.

An additional simulation was also carried out, to investigate the runoff computations from the predicted image, as derived after applying proposed ParetoEnsemble classification model in this research study, on Feock image, in order to quantify the runoff modelling results from the classification predictions of each polygon. Feock image was comprised of 54 polygons according to the InfoWorks map. The classification of Feock image was carried out by using SLIC 50,000 which is the selected SLIC count after applying analysis of results from multiple counts in classification phase. The proposed ParetoEnsemble classification algorithm utilised 'Bagged Trees', 'Coarse KNN' and 'Fine KNN' classifiers as base algorithms which were selected based on diversity and classification accuracy comparison of 8 different test classification models as elaborated in section 4.1.2.4 of Chapter 4. The weighted ParetoEnsemble applied on Feock image provided a validation accuracy of 82.4% for Feock image compared to the maximum individual classifier’s accuracy of 81%.

Figure 6.6 shows the simulation hydrographs for the estimations of runoff, from predicted image results of subcatchment 46. In this case, the whole Feock image is divided into 54 polygons, while the result of every subcatchment is used individually in the ICM model.



	Rainfall			Runoff from surface 01 (m3/s)			Runoff from surface 02 (m3/s)			Runoff from surface 03 (m3/s)		
	Depth (mm)	Peak (mm/hr)	Average (mm/hr)	Min	Max	Volume (m3)	Min	Max	Volume (m3)	Min	Max	Volume (m3)
Rain —	15.288	60.963	4.958									
Subcatchment —				0.000	0.015	13.435	0.000	0.011	10.109	0.000	0.006	9.895

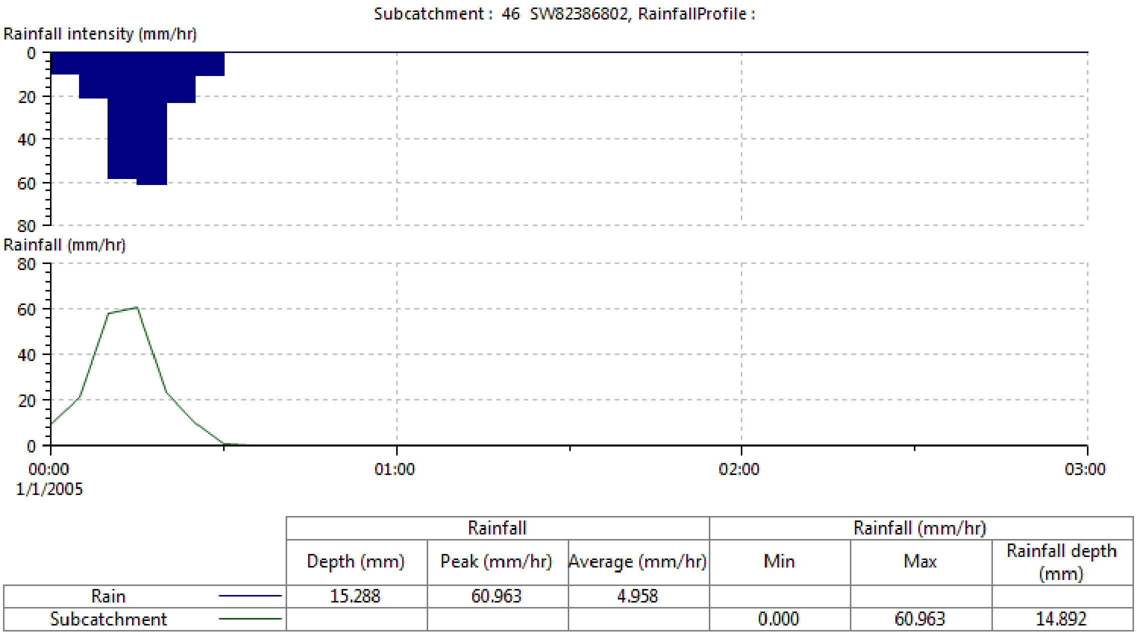
Figure 6.6: Simulation hydrograph results of the predicted classification data of subcatchment 46 on Feock area.

The hydrographs in figures 6.5 and 6.6 demonstrate that the runoff flow from vegetation is the lowest among all three, which was expected as the soil type used for this simulation is set as sand which tends to give a very low runoff flow. This catchment model is originally built in the autumn of 2001, which was an extremely wet period (circa 1 in 200-year 3-month event), in the Feock part of Cornwall. This is likely to have led to additional runoff, being generated from surrounding pervious areas (fields and/or slopes), which does not normally occur.

The total rainfall over the area of interest map (i.e., Feock) is defined by the following relationship in Equation (6.2) [192]:

$$Total\ rainfall\ volume = depth\ of\ rainfall * area\ of\ Feock \tag{6.2}$$

Where, area of Feock =36.1 ha; depth of rainfall for M5-30 event =15.3 mm, as shown in Figure 6.7.



		Rainfall			Rainfall (mm/hr)		
		Depth (mm)	Peak (mm/hr)	Average (mm/hr)	Min	Max	Rainfall depth (mm)
Rain	──	15.288	60.963	4.958			
Subcatchment	──				0.000	60.963	14.892

Figure 6.7: Rainfall volume estimations for subcatchment 46, ID SW82386802 of Feock image.

Hence, the total rainfall volume for a catchment of Feock image can be computed in Equation (6.3) as:

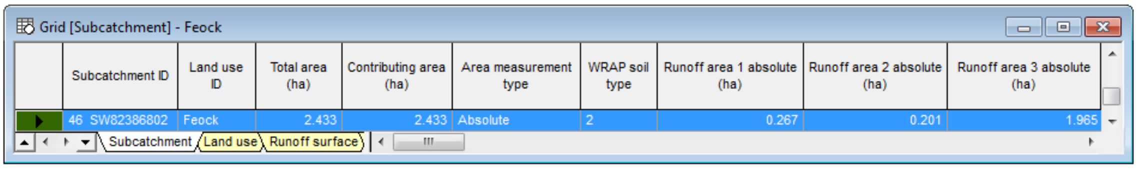
$$36.07809 * 10000 * \left(\frac{15.288}{1000}\right) = 5515.5m^3 \qquad (6.3)$$

6.5 Typical Implementation

This section includes a step-by-step manual estimation of runoff, for subcatchment 46 of Feock image. The same process will be followed for all subcatchments, one by one, to estimate the runoff for the whole Feock image. In this example, the overall runoff and individual class runoff values, for the three classes, are estimated. These estimated values, based on manual calculation, are then compared to the automated estimation values of runoff, to check the validity of this model.

Table 6.1 shows the characteristics of subcatchment 46, used in this example, where (ha) is the area in hectare for Feock image. This example considers the values of area measurement as ‘absolute’, representing the Contribution area of each surface. There are, therefore: 0.267 ha of runoff area 1, 0.201 ha of runoff area 2, and 1.965 ha of runoff area 3 (Table 6.1). One thing to be considered here is that, the Total area value represents just a label and is not linked to any calculations [203].

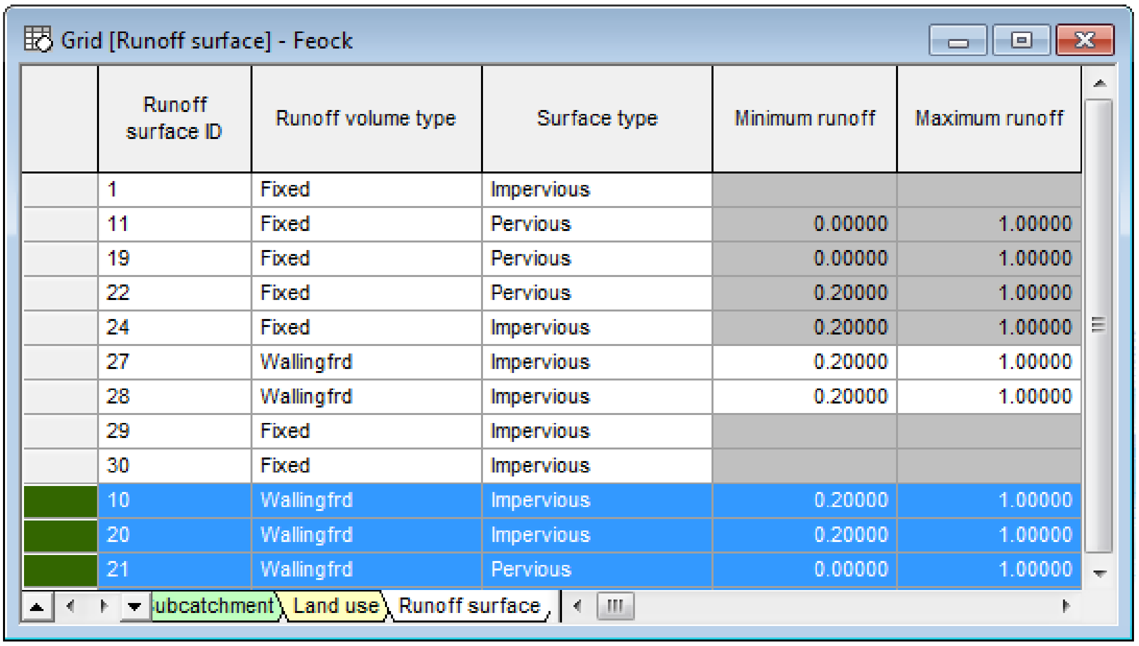
Table 6.1: Subcatchment characteristics for SLIC 50,000 ParetoEnsemble prediction of Feock.



	Subcatchment ID	Land use ID	Total area (ha)	Contributing area (ha)	Area measurement type	WRAP soil type	Runoff area 1 absolute (ha)	Runoff area 2 absolute (ha)	Runoff area 3 absolute (ha)
▶	46 SW82386802	Feock	2.433	2.433	Absolute	2	0.267	0.201	1.965

Table 6.2 represents the types of surfaces linked to Feock, where each of the three surfaces denote one of the three classes studied. Conventionally, the ICM considers that, the road, roofs, and pervious surfaces have runoff surface indices of 10, 20, and 21, respectively, as explained in section 2.7.3.1 of Chapter 2. These values of surfaces are reserved for the Wallingford PR equation, considering the range of indices, associated with permeable and impermeable surfaces. This table shows that runoff areas 1, 2 and 3 use Wallingford equation, where the minimum and maximum runoff limits for all three areas are depicted.

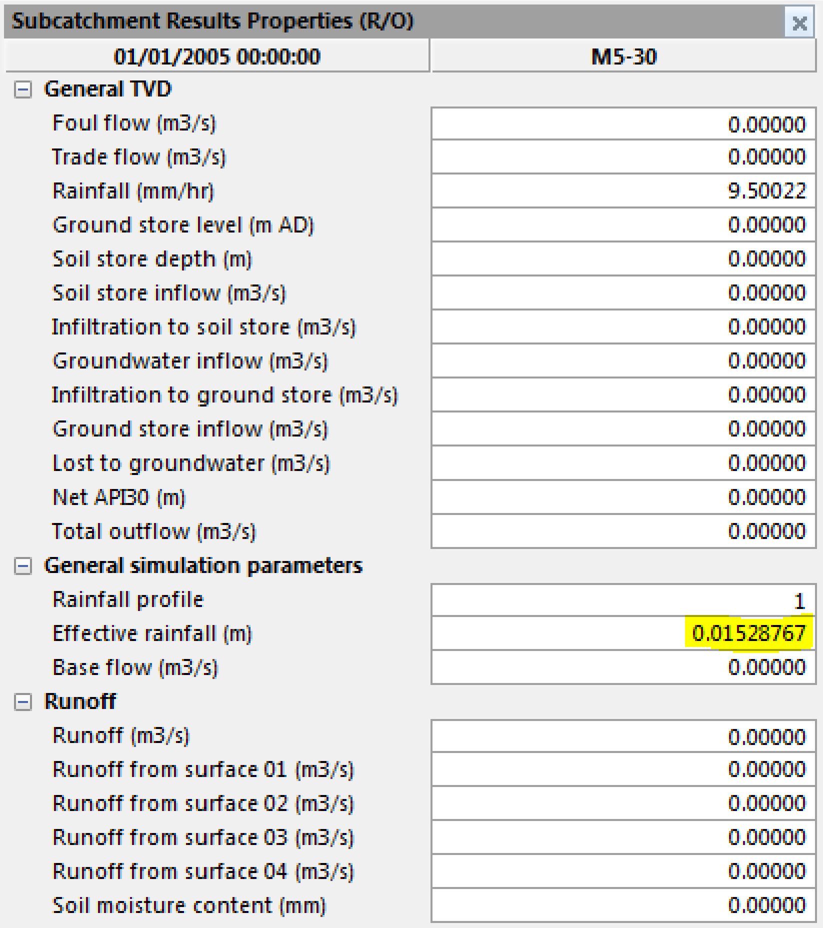
Table 6.2: Runoff surfaces details table of Feock.



	Runoff surface ID	Runoff volume type	Surface type	Minimum runoff	Maximum runoff
	1	Fixed	Impervious		
	11	Fixed	Pervious	0.00000	1.00000
	19	Fixed	Pervious	0.00000	1.00000
	22	Fixed	Pervious	0.20000	1.00000
	24	Fixed	Impervious	0.20000	1.00000
	27	Wallingfrd	Impervious	0.20000	1.00000
	28	Wallingfrd	Impervious	0.20000	1.00000
	29	Fixed	Impervious		
	30	Fixed	Impervious		
	10	Wallingfrd	Impervious	0.20000	1.00000
	20	Wallingfrd	Impervious	0.20000	1.00000
	21	Wallingfrd	Pervious	0.00000	1.00000

To apply the Wallingford PR equation, shown in Equation (2.8) and elaborated in section 2.5 of Chapter 2, for the subcatchment 46, first the rainfall volume has to be calculated which is obtained after simulation of Feock, as shown in Table 6.3, where the effective winter storm rainfall (m) event, for M5-30, is 0.01528767.

Table 6.3: The effective rainfall event of M5-30 storm duration in the subcatchment 46.



Subcatchment Results Properties (R/O)

01/01/2005 00:00:00	M5-30
⊟ **General TVD**	
Foul flow (m3/s)	0.00000
Trade flow (m3/s)	0.00000
Rainfall (mm/hr)	9.50022
Ground store level (m AD)	0.00000
Soil store depth (m)	0.00000
Soil store inflow (m3/s)	0.00000
Infiltration to soil store (m3/s)	0.00000
Groundwater inflow (m3/s)	0.00000
Infiltration to ground store (m3/s)	0.00000
Ground store inflow (m3/s)	0.00000
Lost to groundwater (m3/s)	0.00000
Net API30 (m)	0.00000
Total outflow (m3/s)	0.00000
⊟ **General simulation parameters**	
Rainfall profile	1
Effective rainfall (m)	0.01528767
Base flow (m3/s)	0.00000
⊟ **Runoff**	
Runoff (m3/s)	0.00000
Runoff from surface 01 (m3/s)	0.00000
Runoff from surface 02 (m3/s)	0.00000
Runoff from surface 03 (m3/s)	0.00000
Runoff from surface 04 (m3/s)	0.00000
Soil moisture content (mm)	0.00000

The rainfall volume for subcatchment 46 calculated in Equation (6.4) is equal to:

$$Rainfall\ volume = Effective\ rainfall * Area\ of\ subcatchment \qquad (6.4)$$

$$0.015287(m) * 2.433(ha) = 0.015287 * 2.433 * 10000 = 371.76(m^3) \qquad (6.4.1)$$

For the 1st part of Wallingford PR shown in Equation (2.8), section 2.7.3.2, Chapter 2, to be calculated the Percentage Impermeability (PIMP) is required, which is calculated by the sum of runoff area 1 and runoff area 2 as shown in Equation (6.5).

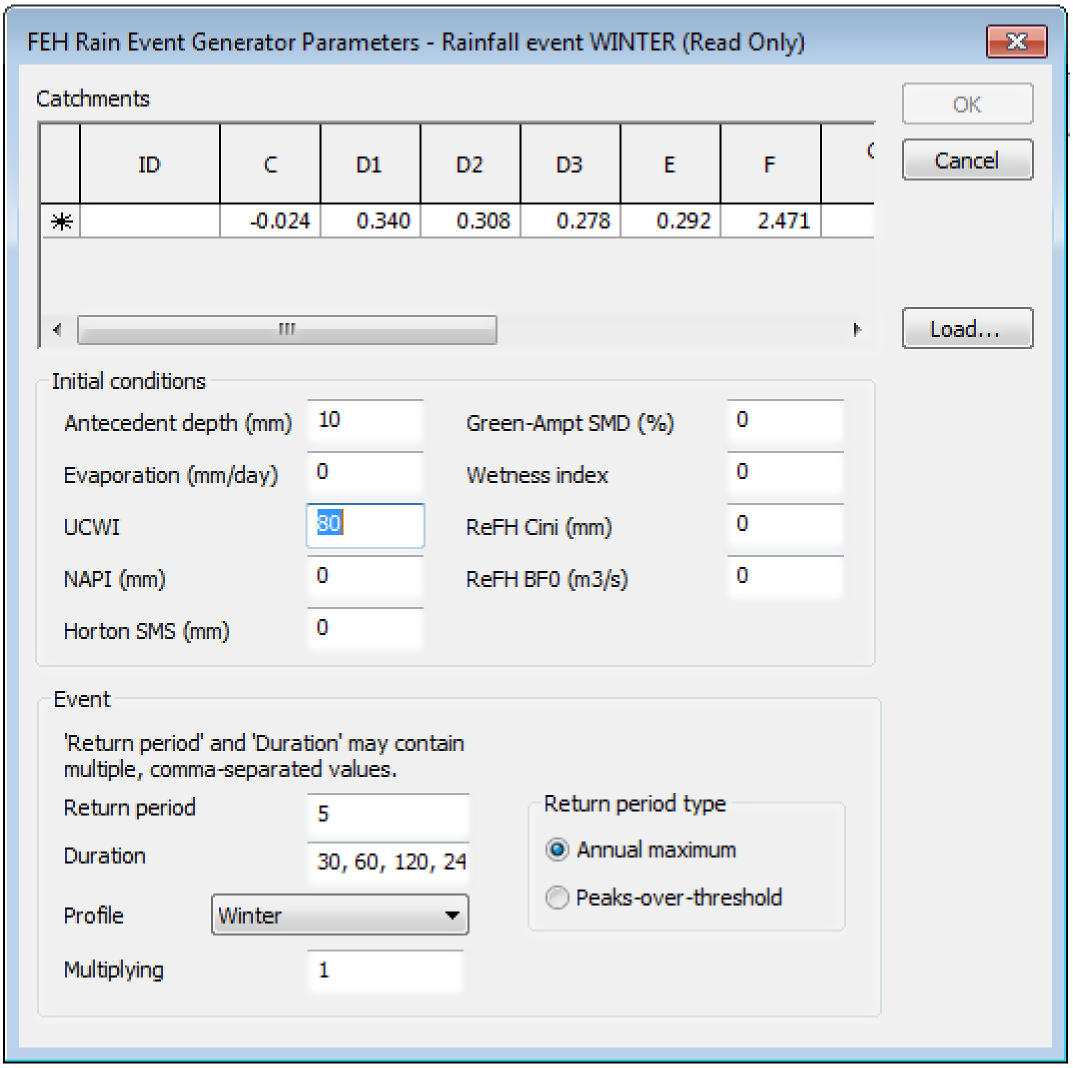
$$PIMP = (0.267 + 0.201)/2.433 = 19.24\% \qquad (6.5)$$

Regarding the calculation of the 2nd part of PR equation, Table 6.1 shows that WRAP soil type is 2, which according to Table 2.3, in Chapter 2, refers to the value 0.30.

For the 3rd part of the PR equation, Table 6.4 shows that the UCWI is equal to 80 according to the Flood Estimation Handbook (FEH) [191] which is a default value for both winter and summer rainfall events. Based on all the parameter values, PR value can be computed as follows in Equation (6.6):

$$\begin{aligned} PR &= (19.24 * 0.829) + (25 * 0.3) + (0.078 * 80) - 20.7 \\ &= 15.94996 + 7.5 + 6.24 - 20.7 = 8.98996\% \end{aligned} \tag{6.6}$$

Table 6.4: UK Rainfall (FEH) Generator input parameters.



FEH Rain Event Generator Parameters - Rainfall event WINTER (Read Only)

Catchments

	ID	C	D1	D2	D3	E	F
✳		-0.024	0.340	0.308	0.278	0.292	2.471

Initial conditions

Antecedent depth (mm)	10	Green-Ampt SMD (%)	0
Evaporation (mm/day)	0	Wetness index	0
UCWI	80	ReFH Cini (mm)	0
NAPI (mm)	0	ReFH BF0 (m3/s)	0
Horton SMS (mm)	0		

Event

'Return period' and 'Duration' may contain multiple, comma-separated values.

Return period: 5

Duration: 30, 60, 120, 24

Profile: Winter

Multiplying: 1

Return period type

● Annual maximum

○ Peaks-over-threshold

Based on Equation (2.9) in Chapter 2, using the default parameters for weighting coefficients of Table 2.3 in Chapter 2, the runoff for surfaces 1, 2 and 3 are computed, from PR equation, as follows in Equations (6.7)-(6.9):

$$\begin{aligned} &PR\ runoff\ for\ area\ 1 \\ &\quad = \Big((1 * 0.267) \\ &\quad /((1 * 0.267) + (1 * 0.201) \\ &\quad + (0.1 * 1.965))\Big) * 8.98996 = 3.623\% \end{aligned} \tag{6.7}$$

$$PR\ runoff\ for\ area\ 2 = \Big((1 * 0.201) / \big((1 * 0.267) + (1 * 0.201) + (0.1 * 1.965)\big)\Big) * 8.98996 = 2.713\% \tag{6.8}$$

$$PR\ runoff\ for\ area\ 3 = \Big((0.1 * 1.965) / \big((1 * 0.267) + (1 * 0.201) + (0.1 * 1.965)\big)\Big) * 8.98996 = 2.654\% \tag{6.9}$$

Equation (6.10) is used to calculate the runoff volume of each one of the three areas, as follows:

$$Runoff\ volume\ of\ area\ i = Rainfall\ volume * PR\ Runoff\ area\ i \tag{6.10}$$

Equations (6.10.1)-(6.10.3) show the calculated Runoff volumes for the three surface areas:

$$Runoff\ volume\ of\ area\ 1 = 371.76 * 3.623\% = 13.4m^3 \tag{6.10.1}$$

$$Runoff\ volume\ of\ area\ 2 = 371.76 * 2.713\% = 10.1m^3 \tag{6.10.2}$$

$$Runoff\ volume\ of\ area\ 3 = 371.76 * 2.654\% = 9.8m^3 \tag{6.10.3}$$

These manually calculated runoff values match with the results shown in Figure 6.8, calculated by InfoWorks ICM model. Figure 6.8 shows the runoff values predicted by the ICM for the three kinds of surfaces separately.

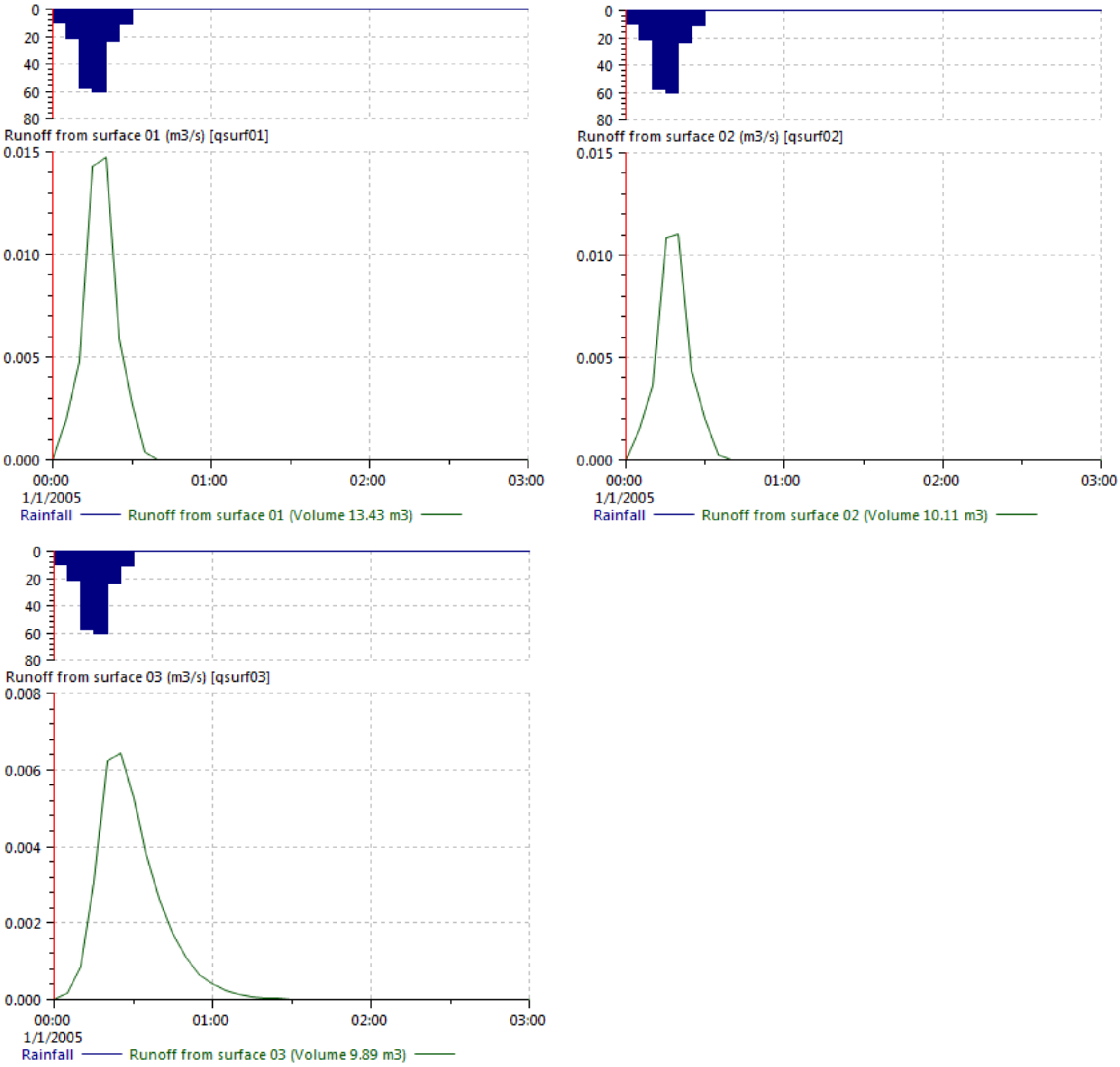


Figure 6.8: Runoff for surfaces 1 (top left), 2 (top right) and 3 (bottom) of subcatchment 46.

6.6 Runoff Estimations Using Multiple SLICs.

The validity of the estimated runoff values, as derived from InfoWorks, is assessed by comparing these, for each subcatchment in the predicted image by the ParetoEnsemble classification, to the runoff values of the corresponding subcatchment in the ground truth image. The comparison regards the relative error between ground truth and ParetoEnsemble predicted subcatchment runoff values. The Relative Error, between the ground truth and the ParetoEnsemble product, is computed by using Equation (6.11), as follows:

$$Relative\ Erro = \big((Ground\ truth\ Runoff - ParetoEnsemble\ Runoff) / Ground\ truth\ Runoff\big) * 100 \qquad (6.11)$$

Overall Error for an image is calculated by considering the Mean of Relative Error values, for all subcatchments in the image, shown in Equation (6.12), as follows:

$$\begin{aligned} &Mean\ Relative\ Error \\ &\qquad = (Relative\ Error\ sum\ of\ all\ subcatchments \\ &\qquad /total\ number\ of\ subcatchments) \end{aligned} \tag{6.12}$$

This Mean Relative Error is converted to Mean Relative Accuracy in Equation (6.13), as follows:

$$Mean\ Relative\ Accuracy = 100 - Mean\ Relative\ Error \tag{6.13}$$

Figure 6.9 illustrates different runoff estimations for Feock classification, based on different SLIC values. ParetoEnsemble classification accuracy, Best Individual classification accuracy, ParetoEnsemble runoff accuracy and Best Individual runoff accuracy for different SLIC values, are applied on Feock image validation. Overall Mean Relative runoff accuracy results, for Feock image, are computed for the design of a winter storm rainfall event, provided from simulation of the probability event 1 in 5 years Return Period (RP), 30 minutes storm duration, based on FSR [253]. In Figure 6.9, red and blue curves represent the ParetoEnsemble and Best Individual classification accuracies, respectively, whereas the value points on the curves show that the ParetoEnsemble classifier provides improved accuracy, compared to individual classification algorithms. The yellow and purple curves, in Figure 6.9, represent ParetoEnsemble runoff accuracy and Best Individual runoff accuracy, for different SLICs, indicating that increase in the SLIC's size leads to increased runoff accuracy. Also, the runoff accuracy increases with the increase in classification accuracy, which shows a linear relationship between classification accuracy and runoff accuracy. Same relationship is observed for both; ParetoEnsemble and Best Individual classification results. In addition, the classification accuracies and the runoff accuracy of the ParetoEnsemble are higher than individual classification runoff accuracy, which indicates that ParetoEnsemble classification is better than individual classification, in terms of both classification and runoff estimation performance. Since the aim of this study is to design a system for surface water modelling, in terms of estimation of runoffs, for any unknown image, there should be one selected SLIC, which can perform well, in terms of classification and runoff performance. In this regard, the illustrated graphs show that, SLICs higher than

50,000 do not lead to a reasonable increase in performance, while the computational costs increase significantly. This is the reason why SLIC 50,000 is selected to be used for any unknown images testing, in the future, while values higher than 50,000 are ignored.

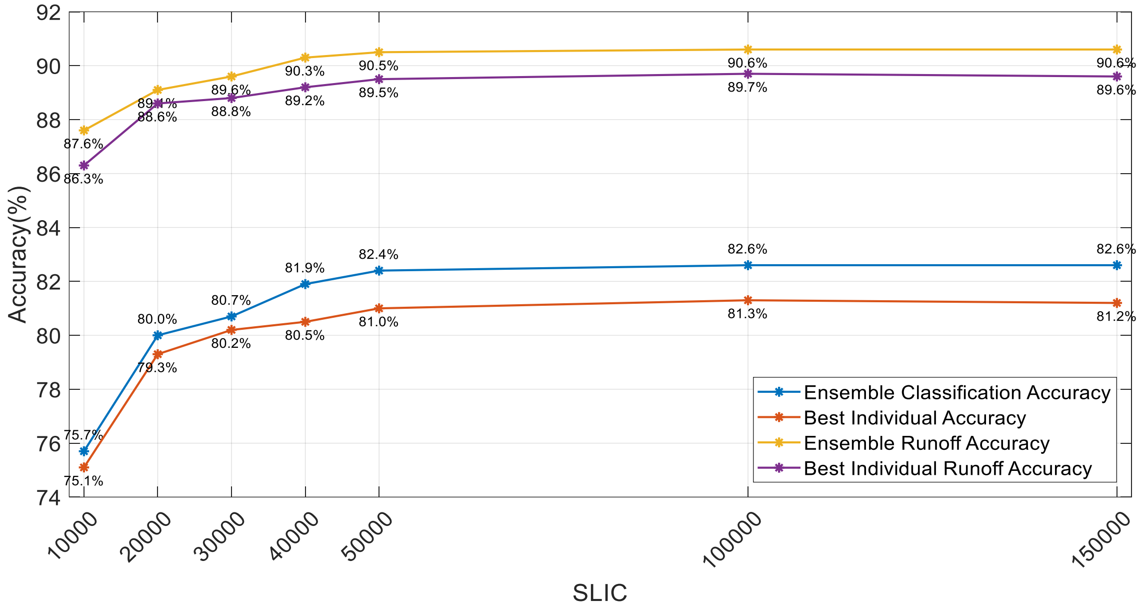


Figure 6.9: Classification and runoff accuracy comparison for different SLICs on Feock image.

Figure 6.10 shows the Percentage Relative Error values based on the comparison between runoff values of ground truth and ParetoEnsemble predicted image for all subcatchments separately, using SLIC 50,000. The low error bars for most of the subcatchments demonstrating that the estimated runoff values, from InfoWorks, for ParetoEnsemble predictions are very close to actual ground truth runoff values. Concluding, InfoWorks model is a good option for simulation and predictions of runoff estimation values.

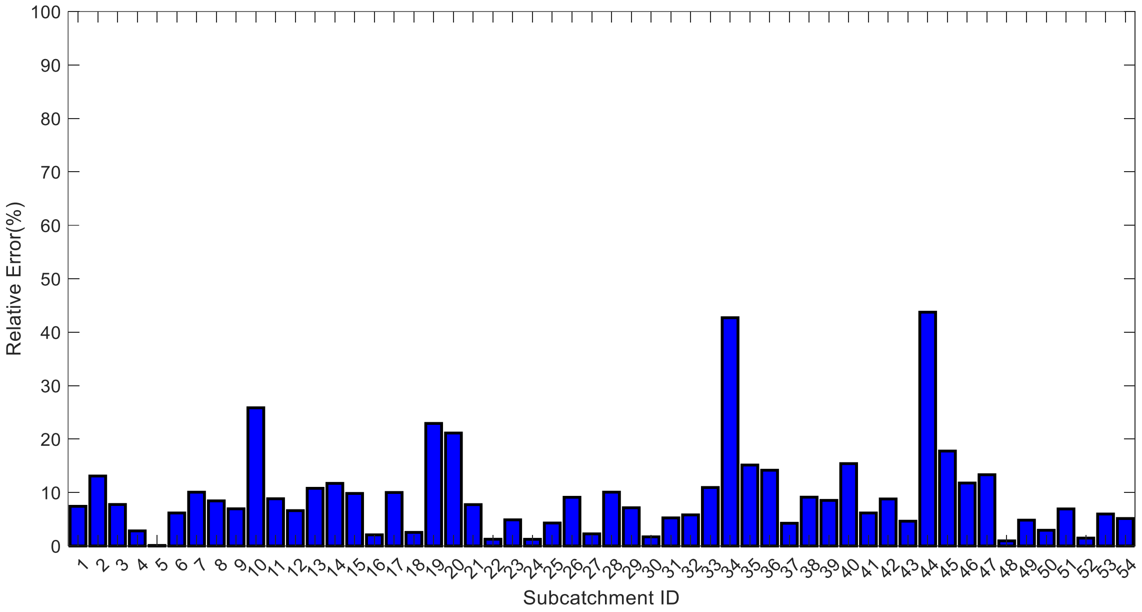


Figure 6.10: Percentage Relative Error between ground truth and ParetoEnsemble predicted runoff values, for each subcatchment in Feock image, using SLIC 50,000.

6.7 Correlation between Classification Accuracy and Runoff Accuracy

The concept of correlation coefficient is used in this research, to estimate the linear dependency, between classification accuracy and runoff accuracy, which can, in turn, provide a highly useful insight, regarding the runoff control parameters. The most commonly used type of correlation coefficient (Pearson's r coefficient) [254] is used for the calculation of the correlation coefficient, between classification and runoff accuracy for multiple SLICs. The value of correlation coefficient lies between -1 to 1, where -1 represents a negative correlation, between the variables, 0 correlation coefficient represents no correlation, between the variables and 1 correlation coefficient value represents strong positive correlation, between the pair of variables under consideration. The correlation coefficient value, between ParetoEnsemble classification accuracy and runoff accuracy, computed by using Equation (6.14), is 0.9918. The coefficient value is closer to 1, indicating high dependency between classification accuracy and runoff accuracy, which means that the changes in classification performance largely affect runoff performance, whereas the runoff accuracy can be improved by increasing the classification accuracy.

$$\rho(A,B) = \frac{1}{N-1}\sum_{i=1}^{N}\left(\overline{\frac{A_i - \mu_A}{\sigma_A}}\right)\left(\frac{B_i - \mu_B}{\sigma_B}\right) \tag{6.14}$$

The parameters μ and σ, in Equation (6.14), represent mean and standard deviation of the two variables A and B. N is the total number of samples in the variables, which should be the same in both cases, while ρ represents the estimated correlation coefficient value [254].

6.8 Runoff Estimations for Unknown Images

Following steps are carried out, for classification and runoff estimations on unknown Penelewey and Playing Place images (Figure 5.14, Chapter 5 top left and bottom left, respectively). First, contribution area percentage calculations, for the unknown images, are carried out, for SLIC 50,000 (selected based on Figure 6.9), producing the predicted images after classification and coloured polygon images. This way, the contribution area percentages sets for all the polygons, included in these images, are computed and introduced to InfoWorks, to estimate the runoff values by comparison of runoff values of each subcatchment and estimation of relative error, for these unknown images. Also, for comparison and results compilation purposes, actual ground truth images and the contribution percentage sets, for polygons of ground truth of these unknown images, are also compiled. This way, ground truth runoff of these unknown images can be computed, in order to calculate the respective overall mean relative runoff accuracy. All these contribution area percentages sets are processed in InfoWorks, in the next phase. In addition, the unknown testing results for classification and runoff accuracy are compared for the two kinds of classifications, i.e., unbalanced and balanced, to select the most suitable kind of classification, based on both classification and runoff prediction performance. Figure 6.11 shows Penelewey and Playing Place maps are extracted from Truro map, as shown on the left side and the right side of this figure, respectively, highlighted in red box as unknown case studies of this research.

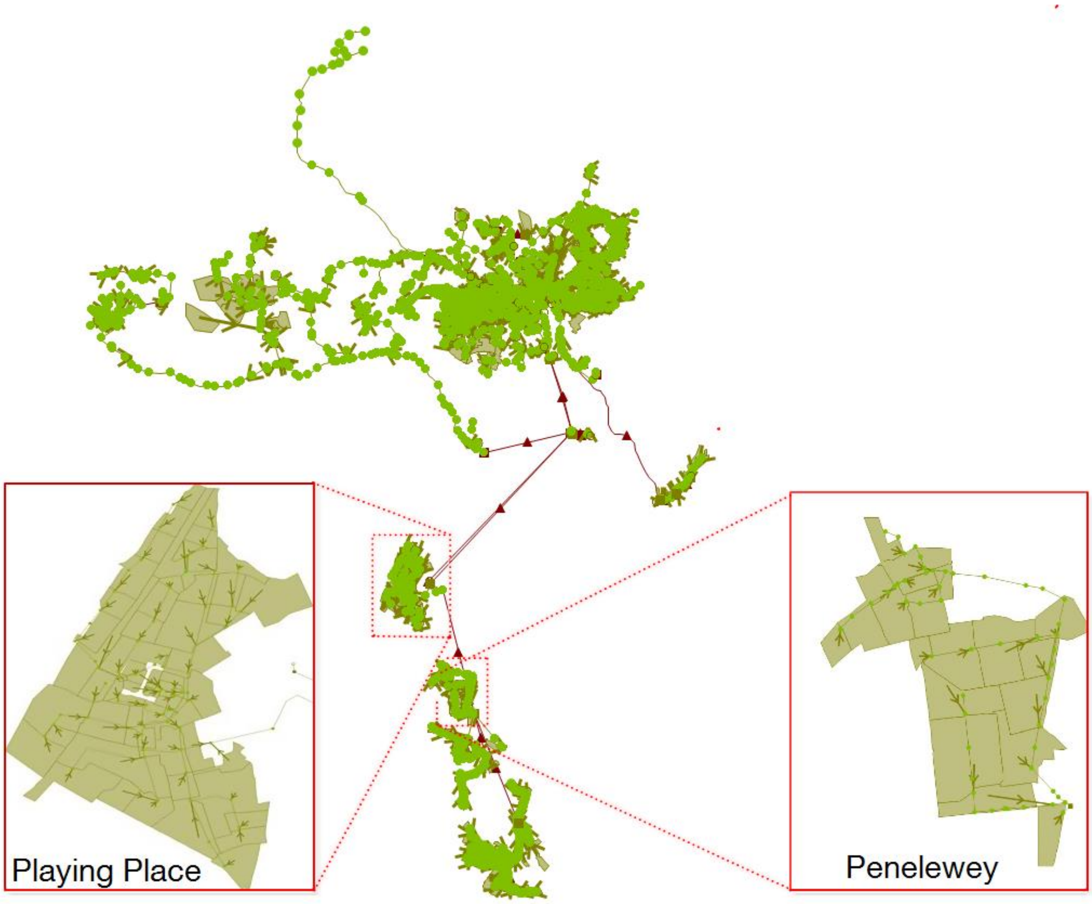


Figure 6.11: Extraction of Playing Place and Penelewey portions from Truro drainage map.

Penelewey and Playing Place images are divided, into 18 and 60 subcatchments/polygons areas, according to the ICM, via terrain analysis, respectively. Both images use different colours for each polygon, manually created to locate pixels inside the satellite image, as shown in Figure 6.12, where the right-side image shows coloured polygons overlay of Playing Place image, while the left side image illustrates coloured polygons overlay of Penelewey image.

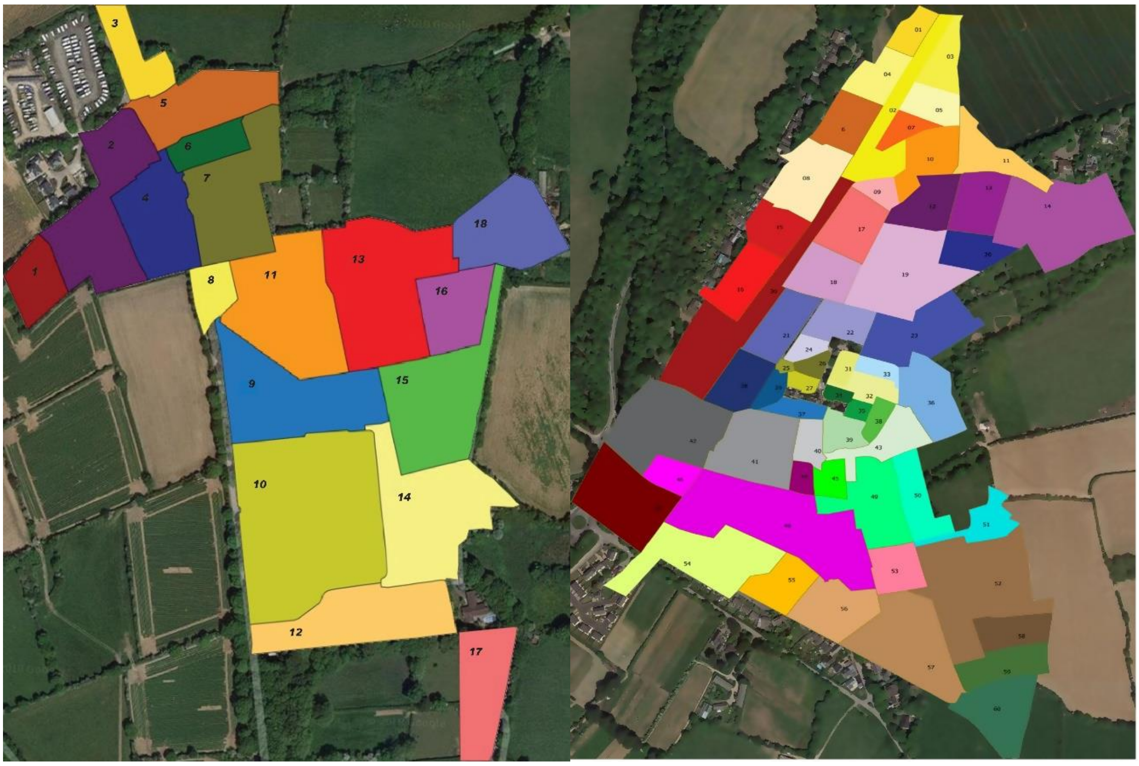


Figure 6.12: Overlay of the 18 coloured polygons, for Penelewey map, on its corresponding satellite image (left) and overlay of the 60 coloured polygons, for Playing Place map, on its corresponding satellite image (right).

Next, the classification of unknown images is performed for unbalanced and balanced cases, where unbalanced case uses the trained classifiers from unbalanced training for testing of unknown images while balanced case classification uses the trained classifiers from the training of balanced samples for unknown images testing. Figure 6.13 includes the actual ground truth image (left), predicted images, according to unbalanced and balanced classification (middle and right, respectively), for Penelewey image. Balanced classification scheme seems to produce over predictions, compared to unbalanced classification scheme, due to the balancing of class samples for training. This approach also trains the classifier to predict all three class pixels, with similar frequency, leading to a high concentration of false-positive predictions, for red and blue class pixels, in the case of balanced classification, affecting the prediction of runoff values, for balanced case modelling.

Testing of Penelewey, using unbalanced trained ParetoEnsemble classifier, for SLIC 50,000, provides 66.3% classification accuracy and 79.9% runoff accuracy, while balanced trained classifier exhibits a classification accuracy of 55.4% and runoff accuracy of 75.3%. Comparing classification and runoff accuracy, in unbalanced and balanced cases, shows that unbalanced case overall runoff accuracy and classification accuracy are higher than in the balanced case, which is why unbalanced case predictions seem to be a better choice for Penelewey image.

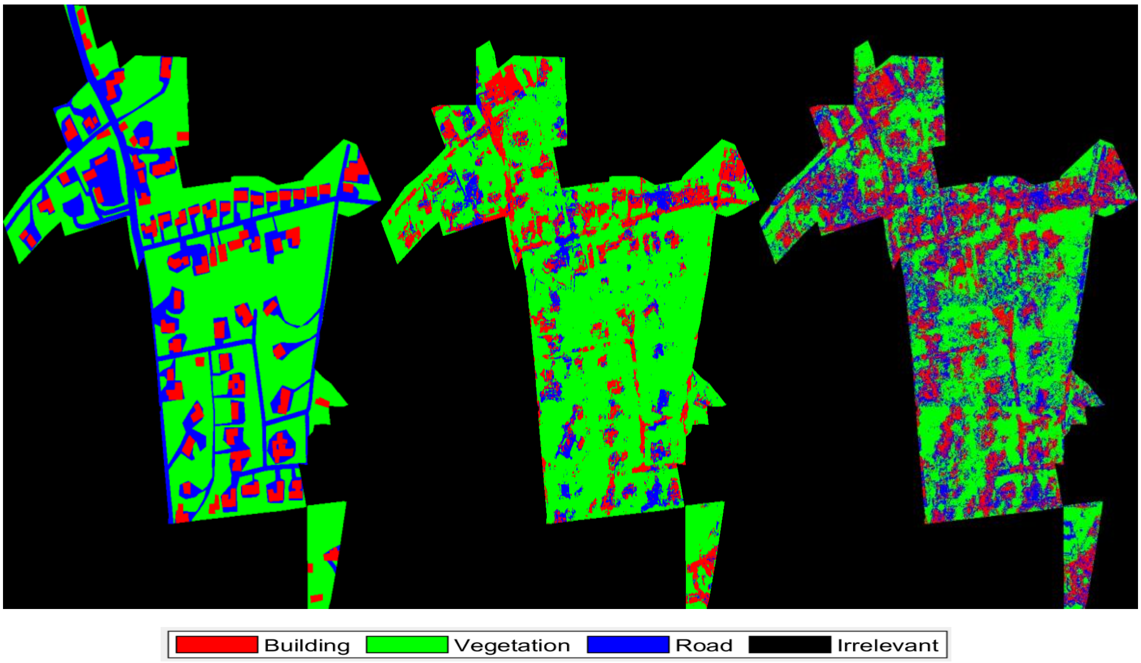


Figure 6.13: Penelewey ground truth image (left), unbalanced case predicted image (middle), and balanced case predicted image (right).

Similarly, Figure 6.14 shows actual ground truth image of Playing Place image (left side), predicted image from unbalanced classification case (middle) and predicted image from balanced classification case (right side). Playing Place image testing provides classification accuracy of 55.8% and runoff accuracy of 78.6%, in the unbalanced classification case, while a classification accuracy of 44.1% and runoff accuracy of 74.3% are achieved in the balanced classification case. The classification of Playing Place image, in the balanced case, also provides over predictions, like in the Penelewey image, while classification and runoff accuracies, in the unbalanced case, are better, compared to the balanced case, which is the reason why unbalanced case-based classification and runoff estimations are preferred in this study.

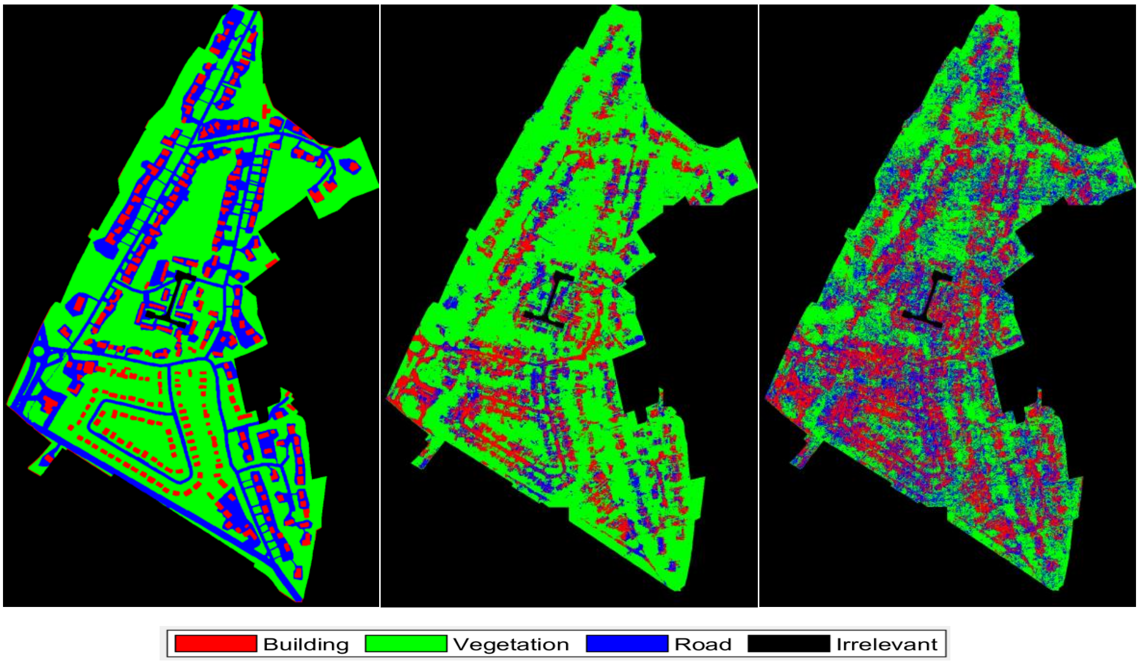


Figure 6.14: Playing Place ground truth image (left), unbalanced case predicted image (middle) and balanced case predicted image (right).

6.9 Conclusions and Suggestions

This chapter provides a detailed account about the combination of classification phase with the surface water modelling, which is one of the main aims of this research study. The process of acquiring maps from InfoWorks software and then, accessing the respective satellite images, is elaborated in this chapter. Furthermore, the classification phase, applied on unknown images, and the conversion process of classification results, into the InfoWorks ICM model input, are described in detail. The selection of optimal SLIC is done considering the following attributes: ParetoEnsemble classification accuracy and ParetoEnsemble runoff estimation accuracy where there is a high correlation between runoff accuracy and classification accuracy which is computed by calculating correlation coefficient value between the two which gives a high correlation value closer to 1. Moreover, the selection of a suitable type of classification, among balanced and unbalanced classification, is based on optimal SLIC based runoff results comparison. The results of test images Penelewey and Playing Place show that, SLIC 50,000 and unbalanced type of classification is the most suitable option. Specifically, testing of Penelewey, using unbalanced trained ParetoEnsemble classifier for SLIC 50,000 gives 66.3% classification accuracy and 79.9% runoff accuracy, while balanced trained classifier gives a classification accuracy of 55.4% and runoff accuracy of 75.3%.

In addition, Playing Place image testing gives classification accuracy of 55.8% and runoff accuracy of 78.9%, in the unbalanced case classification, and classification accuracy of 44.1% and runoff accuracy of 74.3%, in the balanced case classification.

Figure 6.15 shows performance comparison of Feock validation with Penelewey and Playing Place testing, in terms of classification and runoff accuracy, for the unbalanced case. There are two sets of bars (Figure 6.15), the first of which represents classification accuracy comparison, while the second one represents runoff accuracy comparison of the unknown case studies (i.e., Penelewey and Playing Place), with the training case study of Feock image. The difference between bar values, in both sets, depicts that there is no overfitting in the training phase, because there is not an out of the ordinary deviation between validation and test accuracies. The runoff accuracy results, for both unknown testing images, clearly show that, InfoWorks ICM modelling tool is able to model the runoff very well, even provided average quality classification results, for unknown test cases, which further concludes that the runoff estimation and modelling can be performed more accurately, by improving the system to provide better classification results.

Another observation from these results is that the runoff accuracy is always higher than the classification accuracy, for both validation and test results. This means that the relation between runoff and classification accuracy is linear, as also proven in Figure 6.9, where the runoff accuracy and classification accuracy are observed for multiple SLICSs on the same image Feock, by using ParetoEnsemble classification. The results showed that, the runoff accuracy kept on increasing with the increase in classification accuracy. Same relation is observed when the runoff accuracy and classification accuracy results are compared for multiple SLICs, by using the best of the 8 individual classifiers. Similar relation follows for test images Penelewey and Playing Place results, where the runoff accuracy is higher than the classification accuracy. The runoff accuracy values for both images is very close, but there is a difference between classification accuracy. This is due to the fact that, both images are different from one another and have got different class samples distribution, which is why it is not right to compare results from both test images directly to each other. The only fair comparison is different parameters testing on same image, as done for Feock image in Figure 6.9, which gives us interesting insights regarding the classification and runoff accuracy relationships.

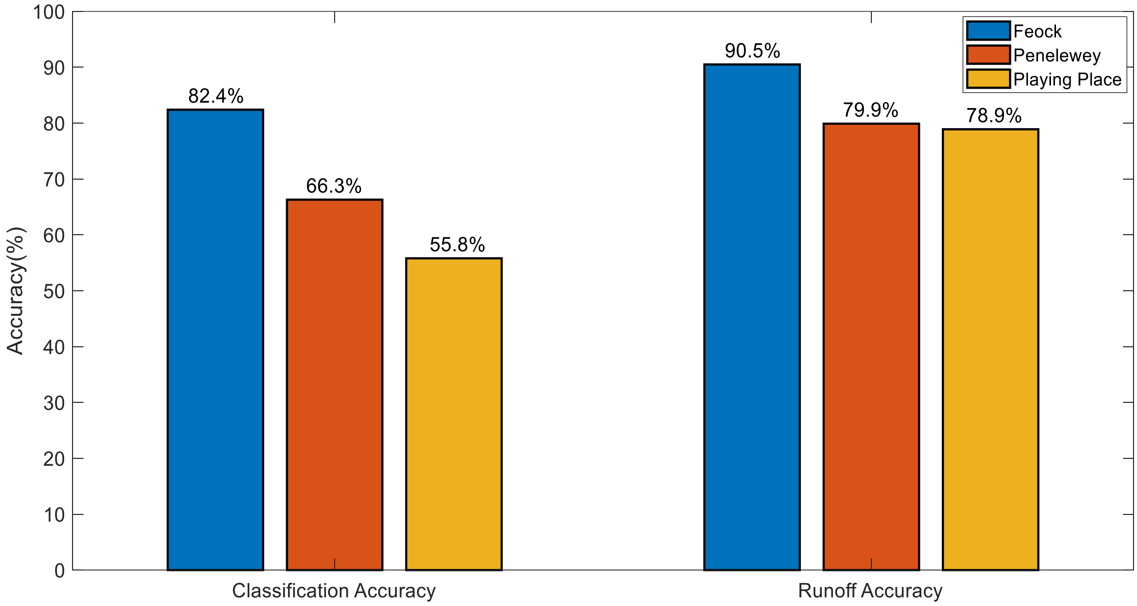


Figure 6.15: Accuracy results comparison between validation image Feock and unknown images.

CHAPTER SEVEN

7 CONCLUSIONS AND FUTURE WORK SUGGESTIONS

7.1 Conclusions

In this thesis, a system for improving urban surface water modelling for runoff estimations is proposed, based on machine learning classification techniques. The surface water runoff estimations can be further used to predict any upcoming natural disaster, in the form of flooding. Normal runoff flow estimation and then, the difference between normal runoff and increased runoff, in case of heavy rainfall, can be a good tool for prediction of an upcoming flood or abnormal conditions. This research study focuses on the design of core models, for classification and runoff predictions, which can be potentially transformed into an actual real-world application for automated flooding predictions, based on runoff estimations for any unknown area, using its satellite image views. The current study is limited to the use of any area map, which is already available in the InfoWorks ICM model. The simulation results show that this approach is a promising method, for obtaining more accurate modelling and runoff estimations of surface water systems, applying a partially automated methodology, reducing the requirement for engineers to manually perform the runoff estimations. A new Pareto, diversity and weighted sum based (ParetoEnsemble) classifier, giving improved classification results, compared to traditional ensemble classifiers, is highlighted as one of the novelties of this work. Another novelty of this study involves the design a fully automated system, for linking classification and runoff estimations, using InfoWorks ICM modelling environment, something that is not described in any prior art literature. The promising classification and runoff prediction results of this research work support the validity of the presented novelties. Conclusions, based on the given material and the objectives achieved are listed as follows:

1. In the first classification scenario, the CT and RF supervised pixel-based classification techniques are applied. CT performance was compared to different RF configurations (i.e., RF10, RF20, RF50 and RF100). Based on the overall respective results, of normal and parallel processing modes, the point RF100 demonstrates better results. However, above a tree count of 20, the performance improvement appears to be slow. On the other

hand, regarding mean computational time, parallel processing, especially for Feock image, performs much faster than the normal processing. Furthermore, after selection of RF model, as the suitable classifier, finding the lowest number of trees that satisfy precision requirements, is an important factor to the final execution time of the algorithm. In terms of precision, more trees and smaller image prove to be the best combination, while in terms of speed, it is less trees and smaller image that perform better. The generalisation of the trained classification models is explored by testing unknown data image with trained classification model, using a different training data image. The results might not be exact, but they produce reasonable approximations, in the calculation of permeable and impermeable areas which highlight the strength of pixel-based classification models in classification of land use data images.

2. In the second scenario of classification, SLIC, a super-pixelling method, that can divide an image into small homogeneous patches, is applied to the satellite images, allowing different size objects and multiple features to be extracted. Next, the objects are categorised, using eight different selected classifiers, to segment the satellite images, instead of processing them, pixel by pixel. The comparison of the pixel- to the object-based approach for multiple SLIC values has shown that the object-based approach, when combined with 10,000 objects using SLIC segmentation, is superior to the pixel-based approach. Specifically, the object-based approach exhibits a much higher degree of accuracy (93.7% versus 88.5%) and a shorter total runtime (869 sec versus 10,855 sec). Pareto dominance analysis further proves this conclusion, since the Knee point for all the test images belongs to the object-based approach. Next, a modified ParetoEnsemble classifier is designed, by selecting a few top performing classifiers (from among the aforementioned eight individual classifiers), based on the highest estimated diversity among them. ParetoEnsemble classifier results were compared to the individual classifiers, used in this section, showing that the ParetoEnsemble classifier exhibits higher total mean accuracy (94.5%), compared to the individual classifiers (93.7%), when the same dataset is employed (the six selected images). The performance evaluation for different SLICs, on

Feock image, show that, values higher than SLIC 50,000 do not provide considerable improvement on the overall classification accuracy, so SLIC 50,000 is selected as optimal SLIC, among all tested SLICs. Further analysis of results proves that, unbalanced data-based classification provides higher accuracy (82.4%), in the case of Feock image, compared to balanced data-based classification (81.9%).

3. In the third classification scenario, a CNN, with an encoder-decoder architecture, is employed, based on SegNet deep learning. Since the dataset (six selected images) is quite limited for neural network training, several augmentation techniques are employed, to make the dataset larger, by increasing the data diversity. The vegetation class appears to be dominant in some images of dataset, while less frequent in other images, raising the issue of unbalanced classes in the training phase. To reduce the classification error, due to unbalanced class samples, weights are assigned to each class, where the less frequent class is assigned weight value higher than other classes, to prevent the network from classifying every pixel to the most frequent class. The total accuracy, achieved in this SegNet network (92.6%), is comparable to the results obtained in the 2nd scenario (94.5%, using the ParetoEnsemble classification), for the same dataset (the six selected images). Many configurations of parameter settings were compared, for the six selected images, while the best-performing set (4 layers with a filter size of 64) was used for the testing of the Feock image, giving a classification accuracy of 78.3%, which is lower than other classification models. The reason for this lower accuracy is that, Feock is a large image and thus requires data from a variety of scenarios, to get a well-trained model. However, the results suggest that this approach, provided suitable parameters tuning and more data, can outperform other state-of-the-art methods.

4. The best performing classification models, among all three scenarios of classification, are compared, performance-wise, for Feock image, in terms of classification validation accuracy. The results showed that, the ParetoEnsemble classifier, with SLIC 50,000, provides best accuracy, among all three tested classification scenarios. Therefore, it is selected to be used as default classification model, during the modelling phase of this

thesis, for the testing of unknown images, after training with Feock image. The generalisation performance of the ParetoEnsemble classifier was assessed, by testing two unknown satellite images, Penelewey and Playing Place, which, considering the training conditions and the limitations of this research scheme, provided fair results (66.3% and 55.8% for Penelewey and Playing Place, respectively).

5. The final objective was to link the classification phase of this research to InfoWorks modelling phase, where an automated stormwater modelling system was compiled, in this research, to predict runoff parameter, through modelling of various parameters. The modelling results can be further used for many real-world applications, such as early flood prediction and safety measures. According to the InfoWorks ICM modelling results, for runoff estimations in Feock image, the mean relative runoff accuracy (computed by comparing runoff predictions for ground truth and ParetoEnsemble runoff for all subcatchments separately) increases as SLIC value rises, until it reaches 50,000. This SLIC value is selected as an optimal SLIC- in terms of both classification and runoff accuracy- for future unknown data testing and predictions. The analysis of the correlation coefficient value (i.e., 0.9918), between ParetoEnsemble classification accuracy and runoff accuracy, depicts that there is a strong dependence of runoff performance on classification performance. The correlation coefficient value is computed for multiple SLIC classification and runoff accuracy results, while a high correlation coefficient value confirms the high dependence of runoff accuracy on classification accuracy.

6. The classification results for non-vegetation area seem to be better, in case of balanced classification in Feock, compared to unbalanced classification, because of the ability of the balanced classifier to predict all classes without any bias. However, the overall classification of unknown images, in case of balanced data-based training, gives an overall classification accuracy of 55.4% and 44.1%, for the unknown images Penelewey and Playing Place, respectively, which is lower than the overall testing accuracy of unknown images, for unbalanced data training-based classification. The reduced accuracy of unknown images is due to

numerous false positive predictions, for less frequent class samples, in the case of balanced training and predictions. Furthermore, the runoff results, for unknown images Penelewey and Playing Place, being 79.9% and 78.6% for unbalanced classification, while 75.3% and 74.3% for balanced classification, respectively, prove that unbalanced data-based training is better, in terms of both unknown classification and runoff prediction. The runoff predictions from the better performing unbalanced training case are reasonable, considering the classification limitations and challenges**.**

7.2 Suggestions for Future research

Based on the presented findings, this study can further proceed into researching issues, such as:

1. The issue of manual ground truth data generation- quite a time consuming task- is an important aspect of this study and a barrier in creating a good amount of training data, for machine learning classifiers. Therefore, it is suggested to use semi-automatic methods of creating ground truth image data, in the future. One suggestion is, to first apply an unsupervised classification model, such as k-means clustering, on the target image, which can produce a suitable number of clusters, equal to the number of classes, while the misclassified pixels can be corrected manually, leading to an accurate ground truth image, with less manual effort. Moreover, this approach could fix the issue of accurate ground truth creation, in image dataset, where some of the roads/buildings are totally/partially covered by trees/shadows, by manually correcting the labels on misclassified pixels.

2. Another important factor to be considered and improved upon, is the selection of feature attributes that can perform even better for the classification, in such type of research problems. In this study, multiple colour, shape and texture-based attributes, in three different colour spaces (RGB, HSV and Lab), have been used. Nonetheless, there is still much to explore and improve, by testing other kind of features, in the future, where the selection is closely related to the performance of the classification model under study. Some other possibilities include more texture features, like Gabor, and Local Ternary Pattern (LTP), along with other non-texture features, such as Speeded Up Robust Features (SURF), etc.

3. Although many types of classification models are explored, in this research, there are still other kinds of algorithms, to be tested, such as hierarchical clustering and other unsupervised models, in order to create a classification model that can learn without any training data. Furthermore, in the presented work, only one image was used, during the training of classifiers. This leaves an issue to be further explored, regarding possible improvement, by adding more than one images having more versatility in terms of capturing environment and quality, during the training phase, producing even better trained classification models. Moreover, better classification results can be derived, if the vegetation class is divided into further subclasses (i.e., fields, trees, grass and soil), in order to produce more balanced classes.

4. Wallingford model has been used, in this study, considering the non-zero vegetation contribution area, during modelling. This assumption creates a prospect to be further explored, regarding the use of some other model, while not considering the contributions from vegetation areas, exploring the impacts of vegetation, which in some cases can be sand (i.e., class-type 1 and 2 in Table 2.3, Chapter 2), behaving as a pervious area, on runoff predictions. This occurrence is possible in many urban situations, especially during summer storms, when the runoff from grassed areas (verges/gardens etc.) tends to be minimal.

5. Another prospect is, calibrate the InfoWorks model by doing a flow survey for Feock catchment, to estimate actual rainfall, usually consisting of a couple of rain-gauges, with at least one flow monitor, in the network which could cost (according to Pell Frischmann Company [209]) about £7000 for 5 weeks check. Then, the flow monitoring is to be used to measure some real rainfall, testing the performance of the proposed classification approach for calibrated model, compared to the original one.

REFERENCES

[1] Chouhan, S. S., & Khatri, R. (2016). Data Mining based Technique for Natural Event Prediction and Disaster Management. *International Journal of Computer Applications*, *139*(14).

[2] Dawod, G. M., Mirza, M. N., & Al-Ghamdi, K. A. (2011). GIS-based spatial mapping of flash flood hazard in Makkah City, Saudi Arabia. *Journal of Geographic Information System*, *3*(3), 217.

[3] Defra. (2012). National policy statement for waste water: A framework document for planning decisions on nationally significant wastewater infrastructure. London: The Stationery Office. Department of Environment, Food and Rural Affairs. ISBN 9780108511080. URL https://www.gov.uk/government/publications/national-policy-s tatement-for-waste-water.

[4] Astaraie-Imani, M., Kapelan, Z., & Butler, D. (2013). Improving the performance of an integrated urban wastewater system under future climate change and urbanisation scenarios. *Journal of Water and Climate Change*, *4*(3), 232-243.

[5] Younis, M. C., Keedwell, E., Savic, D., & Raine, A. (2017). Evaluating Classification Algorithms for Improved Wastewater System Calibration. *CCWI conference publication*, 15th International Computing & Control for the Water Industry Conference. Proceedings from the conference held 5th – 7th September 2017 in Sheffield University, Sheffield, England.

[6] Mark, O., Hénonin, J., Domingo, N. D. S., Russo, B., Chen, A. S., & Djordjević, S. (2014). Report and papers with guidelines on calibration of urban flood models.

[7] Jacobson, C. R. (2011). Identification and quantification of the hydrological impacts of imperviousness in urban catchments: A review. *Journal of environmental management*, *92*(6), 1438-1448.

[8] Lippitt, C. D., Stow, D. A., & Coulter, L. L. (Eds.). (2015). *Time-sensitive remote sensing*. Springer.

[9] Thapa, R. B., & Murayama, Y. (2009). Urban mapping, accuracy, & image classification: A comparison of multiple approaches in Tsukuba City, Japan. *Applied geography*, *29*(1), 135-144.

[10] Lu, D., Hetrick, S., & Moran, E. (2010). Land cover classification in a complex urban-rural landscape with QuickBird imagery. *Photogrammetric Engineering & Remote Sensing*, *76*(10), 1159-1168.

[11] Myint, S. W., Gober, P., Brazel, A., Grossman-Clarke, S., & Weng, Q. (2011). Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery. *Remote sensing of environment*, *115*(5), 1145-1161.

[12] Myint, S. W., Galletti, C. S., Kaplan, S., & Kim, W. K. (2013). Object vs. pixel: a systematic evaluation in urban environments. *Geocarto International*, *28*(7), 657-678.

[13] Fonji, S. F., & Taff, G. N. (2014). Using satellite data to monitor land-use land-cover change in North-eastern Latvia. *Springerplus*, *3*(1), 61.

[14] Ghosh, J., & Porchelvan, P. (2017). Remote sensing and GIS technique enable to assess and predict landuse changes in Vellore district, Tamil Nadu, India. *IJAER*, *12*(12), 3474-3482.

[15] Sujatha, N. (2015). Evolutionary Approach for Land cover classification using GA based Fuzzy Clustering Techniques.

[16] Miljković, O. (2009). Image pre-processing tool. *Kragujevac Journal of Mathematics*, *32*(32), 97-107.

[17] L. Kooistra. "RS Pre-processing. Lecture on Pre-processing in remote sensing (ppt). Introduction Geo-Information (GRS-10306)." 2016.

[18] Liping, C., Yujun, S., & Saeed, S. (2018). Monitoring and predicting land use and land cover changes using remote sensing and GIS techniques—A case study of a hilly area, Jiangle, China. *PloS one*, *13*(7), e0200493.

[19] Chormanski, J., Van de Voorde, T., De Roeck, T., Batelaan, O., & Canters, F. (2008). Improving distributed runoff prediction in urbanized catchments with remote sensing based estimates of impervious surface cover. *Sensors*, *8*(2), 910-932.

[20] Paolini, L., Grings, F., Sobrino, J. A., Jiménez Muñoz, J. C., & Karszenbaum, H. (2006). Radiometric correction effects in Landsat multi-date/multi-sensor change detection studies. *International Journal of Remote Sensing*, *27*(4), 685-704.

[21] Ballhorn, U. (2017). Pre-Processing of Remote Sensing Data. Asia Link Project FORRSA. Bogor Agricultural University (IPB), Indonesia. 2007. URL http://www.helsinki.fi/vitri/research/Educational_Projects/forrsa/GIS_AL2_course%20proceedings/cd/Course/lecture%20pre%20processing%20Bogor%202007.pdf.

[22] Baatz, M., Hoffmann, C., & Willhauck, G. (2008). Progressing from object-based to object-oriented image analysis. In *Object-Based Image Analysis* (pp. 29-42). Springer, Berlin, Heidelberg.

[23] Mendoza, F., & Lu, R. (2015). Basics of image analysis. In *Hyperspectral Imaging Technology in Food and Agriculture* (pp. 9-56). Springer, New York, NY.

[24] Machajdik, J., & Hanbury, A. (2010, October). Affective image classification using features inspired by psychology and art theory. In *Proceedings of the 18th ACM international conference on Multimedia* (pp. 83-92). ACM.

[25] Solomon, C., & Breckon, T. (2011). *Fundamentals of Digital Image Processing: A practical approach with examples in MatLab*. John Wiley & Sons.

[26] Pal, N. R., & Pal, S. K. (1993). A review on image segmentation techniques. *Pattern recognition*, *26*(9), 1277-1294.

[27] Choi, C., Jennings, A., & Hulskamp, J. (1996). Learning To Segment using Fuzzy Boundary Cell Features. *Proc. of Complex Systems*.

[28] Despotović, I., Goossens, B., & Philips, W. (2015). MRI segmentation of the human brain: challenges, methods, and applications. *Computational and mathematical methods in medicine*, *2015*.

[29] Al-Barhawee, D. H. (2007). *Vectorization of Remote Sensing Data for GIS Use*. PhD thesis, Department of Computer Sciences, College of Computer

Sciences and Mathematics. University of Mosul. Iraq. (Unpublished doctoral dissertation).

[30] Fu, K. S., & Mui, J. K. (1981). A survey on image segmentation. *Pattern recognition*, *13*(1), 3-16.

[31] Haralick, R. M., & Shapiro, L. G. (1985). Image segmentation techniques. *Computer vision, graphics, and image processing*, *29*(1), pp.100-132.

[32] Fan, J., Yau, D. K., Elmagarmid, A. K., & Aref, W. G. (2001). Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE transactions on image processing*, *10*(10), 1454-1466.

[33] Bhalerao, A., & Wilson, R. (2001). Unsupervised image segmentation combining region and boundary estimation. *Image and Vision Computing*, *19*(6), 353-368.

[34] Sherman, A. B., Koss, L. G., Abbott, M. C., & Liao, M. W. (1981). A method of boundary determination in digital images of urothelial cells. *Pattern Recognition*, *13*(4), 285-291.

[35] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Susstrunk, S. (2010). *Slic superpixels. EPFL, Lausssanne*. Switzerland, Technical Report. 149300.

[36] Yokoya, N., & Levine, M. D. (1989). Range image segmentation based on differential geometry: A hybrid approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*(6), 643-649.

[37] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, *34*(11), 2274-2282.

[38] Bhargavi, K., & Jyothi, S. (2014). A survey on threshold based segmentation technique in image processing. *International Journal of Innovative Research and Development*, *3*(12), 234-239.

[39] Abubakar, F. M. (2013). Study of image segmentation using thresholding technique on a noisy image. *International journal of science and research (IJSR)*, *2*, 49.

[40] Upadhyay, P. R., & Kumar, M. S. (2017). Survey on Various Image Segmentation Techniques, *2*(1).

[41] Long, Z. (2001). *The Design and Implementation of an Image Segmentation System for Forest Image Analysis* (Doctoral dissertation, Mississippi State University).

[42] Dulyakarn, P., Rangsanseri, Y., & Thitimajshima, P. (1999, November). Histogram transformation based threshold selection for image segmentation. In *Proc. Asian Conference on Remote Sensing', Hong Kong, China*.

[43] Sarabi, A., & Aggarwal, J. K. (1981). Segmentation of chromatic images. *Pattern recognition*, *13*(6), 417-427.

[44] Lauterbach, B., & Anheier, W. (1994). Segmentation of Scanned Maps in Uniform Color Spaces. *MVA*, *94*, 322-325.

[45] Zaniewski, K. (2001). Multispectral classification algorithms and their application to thin section imagery. Prairie Perspectives.

[46] Hosseini, A., & Ghassemian, H. (2012, May). Classification of hyperspectral and multispectral images by using fractal dimension of spectral response curve. In *20th Iranian Conference on Electrical Engineering (ICEE2012)* (pp. 1452-1457). IEEE.

[47] Verma, N., & Sharma, D. (2013). Region Merging Based Image Segmentation Using Maximal Similarity Mechanism. *International Journal of Engineering Research and Development e-ISSN*: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com, 7(7), PP. 09-13

[48] Carvalho, E. A., Ushizima, D. M., Medeiros, F. N., Martins, C. I. O., Marques, R. C., & Oliveira, I. N. (2010). SAR imagery segmentation by statistical region growing and hierarchical merging. *Digital Signal Processing*, *20*(5), 1365-1378.

[49] Umbaugh, S. E. (2010). *Digital image processing and analysis: human and computer vision applications with CVIPtools*. CRC press.

[50] Lew, M. S. (Ed.). (2013). *Principles of visual information retrieval*. Springer Science & Business Media.

[51] Malpica, N., Ortuño, J. E., & Santos, A. (2003). A multichannel watershed-based algorithm for supervised texture segmentation. *Pattern Recognition Letters*, *24*(9-10), 1545-1554.

[52] Wang, Y. H. (2010). Tutorial: Image Segmentation. *National Taiwan University, Taipei*, 1-36.

[53] Kaur, D., & Kaur, Y. (2014). Various image segmentation techniques: a review. *International Journal of Computer Science and Mobile Computing*, *3*(5), 809-814.

[54] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, *8*(6), 679-698.

[55] Baugher, E. S., & Rosenfeld, A. (1986). Boundary localization in an image pyramid. *Pattern Recognition*, *19*(5), 373-395.

[56] Sun, W., Liao, Q., Xue, J. H., & Zhou, F. (2018). SPSIM: A superpixel-based similarity index for full-reference image quality assessment. *IEEE Transactions on Image Processing*, *27*(9), 4232-4244.

[57] Plot Land Classification with Color Features and Superpixels- MATLAB & Simulink- MathWorks United Kingdom. (R2018b). [Online] Available: https://uk.mathworks.com/help/images/land-classification-with-color-features-and-superpixels.html.

[58] Gigandet, X., Bach Cuadra, M., & Thiran, J. (2004). *Satellite image segmentation and classification* (No. STUDENT). URL http://ltswww.epfl.ch/~bach/Gigande t2005.pdf.

[59] Viet Tran, L. (2003). *Efficient image retrieval with statistical color descriptors* (Doctoral dissertation, Linköping University Electronic Press).

[60] Tou, J. T., & Gonzalez, R. C. (1974). Pattern Recognition Principles Addison-Wesley. *Reading, MA*, *377*.

[61] Traina, C., Figueiredo, J. M., & Traina, A. J. (2005). Image domain formalization for content-based image retrieval. In *Proceedings of the 2005 ACM symposium on applied computing,* pp. 604-609. ACM.

[62] Kavya, R. (2015). Feature Extraction Technique for Robust and Fast Visual Tracking: A Typical Review. *Int. J. Emerg. Eng. Res. Technol.*, *3*(1), 98-104.

[63] García, M. A., & Puig, D. (2002, August). Improving texture pattern recognition by integration of multiple texture feature extraction methods. In *Object recognition supported by user interaction for service robots* (Vol. 3, pp. 7-10). IEEE.

[64] Battiato, S., Gallo, G., & Nicotra, S. (2003, September). Perceptive visual texture classification and retrieval. In *ICIAP* (Vol. 3, pp. 524-529).

[65] Lepistö, L., Kunttu, I., Autio, J., & Visa, A. (2003). Rock image classification using non-homogenous textures and spectral imaging.

[66] Liu, X., & Wang, D. (2003). Texture classification using spectral histograms. *IEEE transactions on image processing*, *12*(6), 661-670.

[67] Liu, Y., Wu, S., & Zhou, X. (2003, June). Texture segmentation based on features in wavelet domain for image retrieval. In *Visual Communications and Image Processing 2003* (Vol. 5150, pp. 2026-2034). International Society for Optics and Photonics.

[68] Zhang, D., Wong, A., Indrawan, M., & Lu, G. (2000). Content-based image retrieval using Gabor texture features. *IEEE Transactions Pami*, *13*.

[69] Alamdar, F., & Keyvanpour, M. (2011). A new color feature extraction method based on QuadHistogram. *Procedia Environmental Sciences*, *10*, 777-783.

[70] Kodituwakku, S. R., & Selvarajah, S. (2004). Comparison of color features for image retrieval. *Indian Journal of Computer Science and Engineering*, *1*(3), 207-211.

[71] Petrakis, E. G. (2002). Fast retrieval by spatial structure in image databases. *Journal of Visual Languages & Computing*, *13*(5), 545-569.

[72] Van Den Broek, E. L., Kisters, P. M., & Vuurpijl, L. G. (2005). Content-based image retrieval benchmarking: Utilizing color categories and color distributions. *Journal of imaging science and technology*, *49*(3), 293-301.

[73] Duan, F. (2018). A Feature Extraction Method Combining Color-Shape for Binocular Stereo Vision Image. *JMPT*, *9*(2), 45-58.

[74] Doiphode, A., Kulkarni, A., & Sunil, Y. (2017). A review on feature extraction techniques using color, texture and shape features. *International Conference On Emanations in Modern Technology and Engineering (ICEMTE-2017), 5*(3):31–34.

[75] Zhang, X. N., Jiang, J., Liang, Z. H., & Liu, C. L. (2010). Skin color enhancement based on favorite skin color in HSV color space. *IEEE Transactions on Consumer Electronics*, *56*(3), 1789-1793.

[76] Jasmine, K. P., & Kumar, P. R. (2014). Localized Rgb color histogram feature descriptor for image retrieval. *International Journal of Advances in Engineering & Technology*, *7*(3), 887.

[77] Lew, M. S. (2001). Feature selection and visual learning. In *Principles of visual information retrieval* (pp. 145-162). Springer, London.

[78] Jain, A. K. (1989). *Fundamentals of digital image processing*. Englewood Cliffs, NJ: Prentice Hall.

[79] Tuan, A. P. (2003). Optimization of Texture Feature Extraction Algorithm. Master's thesis, Department of Electrical Engineering, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, the Netherlands.

[80] Tang, H. H., Lee, Y. Y., & Gero, J. S. (2011). Comparing collaborative co-located and distributed design processes in digital and traditional sketching environments: A protocol study using the function–behaviour–structure coding scheme. *Design Studies*, *32*(1), 1-29.

[81] Hazra, A., & Gogtay, N. (2016). Biostatistics series module 6: correlation and linear regression. *Indian journal of dermatology*, *61*(6), 593.

[82] Notes on boxplots. URL http://web.pdx.edu/~stipakb/download/PA551/boxplot.html. [Online; posted 05-December-2018].

[83] Linh, N. (2019). How to compare box plots. URL https://blog.bioturing.com/2018/ 05/22/how-to-compare-box-plots/. [Online; posted 05-December-2018].

[84] Williamson, D. F., Parker, R. A., & Kendrick, J. S. (1989). The box plot: a simple visual method to interpret data. *Annals of internal medicine*, *110*(11), 916-921.

[85] Potter, K., Hagen, H., Kerren, A., & Dannenmann, P. (2006). Methods for presenting statistical information: The box plot. *Visualization of large and unstructured data sets*, *4*, 97-106.

[86] Frigge, M., Hoaglin, D. C., & Iglewicz, B. (1989). Some implementations of the boxplot. *The American Statistician*, *43*(1), 50-54.

[87] The Wellbeing@School team. Understanding and interpreting box plots. URL https://www.wellbeingatschool.org.nz/information-sheet/understanding-and-interpreting-box-plots.

[88] Wang, H., Wang, Y., Zhang, Q., Xiang, S., & Pan, C. (2017). Gated convolutional neural network for semantic segmentation in high-resolution images. *Remote Sensing*, *9*(5), 446.

[89] Younis, M. C., Keedwell, E., & Savic, D. (2018, October). An Investigation of Pixel-Based and Object-Based Image Classification in Remote Sensing. In *2018 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 449-454). IEEE.

[90] Cleve, C., Kelly, M., Kearns, F. R., & Moritz, M. (2008). Classification of the wildland–urban interface: A comparison of pixel-and object-based classifications using high-resolution aerial photography. *Computers, Environment and Urban Systems*, *32*(4), 317-326.

[91] Benvenuto, F., Piana, M., Campi, C., & Massone, A. M. (2018). A hybrid supervised/unsupervised machine learning approach to solar flare prediction. *The Astrophysical Journal*, *853*(1), 90.

[92] Mehrkanoon, S., Alzate, C., Mall, R., Langone, R., & Suykens, J. A. (2014). Multiclass semisupervised learning based upon kernel spectral clustering. *IEEE transactions on neural networks and learning systems*, *26*(4), 720-733.

[93] Zaid, M. A., & Zeki, A. M. (2015). An unsupevised package for multi-spectral image processing for remote data. *Journal of Advanced Computer Science and Technology Research (JACSTR)*, *5*(4), 113-122.

[94] Lillesand, T., Kiefer, R. W., & Chipman, J. (2015). *Remote sensing and image interpretation*. John Wiley & Sons.

[95] Weng, Q. (2012). Remote sensing of impervious surfaces in the urban areas: Requirements, methods, and trends. *Remote Sensing of Environment*, *117*, 34-49.

[96] Dean, A. M., & Smith, G. M. (2003). An evaluation of per-parcel land cover mapping using maximum likelihood class probabilities. *International Journal of Remote Sensing*, *24*(14), 2905-2920.

[97] Maulik, U., & Chakraborty, D. (2010). A robust multiple classifier system for pixel classification of remote sensing images. *Fundamenta Informaticae*, *101*(4), 286-304.

[98] Varma, M. K. S., Rao, N. K. K., Raju, K. K., & Varma, G. P. S. (2016, February). Pixel-based classification using support vector machine classifier. In *2016 IEEE 6th International Conference on Advanced Computing (IACC)* (pp. 51-55). IEEE.

[99] Veljanovski, T., Kanjir, U., & Ostir, K. (2011). Object-based image analysis of remote sensing data. *Geodetski vestnik*, *55*(4), 678-688.

[100] Blaschke, T. (2001). What's wrong with pixels? Some recent developments interfacing remote sensing and GIS. *GeoBIT/GIS*, *6*, 12-17.

[101] Qian, Y., Zhou, W., Yan, J., Li, W., & Han, L. (2015). Comparing machine learning classifiers for object-based land cover classification using very high resolution imagery. *Remote Sensing*, *7*(1), 153-168.

[102] Krishna, T. S., & Babu, A. Y. (2016, March). Three phase segmentation algorithm for high resolution satellite images. In *2016 International*

Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 2217-2223). IEEE.

[103] Wieland, M., & Pittore, M. (2014). Performance evaluation of machine learning algorithms for urban pattern recognition from multi-spectral satellite images. *Remote Sensing*, *6*(4), 2912-2939.

[104] Körting, T., Fonseca, L., Castejon, E., & Namikawa, L. (2014). Improvements in sample selection methods for image classification. *Remote Sensing*, *6*(8), 7580-7591.

[105] Sokolova, M., & Lapalme, G. (2007, May). Performance measures in classification of human communications. In *Conference of the Canadian Society for Computational Studies of Intelligence* (pp. 159-170). Springer, Berlin, Heidelberg.

[106] Sagar, R. (2018). Understanding decision tree, algorithm, drawbacks and advantages. URL https://medium.com/@sagar.rawale3/understanding-decision-tree-algori thm-drawbacks-and-advantages-4486efa6b8c3. [Online; posted 30-May-2018].

[107] Loh, W. Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(1), 14-23.

[108] Improving Classification Trees and Regression Trees- MATLAB & Simulink- MathWorks United Kingdom. (R2018b). [Online] Available: https://uk.mathworks.com/help/stats/improving-classification-trees-and-regression-trees.html.

[109] Kulkarni, V. Y. (2014). Effective learning and classification using random forest algorithm. *International Journal of Engineering and Innovative Technology (IJEIT)*.

[110] Ahmad, A. (2014). Decision tree ensembles based on kernel features. *Applied intelligence*, *41*(3), 855-869.

[111] MindTools. (2020). Decision trees: Choosing by projecting “expected outcomes”. URL https://www.mindtools.com/dectree.html.

[112] Choose Classifier Options- MATLAB & Simulink- MathWorks United Kingdom. (R2018b). [Online] Available: https://uk.mathworks.com/help/stats/choose-a-classifier.html.

[113] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*, 3-24.

[114] Cracknell, M. J., & Reading, A. M. (2014). Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information. *Computers & Geosciences*, *63*, 22-33.

[115] Thanh Noi, P., & Kappas, M. (2018). Comparison of random forest, k-nearest neighbour, and support vector machine classifiers for land cover classification using Sentinel-2 imagery. *Sensors*, *18*(1), 18.

[116] Ponti Jr, M. P. (2011, August). Combining classifiers: from the creation of ensembles to the decision fusion. In *2011 24th SIBGRAPI Conference on Graphics, Patterns, and Images Tutorials* (pp. 1-10). IEEE.

[117] Xu, L., Krzyzak, A., & Suen, C. Y. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE transactions on systems, man, and cybernetics*, *22*(3), 418-435.

[118] Kuncheva, L. I. (2014). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

[119] Giannakopoulos, G., Sakas, D., Anagnostopoulos, T., & Skourlas, C. (2014). Ensemble majority voting classifier for speech emotion recognition and prediction. *Journal of Systems and Information Technology*.

[120] Soto, V., Suárez, A., & Martínez-Muñoz, G. (2016). An urn model for majority voting in classification ensembles. In *Advances in Neural Information Processing Systems* (pp. 4430-4438).

[121] Zhou, Z. H. (2012). *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC.

[122] Shahzad, R. K., & Lavesson, N. (2013). Comparative analysis of voting schemes for ensemble-based malware detection. *Journal of Wireless Mobile*

Networks, Ubiquitous Computing, and Dependable Applications, *4*(1), 98-117.

[123] Yang, C., Yin, X. C., & Hao, H. W. (2014, August). Diversity-based ensemble with sample weight learning. In *2014 22nd International Conference on Pattern Recognition* (pp. 1236-1241). IEEE.

[124] Zhang, Y., Burer, S., & Street, W. N. (2006). Ensemble pruning via semi-definite programming. *Journal of Machine Learning Research*, *7*(Jul), 1315-1338.

[125] Giacinto, G., & Roli, F. (2001). An approach to the automatic design of multiple classifier systems. *Pattern recognition letters*, *22*(1), 25-33.

[126] Dos Santos, E. M., Sabourin, R., & Maupin, P. (2008, July). Pareto analysis for the selection of classifier ensembles. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation* (pp. 681-688). ACM.

[127] Cheng, J., Liu, J., Xu, Y., Yin, F., Wong, D.W.K., Tan, N.M., Tao, D., Cheng, C.Y., Aung, T. & Wong, T.Y. (2013). Superpixel classification based optic disc and optic cup segmentation for glaucoma screening. *IEEE transactions on medical imaging*, *32*(6), 1019-1032.

[128] Petrakova, A., Affenzeller, M., & Merkurjeva, G. (2015). Heterogeneous versus homogeneous machine learning ensembles. *Information Technology and Management Science*, *18*(1), 135-140.

[129] Mao, S., Jiao, L. C., Xiong, L., & Gou, S. (2011). Greedy optimization classifiers ensemble based on diversity. *Pattern Recognition*, *44*(6), 1245-1261.

[130] Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, *51*(2), 181-207.

[131] Özgür, A., Erdem, H., & Nar, F. (2016). Sparsity-driven weighted ensemble classifier. *arXiv preprint arXiv:1610.00270*.

[132] Tavish S. (2015). Finding optimal weights of ensemble learner using neural network. Technical report. URL

https://www.analyticsvidhya.com/blog/2015/08/optimalweights-ensemble-learner-neural-network/. [Online; posted 24-September-2015].

[133] Ozgur, A., & Erdem, H. (2018). Feature selection and multiple classifier fusion using genetic algorithms in intrusion detection systems. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *33*(1), 75-87.

[134] Li, L., Hu, Q., Wu, X., & Yu, D. (2014). Exploration of classification confidence in ensemble learning. *Pattern recognition*, *47*(9), 3120-3131.

[135] Li, Y., Gao, S., & Chen, S. (2012, July). Ensemble feature weighting based on local learning and diversity. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[136] Zeng, X., Wong, D. F., & Chao, L. S. (2014). Constructing better classifier ensemble based on weighted accuracy and diversity measure. *The Scientific World Journal*, *2014*.

[137] Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, *20*(4), 2923-2960.

[138] Neverova, N. (2016). *Deep learning for human motion analysis* (Doctoral dissertation).

[139] Bai, M., & Urtasun, R. (2017). Deep watershed transform for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5221-5229).

[140] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

[141] Noh, H., Hong, S., & Han, B. (2015). Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision* (pp. 1520-1528).

[142] Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(12), 2481-2495.

[143] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[144] Hijazi, S., Kumar, R., & Rowen, C. (2015). Using convolutional neural networks for image recognition. *Cadence Design Systems Inc.: San Jose, CA, USA*.

[145] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354-377.

[146] O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

[147] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[148] Xu, Z., Wang, R., Zhang, H., Li, N., & Zhang, L. (2017). Building extraction from high-resolution SAR imagery based on deep neural networks. *Remote Sensing Letters*, *8*(9), 888-896.

[149] Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. M. (2016). The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3234-3243).

[150] Couprie, C., Farabet, C., Najman, L., & LeCun, Y. (2013). Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*.

[151] Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. (2017). A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*.

[152] Ros, G., Ramos, S., Granados, M., Bakhtiary, A., Vazquez, D., & Lopez, A. M. (2015, January). Vision-based offline-online perception paradigm for

autonomous driving. In *2015 IEEE Winter Conference on Applications of Computer Vision* (pp. 231-238). IEEE.

[153] Muruganandham, S. (2016). Semantic segmentation of satellite images using deep learning. Space Engineering, master’s level. Luleå University of Technology, Department of Computer Science, Electrical and Space Engineering.

[154] Masko, D., & Hensman, P. (2015). The impact of imbalanced training data for convolutional neural networks. Degree Project, In Computer Science. Stockholm, Sweden.

[155] Wieland, M., Torres, Y., Pittore, M., & Benito, B. (2016). Object-based urban structure type pattern recognition from Landsat TM with a Support Vector Machine. *International Journal of Remote Sensing*, *37*(17), 4059-4083.

[156] Ding, C., Li, Y., Xia, Y., Zhang, L., & Zhang, Y. (2018). Automatic kernel size determination for deep neural networks based hyperspectral image classification. *Remote Sensing*, *10*(3), 415.

[157] Soekhoe, D., Van Der Putten, P., & Plaat, A. (2016, October). On the impact of data set size in transfer learning using deep neural networks. In *International Symposium on Intelligent Data Analysis* (pp. 50-60). Springer, Cham.

[158] Athanasiadis, T., Mylonas, P., Avrithis, Y., & Kollias, S. (2007). Semantic image segmentation and object labeling. *IEEE transactions on circuits and systems for video technology*, *17*(3), 298-312.

[159] Mičušĺík, B., & Košecká, J. (2009, September). Semantic segmentation of street scenes by superpixel co-occurrence and 3d geometry. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops* (pp. 625-632). IEEE.

[160] Shotton, J., Johnson, M., & Cipolla, R. (2008, June). Semantic texton forests for image categorization and segmentation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-8). IEEE.

[161] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A. & Blake, A. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, *56*(1), 116-124.

[162] Bhandari, A. K., Singh, V. K., Kumar, A., & Singh, G. K. (2014). Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. *Expert Systems with Applications*, *41*(7), 3538-3560.

[163] Bhandari, A. K., Kumar, A., & Singh, G. K. (2015). Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's, Otsu and Tsallis functions. *Expert Systems with Applications*, *42*(3), 1573-1601.

[164] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning. MIT press, Cambridge, Massachusetts.*

[165] Gamba, P., & Houshmand, B. (2002). Joint analysis of SAR, LIDAR and aerial imagery for simultaneous extraction of land cover, DTM and 3D shape of buildings. *International Journal of Remote Sensing*, *23*(20), 4439-4450.

[166] Thomas, N., Hendrix, C., & Congalton, R. G. (2003). A comparison of urban mapping methods using high-resolution digital imagery. *Photogrammetric Engineering & Remote Sensing*, *69*(9), 963-972.

[167] Matikainen, L., & Karila, K. (2011). Segment-based land cover mapping of a suburban area—Comparison of high-resolution remotely sensed datasets using classification trees and test field points. *Remote Sensing*, *3*(8), 1777-1804.

[168] Hernandez, C. S., Gladstone, C., & Holland, D. (2007, April). Classification of urban features from Intergraph's Z/I Imaging DMC high resolution images for integration into a change detection flowline within Ordnance Survey. In *2007 Urban Remote Sensing Joint Event* (pp. 1-8). IEEE.

[169] Mylonas, S., Stavrakoudis, D., Theocharis, J., & Mastorocostas, P. (2015). A region-based genesis segmentation algorithm for the classification of remotely sensed images. *Remote Sensing*, *7*(3), 2474-2508.

[170] Li, X., & Shao, G. (2014). Object-based land-cover mapping with high resolution aerial photography at a county scale in midwestern USA. *Remote Sensing*, *6*(11), 11372-11390.

[171] Huang, M. J., Shyue, S. W., Lee, L. H., & Kao, C. C. (2008). A knowledge-based approach to urban feature classification using aerial imagery with lidar data. *Photogrammetric Engineering & Remote Sensing*, *74*(12), 1473-1485.

[172] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, *7*(Jan), 1-30.

[173] Lessmanna, S., Seowb, H., Baesenscd, B., & Thomasd, L. C. (2013). Benchmarking state-of-the-art classification algorithms for credit scoring: A ten-year update. In *Credit Research Centre, Conference Archive*.

[174] Labatut, V., & Cherifi, H. (2011). Evaluation of performance measures for classifiers comparison. *arXiv preprint arXiv:1112.4133*.

[175] Markham, K. (2014). Simple guide to Confusion Matrix terminology. *data school*, *25*.

[176] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, *45*(4), 427-437.

[177] Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*(2), 83-85.

[178] Mitchell, T. M. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, *45*(37), 870-877.

[179] Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H. T. (2012). Learning from data vol. 4: AMLBook New York. *NY, USA*.

[180] Richard, G. (2014). Lm101-012: How to evaluate the ability to generalize from experience (cross-validation methods). URL https://www.learningmachines101.com/lm101012-evaluate-ability-generalize-experience-cross-validation-methods/. online; posted 08-September-2014.

[181] Bartlett, J., & Holloway, E. (2019). Generalized Information. *Communications of the Blyth Institute*, *1*(2), 13-22.

[182] Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, *5*(4), 8-36.

[183] Prashant, S. (2019). Refine your deep learning model. online; posted 12-February-2019.

[184] Zhang, C., Vinyals, O., Munos, R., & Bengio, S. (2018). A study on overfitting in deep reinforcement learning. *arXiv preprint arXiv:1804.06893*.

[185] Knoll, F., Hammernik, K., Kobler, E., Pock, T., Recht, M. P., & Sodickson, D. K. (2019). Assessment of the generalization of learned image reconstruction and the potential for transfer learning. *Magnetic resonance in medicine*, *81*(1), 116-128.

[186] Todorovic, S., & Nechyba, M. C. (2004, August). Detection of artificial structures in natural-scene images using dynamic trees. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* (Vol. 1, pp. 35-39). IEEE.

[187] Allitt, A., Blanksby, J., Djordjević, S., Maksimović, Č., & Stewart, D. (2009). Investigations into 1D-1D and 1D-2D urban flood modelling, *In WaPUG Autumn Conference,* vol. 25, 2009, pp. 1-12, Blackpool, UK.

[188] Bisht, D. S., Chatterjee, C., Kalakoti, S., Upadhyay, P., Sahoo, M., & Panda, A. (2016). Modeling urban floods and drainage using SWMM and MIKE URBAN: a case study. *Natural Hazards*, *84*(2), 749-776.

[189] Eldho, T. I., Zope, P. E., & Kulkarni, A. T. (2018). Urban Flood Management in Coastal Regions Using Numerical Simulation and Geographic Information System. In *Integrating Disaster Science and Management* (pp. 205-219). Elsevier.

[190] Schmitt, T. G., Thomas, M., & Ettrich, N. (2004). Analysis and modeling of flooding in urban drainage systems. *Journal of hydrology*, *299*(3-4), 300-311.

[191] Schmitt, T. G., & Thomas, M. (2009). Urban drainage modeling and flood risk management. In *Visualizing sustainable planning* (pp. 109-125). Springer, Berlin, Heidelberg.

[192] Andrew, W. (2012). Introduction to integrated catchment modelling. Technical report. URL https://slideplayer.com/slide/6392648/.

[193] Brown, R. J., Chanson, H., McIntosh, D., & Madhani, J. (2011). Turbulent velocity and suspended sediment concentration measurements in an urban environment of the Brisbane River Flood Plain at Gardens Point on 12-13 January 2011 [Report, CH83/11]. School of Civil Engineering, The University of Queensland, Australia.

[194] Werner, M. G. F., Hunter, N. M., & Bates, P. D. (2005). Identifiability of distributed floodplain roughness values in flood extent estimation. *Journal of Hydrology*, *314*(1-4), 139-157.

[195] Van de Voorde, T., Chormanski, J., Batelaan, O., & Canters, F. (2006, March). Multi-resolution impervious surface mapping for improved runoff estimation at catchment level. In *Proceedings of the 1st Workshop of the EARSeL Special Interest Group on Urban Remote Sensing: Urban remote sensing, challenges and solutions* (pp. 2-3).

[196] Lu, D., Li, G., Moran, E., Batistella, M., & Freitas, C. C. (2011). Mapping impervious surfaces with the integrated use of Landsat Thematic Mapper and radar data: A case study in an urban–rural landscape in the Brazilian Amazon. *ISPRS Journal of Photogrammetry and Remote Sensing*, *66*(6), 798-808.

[197] Yuan, F., & Bauer, M. E. (2006, May). Mapping impervious surface area using high resolution imagery: A comparison of object-based and per pixel classification. In *American society for photogrammetry and remote sensing annual conference proceedings 2006, Reno, Nevada*.

[198] Cablk, M. E., & Minor, T. B. (2003). Detecting and discriminating impervious cover with high-resolution IKONOS data using principal component analysis and morphological operators. *International Journal of Remote Sensing*, *24*(23), 4627-4645.

[199] Hester, D. B., Cakir, H. I., Nelson, S. A., & Khorram, S. (2008). Per-pixel classification of high spatial resolution satellite imagery for urban land-cover mapping. *Photogrammetric Engineering & Remote Sensing*, *74*(4), 463-471.

[200] Jiang, L., Chen, Y. and Wang, H. (2015). Urban flood simulation based on the SWMM model. *Proceedings of the International Association of Hydrological Sciences*, *368*, 186-191.

[201] Paquier, A., Mignot, E., & Bazin, P. H. (2015). From hydraulic modelling to urban flood risk. *Procedia Engineering*, *115*, 37-44.

[202] Gorton, E. & Clark, A. (2015). Ryde surface water management plan. Technical report. URL https://www.iow.gov.uk/azservices/documents/2821-Ryde-SWM P-Final-Report.pdf.

[203] Innovyze Ltd. (2012 June). Implementation of the New PR equation in InfoWorks ICM and CS. Technical report. URL http://blog.innovyze.com/wpcontent/uploads/2012/06/Implementation_of_the_New_PR_equation_in_InfoWorks_ICM_and_CS1.pdf.

[204] Innovyze Ltd. (2018). InfoWorks ICM (Integrated Catchment Modeling). URL https://archive.innovyze.com/products/infoworks_icm/.

[205] Christierson, B. V., Vidal, J. P., & Wade, S. D. (2012). Using UKCP09 probabilistic climate information for UK water resource planning. *Journal of Hydrology*, *424*, 48-67.

[206] Abdellatif, M., Atherton, W., Alkhaddar, R. M., & Osman, Y. Z. (2015). Quantitative assessment of sewer overflow performance with climate change in northwest England. *Hydrological Sciences Journal*, *60*(4), 636-650.

[207] Butler, D., Digman, C. J., Makropoulos, C., & Davies, J. W. (2018). *Urban drainage*. Crc Press.

[208] Kellagher, R. B. B. (2002). Storage requirements for rainfall runoff from greenfield development sites.

[209] Pell Frischmann. (2006a). Feock impermeable area plan. Truro drainage area study. URL https://www.yell.com/biz/pell-frischmann-exeter-69350/.

[210] Feock Parish Council. (2017). Feock neighbourhood development plan 2017–2030. Technical report. URL https://www.cornwall.gov.uk/media/27336285/app2-submission-f eock-ndp-plan-part-1-250417-v33.pdf. [online; posted 25-April-2017].

[211] Google Map Customizer. How to customize google map and export high-quality images using google map customizer. URL http://www.chengfolio.com/google_map_ customizer.

[212] Tessa Mero. (2012). How to use the Fireshot addon for Firefox, 2012. URL https://www.ostrai ning.com/blog/webdesign/fireshot-firefox/. [Online; posted 20-January-2012].

[213] Bache, K., & Lichman. M. (2013). UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences.URL http://archive.ics.uci.edu/ml. [Access: 15- Jan- 2018].

[214] Zhang, H., Fritts, J. E., & Goldman, S. A. (2005, March). A fast texture feature extraction method for region-based image segmentation. In *Image and Video Communications and Processing 2005* (Vol. 5685, pp. 957-968). International Society for Optics and Photonics.

[215] Chary, R., Lakshmi, D. R., & Sunitha, K. V. N. (2012). Feature extraction methods for color image similarity. *arXiv preprint arXiv:1204.2336*.

[216] Ping Tian, D. (2013). A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, *8*(4), 385-396.

[217] Acuña, E., & Rodríguez, C. (2005). An empirical study of the effect of outliers on the misclassification error rate. *Submitted to Transactions on Knowledge and Data Engineering*.

[218] Haralick, R. M., Shanmugam, K., & Dinstein, I. H. (1973). Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6), 610-621.

[219] Raut, M. A., Patil, M. M. A., Dhondrikar, M. C. P., & Kamble, M. S. D. (2016). Texture Parameters Extraction of Satellite Image. IJSTE - International Journal of Science Technology & Engineering, 2(11).

[220] MIKE Powered by DHI. (2016). Danish Hydraulic Institute (DHI), Denmark. URL http://releasenotes.dhigroup.com/2016/MIKEURBANrelinf.htm.

[221] Ustuner, M., Sanli, F. B., & Abdikan, S. (2016). BALANCED VS IMBALANCED TRAINING DATA: CLASSIFYING RAPIDEYE DATA WITH SUPPORT VECTOR MACHINES. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, *41*.

[222] Akbani, R., Kwek, S., & Japkowicz, N. (2004, September). Applying support vector machines to imbalanced datasets. In *European conference on machine learning* (pp. 39-50). Springer, Berlin, Heidelberg.

[223] Jabbar, H., & Khan, D. R. Z. (2015). Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). *Computer Science, Communication and Instrumentation Devices*.

[224] Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), pp. 5-32.

[225] Daumé III, H. (2012). A course in machine learning. *Publisher, ciml. info*, *5*(69).

[226] Längkvist, M., Kiselev, A., Alirezaie, M., & Loutfi, A. (2016). Classification and segmentation of satellite orthoimagery using convolutional neural networks. *Remote Sensing*, *8*(4), 329.

[227] Mnih, V. (2013). *Machine learning for aerial image labeling*. Ph.D. dissertation, University of Toronto (Canada). URL https://www.cs.toronto.edu/~vmnih/data/. [Accessed 15 Jan 2018].

[228] Kraus, K., & Waldhäusl, P. (1996). International Society for Photogrammetry and Remote Sensing. Committee of the Congress. 2D Semantic Labeling Contest. URL http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html. [Accessed 15 Jan 2018].

[229] Sharma, R. C., Hara, K., & Hirayama, H. (2017). A machine learning and cross-validation approach for the discrimination of vegetation physiognomic types using satellite based multispectral and multitemporal data. *Scientifica*, *2017*.

[230] Pirotti, F., Sunar, F., & Piragnolo, M. (2016). Benchmark of Machine Learning Methods For Classification of a Sentinel-2 Image. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, *41,* pp. 335–340, 2016.

[231] Brownlee, J. (2018). A gentle introduction to k-fold cross-validation. Vol. *7*, p. 2018.

[232] Jin, Y., & Sendhoff, B. (2008). Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(3), 397-415.

[233] Deb, K., & Gupta, S. (2011). Understanding knee points in bicriteria problems and their implications as preferred solution principles. *Engineering optimization*, *43*(11), 1175-1204.

[234] Legriel, J., Le Guernic, C., Cotton, S., & Maler, O. (2010, March). Approximating the pareto front of multi-criteria optimization problems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (pp. 69-83). Springer, Berlin, Heidelberg.

[235] Branke, J., Deb, K., Dierolf, H., & Osswald, M. (2004, September). Finding knees in multi-objective optimization. In *International conference on parallel problem solving from nature* (pp. 722-731). Springer, Berlin, Heidelberg.

[236] Van Veldhuizen, D. A., & Lamont, G. B. (1998, July). Evolutionary computation and convergence to a pareto front. In *Late breaking papers at the genetic programming 1998 conference* (pp. 221-228).

[237] Yun, Y., Nakayama, H., & Arakava, M. (2004). Generation of pareto frontiers using support vector machine. *MCDM'04*.

[238] Hsiao, K. J., Calder, J., & Hero, A. O. (2014). Pareto-depth for multiple-query image retrieval. *IEEE Transactions on Image processing*, *24*(2), 583-594.

[239] Kim, I. Y., & De Weck, O. L. (2006). Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation. *Structural and multidisciplinary optimization*, *31*(2), 105-116.

[240] Aquino, N.R., Gutoski, M., Hattori, L. & Lopes, H.S. (2017). The Effect of Data Augmentation on the Performance of Convolutional Neural Networks. 10.21528/CBIC2017-51.

[241] Tran, P. V. (2016). A fully convolutional neural network for cardiac segmentation in short-axis MRI. *arXiv preprint arXiv:1604.00494*.

[242] Kamphuis, C. (2018). Automatic Segmentation of Retinal Layers in Optical Coherence Tomography using Deep Learning Techniques. Master’s thesis, Computing Science Data Science, Radboud University, The Netherlands.

[243] Semantic Segmentation Using Deep Learning- MATLAB & Simulink- MathWorks United Kingdom. (R2018b). [Online] Available: https://uk.mathworks.com/help/vision/examples/semantic-segmentation-using-deep-learning.html.

[244] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

[245] Jang, H. U., Choi, H. Y., Kim, D., Son, J., & Lee, H. K. (2017, March). Fingerprint spoof detection using contrast enhancement and convolutional neural networks. In *International Conference on Information Science and Applications* (pp. 331-338). Springer, Singapore.

[246] Sameen, M. I., Pradhan, B., & Aziz, O. S. (2018). Classification of very high resolution aerial photos using spectral-spatial convolutional neural networks. *Journal of Sensors*, *2018*.

[247] Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, *106*, 249-259.

[248] Taghanaki, S.A., Zheng, Y., Zhou, S.K., Georgescu, B., Sharma, P., Xu, D., Comaniciu, D. & Hamarneh, G. (2019). Combo loss: Handling input and output imbalance in multi-organ segmentation. *Computerized Medical Imaging and Graphics*, *75*, 24-33.

[249] Akarachai, A. & Daricha, S. (2007). Avoiding local minima in feedforward neural networks by simultaneous learning. In Australasian Joint Conference on Artificial Intelligence, pp. 100–109. Springer, 2007.

[250] Landis, J. R., & Koch, G. G. The measurement of observer agreement for categorical data. biometrics, 33 (1): 159–174, 1977. *Cited on pages xi*, *171*.

[251] Pell Frischmann. (2006b) Penelewey impermeable area plan. Truro drainage area study. URL https://www.yell.com/biz/pell-frischmann-exeter-69350/.

[252] Pell Frischmann. (2006c) Playing Place impermeable area plan. Truro drainage area study. URL https://www.yell.com/biz/pell-frischmann-exeter-69350/.

[253] Younis, M. C., Keedwell, E., & Savic, D. (2018, April). Evaluating Image Classification Techniques for Improved Urban Wastewater System Model Calibration. In *EGU General Assembly Conference Abstracts* (Vol. 20, p. 16327).

[254] Hall, G. (2015). Pearson’s correlation coefficient. *other words*, *1*(9).