# Feature Extraction and Classification Approach for the Analysis of Roadside Video Data

**PhD Thesis**

**By**

Sujan Chowdhury

**A thesis submitted to the**

**School of Engineering and Technology**

**Central Queensland University, Australia**

In fulfilment of the requirements for the degree of
Doctor of Philosophy

**August 2016**

# Acknowledgements

I express the deepest gratitude to my principal supervisor Professor Brijesh Verma for his willingness to accept me as a research student and give me an opportunity to do research under his supervision. I thank him for his constant and sincere guidance, encouragement and valuable suggestions at every stage of this work.

I also express utmost gratitude to Dr Ligang Zhang, Dr Mary Tom and Dr David Stockwell for their advice, useful suggestions, and constant encouragement from the very beginning of my research. I am also grateful to all my colleagues for giving me an academic environment, encouragement, and inspiration. I want to thank research office staff and the Centre for Intelligent Systems (CIS) team for their friendship and support during the study time.

I wish to thank the Central Queensland University, Australia, for providing me an International Postgraduate Research Award (IPRA) and Central Queensland University Postgraduate Research Award (CQUPRA) to support my study. I am also grateful to the University for granting funds to support this research.

Finally, I would like to express my heartiest gratitude to my parents, my wife and other members of the family for their unmeasured sacrifices, continuous inspiration, and support.

# Statement of Originality

I certify that this thesis does not, to the best of my knowledge and belief:

a) incorporate without acknowledgement any material previously submitted for a degree in any institution of higher education;
b) contain any material previously published or written by another person except where expressly referenced within this thesis;
c) contain any defamatory material; or
d) disclose or divulge any material of a private or confidential nature that could compromise any individual or group's privacy.

Signature: [redacted]

Sujan Chowdhury

Date: August 30, 2016

# Abstract

Video analysis plays a significant role in a number of real-world applications, such as object tracking, scene analysis, and motion detection. Recently, roadside video analysis is becoming more important for fire risk management, infrastructure planning, tree trimming, and vegetation control. However, manual analysis of the huge volume of video data makes it unsuitable for many applications. Hence, automatic video analysis is becoming a very popular research area in image processing and machine learning fields.

One of the key components for video analysis is the extraction of appropriate features from the video frame/images. Extracting features from the selected regions in video frames and matching the extracted features with similar regions can solve many object detection and recognition problems. There is no specific feature that can be applicable in any domain. This is the reason that researchers look for a good feature representation based on specific applications. Features that are perceptually meaningful, have special analytical ability, are identifiable on different images and are scale invariants can be defined as good features.

In recent years, many applications have used video processing and remote sensing techniques for vegetation area identification either from aerial images or from satellite images. Some other applications used different types of vehicles and various types of sensors for specific applications such as weed and crop identification. However, getting high accuracy using existing methods is still a great challenge on the detection of target objects. Moreover, finding the fire-prone regions by analysing roadside video data is still an open challenge for researchers. Finding all the dense regions using a manual monitoring method is expensive, labour-intensive and time-consuming. To mitigate the above-mentioned problems, there have been increasing demands for automating the system to monitor the dense regions from the video data using image processing and machine learning techniques.

The research in this thesis focused on developing an automatic video analysis approach for roadside object detection and classification. It investigated the problem of detecting roadside objects in unstructured environments. The major problem during the detection of roadside objects is the extraction of appropriate features. Successful classification of objects heavily depends on good feature representation. Learning algorithms produce low accuracy if the feature representation is poor. Hence, the main objective of this research is to develop novel feature extraction and classification techniques which can perform an accurate detection that is fast enough for real-time application.

This thesis presents a number of novel techniques. The first technique described is the identification of vegetation areas from roadside images using the combination of novel feature extraction and

ensemble classifier technique. This technique achieved comparable performance for grass and non-grass area separation. The second technique is an extension of the first technique in respect to features. The third technique introduces bit quantisation and sequence features for object classification from roadside images. The fourth technique is based on a directional connectivity algorithm for grass density estimation. The fifth and final technique presented here is a perceptual feature extraction technique which can be applied to identify any objects on the roadside data frames.

The proposed techniques are applied to identify the roadside regions and objects on local roadside video data and on several popular benchmark datasets including Stanford, MSRC, and SIFT Flow. The results showing the effectiveness of the proposed techniques and detailed analysis are presented in this thesis. The need for future research is also highlighted, including the automation of sending alarms and the investigation of handling the shadow problem.

# List of Publications

Journals:

- Sujan Chowdhury, Brijesh Verma, David Stockwell, "**A novel texture feature based multiple classifier technique for roadside vegetation classification**", Expert Systems with Applications, vol. 42, issue 12, pp. 5047-5055, 2015.
- Sujan Chowdhury, Brijesh Verma, David Stockwell, "**A novel hybrid technique for roadside vegetation classification**", Australian Journal of Intelligent Information Processing Systems (AJIIPS), vol. 14, issue 1, pp. 1-6, 2014.

Conferences:

- Sujan Chowdhury, Brijesh Verma, "**Position gradient and plane consistency based feature extraction**", 23rd International Conference on Neural Information Processing (ICONIP 2016), pp. 673-681, 2016.
- Ligang Zhang, Brijesh Verma, David Stockwell, Sujan Chowdhury, "**Aggregating pixel-level prediction and cluster-level texton occurrence within superpixel voting for adaptive roadside vegetation classification**", International Joint Conference on Neural Networks (IJCNN 2016), pp. 3249-3255, 2016.
- Ligang Zhang, Brijesh Verma, David Stockwell, Sujan Chowdhury, "**Spatially constrained location prior for scene parsing**", International Joint Conference on Neural Networks (IJCNN 2016), pp. 1480-1486, 2016.
- Sujan Chowdhury, Brijesh Verma, Mary Tom, Mengjie Zhang, "**Pixel characteristics based feature extraction approach for roadside object detection**", International Joint Conference on Neural Networks (IJCNN 2015), pp. 1-8**,** 2015.
- Sujan Chowdhury, Brijesh Verma, David Stockwell, "**Analysis of hybrid classification technique on roadside vegetation to differentiate between dense and non-dense Region**", 10th International Conference on Simulated Evolution and Learning (SEAL 2014)**,** pp. 835-846**,** 2014.
- Sujan Chowdhury, Brijesh Verma, "**A novel feature extraction technique to retrieve vegetation class for fire risk assessment**", 8th International Conference on Signal Processing and Communication Systems (ICSPCS 2014), pp. 1-6, 2014.

# Contents

**Acknowledgements** ..... i

**Statement of Originality** ..... ii

**Abstract** ..... iii

**List of Publications** ..... v

**List of Figures** ..... ix

**List of Tables** ..... xii

**Chapter 1 Introduction** ..... 1

1.1 Background ..... 1

1.2 Problems and Motivation ..... 3

1.3 Research Aims ..... 5

1.4 Research Questions ..... 6

1.5 Original Research Contributions ..... 6

1.6 Thesis Structure ..... 7

**Chapter 2 Review of Related Work** ..... 9

2.1 Image Segmentation Techniques ..... 9

2.1.1 Thresholding Technique ..... 10

2.1.2 Boundary Based Technique ..... 10

2.1.3 Region Based Technique ..... 11

2.1.4 Super Pixel Based Technique ..... 12

2.1.5 Summary ..... 15

2.2 Feature Extraction Techniques ..... 16

2.2.1 Gray-level Co-occurrence Matrix ..... 16

2.2.2 Gabor Filter ..... 17

2.2.3 Local Binary Pattern ..... 17

2.2.4 Histogram of Gradient ..... 19

2.2.5 Scale Invariant Feature Transform ..... 20

2.2.6 Summary ..... 22

2.3 Classification Techniques ..... 23

2.3.1 Artificial Neural Network ..... 23

2.3.2 K-Nearest Neighbour ..... 23

2.3.3 Support Vector Machine ..... 24

2.3.4 Markov Random Field ..... 24

2.3.5 Neighbourhood-constrained k-means (NC-k-means) classification ..... 24

2.3.6 Fuzzy Classification Technique ..... 24

2.3.7 Summary ..... 26
**Chapter 3 Multiple Texture Features Extraction Techniques ..... 27**
3.1 Introduction ..... 27
3.2 Co-occurrence of Binary Pattern (CBP) and Ensemble Based Feature Extraction Technique 30
3.2.1 Introduction ..... 30
3.2.2 Proposed CBP Technique ..... 34
3.2.3 Feature Extraction ..... 34
3.2.4 Experiments and Results ..... 37
3.2.5 Result Analysis ..... 41
3.2.6 Summary ..... 48
3.3 Distance and Cross Correlation (DCC) Based Feature Extraction Technique ..... 49
3.3.1 Introduction ..... 49
3.3.2 Proposed DCC Technique ..... 50
3.3.3 Feature Extraction ..... 54
3.3.4 Experiments and Results ..... 56
3.3.5 Result Analysis ..... 56
3.3.6 Summary ..... 58
3.4 Quantisation Feature and Neural Network (QFNN) Based Feature Extraction Technique ..... 59
3.4.1 Introduction ..... 59
3.4.2 Proposed QFNN Based Technique ..... 60
3.4.3 Feature Extraction ..... 61
3.4.4 Post-Processing Technique ..... 63
3.4.5 Experiments and Results ..... 64
3.4.6 Result Analysis ..... 70
3.4.7 Summary ..... 71
**Chapter 4 Directional Connectivity Feature Extraction Technique ..... 72**
4.1 Introduction ..... 72
4.2 Proposed Directional Connectivity Feature Extraction Technique ..... 73
4.3 Feature Extraction ..... 76
4.4 Experiments and Results ..... 79
4.4.1 Data Collection and Setup ..... 79
4.4.2 Retrieval of Feature Set ..... 80
4.4.3 Training Feature Set ..... 82
4.4.4 Grass and Non-Grass Region Separation ..... 84
4.4.5 Window Selection from the Segmented Area ..... 86
4.4.6 Ground Reference Data ..... 88

4.5 Result Analysis ..... 89
4.5.1 Accuracy of Human versus Automated Survey ..... 93
4.5.2 Comparative Analysis ..... 94
4.6 Summary ..... 95
**Chapter 5 Multi-Scale Perceptual Features Extraction Technique ..... 97**
5.1 Introduction ..... 97
5.2 Multi-Scale Deep Learning Feature Extraction Technique ..... 99
5.3 Proposed Multi-Scale Perceptual Features ..... 101
5.3.1 Position Gradient Histogram (PosGH) ..... 102
5.3.2 Plane Consistency Estimation (PCE) ..... 104
5.3.3 Other Features ..... 108
5.4 Experiments and Results ..... 110
5.4.1 Datasets ..... 110
5.4.2 Evaluation Criteria ..... 111
5.4.3 Results ..... 111
5.5 Result Analysis ..... 112
5.5.1 Result on Stanford Background Dataset ..... 112
5.5.2 Comparative Analysis on the Stanford Background Dataset ..... 113
5.5.3 Result on MSRC Dataset ..... 114
5.5.4 Comparative Analysis on MSRC Dataset ..... 115
5.5.5 Result on SIFT Flow Dataset ..... 116
5.5.6 Comparative Analysis on the SIFT Flow Dataset ..... 117
5.5.7 Result on Vegetation Dataset ..... 117
5.5.8 Comparative Analysis on Vegetation Dataset ..... 118
5.6 Summary ..... 119
**Chapter 6 Conclusion ..... 120**
6.1 Contributions and Findings ..... 120
6.2 Future Research Directions ..... 125
**References ..... 127**

# List of Figures

Figure 1.1 Images Extracted from Roadside Video ..... 3

Figure 2.1 Image Segmentation using SLIC ..... 14

Figure 2.2 SIFT Feature Extraction from Input Image ..... 21

Figure 3.1 Dense and Sparse Area Detection: Top Image is an Example Roadside Image collected during the survey; Bottom Image shows Dense and Sparse Regions bounded in green ..... 28

Figure 3.2 Dense Regions Cropped from various parts of the Roadside ..... 28

Figure 3.3 Sparse Regions Cropped from various parts of the Roadside ..... 29

Figure 3.4 Roadside Object Detection: Left Image shows a Sample Roadside Image and the Right Image shows the Annotated Area with identifying colours ..... 29

Figure 3.5 Proposed CBP Technique ..... 34

Figure 3.6 Formation of Basic LBP Operator against its Neighbouring Pixels and Result Interpretation as a Binary Number ..... 35

Figure 3.7 Gray-level Co-occurrence Matrix Formulation for Texture Feature Extraction ..... 37

Figure 3.8 Sample Image Data for Dense Grass Regions ..... 38

Figure 3.9 Sample Image Data for Sparse Grass Regions ..... 38

Figure 3.10 Results using SVM ..... 41

Figure 3.11 Fivefold Cross Validation Accuracy for Each Classifier ..... 42

Figure 3.12 Results using k-NN ..... 44

Figure 3.13 Performance Analysis of Individual Classifiers and Hybrid Technique ..... 45

Figure 3.14 Sparse Grass Misclassified as Dense Grass ..... 46

Figure 3.15 Dense Grass Misclassified as Sparse Grass ..... 46

Figure 3.16 Vegetation Classification Flow Chart ..... 49

Figure 3.17 Feature Extraction Technique ..... 50

Figure 3.18 a) YCbCr Image b) YCbCr Image after Histogram Equalisation ..... 51

Figure 3.19 K-means Clustering ..... 52

Figure 3.20 Cross-Correlation Score Calculation ..... 55

Figure 3.21 Overview of the Proposed Technique: (a) Original Image (b) Converted YCbCr Image (c) Enhanced Image (d) Binary Image ..... 57

Figure 3:22 Feature Vector Extraction from the Block ..... 58

Figure 3.23 Block Value Calculation ..... 58

Figure 3.24 Most Significant Bit (MSB) Pattern Generation Technique ..... 61

Figure 3.25 Feature Vector Extraction Technique ..... 63

Figure 3.26 Post Processing of Pixels ..... 64

Figure 3.27 Post Processing of Neighbourhood Pixels ..... 64

Figure 3.28 Architecture of Neural Network ..... 66

Figure 3.29 Feature Vector Formation Technique ..... 66

Figure 3.30 Performance Curve during Training Phase ..... 68

Figure 3.31 Confusion Matrix after Training of Each Class ..... 68

Figure 3.32 Overview of Scene Labelling on the Original Frame ..... 69
Figure 3.33 Experimental Results ..... 70
Figure 4.1 Edges found by Horizontal Gradient Detection ..... 73
Figure 4.2 Grass Density Estimation: (a) Dense (b) Moderate (c) Sparse ..... 74
Figure 4.3 System Workflows for Frame Categorisation from Video ..... 75
Figure 4.4 Workflow for Grass Height Measurement ..... 76
Figure 4.5 High Connectivity for Dense Grass Region ..... 77
Figure 4.6 Low Connectivity for Sparse Grass Region ..... 79
Figure 4.7 Medium Connectivity for Moderate Grass Region ..... 79
Figure 4.8 Study Area ..... 80
Figure 4.9 Cropped Grass Regions ..... 81
Figure 4.10 Cropped Non-Grass Regions ..... 81
Figure 4.11 Features for Grass Region ..... 83
Figure 4.12 Features for Road as Non-grass Region ..... 83
Figure 4.13 Output for Sample #30 taken during Survey ..... 84
Figure 4.14 Output for Sample #26 Taken During Survey ..... 86
Figure 4.15 Output for Sample #54 Taken During Survey ..... 86
Figure 4.16 Window Selections from Frame ..... 87
Figure 4.17 Overview of Window Selection from Samples #26, #30 and #54 ..... 88
Figure 4.18 Ground Truths on 15 Windows ..... 88
Figure 4.19 Experimental Results with Segmented Output ..... 89
Figure 4.20 Experimental Results with Directional Connectivity Output ..... 89
Figure 4.21 Dense Grass taken from Video Data ..... 90
Figure 4.22 Dense Grass Classifications ..... 90
Figure 4.23 Sparse Grass taken from Video Data ..... 91
Figure 4.24 Sparse Grass Classifications ..... 91
Figure 4.25 Moderate Grass taken from Video Data ..... 92
Figure 4.26 Differences between the Thresholds for three types of Grass ..... 92
Figure 4.27 Performance Evaluation ..... 93
Figure 5.1 Multi-scale Deep Feature Learning Model ..... 99
Figure 5.2 Proposed Multi-scale Perceptual Features ..... 101
Figure 5.3 Example of Orientation Bins used in the Proposed Model ..... 102
Figure 5.4 Illustration of using the Gradient Histogram to Distinguish Superpixels with Similar Visual Appearance ..... 104
Figure 5.5 (a) Original Image (b) Entropy Image (c) Intensity Image (d) Binary Image after Opening Operations (e) Binary Image after Closing Operation (f) Label Probability Image (g) The Vertical Area Covered with Cyan Colour and Horizontal with Original Colour ..... 105
Figure 5.6 Superpixel Location Information ..... 108
Figure 5.7 Accuracy on Stanford Background Dataset ..... 112
Figure 5.8 Parameter Observations for Neural Network on Stanford Background Dataset ..... 112
Figure 5.9 Labelling result on Stanford Background Dataset ..... 114

Figure 5.10 Example of Bad Labelling on Stanford Background Dataset ..... 114
Figure 5.11 Class-wise Accuracy on MSRC Dataset ..... 114
Figure 5.12 Labelling Results on MSRC Dataset ..... 115
Figure 5.13 Labelling Results on SIFT Flow Dataset ..... 116
Figure 5.14 Labelling Results on Vegetation Dataset ..... 117
Figure 5.15 Class-wise Accuracy on Vegetation Dataset ..... 118

# List of Tables

Table 2.1 Summary of Image Segmentation for Vegetation Classification ..... 15
Table 2.2 Summary of Feature Extraction Technique for Vegetation Classification ..... 22
Table 2.3 Summary of Classification Technique used for Vegetation Classification ..... 25
Table 3.1 Data for Training and Testing ..... 38
Table 3.2 Results using Neural Network ..... 43
Table 3.3 Results using Proposed Hybrid Technique ..... 45
Table 3.4 Comparisons Chart for Classification Performance for Different Classifiers ..... 46
Table 3.5 Single Factor ANOVA Summaries ..... 47
Table 3.6 ANOVA Analysis Details ..... 47
Table 3.7 Pixel Value Assign ..... 52
Table 3.8 Dense and Non-dense Separations ..... 53
Table 3.9 Result Comparison ..... 56
Table 3.10 Performances (%) on the Annotated Dataset ..... 69
Table 3.11 Confusion Matrix within the Class ..... 70
Table 3.12 Result Comparison ..... 71
Table 4.1 Ground Truth for Survey Data ..... 88
Table 4.2 Accuracy on Trained Data ..... 90
Table 4.3 Accuracy on Test Data ..... 90
Table 4.4 Ground Truth for 100 images (939 windows) ..... 94
Table 4.5 Confusion Matrix for 100 images (939 windows) using Proposed Technique ..... 94
Table 4.6 Accuracy for 100 images (939 windows) using Proposed Technique ..... 94
Table 4.7 Confusion Matrix for 100 images (939 windows) using Gabor-based Technique ..... 95
Table 4.8 Accuracy for 100 images (939 windows) using Gabor-based Technique ..... 95
Table 4.9 Time Comparison Chart ..... 95
Table 5.1 Performance Comparision on Stanford Dataset ..... 113
Table 5.2 Performance Comparison on MSRC Dataset ..... 115
Table 5.3 Performance Comparison on SIFT Flow Dataset ..... 117
Table 5.4 Performance Comparison on Vegetation Dataset ..... 118

# Chapter 1 Introduction

This chapter presents the background, research aims and research questions related to roadside vegetation classification and propose a reliable and cost-effective solution using image processing and machine learning techniques. Section 1.1 provides the introductory background and Section 1.2 describes the significance of the study. Section 1.3 defines the research aims and Section 1.4 then presents the research questions. Contributions of the study are highlighted in Section 1.5. Finally, Section 1.6 provides an overview of the remaining chapters of the thesis.

## 1.1 Background

Roadside objects analysis [1] is an important and challenging research area in the field of computer vision. Classification of objects such as trees, shrubs, crops, weeds, fruits, road signs and vegetables have numerous applications in the field of agricultural systems as well as in daily applications. The aim of this research is classification of vegetation which will be used for fire-prone regions identification. The output of this research will be fruitful if it can be successfully implemented. This will be helpful in preventing roadside fires and, as a consequence, it can save property and lives. Every year, in Australia, bushfires cause huge losses. We can define fire-prone region is as an area of land where fire can occur due to the type and length of grass. Moreover coverage of grass region and biomass creates a big impact on defining the fire-risk location. The fire usually occurs in the dry season and becomes a frequent event. Ongoing researches mainly focus on satellite images for fire-prone regions identification. This research focuses on roadside fire risk instead of satellite image. Hence, this thesis provides new concepts for fire risk region identification. The video is recorded using vehicle where video cameras are fitted on four directions. Cameras are mounted in such a way that, it can take clear view of roadside object closure to road. An efficient roadside vegetation management system can save a huge amount of cost. Knowledge on different vegetation types is a prerequisite for successful implementation of this research. In rural areas, grasses are the primary visible component along the roadside. There are some other components like trees, shrubs, and weeds also visible on the roadside. Depending on the density and height, grasses are categorised into various types: e.g. dense grass, moderate grass, and sparse grass. Based on the types of grass, fire risk can be determined. Sparse grasses are less sensitive to fire while dense grasses are more sensitive to produce fire. Especially in the dry seasons, the risk of fire is high. When dense grasses are growing and come closer to the road they can result in fire and lead to the closure of road. A short-term solution is to inspect the area and manually cut the risk region around the roadside. The task is difficult to inspect in rural areas where the growth of grass is high. Usually, calendar-based grass trimming is performed on such areas. But the process is not only time-consuming, but also expensive, and needs enormous manpower. The

required viewing information about those roads cannot be obtained from satellite images. Hence, the use of image processing and machine learning techniques have great potential in assisting roadside vegetation management from video data.

Image processing is the art of understanding the world by using images to produce numerical or symbolic information. Image processing [2] is a key enabling research discipline for many applications, including process control, visual surveillance, automatic inspection, medical image analysis, indexing databases of images and image sequences, computational photography, data mining, autonomous vehicle or mobile robot control, etc. Besides these above-mentioned applications on image processing, it is closely related to event detection, object recognition, scene reconstruction, video tracking, image restoration, etc. In spite of these numerous specific applications, the basic goal of utilising computer vision is common. The underlying theme of this research field has been dedicated to developing an automated machine vision system with a view to duplicating the abilities of human vision by extracting abstract information from images for understanding what is happening in the scene. Hence, saving the man-hours and avoiding the manual and boring activities are the main focus of automatic vision-based approaches. In order to achieve this goal, it is necessary to train and teach the machine using a discriminant dataset with reliable information. In recent years, machine learning has become a fast growing research area due to the variety of applications such as engineering, scientific, image processing, and remote sensing etc. However, roadside vegetation analysis and differentiating grass regions is still in its infancy within the pattern-recognition based approaches. Due to the increase in roadside fire episodes, the specific application has grown in interest and also opens the area of various environment-related issues. During dry seasons in Australia, bushfires emanating from roadsides are a major ecological issue and have been identified as a risk. Probabilities of spreading bushfire heavily depend on roadside vegetation. Concerns about taking precautionary measures have become an important consideration for policy makers and developing a fire-risk mapping tool has therefore become an important research area. Policymakers have introduced some methods and taken some steps towards developing a tool for automatic identification of fire risk regions. However, no techniques have been developed to solve the ongoing issues and it is still an active research area. The aim of this research is not only to develop a tool for fire-risk identification, but also develop novel feature extraction techniques which can be applied to many other object identification issues. Moreover, the task is currently quite challenging and it is very hard to differentiate one object from another using existing feature extraction techniques. Figure 1.1 shows example images taken from the study area clearly shows what types of challenges need to be faced. Some challenges that can mention from this viewpoint include vegetation articulation, lighting variations, intra-class variations, multiple viewpoints, soil region identification, and varied appearance. Preliminary research is needed to explore the vegetation classification area. Initially, it is necessary to focus on identifying the different types of grasses using pattern recognition methods. In the agricultural research sector, differentiating

weeds from crops and categorising types of weeds (broad or narrow weed) is an active research area. This research will also be helpful in automatic weed identification strategies.

This research focuses on developing individual modules, where each one can produce different primitives. Then, classifiers will be developed to operate as a black-box to work on sets of training data that are labelled with ground truth, and will finally be used for test set classification. The individual modules interact with each other, and if one fails to provide a better solution, the whole process will be affected. Initially, this research will focus on region segmentation and object identification, and will explore the area for application. In order to improve the accuracy in detecting objects, this study might use the data on surrounding image regions. After potential identification of all the primitives from a scene, a more-refined categorisation can make the decision for scene content understanding.

Figure 1.1 Images Extracted from Roadside Video

## 1.2 Problems and Motivation

Image classification is a common research problem in the field of pattern recognition. Many image classification algorithms have already been developed for specific applications. For roadside vegetation classification several classification algorithm already applied for object classification. Moreover, a single classifier is not suitable for many applications. Rather than this, combinations of

classifiers are also creating great impact due to their accuracy. Hence, automatic image classification algorithms have become an important research field. To solve the classification problem several techniques have been developed such as K-Nearest Neighbour (K-NN), Adaptive boosting (AdaBoost), Support Vector Machine (SVM), Artificial Neural Network (ANN), and Wavelet based techniques. Rather than using a single classifier, fusions of classifiers create a big difference in classification problems and have received much attention in recent years.

Extracting meaningful and important information from an image is one of the great challenges. This information is useful to represent a pattern of objects. A better representation of features not only helps with recognition of objects but also reduces the computational cost. A lot of feature extraction techniques exist and play an important role to improve classification effectiveness and computational efficiency. At present, several strategies exist for feature extraction such as Local Binary Pattern (LBP), Local Ternary Pattern (LTP), Local Directional Pattern (LDP), Scale Invariant Feature Transform (SIFT), Gray-Level Co-occurrence Matrix (GLCM), Fast Fourier Transform (FFT), Gabor Wavelet (GW), Histogram of Gradient (HOG) and so on. Based on the strategy of extracting features, they are divided into two groups – high-level and low-level features. Low-level features directly extract information from images whereas high-level features are calculated based on low-level features.

Based on the characteristics of features, they can be grouped into general or domain specific features. Pixel-level, global and local features are considered as general features. From each pixel, it is possible to extract colour information, intensity, and first/second order derivative information which are considered as pixel-level features. This information has been successfully used in many machine vision applications. Rather than extracting features from a single pixel, but instead consider a patch or a region of interest and extract spatial information, these can be called as local features. This representation is powerful as it is ideally invariant to illumination, scale changes, rotation, and occlusion. Examples of local features include blobs, shapes, edge pixels and corners.

Global features describe the overall information from an image. These features include histogram, moment, energy, mean, and standard deviation of the whole image. Global feature descriptors are not very robust as, if there is a change on a part of the image; the whole resulting descriptor will change. If features are extracted from the image for any particular application, these features are known as domain-specific features. Depending on the application, features can be used individually or they can be grouped one after another for better differentiation between objects. As our application is domain specific, proposed techniques will use cascade features.

Existing feature extraction techniques have difficulty in recognising different roadside objects such as grasses, trees, roads, etc. from video data. The situation also becomes complex if the objects have different sizes, shapes, and colours. Hence, the main objective of this thesis is to propose new feature extraction and classification approaches for analysing roadside video data.

According to existing literature, to the best of our knowledge, there is no similar research on vegetation classification presented where the primary focus was on roadside video data. The application area is novel and adds new value in vegetation research. Furthermore, all existing features primarily focus on specific shape, colour, and structure, whereas the research focuses on developing novel features. Proposing a fusion of classifiers for vegetation classification is also a new contribution to this research.

In the existing literature, rather than remote sensing and satellite images, detection of roadside vegetation from vehicle mounted video cameras have been presented, where aerial images have been used. As there is a drastic difference between the viewpoint of aerial images and ground vehicle images, an effective method is essential for accurate classification. Although the existing methods can detect green vegetation, they may not identify all types of grasses and they cannot measure the grass height. The research will help to analyse roadside videos for a better decision-making in roadside vegetation management. It also seeks to identify the pros and cons related to roadside vegetation research and practices.

This thesis extends to multi-class object classification by expanding the object class. Scene labelling plays an important role in image understanding. However, the task is quite challenging as it faces some common issues like differentiating visually similar objects and differentiating vertical and horizontal objects. Thus, designing appropriate features for a particular object is still an open challenge for computer vision researchers. During scene labelling, some parts of various objects in real world images look very similar to each other, so it is very difficult and challenging to correctly label the image. For example, in roadside images, some portions of road and water are confusing and difficult to differentiate. Using existing features it is difficult to assign a class label for each pixel. The situation is more challenging when objects are similar in all aspects except in respect to the plane. For example, differentiating grass and tree objects from small superpixels is often very difficult as they may have different plane orientations. To address the above-mentioned problems, it is proposed to use multi-scale perceptual features which can solve these problems and improve accuracy.

## 1.3 Research Aims

There is no known method that can efficiently identify roadside objects like trees, high grasses, low grasses, medium grasses and shrubs, from the roadside video data and eventually identify the high-risk regions that could help taking precautionary measures to avoid fire risks on the roadside.

The main aim of this research is to develop new feature extraction and classification techniques to improve the classification accuracy of different roadside objects. The specific aims of the research presented in this thesis are as follows:

- Review the existing segmentation, feature extraction, and classification techniques relevant to object detection and classification from the roadside as well as from scene data.

- ❒ Propose novel feature extraction techniques for classification of roadside objects from the video data.
- ❒ Investigate different classifiers including ensemble classifiers and parameters for classification of roadside objects from the video data.
- ❒ Evaluate the proposed techniques on benchmark datasets and a local dataset collected from different parts of Queensland roads.

## 1.4 Research Questions

The main questions of this research are:

- ❒ What is the best way to separate the Region of Interest (ROI) or roadside objects from the video data?
- ❒ Why are existing feature extraction techniques not suitable to identify vegetation regions from roadside video data?
- ❒ What is the most suitable feature extraction technique or combination of techniques that can identify roadside objects like trees, grasses, shrubs, or any other objects on the roadside?
- ❒ What are the most suitable parameters for a classifier that can effectively classify the objects from video data in terms of efficiency and accuracy?
- ❒ How can the risk location be identified from the video?

## 1.5 Original Research Contributions

This research is a comprehensive study of developing feature extraction and classification techniques for analysing roadside video data. The major focus of the thesis is on inventing suitable feature extraction and classification techniques. The major contributions in terms of new techniques are highlighted in this section. The first is the use of multiple texture features. Introducing a new directional connectivity feature is highlighted as the second major contribution. The third contribution is the development of a perceptual feature extraction technique for object classification from the road scene. This novel approach has the potential of using the perceptual features on different types of data and different applications. Several journals and conference articles have been published based on original contributions presented in this thesis.

The specific original research contributions of this thesis are described below:

- ❒ A comprehensive literature review

  Reviews of existing techniques including segmentation, feature extraction and classification for object detection and classification have been conducted.

- Novel feature extraction techniques
  - Co-occurrence of Binary Pattern (CBP) based Feature Extraction Technique: a novel texture feature extraction technique with multiple classifiers is introduced and applied to roadside vegetation classification.
  - Distance and Cross-correlation (DCC) based Extraction Technique: a novel distance and cross-correlation based feature extraction was introduced to enhance the classification accuracy with cropped regions.
  - Quantisation Feature and Neural Network (QFNN) based Feature Extraction Technique: a combined most significant bit and sequence of colour channel based technique was introduced. This new technique helps to distinguish different roadside objects.
  - Directional Connectivity Feature Extraction Technique: the directional connectivity feature was introduced which helps to identify grass density estimation for a further decision about the grass depth and height.
  - Multiscale Perceptual Feature Extraction Technique: the introduction of the multiscale perceptual feature is another original contribution of the thesis. It helps to identify roadside objects from the scene.
- A comparative evaluation of the proposed techniques on local and benchmark datasets: a comparative analysis using the classification accuracies obtained on a local roadside dataset and three benchmark datasets (Stanford, MSRC, SIFT Flow) have been conducted and presented.

## 1.6 Thesis Structure

**Chapter 1** provides an overview of the background, motivation, research objectives and contributions of this research.

**Chapter 2** reviews the segmentation, feature extraction and classification techniques relevant to roadside object classification. It begins by examining the current challenges in vegetation classification as well as object identification, and explains why it is so difficult to achieve with existing features. It then defines the advantages and disadvantages of existing techniques in detail and explores why new feature extraction techniques are important.

**Chapter 3** presents the concepts of three different feature extraction techniques. The techniques are co-occurrence of binary pattern (CBP), direction and cross-correlation (DCC) and quantisation feature and neural network (QFNN). The chapter initially describes each technique, and then presents the experimental results and analysis of the results.

**Chapter 4** presents and discusses the concept of directional connectivity feature (DCF) for grass density estimation and its application on roadside video data to distinguish dense, sparse and moderate grasses in complex outdoor environments. It also covers how to use the DCF feature for fire risk region identification.

**Chapter 5** discusses multi scale perceptual feature (MSP) and deep learning technique in scene labelling tasks. To evaluate the effectiveness of the proposed technique, a series of experiments and a comparison of the results are also presented in this chapter.

**Chapter 6** concludes the thesis by summarising the contributions and findings. Finally, an outline for future research directions to extend the research is presented.

# Chapter 2 Review of Related Work

This chapter presents and explores background information and prior research relevant to image segmentation, feature extraction and classification techniques for object identification from images. These literature reviews particularly focus on roadside object detection and benefits associated with the information relevant to roadside vegetation classification. Additional research into the methodology involved in this thesis includes the use of segmentation, feature extraction and classification techniques to classify objects from complex scenes. In Section 2.1, a comprehensive literature review on segmentation techniques along with their advantages and disadvantages has been presented. Sections 2.2 focuses on feature extraction techniques with their relative advantages and disadvantages. Lastly, Section 2.3 emphasises on the classification techniques.

## 2.1 Image Segmentation Techniques

In object-based image analysis, one of the most important and critical steps is the ability to partition or extract a region of interest (ROI) from the image. Therefore, image segmentation is one of the fundamental problems in the area of image analysis and has been studied extensively during the last 40 years [3]. Like a human being, it is really difficult for a machine to segment the whole universe of existing objects from an image. It is an ongoing field of research and a lot of work still needs to be done to develop a complete solution. Being a well-studied problem, reviewing all of the literature is completely out of the scope. This research mainly focuses on roadside image segmentation. In this research, objects of special interest are trees, shrubs, grasses and road signs etc. Therefore, successful detection and segmentation of those objects from images are the most recurrent prerequisites of this research. According to [4], segmentation of an image is the partitioning of the image into a set of connected regions, where each region is uniform according to given features such as shape, grey level, colour, intensity or texture. The result of a segmentation method is usually a list of equivalence classes where each class represents an object or the background. Indeed, the classification and its underlying motivation depend on the author and sometimes on a specific goal. A wide range of segmentation algorithms has been developed to derive the information from remotely sensed images. As explained by [5], image segmentation techniques can be classified into four major categories:

1. Thresholding technique
2. Boundary-based technique
3. Region-based technique
4. Superpixel based technique

### 2.1.1 Thresholding Technique

The main principle behind thresholding is that image pixels are partitioned depending on their intensity values [6]. Given a threshold value T, a pixel gray value greater than the threshold is classified to category 1, while, a pixel gray value less than the threshold is classified to category 2. A thresholding function can be formally defined using Equation 2.1:

$$g(x,y) = \begin{cases} 1, & f(x,y) > T \\ 0, & f(x,y) \leq T \end{cases} \qquad (2.1)$$

In many cases, defining the threshold value varies depending on the images and the task is done by manually seeing which one works best in identifying the specific objects. The threshold can be defined using single intensity values or the combination of several intensity ranges. In order to represent a binary image, two-pixel intensity values 0 (black) and 1 (white) respectively will be used.

Extraction of the object from images is a complex problem in the area of image processing. The situation is more difficult if the objects are related to vegetation. To deal with such a complex problem, some researchers have used thresholding based methods for vegetation classification. In [7], Xie *et al.* described a method for oasis vegetation extraction based on a thresholding method. According to their observation, the Otsu method, and an iterative based threshold segmentation method show poorer performance than edge-based detection. The technique used the Roberts operator. The limitation of their work was that it cannot extract a variety of objects at the same time, and thus they proposed extending the work using multi-threshold determination. Montalvo *et al.* [8] proposed a framework for weed/crops identification from maize fields. They argued that their proposed system could identify plants (weeds and crops) where plants were contaminated with materials due to artificial irrigation or natural rainfall. Their system used the Otsu method [9] for thresholding with a combination of vegetation indices. Another similar work [10], instead of using automatic thresholding (such as Otsu), used a statistical mean value of the transformed image.

From the above discussion, it is clear that all vegetation classification techniques that mainly focus on weed identification from crops using vegetation indices and specific thresholds. Therefore, it cannot extract a variety of objects at the same time.

### 2.1.2 Boundary Based Technique

Boundary based image segmentation focus on edges which form a boundary between two regions with distinct properties of the image that might be considered an ROI. According to this technique when the difference between dark and light pixels is easily visible, a digital edge will be formed. Siddiqi *et al.* [11] developed an edge-based weed classification and recognition method based on morphological operation and edge linking algorithm. The limitation of their work lies in the fact that it cannot classify mixed weeds.

From the literature cited above, it is clear that, as fields of mixed vegetation have no structure, it is inappropriate to choose edge based techniques for general vegetation extraction.

### 2.1.3 Region Based Technique

Region-based segmentation is also known as the seeded region growing method, as it selects a set of seeds to partition an image into regions. The selection of the seeds can be operated manually or using automatic procedures based on appropriate criteria. According to [5], region splitting, merging and region growing are the two types of region-based segmentation. Region splitting and merging techniques subdivide the homogenous regions in an iterative process of division into arbitrary regions (splitting) and then join or further split these regions (merging) until some predefined conditions are fulfilled. The region growing technique aggregates neighbouring pixels into regions based on a homogeneity criterion that must take regard of the particular application involved.

The Watershed algorithm [12] has been proven a fast and powerful region merging image segmentation method [13] and effectively used in many kinds of literature for vegetation classification. [14]. Li *et al.* [15] adopted the idea of watershed algorithm for individual tree crown segmentation. An alternative method for tree species classification based on a watershed algorithm using a gradient of brightness is presented in [16]. In their study, they used high-resolution forest imagery taken from a helicopter and used for tree species classification named as a broad-leaved and needle-leaved tree.

In addition to the above-mentioned approaches, various segmentation techniques have been employed for vegetation classification by different researchers. Mean-shift based segmentation is widely discussed in different literature [17] [18] [19]. A hybrid segmentation algorithm based on Mean Shift (MS) with the Fisher Linear Discriminant (FLD) known as MMS-FLD is presented in [18] for crop image segmentation. The accuracy achieved using the method exceeds 97% which is higher than index-based methods. A serious drawback of this method is that it failed in multi-class cases and datasets used in this literature were too few. Zheng *et al.* [19] proposed an algorithm using multiple features for green and non-green vegetation segmentation using mean-shift. Although the proposed method outperformed than index-based method, but computation cost was too high and it cannot be used in real time. Many vegetation indices such as Normalised Difference Vegetation Index (NDVI), VI (Vegetation Index), and SVI (Soil-vegetation Index), have been widely used for segmentation of vegetation cover from different data sources. Zhang and Feng presented an approach based on NDVI and VI for vegetation and non-vegetation region segmentation [20]. More precisely, from vegetation images, it can also classify grasses and trees. The result of an accuracy assessment showed that the proposed method produced 97% accuracy over distributed vegetation. A combination of normalised cut and mean-shift based automated segmentation of vegetation has been presented in [21] and showed outstanding performance in the urban area. Similar kind of applications have been found from satellite image where using remote sensing techniques urban environments [22] have been segmented into different objects to avoid general obstacles. In the proposed method they used KITTI benchmark

object detection dataset for result comparison. Proposed system consists of four steps. Initially, a disparity map was created and in the next step using the disparity value, a graph has been created. Later, probability of each pixel that belongs to that object was calculated. In the final step, clustering was applied to find the object. Extensive object based image analysis from remote sensing images have been done in [23]. In [23] detailed study shows the use of spectral and contextual information in an integrated way. In recent years, Object Based Image Analysis (OBIA) become more popular rather than pixel-based image analysis [24]. OBIA approach comprising of two steps: segmentation and classification. During segmentation entire image region is segmented into image objects based on the homogenous spectral value of pixels. Knowledge-based or supervised training are conducted to classify the segmented objects. Minho Kim et al. [24] proposed an approach which achieved overall accuracy of 76.6 % and a Kappa of 0.57 at a scale of 48 using OBIA. Tree species maintenance shows great importance for policy maker in terms of urban planning, disaster management and environment protection. Hence, research on tree species separation from remote sensing image becomes an important research field in the area of computer vision. Corina et al. [25] proposed a system which can extract, segment and classify vegetation from high-resolution color infrared digital images. Vegetation was extracted using supervised classification model based on Support Vector Machine (SVM) and later to separate tree from the lawn Digital Surface Model (DSM) were used.

### 2.1.4 Super Pixel Based Technique

The super pixel based image segmentation [26] technique has gained popularity due to its accuracy and computational efficiency. It has proved increasingly useful for applications such as 3D reconstruction [27], object localisation [28], image parsing [29], and depth estimation [27]. Usually, the super pixel technique groups the pixels into small patches so that object boundaries become perceptually meaningful [30]. To make the super pixels useful, they must be computationally and representationally efficient, perceptually consistent and produce high-quality segmentations without overlapping [31]. Successful segmentation can give a better support for feature extraction and extract spatial information which will be helpful for the machine to learn and distinguishing objects.

For generating super pixels from images, many existing approaches have been developed. Every method has some advantages as well as some drawbacks and each may be better suited to a particular application [32] [33] [34]. Based on the existing super pixel generating algorithms, they can be broadly categorised as follows:

1. Graph-Based Algorithm
2. Gradient-Ascent-Based Algorithm
3. Simple Linear Iterative Clustering (SLIC) Algorithm

#### 2.1.4.1 Graph-Based Algorithm

In the graph-based approach, each pixel is treated as a node in a graph and superpixels are created based on the minimised cost function defined over the graph. Among these normalised cut [35] [36], graph cut [33] and bipartite graph [37] based methods are popular. In the normalised cut [35] based method, contour and texture cues were used to partition all pixels from a graph. One limitation of this method is that it seems slower than any other method for large images [34]. Another graph based approach proposed by Huttenlocher *et al.* [33] is where clustering for pixels is done based on the minimum spanning tree of the raw pixels. Although compared to normalised cut it is faster but it produces superpixels with irregular shape, which was not useful for feature extraction. Moreover, there was no option for controlling the number of superpixels in an image. In order to generate superpixels, Moore *et al.* [38] proposed a method by finding optimal paths to form the grid from the graph. The method is not computationally efficient as it is similar to the graph cut method. Another superpixel generating method proposed by Veksler *et al.* [39] is where overlapping image patches were stitched together to generate the superpixels.

#### 2.1.4.2 Gradient-Ascent Based Algorithm

In the gradient ascent based method, superpixels are generated initially by clustering the pixels randomly and the clusters are then refined iteratively based on some convergence criteria. For example in [32], superpixels were generated based on colour and intensity features. The performance of this method is relatively slow and it produced irregular shaped superpixels. Furthermore, the method does not have any control over the number of superpixels. Vincent *et al.* proposed a watershed-based [40] approach where local minima are used to produce superpixels. Resulting superpixels do not produce good boundary adherence. Although Vedaldi *et al.* [41] proposed a quick shift based approach that produced good boundary adherence, it is quite slow and does not allow control over the superpixel size. The Turbo pixel method [34] used local image gradients to generate the superpixels. Like other methods, it also has no control over the size and produced poor boundary coherence.

#### 2.1.4.3 Simple Linear Iterative Clustering (SLIC) Algorithm

To generate the superpixels using the SLIC segmentation algorithm [30], colour similarity and proximity was used. For clustering the pixels, five dimensional (5D) [*labxy*] colour space was used. Here [*lab*] is the CIELAB colour space and [*xy*] is the pixel position. One of the advantages of the SLIC method is that it produces equal cluster size and it has been done by the grid interval $S = \sqrt{\frac{N}{k}}$. Here N is the number of pixels and q is the number of superpixels it is desired to generate. To calculate the cluster in 5D space, a new distance measurement technique was applied as simple Euclidean distance

was not applicable without normalising the spatial distances. The distance measure can be defined as follows:

$$dist_{LAB} = \sqrt{(t_q - t_i)^2 + (x_q - x_i)^2 + (y_q - y_i)^2} \quad (2.2)$$

$$dist_{xy} = \sqrt{(a_q - a_i)^2 + (b_q - b_i)^2} \quad (2.3)$$

$$D_{istq} = dist_{LAB} + \frac{m}{q} dist_{xy}, \text{ m} = 10 \quad (2.4)$$

The overall procedure for simple linear iterative clustering is moving the cluster centres based on the lowest gradient in a 3 X 3 neighbourhood. The image gradient is computed as follows:

$$Grad(a,b) = \|I(a+1,b) - I(a-1,b)\|^2 + \| I(a,b+1) - I(a,b-1)\|^2 \quad (2.5)$$

Here, I (a, b) is the pixel value for lab colour at position (a, b) and $\|.\|$ is the $L_2$ norm. Based on the average labxy vector, an initial cluster centre was selected and the process is iteratively repeated until all the associating pixels are moved. The method is computationally efficient and there is an option for setting the number of superpixels for a given image. An example of the SLIC image segmentation process is shown in Figure 2.1.

Original Image

SLIC based Image Segmentation

Figure 2.1 Image Segmentation using SLIC

Table 2.1 provides a summary of the literature reviewed above in relation to existing image segmentation techniques for vegetation classification.

Table 2.1 Summary of Image Segmentation for Vegetation Classification

| Year | Author | Segmentation Technique | ROI extracted |
|---|---|---|---|
| 2016 | Zhang *et al.* [42] | SLIC | Grass, tree, road, sky etc. |
| 2013 | Montalvo *et al.* [8] | Otsu, VI | Weed, crop |
| 2013 | Ponti *et al.* [17] | Mean-shift | Green coverage, gaps and degraded areas |
| 2011 | Guijarro *et al.* [10] | Statistical mean | Weed, crop |
| 2011 | Li *et al.* [14] | Watershed | Tree |
| 2010 | Zheng *et al.* [18] | Mean-shift | Crop |
| 2010 | Xie *et al.* [7] | Otsu | Weed |
| 2009 | Zheng *et al.* [19] | Mean-shift | Green and non-green vegetation |
| 2009 | Siddiqi *et al.* [11] | Edge based technique | Broadleaf and wide leaf |
| 2008 | Omerevi *et al.* [21] | Novel segmentation method | Vegetation |
| 2005 | Zhang and Feng. [20] | NDVI, VI | Tree, grass |
| 2004 | Kanda *et al.* [16] | Watershed | Broad-leaved, needle-leaved |

### 2.1.5 Summary

Segmentation of vegetation for the automated detection of the trees, grasses, shrubs and other objects can be typically performed using a variety of approaches as discussed in Section 2.1. From the literature reviewed in Section 2.1, this section identifies the most suitable segmentation technique for the purpose of vegetation segmentation. The SLIC superpixel segmentation algorithm is a good choice for the purpose of vegetation segmentation and roadside object segmentation.

## 2.2 Feature Extraction Techniques

Intensive research has been done on feature extraction as it is a most important part of proper classification of objects. This section covers an extensive literature survey of existing feature extraction techniques related to vegetation and object classification. Although many techniques have been available for decades, this section only presents the most recent and successful approaches. Basic feature extraction techniques include colour, texture, and edge.

In order to find the meaningful information from images, texture [43] plays an important role in image processing. In terms of vegetation classification, several textural analysis techniques comprising only most recent and successful approaches are described in the literature review. The texture-based approaches are:

1. Gray-level Co-occurrence Matrix (GLCM)
2. Gabor Filter
3. Local Binary Pattern (LBP)
4. Histogram of Gradient (HOG)
5. Scale Invariant Feature (SIFT)

### 2.2.1 Gray-level Co-occurrence Matrix

In vegetation management, the popular and widely used texture feature is based on the Gray-Level Co-occurrence Matrix (GLCM). In 2004, tree species classification technique from high-resolution forest imagery was developed by Kanda *et al.* [16] where they used GLCM as the texture feature. Although many features can be generated from GLCM according to the literature, they selected homogeneity as their feature vector. Yu *et al.* [44] studied the Digital Airborne Imaging System (DAIS) with high spatial resolution imagery and conducted vegetation classification using 52 features including nine GLCMs features. In 2007, Ghazali *et al.* [45] introduced a 2 Dimensional Discrete Wavelet Transform (2D-DWT) based feature extraction technique to investigate the characteristic of narrow and broad weeds. According to [46], GLCM and FFT have been used to recognise types of weeds as either narrow or broad. Recognition results showed that, using the FFT-based technique, narrow and broad weed recognition achieved 89.2% and 91% classification accuracy. On the other hand, the GLCM based approach achieved 81% and 81.5% classification accuracy. Hence, for real-time application, FFT has been chosen. Moreover, Ghazali *et al.* [47] studied an oil palm plantation using the combination of statistical gray-level co-occurrence matrix (GLCM), scale-invariant feature transform (SIFT) and Fast Fourier Transform (FFT) features and obtained above 80% accuracy in the real-time weeds control system. Apart from this, Wu and Wen [48] utilised GLCM and histogram statistics-based texture features extracted from four spatial orientations corresponding to 0°, 45°, 90° and 135° respectively for weed and corn seedling recognition with a SVM classifier. The authors of Li *et al.* [49] investigated state-of-art texture descriptors such as Local Binary Pattern (LBP) and Gray-Level Co-Occurrence

Matrix (GLCM) for object-based vegetation species classification and evaluated their performance. In this work, accuracy was calculated using SVM on 10 spectral and texture feature descriptors. In contrast, 3 texture features (GLCM, Gabor Wavelet (GW), Uniform LBP (ULBP)) were used in [50] to classify vegetation species. The evaluation results suggest the need for selecting appropriate feature and classification algorithm for different categories. At a glance, accuracies obtained using existing texture features with classifiers were not as good as expected. The reason lies in the fact that the appearance of trees varied from season to season as well as with changing health status.

### 2.2.2 Gabor Filter

In 2003, to classify vegetation into different categories Tang *et al.* [51] performed texture based weed classification. In their proposed method to classify vegetation into broadleaf and grass categories, they used low-level Gabor wavelets features. For their research purpose, three different types of broadleaf weeds namely cocklebur, velvetleaf and ivy leaf and two different kinds of grass namely foxtail and crabgrass were used. The performance of the proposed method was promising but sample images used for this research were very low, which is one of the limitations of the proposed method. Another limitation of the proposed method was the size of the image dataset where 20 images were taken from each class, and for the two classes (weeds and grass) total images were forty (40). Moreover, their proposed method used a filter bank with four frequency levels to classify the images and, for that reason, the computational cost of the proposed method was high. Overcoming those limitations, Mustapha and Mustafa [52] achieved 88.17% accuracy for broad and narrow leaf categorisation using texture features based on Gabor Wavelet.

Later Ishak *et al.* [53] later proposed a new feature vector extraction process combining Gabor Wavelet (GW) and Gradient Field of Distribution (GFD). Their proposed method used Artificial Neural Network (ANN) as a classifier. Their dataset consisted of 400 images of 200 grasses and 200 broadleaf weeds with different lighting conditions were used. Results were promising and accuracy obtained using proposed method is the highest accuracy (93.75%) achieved for grass and weed classification.

### 2.2.3 Local Binary Pattern

Local Binary Pattern (LBP) is a promising technique for texture analysis. The automated weed classification method presented by Ahmed *et al.* [54] is based on a LBP operator which shows promising performance in terms of accuracy and computational efficiency. However, the proposed method was not used on mixed types of weeds. Most recently, Ahmed *et al.* [55] proposed an efficient weed classification method using local texture descriptors with different parameter settings. For their study, they used three different types of local patterns, namely Local Binary Pattern (LBP), Local Directional Pattern (LDP) and Local Ternary Pattern (LTP). The dataset used for this purpose was 400 images and showed robust performance in classifying the dominant category of broadleaf weeds or grass. The following briefly reviews previous work done in the area of vegetation extraction using

GLCM, Gabor, and LBP. Rather than this approach, many other approaches exist for vegetation feature extraction.

For object-based vegetation classification, spectral moment features have been presented in [14]. The proposed method was compared with state-of-the-art texture features such as Gabor filters, local binary patterns and gray-level co-occurrence matrix. Another approach described by Li *et al.* [56] is vegetation spectral feature extraction for classifying vegetation based on a decision tree algorithm. Søgaard *et al.* [57] investigated a Danish agricultural field for weed classification and introduce an active shape model from nineteen (19) most important weed species and obtained accuracy that ranged from 65% to 90%. Later, Rumpf *et al.* [58] showed promising results for weed classification using SVM based decision-making.

A morphological operation and binarisation based approach was introduced in [59] to detect a special kind of weed known as avena sterilis where SVM was used as a classifier for categorisation purposes. Compared to existing methods, the proposed method showed promising performance in terms of memory and computational power. In [60], the authors introduced a similar concept for weed recognition. Their proposed method used erosion and dilation based segmentation algorithm. The proposed algorithm used 240 images for evaluation to differentiate between broad and narrow grass images and obtained 89% accuracy. The performance of the proposed classifier was degraded by varying illumination conditions and other natural environment parameters. Ishak *et al.* extended the work in this area by introducing various types of feature extraction and classification techniques [61-63].

More recently, an automated machine vision system using SVM is presented to distinguish crops and weeds in [64]. To find out the different characterisation of weeds and crops, their proposed approach analysed 14 features to determine the optimal combination of features and achieved more than 97% accuracy. For evaluation, the methods used a set of 224 test images were used. Another author proposed a new statistical based weed classifier in [65]. The proposed method determined the sample variance of each image and proposed a new threshold value by processing 140 images (70 of each category). The category was selected according to a threshold value and achieved 97% classification accuracy for the broad and narrow weeds.

Weed detection from lawn areas has not been studied extensively. In [66], weed detection in a lawn is presented using a morphological operation and a Bayesian-based approach. The proposed method achieved accuracies between 77.71-91.11%. Using Dempster-Shafer's theory and Ant Colony Optimisation algorithm Li *et al.* [67, 68] presented a multi-feature and shape features fusion based approach. Rather than applying a learning-based approach, the authors of [8] proposed a new automatic method based on several sequential stages for weeds/crops identification in images from maize fields. Results obtained using their method showed that it works favourably.

### 2.2.4 Histogram of Gradient

The Histogram of Gradient Feature (HOG) is one of the widely used feature extraction techniques for object classification. HOG has achieved great success on sign detection [69], vehicle detection [70] and pedestrian detection [71]. The method was first proposed by Dalal *et al.* for pedestrian detection [71] and later successfully applied to other object detection and localisation [72] work. In recent years, it has been successfully applied to vegetation classification [73] from remote sensing images. The idea is quite simple but has a high impact on object detection. A good feature makes the classification job much easier. Calculation of the HOG feature can be summarised as follows. Initially, the image gradient for each pixel in a $D_x \; X \; D_y$ detection window is calculated as follows:

$$\text{For x direction:} \qquad D_x\,(r,c) = D(r,c+1) - D(r,c-1) \quad (2.6)$$

$$\text{For y direction:} \qquad D_y\,(r,c) = D(r-1,c) - D(r+1,c) \quad (2.7)$$

Gradients are transformed to polar coordinates of angle and magnitude and the angle is constrained to be between zero and 180 degrees as follows:

$$M(x,y) = \sqrt{{D_x}^2 + {D_y}^2} \quad (2.8)$$

$$\theta = \frac{180}{\pi}\left(\tan_2{}^{-1}\frac{T_y}{T_x} \;\; mod\ \pi\right) (2.9)$$

where $\tan_2^{-1}$ is the four-quadrant inverse tangent, thus yielding values between π and −π.

The next step is to accumulate the pixels whose orientation is close to the bin boundary over non-overlapping cells of size C×C pixels (C = 9), then normalise the block feature using its Euclidean norm as follows:

$$\text{L1 norm: } c \leftarrow \frac{c}{(\|c_k\| + \epsilon)} \quad (2.10)$$

$$\text{L2 norm: } c \leftarrow \frac{c}{\sqrt{(\|c_k\|^2 + \epsilon)}} \quad (2.11)$$

where $\epsilon$ is a small positive constant that prevents division by zero is gradient-less blocks.

Finally, to construct the final HOG feature h, all normalised block features are concatenated as follows:

$$h \leftarrow \frac{h}{\sqrt{(\|h\|^2 + \epsilon)}} \quad (2.12)$$

Using only the HOG feature may not give state of the art results, so some researchers propose a combination of features such as HOG-LBP [74] or HOG-SIFT based features for object detection.

### 2.2.5 Scale Invariant Feature Transform

To detect and describe the local features in an image, Lowe *et al.* [75] established an algorithm which is invariant to scaling, rotation, and translation and is popularly known as the Scale Invariant Feature Transform (SIFT) algorithm. In recent years, SIFT has been successfully used in many applications which include scene modelling [76], object recognition [77], robot localisation [78], and object tracking [79]. Scene classification [80] [81] using SIFT descriptors has shown its effectiveness and achieved competent accuracy in object recognition. Quelhas *et al.* [82] proposed an approach to integrate the SIFT descriptor and probabilistic latent space [83] model. Using local feature descriptors, they achieved good accuracy for scene classification. The probabilistic model gives a good low-level scene representation which can capture meaningful scene aspects. SIFT feature was also used to construct the Bag-of-Visual-Words (BOVWs) model [84] which has been successfully used in remote sensing based scene classification. Although the BOVWs method narrows the gap between high-level and low-level features, it does not consider spatial information from images. To overcome that limitation of the existing BOVWs method, some researchers incorporate spatial information. However, the process is time-consuming and complex. Research also shows that a combination of features (SIFT, LBP, and colour) [85] generated from multiple local descriptors does better than single descriptors. To define different types of terrain SIFT descriptors have been used and compare the performances compared with Local Ternary Patterns (LTP) descriptor [86], and the Local Adaptive Ternary Patterns (LATP) descriptor [87]. According to Bosch *et al.* [81], using a hybrid generative/discriminative approach for scene classification achieved superior performance. From the above-mentioned applications, it is obvious that SIFT can be used as a powerful local descriptors and detectors. To detect the interesting points on the object, the first step is scale-space extreme detection which can be defined by the function:

$$W(x, y, \sigma) = Gauss(x, y, \sigma) * I(x, y) \quad (2.13)$$

Here, the convolution operator is defined by (*) and the input image is defined as I(x, y). $Gauss(x, y, \sigma)$ denotes the Gaussian kernel and is described as follows:

$$Gauss(x, y, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{x^2 + y^2}{2\sigma^2}\right] \quad (2.14)$$

where $\sigma$ denotes the standard deviation.

Difference of Gaussians (DOG) can be calculated by taking the difference between the Laplacian operators and is given by:

$$DOG\ (x, y, \sigma) = W(x, y, k + \sigma) - W(x, y, \sigma) \quad (2.15)$$

To detect the extreme all the points need to be checked and if they either minimum or maximum those points can be treated as extreme and the location of extreme z given by:

$$z = -\frac{\delta^2 D^{-1}}{\delta x^2}\frac{\delta D}{\delta x} \quad (2.16)$$

Based on the local image properties, key points can be described as follows.

From the Gaussian smooth image, compute the gradient magnitude, m given by:

$$m(x,y) = \sqrt{(W(x+1,y) - W(x-1,y))^2 + (W(x,y+1) - W(x,y-1))^2} \quad (2.17)$$

And compute orientation $\theta$ $as$

$$\mu\,(x,y) = \tan^{-1}\left(\frac{W(x+1,y)-W(x-1,y)}{W(x,y+1)-W(x,y-1)}\right) \quad (2.18)$$

To create the key point, take the highest local peak within the orientation.

A scenario for feature extraction using SIFT is shown in Figure 2.2. The scenario represents a vegetation area comprising grass, tree, soil, and sky.

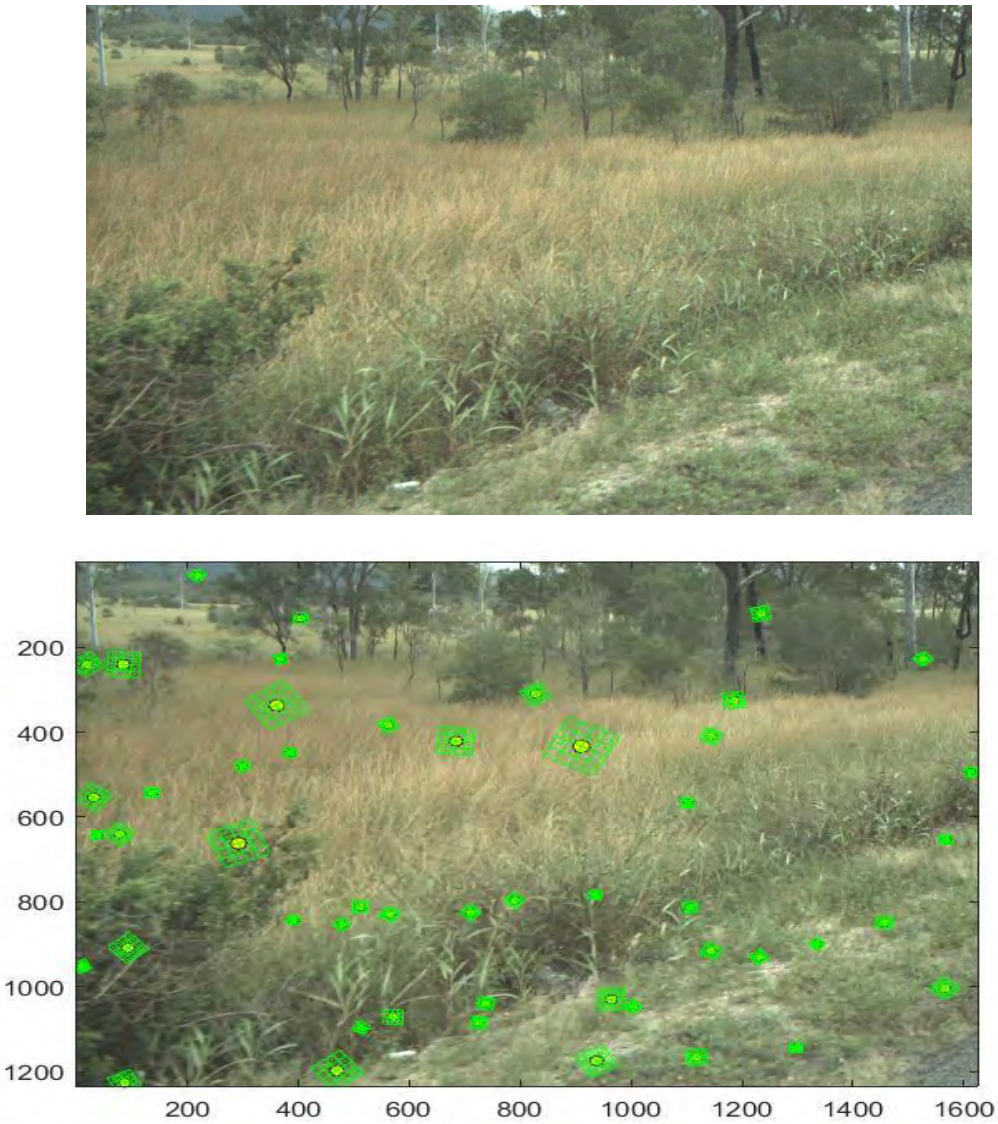



Figure 2.2 SIFT Feature Extraction from Input Image

Table 2.2 provides a summary of the literature reviewed above in relation to existing image feature extraction techniques for vegetation classification.

Table 2.2 Summary of Feature Extraction Technique for Vegetation Classification

| Year | Author | Feature Extraction Technique | ROI |
|---|---|---|---|
| 2016 | Zhang *et al.* [42] | SIFT | Grass and non-grass |
| 2014 | Ahmed *et al.* [55] | LBP, LTP, LDP | Broadleaf and grass |
| 2011 | Ahmed *et al.* [54] | LBP | Broadleaf and grass |
| 2010 | Li *et al.* [49] | Spectral Texture Feature | Vegetation species (Eucalyptus tereticornis, Eucalyptus melanophloia, and Corymbia tesselaris) |
| 2010 | Li *et al.* [50] | GLCM, GW, ULBP | Vegetation species |
| 2009 | Ishak *et al.* [53] | Gabor and GFD | Broadleaf and grass |
| 2009 | Wu *et al.* [48] | GLCM, Histogram | Weed and corn |
| 2008 | Ghazali *et al.* [47] | GLCM, FFT, SIFT | Narrow and broad weed |
| 2007 | Mustafa *et al.* [46] | GLCM, FFT | Narrow and broad weed |
| 2006 | Yu *et al.* [44] | GLCM | Forest, shrub, herb, and non-vegetation |
| 2005 | Mustapha *et al.* [52] | Gabor | Broad and narrow leaf |
| 2004 | Kanda *et al.* [16] | GLCM | Broad-leaved and needle-leaved trees |
| 2003 | Tang *et al.* [51] | Gabor | Broadleaf and grass weeds |

### 2.2.6 Summary

In computer vision, the development of feature extraction techniques plays an important role for successful identification of roadside objects. The most popular techniques for pattern recognition are texture features and statistical features. Commonly used texture-based feature extraction techniques include: GLCMs, LBPs, and HOG. The thesis proposed novel feature extraction techniques which were efficiently employed in the dataset and accuracy is proved using validation technique.

## 2.3 Classification Techniques

Many automated classification techniques have been investigated for the classification of vegetation regions during the last decade. These techniques include: Support Vector Machines (SVMs) ( Wu *et al.* (2009) [48]) (Li *et al.* [49]), Artificial Neural Networks (ANNs) (Tang *et al.* [51]), Pulse-Coupled Neural Networks (PCNNs), k-Nearest Neighbour (k-NN) (Yu et al. [44]), Maximum Likelihood Classifier (MLC) (Kanda *et al.* (2004)[16]) (Yu *et al. (*2006) [44]), Decision Tree Forest (DTF). Other techniques include statistical methods based on the use of statistical models.

### 2.3.1 Artificial Neural Network

In 2003, the authors of Tang *et al.* [51] used feed forward back propagation ANN classifier to classify weeds into broadleaf and grass classes. A similar type of NN-based approach was also presented by Mustapha *et al.* [52] to classify broad and narrow leaf weeds. The authors of Kanda *et al.* [16] used the supervised classification using Maximum Likelihood (ML) decision rules for the detection of the broad-leaved tree and needle-leaved tree.

### 2.3.2 K-Nearest Neighbour

Yu *et al.* [44] proposed a method for distinction of the forest, shrub, herb, and non-vegetation using *k*-Nearest Neighbour (*k*-NN) classifier and pixel based Maximum Likelihood Classifier (MLC) was used as a benchmark to evaluate their performance. The authors of Mustafa *et al.* [46] proposed a Line Measuring Technique (LMT) approach based on Fast Fourier Transform (FFT) for classification between narrow and broad weed in both offline images and recorded video. A comparison between GLCM based approach also presented on this research and illustrate the logic for choosing FFT based technique. The authors of Ghazali *et al.* [47] applied a Continuity Measure (CM) technique. To determine the best threshold equation a combination of linear classification tool was proposed. In their proposed method, for narrow and broad weed classification the best result with a correct classification rate of 86.1% and 88.4% respectively obtained with the angle of 45° and scale 3. Studies on weed/corn classification proposed by Wu *et al.* [48] have shown the superiority of SVMs over Back Propagation Neural Network (BPNN), suggesting that the SVM shows promising results to identify in-field weed/corn images. The authors of Li *et al.* [49] used SVM to detect Vegetation species (*Eucalyptus tereticornis, Eucalyptus melanophloia,* and *Corymbia tesselaris*) based on incorporating spectral moment features. A empirical comparison of seven machine learning algorithms namely K-Means Clustering (KM), Multilayer Perceptron Neural Networks (MLPNN), Support Vector Machines (SVM), Radial Basis Function Networks (RBFN), Single Decision Tree (SDT), Linear Discriminant Analysis (LDA), and Decision Tree Forest (DTF) with 3 texture features (GLCM, Gabor and ULBP) by means of classifying vegetation species in a power line corridor using high resolution aerial imagery has been presented in Li *et al.* [50]. According to the the study, classification performance varied based on

performance matrix, characteristics of datasets and the feature(s) used for classification. Ishak *et al.* [53] have implemented the weed image classification system utilising the combination of Gradient Field Distribution (GFD) and Gabor Wavelet (GW) techniques with a Single Layer Perceptron (SLP) model.

### 2.3.3 Support Vector Machine

Recently an automated detection of broadleaf and grass have been developed by Ahmed *et al.* [54] based on LBP feature and results is evaluated using template matching and SVM classifier. Analysis on results showed that SVM yields better classification rate than template matching. The work is extended by the same authors and gets highest classification accuracy using Local Directional Pattern (LDP). SVM also used for vegetation extraction using the Texture Measures (TM) [25]. Texture measures composed of different feature vector which includes Mean, Contrast, Angular Second Moment, Entropy, Inverse Different Moment, Correlation, Range and Standard Deviation. Four color spaces (RGB, XYZ, Lab and HSV) were considered separately to compute the feature vector.

### 2.3.4 Markov Random Field

Markov Random Field (MRF) model have been successfully applied in the field of image processing and computer vision [88] from early 90's [89] and recently been applied for classification of vegetation from remote sensing images [90]. In such applications, spatial-contextual information was incorporated with MRF [91]. MRFs have the ability to solve many problems in remote sensing applications, e.g. segmentation, sub-pixel analysis, change detection and classification. Usually it examines the global and local properties by quantifying spatial autocorrelation among pixels [92]. Recent studies show that MRF based image classification can achieve better results compared to conventional classification technique [90]. In some applications computational cost is so high for MRF based land cover classification [92].

### 2.3.5 Neighbourhood-constrained k-means (NC-k-means) classification

NC-k-means classification algorithm comprises of four steps [93]. Initially traditional k-means algorithm was performed according to a distance criterion. In the next step, depending on the neighbourhood size pure neighbourhood index (PNI) and non-overlapping pure neighbourhoods ($\delta_k$) was calculated. Depending on the pre-defined value either neighbourhood based k-means clustering or pixel-based k-means clustering was performed. In final stage, depending on the objective function, iteration will be stopped.

### 2.3.6 Fuzzy Classification Technique

To extract the forest and grassland, Chengfan li et al. [94] proposed a fuzzy classification technique based on multi-thresholds method using high resolution remote sensing image.

Table 2.3 provides a summary of the literature reviewed above in relation to existing image classification techniques for vegetation classification.

Table 2.3 Summary of Classification Technique used for Vegetation Classification

| Year | Author | Classification Technique used | Classified Objects | Accuracy |
|---|---|---|---|---|
| 2014 | Ahmed *et al.* [55] | Template Matching and SVM | Broadleaf and grass | LDP (98.5%) |
| 2011 | Ahmed *et al.* [54] | Template Matching and SVM | Broadleaf and grass | Template Matching (88.3%),<br>SVM (98.5%) |
| 2010 | Li *et al.* [49] | SVM | Vegetation species (Eucalyptus tereticornis, Eucalyptus melanophloia, and Corymbia tesselaris) | 95% |
| 2010 | Li *et al.* [50] | KM, LDA, RBFN, MLPNN, SVM, SDT, DTF | Vegetation Species | DTF (71.07%) |
| 2009 | Ishak *et al.* [53] | SLP | Broadleaf and grass | 94% |
| 2009 | Wu *et al.* [48] | SVM | Weed and corn | 92.31 - 100% |
| 2008 | Ghazali *et al.* [47] | CM | Narrow and Broad weed | Narrow (86.1%)<br>Broad (88.4%) |
| 2007 | Mustafa *et al.* [46] | LMT | Narrow and Broad weed | Offline:<br>Narrow (89.2%)<br>Broad (91%)<br>Playback Recorded Video:<br>Narrow (80.6%)<br>Broad (81.1%) |
| 2006 | Yu *et al.* [44] | *k*-NN | Forest, shrub, herb, and non-vegetation | 51 - 58% |
| 2005 | Mustapha *et al.* [52] | ANN | Broad and narrow leaf | 88.17% |
| 2004 | Kanda *et al.* [16] | ML | Broad-leaved and needle-leaved tree | 80 - 90% |
| 2003 | Tang *et al.* [51] | ANN | Broadleaf (velvetleaf and ivyleaf) and grasses (giant foxtail and crabgrass) | 95% |

### 2.3.7 Summary

Classification accuracy mostly depends on suitable classifier selection and tuning of appropriate parameters. Sometimes using only one classifier will not produce good accuracy. Hence fusion of classifiers is necessary to achieve good classification accuracy.

# Chapter 3 Multiple Texture Features Extraction Techniques

This chapter presents multiple texture feature extraction techniques for dense and sparse vegetation region identification from roadside images. Three different types of texture feature extraction techniques are proposed to solve roadside object classification issues. Initially, from whole images, a small portion of grass regions was cropped and differentiated based on the density of grasses. Later, all objects from the entire images are classified. The focus of this chapter is on feature extraction techniques and how they are applied to the target images using machine learning. The overall feature extraction strategies can be divided into three types:

1. Co-occurrence of Binary Pattern (CBP) and ensemble based technique
2. Distance and Cross Correlation (DCC) based technique
3. Quantisation Feature and Neural Network (QFNN) based technique

This chapter is organised into the following sections. In Section 3.2, provides the details of the Co-occurrence of Binary Pattern (CBP) and ensemble based technique along with the experimental results. Section 3.3 presents the Distance and Cross Correlation (DCC) based technique with experimental setup and performance evaluations. In Section 3.4, a novel Quantisation Feature and Neural Network (QFNN) based technique is presented and tested on an annotated dataset collected from various parts of the roadside.

## 3.1 Introduction

The aim of feature extraction is to identify an object within an image or differentiate one object from another using their properties. To fulfil this aim, three goals are set: it is first necessary to manually crop images of dense and sparse regions and create a database. Hence, start finding an appropriate property which can differentiate between both regions. Later, to increase the performance, several different properties were tested and new feature extraction strategies are proposed. Finally, all objects from the image were segmented only the particular segmented region will process for further decision will process. Figure 3.1 shows an example of the sample images collected during the survey. In the same figure, dense and sparse target regions are also shown. Figures 3.2 and 3.3 show some dense and sparse regions collected from different parts of the roadside. Although it is sometimes easier for the differentiation to be undertaken by humans, but such continuous monitoring for long periods is not feasible. For this reason, it is essential to develop an intelligent system to avoid the boredom of this task and make the system more efficient and reliable. The initial focus is on dense and sparse region differentiation using appropriate features. It is then necessary to segment the grass, road, soil, sky and

tree regions from the images. As the target area is grass, the focus is on locating the grass regions. Once the grass regions are identified correctly, it is necessary to decide whether the identified grass regions are dense or sparse.

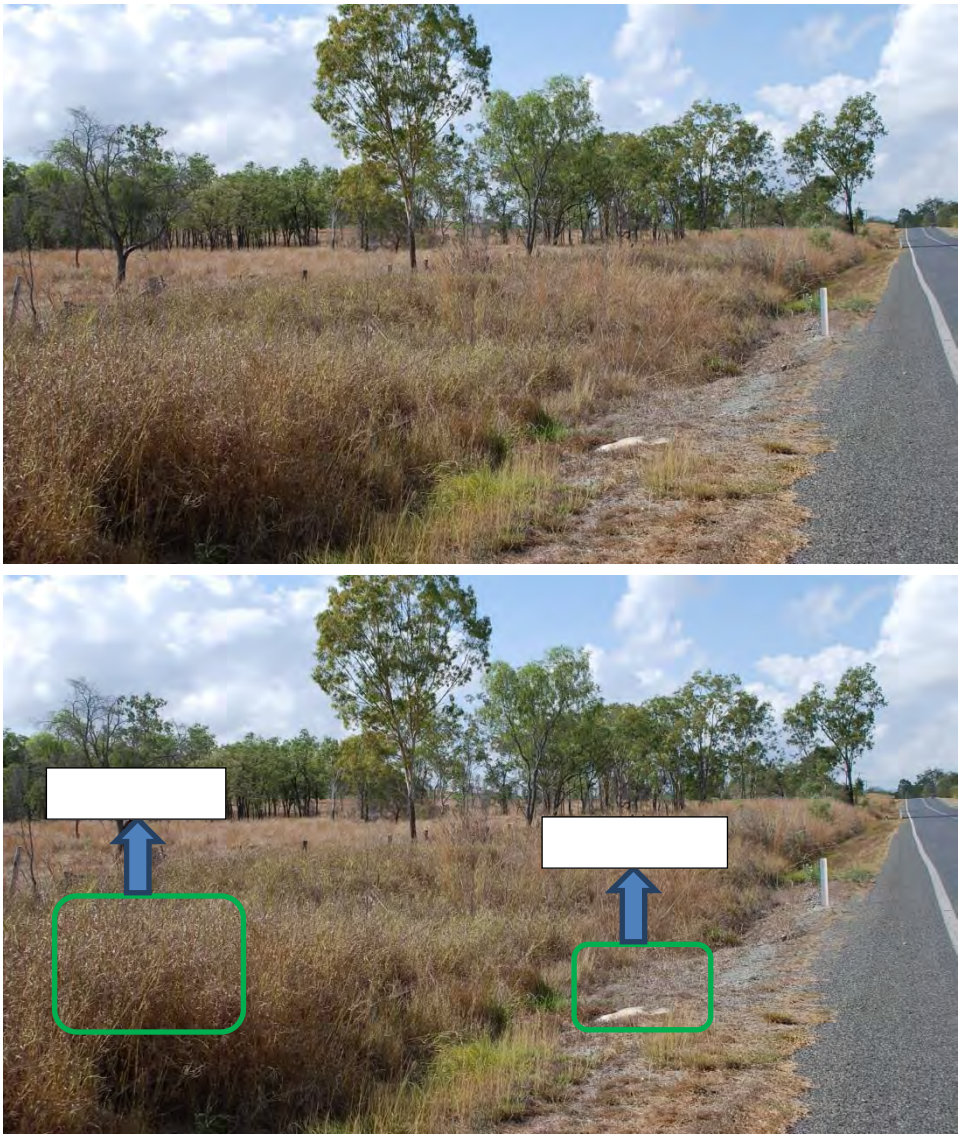


Figure 3.1 Dense and Sparse Area Detection: Top Image is an Example Roadside Image collected during the survey; Bottom Image shows Dense and Sparse Regions bounded in green

Identifying the dense and the sparse regions is really crucial as the level of fires risk will be highly dependent on the accuracy of this process. Hence, appropriate feature representation plays a vital role in differentiating those regions.

Figure 3.2 Dense Regions Cropped from various parts of the Roadside

Figure 3.3 Sparse Regions Cropped from various parts of the Roadside

Figure 3.4 shows a sample image with its corresponding ground truth. Common objects like grass, trees, roads, soil and sky are seen in roadside images. Hence the primary aim of this research is finding appropriate features for those objects. The technique can then be generalised so that it will work on any objects. As the primary target region is grass, the initial focus is on accurate classification of grass regions.

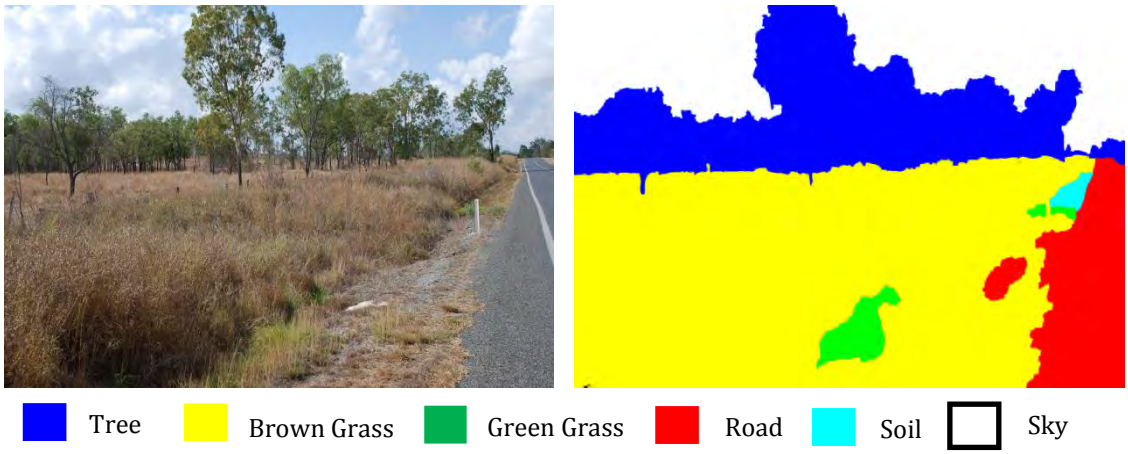



Figure 3.4 Roadside Object Detection: Left Image shows a Sample Roadside Image and the Right Image shows the Annotated Area with identifying colours

## 3.2 Co-occurrence of Binary Pattern (CBP) and Ensemble Based Feature Extraction Technique

This section presents a novel texture feature and ensemble classifier technique for dense and sparse grass region classification [95]. Fire-risk region identification mostly depends on it. The technique was evaluated with survey data collected from various parts of Queensland. To evaluate the overall performance using both quantitive and qualitative methods, CBP technique applied to recognise certain types of grass regions. The overall framework consists of five steps. In step one, proposed technique applied some pre-processing which includes median filtering, RGB to grayscale conversion and image resizing. In step two, features were extract using the proposed technique and named as Co-occurrence of Binary Pattern (CBP) as it is based on Gray-Level Co-occurrence Matrix (GLCM) and Local Binary Pattern (LBP). Training of feature vector using multiple classifiers has been done in step three. Classification results are described in step four. In step five, validation and statistical analysis are described. To fuse the decision and make the ensemble method three different classifiers both for training and testing were used. A majority voting was applied to take the final decision from the classifiers. The classifiers are Support Vector Machine (SVM), Feed Forward Back-Propagation Neural Network (FF-BPNN) and k-Nearest Neighbour (k-NN). Incorporation of three classifiers increases the diversity and improves the classification accuracy. Results were evaluated using five-fold cross-validation and obtained accuracies were listed. The accuracy was promising and to prove its significance, an ANOVA (Analysis of Variance) test was also conducted.

### 3.2.1 Introduction

In computer vision, multiple object detection from images is increasing in interest due to its numerous applications. Roadside image analysis and the capability of identifying every object from roadside would have a great impact on real world applications. Hence, lots of research opportunities related to roadside object analysis have been studied extensively. However, there is no existing method which can classify all objects from roadside images. All existing methods focus on a particular application and each method is developed based on the target object. Appropriate feature extraction is the key challenge for classifying those objects. As the main focus here is on fire risk identification, identifying grass from the images is the key concern of this research. Although there is no known method related to grass region identification, similar kinds of research have been done in the areas of tree identification, weed identification, crop identification etc. A comprehensive review of those applications helped to identify the related advantages and disadvantages of existing methods [96]. Moreover, it provided a clear idea about the existing feature extraction and classification techniques for successful vegetation identification [97].

Gabor Wavelet (GW) based feature extraction was proposed by Burks *et al.* [98] to classify between broadleaf and grass. There were two limitations with their proposed method. The first was that the number of sample images used for training and testing was very low (40 images where 20 from each class) and the second limitation was the processing time. A similar approach based on Gabor Wavelet was proposed by Mustafa *et al.* [52] to categorise broad and narrow leaf weeds and achieved an accuracy of 88.17%. Søgaard *et al.* [57] proposed an active shape models that could classify 19 different weeds from the images with an accuracy of 65% to 90%. Due to low processing speed, it cannot apply on real time weed control applications. Ghazali *et al.'s* [47] proposed a method that achieved 80% accuracy for narrow and broad weed differentiation using a combination of features. The features used in the proposed method were: Fast Fourier Transform (FFT), Gray-Level Co-occurrence Matrix (GLCM), and Scale Invariant Feature Transform (SIFT). The proposed method was mostly rule-based - which cannot apply to general classifications. Rumpf *et al.* [58] proposed a sequential classification based approach that showed promising results and achieved an overall classification accuracy of 97.7%. For simplification, their proposed approach divided the work into two phases. In the first phase, weed areas were identified and the later phase classified the weeds into species based on shape. Data was trained based on some specific shapes and, as the species grow, the method fails to recognise them.

To extract a new set of the feature vector for weed classification, Ishak *et al.* [53] proposed a combination of a Gradient Field of Distribution (GFD) and Gabor Wavelet (GW). In their approach, the dataset was quite large and more than 400 images were used for training and testing. For training and classification, Artificial Neural Network (ANN) was applied and listed the accuracy was 93.75%. Some misclassifications occurred between broadleaf and weeds. Some highly dense weed regions were classified as broadleaf due to overlapping and, due to stems on broadleaf, some were misclassified as grass. Further developments were required to overcome that situation. Ishak et al. extensively studied weed classification and introduced several feature extraction techniques for broad and narrow weed classification which have been listed in [61-63]. One approach used a combination of Gradient Field Distribution (GFD) and Gray-Level Co-occurrence Matrix (GLCM) and used Neural Network as a classifier, while another approach used a Gabor Filter and Fast Fourier Transform (FFT) combination with Support Vector Machine (SVM) as a classifier. The third approach used a curve detection method to identify the region of interest. The proposed curve detection method was based on the quadratic equation and two degrees of freedom were considered. For learning and classification of data, the proposed approach used a single layer perceptron (SLP) classifier. Edge link based weed classification has been introduced in [11] to identify the region of interest. Some previous works [59] [60] used morphological operation based weed classification. Compared to the previous approach, Tellaeche *et al.* [59] proposed an approach that showed promising performance in terms of memory and computational power. In their proposed approach, binarisation, morphological opening and closing along with a SVM classifier were used for segmentation. On the other hand, Siddiqi *et al.* [60] proposed

an approach for broad and narrow weed classification and achieved over 89% accuracy. Their proposed method was designed based on erosion and dilation segmentation algorithm. The performance of the proposed classifier was degraded due to variations in illumination conditions, wind and other natural environment parameters. Moreover, the dataset used by the proposed system was very low.

More recently, in order to classify crops and weeds from digital images, a new method has been presented in [64]. To form the feature vector, a combination of size and rotation invariant shape, colour, and moment features were considered. Among them, nine features were chosen using a forward selection and backward elimination feature selection method. Using Support Vector Machine (SVM) as a classifier, their proposed approach achieved around 97% accuracy over a set of 224 test images. These nine features were: Solidity, Mean value of 'r', Mean value of 'b', Standard deviation of 'r', Standard deviation of 'b', second-order moment invariants (ln(Ø1) of area, ln(Ø2) of area), third-order moment invariants (ln(Ø3) of area, and ln(Ø4) of area) [64]. The proposed method fails and produced segmentation errors when backgrounds were noisy and holes. But it performed better while segmenting from soil backgrounds. Therefore, in real-time implementation prior to feature extraction more effective and efficient image enhancement techniques should be introduced.

In order to recognise the presence of weeds and to differentiate between weeds with broad leaves and narrow leaves, Ahmad *et al.* [65] proposed a statistically based weed classifier approach. Their proposed approach used some threshold to determine the narrow and broad category grasses and 140 sample images were used to evaluate the classification performance. Due to a lower number of samples classification performance and accuracy is so high. Ahmad *et al.* [54] proposed an approach for weed classification based on Local Binary Pattern (LBP) and Support Vector Machine (SVM). In terms of accuracy and computational efficiency, their proposed method shows promising performance. But the method shows poor performance for mixed weed images.

Moreover, vision-based approaches have been applied in crop field for crop row area identification. Many researchers have proposed different strategies, but no method is workable in all cases. Among them Guerrero *et al.* [99] proposed an approach for crop row identification. Intrinsic and extrinsic parameters along with their perspective projections were used to determine the crop lines. As the decisions are based on the horizontal lines, it fails in a complex scenario. To identify the greenness, Romeo *et al.* [100] proposed an approach based on fuzzy clustering. Dynamic threshold adjusting is the main finding of their research. Although their method performs well when grass is green, it fails in any other case and in real scenarios grasses are not always green. Various kinds of grass with various shapes and colours are found in a real scenario. Other researchers proposed [101] a multi-region of interest based approach which shows superior performance than the Hough transform based approach. The method is not robust as they assume that grass will be green.

To achieve an appropriate discrimination between weeds, crops, and soil and make the system robust, the main challenge is making the system workable under varying conditions of lighting within the less processing time. To address those issues, Burgos *et al.* [102] proposed a system which consisting of two independent subsystems. Their proposed system is the combination of Fast Image Processing (FIP) and Robust Crop Row Detection (RCRD). The first subsystem is used to classify the weeds and crops more quickly and deliver the results while the second subsystem is a slower process and used to correct the first subsystem's mistakes. Using the system, they achieved 95% accuracy on weeds and 80% accuracy on crops. But the key robustness of the proposed system is the segmentation method which shows better performance under varying lighting conditions. The colour indices ($r = -0.884$, $g = 1.262$, $b = -0.311$) used in the proposed system can create a gray image which can be easily transformed into a binary image by a simple image thresholding adjustment method.

Detection of roadside vegetation based on the visible spectrum has been used successfully. Colour and texture features based approaches were proposed by Harbas *et al.* [103] to detect the vegetation. One limitation of the proposed method was speed. Moreover, if the vegetation becomes green, their proposed method cannot correctly detect the vegetation area. The authors extended their method by adding a new texture feature and considering the distance from the camera and has been discussed in [104] and [105]. In their new extended method, a two-dimensional continuous wavelet transform with oriented wavelets was used instead of using an entropy feature. However, in terms of computational cost, it requires significant computational resources.

In recent years, calculating per-pixel accuracy on crop/weed discrimination from hyperspectral data has been demonstrated with accuracies of over 80%. However, the vast majority of methods used supervised methods using different sensors including monochrome, colour, multispectral and hyperspectral cameras. Wendel *et al.* [106] proposed an unsupervised method which allowing the classifier to continually update weed appearance models as conditions changed. Their proposed method did not compare the results with static training data. Therefore, further investigation is needed to improve the classification performance and refine the training data.

According to existing literature, there are no existing methods which can segment vegetation data from roadside video. Moreover, compared to existing research the method of data collection and application area also a key issue of our research. Existing research on weed and crop identification, either from satellites or from aircraft and ground mounted cameras, uses a single classifier. Our proposed method uses a novel texture feature along with an ensemble classifier to classify roadside vegetation and eventually identify fire risk areas.

### 3.2.2 Proposed CBP Technique

The proposed method aims to distinguish between dense and sparse regions by the fusion of texture features and use of an ensemble classifier. The overall process is described below. Dense and sparse regions were initially selected by cropping from roadside images and creating the associated database. The next step involves some pre-processing of the images for further decision making. Then features are extracted using the proposed feature extraction model. Those extracted features are then trained using three base classifiers. Finally, the test images are classified using majority voting for a final decision. The overall scenario has been presented in Figure 3.5.

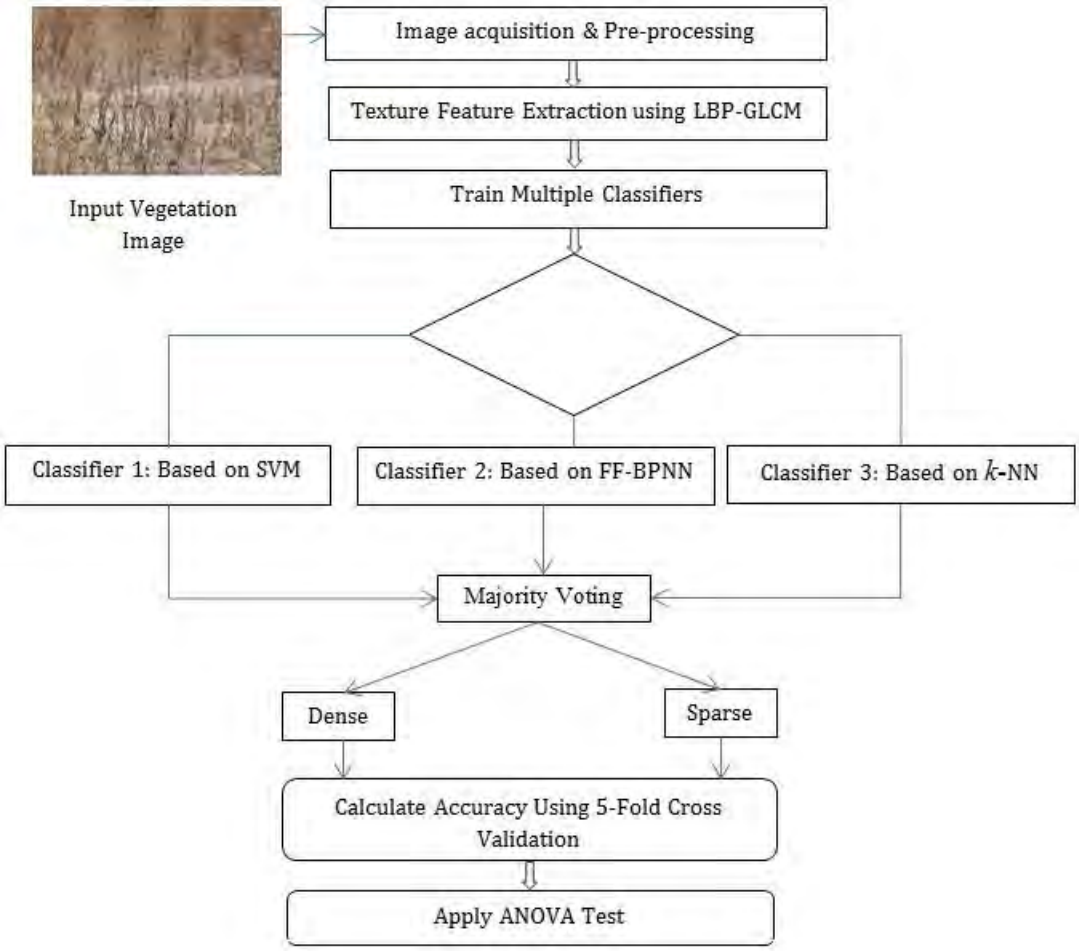



Figure 3.5 Proposed CBP Technique

### 3.2.3 Feature Extraction

Extracting appropriate feature from target images is one of the crucial steps for vegetation classification. To extract appropriate features, it is necessary to analyse the many features which will be suited with the target objects. Texture feature is the best choice for vegetation classification. From existing literature research shows that no single technique has appropriate features which can optimally determine the difference between two objects. Hence, a combination of features were used which serve our specific purpose. The proposed technique is based on Local Binary Pattern (LBP) and

Gray-Level Co-occurrence Matrix (GLCM) and named as Co-occurrence of Binary Pattern feature extraction.

The basic idea for developing the LBP feature extractor was transforming a gray scale image into an array or image of integer labels which will be rotation invariant. Two complementary measures can be described using two-dimensional surface textures. Local spatial patterns and grayscale contrast are the two features developed through the LBP feature descriptor. Formation of the LBP feature descriptor is very simple. From a patch of 3 X 3 pixels, the LBP operator forms labels of image pixels by thresholding the 3X3 neighbourhood of each pixel with the centre value. The obtained value will form a binary number which will concatenate binomially in a clockwise direction. The centre pixel is then assigned with the resultant binary value.

The formation of the LBP code of a pixel $(x_c, y_c)$ is given by:

$$LBP_{P,R}(x_c, y_c) = \sum_{P=0}^{P-1} s(i_p - i_c)2^p \quad (3.1)$$

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (3.2)$$

Here, $i_c$ represents the gray value of the centre pixel $(x_c, y_c)$, $i_p$ is the gray value of its neighbours, P is number of neighbours and R is the radius of the neighbourhood. If it becomes difficult to determine the location of neighbouring pixels, their position is estimated using bilinear interpolation. For each pixel, the LBP values are computed for the entire image along with the histogram to describe the texture of the image. The basic formation of the LBP operator against its neighbouring pixels is illustrated in Figure 3.6.

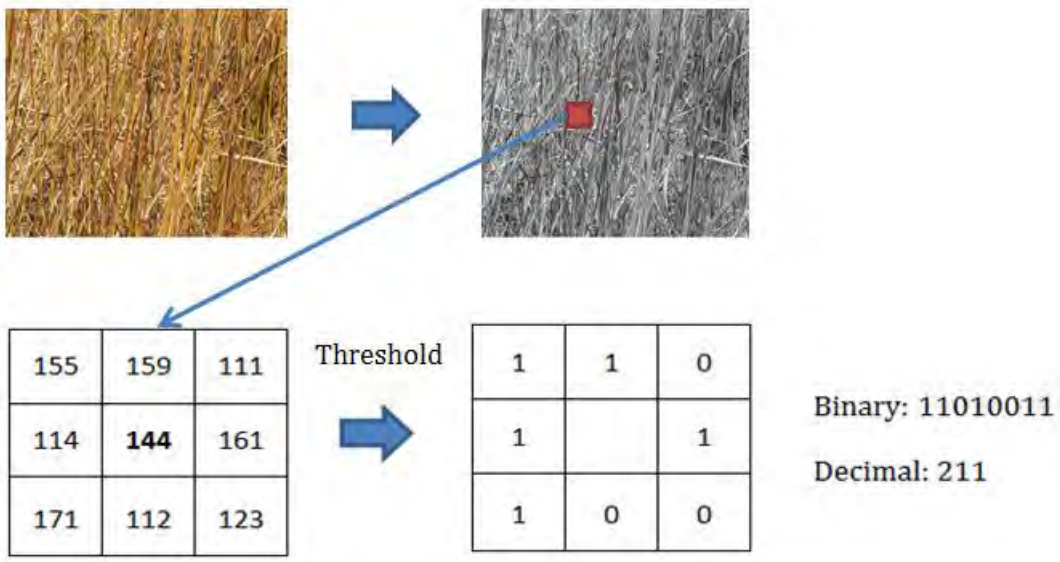



Figure 3.6 Formation of Basic LBP Operator against its Neighbouring Pixels and Result Interpretation as a Binary Number

For example, for an input image I with width and height of M x N pixels, the LBP value is computed for each pixel (x, y) an encoded image representation is obtained. A histogram *H* is then obtained from the encoded image using:

$$H(b) = \sum_{x=1}^{M} \sum_{y=1}^{N} f(LBP_{P,R}(x,y), b), \qquad f(a,b) = \begin{cases} 1, a = b \\ 0, a \neq b \end{cases} \qquad (3.3)$$

Here, b is the LBP code value. The texture information of the image is described using the resulting histogram H. The feature vector is used as an input matrix to compute the GLCM value. If the image patches have different sizes and we need to compare between those images patches, it will be better to make them normalise them for a coherent description using:

$$N_i = \frac{H_i}{\sum_{j=0}^{n-1} H_j} \qquad (3.4)$$

In the next phase, on the extracted LBP feature descriptor the co-occurrence of pattern was calculated. To calculate the Co-occurrence of LBP pattern the concepts of GLCM are used. The calculation of the distribution of co-occurring of values forms a new matrix and known as the co-occurrence matrix. Mathematically, for a given offset $(\Delta x, \Delta y)$ over an image $I$ with a width n, and height m, a new co-occurrence matrix C will be formed using:

$$C_{\Delta_x \Delta_y}(i,j) = \sum_{p=1}^{n} \sum_{q=1}^{m} \begin{cases} 1, & if\ I(p,q) = i\ and\ I(p + \Delta x, q + \Delta y) = j \\ 0, & otherwise \end{cases} \qquad (3.5)$$

In the given image, image intensity values are denoted by $i$ and $j$ and spatial positions are denoted by $p$ and $q$. Depending on the direction $\theta$ offset $(\Delta x, \Delta y)$ value will be determined. As the co-occurrence matrix is sensitive to the offset $(\Delta x, \Delta y)$ parameter, one offset vector is chosen to make the generated matrix rotationally invariant. From the above matrix, the spatial structure of the local texture of a given image can be extracted. The information of adjacency between two pixels can be determined using gray-level co-occurrence matrices. GLCM indicates the frequency of pixels horizontally adjacent with each other in terms of gray-level (gray intensity) value. Within a given window, with distance d and orientation θ are used to calculate the number of co-occurrence for all pixel pairs. Using the above technique, a gray-level co-occurrence matrix for each LBP code is generated from the entire image. To specify the number of gray levels, number levels were used to define the segment level of the matrices. The proposed technique used segment size hundred (100), and the distance between a pixel and its neighbour. Finally, to construct the co-occurrence matrix, the proposed method used the numbers of instances of all possible neighbourhoods. In Figure 3.7 demonstrates an example of gray-level co-occurrence matrix generation from a sample image. In the given figure the last column has no neighbours to the right side. So the number of pixel pairs that need to be computed can be calculated as follows:

$$N_y \text{ X } (N_x - 1) = 5 * (5 - 1) = 20 \qquad (3.6)$$

The co-occurrence matrix is calculated based on the probability. In a horizontal combination, for a specific pattern number of times the outcome occurs divided by the possibility of total outcomes over the image. Mathematically, it can be described by Equation 3.7:

$$P_{i,j} = \frac{V_{i,j}}{\sum_{i,j=0}^{N-1} V_{i,j}} \qquad (3.7)$$

Here i and j denote the row and column number and V denotes the pattern.

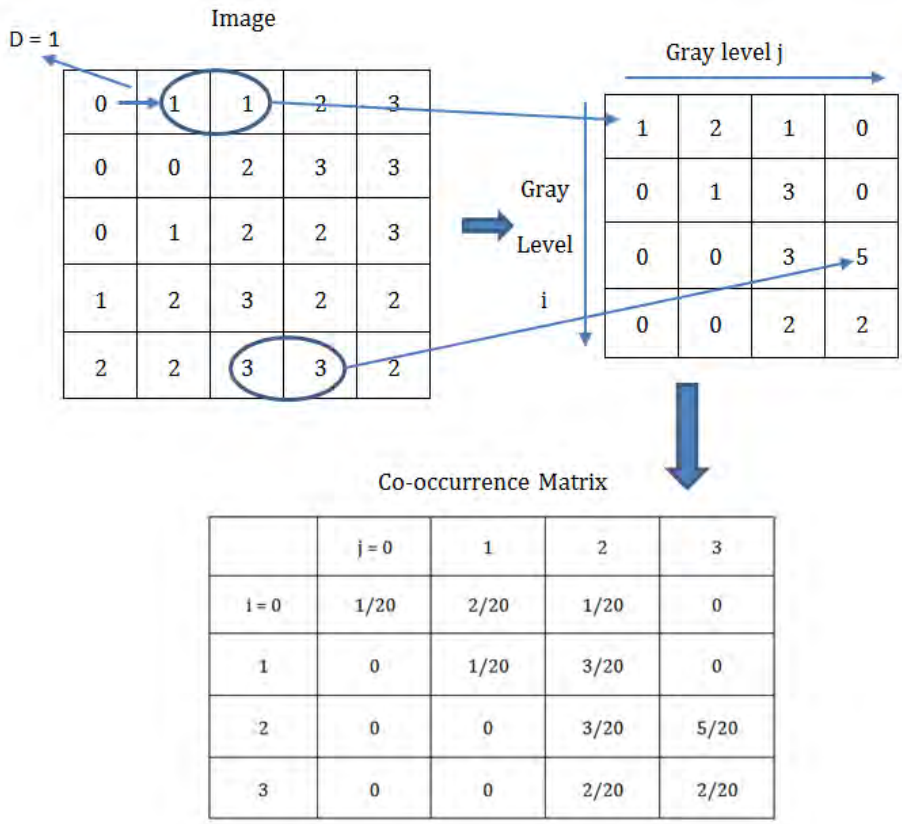



Figure 3.7 Gray-level Co-occurrence Matrix Formulation for Texture Feature Extraction

The idea of the CBP technique is combining the local binary pattern and the gray-level co-occurrence matrix for texture feature extraction. Initially, the local texture feature is extracted using the LBP code and the gray-level co-occurrence matrix concept is then applied on the extracted LBP code to generate the final feature vector. This feature vector was used for training the model and then for classification purposes.

### 3.2.4 Experiments and Results

#### 3.2.4.1 Dataset

To test the efficiency of the proposed feature extraction model, images were collected from real environments, mostly from roadsides in the Central Queensland region. Initially, a small set of images were collected which comprised 110 (one hundred and ten) images where 60 (sixty) images were chosen from dense regions and (50) fifty images were chosen from sparse regions. All the images were collected during daylight and all were colour images.

During dataset collection, various types of grasses were cropped from various locations. This will increase the diversity. All the cropped images were stored in JPEG format with an equal width x height

size of 900 X 500 pixels. The proposed technique was used to classify dense and sparse regions based on their properties. Table 3.1 shows the total images used for evaluation.

Table 3.1 Data for Training and Testing

| | **Dry Grass** | | |
|---|---|---|---|
| | **Dense Grass** | **Sparse Grass** | **Total** |
| Training | 50 | 40 | 90 |
| Testing | 10 | 10 | 20 |
| Total | 110 | | |

Figure 3.8 Sample Image Data for Dense Grass Regions

Figure 3.9 Sample Image Data for Sparse Grass Regions

From Table 3.1, it is clear that for experiment fifty a total of hundred ten (110) images were used where sixty (60) images from dense and fifty (50) from sparse grass were cropped. For training, among the sixty (60) images, forty (40) images were used from the dense region and from the fifty (50) sparse regions forty (40) images were used. For testing the proposed technique a new set of images were used which consists of ten (10) different images from each category. For statistical

analysis and accuracy calculation, fivefold cross-validation was used on the whole dataset. Some sample images that were used in the proposed technique are shown in Figures 3.8 and 3.9.

#### 3.2.4.2 Image Pre-processing

Image pre-processing can significantly increase the reliability of feature extraction. Hence, filtering, scale conversion and resizing were undertaken before feature extraction. Steps are described below.

##### 3.2.4.2.1 Median Filtering

To make the images smooth and noise free median filtering was applied which removes the noise from the images. Median filtering is an effective smoothing technique as it preserves the edges.

##### 3.2.4.2.2 RGB to GRAY Conversion

All original images need to be converted into gray-scale images as the proposed feature extraction technique mostly depends on the grayscale image. The proposed method converted the images into grayscale before the training and feature extraction phase.

##### 3.2.4.2.3 Image Scaling

In order to process the image patches on a standard platform, all the images were normalised and rescaled into a specific size. The initial size of the images was 900 X 500 pixels. The images were normalised into 200 X 200 to reduce the computational time.

#### 3.2.4.3 Ensemble Model

Three classifiers were chosen for investigation to form the ensemble model and the decision was optimised based on majority voting for a final decision. These classifiers were Neural Network (NN), Support Vector Machine (SVM), and k- Nearest Neighbour (k-NN).

##### 3.2.4.3.1 Train and Classify with Support Vector Machine

The SVM classifier is the first choice of classifier for the model to be used to process the separation between the two classes. Some notations are modified here for simplification and different labelling is also used to denote the two classes; instead of using $y_i \in \{0,1\}$ as appears elsewhere, $y_i \in \{1,2\}$ is used to denote the class labels for training. Here two different labelling was used to denote two classes. Dense grass belongs to class "1" and sparse grass belongs to class "2". Other parameters w, b with the vector $\theta$ will be used for the classifier. In this process, let S $=\{(x_{i,}\, y_i)\,|x_i \in R^n\}$ $and\ labeled\ with\ \ y_i \in$ {1,2}. To map the training data into kernel space, the kernel function "svmtrain" was used. To choose the best kernel that will fit with the problem, linear, polynomial and radial basis functions are investigated. After several analyses with training and test accuracy, the linear kernel function was chosen for training and classification purposes. The linear function can be defined using Equation 3.8:

$$f(x) = w^T x + b \quad (3.8)$$

Here, f(x) = 1 if $(w^T x + b) \geq 0$, and f(x) = 2 otherwise. For training, Matlab's "svmtrain" was used, and the "SVM classify" function was used for classification.

#### 3.2.4.3.2 Train and Classify with Neural Network

Feed forward neural network is the second choice proposed for the ensemble model. Generally, the network consists of three layers which are: an input layer, a hidden layer, and an output layer. The input layer consists of the feature vector and the output layer consists of target label. Here, u= $[u_{1,}u_{2,}u_{3,} \dots u_p]^T$ constitutes the input layer with the extracted feature while y= $[y_{1,}y_{2,}y_{3,} \dots y_m]^T$ is the label that denotes the output vector. Here, p and m denotes the number of elements and number of classes respectively. For this implementation awe used p value of 110, and m =2 were used. For proper training and achieving the best classification accuracy, various numbers of hidden neurons, and different numbers of iterations were used and finally best combination was finally chosen. The network repeatedly trained until it reached the root mean square or reached to the total number of epochs. For training purposes, the proposed technique used back propagation. For training proposed technique used one hundred ten images as input matrix. For target matrix generation, the first sixty images treated were from as dense class and were denoted as one ('1'); and rest of the images treated as sparse grass and were denoted as ('2'). Different parameter for hidden neurons and epochs were used for training and finally best parameter was selected for testing. For validation, the whole dataset was divided into five fold and each time 88 images were used for training and the remaining 22 images were used for testing, and finally the average was taken to calculate the overall accuracy.

#### 3.2.4.3.3 Train and Classify with k-Nearest Neighbour

The k-Nearest Neighbour (k-NN) classifier was chosen as the final classifier for the proposed ensemble model. It shows promising performance in general pattern recognition tasks. In k-NN, objects are classified based on the closeness to the feature space. k and distance metric are the only parameters needed to be tuned for k-NN implementation. Success depends mostly on the selection of k values. The basic principle of selecting the k value is that it should be an odd number which will be helpful to avoid draw voting. The best value of k will be determined by the classification performance. For analysis purposes, different k values e.g. k = 5, 7 and 9 were used. The same k values were used for both training and testing phases. Furthermore, in the proposed system both the Standard Euclidean distance metric and City Block distance metric were used to calculate the distance. It was observed that, in both cases, results slightly varied.

The major limitation of k-NN is the number of samples required for training. If one class dominates in the training phase, there is more possibility of misclassifications occurring in the test phase. To avoid this problem in the proposed method tried to make the sample numbers equal in each class.

#### 3.2.4.3.4 Combine the Classifiers

To achieve the best overall accuracy from the three classifiers, weighted majority voting [107] [108] was introduced. The basic idea of majority voting is very simple; a majority decision will win the competition. If two classifiers produce the same class, we assign that label for the corresponding test image.

#### 3.2.4.3.5 Calculate Accuracy

The formula used for the accuracy calculation for the proposed ensemble technique and individual classifiers can be described as follows.

$$Accuracy = \frac{correctly\ classified\ images}{total\ number\ of\ images} * 100 \quad (3.9)$$

### 3.2.5 Result Analysis

The efficiency of the proposed method relies on two steps. In the first step, best parameters from each classifier were chosen. In the second step, best parameters were chosen and results were integrated for final classification results. For robustness of the system, fivefold cross-validation was applied. As it is a binary classification, output from the classifier will have only two values. If the dense region is identified, it will return "1"; otherwise, it will return "2" for a sparse region. There were some misclassifications with dense regions misclassified as "2" and vice versa with sparse regions misclassified as "1". In the fivefold cross-validation, four-fold data were used as training and the data from the other fold were used as testing. The overall classification accuracy was calculated by taking the average. In this section, classification accuracy was analysed. The dataset used for this experiment is described in the experimental setup portion (Section 3.2.4). The classification results obtained from individual classifiers and from using the hybrid technique are shown in with statistical results in tables and graphical comparisons in figures.

Results obtained using different kernel functions for the SVM classifier are presented in Figure 3.10. Both train and test accuracies are listed and showing the reason for choosing the best kernel function for the ensemble model. From the Figure 3.10, it is seen that the linear kernel function surprisingly shows much better performance than both radial basis and polynomial function.

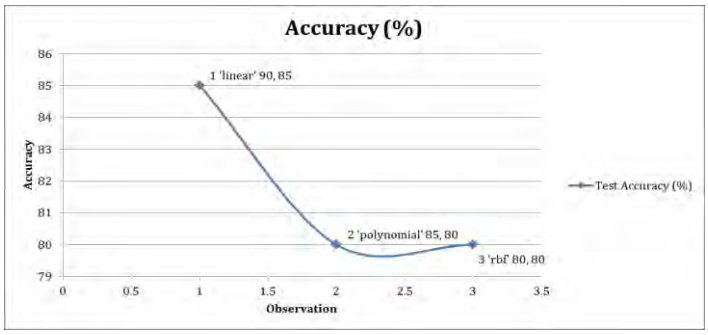



Figure 3.10 Results using SVM

Using the linear kernel function achieved 90% accuracy for training and 85% for testing. While test accuracy for the other kernel functions was almost equal, there was a variation on training accuracy for both kernels. Proposed method achieved 85% accuracy for training and 80% for testing using the polynomial function. On the other hand, proposed method achieved 80% accuracy for both sets using the radial basis function. Based on these results, it was decided to use the linear kernel function for the SVM classifier and its fivefold cross-validation accuracy calculation outcome is shown in Figure 3.11. Here accuracy for six observations was listed where the first three shows each fold accuracy and the last one is for the hybrid classifier.

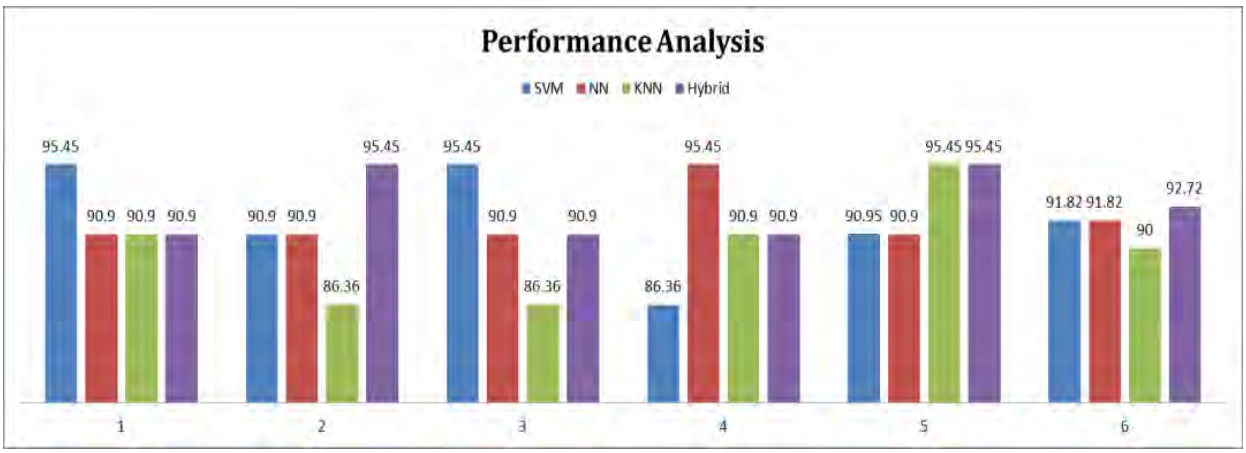



Figure 3.11 Fivefold Cross Validation Accuracy for Each Classifier

The results using the neural network with different parameters are shown in Table 3.2. Initially, started with a random parameter for hidden units, number of iterations, and root mean square and proceed based on accuracy. There are various options for changing parameters so we fixed hidden neurons and changed the number of iterations and RMS error and listed the training and testing accuracy. A larger number of iterations generally increase accuracy, but it also depends on hidden neurons. After several observations we get the best classification performance with number of hidden units=12, learning rate=0.01, epochs=3500, momentum=0.15 and Root Mean Square error=0.0001. We got 90% accuracy for training accuracy and 85% for testing accuracy. From the results, we observed that, if the proper parameter is chosen for the neural network, we achieved similar accuracy as for SVM. Figure 3.11 shows the individual fold accuracy along with the ensemble classification accuracy.

Table 3.2 Results using Neural Network

| Exp# | Hidden Units | Iterations | RMS Error | Accuracy on Train Data (%) | Accuracy on Test Data (%) |
|---|---|---|---|---|---|
| 1 | 6 | 500 | 0.0003 | 80.00 | 75.00 |
| | | 1000 | 0.0004 | 75.00 | 75.00 |
| | | 3500 | 0.0001 | 80.00 | 75.00 |
| 2 | 10 | 500 | 0.0005 | 80.00 | 80.00 |
| | | 1000 | 0.0001 | 80.00 | 80.00 |
| | | 3500 | 0.0003 | 85.00 | 85.00 |
| 3 | 12 | 500 | 0.0001 | 85.00 | 80.00 |
| | | 1000 | 0.0002 | 85.00 | 85.00 |
| | | **3500** | **0.0001** | **90**.00 | **85**.00 |
| 4 | 15 | 500 | 0.0002 | 85.00 | 80.00 |
| | | 1000 | 0.0003 | 90.00 | 80.00 |
| | | 3500 | 0.0001 | 90.00 | 80.00 |
| 5 | 20 | 500 | 0.0004 | 85.00 | 80.00 |
| | | 1000 | 0.0003 | 85.00 | 80.00 |
| | | 3500 | 0.0002 | 85.00 | 80.00 |

Results obtained using k-NN is shown in Figure 3.12. Different k values were chosen to check the performance and based on the accuracy best parameters were chosen. We conducted three analyses for three different k values and results are shown in Figure 3.12. From the figure, it is clear that we obtained the highest accuracy 85% for train and 80% for a test when k value equal 7. Compared to other classifiers k-NN shows lower accuracy. In respect to other classifier the rate is acceptable. We achieved 85% for training and 80% for testing.

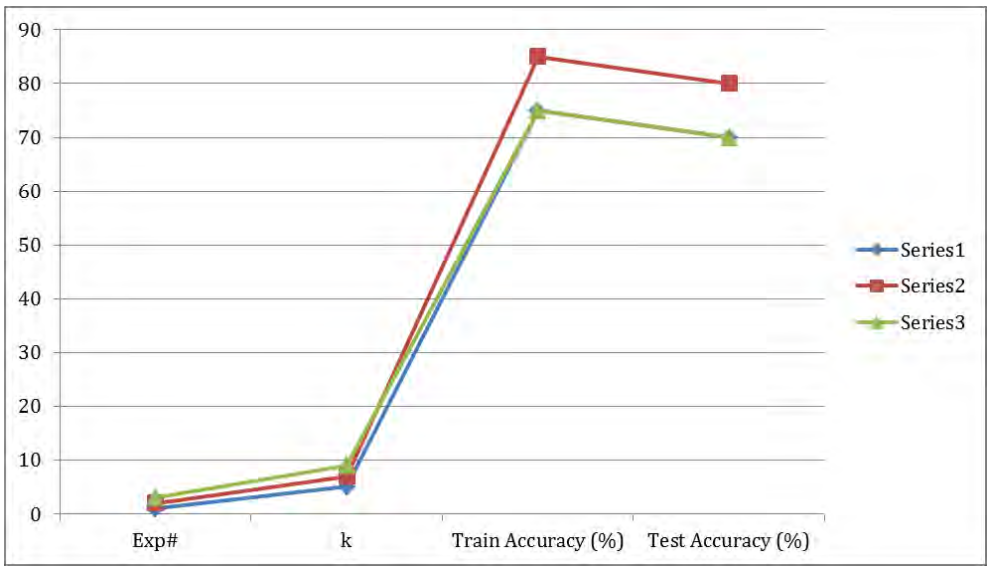



Figure 3.12 Results using k-NN

After analysing the individual classifier performance we combined the best parameters selected from individual classifiers and the results of the resulting hybrid ensemble model are summarised in Table 3.3. From the table, it is clear that, for various combinations, we observe lots of variation in accuracy. We achieved the optimum accuracy outcome by choosing the linear kernel function for SVM, a k value of 7 for k-NN, and setting the number of the hidden units as 12 and epochs as 3500 for Neural Network.

Figure 3.13 shows the performance comparison chart of the proposed hybrid technique along with the individual classifiers. The hybrid technique achieves 92.72% accuracy which is significantly higher than any individual classifier.

Although good accuracy was achieved with the best parameters, some misclassifications still occurred as shown in Figures 3.14 and 3.15. Table 3.4 shows the number of misclassified images in respect to total images for the proposed hybrid technique and the individual classifiers and the resulting success rate percentages.

Table 3.3 Results using Proposed Hybrid Technique

| Exp# | SVM Parameter | ANN Parameters | *k*-NN Parameter | Accuracy on Train Data (%) | Accuracy on Test Data (%) |
|---|---|---|---|---|---|
| 1 | Linear | H.U= 12, Iterations= 3500 | 7 | 95.00 | 90.00 |
| | polynomial | H.U= 10, Iterations= 3500 | 5 | 85.00 | 80.00 |
| | Rbf | H.U= 15, Iterations= 1000 | 9 | 80.00 | 75.00 |
| 2 | Linear | H.U= 10, Iterations= 3500 | 9 | 85.00 | 80.00 |
| | polynomial | H.U= 15, Iterations= 1000 | 5 | 80.00 | 75.00 |
| | Rbf | H.U= 12, Iterations= 3500 | 7 | 85.00 | 80.00 |
| 3 | Linear | H.U= 15, Iterations= 1000 | 5 | 80.00 | 75.00 |
| | polynomial | H.U= 12, Iterations= 3500 | 9 | 80.00 | 80.00 |
| | Rbf | H.U= 10, Iterations= 3500 | 7 | 80.00 | 75.00 |

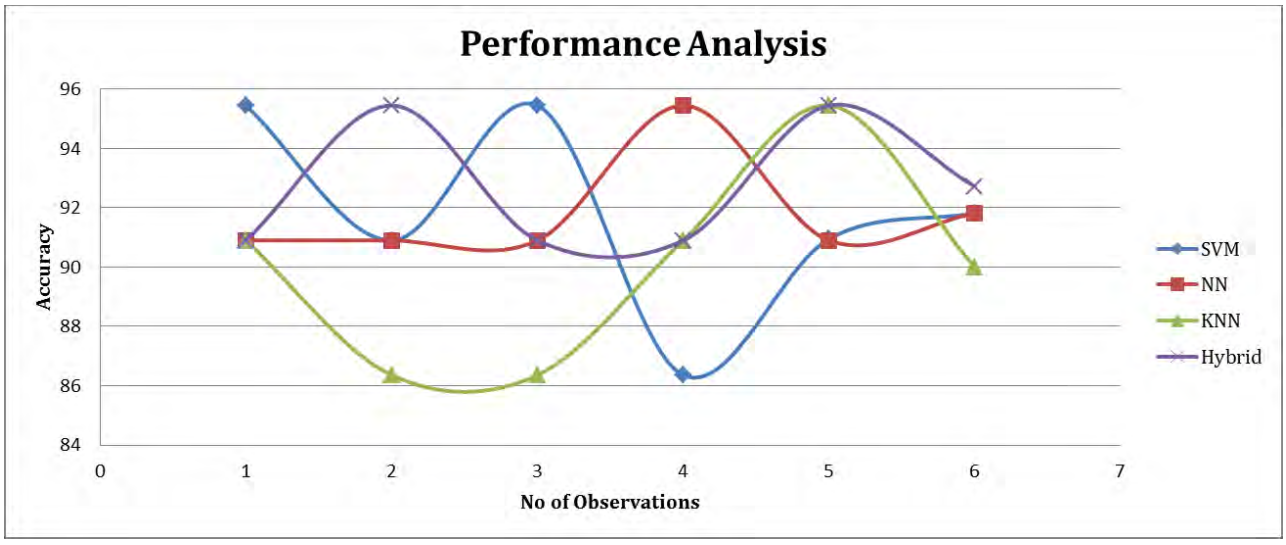



Figure 3.13 Performance Analysis of Individual Classifiers and Hybrid Technique

Figure 3.14 Sparse Grass Misclassified as Dense Grass

Figure 3.15 Dense Grass Misclassified as Sparse Grass

The reason for misclassification is showing high texture for the sparse region. If the grass region is very low, but if the density high, it creates high texture and the possibility of misclassification arises. Table 3.4 shows the comparison results for different classifiers versus hybrid classifier.

Table 3.4 Comparisons Chart for Classification Performance for Different Classifiers

| **Method** | **Number of Samples** | **Number of Misclassified Samples** | **Success Rates (%)** |
|---|---|---|---|
| SVM Classifier | 110 | 9 | 91.82 |
| NN Classifier | 110 | 9 | 91.82 |
| *k*-NN Classifier | 110 | 11 | 90.00 |
| Proposed Hybrid Classifier | 110 | 8 | 92.72 |

To prove the effectiveness of the proposed system as being statistically significant, an ANOVA (Analysis of Variance) was also conducted. Tables 3.5 and 3.6 show the statistical summaries using the ANOVA analysis.

Table 3.5 Single Factor ANOVA Summaries

| Group | Count | Sum | Average | Variance |
|---|---|---|---|---|
| SVM | 5 | 459.11 | 91.822 | 14.44217 |
| NN | 5 | 459.05 | 91.81 | 4.1405 |
| *k*-NN | 5 | 449.97 | 89.994 | 14.45538 |
| Hybrid | 5 | 463.6 | 92.72 | 6.21075 |

We set a hypothesis to compare the performance of the proposed ensemble technique with the NN, SVM, and k-NN classifiers in terms of classification accuracy (H1). We set the null hypothesis as follows:

*There is no significant difference between individual classifier accuracy in respect to hybrid technique classification accuracy ($H_0$).*

Table 3.6 ANOVA Analysis Details

| ANOVA: SINGLE FACTOR | | | | | | |
|---|---|---|---|---|---|---|
| Group Variation | | | | | | |
| **Source of Variation** | **SS** | **df** | **MS** | **F** | **P-value** | **F-Crit** |
| **Between** | 19.6314 | 3 | 6.543818 | 0.6669 | 0.584529 | 3.2388 |
| **Within** | 156.9952 | 16 | 9.8122 | | | |
| | | | | | | |
| **Total** | 176.6267 | 19 | | | | |

The alternative hypothesis will be proven if it can be shown that there is a significant difference between the classification accuracies $H_0$ and $H_1$. From the ANOVA analysis we need to determine the p-value which will show whether or not there is strong evidence that there is a significant difference between the classification accuracies.

From Table 3.6, we can analyse the statistical significance between the proposed hybrid technique accuracy and individual classifier accuracy. It is clear that the p-value obtained is greater than the critical value which proves the significance of the batch effect. So we can reject the null hypothesis and confirm that the classification accuracy obtained using the proposed hybrid technique is statistically significant.

### 3.2.6 Summary

This technique shows an approach for classifying dense and sparse grasses. A new texture feature extraction technique is proposed which shows a significant improvement over existing techniques. Initially, the original image was converted into a grayscale image and a binary pattern was obtained using histogram equalisation against its neighbouring pixels which interprets the result as a decimal number. To calculate the co-occurrence of binary pattern, the GLCM technique was applied and a texture feature was generated. Finally, performance was evaluated using a hybrid classification technique with respect to the individual classifiers.

For experimental purposes, a real dataset was used where all the images were collected from Central Queensland region. Resulting analysis shows that using the proposed technique can achieve 92% accuracy. All classification performances conducted were validated using fivefold cross-validation. The proposed method was also statistically significant as proven using the ANOVA test.

There are some limitations of the proposed method. It considers only part of the image, not the whole image. Sometimes it is difficult to judge as, if it works on a small scenario, it may still fail in the larger scenario. Moreover, the method will not work on any other images, e.g. satellite images. While doing the experiments we did not consider any adverse weather conditions and ignored shadows and rainy conditions. In the real world, so many other situations will arise and the currently proposed model cannot handle all those situations.

The proposed technique cannot classify all objects from the roadside scene. It can only differentiate cropped dense and sparse regions. This can be usable as a part of identifying the fire-prone regions. Now we need to focus more on object identification from the scene and we need to separate the grass regions from those identified objects. Once the separated grass regions are obtained, we can apply the CBP technique to identify dense and sparse regions. In the real roadside scenario, rather than just grass, several complex objects will be found. Using only a small portion of data will not give enough evidence and adequate performance when experiments need to be conducted on a large dataset. Furthermore, thorough analysis is required to add more texture features to make the system sufficiently robust. The application area can be extended and can be used in other environment sensing technology. Moreover, we can apply the technique on satellite images and compare the performance.

## 3.3 Distance and Cross Correlation (DCC) Based Feature Extraction Technique

This section presents another feature extraction technique [109] which is an extension of previous work. We add two new proposed features and improve the overall accuracy. From the previous analysis, we identified that the major problems are in differentiating regions of grasses due to the similar spectral signature. A simple example of the new proposed method is shown in Figure 3.16. To evaluate the performance of the new proposed feature we used the same dataset collected as set out in the image acquisition part. For this new feature extraction, work was undertaken on several colour channels and good performance was obtained using the YCbCr model. Before moving to the feature extraction phase, all images are rescaled to 384 X 384 pixels. After feature extraction, an effective classifier is required to discriminate between the types of grass. This study helps to identify suitable features for vegetation classification which will be helpful in most agricultural research fields. In the future, the proposed technique can be applied for an automatic weeding strategy.

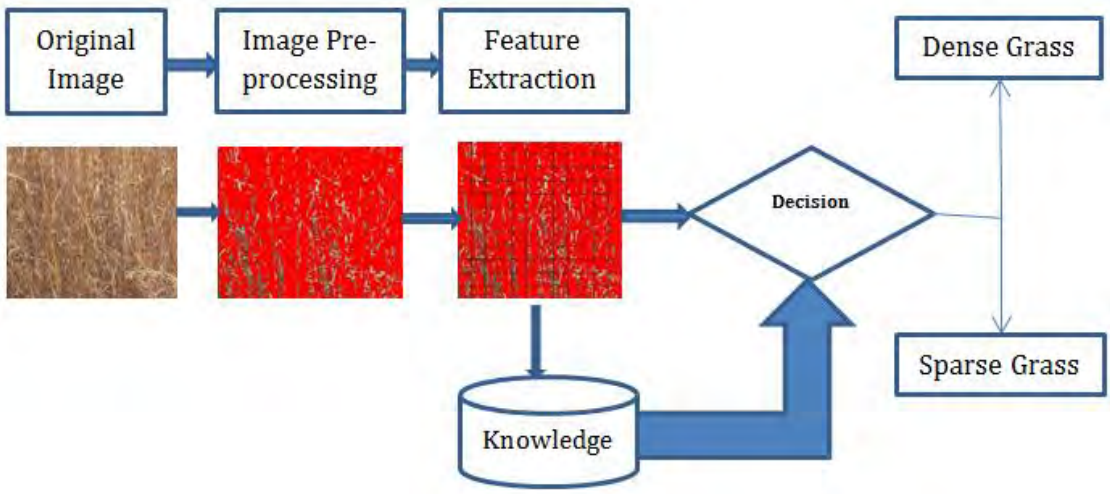



Figure 3.16 Vegetation Classification Flow Chart

### 3.3.1 Introduction

The ultimate goal of the automatic machine vision based approach is to minimise the man-hours of human involvement. If people are doing the same repetitive job for a long period of time, they will feel bored and there will be a high possibility of making mistakes. In a machine vision based approach it is usually necessary to train a dataset based on some features and to later classify test sets efficiently. At the present time, machine vision based intelligent systems have gained popularity for numerous applications on various engineering and scientific applications. However, success depends on appropriate feature extraction. Although we have seen the success in digital and face recognition, still some application areas remain relatively unexplored. One such application area is the vegetation classification sector. The area is become an ongoing research area due to its sensitive application. Roadside bushfires become a frequent issue during the dry season in Australia. Policy makers desire

an optimal solution which can automatically identify fire-prone regions with minimal effort and cost-effectively. Although we have seen some applications from satellite image tries to find out the fire-prone regions but still cannot give an exact location and appropriate information. This research will help to identify various objects from roadside images. The task is not easy as outdoor scenes are sensitive to various environmental issues. Some of these scenarios are presented in Figures 3.8 and 3.9. Vegetation articulation, multiple viewing points, difficult lighting, intra-class variations, soil region identification, and varying seasonal appearance are common issues we need to consider during vegetation classification.

### 3.3.2 Proposed DCC Technique

The whole process for feature vector generation is presented in Figure 3.17. Here the main focus is on creating two new feature vectors which are distance and cross-correlation feature. To generate those feature vectors we need to do some pre-processing steps which are shown in the figure. These feature vectors, along with previously studied feature vectors, will jointly be used for training the classifier and later applied on test images for further classification.

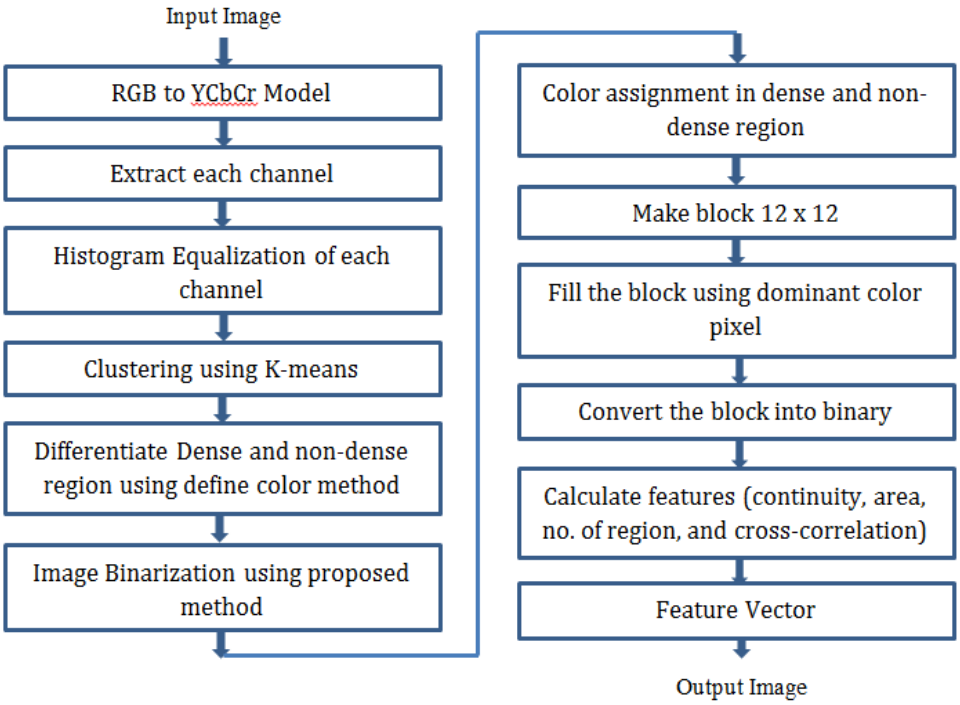



Figure 3.17 Feature Extraction Technique

#### 3.3.2.1 RGB to YCbCr Conversion

The world currently uses several kinds of image colour channel such as RGB, YIQ, HIS, YCbCr, L*a*b*. Feature extraction and classification greatly depends on the appropriate colour space. Working with only the RGB image will not provide appropriate information about the context of grass images. As grasses in the real environment do not have any specific colour so using only RGB colour images will

not be effective. Moreover, using grayscale images will also fail in providing adequate decisions in foreground and background (soil region) separation. To differentiate grass and the non-grass regions as well as the gap between the pixels, YCbCr colour space has been chosen as it will help with extracting the intensity or brightness (luminance) and colour difference (chrominance) information of the image. The conversion used to change the colour space is shown using Equation 3.10:

$$[Y\ Cb\ Cr] = [R\ G\ B]\begin{bmatrix} 0.299 & -0.169 & 0.4998 \\ 0.587 & -0.332 & -0.419 \\ 0.114 & -0.501 & -0.081 \end{bmatrix} \quad (3.10)$$

#### 3.3.2.2 Extract Channel Information

As there are three different channels in YCbCr colour space we need to extract luminance (Y) and chrominance (Cb and Cr) colour values as columns. Each row represents corresponding colour value for RGB colour space where a range of each channel is different where Y is in the range [16 235], and Cb and Cr are in the range [16 240]. To extract the appropriate image information it is necessary to enhance the contrasts of each channel.

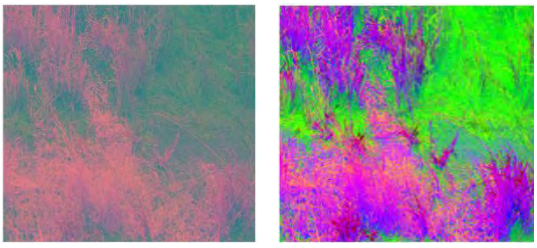
#### 3.3.2.3 Histogram Equalisation

Figure 3.18 shows the importance of enhancing the contrast of an image using histogram equalisation. Different objects with visual similarity will be presented by different highlighted colour which will be easily differentiable from the YCbCr enhanced image. The equation used for histogram equalisation is as follows:

$$g_{i,j} = floor\left((L-1)\sum_{n=0}^{f_{i,j}} P_n\right) \quad (3.11)$$

where $$P_n = \frac{number\ of\ pixels\ with\ intensity\ n}{total\ number\ of\ pixels}\ n = 0, 1, \ldots, L-1.$$

Here, f is the candidate image and g represents the histogram equalised image.



a b

Figure 3.18 a) YCbCr Image b) YCbCr Image after Histogram Equalisation

#### 3.3.2.4 K-means Clustering

After enhancing the image to separate the different colour channels, we use k-means clustering where three different clusters represent the presence of three different types of objects. We consider the dominating pixels as grass region pixels and non-dominating pixels as non-grass region pixels.

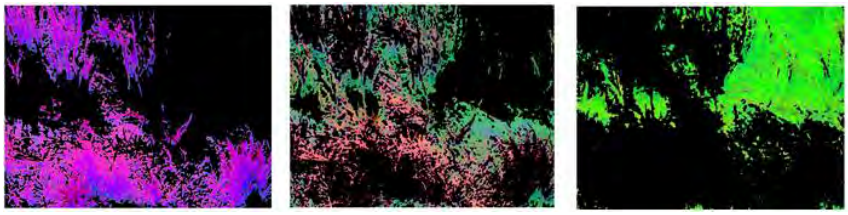


Figure 3.19 K-means Clustering

Figure 3.19 shows the example result of k-means clustering. To compute the clusters among different distance parameter techniques, squared Euclidean distance was used.

#### 3.3.2.5 Differentiate Regions

In order to differentiate between dominating and non-dominating pixel regions, we need to consider pixel value colour information. We need to choose some colour range from chromaticity colour information. For convenience and after thorough analysis, we set some ranges as shown in Table 3.7. While choosing the range of colour, we tried to choose those pixel colours which will remain the same in any image. This makes the calculations easier during further processing and producing the binary image for feature extraction.

Table 3.7 Pixel Value Assign

| Pixel Value | Value Assign |
|---|---|
| 0-63 | 0 |
| 64-127 | 1 |
| 128-191 | 2 |
| 192-255 | 3 |

#### 3.3.2.6 Image Binarisation

To create the binarised image we have not followed any existing technique. We developed a new approach based on the combination which is shown in Table 3.8. The table shows, from among the 64 combinations, only for those combinations to which was assigned the value zero (0). All other combinations were set to one (1). Using the above-mentioned procedure we obtained the binary image which will help to extract a further feature from the image.

Table 3.8 Dense and Non-dense Separations

| Value 1 | Value 2 | Value 3 | Assigned Value |
|---|---|---|---|
| 0 | 3 | 0 | 0 |
| 0 | 3 | 1 | 0 |
| 1 | 3 | 0 | 0 |
| 1 | 3 | 1 | 0 |
| 2 | 3 | 0 | 0 |
| 2 | 3 | 1 | 0 |
| 3 | 3 | 0 | 0 |
| 3 | 3 | 1 | 0 |

#### 3.3.2.7 Colour Assignment

To check the efficiency of the regions separation, we tried to assign a different colour to different regions and compare the performance visually in respect to the original RGB image. We tried to colour the grass regions using red colour with all other regions remaining in their original RGB colour. From the pictorial analysis, we achieved better performance. Hence, technique is applied for further processing of feature extraction.

#### 3.3.2.8 Blocking

The next step was block size selection for feature extraction from each block. To capture useful texture information selection of block size plays a vital role in differentiating dense and sparse grass. However, if the block size becomes too small it will not provide useful information: on the other hand, if the block size is too big it will be overlapped with other information. By considering these facts, we choose a block size for the whole image of 12 X 12. Each block contains 32 X 32 pixel values and contains useful information about the image.

#### 3.3.2.9 Block Filling

In order to finalise the block value, we checked the dominating pixel values. The final value will be one (1) if the total number of non-zero pixels is greater than zero pixels and vice versa.

#### 3.3.2.10 Block Minimisation

To determine the final block value among the 32X 32-pixel values we considered the dominating pixel values and assign one (1) if all are ones and assign zero (0) if all are zeros. From the 384 X 384 pixel values we minimised these into 12X12 pixels. The generated matrix using the proposed block minimisation technique contains the texture information for the whole image.

### 3.3.3 Feature Extraction

To distinguish between different object categories, extracting the appropriate feature plays a vital role. In the previous approach, we used pixel-based features; here we introduce the patch-based feature. The combination of both features forms a new foundation of robustness. The colour is one of the widest features for pixel-level image segmentation and classification, but choosing a suitable colour space is still a challenging task. One principle is that the colour space should be (approximately) uniform to human colour perception (i.e. equal distances in the colour space correspond to equal colour differences perceived by humans). This is because humans are very adept at distinguishing different types of objects with no difficulty using solely colour information, and thus the segmented results should be consistent with the understanding of humans. Therefore, we choose the CIE Lab colour space, which has been proved as having high consistency with human vision perception and successfully applied in many studies. In addition, we also include the R, G, B colour channels to compensate for the possible information lost in the Lab space. For a pixel at the coordinates in an image, its corresponding 6-dimension pixel-based features are:

$$F1^{I}_{x,y} = [R, G, B, L\ , a, b, Y, Cb, Cr]$$

Patch based features are extracted by taking information from neighbouring pixels. It is increasingly agreed that spatial texture information in neighbouring pixels plays an essential role in object recognition, particularly for real-world applications. We extract patch-based features based on Local Binary Pattern and Gray Label Co-occurrence, which are capable of encoding both shape and colour information, being scaling and rotation invariant, and having high robustness against changing lighting conditions. Moreover, we included two new features extracted using the new proposed technique.

$$F2^{I}_{x,y} = [CBP, Distance, Area, Continuity, Crosscoorelation\ Score]$$

The final feature vector will be the combination of both features which can be presented as:

$$F = [F1^{I}_{x,y}, F2^{I}_{x,y}]$$

Calculation of distance feature and cross-correlation feature are described below:

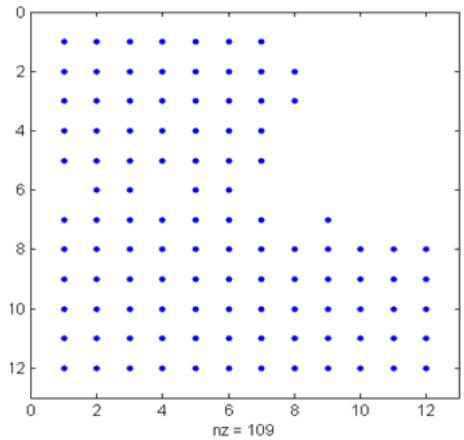
#### 3.3.3.1 Distance Feature

From the 12 X 12 block, distance can be calculated by summing the distance of each block value from the whole patch. Initially, we proceed with a 2-by-2 block and consider its neighbourhood and calculate the distance. Different types were considered and corresponding distance values are listed:

1. If zero for all neighbourhood pixels, assign value for distance = 0
2. If the block contains one-pixel value “1”, assign value for distance = 1/4
3. If the block contains two adjacent pixels value “1”, assign value for distance = 1/2
4. If the block contains two diagonal pixels values “1”, assign value for distance = 3/4
5. If the block contains with three-pixel values “1”, assign value for distance = 7/8
6. If the block contains with all four-pixel values “1”, assign value for distance = 1

#### 3.3.3.2 Cross-Correlation Score

Figure 3.20 shows a cross-correlation score calculation from a patch. The overall procedure is very simple. Initially, we check column-wise block value: for each column we will assign either ‘D’ to represent dense or ‘S’ to represent sparse. If within a column the block value contains more than 30% zero values, we consider as sparse and assign ‘S’ for that particular column. Otherwise, we will assign ‘D’ to represent dense. The similar procedure follows for the rest of the columns and then applied on each row. The final cross-correlation score will be determined by summing the individual score.




| | D | D | D | D | D | D | D | S | S | S | S | S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| D | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| D | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| D | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| D | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

Figure 3.20 Cross-Correlation Score Calculation

### 3.3.4 Experiments and Results

After feature extraction, the most crucial part is appropriate classifier selection and proper training. We tested with the different classifiers and got good performance using Support Vector Machine (SVM). SVM is as a good classifier as it can construct a hyper plane between two classes and make the classes separable. This hyper plane can be used as a decision maker to discriminate between the two classes. Let A= {($x_i$, $y_i$), i=1, 2, . . m} be the set of training samples, where $x_i$ € $R_p$. The labelling for the dataset can be defined as $y_i$ € {1, 2}. For classification of the test set data, x can be defined as:

$$f(x) = sign(\sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b) \quad (3.12)$$

Among the kernels for the classification, we choose the Radial Basis Function (RBF) kernel. The kernel function K can be expressed as:

$$K(xi, x) = \exp(-\frac{\|xi-x\|2}{2\sigma^2}) \quad (3.13)$$

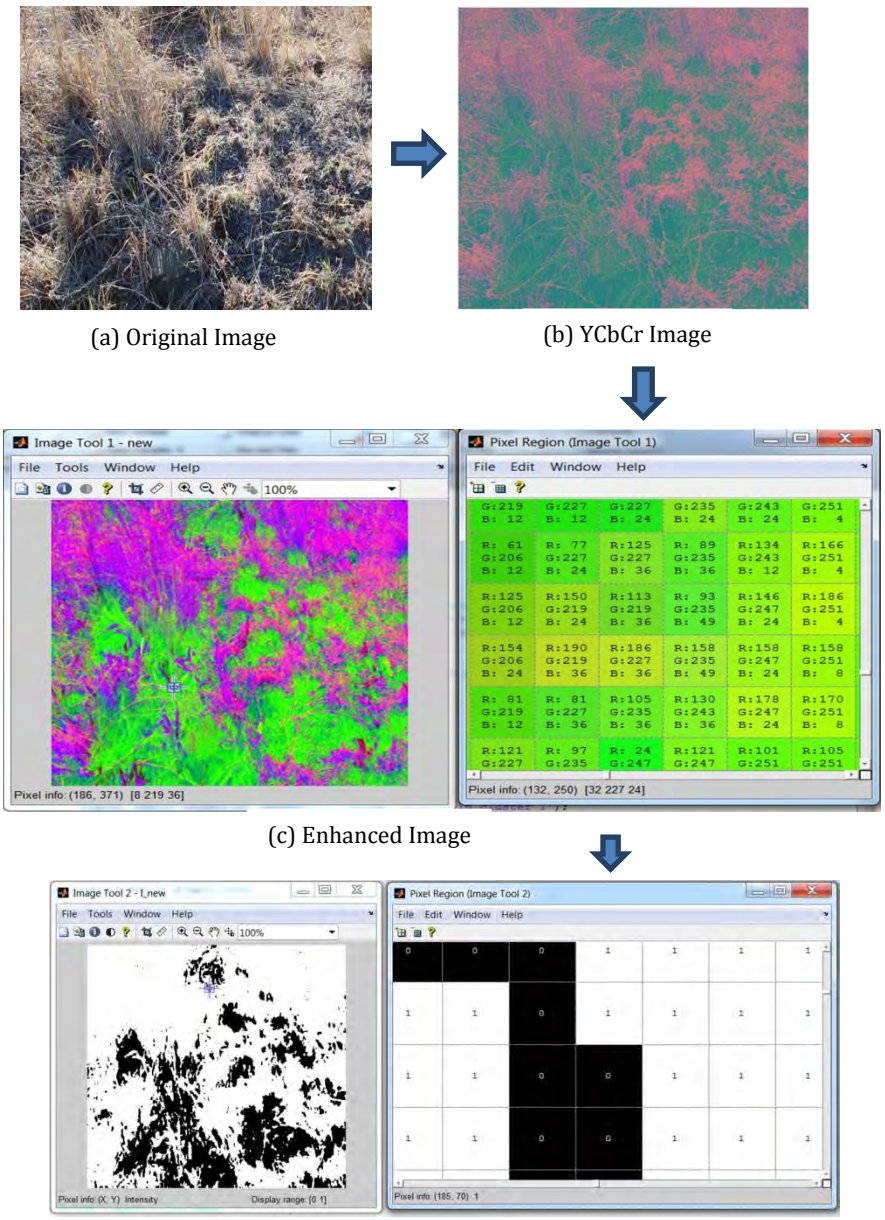
Here $\sigma$ is a kernel parameter and distance can be calculated using Euclidean distance algorithm and can be calculated using $||xi-x||^2$. For training we use the Matlab built in function "svmtrain", and for classification we use 'svmclassify'. During training we put both feature vectors together for all one hundred and ten images. For better understanding, we use numeric values '1' and '2' for the training levels.

### 3.3.5 Result Analysis

In order to validate the performance of the above algorithm, an experiment using the same dataset described in Section 3.2.4.1 is conducted. Table 3.9 shows the result comparison between the two methods. Figure 3.21 shows the experimental output for the proposed technique. Initially, the original image was converted into an YCbCr image. The latter image was enhanced using histogram equalisation. Based on the proposed range, the enhanced image was binarised to obtain the target image.

Table 3.9 Result Comparison

| Serial No | Approach | Pixel-wise Accuracy (%) |
|---|---|---|
| 1 | Hybrid Classifier [95] | 92.72 % |
| 2 | Proposed DCC Technique [109] | 93.00 % |




(a) Original Image

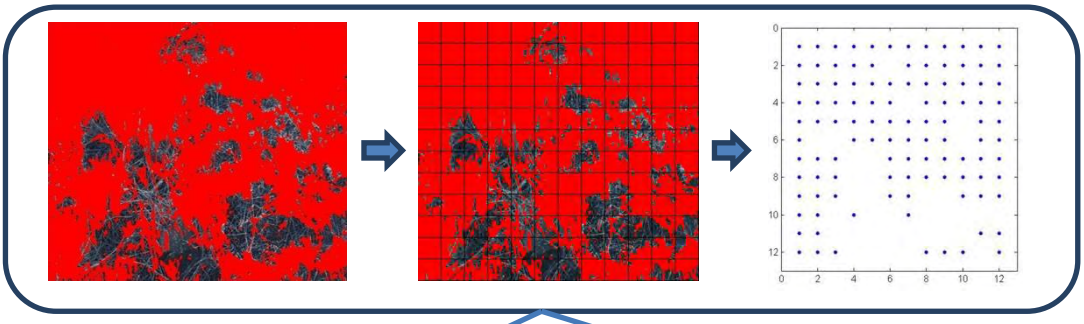
(b) YCbCr Image

(c) Enhanced Image

(d) Binary Image

Figure 3.21 Overview of the Proposed Technique: (a) Original Image (b) Converted YCbCr Image (c) Enhanced Image (d) Binary Image

From the binary image, we found isolated pixels which can distinguish different pixel regions. Then we convert the image into the blocks and determine the block values. Figure 3.22 shows a sample example of dense and sparse region separation and the dense and sparse regions are clearly visible from the captured area. Finally, feature vectors were extracted from the block. Figure 3.23 shows some more experimental results from different images. From the figure, we can assume that the recognition rate improves due to adding the new features within the feature vector.



(a) Distance Feature

| | D | D | D | S | S | S | S | D | S | S | D | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| D | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| D | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| D | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| D | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| D | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| D | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| D | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(b) Cross-correlation Feature

Figure 3:22 Feature Vector Extraction from the Block

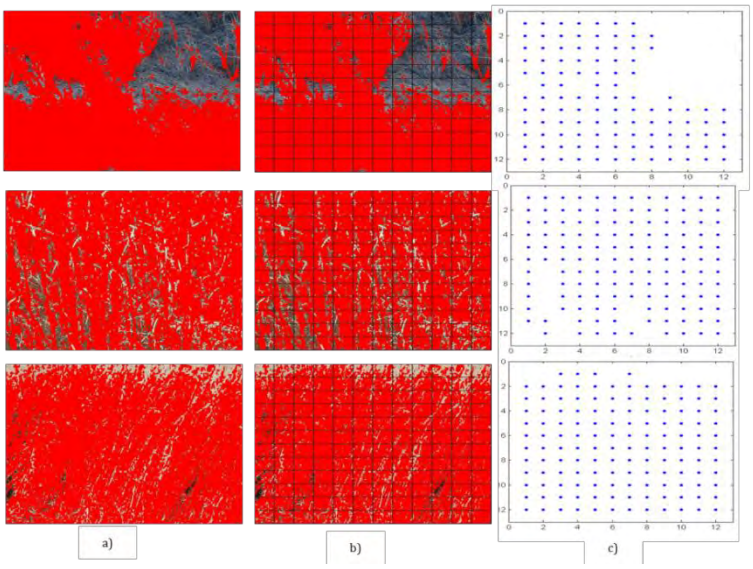



Figure 3.23 Block Value Calculation

### 3.3.6 Summary

The task of grass classification is challenging as grass has no specific shape, colour and sometimes exhibits overlaps in images. To overcome these difficulties, a new feature extraction technique for grass classification has been presented which is helpful to solve the complex problem. Evaluation on the test dataset shows that the proposed feature extraction technique, along with an SVM classifier, shows promising performance. Further investigation is required towards making the technique more robust against illumination variability.

## 3.4 Quantisation Feature and Neural Network (QFNN) Based Feature Extraction Technique

This section presents the new feature extraction technique for roadside object classification. Initially, existing features were investigated and performance was recorded. Further investigation on the misclassified objects helped to propose new features which can overcome the challenges. Two new techniques with an effective classifier were introduced.

### 3.4.1 Introduction

Relevant works on roadside object detection [110] [111] [112] were reviewed and existing problems were identified. In [113], the authors mentioned the procedure of classifying vegetation and non-vegetation from the structured environment. The idea was extended by another author and incorporates the idea of the fusion of multiple features. This idea was described in [114] and, for their fusion process they used colour, texture, and 3D distribution information. Although the idea is promising, it is not practically suitable due to its high processing time. Moreover, environments and sensors affect the overall performance of the proposed feature vector [110]. Recently, vegetation classification shows promising performance on navigation. Due to its high performance, it can also be used in vegetation management. In [115] and [116], the authors proposed a method for grass detection from video data using a combination of texture and colour features. Both papers used different types of strategies in order to detect the grass regions. The authors of [115] used a multi-scale texture analysis based adaptive colour and positioning model while the authors of [116] used a probability-based colour and texture feature model. The similarity between the two papers is the use of YUV colour space in their proposed colour model. The image was enhanced by changing the brightness of the colour.

A recent improvement on vegetation classification is the use of features in the visible spectrum. One limitation of the visible spectrum is that it cannot differentiate between two different objects with similar colour. For example, it fails to differentiate between a green car and tree leaves under various lighting conditions. Use of the invisible spectrum can solve the above-mentioned problem. The invisible spectrum has been used in remote sensing techniques for vegetation area identification. One such invisible spectrum is the NDVI (Normalised Difference Vegetation Index) [114] [117]. The invisible spectrum was used to identify the presence of chlorophyll within the vegetation. Visible spectral reflectance (VIS) and near-infrared regions (NIR) were used for NDVI calculation. According to published vegetation research, vegetation areas will be dense and healthy if their NIR value is high and VIS value is low. If the NIR value is low and VIS value is high, the vegetation will be sparse. Nguyen *et al.* [114] used the concept of NDVI for vegetation detection. In their proposed method they used the combination of vegetation indices along with colour and texture features. To collect more information from the images, LiDAR and 3D scanner data was also used. Bradley *et al.* [118] indicated that the data

collected from satellites is different from ground image data. Shadow, shininess and underexposure effects are common problems in the ground images, while satellite images are free from those effects. To overcome that above-mentioned problem, Nguyen *et al.* proposed a modified vegetation index MNDVI [119] which considers NIR intensity and colour information as a feature by adaptive learning. From the above discussion, we can summarise the situation as follows. Appropriate feature selection [120] is the key factor for successful vegetation classification. The idea of using probabilistic superpixels and Markov random field [121] helps with segmenting crops from the field under natural illumination conditions. The concept of the proposed method was based on the assumption that colour starts changing from the highlighted area to the non-highlighted area. This information will be useful on extracting crop regions from shadow regions. The method proposed in this thesis will be helpful for scene labeling and scene recognition [122] applications. The proposed method can also be applied on intelligence transport systems [123] as well as in weed identification [59] [124]. Furthermore, it can apply for change detection in remote sensing images [125].

### 3.4.2 Proposed QFNN Based Technique

The proposed quantisation and neural network (QFNN) based technique involves two new ideas. Incorporating the idea of new feature vector generation is one of the new concepts which are described in Section 3.4.3. And the idea of introducing the radial basis function in the hidden layer also adds a new dimension in vegetation classification which is described in Section 3.4.5.2. These two new concepts make the task easier for efficient vegetation classification from roadside video images. Finally, to improve the overall accuracy, a post-filtering technique was introduced which is also a new idea. It helps the proposed method to detect incorrect predictions and restore the correct prediction. As we are dealing with each pixel during training and classification, there is always a possibility for the wrong prediction of one pixel. This post filtering technique helps to remove those false predictions. Eight-bit colour information from both RGB and HSB channels were used to build the new feature vector. Not only the individual pixel information but also their pixel differences were considered. The original contribution of this research is introducing the idea of Most Significant Bit (MSB) quantisation with colour channel sequence into the feature vector. The proposed feature vector was tested on a roadside vegetation dataset collected from different parts of Queensland and the experimental results show that the new feature vector and a radial basis activation function in the neural network helped to achieve state-of-the-art performance. We also do some comparison with some proposed method and it shows an improvement in results over the benchmark dataset.

### 3.4.3 Feature Extraction

The new feature vector is an extension of the previous pixel characteristic based feature extraction technique published by the thesis author and others in [126]. To increase the overall accuracy and make the system compatible, the new feature has been introduced along with an existing feature which will help to improve the overall accuracy. For a particular cropped region I, the feature vector for a pixel p from image I can be calculated as follows:

$$\alpha_p = \left[R, G, B, |R-G|, |R-B|, |G-B|, \frac{1}{3}(R+G+B), MSB\ pattern, sequence, H, S, V\right]$$

The complete feature vector can be expressed as:

$$F = \alpha_p$$

If a feature can be extracted from a class and identified as a pattern which will be the same and followed for all similar classes, then that feature is termed compatible. The proposed method uses both RGB and HSV colour images for finding the pattern among the classes. At the very beginning, we extract the colour information in exactly the same way as for the previous technique. It carries the R, G, and B colour information from each pixel p, hence the initial feature vector can be expressed as $\alpha_p\{p \in \{R, G, B\}\}$. The feature vector can then be enriched by adding the absolute difference between each colour channel which later helps to decide the corresponding class K for a new pixel. The gray scale information can also be used by converting the colour channel using a simple calculation of $\frac{1}{3}(R+G+B)$. Figure 3.24 shows the feature extraction generation technique pictorially.

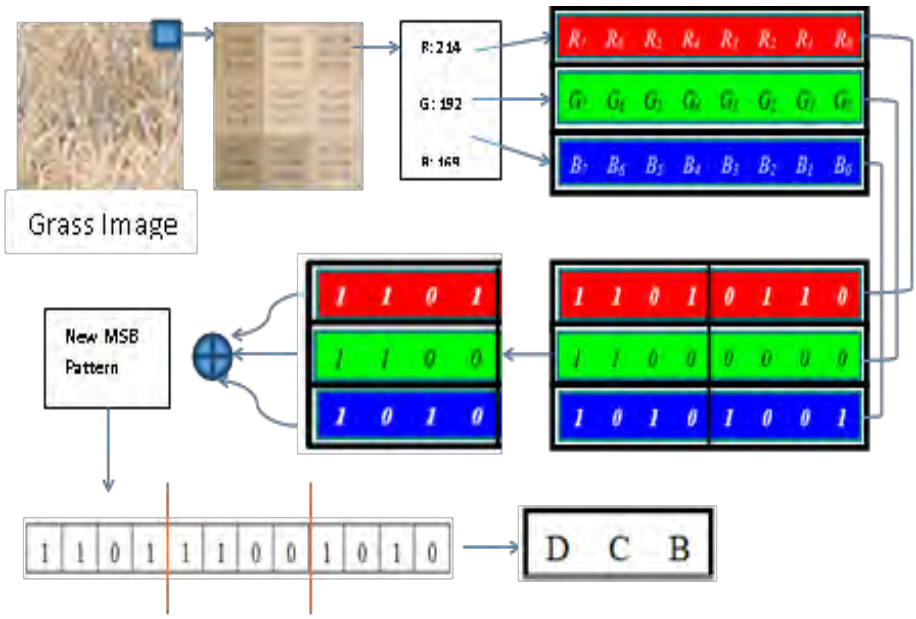



Figure 3.24 Most Significant Bit (MSB) Pattern Generation Technique

The novelty of the new feature vector generation technique is the introduction of the Most Significant Bits (MSB) quantisation and colour sequence generation. This will be useful for generating

a common bit pattern for a similar class. To do that it is necessary to initially convert all the decimal values of R, G, and B into a binary value using 8-bit code (128, 64, 32, 16, 8, 4, 2, 1, 0). Here each colour channel is represented by 8-bit code using the following order C ∈ {$C_7$ ….. $C_0$} where C ∈ $\{R, G, B\}$. The most significant bit is defined by $C_7$ to $C_4$ notation and the lowest significant bit is defined by $C_3$ to $C_0$ notation. Now each binary value for each colour channel is converted into a hexadecimal value by considering only the most significant bit values. The overall calculation can be mathematically expressed as follows:

$$\sum_{i=8,4,2,1,\ \ k=7,6,5,4} C_k * i\ \ MSB_R, \quad \sum_{i=8,4,2,1,\ \ k=7,6,5,4} C_k * i\ \ MSB_G\ , \quad \sum_{i=8,4,2,1,\ \ k=7,6,5,4} C_k * i\ MSB_B$$

The next contribution of the proposed feature vector is the inclusion of colour channel sequence. This research on cropped image regions has shown that, in some image regions, pixel difference is showing same as we are taking the absolute difference. Moreover, from the observations, it is clear that colour channel sequence is varied among different image regions. This is one of the key issues for showing the visual dissimilarity between different image regions. Usually, the similar colour sequence will be found between similar types of objects. For the proposed method a code was designed to represent the colour channel sequence. As three colour channel information is being used, there will be six different combinations within the sequence. For the RGB colour channel, to represent R we used the value 1, for G we used the value 2 and for B we used the value 3. Hence, for a colour sequence that followed {R>G>B}, according to proposed method the generated code will be 123. To construct the code, it is necessary to initially sort the R, G and B values with its index using a sorting algorithm. We achieved the sorted value like this $[sorted_{value}, index] = \{sort(A), here\ A \in\ R, G \text{ and } B\}$. The HSV colour information is also included. The reason for adding the HSV colour information is to minimise the error against illumination condition. Common equations that were used for this conversion process can be described as follows:

$$R' = R/255,\ G' = G/255,\ B' = B/255;$$

$$T_{max} = \max(R', G', B')$$

$$T_{min} = \min(R', G', B')$$

$$\Delta = T_{max} - T_{min}$$

Hue calculation:

$$Hue = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G' - B'}{\Delta}\ mod6\right), & \text{Tmax} = \text{R}' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2\right), & \text{Tmax} = \text{G}' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4\right), & \text{Tmax} = \text{B}' \end{cases}$$

Saturation calculation:

$$\text{Saturation} = \begin{cases} 0\,, \text{Tmax} = 0 \\ \frac{\Delta}{\text{Cmax}}\,, \ \text{Tmax} \neq 0 \end{cases}$$

Value calculation:

$$\text{Value} = C_{max}$$

The feature vector was constructed using the pixel information from the HSV colour information with the following equation set:

$$H' = H*255,\ S' = S*255,\ V' = V*255.$$

Figure 3.25 shows the procedure of feature vector generation pictorially.

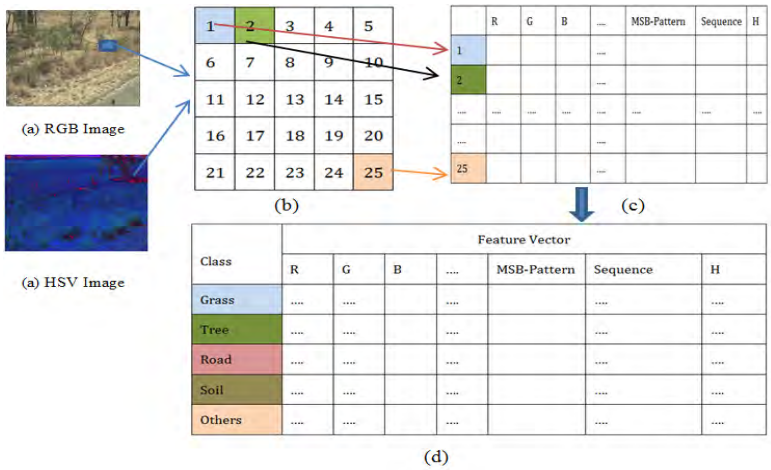



Figure 3.25 Feature Vector Extraction Technique

### 3.4.4 Post-Processing Technique

For the proposed method of dealing with pixel-wise classification, there is a possibility of misclassification of one pixel around the surrounding pixels. This will degrade the overall performance and classification accuracy. The reason is that, by considering only individual pixels, some of the pixels seem the same with different objects e.g. tree stumps with the road, green grass with leaf etc. Two strategies are used to solve this issue. The first strategy is to use the concept of location of a pixel and the second strategy considers the probability of neighbourhood pixels. It is obvious and natural that sky pixels cannot be located on the bottom of an image and road and soil pixels cannot be located at the top of the image. This strategy will remove some confusion. An example of post processing is presented in Figure 3.26.

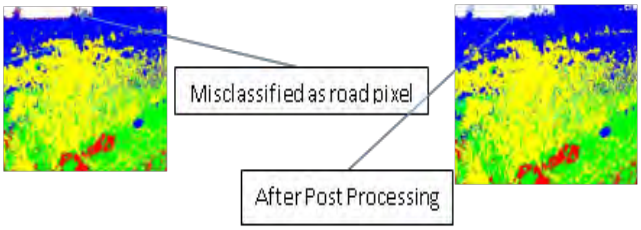



Figure 3.26 Post Processing of Pixels

The first strategy helps to overcome some common mistakes and overall accuracy is improved. The misclassification issues were further investigated and another interesting concept was introduced which helps to achieve a state of the art performance. The new filtering idea of neighbourhood pixel localisation can be described as follows.

Figure 3.27 Post Processing of Neighbourhood Pixels

The concept of the superpixel was used to extract neighbourhood pixel information. Both the classified image and the original image with superpixel are available. Now from the original image, we proceed with each superpixel and determined the classification value from the classified image. It was expected that within a superpixel, all values should be classified as the same class. So we start checking the corresponding class within each superpixel and, if a variation was found among the class information, majority voting was applied. The idea is shown in Figure 3.27. Here within a block, if the majority says it belongs to class 2 it should be class 2, and any other class will be replaced by the dominating class value.

### 3.4.5 Experiments and Results

#### 3.4.5.1 Data Collection

The data used for the experiment was taken from the locally created dataset [127]. The data was collected by the industry partner and it was collected from the real field. The strategy of data collection was not new as many existing methods used cameras mounted over a vehicle for data collection [128]. But the ways of cameras are set up and their position selection was varied depending

on the applications. The local technique used four different cameras in four different positions. The vehicle was driven all over rural areas of Queensland to collect data. For data collection, they put the camera in all four directions. The front camera was used to capture the front view and it indicates the overall scenario of the road. A left facing camera indicates the actual situations of grass, trees, road and soil positions and roadside vegetation risk can be determined from the left camera information. The right facing camera indicates conditions on the opposite side of the road and gives an overall impression about the road. The rear camera gives a similar view to that of the front camera. As the main concern is with vegetation classification, the left camera information was of most interest. Collected data was video data and, for analysis purposes, these were converted into the frames. There are lots of variation among the collected data and location-wise vegetation information also varied. The entire collection of frames has the same resolution (1632x1238) and fifty (50) regions were cropped for each class from 10,000 images.

#### 3.4.5.2 Training and Classification of Feature Vector

Neural networks have been successfully used in pattern recognition problems for many years. A set of the feature vector is needed along with the target class to build the overall neural network [129] architecture. The first layer contains the input feature vector, and in second layer, we need to determine the hidden layer which may vary depending on the the specific problem being examined as well as particular choiches made by the user. The last layer contains the target class. In most cases, classification accuracy depends on the number of hidden neurons and the number of iterations needed to train the model properly. Selection of training algorithm and activation function also has a great impact on classification accuracy. As the problem here is a multiclass classification problem, it is necessary to design an output matrix according to the number of classes. Eleven different features wre extracted which will be used for the input matrix and six classes as output matrix. Many neural networks currently exist and show promising performance in different applications. In all cases, based on the problem complexity, researchers have proposed different types of activation function [130].

To solve the vegetation classification problem the radial basis activation function was used which would solve the problem. The strategy of the radial basis function is that it computes the similarity between the test set input and a prototype vector. This prototype vector will be taken from the training set. The output value will be high if the similarity is high and returns a value close to one (1). The Gaussian function was used to measure the similarity. The radial basis activation function can be expressed as:

$$\varphi(X) = e^{-\beta||x-\mu||^2} \quad (3.14)$$

Here $\mu$ is the prototype vector which is at the centre of the curve.

Figure 3.28 shows the overview of the neural network architecture that consists of the input layer, hidden layer, and an output layer. It also shows the activation function in different layers. The

activation functions in both layers are different. The hidden layer used the radial basis activation function while the output layer used the softmax activation function. The softmax function can be described using the following equation:

$$\sigma\,(X) = \frac{e^{X}}{\sum_{k=1}^{K} e^{X}}\ for\ j = 1, \ldots K \quad (3.15)$$

To train the overall network a two layer feed forward neural network was used. To reach the target and get a compatible accuracy we needed to change the number of epochs and the RMS error value in an iterative fashion.

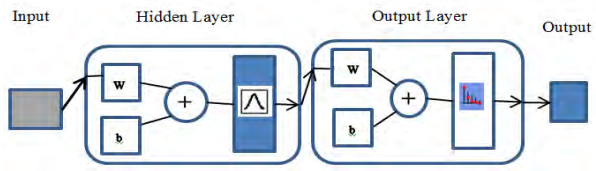



Figure 3.28 Architecture of Neural Network

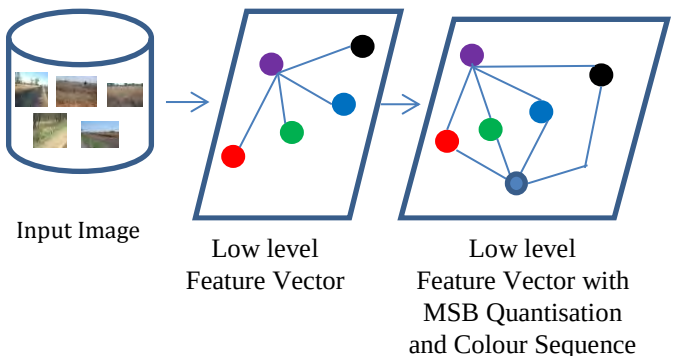



Figure 3.29 Feature Vector Formation Technique

Figure 3.29 shows the overall feature vector formation technique. Later 50 regions were cropped from each class and the proposed technique had L = {l1, l2 ....l50} cropped regions. Later, for each cropped region, the feature vector was extracted which can be denoted by F. For each cropped region there should be a corresponding class and denoted by K = {K1, K2 ... K6}. Let T be the total dataset which consists of all the training cropped images. The overall training procedure is described in Algorithm 1. Appropriate functions were used along with different parameters to train the network.

**Algorithm 1** Training Feature Vector

***Inputs***: *Dataset of low-level feature with MSB quantisation and colour sequence features*

$$\alpha_p = \left[R, G, B, |R-G|, |R-B|, |G-B|, \frac{1}{3}\,(R+G+B), MSB\ pattern, sequence, H, S, V\right]$$

```
 for i=1...T
        image(i) = database(i);
        levelling(i) = GroundTruth(i);
      for each pixel p(i ,j ) do
          extract feature vector α_p
          extract ground truth
         if GroundTruth(i ,j)==k
              Feature_Vector(k) ← [Feature_Vector(k);  α_p];
       end if
     end for
end for
train_feature_vector(F_V,label)
   network = patternnet(50);
   network.trainParam.showCommandLine = true;
    network.divideFcn ←'dividerand';
    network.trainParam.mc  ← 0.15;
     network.performFcn ← 'mse';
    network.trainParam.goal ← 0.001;
    network.trainParam.lr ←  0.10;
     network.trainParam.epochs ← 1000;
     network.trainFcn ←  'traingdm';
    network.layers{1}.transferFcn ←  'radbas';
    network.layers{2}.transferFcn ←  softmax;
[neuralnetwork t,tr] ←  train(network,double(input),double(target));
save('neural_nn.mat', neuralnetwork);
end
```

### 3.4.5.3 Results

To validate the performance of the proposed technique, the dataset was trained using the cropped region and tested on scene images collected during data collection. The strategy of training and testing is different compared to existing methods. The dataset has been uploaded in Google drive and made public so that anyone can use the dataset and compare the performance using their proposed method. During training of the feature vector, the performance of overall training was observed and the final performance curve is shown in Figure 3.30. Figure 3.31 shows the training confusion matrix after final epochs. The overall accuracy is 89% which is good enough for a trained network. The class-wise accuracy achieved more than 90% accuracy for some classes. Two classes show almost 80% accuracy and one class shows 70% accuracy. The accuracy dropped due to the similarity between green grass pixel information and tree leaf information as they have the same property of pixel colour.

As soon as training finished we applied the trained network on the test image and the overall performance was recorded and is shown in Table 3.9. From the table, we can conclude that the proposed novel quantisation feature and neural network based technique (QFNN) shows promising performance on overall object classification. Figures 3.32 and 3.33 show some experimental results with respect to ground truth images. From the visual representation, it is obvious that the proposed feature extraction technique performs satisfactorily.

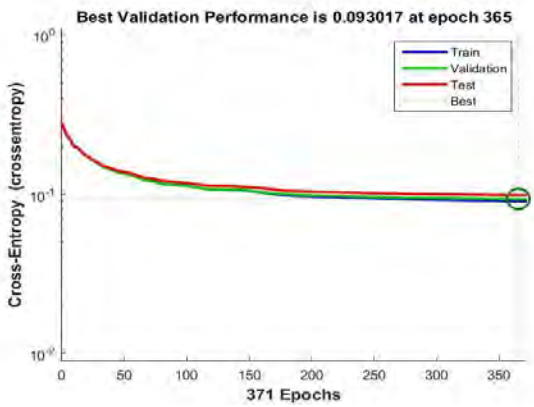



Figure 3.30 Performance Curve during Training Phase

Figure 3.31 Confusion Matrix after Training of Each Class

Two video frames are shown in Figure 3.32 to present the classification result pictorially. For a better understanding of the different classes, we used a different colour code for each object. Although the classification results give outputs as numerical values, different colour codes have been assigned here for different numerical values. Figure 3.32 shows the colour codes for four of the classes. Six different classes were used in the experiment and six different colours for each class. Tree leaf and tree stump are represented with blue colour, while brown and green grass used yellow and green colour respectively. The sky region is displayed with white colour and the road with red colour. Cyan was used for soil colour.

Table 3.10 Performances (%) on the Annotated Dataset

| **Type** | **Overall** | **Sky** | **Tree (Leafs and Tree Stump)** | **Grass (Brown and Green)** | **Road** | **Soil** |
|---|---|---|---|---|---|---|
| Pixel-wise | 78.01 | 93.29 | 80.5 | 67.84 | 73.09 | 75.32 |

Although 7 different cropped regions were extracted for a better understanding of the objects, the final accuracy calculation only used six different classes. Brown and green grasses are merged and make up the grass class, and tree stump and tree leafs are merged into the tree class. Some experimental outputs along with annotated dataset and predicted output are shown in Figure 3.32 for visualisation.

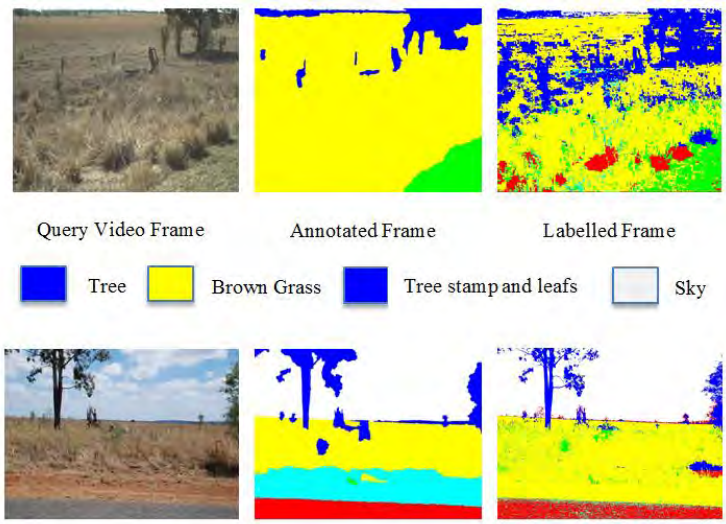



Figure 3.32 Overview of Scene Labelling on the Original Frame

In Figure 3.32, the left column shows the original images which need to be classified. The middle column with the label "Annotated Frame" represents the corresponding ground truth which is done manually before testing the proposed system. The right column with the label "Labelled Frame" shows the corresponding outputs from the proposed system. Manual annotation takes a long time as the image size is bigger than any existing dataset. This is why a small test dataset was used for validation.

But performance of the test set shows that the proposed method can classify any image from the collected video within seconds.

Some misclassifications occurred due to intra-class variations. Further investigation is needed to improve the overall accuracy. Figure 3.33 shows some more experimental results along with the corresponding ground truths and final outputs. Some misclassifications between soil and road can also be seen within the images.

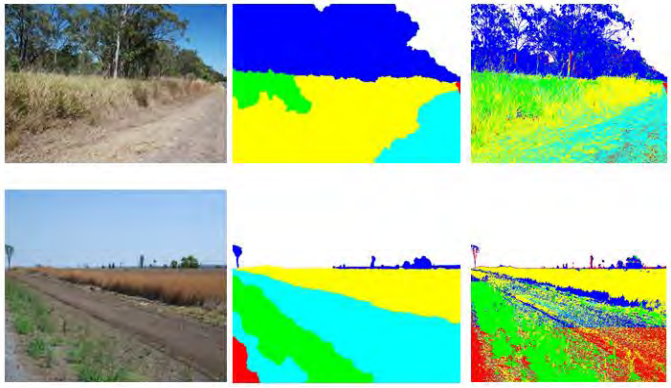


Figure 3.33 Experimental Results

### 3.4.6 Result Analysis

Table 3.10 shows the overall confusion matrix between the classes. For accuracy calculation within the class, the ratio of correctly classified pixels versus total pixels was used and can be expressed using the following Equation 3.16:

$$A_c = \frac{\sum_j Arr_{i,j}}{\sum_j \sum_k Arr_{j,k}} \qquad (3.16)$$

Table 3.11 Confusion Matrix within the Class

| | **Sky** | **Tree** | **Grass** | **Road** | **Soil** |
|---|---|---|---|---|---|
| **Sky** | **93.29** | 1.32 | 2.80 | 1.51 | 1.08 |
| **Tree** | 1.00 | **80.51** | 13.26 | 3.52 | 2.50 |
| **Grass** | 0.99 | 23.15 | **67.84** | 3.52 | 4.50 |
| **Road** | 2.16 | 6.52 | 7.53 | **73.09** | 10.70 |
| **Soil** | 0.2 | 5.38 | 6.56 | 12.54 | **75.32** |

#### 3.4.6.1 Comparative Analysis

Table 3.11 shows the comparison results of the proposed technique with existing techniques.

Table 3.12 Result Comparison

| Serial No | Approach | Pixel-wise Accuracy (%) |
|---|---|---|
| 1 | Colour Feature [109] | 70.47 |
| 2 | Texture Feature [131] | 73.30 |
| 3 | Colour Texture Feature [131] | 74.50 |
| 4 | Colour Feature with Proposed Neural Network | 72.36 |
| 5 | Proposed Feature with Feed Forward Neural Network | 74.34 |
| 6 | **Proposed Quantisation Feature and Neural Network (QFNN) based approach** | **78.01** |

Some techniques to check the performance in respect to the proposed technique were also developed. For example, experiments using standard colour and texture features with a feed forward neural network and a feed forward neural network with radial basis activation function on the same dataset were conducted to validate the results.

### 3.4.7 Summary

This section has presented the overall scenario of the proposed QFNN based technique. Two new feature vectors were introduced along with an existing feature vector to improve the overall vegetation classification performance. The ideas of the Most Significant Bit Sequence and colour channel sequence were included to give a good pattern for individual class differentiation. The idea of a post processing technique also improved the overall accuracy and was able to reduce the misclassifications of class. Moreover, comparison analysis with existing methods shows the improvement of the overall classification accuracy of the proposed technique. One limitation of the proposed technique is that experiments were conducted on a small dataset and considered only a few classes from the roadside. Further investigation is also needed to test whether the proposed method works robustly under certain environmental conditions. All the images that were used for training and validation were taken under natural daytime light and with good lighting conditions.

# Chapter 4 Directional Connectivity Feature Extraction Technique

This chapter presents a novel and effective approach to solve the problem of grass density estimation in categories of dense, sparse, and moderate from video data collected using 2D car-mounted video cameras. The approach can easily be scaled to datasets with thousands of frames from a video and can assign a category for each frame. First, colour feature sets are extracted from training images that are mostly similar to the query images. Second, a network is trained using that colour feature set. Pixel labelling is then performed on the query images using the trained network and the segmented grass region is divided into window regions. Finally, the proposed contextual feature (i.e. grass pixel vertical directional connectivity) is extracted and the appropriate grass density category is determined. The novel contributions of this process are: 1) directional connectivity feature extraction technique to estimate grass density; and 2) machine learning based classification technique to improve grass segmentation accuracy. The proposed approach has been evaluated on images with ground truths collected from a field survey and images from Transport and Main Roads video data. The experimental results show satisfactory performance with higher accuracy than human observation and promising performance on a real-life video dataset.

## 4.1 Introduction

A directional change in image intensity/colour can be calculated by gradient calculation. To illustrate this, Let's consider a pixel I(a, b) with 3 X 3 neighbourhood pixels (as shown by the diagram below), individual grass pixels are defined by I = { $I_{1:n}^{s}$ }s € S for the set of pixel observations, , the direction in which the intensity values are changing is defined by K={$k_s$ } s € S and Θ represents the model of appearance along the vertical direction.

The first order derivative in the horizontal x direction can be computed as:

$$\frac{\partial I\,(a,b)}{\partial a} = (\frac{1}{2}\,(I\big((a,b+1) - I(a,b)\big) + (I(a+1,b+1) - I(a+1,b))) \quad (4.1)$$

| | | |
|---|---|---|
| (a-1, b-1) | (a-1, b) | (a-1, b+1) |
| (a, b-1) | (a, b) → | (a, b+1) |
| (a+1, b-1) | (a+1, b) | (a+1,b+1) |

Similarly, the first order derivative in the vertical y direction can be computed as:

$$\frac{\partial I\,(a,b)}{\partial b} = (\frac{1}{2}\,(I\big((a+1,b) - I(a,b)\big) + (I(a+1,b+1) - I(a,b+1))) \qquad (4.2)$$

The direction in which way the pixel intensity is changing can be calculated by following:

$$\Theta = \tan^{-1}(\frac{\partial I\,(a,b)}{\partial b} / \frac{\partial I\,(a,b)}{\partial a}) \qquad (4.3)$$

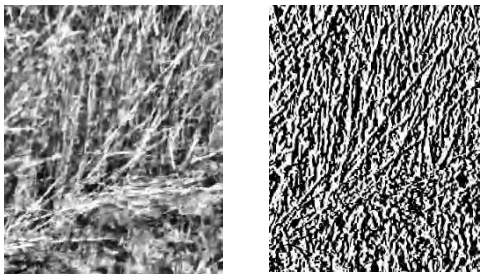


Figure 4.1 Edges found by Horizontal Gradient Detection

The edges produced by the model of horizontal edge collection are shown in Figure 4.1. These edges were detected by using a thresholding operation. The strongest and our targeted horizontal line was found by setting the threshold of T=5 for the model.

## 4.2 Proposed Directional Connectivity Feature Extraction Technique

The examples of roadside images in Figure 4.2 depict the target grass regions more precisely while Figure 4.3 shows the proposed workflow framework for frame categorisation from video. If we analyse the intended target more deeply, it is clear that only identifying areas of grass coverage cannot fulfil the requirements. It is necessary to calculate the density or grass pixel direction from the input images. This is one of the great challenges of this type of complex image processing. The main aim is to find the grass pixel direction accurately, thus solving the problem of grass density estimation. In order to perform this task, a huge dataset was retrieved from videos and sample images collected during the initial field survey to create the database for feature set extraction. Then those features were trained using several machine learning algorithms from which the best feature was chosen. A local window was then applied on the segmented frame for making the decision about the frame. The concept for considering a window area comes from a survey. During the survey, all of the grass in areas of one metre square was trimmed and collected; the biomass was calculated from these samples and these grass regions were categorised based on the levels of biomass. For doing this we need to extract the grass pixel direction and grass coverage. Finally, based on the ground truth we calculate the results and compare human versus machine accuracy. The process is described as follows.

(a) Dense (b) Moderate (c) Sparse

Figure 4.2 Grass Density Estimation: (a) Dense (b) Moderate (c) Sparse

It is clear from Figure 4.2 that the grass coverage areas in the images will be the same in most cases. Almost 75% grass coverage will be found from all the images, but all grasses do not grow in the same way. Considering the problem in terms of grass biomass, it will be different in all three images. This happens due to the different grass density. In the dense grass (Figure 4.2(a)), both grass coverage and the grass height are high. On the other hand, the scenarios are different for the moderate (Figure 4.2(b) and sparse grasses (Figure 4.2(c)). Finding the proper grass height is a vital issue for density estimation. Hence, we need to focus not only on proper identification of grass coverage, but also need to ensure proper estimation for grass height measurement to achieve proper grass density estimation. To serve the specific purpose, a new feature extraction strategy is proposed which can solve the problem.

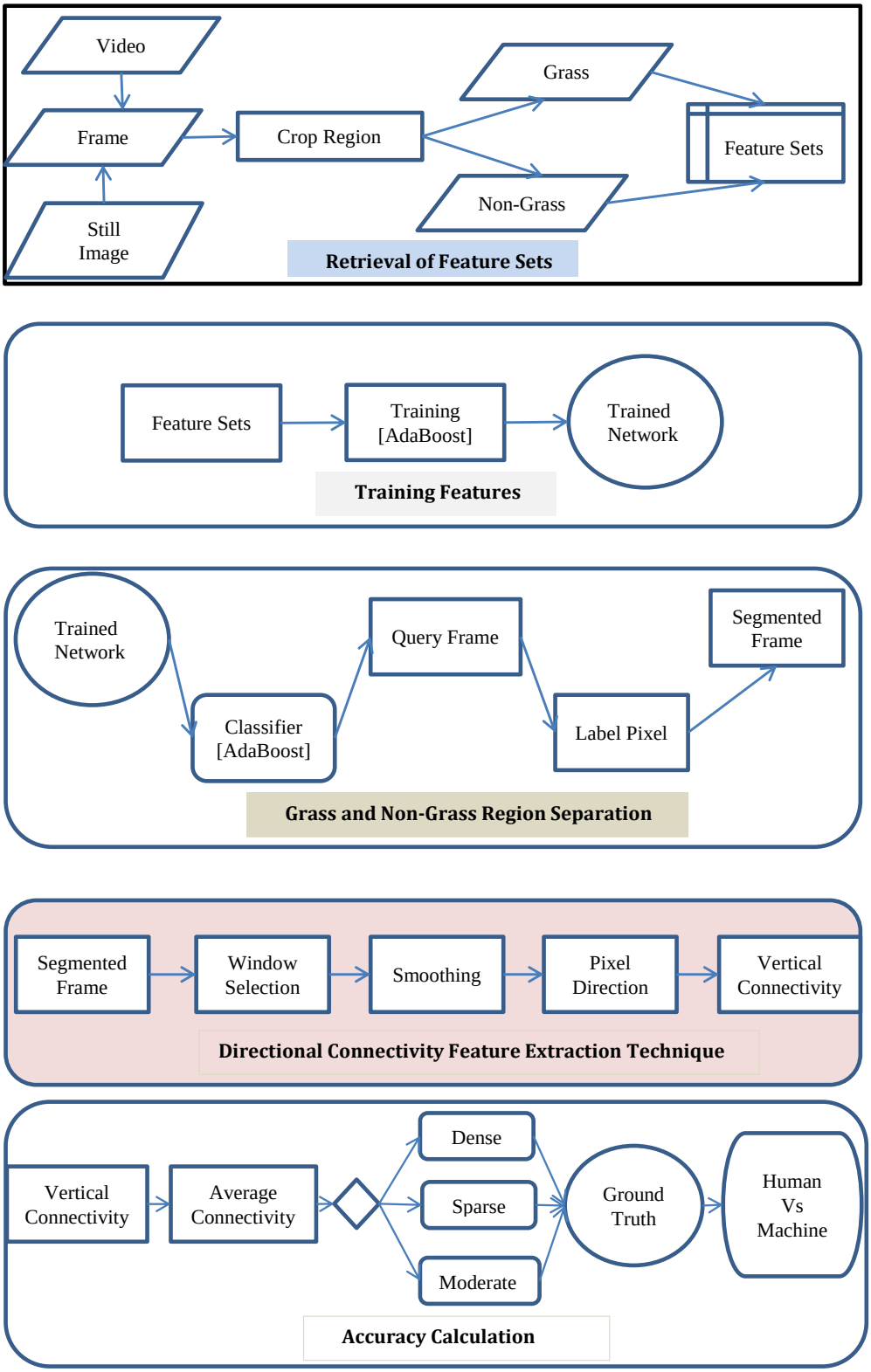



Figure 4.3 System Workflows for Frame Categorisation from Video

## 4.3 Feature Extraction

In the last decade, tree stem height estimation has been extensively investigated [132] [133] [134] [135] [136]. Among them, Paris *and* Bruzzone [121] proposed a method for the reconstruction of tree-top height by fusing low-density LiDAR data and high-resolution orthophotos. Their proposed method cannot measure tree height from normal images taken straight using cameras or taken a frame from video data. A similar approach has been presented by Chen *et al.* [136] to retrieve canopy height from large-footprint satellite LiDAR waveforms over mountainous areas.

After selection of windows, the major contribution of this research is calculating the grass pixel directional connectivity features. Finally, we use the features for categorising grass density. For grass density estimation, we calculate the features in each window.

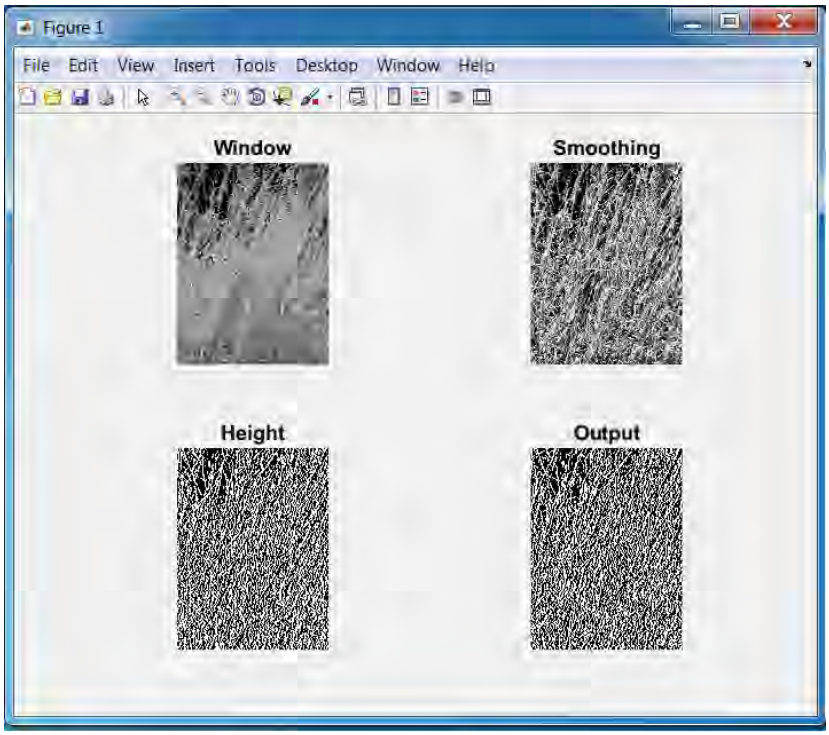



Figure 4.4 Workflow for Grass Height Measurement

Figure 4.4 shows a basic workflow for grass vertical directional connectivity measurement from the window region. Initially, we start from the left side top corner of our window. As soon as the window region is selected we take the sub-image of that window from the segmented region. This sub-image contains only grass and non-grass regions. Here the grass region is shown with the original colour of the grass and the non-grass region is highlighted with black colour. To make the window region more clearly defined, we apply adaptive thresholding in the local region for smoothing of the image. The effect of this smoothing can be seen in Figure 4.4. The grass part is now more visible and can be easily differentiable for further calculations. In order to calculate the grass part, we need to recognise the

grass part. The Canny edge detection technique can be applied but it comes with many noises and fails to work in fulfilling our target. So we proceed with pixel direction and magnitude and try to determine the connectivity and relationship within similar grass and non-grass pixels. This final strategy helps to achieve the desired goal with good accuracy of grass height calculation, giving a high value where the grass region is high and a low value where the grass region is low. After deep analysis, it is seen that the structure of grass height is not always straight. For our purpose, we only consider the 90° orientation of the grass region.

To measure the grass pixel directional connectivity it is necessary to initially extract gradients in both x and y directions. The formula for image gradient calculation is:

$$\nabla f = \frac{\partial f}{\partial a}\hat{a} + \frac{\partial f}{\partial b}\hat{b} \quad (4.4)$$

where:

$\frac{\partial f}{\partial a}$ is the gradient in the x direction

$\frac{\partial f}{\partial b}$ is the gradient in the y direction.

Gradient direction can be defined as follows:

$$\theta = \tan^{-1}\left(\frac{\frac{\partial f}{\partial b}}{\frac{\partial f}{\partial a}}\right) \quad (4.5)$$

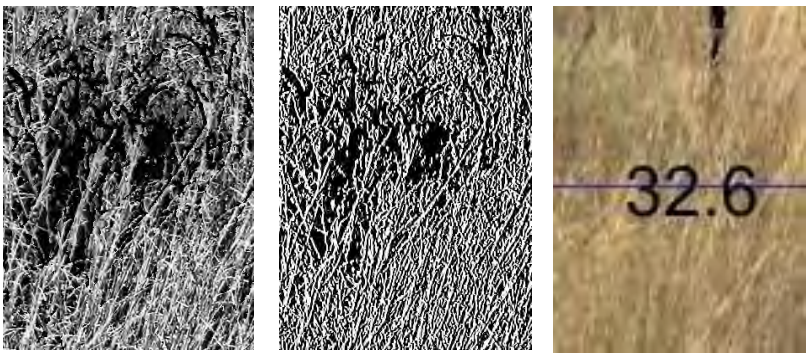



Figure 4.5 High Connectivity for Dense Grass Region

Figure 4.5 shows an example of high connectivity for a dense grass region. Finally, to calculate the grass height we consider only the 90° orientation and this can be explained as follows.

Algorithm 1: Pseudo-code of Directional Connectivity Feature Extraction Technique for Roadside Grass Density Estimation in all basic windows

*Algorithm : Directional Connectivity Calculation*

***Input**: Let $w_1$, $w_2$ ……$w_{15}$ are the fifteen windows. We have to calculate directional connectivity for each window.*

```
procedure MyProcedure
input: Classified picture P € C^{M X N}
Initialisation
        C ← 1, winccp ← zeros(15,1,'single');
        maxVerCon ← zeros(winWidth,1,'uint16');
begin
    for all the window w € |1,M| do
        smoothing the classified window
        calculate the vertical direction of each pixel
        for each F(a ,by) calculate ∇f = ∂f/∂a â + ∂f/∂b b̂
        for each F(a , b) θ ← ((atan2(∂f/∂b b̂, ∂f/∂a â)+pi)*180)/pi;
        output image: image with direction and magnitude
        T^{R,C}_{row,col} ← { };
        for all row ← 1: Width
            for all column ← 1: Height
                check for 90 °vertical orientation
                        T^{R,C}_{row,col} ← 255;        %% save the direction
            end for
        end for
    Length_col ← 0 ; C^C_col ← { };
    for all col ← 1: Width
        for all row ← 1: Height
            if T_{col,row} ← 255;
                Length_col ← Length_col + 1;
                if T_{col,row} ← 0;
                  if consecutive 5 zeros in a column
                  save the previous length
                  Check whether this length is higher than old length
                  Update the length
                end if
            end if
            save the highest length for each column
                C^C_col ← max C^{k,l}_row {Length_col};
        end for
    end for
Get the mean value of all column M_{w_k} elements in C^c_col:
        winccp_{W_k} = 1/M_{w_k} Σ_{i=1}^{M_{w_k}} C^c_col;
```

***Output:** Vertical Connectivity Value for each Window*

$$\langle \boldsymbol{winccp}\ (\mathbf{1} \ldots \mathbf{15}) \rangle = \frac{1}{M_{w_k}} \sum_{i=1}^{M_{w_k}} C^c_{col}\,;$$

From Figures 4.6 and 4.7, it is obvious that a sparse region gives a lower value compared to a dense region. A moderate grass region lies between the sparse and dense regions. In a real scenario, it is hard to differentiate these moderate regions. But very little confusion arises between sparse and dense region separation. If we can successfully differentiate these two regions, it would be a great success. If we can also minimise misclassifications between dense and sparse regions, our effort would be generally successful.

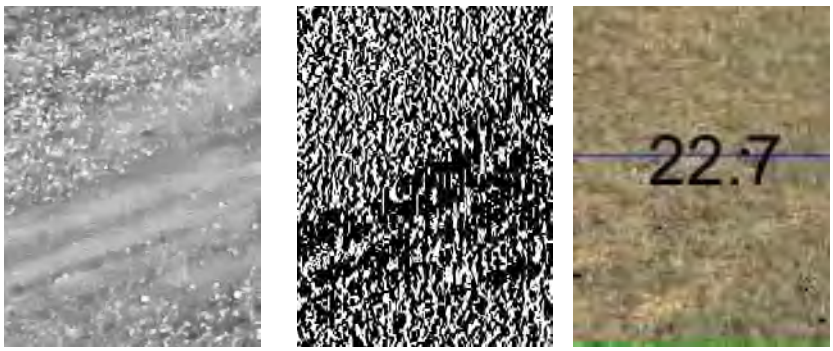



Figure 4.6 Low Connectivity for Sparse Grass Region

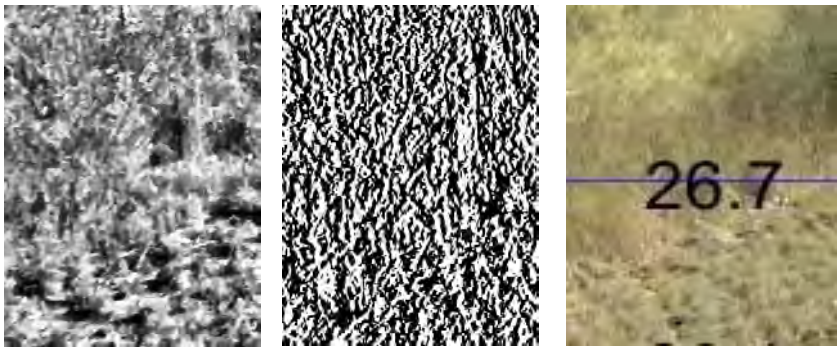



Figure 4.7 Medium Connectivity for Moderate Grass Region

## 4.4 Experiments and Results

An accurate data collection process is critical to prove the effectiveness of the proposed method. The next sections explain the data collection procedure and the study area..

### 4.4.1 Data Collection and Setup

Data collection for this study included gathering roadside video data as well as collecting biomass samples from specific locations. A field survey was also conducted to evaluate the accuracy of the vegetation classification process. A large amount of roadside images were taken and, among them, 61 sites were selected as locations from which to take biomass samples. The study area mostly covered the Central Queensland region as shown in Figure 4.8 which includes Rockhampton, Baralaba, Blackwater, Emerald, Springsure, Biloela, Dingo, Duaringa, Gladstone, and Yeppoon. In the first phase, a series of data were collected and ground truths were set according to human observation. In the second phase, the collected samples were dried at 70° Celsius for more than 72 hours and biomasses in tonnes/ha have been calculated. According to the biomass samples, ground truth was refined. In the

final phase, collected samples were organised and processed with the developed technique to measure the performance in respect to human observation. From the collected samples, thresholds were chosen from the biomass sample results to indicate high, medium and low risk regions.

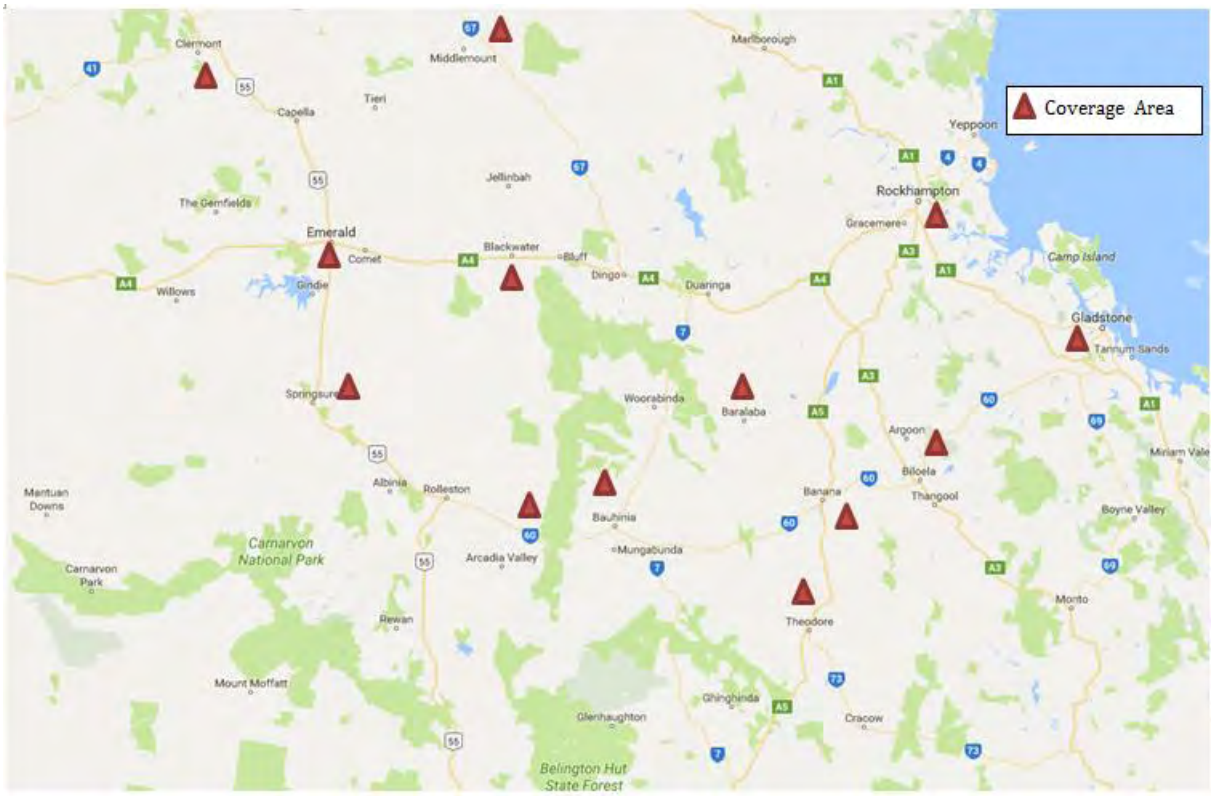



Figure 4.8 Study Area

### 4.4.2 Retrieval of Feature Set

For successful classification of objects [137], the most important step is to find the feature sets which can be retrieved from a set of training images. This helps to differentiate one object from another as well as improve the computational efficiency. A good retrieval set will contain images from all sets depending on the target. For this purpose, our target is the separation of grass and non-grass regions. So the feature set covers grass regions mostly found on the roadside, and non-grass regions like soil, sky, tree, road signs, etc. Different types of global features have been analysed to capture the types of similarity among grass and non-grass regions: colour space, entropy, and colour histogram. Finally, the RGB colour features along with thresholds are used for feature sets. Some cropped regions for the retrieval set are shown in Figure 4.9. These regions are cropped manually from the video data taken using a 2D car-mounted video camera. During cropping, we were careful enough so that non-grass region comes within the cropped region and a similar strategy is followed for non-grass regions also. There are 356 cropped grass regions and 295 cropped non-grass regions taken from different locations from video data as well as from survey images. Figure 4.9 represents the grass regions and Figure 4.10 represents non-grass regions.

Figure 4.9 Cropped Grass Regions

Initially, we create the feature set with colour features. The colour feature vector corresponding to a pixel at coordinates (x, y) is extracted from the cropped region I as follows:

$$F1\ (x, y) = \left[R, G, B, |R - G|, |R - B|, |G - B|, \frac{1}{3}\ (R + G + B)\right]$$

The complete feature vector can be defined using the following notation:

$$F = \ [F1]$$

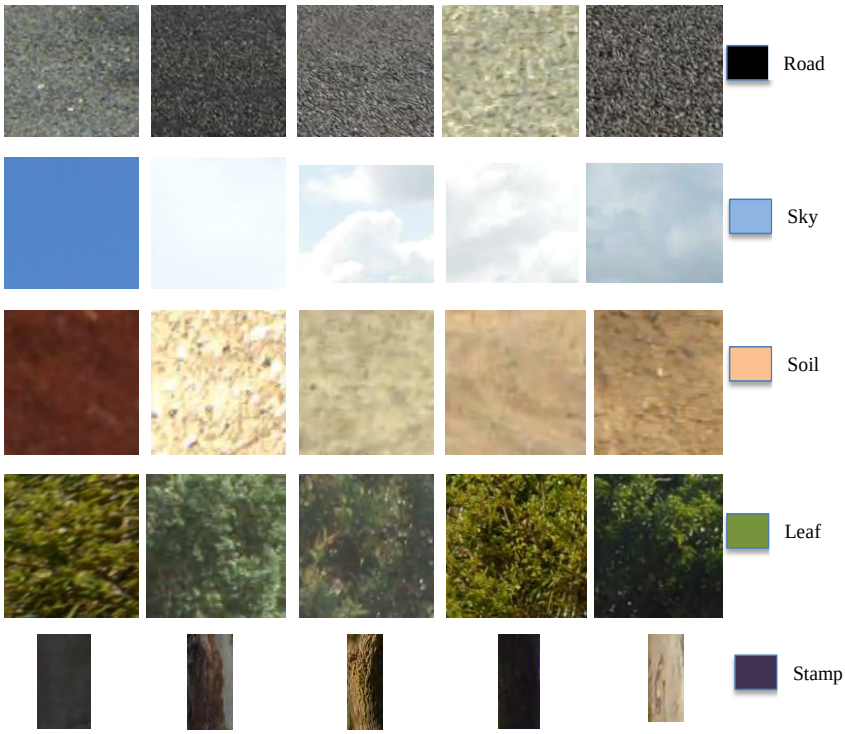



Figure 4.10 Cropped Non-Grass Regions

### 4.4.3 Training Feature Set

In order to label the feature sets based on the content of the retrieval set, labels are assigned for each region. Here we use the colour feature for retrieval of feature sets. Two types of colour features are shown in Figure 4.11 and Figure 4.12. Figure 4.11 presents the colour feature for grass regions whereas Figure 4.12 presents the corresponding detail for the road as an example of non-grass regions. From the two images, we can easily differentiate grass and non-grass regions. So this is a better feature for applying in machine learning for training the feature sets. To achieve better accuracy, three machine learning techniques were applied - Support Vector Machine, Neural Network and AdaBoost. The best accuracy was achieved using AdaBoost for the interclass classification.

As our target object is grass, from the image dataset we crop data from grass and non-grass regions and thus we divide into two classes grass and non-grass and the problem becomes a binary classification. We define our dataset as follows $\{(a_1, b_1)\ , \ldots\ldots (a_n, b_n)\}$ where $b_n$ € {1,2} represents the true class label for each feature set. For training, we need to set a weight function which is updated to minimise the error. Here w is used for weight observations and expressed as $i_n$. We also need also an error function which is defined by E.

The error function is the exponential loss of each data point and is given as:

$$E\,(f\,(a), b, n) = \sum e^{-b_n f(a_n)} \qquad (4.6)$$

Let $w_i^{(1)} = 1\ and\ w_i^{(n)} = \ e^{-b_n f(a_n)}\ for\ n > 1. Then\ we\ have$

$$E = \sum_{n=1}^{N} w_n e^{-b_n f(a_n)} \qquad (4.7)$$

Here $f(a_n)$ is the predicted classification score and the computation of prediction for data uses:

$$f(a) = \sum_{t=1}^{T} \alpha_t h_t(a) \qquad (4.8)$$

Where

$$\alpha_t = \frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t} \qquad (4.9)$$

and

h: a → [1, 2]

Updating of weights for all t in 1 ……. T uses:

$$w_{i,\ t+1} = w_{i,t}\, e^{-b_n \alpha_t h_t(a)} \qquad (4.10)$$

Table 4.1 describes details regarding the ground truth. The accuracy of trained data is listed in Table 4.2, whereas Table 4.3 represents the accuracy of test data. We calculated the accuracy in two ways. First, we applied the trained network on the trained data and listed the accuracy. Then, in order to test whether it worked on non-trained images, we created a test set with 57 cropped regions which are totally different from the training cropped region and list the accuracy. For further analysis of whether grass and non-grass can be differentiable from real frames, we applied the trained network on real frames and observed the classification accuracy.

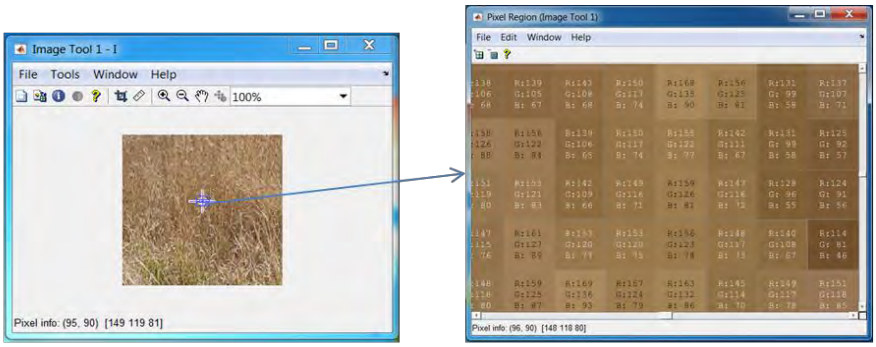



Figure 4.11 Features for Grass Region

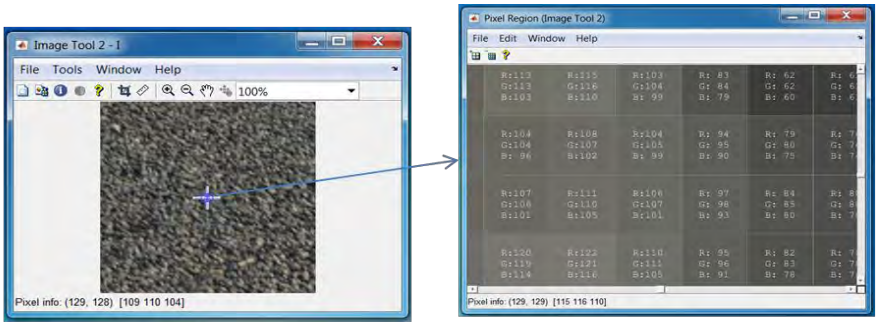



Figure 4.12 Features for Road as Non-grass Region

Table 4.2 shows that we have samples from various grass regions and the number of cropped regions for representing grass is 356. Looking at Figure 4.4 for grass regions, we see that the grass is not limited to one type. It may be deep green, light green, deep brown, light brown and mixed with brown and green. Within green and brown lots of variation can also be found in real scenarios. Again, choosing the objects for creating the feature property for non-grass regions is also a difficult task. Non-grass objects can be anything on the roadside. Here we consider road, sky, soil, tree leaf and tree stump. Although in a real scenario there can be more objects, it is necessary to determine the common items on the roadside and cover as many items as possible. From the figure, it is clear that there are lots of variations on road data within the road region. The same scenario is found for soil regions, tree leaf regions, tree stump and sky regions.

### 4.4.4 Grass and Non-Grass Region Separation

To test our trained classifier we take some query frames from both survey data and video data. For the convenience of understanding, we assign 255 frames for grass regions and 0 for the non-grass regions. In order for proper verification, we retrieve the original images with only segmented pixels. Some examples of the results are shown in Figures 4.13, 4.14 and 4.15.

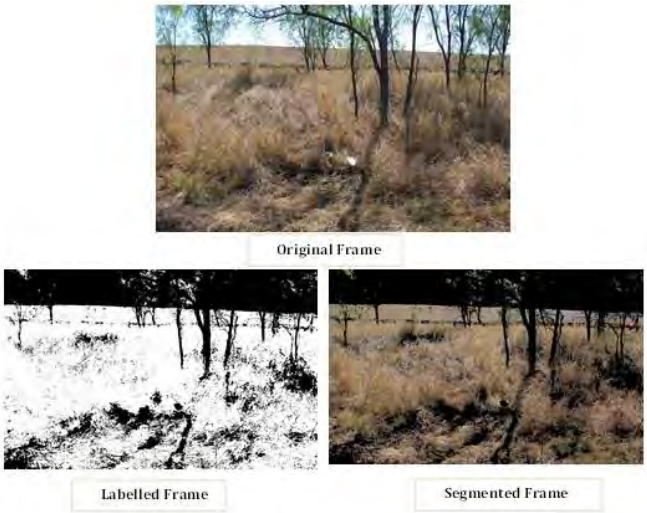



Figure 4.13 Output for Sample #30 taken during Survey

The first image is taken from sample F030. Sample numbers were given during survey data collection and are detailed in Table 4.1. Here the original frame contains brown grass, tree, and sky. There is no soil and the grass density is near to moderate. The second figure, called the labelled frame, is the output frame after being classified by the proposed classifier. Here the output is for two options - grass and non-grass. For convenience, we define these options as 1 for grass and 2 for non-grass. To generate the image for grass we assign R ⟨x,y⟩ equals 255, G ⟨x,y⟩ equals 255 and B ⟨x, y⟩ equals 255 and for non-grass we assign R⟨x,y⟩ equals 0, G ⟨x,y⟩ equals 0 and B ⟨x,y⟩ equals 0. This is so that we can represent the images using two dimensional figures. Here the labelled frame shows that, by using the proposed classifier, we are able to differentiate grass and non-grass regions near to accurate. This means our network is trained well and gives satisfactory performance. Here we observe some misclassifications on differentiating between grass and non-grass due to shadows. This can be properly investigated from a third image called the segmented frame. Here we reconstruct the original image from the classified pixels and observe the difference. As colour is one of the important factors for human perception, we applied original colour on the labelled frame. When trying to find the coordinates which represents a grass region, it is necessary to specifically look for R ⟨x, y⟩, G ⟨x, y⟩ and B ⟨x, y⟩ equals 255.

$$\langle Nrows, Ncols\rangle = find\ (R\langle x, y\rangle == 255\ \&\ G\langle x, y\rangle == 255\ \&\ B\langle x, y\rangle == 255)$$

Assigning original colours can be done by following

segmented image $\langle p,\ q\ , i\rangle$ = original image $\langle p,\ q\ , 1\rangle$

Here {p∈ $R$ , q∈ $C$} and i ∈ {1, 2, 3} are image channel numbers. R represents Row and C represents Column.

The mathematics behind the class prediction is as follows:

The class prediction depends on the probability function. In our scenario, we are given a set of training data {($a_1$, $b_1$) . . . ($a_n$, $b_n$)} where input is feature vector $a$ € $R^p$ and output is assumes value $b_n$ € {1, 2} which represents the true class label for each feature set. The trained network is loaded and the prediction class is calculated as follows:

$$\hat{y} = arg_{y=1\ldots K}\ min \sum_{k=1}^{K} P(k|a)\ \ C(b|k) \quad (4.11)$$

Here

*$\hat{y}$ is the predicted class $\hat{y}$ $\in \{1, 2\}$*

*K is the number of classes here we have two classes (grass, non-grass)*

$P(k|a)$ *is the posterior probability function that a observation p belongs to class k*

*This can be defined using*

$$P(k|a) = \frac{P(a|k)\ P\ (k)}{P(a)} \quad (4.12)$$

*Where*

$P(a|k)$ *is the prior probability which can be calculated by*

$$P\ (a|k) = \frac{1}{(2\pi\ |\sum_k|)^{\frac{1}{2}}} \exp(-\frac{1}{2}(x-\mu_k)^T\ \ \sum_k^{-1}(x-\mu_k)) \quad (4.13)$$

*whereas $|\Sigma_k|$ is the determinant of $\Sigma_k$, and $\Sigma^{-1}{}_k$ is the inverse matrix.*

$C(b|k)$*is the* <u>*cost*</u> *of classifying an observation as y when its true class is k.*

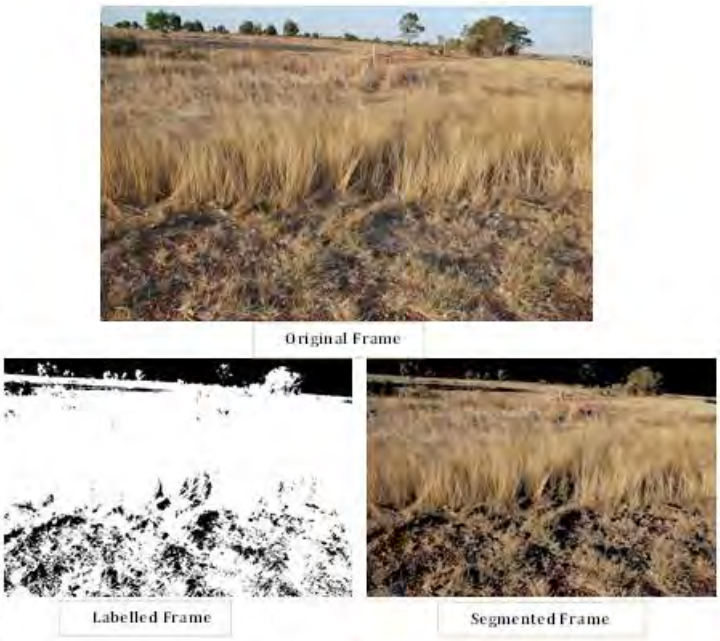



Figure 4.14 Output for Sample #26 Taken During Survey

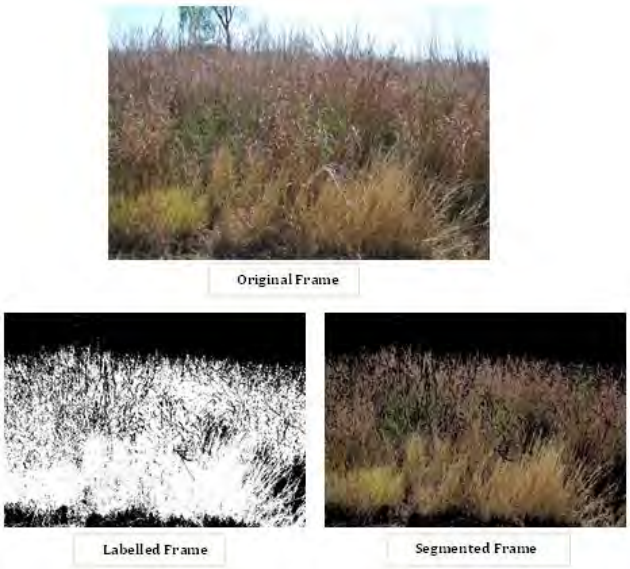



Figure 4.15 Output for Sample #54 Taken During Survey

### 4.4.5 Window Selection from the Segmented Area

Recently, estimating and preserving the aboveground biomass from satellite data using remote sensing techniques has been proposed [138]. In their proposed approach, the authors added two parameters: canopy height and leaf area index. However, our proposed approach needs to incorporate the relationship of biomass and directional connectivity value. Biomass was collected during the survey and used as ground truth for our experiments. Here we tried to calculate the grass pixel direction and grass area coverage for a specific area and match them with the biomass and thus we developed a method for calculating biomass from real world images.

To obtain acceptable accuracy of our method, we need to select local windows from the segmented image region. Initially, we proceeded by selecting the highest grass height from the segmented region and drew the first window from the bottom of the grass height with a width and height of 300 and 400 respectively. Then, we drew the second and third windows on the left and right sides respectively by making a pixel distance of around 100 on both sides. If the first window reached either the left or right side border, we took the 2nd and 3rd windows on the other side. Figure 4.16 shows the window selection from an original image. Here the tall grass is in the middle and this should therefore be the first window which is selected automatically from the whole frame. The only challenge for this type of window is selecting the first window. The second and third windows are just added using some constant values in x and y directions. For a better understanding, and to keep consistency with the biomass collection, we keep the window width and height at a ratio of 300 (W) X 400 (H).

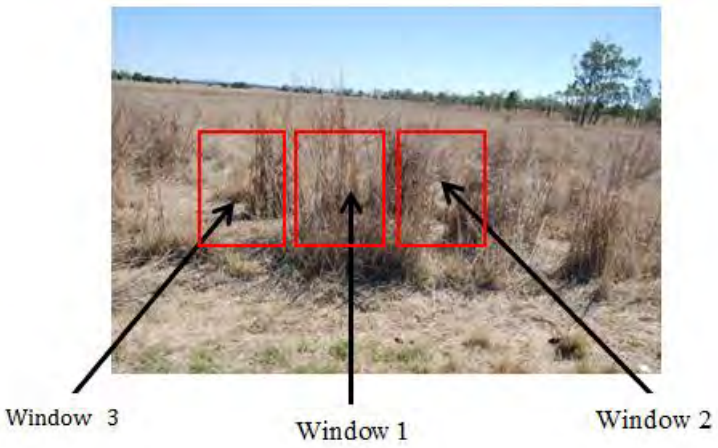



Figure 4.16 Window Selections from Frame

The reason behind the scene is that, during the survey we randomly selected an area, cut the grass, calculated the biomass, and categorised the area into dense, moderate and sparse according to the biomass. For better decision making, we divided the whole area into 15 windows which are shown in Figure 4.17 for samples F026, F030 and F054 respectively. During the window selection, we tried to avoid both the upper and lower parts of the image. This was on the basis that the upper part mostly contained sky region and some far distant grass regions which are not our concern. The reason for avoiding the lower part is that we found by analysis that most of the lower part of the frames contained road which is also not our concern. Then the main challenge was selecting an appropriate window size. After a thorough analysis, we decided that a window size with a width of 300 and height of 400 would serve our purposes for decision making. If the size was too small it would decrease the accuracy, and if too big, decisions may not be correct. Figure 4.18 shows not only the selected window from the segmented frame but also the grass pixel direction value for each window. The procedure to ascertain the value calculation has been described under directional connectivity feature extraction (Section 4.2). Here the entire window is overlapping. We do not want to miss any grass region during decision making. .

Figure 4.17 Overview of Window Selection from Samples #26, #30 and #54

To prove the accuracy of the proposed method, a dataset with a hundred images was created and the ground truths for each window were drawn and the results finally compared with the automated outputs. Figure 4.18 shows a sample image with ground truths set by a human. It is a mixture of dense, moderate and sparse regions.

Figure 4.18 Ground Truths on 15 Windows

### 4.4.6 Ground Reference Data

Table 4.1 shows ground truth for sample images collected during survey. Here we collected samples sequentially and marked as F001, F002 and so on. Based on the characteristic of the grass and finally after calculating biomass in lab, we categorize the collected samples into three categories which is depicted in Table 4.1. We also took picture for each sample, so that we can analyze the image using our proposed technique and can verify the ground truth with biomass. Categories of grass mostly depend on grass height and biomass.

Table 4.1 Ground Truth for Survey Data

| *Grasses* | *Sample Numbers* |
|---|---|
| Sparse Grass | F007, F008, F014, F017, F019, F022, F026, F030, F033, F034, F035, F038, F039, F043, F048, F050, F051, F056, F058, F060 |
| Moderate Grass | F002, F004, F006, F011, F013, F018, F020, F024, F027, F029, F032, F036, F040, F041, F042, F045, F047, F049, F052, F055, F061 |
| Dense Grass | F001, F005, F009, F012, F015, F016, F021, F025, F028, F031, F037, F044, F046, F053, F054, F057, F059 |

We tested the effectiveness of the grass density estimation technique in our created image database with manually annotated images. Figure 4.19 provides some visual examples of the obtained results. The first column presents the input images with original image colour; the detected grass region is shown in the second column where black represents the non-grass region and the grass region is shown with original colour. The third column presents the results of the segmented region as a binary image associated with each input image of the first column. In Figure 4.20, more experimental outputs are shown not only with the segmented region, but also with directional connectivity values.

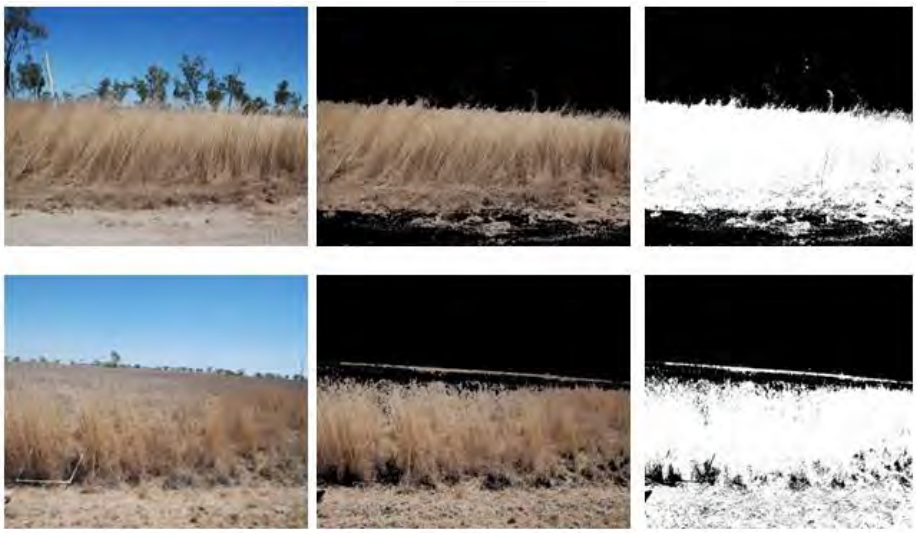
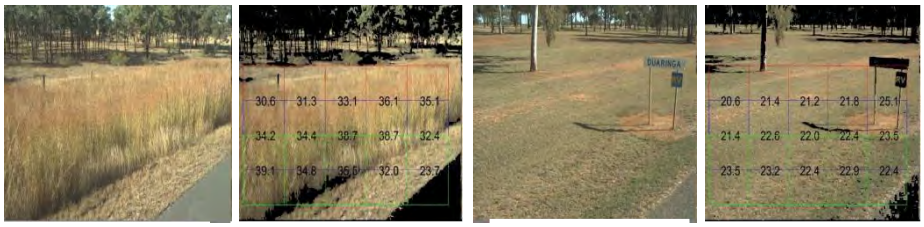


Figure 4.19 Experimental Results with Segmented Output




(a) Original Image (b) Connectivity Value (a) Original Image (b)Connectivity Value

Dense Region Sparse Region

Figure 4.20 Experimental Results with Directional Connectivity Output

## 4.5 Result Analysis

To fairly assess of the proposed classification system's performance, classification accuracy was evaluated using not only the pixel-wise classification frequency, but also the density estimation over all classes.

Figure 4.21 shows the connectivity value variations for those images categorised as dense grass regions. It is obvious from the figure that the connectivity is around 28 for dense grass. Misclassifications sometimes occurred and this has reduced the performance. From the figure we see a value as low as 15 and some others are below 25. This occurs due to shadow and illumination

conditions. At this stage, we did not focus on removal of that type of scenario during segmentation. This resulted in some inaccurate values and dropped the classification performance.

Table 4.2 Accuracy on Trained Data

| ***Sequence*** | ***Classifier*** | ***No of images Grass + Non-Grass*** | ***Accuracy (%)*** |
|---|---|---|---|
| 1 | Support Vector Machine | 356 + 295 = 651 | 90 |
| 2 | Neural Network | 356 + 295 = 651 | 92 |
| 3 | AdaBoost | 356 + 295 = 651 | 95 |

Table 4.3 Accuracy on Test Data

| ***Sequence*** | ***Classifier*** | ***No of Grass Images*** | ***Accuracy (%)*** |
|---|---|---|---|
| 1 | Support Vector Machine | 57 | 70 |
| 2 | Neural Network | 57 | 73 |
| 3 | AdaBoost | 57 | 77 |

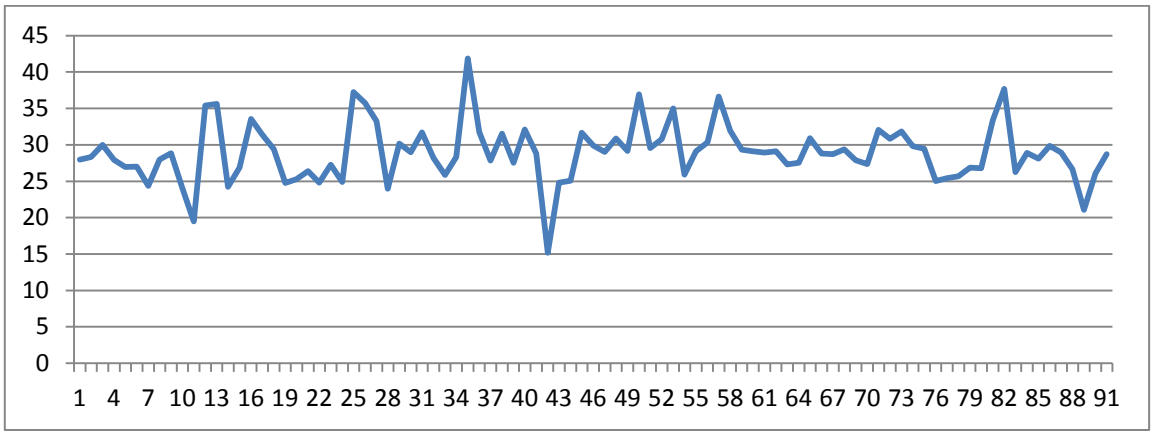
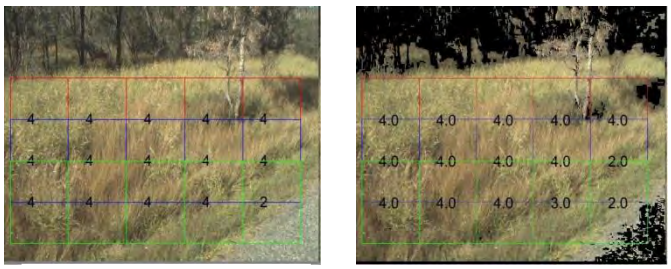



Figure 4.21 Dense Grass taken from Video Data



(a) Ground Truth (b) Output

Figure 4.22 Dense Grass Classifications

Figure 4.22 shows some ground truth annotated manually for testing and the corresponding machine output for the selected image. Results show that most of the ground truth was successfully retrieved. Some misclassification occurred for window 10 and window 14. Although ground truth from Figure 4.22(a) showed that the grass region should be dense (4) for window 10 it was categorised as

sparse (2) and a similar scenario occurred for window 14 where the ground truth represented the grass as moderate and the machine recognised it as dense.

Similar kinds of surveys were done for sparse regions and the results recorded are shown in Figure 4.23 from which it is clear that the threshold for sparse grass was around 26.5 and below. Some misclassification results occurred for some images and gave high values of about 36. Figure 4.24 shows the results of ground truth versus classified output. The output shows that it fulfilled all of the ground truth requirements which constituted one of our major concerns for this research.

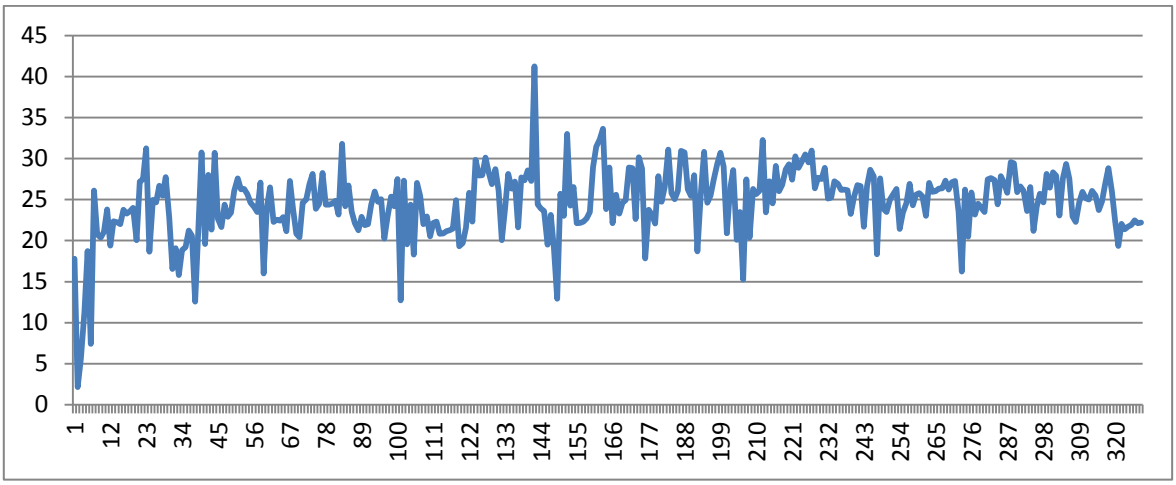
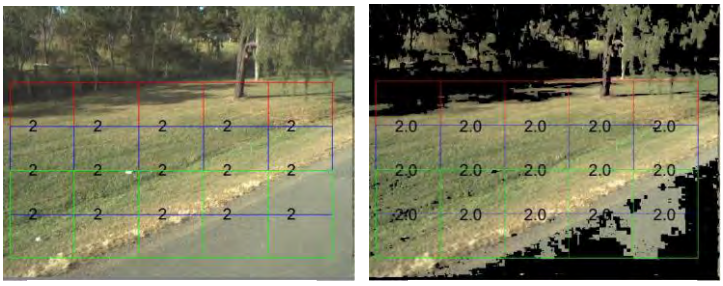



Figure 4.23 Sparse Grass taken from Video Data

Here, although grass coverage is high, the grass density is low; while this should be differentiable, it was found to be difficult to differentiate from a 2D image. A major success of this research was incorporating a new feature to make this differentiable. Overall performance shows that good accuracy has been achieved by the machine in the overall scenario from the real-time environment.




(a) Ground Truth (b) Output

Figure 4.24 Sparse Grass Classifications

We also analysed the performance for moderate grass regions. To set any rules for moderate grass regions in a real scenario is complicated. In most of these cases, confusion was seen when differentiating moderate grass. Sometimes the same grass was categorises as moderate by one person, whereas it seemed dense for someone else. Similar things occurred in differentiating between sparse

and moderate regions. Figure 4.25 shows the variation in threshold with the moderate grass regions where the value is generally greater than 26.5 and less than 28.

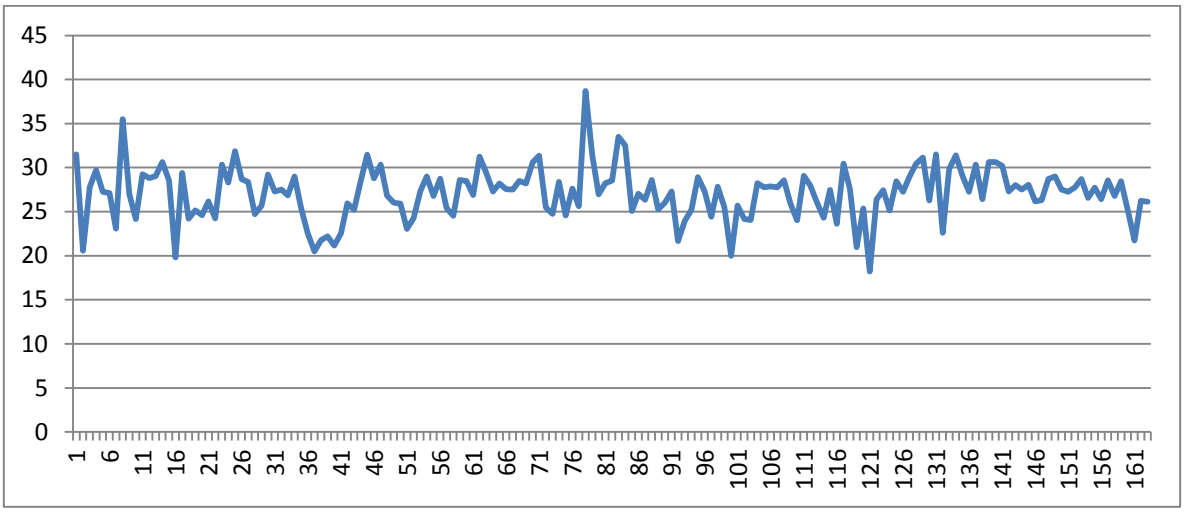



Figure 4.25 Moderate Grass taken from Video Data

An overall comparison between the thresholds is shown in Figure 4.26. Where dense occurs in all windows, we get a higher value - which is greater than the threshold for dense as expected. Similar scenarios were also obtained for moderate and sparse grass regions. From observation, it was clear that first figure looked like dense grass, the second figure looked like moderate grass and the third one was obviously sparse grass. These classifications were also differentiable by the values provided by automated output. So we can say that the machine exhibits satisfactory performance compared with human observation. .

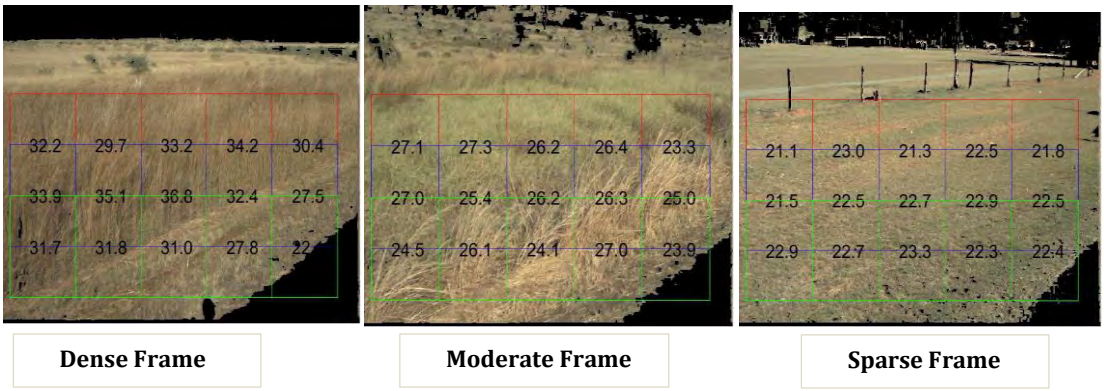



Figure 4.26 Differences between the Thresholds for three types of Grass

### 4.5.1 Accuracy of Human versus Automated Survey

This section presents a detailed evaluation of classification accuracy of our system. The evaluation was conducted on both human annotated data and machine annotated data. The results were recorded and, for better understanding, we drew a linear regression as shown in Figure 4.27. According to this figure, it is clear that the accuracy of the machine was higher than human observation.

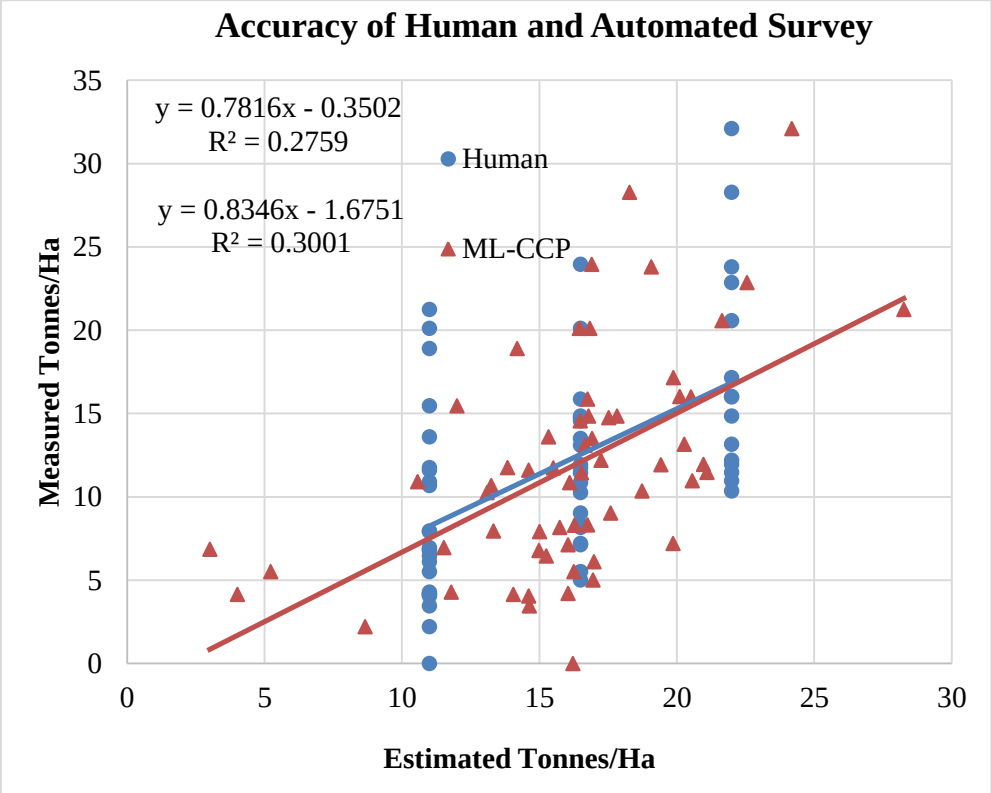



Figure 4.27 Performance Evaluation

Here the blue dot points represent ground truth for a human which was collected during survey data analysis. Actually, the ground truth was set according to biomasses which were calculated in a lab environment. Then it was converted into a similar scale to make it comparable to automated data. Here, the red triangle points were obtained as a result of machine learning technique. Then, using regression analysis, we tried to figure out the similarity and measure the performance. The straight line shows that the automated output almost overlaps the ground truth. Hence, we can say that the automated technique performs almost the same as a human. Moreover, the accuracy depends on R2. If the value of R2 is almost equal, we can say that the machine performs well and if it exceeds it is performing better than a human. Here in our scenario, the R2 value exceeds human observation, so it is obvious that the machine performs better than a human. In reality, a human cannot perform consistently with such types of data. When watching the same types of scenario over and over, humans usually lose concentration and focus and thus they make mistakes. If we can employ a machine in such scenarios, it could save time as well as process the data more precisely.

### 4.5.2 Comparative Analysis

Although some literature was found for object height measurement from the man-made environment using a vanishing point [139], there is no established method for grass height calculation. That's why we were motivated to compare our results by creating our own ground truth. The obtained result shows outstanding performance in terms of running time and accuracy. Though the proposed technique works well in some scenarios, it sometimes fails to classify properly. The threshold for grass density estimation can be defined using:

$$D(W_i) = \begin{cases} sparse, & V_{W_i} \leq 26.5 \\ moderate, & 26.5 < V_{W_i} < 28, \\ dense, & V_{W_i} \geq 28 \end{cases} \quad (4.14)$$

Table 4.4 Ground Truth for 100 images (939 windows)

| | | |
|---|---|---|
| *533* | *0* | *0* |
| 0 | 20 | 0 |
| 0 | 0 | 386 |

Table 4.5 Confusion Matrix for 100 images (939 windows) using Proposed Technique

| | | |
|---|---|---|
| *436* | *73* | *24* |
| 4 | 13 | 3 |
| 40 | 44 | 302 |

Table 4.6 Accuracy for 100 images (939 windows) using Proposed Technique

| | | |
|---|---|---|
| *81.80%* | *0* | *0* |
| 0 | 65 % | 0 |
| 0 | 0 | 78.24% |

Table 4.4 shows the ground truth for 100 images where 939 windows were chosen without confusion among the grass categories. Table 4.5 shows the confusion matrix which is generated from the proposed technique and Table 4.6 shows the overall accuracy for the three classes. Tables 4.7 and 4.8 show the accuracy achieved using the Gabor-based grass density estimation technique. It is obvious that Gabor shows poor performance in the overall scenario. While it shows higher performance than our proposed technique in terms of sparse grass region identification, it shows very poor performance in terms of dense and moderate grass region identification. It only can identify 193

from 386 windows which is a very poor result. But, most importantly, most of the dense grass is identified as sparse - which is a significant misclassification. If it goes for moderate, we can assume that it's confused with moderate. On the other hand, our proposed technique performs better than the Gabor-based technique, which can identify 302 among 386 windows which is promising. Only 40 window regions were misclassified as sparse which is also a good sign for claiming the effectiveness of our proposed method.

Table 4.7 Confusion Matrix for 100 images (939 windows) using Gabor-based Technique

| | | |
|---|---|---|
| *438* | *59* | *36* |
| 4 | 11 | 5 |
| 117 | 76 | 193 |

Table 4.8 Accuracy for 100 images (939 windows) using Gabor-based Technique

| | | |
|---|---|---|
| *82.18%* | *0* | *0* |
| 0 | 55 % | 0 |
| 0 | 0 | 50.00% |

Accuracy Comparison in terms of running time is recorded in Table 4.9. The proposed method is more efficient than the Gabor-based technique in terms of running time.

Table 4.9 Time Comparison Chart

| **Method** | **Running Time** |
|---|---|
| Proposed Method | 445.47 s |
| Gabor-Based Technique | 709.80 s |

## 4.6 Summary

This chapter described a novel directional connectivity feature extraction technique for grass density estimation and showed how to apply it to images taken from real-world video data. The proposed technique segments video streams into the grass and non-grass regions and detects grass density. Furthermore, the proposed method helps to identify grass density from a complex roadside scenario. The development of this new technique along with its satisfactory results can save time and produce better accuracy than humans. The proposed system also has the potential of differentiating between grass and non-grass areas in each frame. At the same time, it reduces the dependency on

human observation. Visual as well as experimental results show the effectiveness of the proposed method and its application in automation on roadside fire-risk monitoring.

The presented method is useful in computing the height of the grass. Future study should further investigate the generalisation of object height measurement and evaluate the proposed technique in field tests.

# Chapter 5 Multi-Scale Perceptual Features Extraction Technique

This chapter presents a novel and effective approach for object feature extraction using multi-scale feature levels [140]. Learning of image features plays an important role in computer vision. Representation-based features have recently gained substantial attention due to their potential real-world applications. One such application is scene labelling where one of the key challenges is to distinguish objects with visual similarities. To address the challenges with visual similarities, this section proposes new Multi-Scale Perceptual (MSP) features and a deep learning architecture. The MSP features are designed specifically to distinguish visually similar objects and improve the overall object identification performance. The MSP features have two main advantages: (1) they can differentiate the objects with the same histogram of gradients: and (2) they can differentiate horizontal and vertical objects. The first advantage is achieved by introducing Position with Gradient of Histogram named (PosGH) and the second advantage is achieved by introducing a Plane Consistency Estimation named (PCE). The proposed approach with a deep architecture and MSP features achieves better performance than the existing scene labelling methods on three real-world benchmark datasets - Stanford, MSRC, and SIFT flow.

## 5.1 Introduction

Learning discriminative features plays a central role for almost all recognition tasks [141]. Due to the rapid development of computer vision technology, a wide variety of applications use the advantages of both features and learning algorithms [142] to solve real world problems [143]. Recently, a lot of works have focused on learning discriminative features. One such application is scene labelling whose primary objective is to detect objects in the scene and recognise the corresponding class for each object i.e. labelling each pixel to one object [144] [145]. Scene labelling plays an important role in image understanding. However, the task is quite challenging as it faces some common issues like differentiating visually similar objects and differentiating vertical and horizontal objects. Thus, designing appropriate features for a particular object is still an open challenge for computer vision researchers. Although many research methodologies have been proposed in the last decade to solve feature extraction challenges, there are two key factors affecting the overall performance [146]. The first and most important challenge is appropriate feature representation [147] [148] of objects which can effectively differentiate visually similar objects. For example, distinguishing grass from a tree, road from water, and buildings from others structures and so on. The other challenge is to improve the global label accuracy based on object spatial relationships, e.g. the possibility of grasses and trees

being located near to each other and a road having foreground objects. Researchers have focused on finding a good representation of features for objects and we see the emergence of many feature extraction techniques, such as SIFT [149], HOG [71], GIST [150] and so on. Although those techniques show promising results their performance varies from domain to domain and shows low performance in some real-world applications.

Recently, deep learning [151] [152] has emerged as a competitive method for classifying objects by simulating the human brain. The deep learning architecture tries to learn feature representations from input data and progressively learn more complex features in higher layers. Two successful deep learning models [153] are Convolutional Neural Networks (CNN) and Deep Belief Networks (DBN). The aim of the deep learning architecture is learning raw features during the training phase. In this section, a new deep learning architecture is presented to learn multi-scale perceptual features to improve the overall pixel label accuracy in scene labelling and achieve high performance compared to existing methods. The main objective of this architecture is to encode new powerful feature descriptors such as PosGH and PCE in the architecture. The proposed architecture is different from existing deep learning architectures as it constructs deep features from extracted multi-scale features, while other deep networks learn feature representations [154] from raw pixels.

The key concepts in this section are based on the following observations. During scene labelling, some parts of various objects in real world images look very similar to each other so it is very difficult and challenging to correctly label the image. For example, some portions of road and water in roadside images are confusing and difficult to differentiate. Using existing features, it is difficult to assign a class label for each pixel. The situation is more challenging when objects are similar in all aspects except in respect to the plane. For example, differentiating grass and tree objects from small superpixels is often very difficult although they have different plane orientations.

To address the above-mentioned problems, we propose multi-scale perceptual features which can solve problems and improve accuracy. The novel contributions of this section are as follows. The first contribution is the introduction of deep architecture for generation and learning of Multi-Scale Perceptual (MSP) features. The second contribution is the development of the new Position-based Gradient Histogram (PosGH) feature. The proposed position-based histogram of gradient solves the problem of intra-class variations. The third and final contribution is the development of the new Plane Consistency Estimation (PCE) feature. The proposed plane consistency estimation solves horizontal and vertical object differentiation problems.

## 5.2 Multi-Scale Deep Learning Feature Extraction Technique

In this section, a deep feature learning model is proposed. The model learns MSP features based on visual features extracted from superpixels, and it is trained using images where each superpixel is labelled with ground truth classes. The core idea is to learn more useful high-level features from superpixel features. The proposed deep learning architecture design is shown in Figure 5.1.

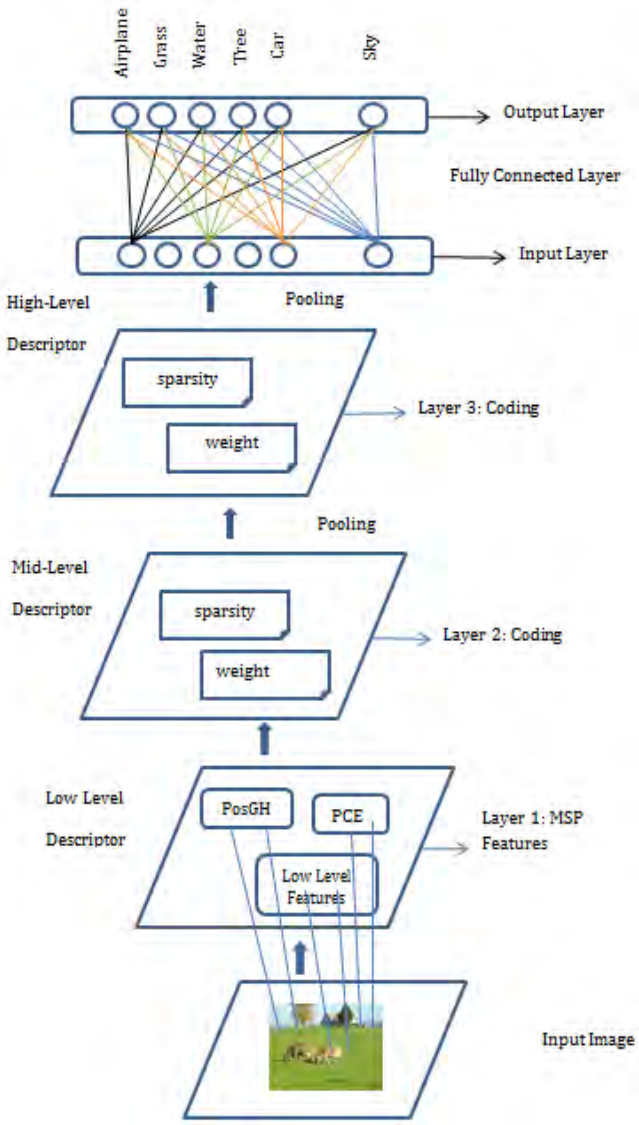



Figure 5.1 Multi-scale Deep Feature Learning Model

Suppose there are n sample images and for each image, there are P superpixels with L labels. Here $P = \{p1, p2, \ldots Pn\}$ and $L = \{l1, l2, \ldots ln\}$, where p denotes the number of superpixels and $l_i \in \{-1, 0, 1, 2, \ldots ln\}$ which defines the number of classes in the dataset. Here $li = -1\ or\ 0$ means an unknown object for this particular superpixel, and is ignored during the calculation of pixel-wise accuracy. For each superpixel, we have a set of low–level features $X = \{x1, x2, x3\ \ldots\ xn\}$ and their known class labels which are fed into neural networks for training. Before passing low-level features to the learning stage as shown in Figure 5.1, we learn deep features through multiple coding stages.

The dimension of features for each layer is reduced using auto encoders leading to deep features. The auto encoder performs unsupervised learning by reducing the cost function. Adopting the same notation as in [155], the encoding and decoding processes for producing MSP features are calculated as shown in Equation 5.1.

$$\hat{f} = h^2(W^2 f + b^2) \qquad (5.1)$$

where, $\hat{f}$, W and b represent the output matrix, weight matrix and bias vector respectively, and the superscript represents the layer number in the proposed architecture. W is the connecting weight matrix and represents local pixel information. The cost function is shown in Equation 5.2 and is calculated by measuring the difference between input features f and the new reconstructed feature $\hat{f}$.

$$E = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \left(f_{kn} - \hat{f}_{kn}\right)^2 + \alpha * \varphi_{weights} + \beta * \varphi_{sparsity} \qquad (5.2)$$

where, $\alpha$ and $\beta$ represent coefficients. In the proposed model, $\alpha = 0.001$ $and$ $\beta = 4$. By default the coefficient for the sparsity regularisation term is set as 1, but we choose a higher value to indicate that the value is not close and very sparse and differentiable. The other two notations $\varphi_{weights}$ $and$ $\varphi_{sparsity}$ represent L2 regularisation and sparsity regularisation and can be computed as:

$$\varphi_{sparsity} = \sum_{i=1}^{\rho^{(1)}} \rho \log\left(\frac{\rho}{\hat{\rho}_i}\right) + (1-\rho)\log\left(\frac{1-\rho}{1-\hat{\rho}_i}\right) \qquad (5.3)$$

$$\varphi_{weights} = \frac{1}{2} \sum_{l}^{2} \sum_{j}^{n} \sum_{i}^{k} \left(W_{ji}^{1}\right)^2 \qquad (5.4)$$

Here, $\rho$ is the desired value and $\hat{\rho}_i$ is the average activation value which can be expressed as:

$$\hat{\rho}_i = \frac{1}{n} \sum_{j=1}^{n} h(W_i^{(1)\,T} x_j + b_i^{(1)}) \qquad (5.5)$$

For training in the encoder we use a logistic sigmoid function and, in the decoder, we use a linear transfer function as shown in the equations below.

The encoder transfer function is defined as:

$$f(z) = \frac{1}{1 + e^{-z}} \qquad (5.6)$$

The decoder transfer function is defined as:

$$f(z) = z \qquad (5.7)$$

Now we get the deep MSP features for training the whole network by applying a softmax function.

To calculate the loss function we use the cross-entropy Equation 5.8:

$$E = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{k} t_{ij} \ln y_{ij} + \left(1 - t_{ij}\right) \ln(1 - y_{ij}) \qquad (5.8)$$

For an input vector $f$, $y_{ij}$ is the output vector for position i. k and n are the number of classes and training samples respectively.

For each superpixel in test images, we extract the MSP features in the same way as we extract from the training data. For each feature, we predict the class based on the maximum value of class probabilities.

## 5.3 Proposed Multi-Scale Perceptual Features

An overview of the proposed feature model is presented in Figure 5.2. One of the original contributions to this model is multi-scale perceptual features which are highlighted with the red box in the figure. In the proposed model, as in many other existing approaches, we use superpixels [156] rather than pixels as the basic units for visual feature extraction. Here every pixel in an image $I$ with width and height $W\ X\ H$ is assigned a discrete label $L$. The output image is denoted by $A \in L\ (WX\ H)$. Initially, we decompose the raw image $I$ into multi-scale superpixels using simple linear iterative clustering [157] $A: A = f\ (I, \theta)$ with an appropriate parameter $\theta$, where $\theta$ represents the ratio and number of superpixels that are used for decomposition. Here $A = [a1, a2, a3\ \ldots\ an], n = 70$. From those superpixels, we extract superpixel features and proposed multi-scale perceptual features.

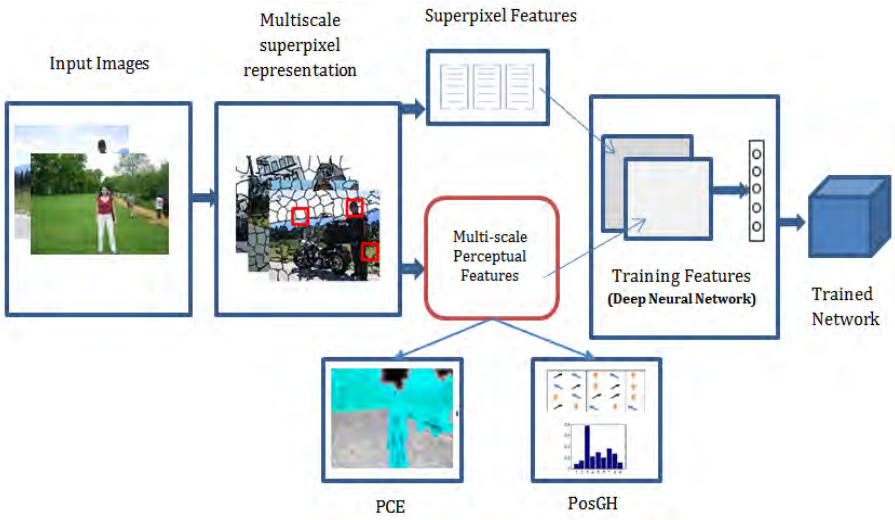



Figure 5.2 Proposed Multi-scale Perceptual Features

Existing feature extraction techniques for scene labelling have some limitations which we address here. One of the most popular methods for feature extraction is HOG, which has been successfully used in a wide variety of applications. One drawback of the HOG feature extraction technique is that the positions of pixel gradients are lost as it counts the sum of the magnitudes for different directions. As a

result, two superpixels with different gradient directions may have very similar histograms. The idea is illustrated in Figure 5.4, from which it is obvious that the gradient orientation is different in both cells, but the number of components is the same. While they have the same number of components, nevertheless the objects belong to different classes. This is the reason that the gradient direction plays a vital role in distinguishing between similar types of gradient histograms. Another vital feature is the similarity between two objects. Although they have the same position of HOG, they are differentiable with respect to their planes of orientation, which is achieved by introducing a plane consistency estimation concept in this section.

### 5.3.1 Position Gradient Histogram (PosGH)

Edge intensity orientations for local regions or patches are calculated based on the orientation of the gradients. For an image patch $a_1$ , its magnitude $m$ and orientation $\theta$ are calculated using Equations 5.9 and 5.10.

$$m(x,y) = \sqrt{(I(x+1,y) - I(x-1,y))^2 + (I(x,y+1) - I(x,y-1))^2} \qquad (5.9)$$

$$\theta(x,y) = \tan^{-1}\left(\frac{I(x,y+1) - I(x,y-1)}{I(x+1,y) - I(x-1,y)}\right) \qquad (5.10)$$

The inter-bin distance and the number of orientation bins used in the proposed model are as follows.

$$-\frac{\Pi}{2} < \theta < \frac{\Pi}{2} \qquad (5.11)$$

$$\left(\theta + \frac{\Pi}{2}\right) \div \Pi \times 9 \qquad (5.12)$$

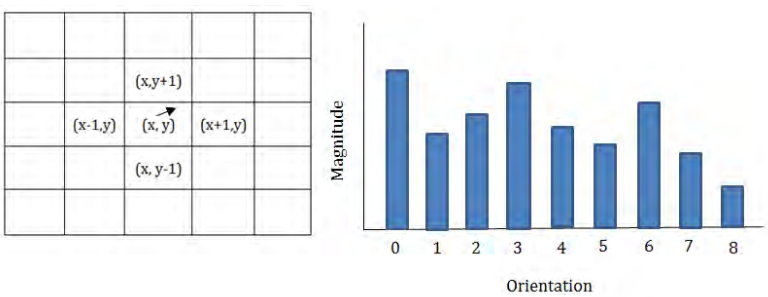



Figure 5.3 Example of Orientation Bins used in the Proposed Model

Figure 5.3 shows an example of bins using visualisation to understand the scene more specifically. Equation 5.13 shows how the normalisation is performed by moving one cell to the entire region.

$$\theta\ (\mathrm{n}) = \frac{\theta\ (\mathrm{n})}{\sqrt{\sum_{k=1}^{3*3*9} \theta\ (\mathrm{n})^2 + 1}} \qquad (5.13)$$

*where* $\theta$ (n) is the magnitude of each direction

As shown in Figure 5.4, a positioning technique is applied at the next stage, with edge intensity information to avoid the visually similar object identification and hence improve the classification accuracy. Figure 5.4 shows an example highlighting the problem we face with currently extracted information.

Assume that for a specific superpixel $a_1$ we have $(w1\ X\ h1)$ pixels. As we are dealing with each block individually within each superpixel, we have 9 small sub-blocks within each superpixel. Let $B$ be one of the blocks with $N$ pixels. As we are adding one of the additional features, so let $\theta(x, y, i)$ be the orientation on position *(x, y)* from the N pixels. We divide the orientation using the range shown in Equations 5.14 and 5.15. For eight orientations, we update the range each time with the interval shown in Equation 5.16.

$$Lower\ range\ =\ -pi + 2 * pi/9 \quad (5.14)$$

$$Upper\ Range\ =\ -pi + 2 * pi/9\ +\ 2 * pi/9 \quad (5.15)$$

$$Interval\ =\ 2 * pi/9 \quad (5.16)$$

Now we find the pixels within the range and the equations for finding the position are given by Equations 5.17 and 5.18.

$$P^r_{x,y} =\ R\ ||\ \theta(x, y, i) == Z(x, y, i) \quad (5.17)$$

$$P^c_{x,y} =\ C||\ \theta(x, y, i) == Z(x, y, i) \quad (5.18)$$

This new position with a gradient of histogram features is added as a new feature. Let $Z(x, y, i)$ be the range for finding position whereas $Z(x, y, i)$ is a member of $\theta(x, y, i)$. The value of $Z(x, y, i)$ is calculated from the lower range and upper range. Now for each row we get the positions $P^r_{x,y}$ from the specific orientation $Z(x, y, i)$. Similar things are done for column position $P^c_{x,y}$. To calculate the feature value we multiply the row position with the number of items in that row position and finally sum the values for the whole block to get the feature value $F^R_{x,y}$. This is also done for column position and we get the feature value $F^C_{x,y}$ and finally both values are added within the histogram of gradient values. Although the gradient of the histogram for the two charts is the same e.g. for 45° rotation the value is 6 in both cases. But they are varied position-wise and there is a significant change in value according to Equations 5.19 and 5.20 For example, in Figure 5.4, for the left histogram the row and column values are 15 and 15 respectively, but for the right histogram they are 14 and 16 respectively. Hence introducing the position creates a feature vector more powerful for differentiating similar objects.

$$F^R_{x,y} =\ \sum_{x=1}^{R_x} \sum_{y=1}^{R_y} C +\ x *\ P^r_{x,y} \quad (5.19)$$

$$F^C_{x,y} =\ \sum_{x=1}^{C_x} \sum_{y=1}^{C_y} C +\ y *\ P^c_{x,y} \quad (5.20)$$

$$F^{R}_{x,y}(left\ hist) = 1*1+2*2+3*2+4*1 = 15$$

$$F^{C}_{x,y}\ (left\ hist) = 1*2+2*1+3*1+4*2+5*0 = 15$$

$$F^{R}_{x,y}(right\ hist) = 1*+2*3+3*1+4*1 = 14$$

$$F^{C}_{x,y}(right\ hist) = 1*2+2*1+3*1+4*1+5*1 = 16$$

While we process the block, we move one cell to another cell as shown in Figure 5.4.

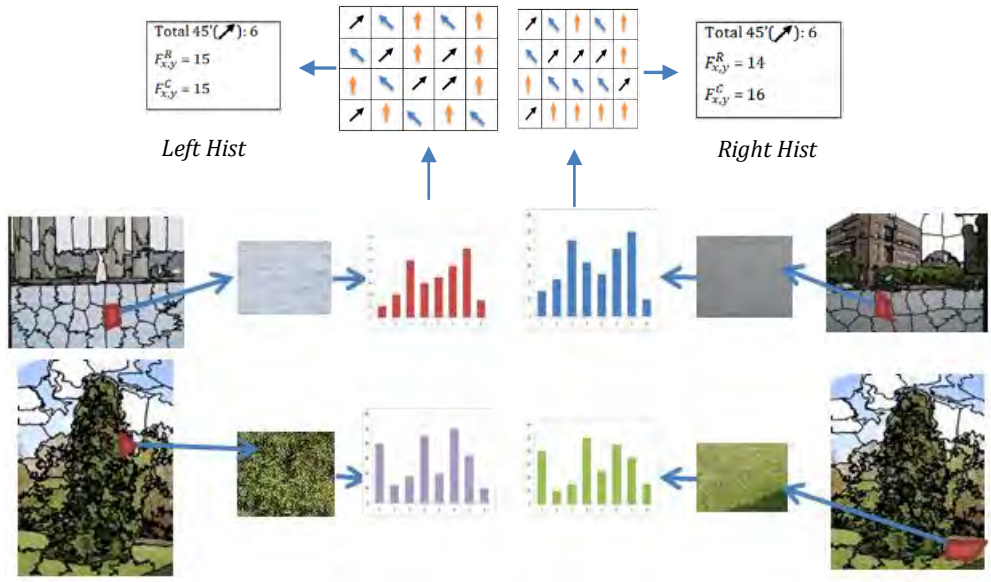



Figure 5.4 Illustration of using the Gradient Histogram to Distinguish Superpixels with Similar Visual Appearance

### 5.3.2 Plane Consistency Estimation (PCE)

The PCE technique is introduced to differentiate between vertical and horizontal objects. This helps to distinguish between numerous confusing objects like trees, grasses, roads and buildings that have a similar visual appearance but different directions. If we look at the superpixels for both tree and grass, in most cases they are identical. There is a similar problem in differentiating between roads and buildings. To solve the problem and separate confusing objects, we developed the PCE technique. The reasoning lies in the fact that tree orientation is always vertical while grass is on the same plane as the ground. This difference is also true for buildings and roads as well as some other objects. For an input image I, the outputs for vertical and horizontal objects are shown in Figure 5.5.

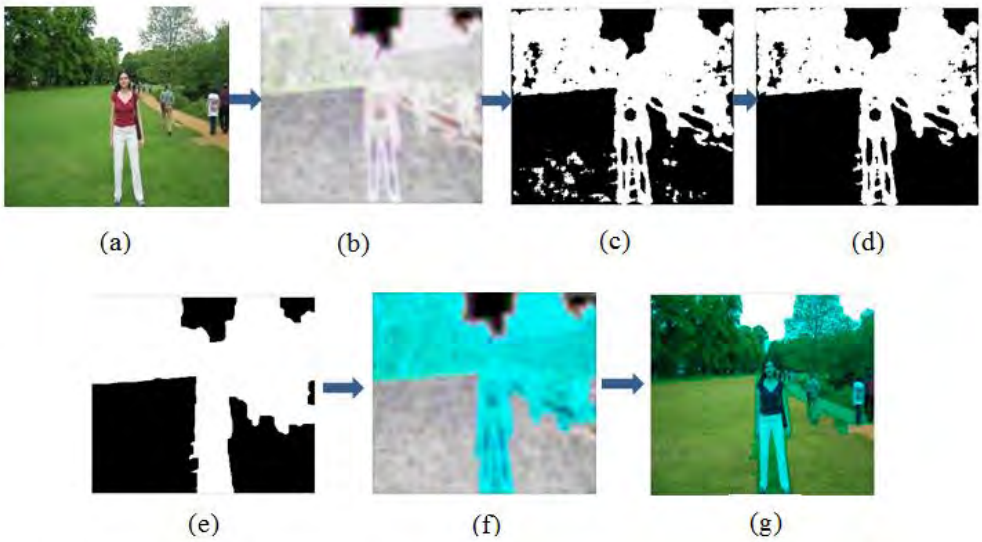



Figure 5.5 (a) Original Image (b) Entropy Image (c) Intensity Image (d) Binary Image after Opening Operations (e) Binary Image after Closing Operation (f) Label Probability Image (g) The Vertical Area Covered with Cyan Colour and Horizontal with Original Colour

Figure 5.5 shows all images after each operation. Figure 5.5(a) shows the original image. Figure 5.5(b) shows the image after applying an entropy function on the input image I. Here, we use output pixels to calculate entropy value for each input pixel *p(x, y)* by considering a 9-by-9 neighbourhood around the pixel of interest in the input image I and assigning the values for each pixel. Equation 5.21 illustrates the calculation process for entropy.

$$E = \sum_{i=1}^{n} P(x_i)I(x_i) = -\sum_{i=1}^{n} P(x_i).log_2 P(x_i) \qquad (5.21)$$

Here $E$ is the entropy value for an image $I$ and $P$ which contains the histogram counts for the image.

Later the entropy image is converted into an intensity image where each value represents intensities within the range between the minimum entropy value and maximum entropy value. The width and height of a pixel are determined as follows.

$$x = \frac{x(j) - x(i)}{size(C,2) - 1};\ \mathrm{j} = 2, \mathrm{i} = 1 \qquad (5.22)$$

$$y = \frac{y(j) - y(i)}{size(C,1) - 1};\ \mathrm{j} = 2, \mathrm{i} = 1 \qquad (5.23)$$

$$limits = \left[min\big(E(:)\big) \quad max\big(E(:)\big)\right] \qquad 5.(24)$$

$$delta = \frac{1}{limits(2) - limits(1)} \qquad (5.25)$$

$$G = x * delta + y * E - limit(1) * delta \ (5.26)$$

Figure 5.5(c) shows the intensity image, ranging from 0 (black) to 1 (white or full intensity).

$$\begin{cases} BW = 1\,; if\ p > T \\ BW = 0; if\ p < T \end{cases} \qquad (5.27)$$

Figure 5.5(d) shows the binary image after applying threshold operations on the intensity image. The threshold selection is an iterative process. The output image replaces all pixels in the input image with 0 or 1. If the input value is greater than 0.9 we replace it with the value 1 (white) and replace all other pixels with the value 0 (black). This is done by using the Equation 5.27. We apply opening and closing operations to determine the actual vertical and horizontal regions in the target image which are shown in Figures 5.5(e) and 5.5(f) respectively. Finally, for the best visualisation, we show the overall region in Figure 5.5(g) with 2 different colours. The vertical region is covered using the blue colour and the horizontal region with its original image colour. This gives an overall scenario for vertical and horizontal objects in an entire scene.

**Algorithm 1** Plane Consistency Estimation Calculation

**Input**:
***I***: Raw Image
***T***: Threshold for binary conversion
***C***: Connected component value
***hood***: Neighbouring Value

```
1.  for total number of image do
2.      calculate the size of image
3.      repeat
a.          E ← calculate entropy using equation 5.21
4.      until image size
5.      E_im ← converts matrix E to Intensity Image using
                equations 5.22,5.23 and 5.24
6.      if upper and lower limits are same
7.          E_im ← E;
8.      else
9.          compute delta using equation 5.25
10.         compute  G using equation 5.26 with linear
            combination
11.     repeat for each pixel in G
12.         calculate binary image using equation 5.27
13.         for each pixel binary image do
14.             calculate connected component
15.             calculate area
16.             idx ← CC.PixelIdxList(area >= C);
17.             idx ← vertcat(idx{:});
18.             BWao ← 0;
19.             BWao(idx) ←1;
20.         end for
21.     until  G1 < size(G)
22.         nhood ← true(9);
23.     for each BWao do
24.         calculate dilated image
            new BWao ← apply erosion on the dilated
                       image
25.     end for
26.     output pixel 1 with a vertical orientation and 0 with
        horizontal orientation
27.     for each superpixel do
28.         c1 ← count no of ones
29.         c2 ← count no of zeros
30.         if c1>c2
31.             vertical ←1;
32.     end for
33. end for
```

**Output**:
***Y***: Label probability after applying PCE

### 5.3.3 Other Features

It is often inadequate to use the superpixel area to describe the characteristic of individual objects. So, for a semantic representation of each superpixel, we incorporate two measurements: 1) perceptual features to differentiate visually similar objects; 2) global and local texture features for each superpixel which include the following.

1) SIFT features which reflect the invariant property for the superpixel (128 dimensions).
2) Multiple appearance features (colour, histogram), geometric features (area, eccentricity), texture features (LBP, energy), and location features.
3) RGB colour features for each channel having a total of three dimensions.
4) The Modified position of gradient histogram features (225 dimensions).
5) Local Binary Pattern features (59 dimensions).
6) Pixel height and centreline calculation in which the position feature is used having two dimensions.
7) PCE feature representation for the image.
8) Location feature which takes the location of each super-pixel in the whole image [158]. For this purpose, we separate the whole image into 6 X 6 blocks as shown in Figure 5.6 and determine the corresponding location of each superpixel and thus immediately know the position of each class within the image. The overall algorithm for extracting location features is shown in algorithm 2.

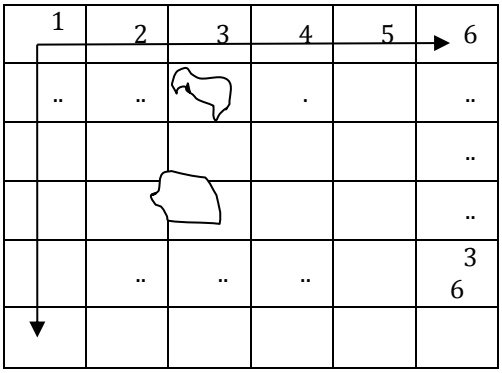



Figure 5.6 Superpixel Location Information

To do this, we need to know the x and y positions of the superpixel and based on the pixel locations, we determine the position within 36 blocks and assign a block value for each superpixel. Some points may overlap in 36 blocks but we take the majority pixels location within the block.

Suppose that one candidate image size is 240 X 320 pixels. As we divide the image into 6 X 6 blocks, so each block contains 40 X 53 pixels. For example, if the candidate superpixels x location is 165 and y

location is 201, then the location is 4. In Figure 5.6, we show two superpixels in different locations. One corresponds to location 9 and other belongs to 21. If we look at the second superpixel it overlaps on some other blocks, but as majority of the pixels fall within block 21, we assign the superpixel to location 21.

**Algorithm 2** Proposed Location Feature Calculation

***Input***: *Original Image and a superpixel block*

1. ***Initialisation***: *nblockcolumn = 6; nblockrow = 6;*
2. ***for*** *each super pixel from an image* ***do***
3. *initialise row, column minimum and maximum value*
4. *calculate size of image*
5. *update* $dcol \leftarrow fix(\frac{column\ no}{nblockcolumn})$

   $drow \leftarrow fix(\frac{row\ no}{nblockrow})$
6. ***repeat***
7. *[rr,cc] = ind2sub([nblockrow,nblockcolumn],index );*

   $block_{row_{\min(index)}} = (rr - 1) * drow + 1$

   $block_row_max(index) = rr * drow$ ;

   $block_{col_{\min(index)}} = (cc - 1) * dcol + 1$

   $block_col_max(index) = cc * dcol$ ;
8. ***until*** *nblockrow* nblockcolumn*
9. *V = zeros(Rs,Cs);*
10. *Connected Component* $\leftarrow$ *Binary connected component of BW;*

    *numPixels* $\leftarrow$ *from pixel list;*

    *Calculate the maximum value from the pixel list;*

    *Assign V*$\leftarrow$*1;*
11. *Calculate centroid for the area;*

    *xc = stat(1).Centroid(2);*

    *yc = stat(1).Centroid(1);*
12. *actual_x = r_min + xc; actual_y = c_min + yc;*

    ***repeat***
13. ***if****((block_row_min(loc)<=uint8(actual_x)&block_row_max (loc)>=uint8(actual_x))&&*

    *(block_col_min(loc)<=uint8(actual_y) && block_col_max(loc)>=uint8(actual_y)))*

    *location(k) = loc;*

    *location_feature = loc;*
14. ***until*** *(loc<36)*
15. ***end for***

## 5.4 Experiments and Results

The effectiveness of the proposed approach is evaluated on three benchmark datasets and compared with existing approaches in this section.

### 5.4.1 Datasets

We conducted experiments on three benchmark datasets [1] to evaluate the performance of the proposed approach. The datasets are Stanford Background [147], MSRC Dataset [144], SIFT Flow [159] and vegetation dataset [131].

The Stanford dataset [147] contains 715 images of outdoor scenes composed of 8 classes: sky, tree, grass, road, water, building, mountain and foreground. The size of each image on the dataset is 320×240 pixels. Some examples of the Stanford dataset along with ground truth and output from the system are discussed and illustrated in Section 5.5.2.

The MSRC dataset [144] is a another popular benchmark dataset for scene labelling, which consists of 591 images including 23 classes: “building”, “grass”, “tree”, “cow”, “horse”, “sheep”, “sky”, “mountain”, “aeroplane”, “water”, “face”, “car”, “bicycle”, “flower”, “sign”, “bird”, “book”, “chair”, “road”, “cat”, ”dog”, ”body” and “boat”. During the training and testing for direct comparison, the pixels with labelled “void” are not considered.

The SIFT flow 33-class dataset [159] is composed of 2688 images, that have been labelled 33 different number and colours. There are 33 semantic categories of objects, including: “sky”, “building”, “mountain”, “tree”, “road”, “sea”, “field”, “grass”, “river”, “plant”, “car”, “sand”, “rock”, “sidewalk”, “window”, “desert”, “door”, “bridge”, “person”, “fence”, “balcony”, “crosswalk”, “staircase”, “awning”, “sign”, “streetlight”, “boat”, “pole”, “sun”, “bus”, “bird”, “moon” and “cow”. The annotated dataset mostly contains outdoor scenes. There is also an “unlabelled” class which is impossible to determine from the scene and was not considered during the training and testing.

The vegetation dataset contains six classes and consists of 50 annotated images collected from various parts of Queensland. The classes included in this datasets are grass (brown and green grass), road, soil, tree, and sky. There are also some unlabeled pixels and put black colour is placed for those unknown regions.

### 5.4.2 Evaluation Criteria

We evaluate our approach with respect to pixel-wise accuracy for each class. The pixel-wise accuracy is calculated as follows:

$$PA = \frac{(TP+TN)}{(TP+FP+TN+FN)} \quad (5.28)$$

Here TP means true positive, which is the number of correct pixels in both reference image and ground truth image. TN means true negative which denotes the number of pixels correctly detected as an unchanged area in both reference image and ground truth image. FP means false positive which incorrectly identified the unchanged pixels as changed pixels, on the other hand, and FN means false negative which incorrectly rejected the changed pixels that were undetected.

### 5.4.3 Results

In order to train the neural network for scene labelling, there are two types of approaches reported in the literature [145] [160]. These approaches are patch-wise and image–wise. In patch-wise approaches, patches are cropped randomly from a sample image whose labels are determined from the central pixels. In image-wise approaches, the whole image is taken as an input and the output. Research shows that image-wise training suffers seriously from an over-fitting [161] problem. The problem occurs due to the strong correlation between the pixels in an image. Hence in our experiments we use the patch-wise approach with two modifications. In the proposed model, the first modification is that instead of choosing the patch randomly we use superpixels to determine the patch. Another modification made in the proposed model is that instead of choosing the central pixel as a label for the corresponding patch, we take the dominating label within the patch. This means the maximum label within a patch determines the label for the corresponding patch.

Experiments are performed using 5-fold cross-validation [147]. For the Stanford background dataset, 572 images were used for training and the remaining 143 were used for testing. For the MSRC dataset, among 591 images, 473 images were used for training and 118 images were used for testing. For the SIFT flow dataset the evaluation procedure is different. We followed the procedure introduced in [145]. From 2688 images, 2488 images were used for training data and the remaining 200 images used for testing data. In all of the datasets, the image sizes are different. In the Stanford dataset, most of the images have a size of 320 X 240 pixels, whereas in the MSRC dataset most of the images have the size 320 X 213 pixels. In the SIFT flow dataset all images have an equal size of 256 X 256 pixels. All of our training and testing processes are run on a PC with dual core i5 with 2.66 GHz CPU and 8 GB RAM.

## 5.5 Result Analysis

### 5.5.1 Result on Stanford Background Dataset

The result obtained on the Stanford background dataset using the proposed deep learning model is shown in Figure 5.7.

| | Sky | Tree | Road | Grass | Water | Building | Mountain | Foreground |
|---|---|---|---|---|---|---|---|---|
| 1 | **94.5** | 0.6 | 0.4 | 0.5 | 2.6 | 1.3 | 0.0 | 0.1 |
| 2 | 3.9 | **83.2** | 0.4 | 8.2 | 0.7 | 1.1 | 1.6 | 0.9 |
| 3 | 0.0 | 0.6 | **88.1** | 0.3 | 4.5 | 1.5 | 2.3 | 2.7 |
| 4 | 0.0 | 8.7 | 3.5 | **83.3** | 0.6 | 0.9 | 2.3 | 0.7 |
| 5 | 1.2 | 0.2 | 14.6 | 2.1 | **72.9** | 2.6 | 4.1 | 2.3 |
| 6 | 0.3 | 0.4 | 9.7 | 0.7 | 3.1 | **79.1** | 2.1 | 4.6 |
| 7 | 4.1 | 11.6 | 5.6 | 5.5 | 10.8 | 13.9 | **37.1** | 11.6 |
| 8 | 0.0 | 1.8 | 4.7 | 1.1 | 0.6 | 10.3 | 5.1 | **76.4** |

Figure 5.7 Accuracy on Stanford Background Dataset

From Figure 5.7, it is clear that we obtained high accuracy for visually similar objects using the proposed model, e.g. tree and grass, road, and water as well as building and road. For tree and grass, we achieved more than 80 percent accuracy in both scenarios and the confusion matrix in Figure 5.7 shows that less confusion occurred during the classification of the two objects. From Figure 5.7, it is obvious that less misclassifications occurred for each object so we achieved an overall accuracy of 82.64 percent where the fivefold accuracies are 80.70 (fold 1), 83.21 (fold 2), 83.03 (fold 3), 82.37 (fold 4) and 83.90 (fold 5) respectively. Figure 5.8 shows the parameter selection for both training and testing phases using the MSP features. We chose 20, 70 and 150 for the number of hidden neurons. For epochs, we set 200, 300 and 500 respectively. The learning rate was set as 0.1 and the target RMS error was 0.001. We tested with a number of 70 and 120 superpixels. We achieved the best accuracy using 20 hidden neurons with 300 epochs and 70 superpixels. The scenario is described in Figure 5.8 using observation 8.

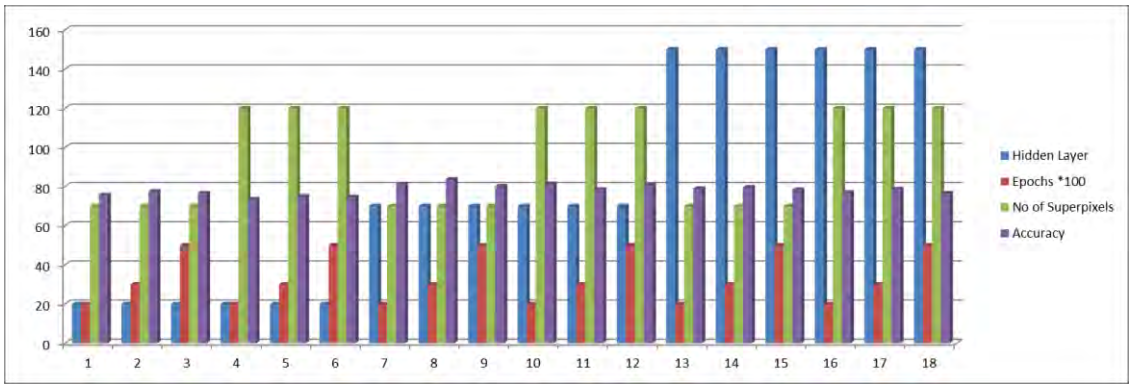



Figure 5.8 Parameter Observations for Neural Network on Stanford Background Dataset

Figure 5.9 shows sample output on the Stanford dataset using the proposed model. The figure visually depicts that the proposed model provides good classified output for individual objects. From the pixel-wise class label accuracy, it is depicted that the proposed approach shows promising performance in respect to ground truths and we obtained more than 80 percent accuracy for trees, roads, grasses, buildings, and sky.

In addition to the good results described above, Figure 5.10 also presents some bad labelling outputs produced by the proposed model. The analysis of results showed that water and mountain were misclassified as road and tree. This problem occurred because of the small patch size because it is difficult to detect and recognise objects from small patches if the illumination varies and no such sample patches are used in training data.

In addition to the good results described above, we also present some bad labelling outputs in Figure 5.10 produced by the proposed model. The analysis of results showed that water and mountain were misclassified as road and tree. This problem occurred because of the small patch size it is difficult to detect and recognise objects from small patches if the illumination varies and no such sample patches are used in training data.

### 5.5.2 Comparative Analysis on the Stanford Background Dataset

We also made comparisons with the overall accuracy of existing methods as shown in Table 5.1 and which indicates our proposed approach outperforms other existing methods.

Table 5.1 Performance Comparision on Stanford Dataset

| **Method** | **Accuracy** |
|---|---|
| Gould *et al.,* 2009 [147] | 76.4% |
| Tighe *et al.*, 2010 [162] | 77.5% |
| Munoz *et al.*, 2010 [150] | 76.9% |
| Kumar *et al.*, 2010 [163] | 79.4% |
| Socher *et al.,* 2012 [164] | 78.1% |
| Lempitsky *et al.,* 2011 [165] | 81.9% |
| Farabet *et al.*, 2013 [145] | 78.8% |
| Pinheiro et al. 2013 [166] | 80.2% |
| Byeon *et al.,* 2015 [167] | 78.56% |
| **Proposed Method** | **82.64%** |

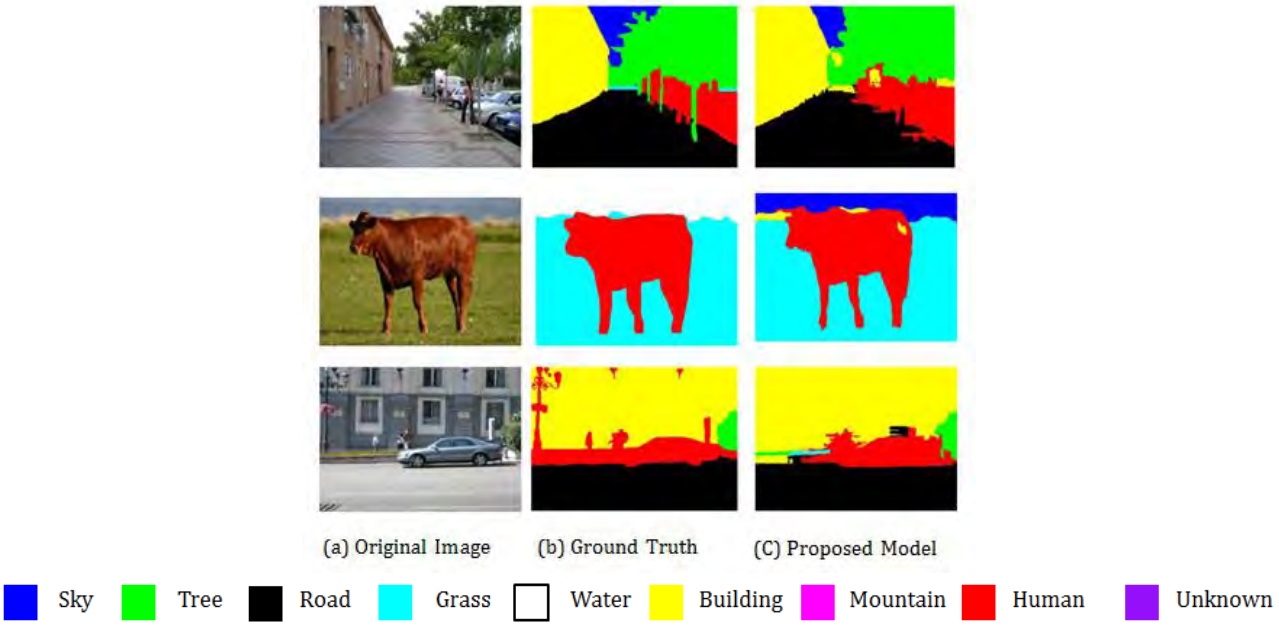



Figure 5.9 Labelling result on Stanford Background Dataset

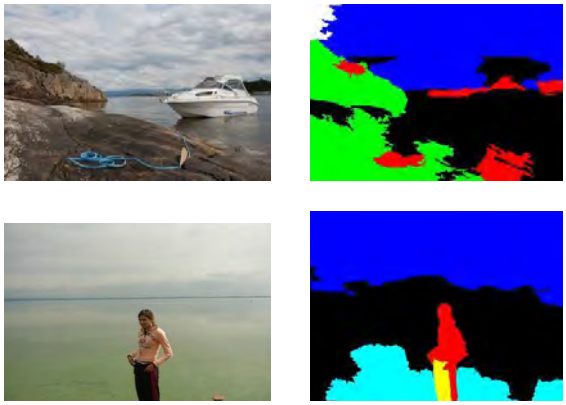


Figure 5.10 Example of Bad Labelling on Stanford Background Dataset

### 5.5.3 Result on MSRC Dataset

Figure 5.11 demonstrates the overall pixel-wise accuracy for each object on the MSRC dataset. Our main focus was on building, grass, tree and water. We obtained promising results compared to other existing methods. Figure 5.12 shows some labelling outputs for the MSRC dataset.

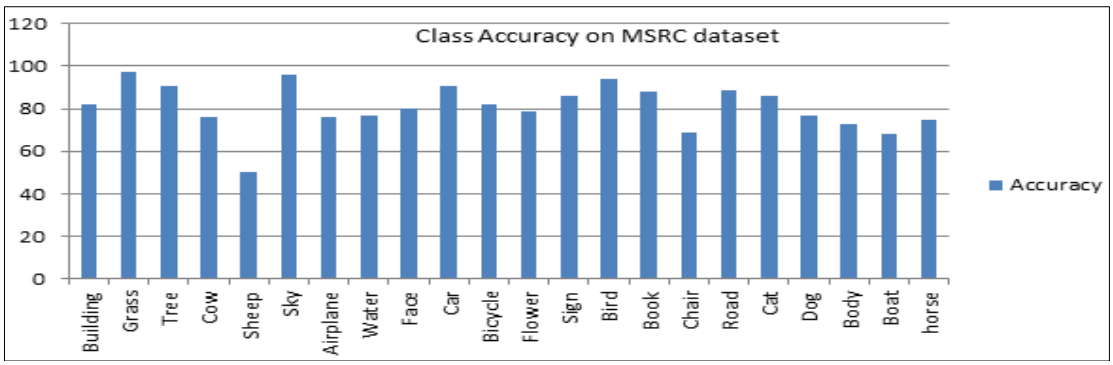



Figure 5.11 Class-wise Accuracy on MSRC Dataset

A remarkable fact of the proposed method is it outperformed than other existing methods. This superior performance was achieved because our new MSP features have helped in solving some of the existing misclassification issues such as tree misclassified as grass, road misclassified as building, water misclassified as road and so on.

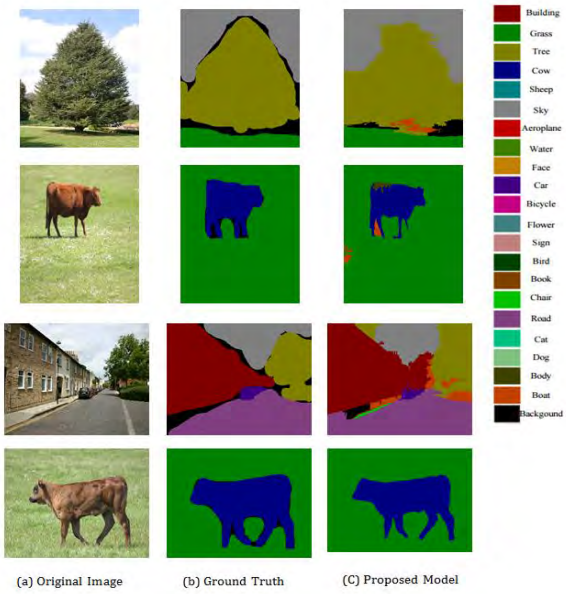



Figure 5.12 Labelling Results on MSRC Dataset

### 5.5.4 Comparative Analysis on MSRC Dataset

Table 5.2 shows the detailed comparison with existing methods on the MSRC dataset.

Table 5.2 Performance Comparison on MSRC Dataset

| **Method** | **Accuracy** |
|---|---|
| Gould *et al.*, 2008 [168] | 64.3% |
| Lempitsky *et al.*, 2011 [165] | 78.2% |
| Krähenbühl *et al.* 2012 [169] | 78.3% |
| Zhou *et al.*, 2013 [170] | 76.4% |
| Zhu *et al.*, 2012 [171] | 74.1% |
| Farabet *et al.*, 2013 [145] | 74.6% |
| Sharma *et al.*, 2015 [172] | 77.6% |
| Long *et al.*, 2015 [173] | 77.9% |
| Zhou *et al.*, 2016 [1] | 79.4% |
| **Proposed Method** | **83.00%** |

### 5.5.5 Result on SIFT Flow Dataset

Finally, we applied our method to the SIFT flow dataset and the overall accuracy and performance comparisons are listed in Table 5.3. Eigen and Fergus also achieved comparable accuracy with our proposed approach. The dataset is more challenging as it has 33 different objects, so we need to expand our proposed approach by analysing confused objects and adding more deep features.

Selected examples of labelling results from the SIFT flow dataset are shown in Figure 5.13.

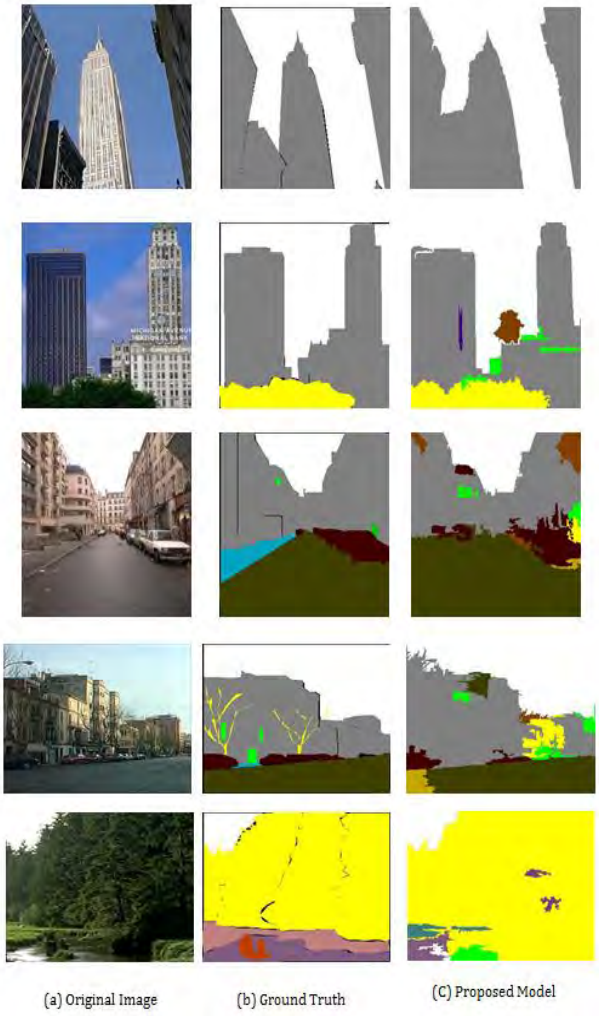



Figure 5.13 Labelling Results on SIFT Flow Dataset

### 5.5.6 Comparative Analysis on the SIFT Flow Dataset

Table 5.3 Performance Comparison on SIFT Flow Dataset

| Method | Accuracy |
|---|---|
| Liu *et al.,* 2009 [159] | 74.75% |
| Tighe *et al.,* 2010 [162] | 76.90% |
| Eigen *et al.*, 2012 [174] | 77.10% |
| Pinheiro *et al.,* 2013 [166] | 76.20% |
| **Proposed Method** | **77.43%** |

### 5.5.7 Result on Vegetation Dataset

Results from the vegetation dataset have been listed to prove the effectiveness of the proposed multiscale perceptual model. Figure 5.14 shows some experimental output using the proposed model. Figure 5.15 shows the class-wise accuracy.

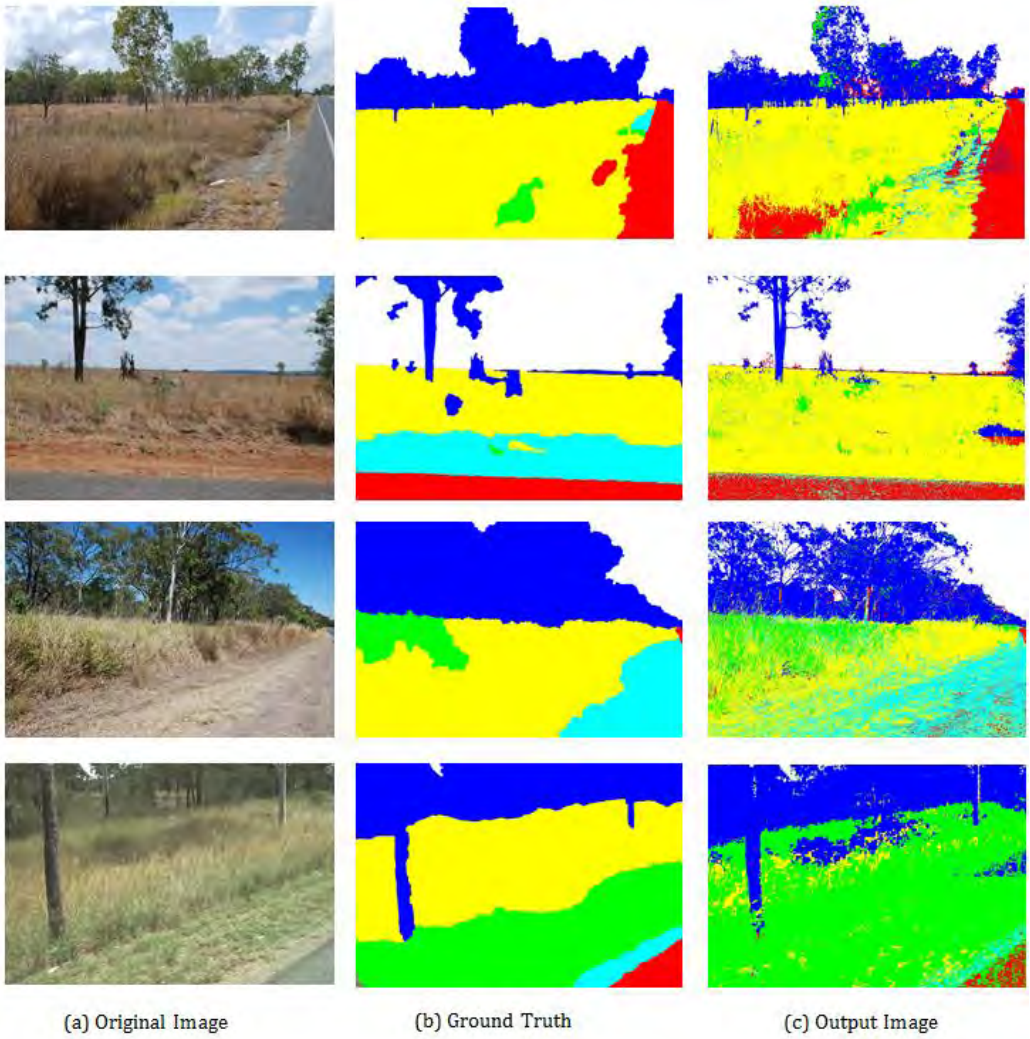



Figure 5.14 Labelling Results on Vegetation Dataset

From the experimental results and class-wise accuracy chart, it is clear that we achieved compatible accuracy compared to the previous approach. Adding the proposed multiscale perceptual feature improved overall accuracy from 78% to 82%. Table 5.4 shows the comparison chart with the existing methods.

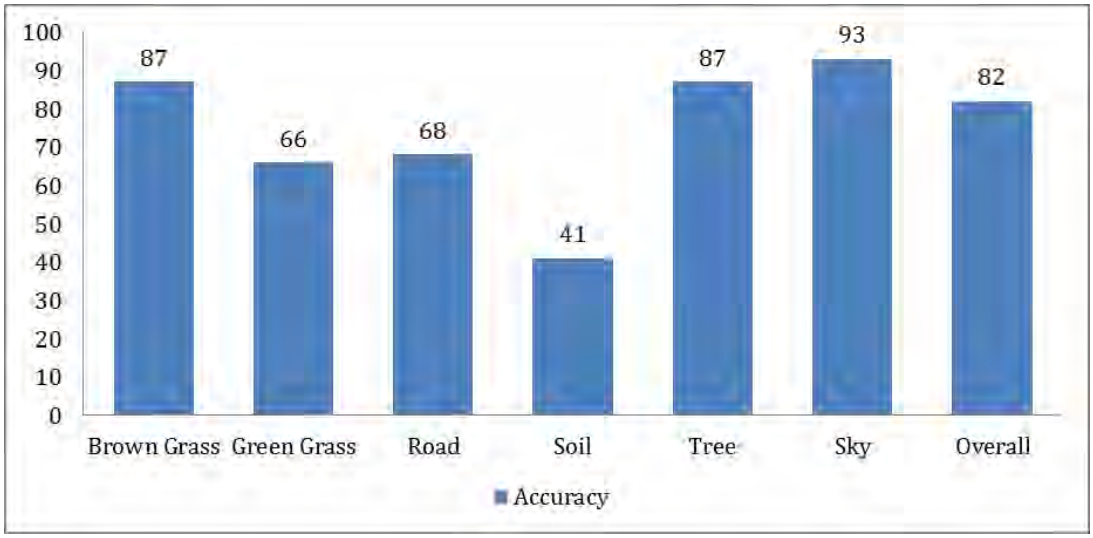



Figure 5.15 Class-wise Accuracy on Vegetation Dataset

### 5.5.8 Comparative Analysis on Vegetation Dataset

Table 5.4 Performance Comparison on Vegetation Dataset

| Serial No | Approach | Pixel-wise Accuracy |
|---|---|---|
| 1 | Colour Feature [109] | 70.47 % |
| 2 | Texture Feature [131] | 73.3 % |
| 3 | Colour Texture Feature [131] | 74.5 % |
| 4 | Novel Quantisation Feature and Neural Network (QFNN) based approach | 78.01 % |
| **5** | **Proposed MSP Method** | **82.00%** |

## 5.6 Summary

A new deep learning model to generate multi-scale perceptual features for scene labelling has been presented. The main contribution of this work is the introduction of new deep multi-scale perceptual features which take into account both Position Gradient Histogram and Plane Consistency Estimation for scene labelling.

It has been shown that multi-scale perceptual features based on position gradient histogram and plane consistency concepts can produce a powerful representation of visually similar objects. The proposed model was trained on superpixels with fully labelled training images in a supervised manner to learn appropriate features. The proposed model has been evaluated on three well-known benchmark datasets and compared with existing approaches. The experimental results demonstrate that the proposed model consistently achieved higher accuracy on all three datasets, confirming that the proposed deep learning model is promising for scene labelling tasks.

Although the proposed model achieved better performance than other approaches, there are still some drawbacks. The possible reasons for those drawbacks are shadows and different illumination conditions. Future work should extend this research by further investigating misclassified cases to find appropriate solutions. One possible way to find a solution is the use of shadow removal techniques. The other possible extension of this research work is the optimisation of the number of layers and network parameters in a deep learning architecture.

# Chapter 6 Conclusion

This thesis investigated feature extraction and classification techniques for roadside object classification and scene labelling. The integration of features and classifiers was used to analyse the roadside video data and provide results for object detection and classification. Both designed and learnt features were intensively studied and successfully applied to the roadside video data analysis application. This chapter presents the contributions and findings of the thesis and proposes future research directions.

## 6.1 Contributions and Findings

This thesis proposed and investigated various feature extraction and classification techniques and evaluated their performances on data collected from various parts of Queensland. The proposed feature extraction techniques were also evaluated on three benchmark datasets (e.g. Stanford dataset, MSRC dataset and SIFT flow dataset) and one local dataset (Vegetation dataset). The proposed techniques achieved better performance than existing techniques. Different types of feature extraction techniques were proposed and applied to vegetation area classification. The results from a series of experiments prove its effectiveness and demonstrate the importance of using the proposed techniques. The contribution of the automated strategy was combining geometry, appearance and spatial texture information rather than using traditional features. Furthermore, spatial image descriptors were extracted using a new mathematical model based on the image properties. Multiple features were extracted and combined to form the final feature descriptors and used for object detection. The evaluations were performed using the real dataset.

The major contributions and findings of this thesis are summarised below:

- CBP based Feature Extraction Technique

Initially, a co-occurrence of binary pattern based feature extraction was proposed to determine the dense and sparse regions from a cropped vegetation dataset. Several feature extraction techniques, namely local binary pattern, gray-level co-occurrence, and Fast Fourier Transform based individual feature extraction were applied individually and performances were recorded. Later, an ensemble model combining the local binary pattern and gray-level co-occurrence based approach was developed and it showed better performance than the existing methods. The best accuracy obtained using that proposed CBP technique was 92.72%. Another investigation undertaken was a classifier selection

process which compared the performances of individual classifiers and recorded the accuracy. Then an ensemble classifier was proposed which showed better performance than any individual classifier. Using Support Vector Machine and Artificial Neural Network, accuracy was 91.72% whereas k-Nearest Neighbour (k-NN) gave accuracy of 90.00%. But the CBP ensemble model gives a higher accuracy of 92.72% [84].

- ❒ DCC based Feature Extraction Technique

In our previous method, a very small dataset was used to evaluate the proposed feature extraction technique. To validate performance and check robustness, the dataset was increased by adding different illumination condition based cropped data. While trying to validate the performance on the enhanced dataset, overall accuracy found to have dropped. It is of great importance when developing a new feature to address the concern of how to improve the performance. Hence, different colour channel based feature extraction techniques were investigated. HIS, HSV, and YCbCr based colour channels were tested with the cropped region and the best performance was achieved using the YCbCr colour model. Two new feature extraction techniques were developed using the YCbCr colour model. The distance feature gives the information about the present pixel condition within a block whereas the cross-correlation feature gives the relationship between different blocks. Both features help to identify sparseness within one cropped region. After introducing these new features, overall accuracy improved to 93% for the enriched dataset [98].

- ❒ QFNN based Feature Extraction Technique

In the next phase of feature extraction, the entire image was considered instead of just the cropped region and the number of object classes was increased. Instead of grass regions, other objects such as trees, roads, soil and sky regions were also considered for feature extraction. Individual pixel values from different regions were considered and new feature extraction techniques were introduced. Initially, existing features with R, G, B colour channels and their combination were extracted and tested with the created dataset [115]. The performance was very low and lots of misclassifications occurred. The reason for misclassifications was investigated and a quantisation based feature extraction was introduced. Instead of using the individual colour channels, the proposed method identifies the colour channel sequences and applies most significant bit quantisation and colour channel sequence which help to overcome the challenges with existing methods. The proposed method achieves 78.01% accuracy for overall region identification. The results were compared with existing feature extraction techniques and performances were measured. A comparative performance analysis proves the effectiveness of the proposed technique.

- DCF based Feature Extraction Technique

Identification of grass regions was not sufficient to find the fire risk regions as the risk depends on the depth and length of grasses. The quantisation feature was extended by introducing the new Directional Connectivity Feature to estimate the grass density. A new dataset was created and experiments were performed for evaluation. The approach differs from the previous approach as it also considers the orientation and calculates the connectivity on the vertical direction. The proposed techniques achieve better accuracy for all types of grasses and the results were compared with the Gabor-based feature extraction technique. Accuracy for dense grass was 81.80%, whereas for moderate and sparse grass, they were 65% and 78.24% respectively. The overall accuracy was 75.01% where the Gabor-based technique achieved only 62%. The performance was evaluated in terms of computational time. The proposed technique showed competitive performance.

- MSP Feature Extraction Technique

Multi-scale perceptual feature fusion plays an important role in increasing the reliability of the extracted information for robust operational performance and decision making in object classification. The use of superpixel features and perceptual features collectively has strong links with human perception. The multi-scale perceptual feature was investigated and effectively applied to the standard dataset and the experimental results demonstrated improvement over the superpixel features. Using the proposed MSP feature obtained 82.64% accuracy on the Stanford dataset and 83.00% and 77.43% for MSRC and SIFT flow datasets respectively. The results were also compared on the vegetation dataset and achieved 82.00%. The proposed technique showed better performance than many existing methods [129].

The findings of this research have provided answers to the research questions in this thesis. The relevant research questions and findings are presented as follows::

- **What is the best way to separate the Region of Interest (ROI) or roadside objects from the video data?**

  This thesis has investigated a number of ways to segment the objects from the video data. The techniques were thresholding based technique, graph-based technique, pixel-based technique, and superpixel-based technique. Preliminary research started with the thresholding based technique and achieved 70% accuracy for the region of interest separation. Later, the graph-based technique was introduced and accuracy improved to 75%. Due to misclassifications on pixel labels, the pixel-based segmentation technique was introduced and achieved an accuracy of 78%. Pixel based processing was too slow; hence superpixel-based segmentation was introduced and gave compatible performance for both vegetation and benchmark dataset. The

accuracy with this technique increased to 82%. Hence, the best way to separate the region of interest is the superpixel-based technique. The derived findings provided further understanding of region of interest separation.

- **Why are existing feature extraction techniques not suitable to identify vegetation regions from roadside video data?**

  The thesis investigated existing feature extraction techniques and found the problem with existing features. For example, a combination of GLCM, FFT, SIFT feature can differentiate narrow and broad weed but failed to differentiate different grass regions because of their irregular shape and distribution in the field. Moreover, Histogram of gradient feature was useful for differentiating different objects but failed when intra-class variation became small. Some recent vegetation classification techniques use LiDAR data for grass and non-grass region identification. As proposed research used only RGB information these features were not suitable to identify vegetation regions from the roadside video data.

- **What is the most suitable feature extraction technique or combination of techniques that can identify roadside objects like trees, grasses, shrubs, or any other objects on the roadside?**

  The thesis investigated a number of feature extraction techniques related to object classification and vegetation classification. These included Local Binary Pattern, Gray-Level Co-Occurrence, Colour Feature, Entropy, Location, Scale Invariant Feature Transform, Histogram of Gradient etc. The proposed CBP used the combination of local binary pattern and gray-level co-occurrence and achieved 92% accuracy. In addition, DCC used distance and cross-correlation feature along with CBP feature and achieved 93% accuracy. Moreover, the QFNN based technique used several features which included $R, G, B, |R-G|, |R-B|, |G-B|, \frac{1}{3}(R+G+B), MSB\ pattern, sequence, H, S, V$ and achieved 78.01% accuracy for grass, tree, soil, road, and sky region identification. The accuracy obtained was promising, but it failed to differentiate dense and sparse region. The sparseness and denseness depend on the grass height and depth. Hence, a new technique for density estimation was introduced. Directional connectivity feature for depth calculation helps to differentiate between dense and sparse region on complex environments. The accuracy for dense, moderate and sparse grass region identification was 75.01% which is comparatively higher than the Gabor based technique and helps to identify risk location from roadside video images. Finally, the multiscale perceptual feature for overall object classification from scene images was introduced and effectively applied on vegetation classification. To improve the overall classification accuracy the proposed technique also used some existing features. During the selection process from existing features, several combinations were used and the best features were incorporated. Those features were SIFT

(128 dimensions), multiple appearance features (colour, histogram), geometric features (area, eccentricity), texture features (LBP, energy), location features, and RGB colour features for each channel which in total of three dimensions. Moreover, modified position of gradient histogram features (225 dimensions), Local Binary Pattern features (59 dimensions), pixel height and centreline calculation in which position feature is used, which has two dimensions. Proposed MSP features showed promising performance on benchmark dataset and accuracy obtained for the vegetation dataset was 82.00%, for the Stanford dataset was 82.64%, for MSRC was 83.00% and for SIFT flow was 77.43%. This feature helps to identify objects like trees, grasses, shrubs or any other roadside objects.

- **What are the most suitable parameters for a classifier that can effectively classify the objects from video data in terms of efficiency and accuracy?**

  The thesis investigated a number of classifiers, examining the performance with different parameters like kernel function, the number of hidden neurons, the number of iterations, root mean square error, training algorithm, etc. As various parameters were changed, results with individual classifiers were investigated and finally the best parameters were chosen. The proposed CBP technique used an ensemble model, but DCC and QFNN used single SVM and neural network classifiers. The proposed MSP feature extraction technique used deep learning architecture. Deep learning strategy achieved better performance than any other technique with 70 hidden neurons, 300 iterations and an RMS error of 0.001.

- **How can the risk location be identified from the video data?**

  The last research question that the thesis needs to address is the risk location from the video data. As one of the possible applications of the research is identifying the risk locations from the engaged video, it is necessary to point out those specific regions. For each frame, the proposed DCF based feature created fifteen windows and analysed each window and calculated the feature value from the grass region. This value indicates the fire risk level as being high, moderate or low. From the video data, the total length of time of the video recording and related GPS coordinates are known. Moreover, the distance of overall road travel distance is known and, from the converted video, the length of road covered can be calculated. Thus the high fire risk locations can be generated from the video.

## 6.2 Future Research Directions

This thesis provides important knowledge about the feature descriptors and their applications to real-world problems. The proposed techniques can be further extended as described below.

- CBP, DCC, and QFNN based Feature Extraction Technique

  The co-occurrence of binary pattern based technique using grayscale images loses the colour information. Colour plays an important role in object differentiation. Hence further investigation is needed to improve the feature. The dataset used to evaluate the proposed technique was small. Further investigation can include more dataset content for training and testing.

  The distance and cross-correlation based feature extraction technique also depends on grayscale images and YCbCr images. Research shows that YCbCr images help with differentiating objects under different illumination conditions. Still, shadows create a big impact on overall classification. Further investigation is needed with other colour channels and performance needs to be compared. The dataset used for this technique was also small. To make the system more robust, larger datasets need to be included.

  Quantisation based feature extraction technique introduced two new features which show better performance on a small dataset. Further investigation is needed to validate the performance by applying it on the standard benchmark dataset. In future, results can be compared with deep learning feature.

- DCF based Feature Extraction Technique

  The directional connectivity based feature extraction technique helps to differentiate between the depths of grass in the images. Further investigation is needed to find whether it is useful for differentiating vertical and horizontal images. If it is possible to further extend the method for plane estimation, this will be helpful in solving many real-world problems.

- MSP Feature Extraction Technique

  Multiscale perceptual features help to improve the overall scene classification accuracy. The proposed technique was verified on three benchmark datasets and a local dataset. Further investigation is needed to check the performance on other benchmark datasets. Moreover, only deep learning architecture was used for classification. Further investigation is needed to determine whether or not an ensemble classifier can produce better results.

- Make the system real time

  The proposed techniques have been designed for offline use. Data collected from the video camera is stored and analysed later for further decisions. As roadside conditions keep changing, the decisions might not be accurate. Therefore, this research can be extended by including some real-time video processing which would be beneficial for the industry.

- Shadow removal from the vegetation image

  The proposed techniques do not remove shadows and other illumination changes as all the videos are collected during the daytime with good illumination conditions. Further investigation is needed to consider those areas which are affected by shadow. Existing shadow removal algorithms will not be applicable as the image properties are different.

- 3D feature extraction from the LiDAR image

  The proposed techniques used 2-D image data. They can be extended to use 3-D data. It might be found to be beneficial to use 3-D data as this might allow calculating the grass height much easier than from RGB image data.

# References


[1] Q. Zhou, B. Zheng, W. Zhu, and L. Jan Latecki, "Multi-scale context for scene labeling via flexible segmentation graph," *Pattern Recognition,* 2016.

[2] L. Zhang and B. Verma, "Class-semantic textons with superpixel neighborhoods for natural roadside vegetation classification," in *Digital Image Computing: Techniques and Applications (DICTA), 2015 International Conference on*, 2015, pp. 1-8.

[3] Y.-J. Zhang, "An overview of image and video segmentation in the last 40 years," in *Advances in Image and Video Segmentation*, ed: IGI Global, 2006, pp. 1-16.

[4] Wesley E. Snyder and H. Qi, "Machine Vision," *Cambridge, UK, Cambridge University Press,* 2004.

[5] R. M. Rangayyan, "Biomedical image analysis," *Biomedical Engineering Series, CRC Press LLC, ISBN 0-8493-9695-6,* 2005.

[6] R. C. Gonzalez and R. E. Woods, "Digital Image Processing. ," *Prentice-Hall, Inc., 3rd edition, ISBN 0-201-18075-8. International Edition,* 2002.

[7] X. Yaowen, L. Linlin, W. Haoyu, and Z. Xiaojiong, "The application of threshold methods for image segmentation in oasis vegetation extraction," in *Geoinformatics, 2010 18th International Conference on*, 2010, pp. 1-4.

[8] M. Montalvo, J. M. Guerrero, J. Romeo, L. Emmi, M. Guijarro, and G. Pajares, "Automatic expert system for weeds/crops identification in images from maize fields," *Expert Systems with Applications,* vol. 40, pp. 75-82, 2013.

[9] N. OTSU, "A threshold selection method from gray-level histograms," *Systems, Man and Cybernetics, IEEE Transactions on,* vol. 9, pp. 62-66, 1979.

[10] M. Guijarro, G. Pajares, I. Riomoros, P. J. Herrera, X. P. Burgos-Artizzu, and A. Ribeiro, "Automatic segmentation of relevant textures in agricultural images," *Computers and Electronics in Agriculture,* vol. 75, pp. 75-83, 2011.

[11] M. H. Siddiqi, I. Ahmad, and S. B. Sulaiman, "Edge link detector based weed classifier," in *Digital Image Processing, 2009 International Conference on*, 2009, pp. 255-259.

[12] A. Bleau and L. J. Leon, "Watershed-based segmentation and region merging," *Computer Vision and Image Understanding,* vol. 77, pp. 317-370, 2000.

[13] N. Hieu Tat, M. Worring, and R. Van den Boomgaard, "Watersnakes: energy-driven watershed segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 25, pp. 330-342, 2003.

[14] Z. Li, R. Hayward, Y. Liu, and R. Walker, "Spectral–texture feature extraction using statistical moments with application to object-based vegetation species classification," *International Journal of Image and Data Fusion,* vol. 2, pp. 347-361, 2011.

[15] Z. Li, "Aerial image analysis using spiking neural networks with application to power line corridor monitoring " *Ph.D. thesis,* 2011.

[16] F. Kanda, M. Kubo, and K.-i. Muramoto, "Watershed segmentation and classification of tree species using high resolution forest imagery," in *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*, 2004, pp. 3822-3825.

[17] M. P. Ponti, "Segmentation of low-cost remote sensing images combining vegetation indices and mean shift," *Geoscience and Remote Sensing Letters, IEEE,* vol. 10, pp. 67-70, 2013.

[18] L. Zheng, D. Shi, and J. Zhang, "Segmentation of green vegetation of crop canopy images based on mean shift and Fisher linear discriminant," *Pattern Recognition Letters,* vol. 31, pp. 920-925, 2010.

[19] L. Zheng, J. Zhang, and Q. Wang, "Mean-shift-based color segmentation of images containing green vegetation," *Computers and Electronics in Agriculture,* vol. 65, pp. 93-98, 2009.

[20] Z. Xiuying and F. Xuezhi, "Detecting urban vegetation from IKONOS data using an object-oriented approach," in *Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International*, 2005, pp. 1475-1478.

[21] D. Omerevi, R. Perko, A. T. Targhi, J.-O. Eklundh, and A. Leonardis, "Vegetation segmentation for boosting performance of mser feature detector," in *Computer Vision Winter Workshop*, 2008, pp. 17-23.

[22] D. A. Ridel, P. Y. Shinzato, and D. F. Wolf, "A clustering-based obstacle segmentation approach for urban environments," in *2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*, 2015, pp. 265-270.

[23] T. Blaschke, "Object based image analysis for remote sensing," *ISPRS journal of photogrammetry and remote sensing,* vol. 65, pp. 2-16, 2010.

[24] M. Kima, B. Xu, and M. Maddena, "Object-based vegetation type mapping from an orthorectified multispectral IKONOS image using ancillary information," ed: GEOBIA, 2008.

[25] C. Iovan, D. Boldo, and M. Cord, "Detection, characterization, and modeling vegetation in urban areas from high-resolution aerial imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 1, pp. 206-213, 2008.

[26] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003, pp. 10-17 vol.1.

[27] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic photo pop-up," *ACM Trans. Graph.,* vol. 24, pp. 577-584, 2005.

[28] Y. Wang and Q. Zhao, "Superpixel tracking via graph-based semi-supervised SVM and supervised saliency detection," in *Multimedia and Expo (ICME), 2015 IEEE International Conference on*, 2015, pp. 1-6.

[29] J. Tighe and S. Lazebnik, "SuperParsing: scalable nonparametric image parsing with superpixels," in *Computer Vision – ECCV 2010: 11th European Conference on Computer Vision,*

*Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V*, K. Daniilidis, P. Maragos, and N. Paragios, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 352-365.

[30] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 2015, pp. 1356-1363.

[31] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. S*, et al.*, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 34, pp. 2274-2282, 2012.

[32] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, pp. 603-619, 2002.

[33] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision,* vol. 59, pp. 167-181, 2004.

[34] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: fast superpixels using geometric flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 2290-2297, 2009.

[35] J. Vadher, "Normalized cut based image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, issue 8, pp. 888-905, 2015.

[36] S. Jianbo and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, pp. 888-905, 2000.

[37] Z. Li, X. M. Wu, and S. F. Chang, "Segmentation using superpixels: A bipartite graph partitioning approach," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 789-796.

[38] A. P. Moore, S. J. D. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1-8.

[39] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V*, K. Daniilidis, P. Maragos, and N. Paragios, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 211-224.

[40] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 13, pp. 583-598, 1991.

[41] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Computer Vision – ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV*, D. Forsyth, P. Torr, and A. Zisserman, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 705-718.

[42] L. Zhang, B. Verma, and D. Stockwell, "Spatial contextual superpixel model for natural roadside vegetation classification," *Pattern Recognition, pp. 40-47,* 2016.

[43] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *Systems, Man and Cybernetics, IEEE Transactions on,* vol. SMC-3, pp. 610-621, 1973.

[44] Qian Yu, Peng Gong, Nick Clinton, Greg Biging, a. Maggi Kelly, and D. Schirokauer, "Object-based detailed vegetation classification with airborne high spatial resolution remote sensing imagery," *Photogrammetric Engineering & Remote Sensing,* vol. Vol. 72, No. 7, pp. pp. 799–811, 2006.

[45] K. H. Ghazali, M. F. Mansor, M. M. Mustafa, and A. Hussain, "Feature extraction technique using discrete wavelet transform for image classification," in *Research and Development, 2007. SCOReD 2007. 5th Student Conference on*, 2007, pp. 1-4.

[46] M. M. Mustafa, A. Hussain, K. H. Ghazali, and S. Riyadi, "Implementation of image processing technique in real time vision system for automatic weeding strategy," in *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, 2007, pp. 632-635.

[47] K. H. Ghazali, S. Razali, M. M. Mustafa, and A. Hussain, "Machine vision system for automatic weeding strategy in oil palm plantation using image filtering technique," in *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*, 2008, pp. 1-5.

[48] L. Wu and Y. Wen, "Weed/corn seedling recognition by support vector machine using texture features," *African Journal of Agricultural Research,* vol. 4, pp. 840-846, 2009.

[49] Z. Li, R. Hayward, J. Zhang, H. Jin, and R. Walker, "Evaluation of spectral and texture features for object-based vegetation species classification using support vector machines": PhD thesis, QUT, 2010.

[50] Z. Li, Y. Liu, R. Hayward, and R. Walker, "Empirical comparison of machine learning algorithms for image texture classification with application to vegetation management in power line corridors": PhD thesis, QUT, 2010.

[51] L. Tang, L. F. Tian, and B. L. Steward, "Classification of broadleaf and grass weeds using Gabor wavelets and an artificial neural network," *Transactions of the ASAE,* vol. 46, p. 1247, 2003.

[52] A. Mustapha and M. M. Mustafa, "Development of a real-time site sprayer system for specific weeds using gabor wavelets and neural networks model," *Proceedings of the Malaysia Science and Technology Congress, Malaysia, pp. 406–413.,* 2005.

[53] A. J. Ishak, A. Hussain, and M. M. Mustafa, "Weed image classification using gabor wavelet and gradient field distribution," *Computers and Electronics in Agriculture,* vol. 66, pp. 53-61, 2009.

[54] F. Ahmed, A. S. M. H. Bari, A. Shihavuddin, H. A. Al-Mamun, and P. Kwan, "A study on local binary pattern for automated weed classification using template matching and support vector machine," in *Computational Intelligence and Informatics (CINTI), 2011 IEEE 12th International Symposium on*, 2011, pp. 329-334.

[55] F. Ahmed, M. H. Kabir, S. Bhuyan, H. Bari, and E. Hossain, "Automated weed classification with local pattern-based texture descriptors," *Int. Arab J. Inf. Technol.,* vol. 11, pp. 87-94, 2014.

[56] W. Li, J. Du, and B. Yi, "Study on classification for vegetation spectral feature extraction method based on decision tree algorithm," in *Image Analysis and Signal Processing (IASP), 2011 International Conference on*, 2011, pp. 665-669.

[57] H. T. Søgaard, "Weed classification by active shape models," *Biosystems Engineering,* vol. 91, pp. 271-281, 2005.

[58] T. Rumpf, C. Römer, M. Weis, M. Sökefeld, R. Gerhards, and L. Plümer, "Sequential support vector machine classification for small-grain weed species discrimination with special regard to cirsium arvense and galium aparine," *Computers and Electronics in Agriculture,* vol. 80, pp. 89-96, 2012.

[59] A. Tellaeche, G. Pajares, X. P. Burgos-Artizzu, and A. Ribeiro, "A computer vision approach for weeds identification through Support Vector Machines," *Applied Soft Computing,* vol. 11, pp. 908-915, 2011.

[60] M. H. Siddiqi, I. Ahmad, and S. B. Sulaiman, "Weed recognition based on erosion and dilation segmentation algorithm," in *Education Technology and Computer, 2009. ICETC '09. International Conference on*, 2009, pp. 224-228.

[61] A. Juraiza Ishak, M. M. Mustafa, and A. Hussain, "Gradient field distribution and grey level co-occurrence matrix techniques for automatic weed classification," in *Mechatronics and Its Applications, 2008. ISMA 2008. 5th International Symposium on*, 2008, pp. 1-5.

[62] A. Juraiza Ishak, M. M. Mustafa, N. M. Tahir, and A. Hussain, "Weed detection system using support vector machine," in *Information Theory and Its Applications, 2008. ISITA 2008. International Symposium on*, 2008, pp. 1-4.

[63] A. Juraiza Ishak, S. S. Mokri, M. M. Mustafa, and A. Hussain, "Weed detection utilizing quadratic polynomial and ROI techniques," in *Research and Development, 2007. SCOReD 2007. 5th Student Conference on*, 2007, pp. 1-5.

[64] F. Ahmed, H. A. Al-Mamun, A. S. M. H. Bari, E. Hossain, and P. Kwan, "Classification of crops and weeds from digital images: A support vector machine approach," *Crop Protection,* vol. 40, pp. 98-104, 2012.

[65] I. Ahmad, A. Muhamin Naeem, M. Islam, and A. Bin Abdullah, "Statistical based real-time selective herbicide weed classifier," in *Multitopic Conference, 2007. INMIC 2007. IEEE International*, 2007, pp. 1-4.

[66] U. Watchareeruetai, Y. Takeuchi, T. Matsumoto, H. Kudo, and N. Ohnishi, "Computer vision based methods for detecting weeds in lawns," in *Cybernetics and Intelligent Systems, 2006 IEEE Conference on*, 2006, pp. 1-6.

[67] L. Xianfeng and C. Zhong, "Weed identification based on shape features and ant colony optimization algorithm," in *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, 2010, pp. V1-384-V1-387.

[68] L. Xianfeng and Z. Weixing, "Multi- feature fusion in weed recognition based on dempster-shafer's theory," in *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, 2010, pp. 127-130.

[69] G. Overett and L. Petersson, "Large scale sign detection using HOG feature variants," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, 2011, pp. 326-331.

[70] J. Gleason, A. V. Nefian, X. Bouyssounousse, T. Fong, and G. Bebis, "Vehicle detection from aerial imagery," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 2065-2070.

[71] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, pp. 886-893.

[72] Y.-F. Pan, X. Hou, and C.-L. Liu, "A robust system to detect and localize texts in natural scene images," in *Document Analysis Systems, 2008. DAS'08. The Eighth IAPR International Workshop on*, 2008, pp. 35-42.

[73] G. Cheng, J. Han, L. Guo, Z. Liu, S. Bu, and J. Ren, "Effective and efficient midlevel visual elements-oriented land-use classification using VHR remote sensing images," *Geoscience and Remote Sensing, IEEE Transactions on,* vol. 53, pp. 4238-4249, 2015.

[74] J. Zhang, K. Huang, Y. Yu, and T. Tan, "Boosted local structured hog-lbp for object localization," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011, pp. 1393-1400.

[75] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1999, pp. 1150-1157 vol.2.

[76] G. T. Flitton, T. P. Breckon, and N. M. Bouallagu, "Object recognition using 3D SIFT in complex CT volumes," in *BMVC*, 2010, pp. 1-12.

[77] G. Flitton, T. P. Breckon, and N. Megherbi, "A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery," *Pattern Recognition,* vol. 46, pp. 2420-2436, 2013.

[78] S. Se, D. Lowe, and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 2001, pp. 2051-2058.

[79] I. Gordon and D. G. Lowe, "What and where: 3D object recognition with accurate pose," in *Toward category-level object recognition*, ed: Springer, 2006, pp. 67-82.

[80] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification via pLSA," in *European conference on computer vision*, 2006, pp. 517-530.

[81] A. Bosch, A. Zisserman, X. Mu, x0F, and oz, "Scene classification using a hybrid generative/discriminative approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, pp. 712-727, 2008.

[82] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool, "Modeling scenes with local descriptors and latent aspects," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) vol. 1*, 2005, pp. 883-890.

[83] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2003, vol. 2, pp. 264-271.

[84] L. J. Zhao, P. Tang, and L. Z. Huo, "Land-use scene classification using a concentric circle-structured multiscale bag-of-visual-words model," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 7, pp. 4620-4631, 2014.

[85] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 24, pp. 971-987, 2002.

[86] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE transactions on image processing,* vol. 19, pp. 1635-1650, 2010.

[87] M. A. Akhloufi and A. Bendada, "Locally adaptive texture features for multispectral face recognition," in *Systems man and cybernetics (SMC), 2010 IEEE international conference on*, 2010, pp. 3308-3314.

[88] M. Li, S. Zang, B. Zhang, S. Li, and C. Wu, "A review of remote sensing image classification techniques: The role of spatio-contextual information," *European Journal of Remote Sensing,* vol. 47, pp. 389-411, 2014.

[89] S. Z. Li, *Markov random field modeling in image analysis*: Springer Science & Business Media, 2009.

[90] B. Zhang, S. Li, X. Jia, L. Gao, and M. Peng, "Adaptive markov random field approach for classification of hyperspectral imagery," *IEEE Geoscience and Remote Sensing Letters,* vol. 8, pp. 973-977, 2011.

[91] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proceedings of the IEEE,* vol. 101, pp. 652-675, 2013.

[92] G. Moser, S. B. Serpico, and J. A. Benediktsson, "Land-cover mapping by Markov modeling of spatial–contextual information in very-high-resolution remote sensing images," *Proceedings of the IEEE,* vol. 101, pp. 631-651, 2013.

[93] B. Zhang, S. Li, C. Wu, L. Gao, W. Zhang, and M. Peng, "A neighbourhood-constrained k-means approach to classify very high spatial resolution hyperspectral imagery," *Remote sensing letters,* vol. 4, pp. 161-170, 2013.

[94] C. Li, J. Yin, and J. Zhao, "Extraction of urban vegetation from high resolution remote sensing image," in *Computer Design and Applications (ICCDA), 2010 International Conference on*, 2010, pp. 403-406.

[95] S. Chowdhury, B. Verma, and D. Stockwell, "A novel texture feature based multiple classifier technique for roadside vegetation classification," *Expert Systems with Applications,* vol. 42, pp. 5047-5055, 2015.

[96] A. M. Cingolani, D. Renison, M. R. Zak, and M. R. Cabido, "Mapping vegetation in a heterogeneous mountain rangeland using landsat data: an alternative method to define and classify land-cover units," *Remote Sensing of Environment,* vol. 92, pp. 84-97, 2004.

[97] P. Kamavisdar, S. Saluja, and S. Agrawal, "A survey on image classification approaches and techniques," *International Journal of Advanced Research in Computer and Communication Engineering* vol. 2, pp. 1005-1009, 2013.

[98] L. Tang, L. Tian, and B. L. Steward, "Classification of broadleaf and grass weeds using Gabor wavelets and an artificial neural network," *Transactions of the ASAE,* vol. 46, p. 1247, 2003.

[99] J. M. Guerrero, M. Guijarro, M. Montalvo, J. Romeo, L. Emmi, A. Ribeiro*, et al.*, "Automatic expert system based on images for accuracy crop row detection in maize fields," *Expert Systems with Applications,* vol. 40, pp. 656-664, 2013.

[100] J. Romeo, G. Pajares, M. Montalvo, J. M. Guerrero, M. Guijarro, and J. M. de la Cruz, "A new Expert System for greenness identification in agricultural images," *Expert Systems with Applications,* vol. 40, pp. 2275-2286, 2013.

[101] G. Jiang, Z. Wang, and H. Liu, "Automatic detection of crop rows based on multi-ROIs," *Expert Systems with Applications,* vol. 42, pp. 2429-2441, 2015.

[102] X. P. Burgos-Artizzu, A. Ribeiro, M. Guijarro, and G. Pajares, "Real-time image processing for crop/weed discrimination in maize fields," *Computers and Electronics in Agriculture,* vol. 75, pp. 337-346, 2011.

[103] I. Harbas and M. Subasic, "Detection of roadside vegetation using features from the visible spectrum," in *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*, 2014, pp. 1204-1209.

[104] I. Harbas and M. Subasic, "Motion estimation aided detection of roadside vegetation," in *Image and Signal Processing (CISP), 2014 7th International Congress on*, 2014, pp. 420-425.

[105] I. Harbas and M. Subasic, "CWT-based detection of roadside vegetation aided by motion estimation," in *Visual Information Processing (EUVIP), 2014 5th European Workshop on*, 2014, pp. 1-6.

[106] A. Wendel and J. Underwood, "Self-supervised weed detection in vegetable crops using ground based hyperspectral imaging," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5128-5135.

[107] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Information fusion,* vol. 6, pp. 63-81, 2005.

[108] L. Lam and S. Suen, "Application of majority voting to pattern recognition: an analysis of its behavior and performance," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 27, pp. 553-568, 1997.

[109] S. Chowdhury and B. Verma, "A novel feature extraction technique to retrieve vegetation class for fire risk assessment," in *Signal Processing and Communication Systems (ICSPCS), 2014 8th International Conference on*, 2014, pp. 1-6.

[110] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 38, pp. 142-158, 2016.

[111] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 17-24.

[112] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," *IEEE transactions on pattern analysis and machine intelligence,* vol. 37, pp. 2071-2084, 2015.

[113] I. Harbaš and M. Subašić, "Motion estimation aided detection of roadside vegetation," in *Image and Signal Processing (CISP), 2014 7th International Congress on*, 2014, pp. 420-425.

[114] D.-V. Nguyen, L. Kuhnert, and K. D. Kuhnert, "Spreading algorithm for efficient vegetation detection in cluttered outdoor environments," *Robotics and Autonomous Systems,* vol. 60, pp. 1498-1507, 2012.

[115] S. Herman, J. Janssen, E. Bellers, and J. Wendorf, "Automatic segmentation-based grass detection for real-time video," ed: Google Patents, 2004.

[116] B. Zafarifar and P. H. de With, "Grass field detection for TV picture quality enhancement," in *2008 Digest of Technical Papers-International Conference on Consumer Electronics*, 2008, pp. 1-2.

[117] A. Bhandari, A. Kumar, and G. Singh, "Improved feature extraction scheme for satellite images using NDVI and NDWI technique based on DWT and SVD," *Arabian Journal of Geosciences,* vol. 8, pp. 6949-6966, 2015.

[118] D. M. Bradley, R. Unnikrishnan, and J. Bagnell, "Vegetation detection for driving in complex environments," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 503-508.

[119] D.-V. Nguyen, L. Kuhnert, and K. D. Kuhnert, "Structure overview of vegetation detection. A novel approach for efficient vegetation detection using an active lighting system," *Robotics and Autonomous Systems,* vol. 60, pp. 498-508, 2012.

[120] X. Ren, L. Bo, and D. Fox, "Rgb-(d) scene labeling: features and algorithms," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 2759-2766.

[121] M. Ye, Z. Cao, Z. Yu, and X. Bai, "Crop feature extraction from images with probabilistic superpixel Markov random field," *Computers and Electronics in Agriculture,* vol. 114, pp. 247-260, 2015.

[122] L. Nanni and A. Lumini, "Heterogeneous bag-of-features for object/scene recognition," *Applied Soft Computing,* vol. 13, pp. 2171-2178, 2013.

[123] C. Wang, X. Li, X. Zhou, A. Wang, and N. Nedjah, "Soft computing in big data intelligent transportation systems," *Applied Soft Computing,* vol. 38, pp. 1099-1108, 2016.

[124] M. Pérez-Ortiz, J. Peña, P. Gutiérrez, J. Torres-Sánchez, C. Hervás-Martínez, and F. López-Granados, "A semi-supervised system for weed mapping in sunflower crops using unmanned aerial vehicles and a crop row detection method," *Applied Soft Computing,* vol. 37, pp. 533-544, 2015.

[125] N. S. Mishra, S. Ghosh, and A. Ghosh, "Fuzzy clustering algorithms incorporating local information for change detection in remotely sensed images," *Applied Soft Computing,* vol. 12, pp. 2683-2692, 2012.

[126] S. Chowdhury, B. Verma, M. Tom, and M. Zhang, "Pixel characteristics based feature extraction approach for roadside object detection," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1-8.

[127] P. Bendich, E. Gasparovic, J. Harer, R. Izmailov, and L. Ness, "Multi-scale local shape analysis and feature selection in machine learning applications," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1-8.

[128] M. Kristan, V. S. Kenk, S. Kovačič, and J. Perš, "Fast image-based obstacle detection from unmanned surface vehicles," *IEEE transactions on cybernetics,* vol. 46, pp. 641-654, 2016.

[129] B. Ahn, "Real-time video object recognition using convolutional neural network," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1-7.

[130] M. Abdechiri and K. Faez, "Efficacy of utilizing a hybrid algorithmic method in enhancing the functionality of multi-instance multi-label radial basis function neural networks," *Applied Soft Computing,* vol. 34, pp. 788-798, 2015.

[131] L. Zhang, B. Verma, and D. Stockwell, "Class-semantic color-texture textons for vegetation classification," in *International Conference on Neural Information Processing*, 2015, pp. 354-362.

[132] C. Paris and L. Bruzzone, "A novel technique for tree stem height estimation by fusing low density LiDAR data and optical images," in *Geoscience and Remote Sensing Symposium (IGARSS), 2013 IEEE International*, 2013, pp. 3022-3025.

[133] H. Balzter, C. S. Rowland, and P. Saich, "Forest canopy height and carbon estimation at Monks Wood National Nature Reserve, UK, using dual-wavelength SAR interferometry," *Remote Sensing of Environment,* vol. 108, pp. 224-239, 2007.

[134] N. Pham Minh, Z. Bin, and C. Yan, "Forest height estimation from PolInSAR image using adaptive decomposition method," in *Signal Processing (ICSP), 2012 IEEE 11th International Conference on*, 2012, pp. 1830-1834.

[135] P. Gärtner, M. Förster, A. Kurban, and B. Kleinschmit, "Object based change detection of central asian tugai vegetation with very high spatial resolution satellite imagery," *International Journal of Applied Earth Observation and Geoinformation,* vol. 31, pp. 110-121, 2014.

[136] Q. Chen, "Retrieving vegetation height of forests and woodlands over mountainous areas in the Pacific Coast region using satellite laser altimetry," *Remote Sensing of Environment,* vol. 114, pp. 1610-1627, 2010.

[137] H. Fan, Y. Cong, and Y. Tang, "Object detection based on scale-invariant partial shape matching," *Machine Vision and Applications,* vol. 26, pp. 711-721, 2015.

[138] G. Zhang, S. Ganguly, R. R. Nemani, M. A. White, C. Milesi, H. Hashimoto*, et al.*, "Estimation of forest aboveground biomass in California using canopy height and leaf area index estimated from satellite data," *Remote Sensing of Environment,* vol. 151, pp. 44-56, 2014.

[139] F. A. Andaló, G. Taubin, and S. Goldenstein, "Efficient height measurements in single images based on the detection of vanishing points," *Computer Vision and Image Understanding,* vol. 138, pp. 51-60, 2015.

[140] S. Chowdhury, B. Verma, and L. Zhang, "Position gradient and plane consistency based feature extraction," *23rd International Conference on Neural Information Processing (ICONIP 2016),* 2016.

[141] P. Zhou, Z. Lin, and C. Zhang, "Integrated low-rank-based discriminative feature learning for recognition," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 27, pp. 1080-1093, 2016.

[142] M. Gong, J. Liu, H. Li, Q. Cai, and L. Su, "A multiobjective sparse feature learning model for deep neural networks," *Neural Networks and Learning Systems, IEEE Transactions on,* vol. 26, pp. 3263-3277, 2015.

[143] Q. Zhu, L. Shao, X. Li, and L. Wang, "Targeting accurate object extraction from an image: A comprehensive study of natural image matting," *Neural Networks and Learning Systems, IEEE Transactions on,* vol. 26, pp. 185-207, 2015.

[144] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision,* vol. 81, pp. 2-23, 2009.

[145] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 35, pp. 1915-1929, 2013.

[146] S. Bu, P. Han, Z. Liu, and J. Han, "Scene parsing using inference Embedded Deep Networks," *Pattern Recognition, pp.56-65,* 2016.

[147] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 1-8.

[148] D. Grangier, L. Bottou, and R. Collobert, "Deep convolutional networks for scene parsing," in *ICML 2009 Deep Learning Workshop*, 2009.

[149] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision,* vol. 60, pp. 91-110, 2004.

[150] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked hierarchical labeling," in *Computer Vision–ECCV 2010*, 2010, pp. 57-70.

[151] H. Goh, N. Thome, M. Cord, and J. H. Lim, "Learning deep hierarchical visual feature coding," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 25, pp. 2212-2225, 2014.

[152] J. Chorowski and J. M. Zurada, "Learning understandable neural networks with nonnegative weight constraints," *Neural Networks and Learning Systems, IEEE Transactions on,* vol. 26, pp. 62-69, 2015.

[153] M. Gong, J. Liu, H. Li, Q. Cai, and L. Su, "A multiobjective sparse feature learning model for deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 26, pp. 3263-3277, 2015.

[154] R. Chakraborty and N. R. Pal, "Feature selection using a neural framework with controlled redundancy," *Neural Networks and Learning Systems, IEEE Transactions on,* vol. 26, pp. 35-50, 2015.

[155] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1," *Vision Research,* vol. 37, pp. 3311-3325, 1997.

[156] L. Zhang, B. Verma, D. Stockwell, and S. Chowdhury, "Aggregating pixel-level prediction and cluster-level texton occurrence within superpixel voting for adaptive roadside vegetation classification," *International Joint Conference on Neural Networks (IJCNN 2016),* pp. 3249-3255, 2016.

[157] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 34, pp. 2274-2282, 2012.

[158] L. Zhang, B. Verma, D. Stockwell, and S. Chowdhury, "Spatially constrained location prior for scene parsing," *International Joint Conference on Neural Networks (IJCNN 2016),* pp. 1480-1486, 2016.

[159] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing: label transfer via dense scene alignment," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 1972-1979.

[160] A. Sharma, O. Tuzel, and M.-Y. Liu, "Recursive context propagation network for semantic scene labeling," in *Advances in Neural Information Processing Systems*, 2014, pp. 2447-2455.

[161] M. Liang, X. Hu, and B. Zhang, "Convolutional neural networks with intra-layer recurrent connections for scene labeling," in *Advances in Neural Information Processing Systems*, 2015, pp. 937-945.

[162] J. Tighe and S. Lazebnik, "Superparsing: scalable nonparametric image parsing with superpixels," in *Computer Vision–ECCV 2010*, 2010, pp. 352-365.

[163] M. P. Kumar and D. Koller, "Efficiently selecting regions for scene understanding," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 3217-3224.

[164] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3d object classification," in *Advances in Neural Information Processing Systems*, 2012, pp. 665-673.

[165] V. Lempitsky, A. Vedaldi, and A. Zisserman, "Pylon model for semantic segmentation," in *Advances in neural information processing systems*, 2011, pp. 1485-1493.

[166] P. H. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene parsing," *International Conference on Machine Learning, pp. 121-128,* 2013.

[167] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki, "Scene labeling with lstm recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3547-3555.

[168] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *International Journal of Computer Vision,* vol. 80, pp. 300-316, 2008.

[169] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *Advance Neural Inormation Processing System, pp.14-20,* 2012.

[170] Q. Zhou, J. Zhu, and W. Liu, "Learning dynamic hybrid Markov random field for image labeling," *Image Processing, IEEE Transactions on,* vol. 22, pp. 2219-2232, 2013.

[171] L. Zhu, Y. Chen, Y. Lin, C. Lin, and A. Yuille, "Recursive segmentation and recognition templates for image parsing," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 34, pp. 359-371, 2012.

[172] A. Sharma, O. Tuzel, and D. W. Jacobs, "Deep hierarchical parsing for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 530-538.

[173] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431-3440.

[174] D. Eigen and R. Fergus, "Nonparametric image parsing using adaptive neighbor sets," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 2799-2806.