Table of Contents

List of Figures

List of Tables

List of Abbreviations

DARPA	Defense Advanced Research Projects Agency
LIDAR	LIght Detection And Ranging
RADAR	RADio Detection And Ranging
GMM	Gaussian Mixture Modeling
MLP	Multi Layer Perceptrons
LTP	Local Ternary Pattern
MRF	Markov Random Field
SURF	Speeded Up Robust Features
BOW	Bag Of Words
SVM	Support Vector Machine
CEDD	Color and Edge Directivity Descriptor
FCTH	Fuzzy Color Texture Histogram
JCD	Joint Composite Histogram
SIFT	Scale Invariant Feature Transform
CSD	Color Structure Descriptor
HTD	Homogeneous Texture Descriptor
ORB	Oriented FAST and Rotated BRIEF
ELM	Extreme Learning Machine
NN	Nearest Neighbor
RGB	Red Green Blue
HSV	Hue Saturation Value
CLTP	Completed Local Ternary Pattern
GLCM	Gray Level Co-occurrence Matrix
MPEG	Moving Picture Experts Group

Chapter 1
Introduction

1.1 Background

Autonomous robot navigation in outdoor environments is a challenging area of active research. The navigation task requires identifying safe, traversable paths that allow the robot to progress toward a goal while avoiding obstacles. Many applications for autonomous mobile robots need good perception mechanism for navigation. Exploration robots, such as the Mars Rovers [1], make use of such an approach, as human tele-operation incurs very long delays with transmitting information back and forth from Mars to Earth. Spacecrafts that orbit around a planet can make observations covering a large area; rovers are capable of microscopic observations and conducting physical experimentations. As a result, Rovers have become one of the most preferred methods for planetary exploration.

In search and rescue scenarios [2], a robot must be able to navigate an unknown environment to carry supplies or to rescue casualties from an open field. Detecting landmines is another example [3]. Handling disaster scenarios, such as the Fukushima nuclear meltdown or the Gulf of Mexico oil spill, is another important application for mobile robotics (both on land and underwater). These vehicles must be able to navigate and monitor these environments, which involves understanding the semantic details of what is around them.

In the paper 'Hough based terrain classification for real-time detection of drivable ground' [2], authors say that "Mobility in this regard is not only a mechatronic problem but also a perception, modeling and planning challenge.."

Operating in the outdoor environments common in these applications requires a vehicle to recognize traversable areas and non traversable areas in the scene that could cause damage to the vehicle. As a consequence, robust terrain identification becomes an important capability that mobile robots must possess and serves as the test bed for state-of-the-art computer vision and machine learning algorithms. This is particularly true if these systems are to be deployed in natural, hostile or unknown environments. In such cases, the safety of the robot might depend on the nature of the immediate terrain. The locomotion strategy of the robot is also intimately tied to the terrain type. Thus, its identification in a robust and efficient manner is of prime importance.

Technological and algorithmic improvements made over the past two decades have tremendously increased the level of intelligence that robotic systems can now display. This made it possible to deploy mobile robotic systems in increasingly challenging outdoor environments [1][2][3][4][5]. A crucial capability of such autonomous systems is learning without human supervision. The reason being: outdoor environments are highly unpredictable and, consequently, robots must be able to quickly adapt to changing situations. One possible way for such autonomous mobile robot systems to adapt is by employing a learning strategy.

1.2 Motivation

DARPA held competitions to promote technologies, in response to meet a congressional mandate that a third of U.S military ground vehicles be unmanned by 2015. DARPA, Grand Challenge winner, Stanley [6] used 5 sick laser range finders to measure cross sections of approaching terrain at different ranges out to 25m in front of the vehicle and a color camera for long road color perception. DARPA, Urban Challenge winner, boss [62] used 11 LIDAR's, 5 RADAR's and 2 high dynamic range cameras for terrain and environment perception. These winning entries successfully navigated the course autonomously.

Terrain perception using range sensors such as LIDAR, RADAR extracts geometric features which are inadequate to distinguish between terrain types. Driving behavior is related to physical properties of the surface. Geometrical description gives no information about these properties leading to the vehicle not altering its driving behavior to the surface.

The traditional vision-based terrain classification relies solely on analysing 3D distribution of point clouds given by a LIDAR, or stereo cameras. Meanwhile, scene interpretation based purely on geometric point of view is very difficult, even for human experience and knowledge. When two or more objects are near each other, they appear as one in the point cloud; many complex objects like vegetation might exist in different shape, so it is not possible to build common 3D models for them. Particularly using a LIDAR, it is prone to the scattering effect of beam scattering angle. Whereby, lacking information of far objects usually causes misclassification. Furthermore, the LIDAR has to sweep up and down to scan the environment, which is extremely time-consuming to acquire the whole frame of point cloud. This hinders many real-time applications.

The performance of stereo-based methods is limited, because stereo-based distance estimation is unreliable above 10 or 12 meter (for typical camera configurations and

resolutions). Recent learning-based research has focused on increasing the range of vision by classifying terrain in the far field according to the color of nearby ground and obstacles. This type of near-to-far color-based classification is quite limited, however. Although it gives a larger range of vision, the classifier has low accuracy and can easily be fooled by shadows, monochromatic terrain, and complex obstacles or ground types [8]. The range data doesn't differentiate a moving body from the terrain. But appearance information readily differentiates the navigable terrain and moving bodies (eg: car, humans, animals...).

Using vision to assist mobile robots and vehicles in navigation has been a popular research field. Recent terrain classification and navigation research has focused on using a combination of range sensors and visual data [1][9][10][11]. The work in [12] uses vibration data from onboard sensors and combines with the appearance of the terrain to predict the load bearing capability of the surfaces in far ranges. It is evident from the literature that vision based sensing and perception plays a crucial role in autonomous mobile system in various ways. This thesis investigates the efficiency of appearance based terrain sensing and perception.

In the paper, 'Vision-based motion planning for an autonomous motorcycle on ill-structured roads' [7] the authors say "The vision-based navigation algorithm that is based on passive sensing has its irreplaceable advantages such as far range perception and appearance information of the environment around it".

The main motivation of this work is to find out an excellent visual descriptor and an effective image classifier for different terrain types under varying illumination conditions for autonomous outdoor mobile robots. It is intuitive that color and texture together gives better semantic information for terrain classification. There are a number of color spaces and local neighborhood statistics and filter bank response methods for texture feature extraction. With multiple options at hand, this thesis answers empirically which color space works the best and worst, which texture works the best and worst and most importantly which combination of color and texture is the most suitable for terrain classification and which one is best suited for varying illumination conditions. It also attempts to answer which classification approach for terrain classification is best suited under varying illumination conditions.

1.3 Problem Description

Terrain classification Problem for Autonomous Mobile Robots - The terrain classification problem consists of categorizing a local terrain based on sensor information. At the lower

level of this description, a multitude of sensors measuring some of the terrain's physical properties are used. Relevant features are extracted from the pre-processed sensory information. Classification of the data into several categories of terrain is accomplished using a statistical classifier, such as mixture of Gaussians or k-Nearest Neighbours or a neural network based classifier. Consequently, a map is computed based on the classified terrain.

In the literature of autonomous mobile robots research, terrain classification is generally categorized as vision-based, range-based, reaction-based or a combination of these methods. Sensors, such as accelerometers or actuator feedback, can then be employed to indirectly estimate a number of physical properties of the surface (elasticity, sub-millimetre roughness) as the robot traverses the terrain. The increased richness of information thereby improves the terrain identification capabilities of robots equipped with such sensors. If the remote terrain identification capability is still required, a visual system can then be self-trained using this local information.

The core problem of terrain classification can be formalized as follows. Let C_{env} represent the set of all possible terrains the robot is expected to encounter. The possible values for the class prediction c are limited to this set C_{env}:

$$c \in C_{env} = \{C_1, C_{-1}\}$$

$C_1, C_{-1} \subset C_{env}$, where, C_1 – Navigable terrain classes, C_{-1} – Not navigable terrain classes

For each terrain $c \in C_{env}$, we expect a particular distribution of sensor measurements z in the sensory space of the robot.

$$z = \emptyset(c)$$

The problem of terrain classification can be summed up as finding the inverse mapping relationship; that is to identify the terrain $c \in C_{env}$ given a particular set of measurements z:

$$c = \emptyset^{-1}(z)$$

1.4 Contribution in this thesis

- Comparison of appearance based features (color, texture and color – texture) for binary terrain classification.
- Pixel based and block based terrain perception and analysis using monocular framework.
- Performance analysis of features and classifiers based on recall and false positive rate.

1.5 Thesis Organization

The rest of the thesis is organized as follows. Related literature survey is reported in chapter 2. In chapter 3, appearance based features to be investigated are described and extraction method is detailed. The features extracted are used for classification using Gaussian Mixture Modeling (GMM) – classifier and Multi Layer Perceptron (MLP) – classifier. Their description and algorithm is presented in chapter 4. In Chapter 5 the datasets used for validation are described. The experimental setup for GMM, MLP are given in chapter 6 followed by the experimental results an analysis in chapter 7. In chapter 8, visual mapping of perception to physical space is briefed that helps in construction a metric map used by the navigation module of the autonomous mobile robot. In chapter 9, conclusions of this thesis are reported.

Chapter 2
Literature Review

2.1 Geometric features based methods

Research in terrain classification reported in literature use a variety of sensors like cameras in both monocular and stereo vision framework, LIDAR, RADAR and tactile sensors. Raw data from different sensors is processed and combined to extract a set of features for terrain classification. Geometric features like elevation, roughness, shape, size, etc have been extracted to form 3D terrain models generated using LIDAR for terrain classification as reported in [13] [14]. Authors in [15], use the notions of terrain certainty and goodness that are combined to derive terrain traversability, [16] uses a correlated approach where certainty in terrain perception is modeled as a function of the number of points and the uniformity of their distribution.

In [14], 3D LIDAR data were classified based on scatterness, surfaceness and linearness for natural terrain analysis. Through the use of ground truth data, Gaussian mixture models of these classes were learnt by empowering Expectation Maximization on a set of static features that were computed from principal component analysis decomposition of sets of neighboring points across the 3D scene. Traversability analysis incorporating vehicle model was done by superposing the robot model across different directions of the elevation map and estimating its roll, pitch and ground clearance. Perception of negative obstacles such as gaps, downward inclined planes or descending steps which are characteristic examples of shapes that belong to this category has been at the focus of scene understanding approaches. This problem is addressed in [17], wherein terrain traversability was determined by the presence of positive and negative obstacles, step edge obstacles, slope steepness and terrain roughness. Detailed traversability analysis models that take vehicle dependent models as input are discussed in [13] [18].

2.2 Hybrid methods

The methodologies that use tactile sensor based learning for terrain classification, asses the traversability characteristics before actually driving over the respective region. In [19][12][20] the author's classify terrain by learning appearance information which combines vehicle wheel terrain interaction on vibration based sensing to identify these terrain classes.

By building upon fine terrain descriptors extracted from the GESTLAT system [1], step, roughness, pitch and border hazards were perceived as described in [21], by using the statistics within goodness map that quantified traversability by fitting planar patches locally across the map.

In [22], algorithms for obstacle detection, color-based terrain typing and obstacle labeling, and ladar-based obstacle detection in grass were proposed. A maximum likelihood strategy on each pixel, using mixture of Gaussians for terrain classification was used. Ladar data was used to discriminate between grass and other surfaces that are likely to belong to obstacles. The exemplar-based approach used in this paper, trains the system with a spectrum of images under many illumination conditions as possible. While this approach was very reliable in their experiments, it required substantial training data collection and labelling effort. In [23], efficient illumination compensation methods have been proposed.

In [24], superpixels based image classification, are demonstrated to be more accurate than the patch based image classification. Only stereo vision sensor is used and features from color (RGB and HSV) and texture feature (LM filters) spaces are extracted. The training data is hand labeled and its feature vector (length 55) is used to train the traversability classification algorithm which supports incremental learning and real-time classification. The learning algorithm incrementally updates prototypes in a codebook method and uses hyper-spheres to represent the prototypes. During the navigation, the UGV classifies a novel image region feature using a probabilistic approach which uses a modified *k*-nearest neighborhood algorithm.

In [12], author introduces a new method for long-term learning in the robot navigation task by leveraging past experience stored in the form of highly specialized, discrete models. Previously learned models, trained on appearance-based features from near field Stereo labels, are carefully selected from memory and applied to the current scene; the models' outputs are combined to create a final classification.

In [25], an online self supervised learning algorithm is trained and used to accurately segment an image in to obstacle and ground patches based on supervised input. In this method, the implicit assumption of independence in local pixel neighborhoods is relaxed by modeling correlations between pixels in the submodular Markov Random Field (MRF) framework. Exact inference is implemented via an efficient max flow computation; and learning, via an averaged-subgradient method. For approximate inference a max-margin is

optimized using a Maximum A Posteriori estimate. Only RGB color feature, form stereo data were used and the algorithms performed better for Markov Random Fields than histograms.

In [9], [10] [11], robustness of road detection is achieved by combining sensor information from a range sensor and a color camera. Using the first two modalities, the system first identifies a nearby patch of drivable surface. Computer Vision then takes this patch and uses it to construct appearance models to find drivable surface outward into the far range using Mahalanobis metric.

In [8], a deep hierarchical network (a multi-layer convolutional network, initialized with deep belief net training) is trained to extract informative and meaningful features from an input image, and the features are used to train a real-time classifier to predict traversability. A feature representation that is robust to irrelevant transformations of the input, such as lighting and viewpoint and at the same time, informative enough to recognize high-level terrain structures such as paths and obstacles is proposed. The online learning framework takes the feature vectors and supervisory labels and trains 5 binary classifiers. The classifier is trained on every frame for fast adaptability. A stochastic gradient descent and a cross entropy loss function are used for classification.

2.3 Appearance features based methods

In [26], comparison of multiple approaches for visual terrain classification using local features is given, using random trees for classification and cross validation for verification of results. Along with texture based descriptors: Local Binary patterns descriptors, Local ternary patterns (LTP) descriptor and Local Adaptive Ternary pattern descriptor key point descriptors: SURF (Speeded Up Robust Features) descriptor and Daisy descriptor and Contrast Context Histogram (CCH) descriptor were also used to classify terrain as gravel, asphalt, grass, big-tiles and small-tiles. Texture descriptors performed better but SURF descriptors were better at discriminating at higher resolutions. CCH performed the worst. It is demonstrated that visual terrain classification can be successfully performed even in extreme conditions, such as motion blur induced by a fast moving robot and its vibrating camera, different weather conditions, both wet and dry ground surfaces and a low camera viewpoint.

In [27], authors address terrain traversability as a classification problem where statistical information extracted from small image patches is combined with soft computing inference techniques. A comparative study of the results obtained with three different classifiers based on (1) a heuristic rule-based terrain classifier (RBTC), (2) artificial neural

networks (ANN), and (3) fuzzy logic reasoning (FL) is presented. Terrain patches were classified into one of the following types: sandy, rough, very rocky, rocky, impassable or uncertain. The results reported in this work suggest a clear relation between the classifier complexity and its performance, with over 98% success rate for the FL classifier, followed by the ANN approach with 93% and the RBTC performing in some cases under 20%.

In [7], a vision-based motion planning system for a motorcycle was designed for autonomous navigation in desert terrains, where uniform road surface and lane markings were not present. The color information and the directional information from prior vehicle tire tracks and pedestrian footsteps in an ill-structured road were used to construct a vision vector space (V2-Space), a unitary vector set that represented local collision-free directions in the image coordinate system. Although the algorithm proposed had 91% successful classification rate it is unfit to be used on planetary exploration environments.

In [28], a neural network classifier is trained using real off-road terrain images. Color and texture features extracted using discrete wavelet transform coefficients are used. In addition, spatial coordinates, where a terrain class is located in the image, are also adopted Experiments showed that using the wavelet features and spatial co-ordinates features improved the terrain cover classification performance. An average classification rate of 81.8% was achieved using wavelet and spatial features.

[29], presents an approach that works with a single, compact camera and maintains high classification rates that are robust to changes in illumination. Terrain is classified using a bag of visual words (BOVW) created from SURF with a support vector machine (SVM) classifier. An adaptive sliding window technique within a 2D image was proposed in order to obtain terrain signatures of constant feature density using a simple gradient descent based approach and iteratively computed the corresponding visual word histogram.

In [30], multiple approaches to visual terrain classification for outdoor mobile robots are compared. Composite descriptors called CEDD, FCTH and JCD, with traditional color and texture descriptors, such as LTP, SCD, EHD and a descriptor called CSD‑HTD generated by late fusion method. Three BOVW models based on SIFT, SURF and ORB, respectively. For performance evaluation three classifiers ELM, SVM and NN are employed and tested on patches of terrain collected in outdoor environment.

Chapter 3
Appearance based feature extraction and representation

3.1 Introduction

Electromagnetic waves have an infinite range of frequencies, but the human eye can only perceive the range of frequencies in the visible spectrum which ranges from about 400 to 700 nm. Each frequency defines a different color as illustrated in Figure 3.1. In the visible spectrum, each wavelength is perceived as a color; the extreme values are perceived as violet and red and between them there are greens and yellows.

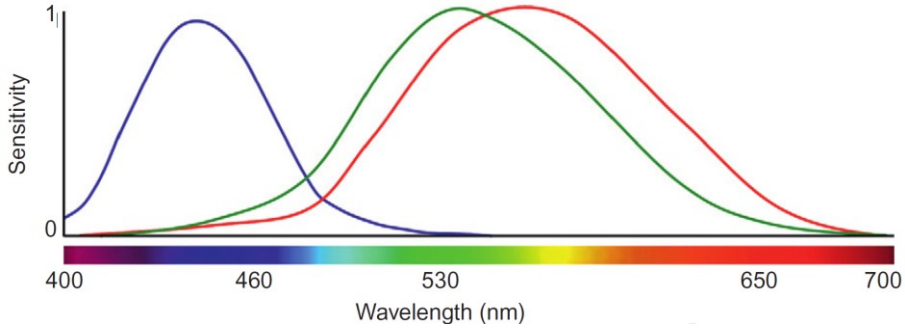


Figure 3.1: Visible spectrum and tristimulus response curves.

Human vision representation of color is created by three types of cell receptors in our eyes that are sensitive to a range of frequencies near the blue, red, and green lights. Thus, instead of describing colors by frequency content or radiometric properties, colors can be represented by three stimuli according to the way we perceive them. This way of organizing colors is known as trichromatic or tristimulus representation. For structured and unstructured (natural outdoor) environments, color representation gives rich scene information. This is advantageous for machine vision algorithms like robot perception for terrain recognition and classification.

In the thesis, 'Learning to visually predict terrain properties for planetary robots' [12], the author says, "The purpose of exteroceptive terrain classification is to classify terrain patches based on features derived from exteroceptive sensor data – in this case color, visual texture, and topography."

In the literature there are two types of features extracted from exteroceptive sensors:

1. Geometric features form range sensors (stereo camera, LIDAR, RADAR)
2. Appearance based features from monocular camera framework.

Some obstacles, however flat, are not traversable (e.g. lawn). Other places like pedestrian crossing should be recognized to apply special strategy for traversing [31]. Appearance based features greatly benefit the classification task by using visual clues to minimize the number of false positive objects [32], non geometric hazards [12] and enhancing the perception range of the autonomous mobile system.

For classification purposes, the visual content of the images, *f* is mapped into feature space, *z*. Five color spaces (RGB, rgb-chromaticity, HSV, opponent and Lab) color spaces, along with, five texture descriptors (LTP, CLTP, GLCM, Gabor filter response and MPEG-7 edge filter response) and combination of color and texture descriptors were investigated. The extracted visual descriptors are applied to and compared with two different classifiers, GMM and MLP.

Due to changing illumination conditions in the environment and non-uniform illumination (shadows, reflection), classification using raw sensor data results in poor classification [31] [12] [11]. Therefore, it is generally useful to map the colors from the RGB space to a more suitable one. 5 color spaces with their descriptions are as follows.

3.2 Color Features

3.2.1 RGB

Data from the camera is available as red, green, and blue (RGB) intensities. The RGB feature vector employed is as given in equation (3.1).

$$\text{Feature vector: } z_{RGB} = \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.1}$$

3.2.2 rgb-chromaticity

An intensity normalized space of the RGB-space, is define by equation (3.2)

$$r = \frac{R}{R+G+B}, \qquad g = \frac{G}{R+G+B}, \qquad b = \frac{B}{R+G+B} \tag{3.2}$$

This space is invariant to viewing direction, surface orientation, illumination intensity and illumination direction [33]. It however lacks important information about intensities close to zero. The feature vector is formed as given in equation (3.3).

$$\text{Feature vector: } z_{rgb} = \begin{pmatrix} r \\ g \\ b \end{pmatrix} \tag{3.3}$$

3.2.3 Opponent color space

The opponent color space is investigated as suggested in [32]. The opponent colorspace [34], given by equation (3.4) is a linear transformation of RGB that matches the physiology of the human visual system.

$$\begin{pmatrix} wb \\ rg \\ yb \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ -1 & -1 & 2 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.4}$$

Moreover it is shown that color opponent coordinate systems can represent natural images very efficiently [35]. The three axes represent the intensity values and two chromatic values. The chromatic values give the red-green and yellow-blue differences. Opponent color spaces tend to decorrelate the RGB components, which is a desirable characteristic for pattern recognition [36]. The feature vector is constructed as shown in equation (3.5).

$$\text{Feature vector: } z_{opponent} = \begin{pmatrix} wb \\ rg \\ yb \end{pmatrix} \tag{3.5}$$

3.2.4 Modified Hue Saturation Value

A modified hue, saturation, and value (eHSV) representation of color is used as in [37]. In this approach, hue (an angle) is represented as two values – sin(hue) and cos(hue) – to eliminate the artificial discontinuity at 2*pi. The transformation is given in equation (3.6), followed by the feature vector in equation (3.7).

$$\begin{pmatrix} V = max(R,G,B) \\ S = \begin{cases} \frac{V - min(R,G,B)}{V} & if\ V \neq 0 \\ 0 & otherwise \end{cases} \\ H = \begin{cases} 60\frac{G-B}{S} & if\ V = R \\ 120 + 60\frac{B-R}{S} & if\ V = G \\ 240 + 60\frac{R-G}{S} & if\ V = B \end{cases} \\ If\ H < 0\ then\ H = H + 360 \\ H = \frac{H}{2}\ \ with\ 0 \leq H \leq 255 \end{pmatrix} \tag{3.6}$$

$$\text{Feature vector: } z_{eHSV} = \begin{pmatrix} \cos(H) \\ \sin(H) \\ S \\ V \end{pmatrix} \tag{3.7}$$

3.2.5 Lab colorspace

The Lab color features are extracted as in [38]. Here the 8-bit images must first be converted into floating-point images with range [0..1]. One common disadvantage of all above mentioned colors spaces is that colors cannot be compared to each other; that is, all above mentioned spaces are not perceptually uniform. In 1976 the CIE defined the LAB-space to overcome this disadvantage. The L*, a* and b* values are calculated in two stages. The first stage calculates the XYZ values as given in equations (3.8) – (3.9). The second stage transforms the XYZ values non-linearly to LAB values as in equation (LAB).

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \frac{1}{255} * \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.8}$$

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{pmatrix} \begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} \tag{3.9}$$

XYZ after CIE L*a*b*,

$$\mathrm{X}' = \frac{\mathrm{X}}{0.950456}, \qquad \mathrm{Z}' = \frac{\mathrm{Z}}{1.088754} \tag{3.10}$$

$$\begin{pmatrix} L' = \begin{cases} 116 + Y^{1/3} \; if \; Y > 0.008856 \\ 903.3 * Y \quad if \; Y \leq 0.008856 \end{cases} \\ a' = 500 * \left(f(X') - f(Y) \right) + 128 \\ b' = 200 * \left(f(Y) - f(Z') \right) + 128 \end{pmatrix} \tag{3.11}$$

Since the result has ranges –

$$\begin{pmatrix} 0 \leq L' \leq 100 \\ -127 \leq a' \leq 127 \\ -127 \leq b' \leq 127 \end{pmatrix} \tag{3.12}$$

Ranges are adjusted accordingly,

$$\begin{pmatrix} L = L' * 255/100 \\ a = a' + 128 \\ b = b' + 128 \end{pmatrix} \tag{3.13}$$

$$\text{Feature vector: } z_{Lab} = \begin{pmatrix} L \\ a \\ b \end{pmatrix} \tag{3.14}$$

3.3 Texture Features

While color is a point property, texture is a local neighbourhood property. In order for a texture descriptor to be useful, it must provide an adequate description of the underlying texture parameters and it must be computed in a neighbourhood which is appropriate to the local structure being described. Visual texture is a measure of the local spatial variation in intensity of an image. Researchers have proposed many metrics for visual texture, such as Gabor filter, Mpeg-7 edge filters, local energy methods LTP, CLTP, GLCM

- The texture features discussed are appropriately adapted for pixel wise image description.
- For illustration purposes image 1 in LAGR DS 3A is used as shown in Figure 3.2.

Figure 3.2: Sample image used from dataset LAGR DS 3A for texture illustration.

3.3.1 Local Ternary Pattern

In [39], authors extended Local Binary Pattern (LBP) to 3-valued codes (−1, 0, 1), LTP as illustrated in Figure 3.3. The gray-levels in a zone of width ±t around i_c are quantized to zero, ones above this are quantized to +1 and ones below it to −1. The new operator is more robust to noise, particularly in near-uniform image regions, and to smooth weak illumination gradients. The mathematical expression of the LTP is given as in equation (3.15).

$$\mathrm{LTP_{r,c}} = \sum_{p=0}^{p-1} 2^{p} s(i_p - i_{r,c}), \text{where } s(x) = \begin{cases} 1 & x \geq t \\ 0 & -t < x < t \\ -1 & x < -t \end{cases} \quad (3.15)$$

Where, $i_{r,c}$ and i_p $(p = 0, \ldots, P-1)$ denote the gray value of the centre pixel and gray value of the neighbour pixels respectively, and P is the number of neighbours.

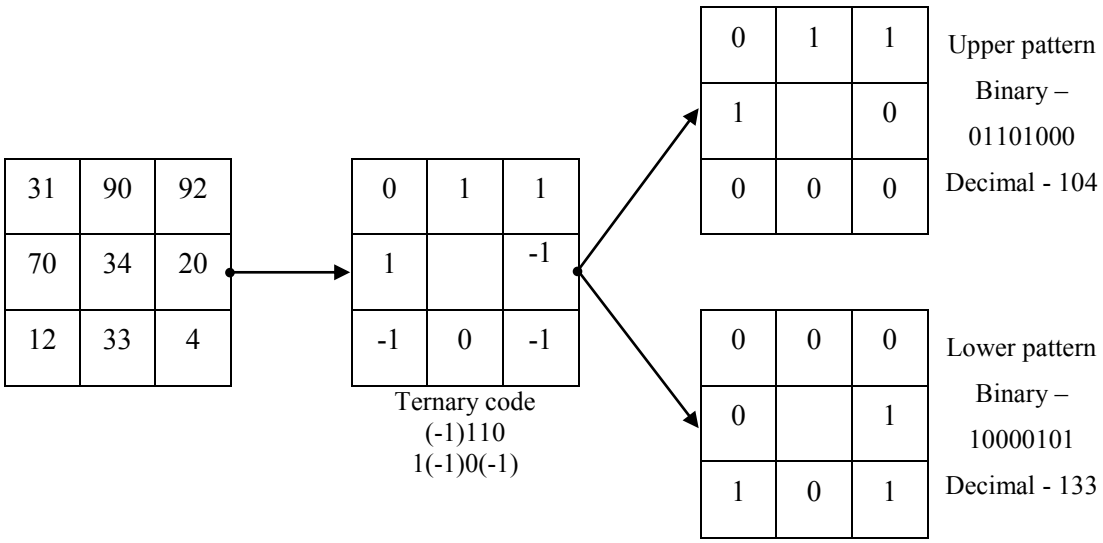
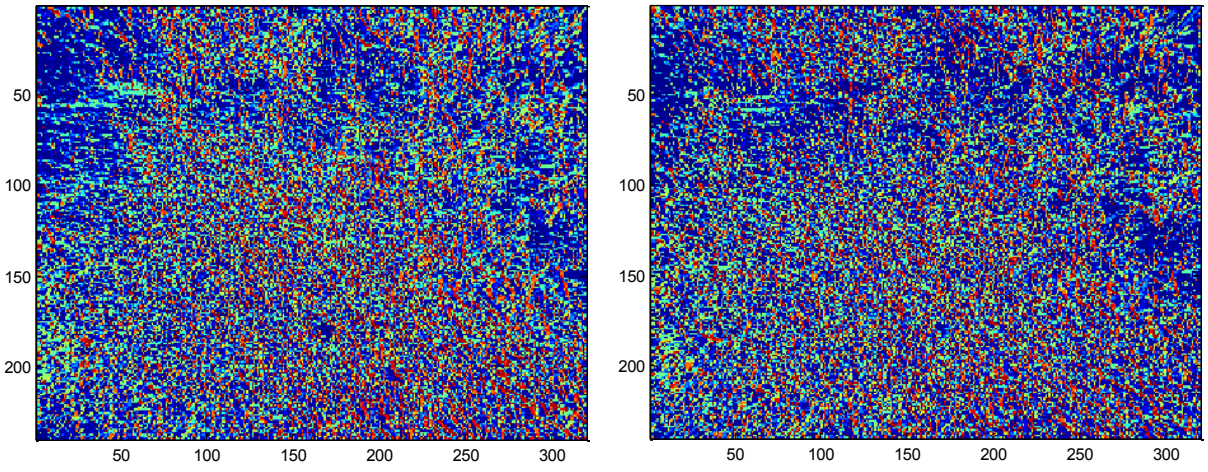


Figure 3.3: Illustration of LTP features extraction

The feature vector of LTP is represented as in equation (3.16) and the extracted feature of the image is as shown in Figure 3.4.

$$\text{Feature vector, } z_{LTP} = \begin{pmatrix} LTP_{upper} \\ LTP_{lower} \end{pmatrix} \tag{3.16}$$



(a) LTP_{upper} (b) LTP_{lower}

Figure 3.4: Feature images - LTP

3.3.2 Completed Local Ternary Pattern

Introduced by [40] the feature is similar to CLBP [41], the LTP is extended to completed modelling LTP (CLTP). As mentioned before, the LTP is more robust to noise

than LBP. Furthermore, constructing the associated completed Local Ternary Pattern will help to enhance and increase it's discriminating property. In CLTP, local difference of the image is decomposed into two sign complementary components and two magnitude complementary components as illustrated in Figure 3.5.

31	90	102
70	34	20
12	33	20

0	1	1
1		0
0	0	0

Sign Component
(a)

3	90	102
36		14
22	1	25

0	1	1
1		0
0	0	0

Magnitude Component
(Assume c = 29)
(b)

Figure 3.5: CLTP illustration, (a) sign component (LBPS code); (b) magnitude components (LBPM code) assuming, threshold = 29.

The feature vector of CLTP is constructed using equations (3.17) – (3.20) and the feature vector is represented as in equation (3.21). The pixel-wise feature image is illustrated in Figure 3.6.

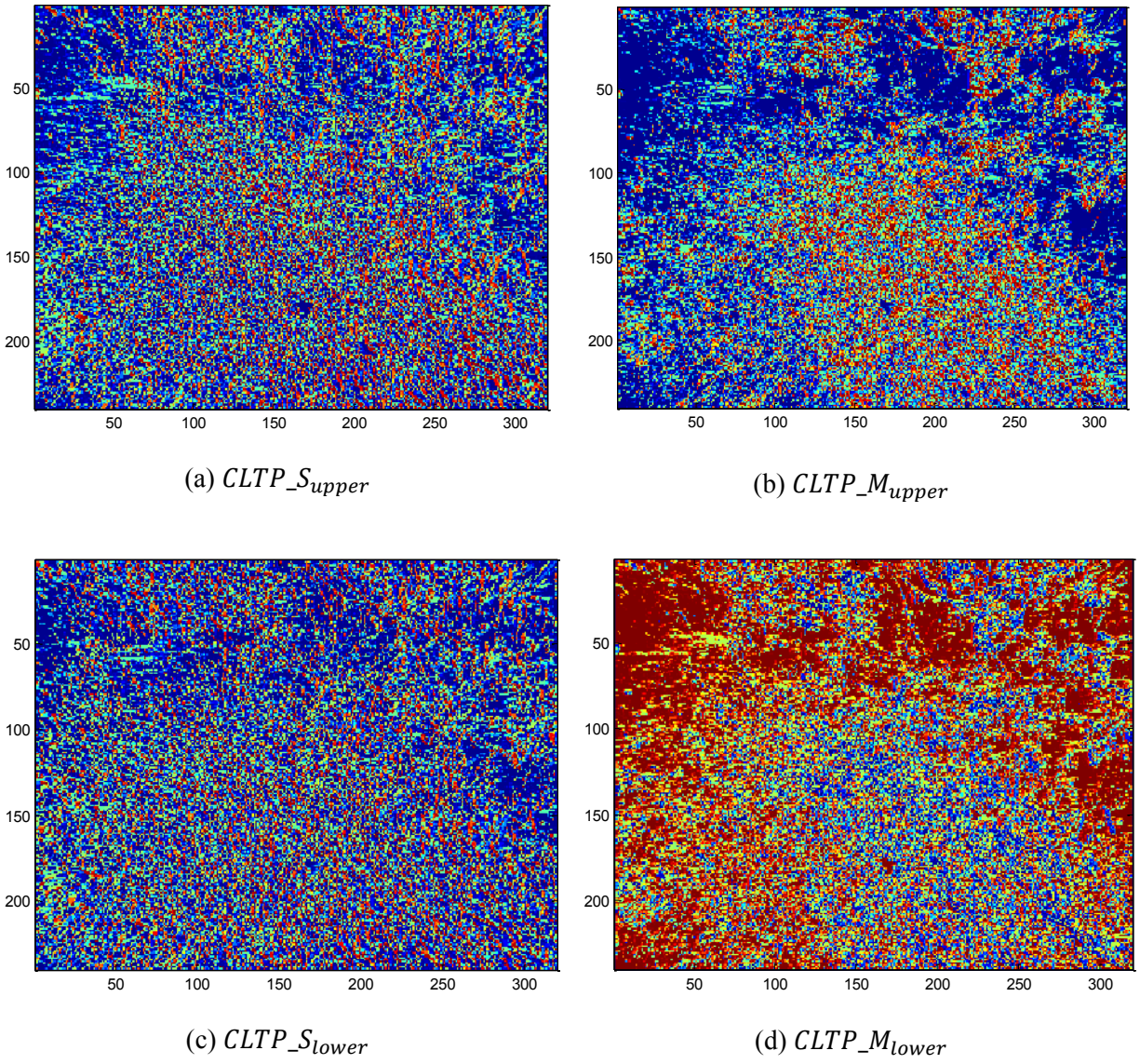
$$CLTP\_S_{r,c}^{upper} = \sum_{p=0}^{p-1} 2^p s_p^{upper}(i_p - (i_{r,c} + t)), where\ s_p^{upper} = \begin{cases} 1, & i_p > i_{r,c} + t \\ 0, & otherwise \end{cases} \quad (3.17)$$

$$CLTP\_S_{r,c}^{lower} = \sum_{p=0}^{p-1} 2^p s_p^{upper}\left(i_p - \left(i_{r,c} - t\right)\right), where\ s_p^{upper} = \begin{cases} 1, & i_p < i_{r,c} - t \\ 0, & otherwise \end{cases} \quad (3.18)$$

$$CLTP_{M_{r,c}}^{upper} = \sum_{p=0}^{p-1} 2^p t_p^{upper}\left(m_p^{upper}, c\right),\ where\ t_p^{upper} = \begin{cases} 1, |\ i_p - (i_{r,c} + t) \geq c| \\ 0, |\ i_p - \left(i_{r,c} + t\right) < c| \end{cases} \quad (3.19)$$

$$CLTP\_M_{r,c}^{lower} = \sum_{p=0}^{p-1} 2^p t_p^{lower}\left(m_p^{lower}, c\right),\ where\ t_p^{lower} = \begin{cases} 1, |\ i_p - (i_{r,c} - t) \geq c| \\ 0, |\ i_p - \left(i_{r,c} - t\right) < c| \end{cases} \quad (3.20)$$

$$\text{Feature vector, } f_{CLTP} = \begin{pmatrix} CLTP\_S_{upper} \\ CLTP\_S_{lower} \\ CLTP\_M_{upper} \\ CLTP\_M_{lower} \end{pmatrix} \quad (3.21)$$



(a) $CLTP\_S_{upper}$ (b) $CLTP\_M_{upper}$

(c) $CLTP\_S_{lower}$ (d) $CLTP\_M_{lower}$

Figure 3.6: Feature images - CLTP

3.3.3 Gray Level Co-occurrence Matrix

The GLCM measures the relative co-occurrence of image values by recording how often different combinations of adjacently occur within the image given a localized orientation. Traditionally, this localized orientation is defined with reference compass directions as *{N,S, E,W,NW,NE,SW, SE}*. The co-occurrence matrix is formed using a set of offsets sweeping through 180 degrees (i.e. 0, 45, 90, and 135 degrees) at the same distance to achieve a degree of rotational invariance. The co-occurrence features as extracted using equations (3.22) – (3.25). Mean of the surfaces formed of co-occurrence matrix, in itself, is able to distinguish between different terrain samples (grass, soil, asphalt) as shown in Figure 3.7.

$$GLCM_{r,c}^{N} = \sum_{i=-1}^{1}\sum_{j=-1}^{1} i_{i+r,j+c} * g_N(i+2,j+2), where\ g_N = \begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 1 & 0\end{pmatrix} \quad (3.22)$$

$$GLCM_{r,c}^{E} = \sum_{i=-1}^{1}\sum_{j=-1}^{1} i_{i+r,j+c} * g_E(i+2,j+2), where\ g_E = \begin{pmatrix}0 & 0 & 0\\ 1 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} \quad (3.23)$$

$$GLCM_{r,c}^{SE} = \sum_{i=-1}^{1}\sum_{j=-1}^{1} i_{i+r,j+c} * g_{SE}(i+2,j+2), where\ g_{SE} = \begin{pmatrix}1 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} \quad (3.24)$$

$$GLCM_{r,c}^{SW} = \sum_{i=-1}^{1}\sum_{j=-1}^{1} i_{i+r,j+c} * g_{SW}(i+2,j+2), where\ g_{SW} = \begin{pmatrix}0 & 0 & 1\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} \quad (3.25)$$

The GLCM itself is not a texture feature descriptor in its own right but is more correctly a 2-D statistical record of co-occurring value variation. In the original work of [42], texture features were in turn calculated from the normalized co-occurrence matrix as a set of 14 summary statistics. There are many ways in which GLCM with Haralick features are extracted (window based, block based). In this work, features are adopted as was suggested in [43], where, the authors down-selected a subset of five summary statistics based on the realization of the inter correlation within the original set of 14.

This subset of five measures (entropy, energy, contrast, correlation, local homogeneity) is used in this thesis as a workable representative texture feature descriptor derived from the original GLCM formulation. Each is described as in equations (3.26) – (3.30). The feature images are shown in Figure3.8. The feature vector of GLCM is given as in equation (3.31).

$$Entropy = -\sum_{i=1}^{cols}\sum_{j=1}^{rows} M(i,j) log_2(M(i,j)) \quad (3.26)$$

$$Energy = \sum_{i=1}^{cols}\sum_{j=1}^{rows} M(i,j)^2 \quad (3.27)$$

$$Contrast = \sum_{i=1}^{cols}\sum_{j=1}^{rows}(i-j)^2 M(i,j) \quad (3.28)$$

$$Correlation = \frac{1}{\sigma_I \sigma_J} - \sum_{i=1}^{cols}\sum_{j=1}^{rows}(i-\mu_I)\left(j-\mu_J\right)M(i,j) \quad (3.29)$$

$$Localhomogeneity = \sum_{i=1}^{cols}\sum_{j=1}^{rows}\frac{M(i,j)}{(1+(i-j)^2)} \quad (3.30)$$

The feature vector for GLCM is represented as in equation (3.31).

$$\text{Feature vector, } z_{GLCM} = \begin{pmatrix} gray \\ \overline{GLCM_{N,E,SE,SW}} \\ entropy \\ energy \\ contrast \\ correlation \\ Local\ homogeneity \end{pmatrix} \tag{3.31}$$

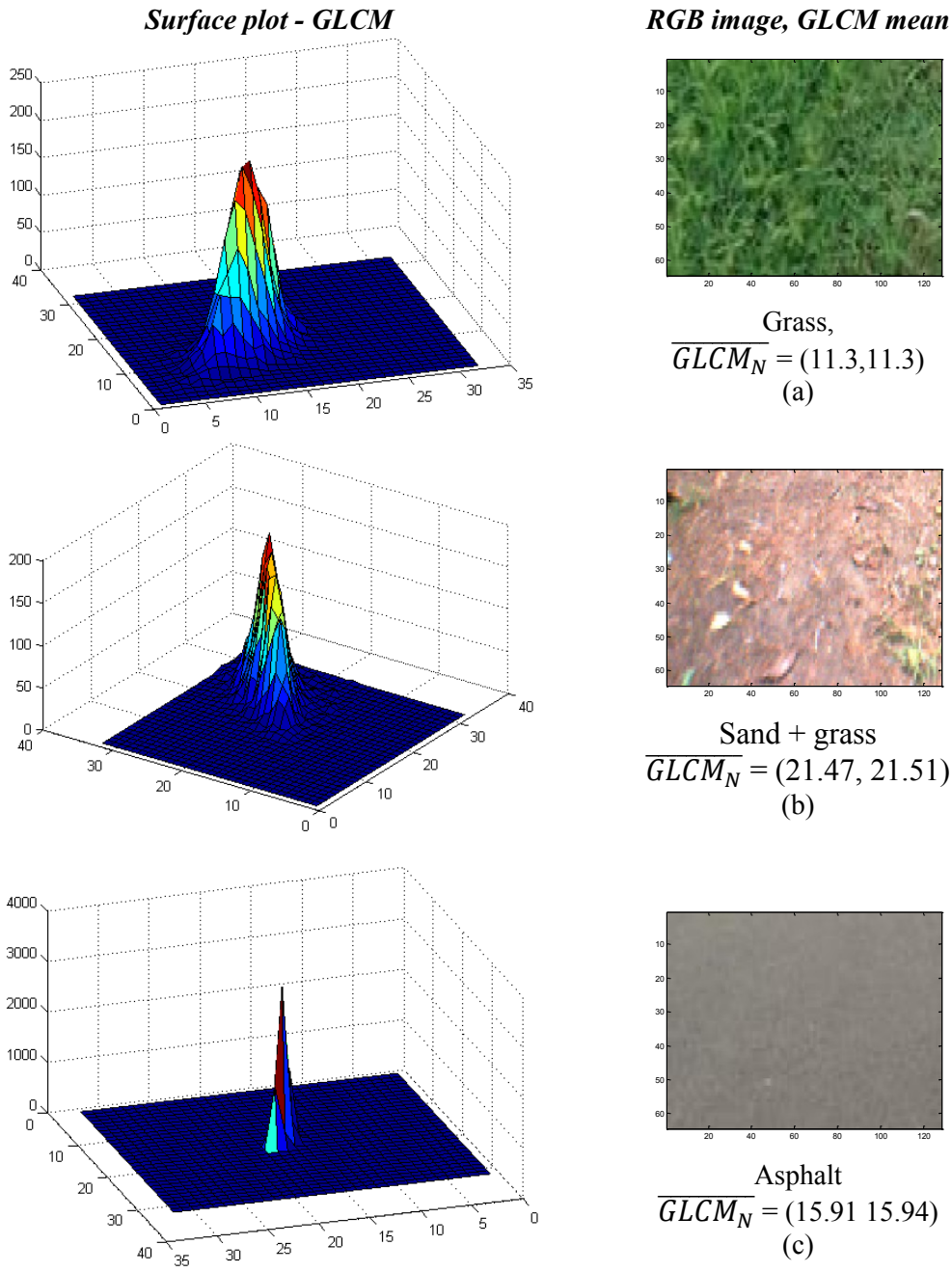
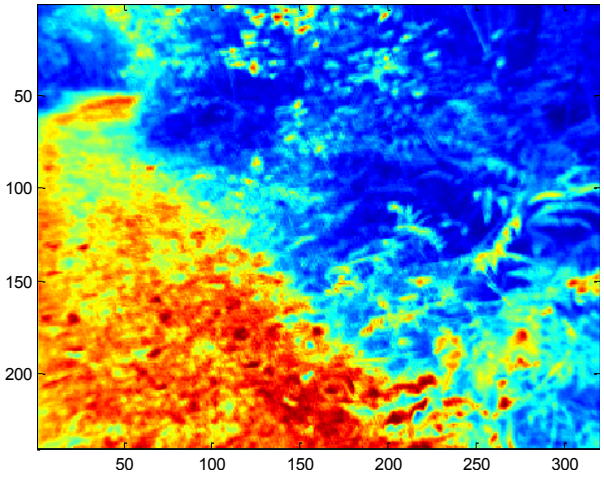
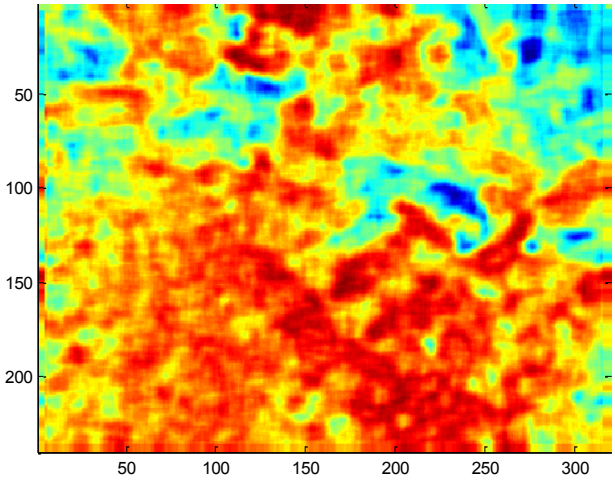


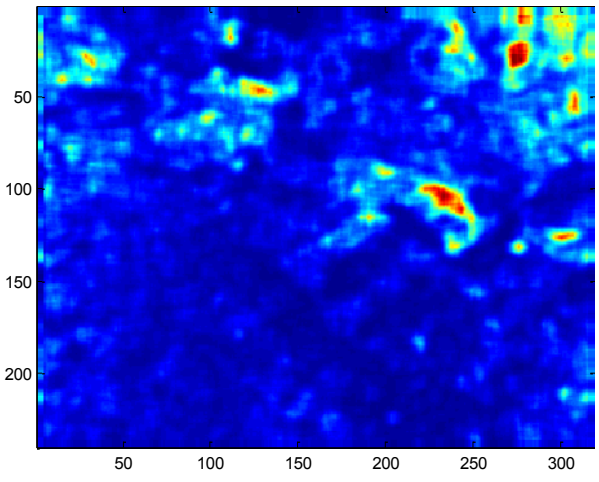
Figure 3.7: Surface plots - GLCM



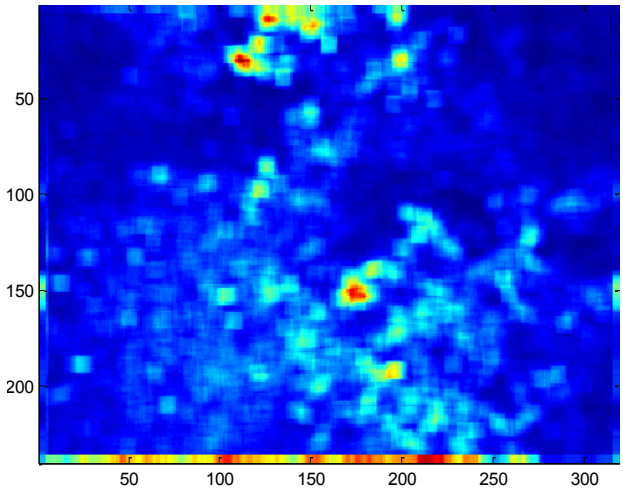
(a) $\overline{GLCM_{N,E,SE,SW}}$



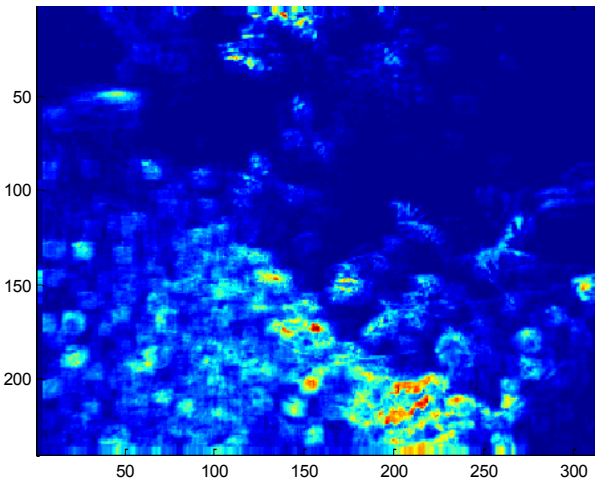
(b) Entropy



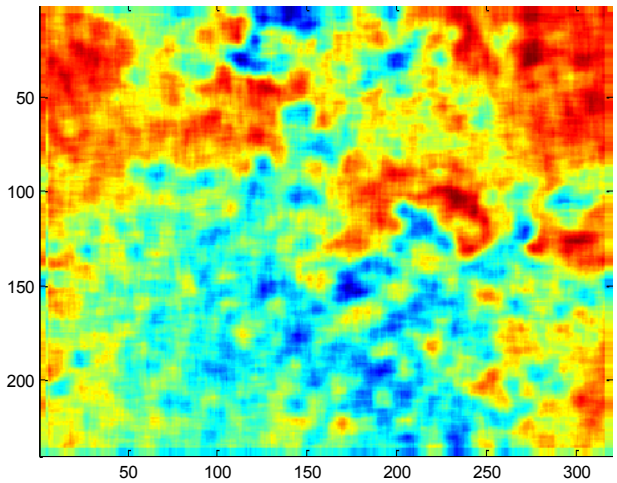
(c) Energy



(d) Contrast



(e) Correlation



(f) Local Homogeneity

Figure 3.8: Feature images - GLCM

3.3.4 Gabor filter response

The 1-D Gabor function was first defined by Gabor [44], and later extended to 2-D by Daugman [45]. A 2-D Gabor filter is an oriented complex sinusoidal grating modulated by a 2-D Gaussian function, as given in equation (3.32).

$$G_{\sigma,\emptyset,\theta}(x,y) = g_\sigma(x,y)e^{2\pi j\emptyset(xcos\theta+ysin\theta)}, \text{ where } g_\sigma(x,y) = \frac{1}{2\pi\sigma^2}e^{\frac{x^2+y^2}{2\sigma^2}} \quad (3.32)$$

The multi channel filtering technique in this paper uses a bank of even symmetric Gabor filters to functionally characterize the different channels. The filtered images are then subjected to a nonlinear transformation. An energy measure is defined on the transformed images in order to compute different texture features. The measure calculates a sum of texture energies over a small window around each transformed pixel in the nonlinearly transformed filtered images. Using the energy measure as features, a vector of Gabor features is defined for each pixel in the original image. The Gabor filter $G_{\sigma,\emptyset,\theta}(x,y)$forms complex valued function. Decomposing $G_{\sigma,\emptyset,\theta}(x,y)$ eq. (3.33) into real eq. (3.34) and imaginary eq. (3.35) parts gives

$$G_{\sigma,\emptyset,\theta}(x,y) = R_{\sigma,\emptyset,\theta}(x,y) + jI_{\sigma,\emptyset,\theta}(x,y) \quad (3.33)$$

Where,
$$R_{\sigma,\emptyset,\theta}(x,y) = g_\sigma(x,y)\cos\big(2\pi\emptyset(x\cos\theta + y\sin\theta)\big) \quad (3.34)$$

$$I_{\sigma,\emptyset,\theta}(x,y) = g_\sigma(x,y)\sin\big(2\pi\emptyset(x\cos\theta + y\sin\theta)\big) \quad (3.35)$$

In this work, for terrain learning, only the real part of the Gabor filter response is used for computational simplicity as given in equation (3.36).

$$G_{\sigma,\emptyset,\theta}(x,y) = exp\left\{-\frac{1}{2}\left[\frac{X^2}{{\sigma_x}^2}+\frac{Y^2}{{\sigma_y}^2}\right]\right\}\cos(2\pi u_0 X + \emptyset) \quad (3.36)$$

Where $X = x\cos\theta + y\sin\theta$, $Y = -x\cos\theta + y\sin\theta$, u_0 and $\emptyset$ are the radial frequency and phase of the sinusoidal plane wave along the x-axis, and $\sigma_x$and σ_y are the space constants of the Gaussian envelope along the x- and y-axis, respectively.

Gabor filter is designed using the parameters: 4 orientations and 2 radial frequencies (scales), Radial frequency, u_0 = $\{2\sqrt{2}, 4\sqrt{2}, 8\sqrt{2}\}$, Orientation, θ = $\{0^0, 30^0, 90^0, 120^0\}$ and $\emptyset = 180^0$ and $\sigma = 0.5\frac{N_c}{u_0}$, block size - N_c = 32, $\sigma_x = \sigma$ and $\sigma_y = 0.3\sigma$,

The magnitude response is obtained using the algorithm as described below and the magnitude response for Gabor filter bank can be seen in Figure 3.9.

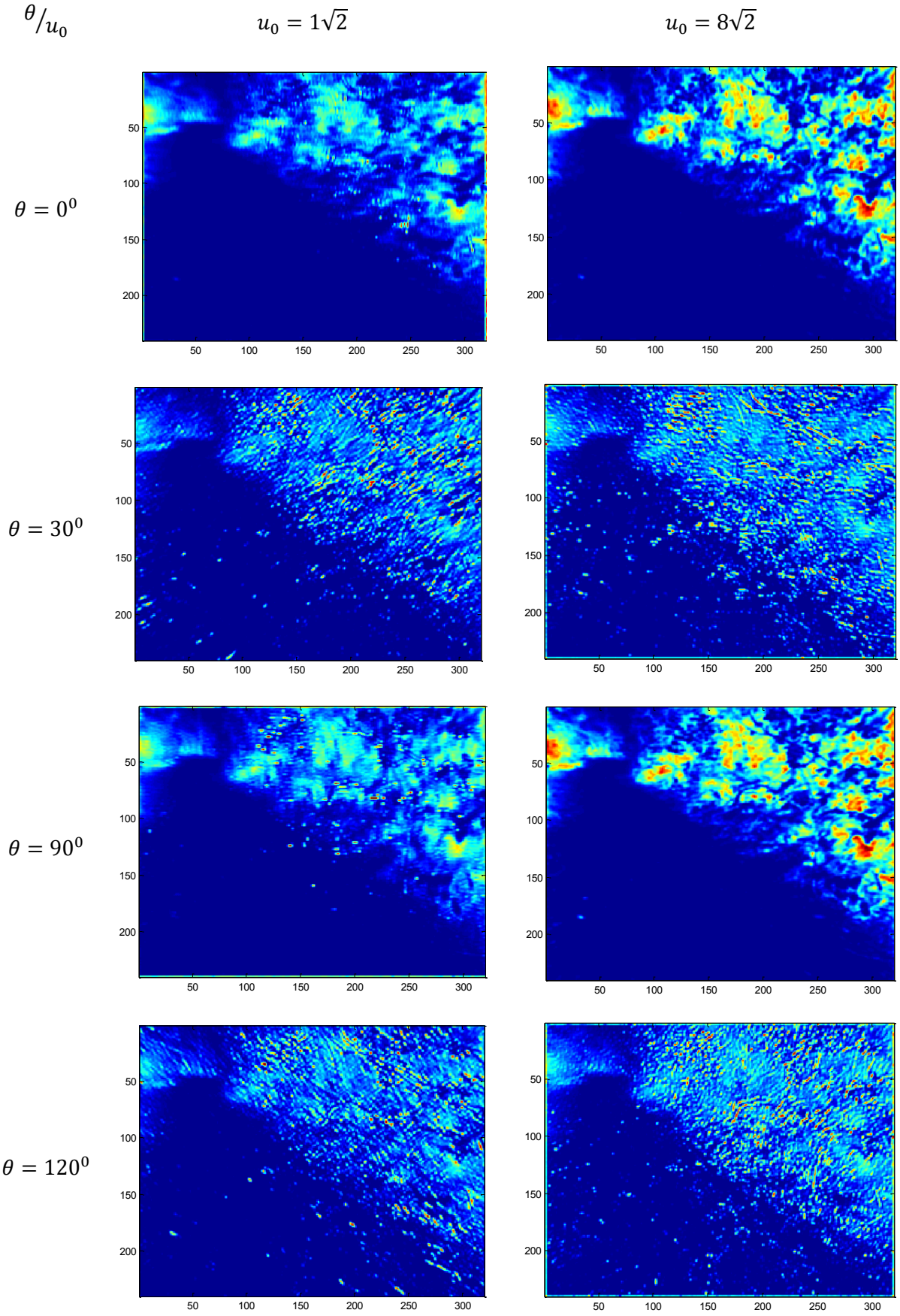


Figure 3.9: Feature images, Gabor filter magnitude response

Algorithm for obtaining magnitude response using filter banks

Start

---step 1: Gabor Prefiltering - Convolution of 8 filters with the image.

---step 2: Non-linear transformation of the filtered images given as in (3.37)

$$\varphi(m) = tanh(\alpha m) = 1 - \frac{1}{1+e^{-2\alpha m}}, \alpha = 0.25 \qquad (3.37)$$

---step 3: Gaussian post filtering of non-linear transformed images.

End

3.3.5 MPEG7 Edge Descriptor

Edges in images constitute an important feature to represent their content. Also, human eyes are sensitive to edge features for image perception [46]. It can describe both shape and texture features. The Edge Histogram Descriptor (EHD) describes five edge types in the image, namely horizontal, vertical, two diagonal and non-directional edge types.

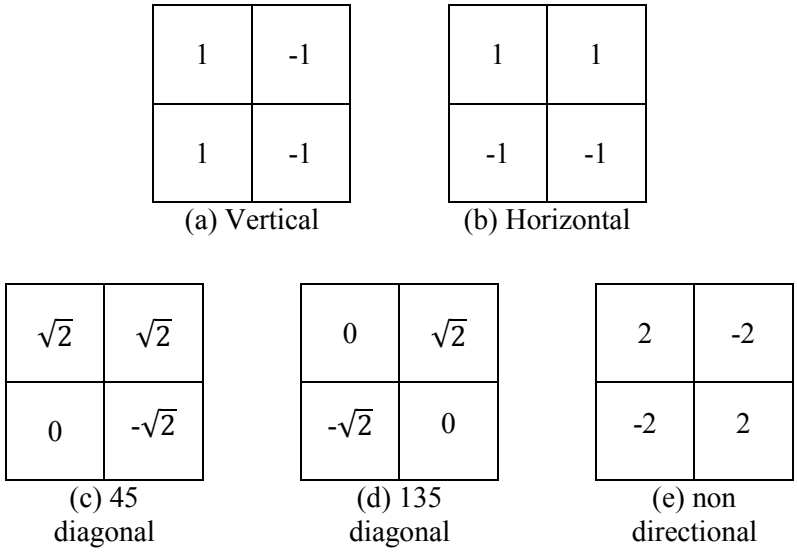
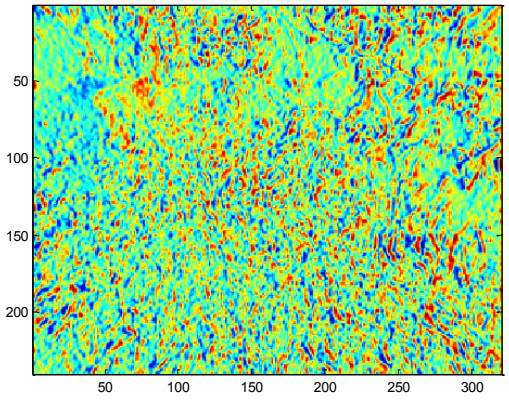


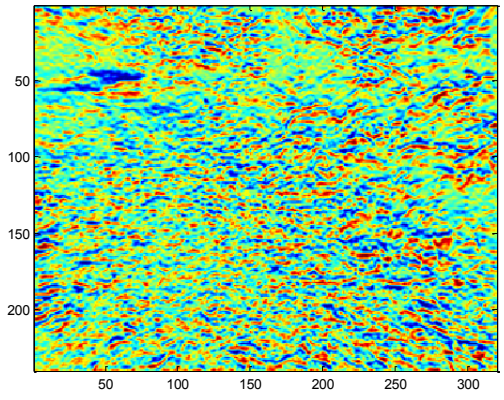
Figure 3.10: Mpeg7 Edge filters

The EHD described in [47] represents local edge distribution in the image. It describes edges in each 'sub-image', which is obtained by dividing the image using 4x4 grid. Edges in the sub-image are classified into five types; vertical, horizontal, 45-degree, 135-degree, and non-directional. Occurrence of each type becomes a histogram bin, producing 80 histogram bins overall. For online learning of different terrain types using GMM, only the edge filter bank described in the MPEG7 EHD descriptor will be used to extract texture features using the above described algorithm. The Filter bank used is given in Figure 3.10 to extract the texture features as discussed in the earlier algorithm. Illustration of the edge filters

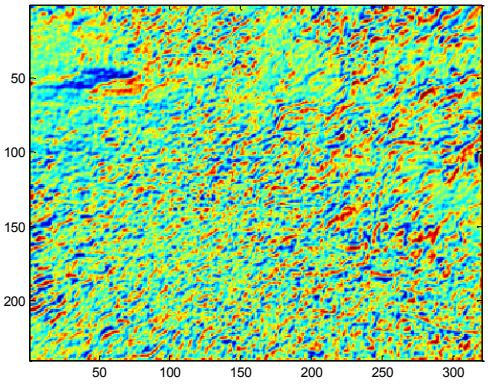
magnitude response is as shown in Figure 3.11. The feature is represented as shown in equation (3.38).



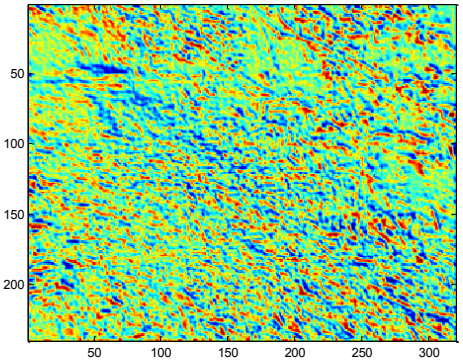
(a) Vertical edge



(a) Horizontal edge

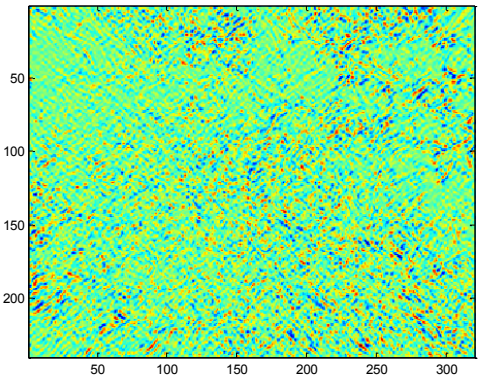


(c) Directional 45



(d) Directional 135

Feature vector, $z_{edge} = \begin{pmatrix} vertical\ edge \\ horizontal\ edge \\ directional\ 45 \\ directional\ 135 \\ non\ directional \end{pmatrix}$ (3.38)



(e) Non directional

Figure 3.11: Feature images - MPEG7 edge filter magnitude responses

Chapter 4

Classifiers description and computational flow

4.1 Introduction

In literature various statistical models have been applied to appearance based terrain classification. Analogously, there are many image classifiers of varying complexity which achieve different levels of success in classification reported in the literature such as simple Fisher and Fisher Linear Discriminant Analysis [48], GMM [9] [10][11], Majority voting [24], rule-based terrain classifier (RBTC) [27], Nearest neighbour classification [48][30] Classification trees – Random forests [26], Fuzzy logic reasoning [27], Feed forward neural networks [49], [27], [28], SVM [12] [48] [29] [30], ELM [30].

In machine learning and statistics, classification is the problem of identifying to which set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. Any classification model is defined on the space N of maps from the image domain to the set C of classes (each class c corresponds to an entity of interest in the scene). Thus each classification $h \epsilon N$ assigns a class $C = h(z) \epsilon N$ to each observation z giving the class of that observation. Two methods of classification is employed in this thesis

1. GMM – Self-supervised online learning and classification.
2. MLP – supervised learning for offline pattern recognition and classification.

GMM captures linear relationships between features used, as a result of using Pearson correlation formula to form the covariance matrix. Hence, a classifier MLP, which learns nonlinearities between the features, is analysed to find a suitable classification mapping for each of the features discussed in chapter 3.

The rest of this chapter is followed by

1. Introduction
2. Block diagram and algorithm description of each of the classifier.

4.2 Gaussian mixture modeling

Autonomous robots have seen numerous applications in recent years as discussed in the previous chapters and they have been increasingly deployed in outdoor environments. Outdoor environments are notoriously unpredictable or unknown or changing and consequently, autonomous robots must be able to quickly adapt to these situations. Hence such robotic systems should be equipped with a learning strategy to adapt to complex environments.

GMM is a self-supervised online learning machine, allowing it to adapt to changing road conditions while making no assumptions about the general structure, geometric or appearance or both of the environment as compared to offline trained systems (e.g. decision trees, neural network based classifiers) which would most likely fail, due to changing illumination conditions and the high variability of the terrain characteristics. In this thesis, GMM follows a clustering analysis using K-means and classification based on Mahalanobis distance.

MOG with multiple components (K = 3) is used to accommodate for modeling heterogeneous terrain surfaces in the feature space, thus allowing to detect and learn terrain sub-classes in addition to terrain segmentation.

The online terrain perception and classification system used in this thesis is similar to the one in [9] [10] [11] where, terrain models of navigable terrain are learnt using self supervised learning. The significant difference between the two is at the training stage. The work by [9] [11] relies on a time of flight - range sensors to identify navigable regions in the perception space. The scheme in this thesis investigates suitability of monocular vision for terrain classification in complex environments.

The algorithm presented in this work relies on a self-supervised learning, trained by pre-filtering pixels in the perception space. The pre-filtered pixels are assumed to correspond to regions of free-space immediately in front of the rover. The vision algorithm learns terrains, as a Mixture of Gaussians. To learn terrain models of maneuverable regions, sub-space clustering is employed. The framework is used to estimate properties of distant terrain based on measurements from similar terrain in the short range. This strategy is popularly known as near to far range learning in the literature and is successfully employed in [8] [9][10][11][12][19][20].

4.2.1 Algorithm description

In this section an end-to-end computational flow of a visual classifier using images produced by a monocular camera is described. It relies on texture and color features to describe the appearance of ground. Ground can be detected at near ranges in the image frames, and also at a significant distance from the camera, thus providing long-range information. In addition, ground subclasses can be identified for terrain typing. The learning phase for the visual classifier is supervised by a range sensor (Kinect depth frame), which provides ground labels. The overview of the computational flow of the GMM - algorithm is as shown in Figure 4.1.

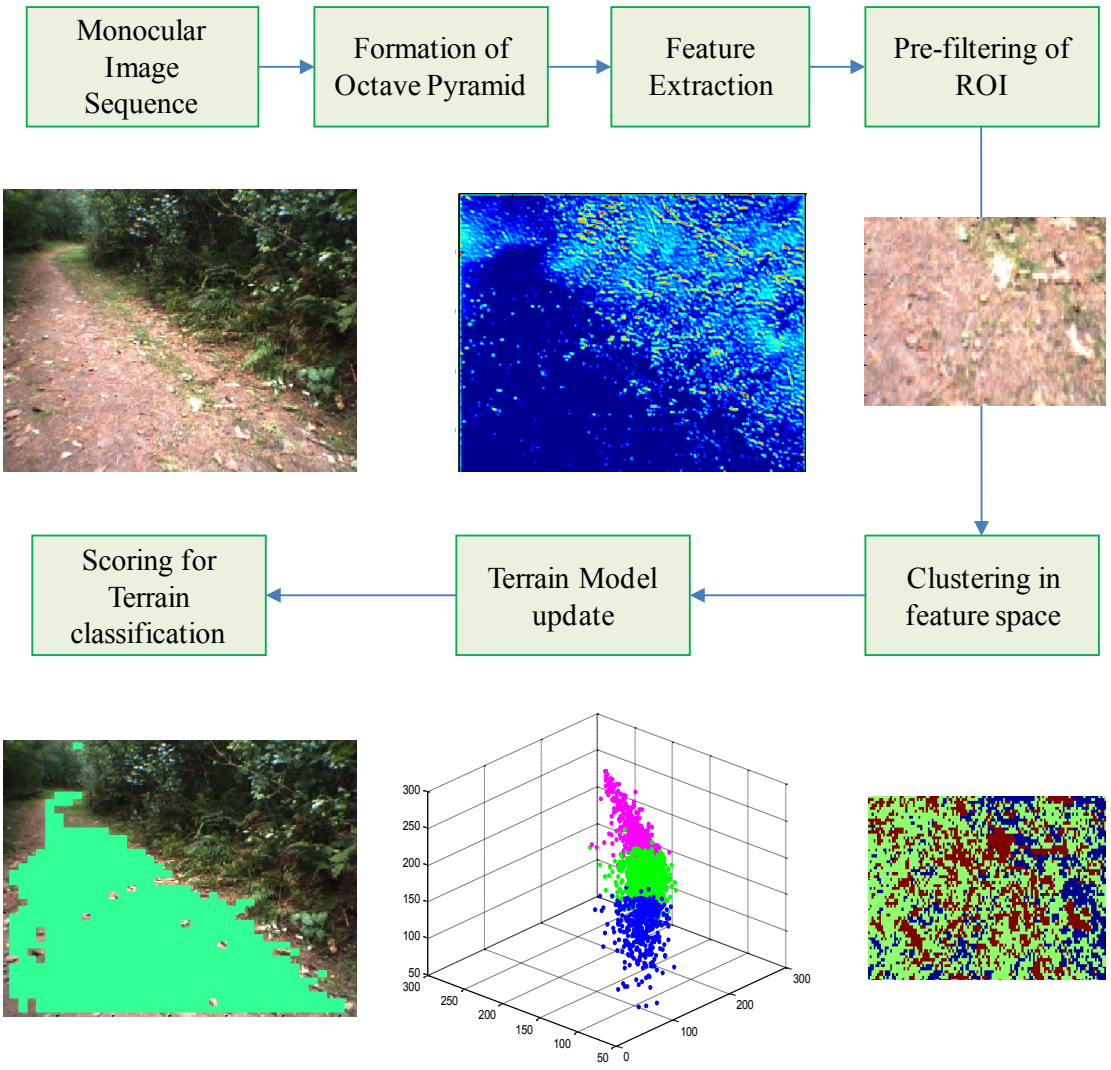


Figure 4.1: Overview of the computational flow of the GMM – terrain classification algorithm

In this thesis, a region of interest based scheme is employed as given in [10]. The major stages of the algorithm are as follows:

- Formation of octave pyramid.
- Feature extraction.
- Pre-filtering near-range observations which correspond to navigable terrain.
- Terrain modeling - K-Means clustering in the perception space.
- Terrain model update.
- Terrain classification - Scoring observations in the perception space, by learned models.

4.2.2 Formation of octave pyramid

Images acquired are convolved with a binomial filter and an octave pyramid is constructed, reducing the size of the image, thus, reducing the time and space complexity of the algorithm. Each frame, *f* in the incoming video feed is decimated as formulated in [50] to reduce the resolution resulting in a smaller search space for clustering. The incoming frame, f, is convolved with a low-pass filter, h as shown in equation 4.1.

$$o(i,j) = \sum_{k,l} f(k,l)h(i-k,j-l) \text{ where, } h = \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} \quad (4.1)$$

In the equation, pixel coordinates are denoted by the ordered-pair *(i, j)*, and the smoothening kernel *h(k, l)*. The input image is a color image of size (480, 640) acquired at 15 frames per second. The notation adopted is as follows: each image is represented as $f_{u,v,c}$ pixel coordinate is denoted by *(u, v)* and the corresponding color channel by *c*. Since the proposed method is applied to each of the color channels *c*, the subscript *c*, is dropped in the discussion, and operations are done on $f_{u,v}$.

4.2.3 Pre-filtering near-range pixels

After features are extracted as discussed in chapter 3, mean of the Region of Interest (ROI) is computed in this stage.

$z_{u,v} = \text{Feature\_extraction}\{o_{u,v}\}$, where $z_{u,v}$ is the feature image with N channels.

Computations are done on a ROI of size 64x128 in the image. It corresponds to the closest navigable region labeled by a proximity sensor, which later will be used as training samples.

Formally, let $\{(u_1, v_1), (u_2, v_2)\ldots, (u_m, v_m)\} \in S$ represent the position of pixel inliers in the ROI. The i^{th} pixel in the feature space is represented by $g(u_i, v_i) \in R^N$. $S = \{(u_i, v_j) \mid (i, j) \in D\}$, where D consists of pixel coordinates inside the rectangular ROI. Specifically, the cardinality of D is chosen to be a power of two, so division operations for the entire ROI reduce to bit-shifts so that it is amenable for real time embedded implementation. The ROI is chosen to be a window of size 64 x 128($m = 2^P$ X 2^Q). Consequently, pixels in the ROI are represented by $X \subset \{S \cap f\}$ (f is current image frame). Individual pixels in the ROI, are represented by $x(u, v) \in X$. The following notations are adopted when defining the ROI:

$\mu_X = [\mu_1 \quad \mu_2 \quad \cdots \quad \mu_N]$ - Mean of the ROI, N is the length of the feature vector.

The mean pixel value in the ROI is computed as given in equation (4.2). This equation reduces to an accumulate step and a bit shift as given in equations (4.3) and (4.4) respectively.

$$\mu_X = \sum_{(u,v)\,\epsilon\, s} z_{u,v}/2^{P+Q} \tag{4.2}$$

$$\text{Accumulate: } a_X = \sum_{(u,v)\in S} z_{u,v} \tag{4.3}$$

$$\text{Shift: } \mu_{ROI} = a_X \gg (P+Q) \tag{4.4}$$

4.2.4 Terrain modeling

The operations in pre-filter stage are done on a perception space X. The mean calculated in the pre-filter stage is used in cluster initialization stage. K-means clustering based on the method described in [51] is performed on X. Specifically, the training data set is $\{t^{(1)}, t^{(2)}, \ldots, t^{(m)}\} \in X$.

The K clusters (K is chosen to be 3 here), with centroid $\{\mu_1, \mu_2, \mu_3 \in R^N\}$ are initialized such that the centroids are uniformly distributed. The centroid of the clusters nearest to the ROI mean is replaced by mean μ_x. In the cluster assignment step, for the i^{th} training sample, the index of the cluster is computed using equation (4.5). After all the training samples have been assigned clusters, the centroids are updated using equation (4.6).

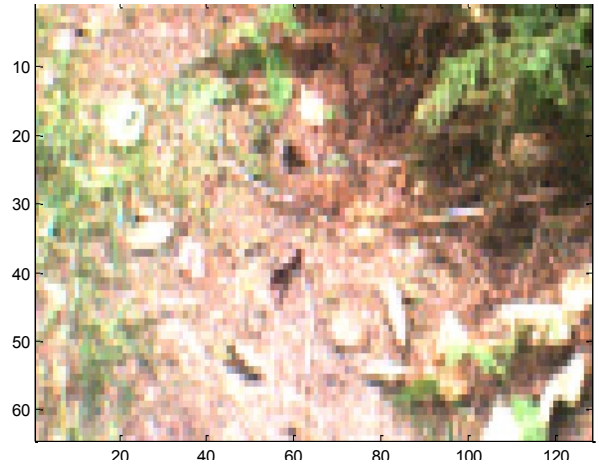
$$C^{(i)} \leftarrow k \text{ that minimizes } ||t^{(i)} - \mu_k|| \tag{4.5}$$

$$\mu_k \leftarrow \frac{1}{|W_k|}\sum_{i\in C_k} t^{(i)} * X\left(t^{(i)}\right), \text{ where, } W_k \text{ is the cluster weight} \tag{4.6}$$

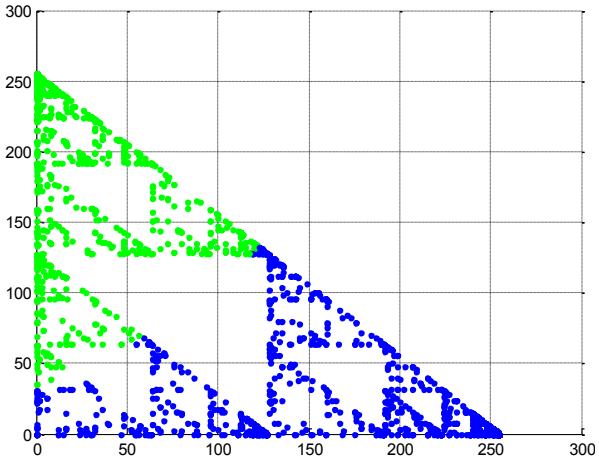
The cost function which is to be minimized is given by equation (4.7).

$$J\left(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots \mu_k\right) = \frac{1}{m}\sum_{i=1}^{m} ||x^i - \mu_{c^{(t)}}|| \tag{4.7}$$

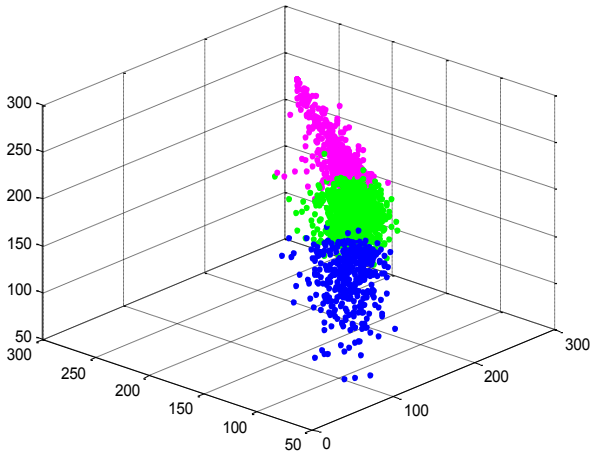
The clusters obtained by this process, as shown in Figure 4.2, forms the three models which are used for terrain learning. For each model, a co-variance matrix, $\sum(N \times N)$ is computed, where N is the length of the respective feature. The clustered images of the ROI employed for all the features investigated is shown in Figure 4.3.



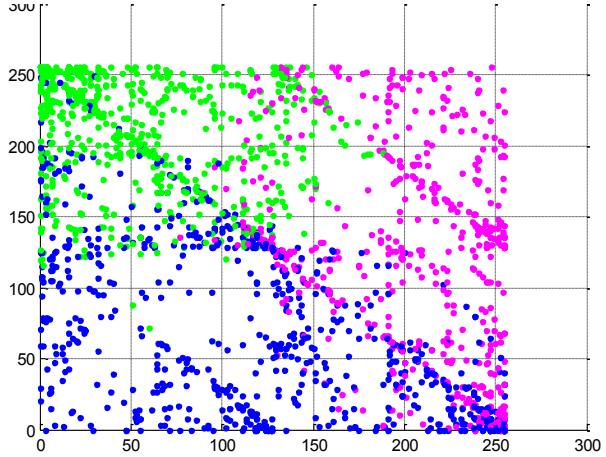
(a) ROI RGB Image



(c) LTP cluster

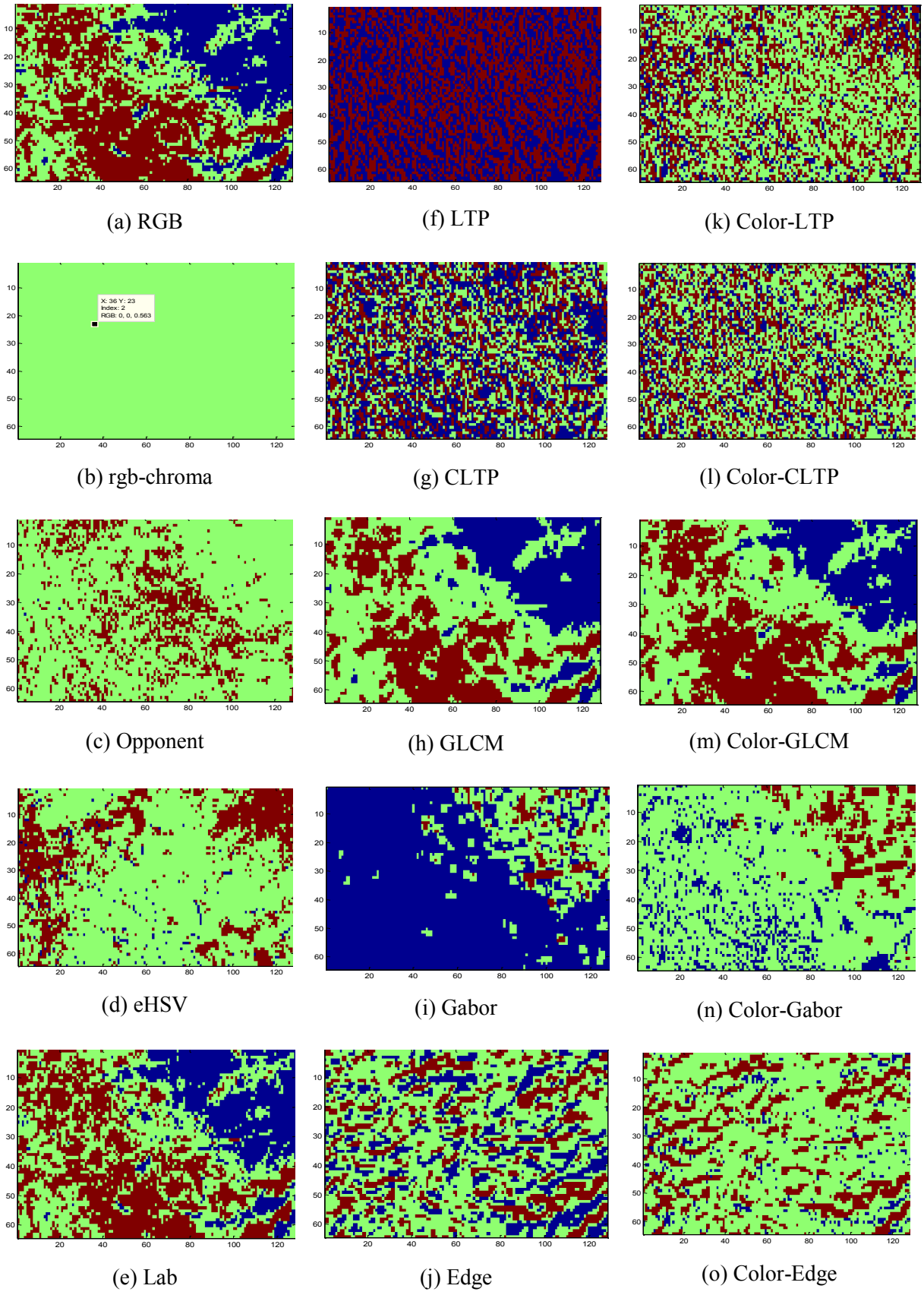


(b) RGB clusters



(d) CLTP Sign component cluster

Figure 4.2: K cluster components in feature space



(a) RGB (f) LTP (k) Color-LTP

(b) rgb-chroma (g) CLTP (l) Color-CLTP

(c) Opponent (h) GLCM (m) Color-GLCM

(d) eHSV (i) Gabor (n) Color-Gabor

(e) Lab (j) Edge (o) Color-Edge

Figure 4.3: Cluster index images

4.2.5 Terrain model update

To adapt to changing environment conditions a learning approach is adapted as in [9] [10] that allows the autonomous robot to continuously update new terrain models maintained in a Look Up Table (LUT). LUT also remembers past terrain models giving the online self-supervised framework better flexibility. Terrain model is automatically initialized at the beginning of the vehicle's operation and progressively updated.

The K color clusters are characterized by their mean (μ), covariance matrix (CM) and number of pixels in the cluster (or cluster weight W). In addition to the K training models, N learned models are present, which incorporate the past history of dominant terrain features, with N > K. When the condition in equation (4.8) is satisfied, learned model is updated using equations (4.9) to (4.11).

$$(\mu_L - \mu_T)(\sum_L - \sum_T)^{-1}(\mu_L - \mu_T)^T \leq 1 \tag{4.8}$$

$$\mu_L \leftarrow \frac{\mu_L W_L + \mu_T W_T}{W_L + W_T} \tag{4.9}$$

$$\sum_L \leftarrow \frac{\sum_L W_L + \sum W_T}{W_L + W_T} \tag{4.10}$$

$$W_L \leftarrow W_L + W_T \tag{4.11}$$

Where, subscripts L and T refer to learned models and training models respectively. If the current training model does not match any of the learned models, as per equation (4.08), then model update is performed using equation (4.12).

$$\left.\begin{matrix}\mu_L \leftarrow \mu_T \\ CM_L \leftarrow CM_T \\ W_L \leftarrow W_T\end{matrix}\right\} \; that\ maximizes\ W_n - \text{Summation of weights in the LUT} \tag{4.12}$$

4.2.6 Terrain classification

Given a new observation z, where z is either a radar feature vector in the radar classifier or a visual feature vector in the visual classifier the classification step is aimed at assessing whether the observation is an instance of ground or not. A Mahalanobis distance-based approach is adopted whereby the Mahalanobis distance and its distribution are employed to predict if a pattern has an extremely low probability of belonging to ground and may be suspected to be an outlier. In detail, the algorithm proceeds as follows. First, the squared Mahalanobis distance of the feature vector z with respect to each component of the current ground model is computed as given in equation (4.13).

$$d_i^2 = if\ (\mu_L - z_{u,v})\sum_i^{-1}(\mu_L - z_{u,v})^T \tag{4.13}$$

where μ_L is the mean value and $\sum_i$ is the covariance matrix of the i^{th} component for $i = 1, 2, \ldots, l$, l being the number of available terrain subclasses. Then, the minimum squared Mahalanobis distance $d_{min}^2 = min\{d_1^2, d_2^2, \ldots, d_k^2\}$ (i.e., the distance of z from the closest ground subclass) is found, and is compared with a cut-off threshold for classification as given in equation (4.14). Hence, any patch with minimum Mahalanobis distance d_{min}^2satisfying the inequality $d_{min}^2 >$ *cut-off threshold* may be suspected to be an outlier. Otherwise, it will be labeled as a ground.

$$f_{navigable=}\begin{cases}1, & if\ d_{min}^2 \leq 5\\ 0, & otherwise\end{cases} \tag{4.14}$$

The two main parameters on the outcome of the visual classifier are the number of Gaussian components for MOG fitting and the cut-off threshold. To introduce uniformity in comparing different feature spaces, the cut-off threshold was fixed.

4.3 Multi-layer Perceptron

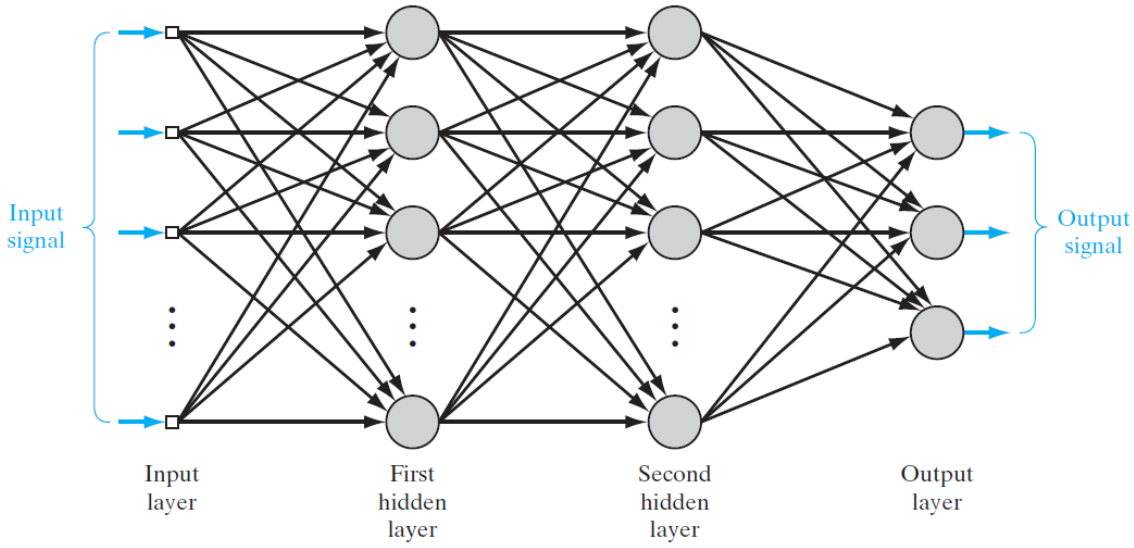


Figure 4.4: Architectural graph of a multilayer perceptron with two hidden layers [54]

Neural networks have become a popular technique for pattern recognition problems, including terrain classification [8][29][30][27][28][49]. Neural networks today are much more than just a simple MLP. Modular architectures, complex learning algorithms, auto associative and compression networks, and networks evolved or pruned with genetic algorithms are all examples of the widespread use of neural networks in pattern recognition.

In application such as road following for autonomous cars [52] [53] or where the perception environment is known a priori online learning will be redundant. Artificial Neural Networks are well suited for non-linear function approximation and therefore are theoretically capable, subject to proper training, of performing complex classification, data fusion and inference tasks. Hence, modeling terrain, using an offline trained architecture such as MLP is explored.

The multi-layer perceptron (MLP) is a representative static neural network that performs recognition or classification via a supervised learning method. The MLP has one or more hidden layers between its input and output layers as shown in Figure 4.4.

The navigable terrain is modelled as a pattern recognition and classification problem. The MLP network is trained offline to search for an optimal set of weights W^*; given an unknown function $g:X \rightarrow Y$ (the ground truth) that maps input instances $x \varepsilon X$ to output labels $y \varepsilon Y$, along with training data $D = \{(x_1,y_1),...,(x_n,y_n)\}$ representing examples of the mapping, produce a function $h:X \rightarrow Y$ that approximates as closely as possible the correct mapping $g:X \rightarrow Y$. The algorithm is divided in to two parts 1) Terrain modeling – Offline training of MLP with back propagation, using gradient descent. 2) Terrain classification – Hyper surface separation. Block diagram for the terrain classification algorithm with MLP architecture is shown in Figure 4.5.

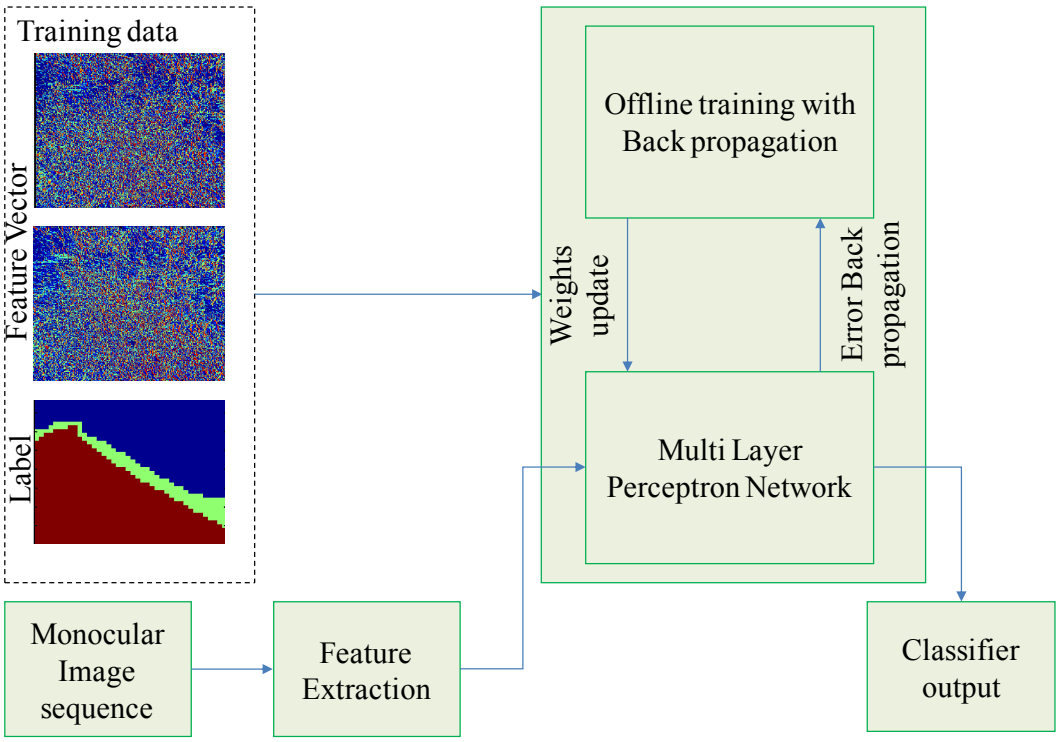


Figure 4.5: Block diagram –MLP Terrain classification

4.3.1 Terrain modeling

In this method terrain is modelled as a hyper surface separating two sets C_1, navigable and C_{-1} not navigable in the feature space, $\mathbb{R}^N$. A MLP network is trained offline with back propagation using gradient descent algorithm to minimize the cost function as given in equation (4.15).

$$\text{minimize } E_{av}(W) = \frac{1}{2}\sum_{n=1}^{N}\|h(z_n, W) - y_n\|^2 \tag{4.15}$$

where, $z \in Z$ and $y \in \mathbb{R}$. The global minimum W^*, is searched for this cost function, iteratively, until a stopping criterion is reached; employing the gradient descent algorithm as given below, as given in [54].

4.3.2 Gradient Descent Algorithm

Forward computation: Let a training example in the epoch be denoted by *(z(n), d(n))*, with the input vector *z(n)* applied to the input layer of sensory nodes and the desired response vector *d(n)* presented to the output layer of computation nodes. Compute the induced local fields and function signals of the network by proceeding forward through the network, layer by layer. The induced local field $v_j^{(l)}(n)$ for neuron *j* in layer *l* is computed as in (4.16).

$$v_j^l(n) = \sum_l w_{jl}^{(l)}(n) y_l^{(l-1)}(n) \tag{4.16}$$

Where $y_l^{(l-1)}(n)$ is the output signal of neuron *i* in the previous layer *l*-1 at iteration n, and $w_{jl}^{(l)}$ is the synaptic weight of neuron *j* in layer *l* that is fed from neuron *i* in layer *l*-1. For *i* = 0, we have $y_0^{(l-1)}(n)$ = +1, and $w_{j0}^{(l)}(n) = b_j^{(l)}(n)$ is the bias applied to neuron *j* in layer *l*. A nonlinear activation function (hyperbolic tangent) for each neuron in every layer is used. The output signal of neuron *j* in layer *l* is computed as in equation (4.17).

$$y_l^{(l)}(n) = \varphi(v_j(n)) \tag{4.17}$$

If neuron *j* is in the first hidden layer (i.e., *l* = 1), set as given in equation (4.18).

$$y_l^{(o)}(n) = x_j(n) \tag{4.18}$$

where $x_j(n)$ is the j^{th} element of the input vector *z(n)*. If neuron *j* is in the output layer (i.e., *l* = *L*, where L is referred to as the depth of the network), set as in equation (4.19).

$$y_l^{(L)}(n) = o_j(n) \tag{4.19}$$

The error signal is computed as in e equation (4.20)

$$e_j(n) = d_j(n) - o_j(n) \tag{4.20}$$

where $d_j(n)$ is the j^{th} element of the desired response vector d(n) which is scalar for our classification purposes.

Backward computation: The local gradients, δs of the network are computed as given in equation (4.21).

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n)\varphi_j'\left(v_j^{(L)}(n)\right) & for\ neuron\ j\ in\ output\ layer\ L \\ \varphi_j'\left(v_j^{(L)}(n)\right)\sum_k \delta_k^{(l+1)}(n)w_{kj}^{(l+1)}(n) & for\ neuron\ j\ in\ hidden\ layer\ l \end{cases} \tag{4.21}$$

where, $\varphi_j'\left(v_j(n)\right) = \frac{b}{a}\left(a - y_l(n)\right)\left(a + y_l(n)\right)$

The synaptic weight of the network in layer l is updated according to the generalized delta rule as given in equation (4.22)

$$w_{jl}^{(l)}(n+1) = w_{jl}^{(l)}(n) + \alpha\left[\Delta w_{jl}^{(l)}(n-1)\right] + \eta\delta_j^{(l)}(n)y_l^{(l-1)}(n) \tag{4.22}$$

where α is the learning-rate parameter and η is the momentum constant.

Iteration: Forward and backward computations are repeated with stochastic online training, by presenting new epochs of training examples to the network until the chosen stopping criterion is met.

4.3.3 Terrain classification

The trained network represented by (4.23) is modeled as a hyper surface, which separates the two classes. After the stopping criterion is met the network is used only in forward computation mode and presented with inputs for classification.

$$h(z_n, W) = \varphi\left(\sum_k w_{0k}\varphi\left(\sum_j w_{kk}\varphi\left(\ldots\varphi\left(\sum_i w_{0k}z_i\right)\right)\right)\right) \tag{4.23}$$

where, z_i is the i^{th} element of the input vector z. The mapping from feature space, Z to set of classes, $C = \{ C_1, C_{-1}\}$ is obtained using equation (4.24).

$$f_{navigable}: Z \rightarrow C = \begin{cases} 1, & h(z_n, W) > 0 \\ -1, & h(z_n, W) \leq 0 \end{cases} \tag{4.24}$$

Chapter 5
Datasets description

Two datasets were used to investigate the feature spaces and classifiers discussed in the previous chapter. The first dataset St Lucia, although structured was a difficult dataset to perform terrain learning as will be discussed in chapter 7. The second dataset employed for terrain learning was LAGR, represents an outdoor unstructured environment, with heterogeneous terrain profiles with grass and soil. The images from these databases were decimated to images of size (240, 320) for faster computation.

5.1 St Lucia

The standard data sets include monocular urban road sequences, from the RAT-SLAM field experiments as described by [55]. The dataset comprises of several miles of urban road, which includes moving vehicles and pedestrians during different times of the day. A segment of the route in the interval of frames 19800 and 19950 were used as test data as this part of the database was non-uniformly illuminated with, shadows and reflections. This dataset was the most difficult as a result of non-uniformity in illumination as seen in the experiments section. This data set is structured dynamic environment.

5.2 LAGR

The natural unstructured datasets in [56] are taken from logged field tests conducted by DARPA evaluators, and have been shown to contain time-varying (drifting) concepts. Each dataset consists of a 100 images and hand-labeled image sequence. Overall, three scenarios are considered. Each scenario is associated with two distinct image sequences, each representing a different lighting condition. There are thus six datasets total. The terrain appearing in the datasets varies greatly, and includes various combinations of ground type (mulch, dirt); foliage; natural obstacles (trees, dense shrubs); and man-made obstacles (hay bales). Lighting conditions range from overcast with good color definition (e.g., DS1B, shown above), to very sunny, causing shadows and saturation (e.g., DS2A). Additional descriptions and representative images from each dataset are available.

Chapter 6

Experimental setup

For combined color and texture classification the feature vector is constructed with the concatenation of color feature, z_{eHSV} and texture feature, $z_{texture}$. For GMM, all the feature vectors were scaled for best performance. For classification using MLP, the feature vectors are scaled and decorrelated.

6.1 Experimental setup for GMM

The algorithms that are discussed in chapter 4 runs through the images in the data base extracting features and performing classification. The parameter of GMM threshold is held constant across features. The LUT size is held constant at 27 models per database. Three clusters are used to learn terrain models from ROI and the entire image is scored using the LUT to classify terrain to navigable and not navigable.

For terrain classification using vision based sensors and image processing techniques the use of blocks instead of pixels introduces spatial dependence leading to better accuracy. In this thesis the features are extracted pixel wise and scoring for classification is also investigated on blocks for faster computation and examined for accuracy. The experiments are conducted for four image sizes – pixel wise, 10 x 10 blocks and 40 x 40 blocks.

6.2 Experimental setup for MLP

6.2.1 Architecture - To solve arbitrarily non-linear pattern classification, 3 different MLP network architectures are investigated, and, tangent sigmoid activation function. The number of nodes of the output layer is one corresponding to the binary classification.

6.2.2 Approach - To measure network performance, each dataset was partitioned into two disjoint parts: Training data and test data. The training data is further subdivided into a training set of examples used to adjust the network weights and a validation set of examples used to estimate network performance during training as required by the stopping criteria. The validation set is never used for weight adjustment. This decision was made in order to obtain pure stopping criteria results. In contrast, in a real application after a reasonable stopping time has been computed, one would include the validation set examples in the training set and retrain from scratch.

6.2.3 Learning Tasks - 15 features were used to learn for terrain classification. The problems have between 2 and 16 inputs, and one output, and between 2800 and 3600 examples.

6.2.4 Datasets sampling - The examples of each problem were partitioned into training (50%), validation (25%), and test set (25% of examples).

6.2.5 Training Algorithm - All runs was done using the backpropagation training algorithm as in [54] using the squared error function and the parameters:

- Weight initialization: $\left\{\frac{-1}{\sqrt{fan\ in}}, \ldots, \frac{1}{\sqrt{fan\ in}}\right\}$ uniformly distributed [57].
- Learning rate, output layer learns faster when compared to previous layers. Therefore η decreases as we approach output layer as in equation (6.1).

$$\eta(L) = \frac{\eta(0)}{\sqrt{\text{synaptic weights} * L}} \quad L = 1 \text{ for first hidden layer... N for the last} \quad (6.1)$$

- Momentum constant, μ= 0.75
- Of the available 1800 samples in training data 50% or less was randomly chosen for training in each epoch. A small learning rate 0.00001 was used as the number of data samples used are large in number and error converges faster without giving the stopping criterion a chance to stop at the appropriate point for generalization.

6.2.6 Stopping criterion - Network is trained until the condition given in equation (6.2) is satisfied.

$$E_{va} < 0.095 \quad (6.2)$$

Where, E_{va} is validation set error.

6.2.7 Datasets sampling for cross dataset training and validation - The network is trained using all the examples from the dataset (DS3B) and the other dataset (DS3A) is partitioned into validation (50% of examples), and test set (50% of examples).

All the computations are carried out in MATLAB run on i5 core processor. No libraries were used for feature extraction and classification. Image processing tool box was used for reading images and displaying. And symbolic math tool box was used for plotting graphs.

Chapter 7
Experimental Results and Analysis

7.1 Performance Metrics

The performance metrics adopted, to analyze the given binary terrain classification are accuracy, recall and false positive rate (FPR, fall-out). FPR is an important performance metric for a task such as terrain classification as the autonomous vehicle employed comes in harm's way, when an obstacle is classified as navigable terrain, endangering the machine. A good performance for this terrain classification task is characterized by high accuracy and high recall rates and low false positive rates.

Accuracy for binary classification is defined as the ratios of the summation of true positives (TP) and true negatives (TN) to the total number of observations as given in equation (7.1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \} \quad (7.1)$$

Where, FP – False positive and FN – False Negative. Recall is defined as the ratios of true positives and the number of true labels as given in equation (7.2).

$$recall = \frac{TP}{Total\ true\ conditions} * 100\% \} \quad (7.2)$$

False positive rate is defined the ratios of false positive and the number of false labels given in equation (7.3)

$$recall = \frac{FP}{Total\ false\ conditions} * 100\% \} \quad (7.3)$$

The chapter follows the results for each database with both the classifiers. Three databases are used for experimentation. The result of each classifier is organized in to three categories for each database Color only features, texture only features and color-texture features. Each classifier is investigated with three architectures.

7.2 Experiments on St Lucia dataset

7.2.1 Classification results of GMM

The classification results of color only features on the dataset for different image sizes are shown in Table 7.1. The results of texture only features classification is given in Table 7.2 and the results for color and texture features combined is given in Table 7.3.

Table 7.1: Color only features GMM - Classification results for St Lucia dataset

Features		Pixel based			Blocks (10, 10)			Blocks (40, 40)		
		Accuracy	Recall	FPR	Accuracy	Recall	FPR	Accuracy	Recall	FPR
RGB	μ	**68.31**	**91.71**	**43.10**	**71.14**	**95.23**	**39.93**	**69.18**	**93.69**	**42.55**
	σ	15.26	10.21	26.30	17.81	7.28	28.40	15.61	8.01	26.26
	min	47.37	69.96	11.40	43.33	80.76	4.47	46.25	76.78	10.83
	max	87.36	98.65	79.39	95.40	100	85	87.23	99.75	81.53
rgb-chroma	μ	**48.42**	**95.80**	**73.07**	**47.46**	**97.67**	**73.63**	**48.12**	**96.82**	**73.62**
	σ	16.93	7.55	28.48	19.19	4.43	29.07	17.83	6.78	29.25
	min	35.59	78.78	11.67	32.92	88.46	10.93	35.59	81.47	10.06
	max	85.45	99.92	98.64	88.88	100	98.33	87.39	100	99.20
Opponent	μ	**67.86**	**91.06**	**43.44**	**70.15**	**95.78**	**41.49**	**68.72**	**92.98**	**42.87**
	σ	14.95	10.22	25.96	17.23	6.05	27.57	15.15	8.52	25.73
	min	46.51	69.53	11.58	42.22	84.61	8.95	45.99	75.66	11.79
	max	85.28	98.20	80.38	91.95	100	86.66	86.15	99.75	81.63
eHSV	μ	**74.41**	**87.77**	**32.09**	**79.92**	**82.21**	**21.50**	**76.78**	**88.27**	**28.74**
	σ	10.58	11.79	18.45	16.03	21.38	19.43	11.43	11.57	18.36
	min	59.16	62.21	6.67	48.89	50	4.68	58.13	64.28	6.04
	max	83.89	96.12	57.58	91.46	100	53.33	86.79	96.46	55.48
LAB	μ	**69.87**	**90.51**	**40.16**	**71.83**	**95.78**	**38.98**	**70.42**	**92.70**	**40.18**
	σ	14.93	10.08	25.02	16.67	6.05	26.72	15.10	8.37	25.08
	min	44.60	69.18	10.94	43.33	84.61	10.93	44.50	75.22	10.73
	max	86.68	98.38	80.70	89.65	100	85	86.29	99.50	82.23

All numbers are in percentage.

The number of iterations for kmeans clustering made a huge difference in pixel wise classification results. A maximum of 3 iterations was employed; for iterations exceeding three the classification results dropped. It is seen that with color only features, the misclassification of non traversable regions as traversable, (i.e., false positives) are high, which is undesirable for an application such as terrain classification for autonomous mobile robots. The feature rgb-chroma although has the highest recall rate, so is the false positive rate, giving low accuracy rates. Although the feature, eHSV has lower recall, its false positive rate is the lowest. Consequently this color feature is preferred and used as the color feature for color and texture features.

Table 7.2: Texture only features GMM - Classification results for St Lucia dataset

Features		Pixel based			Blocks (10, 10)			Blocks (40, 40)		
		Accuracy	Recall	FPR	Accuracy	Recall	FPR	Accuracy	Recall	FPR
LTP	μ	**31.32**	**99.67**	**99.04**	**31.77**	**99.45**	**96.20**	**33.76**	**99.36**	**94.98**
	σ	4.76	0.86	2.52	8.12	1.45	10.03	9.77	1.68	13.26
	min	25.27	97.70	93.33	22.98	96.15	73.43	24.93	95.53	64.90
	max	38.30	100	100	46.67	100	100	53.25	100	100
CLTP	μ	**34.19**	**96.48**	**93.65**	**33.04**	**100**	**94.64**	**33.30**	**99.42**	**95.67**
	σ	5.22	2.24	7.35	11.01	0	14.17	8.72	1.51	11.45
	min	28.07	91.92	78.41	22.98	100	62.50	24.93	95.98	69.70
	max	42.89	98.46	98.38	55.55	100	100	50.03	100	100
GLCM	μ	**51.22**	**80.75**	**62.62**	**48.93**	**79.80**	**63.90**	**48.58**	**85.99**	**68.44**
	σ	12.18	9.03	22.62	14.15	23.40	26.25	12.83	9.57	23.62
	Min	42.14	62.30	21.82	38.83	38.46	17.18	37.40	65.84	25.50
	max	73.36	89.36	82.00	70	100	82.53	71.89	94.37	87.72
Gabor	μ	**51.11**	**89.78**	**66.72**	**43.73**	**91.44**	**76.23**	**50.24**	**90.39**	**67.90**
	σ	8.51	5.71	16.38	16.59	20.09	31.66	11.87	8.30	21.60
	min	41.24	79.34	37.06	26.82	46.15	7.81	35.68	75.66	27.99
	max	67.90	95.37	88.03	78.88	100	100	73.10	99.64	89.39
Edge	μ	**46.59**	**89.75**	**73.14**	**29.07**	**98.90**	**99.77**	**35.09**	**96.67**	**92.08**
	σ	5.30	4.34	11.07	4.81	2.90	0.59	5.98	5.19	10.33
	min	40.90	84.21	54.24	22.98	92.30	98.43	29.79	85.04	69.70
	max	57.40	97.41	85.57	37.07	100	100	46.74	99.82	99.20

All numbers are in percentage.

It can be seen from the Table 7.2 that, terrain classification using only texture features using GMM is inefficient. But it can be seen that Gabor texture feature and GLCM feature have the best accuracy and false positive rates. It can be seen form Figure 4.2 that LTP and CLTP do not form clusters to be modeled with GMM. These features are not well suited for GMM. Pixel features are better suited (exception: LTP) in comparison to mean of blocks for terrain classification using only texture features.

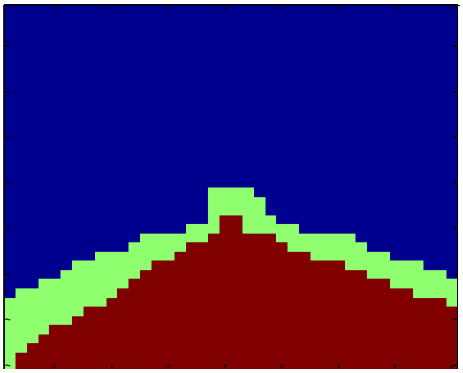
Table 7.3: Color and Texture features GMM - Classification results for St Lucia dataset

Features		Pixel based			Blocks (10, 10)			Blocks (40, 40)		
		Accuracy	Recall	FPR	Accuracy	Recall	FPR	Accuracy	Recall	FPR
Color-LTP	μ	**79.90**	**82.15**	**21.56**	**84.30**	**84.06**	**15.73**	**81.67**	**87.82**	**21.53**
	σ	8.67	13.36	14.25	14.63	23.66	15.36	9.67	11.41	15.58
	min	64.25	53.38	4.65	54.44	46.15	1.49	63.56	63.83	5.17
	max	89.34	91.67	44.67	97.70	100	45	92.27	97.35	48.40
Color -CLTP	μ	**82.65**	**75.79**	**14.98**	**84.97**	**89.12**	**16.93**	**81.87**	**86.53**	**20.78**
	σ	8.85	12.17	12.61	12.59	15.82	14.77	9.25	12.39	15.55
	min	64.88	58.14	2.20	60	63.33	1.56	65.37	60.04	1.82
	max	89.76	94.86	36.93	96.55	100	41.66	92.54	97.53	45.80
Color -GLCM	μ	**80.11**	**48.18**	**5.53**	**86.02**	**60.53**	**2.54**	**84.26**	**60.11**	**4.90**
	σ	4.97	11.76	4.09	9.08	26.45	2.46	5.21	14.14	3.78
	min	73.34	29.55	1.16	75.28	19.23	0	76.27	37.5	1.34
	max	87.06	57.92	10.79	96.34	90.47	6.66	90.38	72.62	10.67
Color -Gabor	μ	**81.25**	**67.82**	**13.74**	**81.40**	**54.99**	**8.13**	**83.04**	**70.87**	**12.37**
	σ	9.02	14.83	13.62	12.08	22.50	11.76	9.34	13.54	13.37
	min	61.70	43.39	0.71	56.66	23.07	0	63.11	44.86	0.76
	max	89.06	91.38	40.53	91.95	82.60	33.33	91.46	88.02	39.28
Color -Edge	μ	**80.19**	**73.38**	**17.64**	**79.55**	**94.89**	**27.41**	**80.15**	**89.74**	**24.88**
	σ	9.69	13.58	15.24	13.97	8.21	22.25	12.13	11.67	20.86
	min	59.93	50.40	1.93	54.44	76.92	3.12	57.29	65.17	2.10
	max	88.77	93.64	46.52	95.40	100	66.66	92.33	99.29	62.27

All numbers are in percentage.

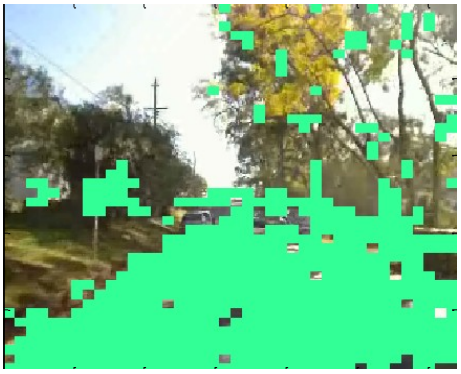
The average classification accuracy, recall and false positive rates for color-Texture features are better than color only and texture only features as seen in Table 7.3. Ternary patterns (LTP and CLTP) have the best classification accuracy and recall. But, their false positive rate is high. Feature, Color-GLCM has the best false positive rate although not the best recall. For features with low false positive rate and low recall, increasing the threshold, increases recall at the cost of increased false positive rate (GLCM) as shown in Table 7.5. It can be seen in Table 7.5, that increasing the threshold gives better recall and the best false positive rate. The classified images of the dataset St Lucia using GMM are shown in Figure 7.1.

(a) RGB 1



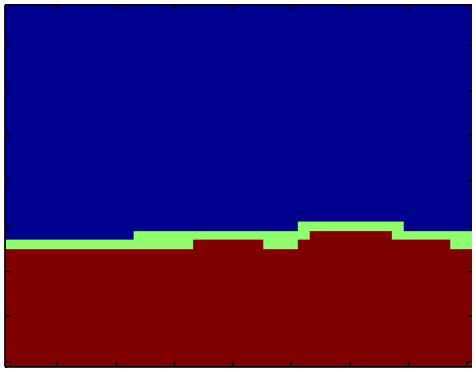
(b) Label 1

(c) Color-GLCM result 1



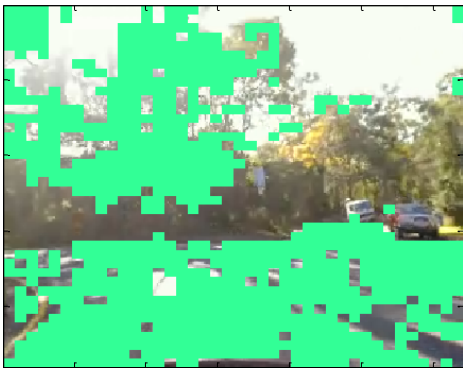
(d) Color-LTP result 1

(e) RGB 2



(f) Label 2

(g) Color-GLCM result 2



(h) Color-LTP result 2

Figure 7.1: Classification results, St Lucia – GMM. (a) and (e) RGB images. (b) and (f) Ground truth labels respectively. (c) and (g) Classified images using GMM with color-GLCM respectively. (d) and (h) Classified images using GMM with color-LTP respectively

A comparison of RGB-texture feature and eHSV texture feature is given in Table 7.4. It complies with the results from Table 7.1 that eHSV color-Texture feature performs better.

Table 7.4: Classification performances of RGB versus eHSV features combined with LTP

Block Size		RGB-LTP			eHSV-LTP		
		Accuracy	Recall	FPR	Accuracy	Recall	FPR
10 x 10	μ	74.9491	93.8978	33.7267	84.3020	84.0609	15.7312
	σ	17.4664	07.9040	27.1663	14.6372	23.6662	15.3601
	min	43.3333	80.7692	02.9851	54.4444	46.1538	01.4925
	max	96.5517	100.0000	83.3333	97.7011	100.0000	45.0000
40 x 40	μ	75.1161	91.4120	32.6560	81.6646	87.8287	21.5205
	σ	13.3418	09.9379	22.6538	09.6739	11.4112	15.5834
	min	51.6796	72.9911	08.6290	63.5659	63.8393	05.1774
	max	90.2554	98.9770	73.2535	92.2715	97.3510	48.4032
Pixel	μ	76.5693	86.2226	28.3868	79.9011	82.1564	21.5695
	σ	11.6346	11.2670	20.3676	08.6787	13.3656	14.2596
	min	55.6531	63.4750	08.0097	64.2560	53.3840	04.6529
	max	89.9244	95.5227	65.0430	89.3495	91.6786	44.6745

All numbers are in percentage.

In the Literature [11][58], a reduced feature set for GLCM as in [11] and for Gabor texture as in [58] is used. In Table 7.5 the classification results of the reduced features are given. Both the reduced feature sets performs better than their original feature counterparts. As a result they are more suitable for real time applications such as terrain classification particularly for GMM.

Table 7.5: Reduced feature size classification results and comparison

Block Size		Color - Reduced GLCM			Color - Reduced Gabor		
		Accuracy	Recall	FPR	Accuracy	Recall	FPR
10 x 10	μ	84.8440	66.3482	6.5687	84.0372	71.3758	10.5149
	σ	9.9947	28.7084	6.1002	13.7945	30.6545	12.2239
	min	71.1111	26.9231	0.0000	60.0000	23.3333	00.0000
	max	96.5116	95.2381	15.0000	98.8889	96.2963	29.5082
40 x 40	μ	83.7322	69.1581	9.6253	81.0613	75.5366	17.0694
	σ	6.4219	14.7562	6.5771	9.2670	12.0273	13.8800
	min	75.8713	45.3125	1.5340	65.6331	51.3393	1.3423
	max	92.6003	84.5475	16.9661	91.6321	86.7958	37.4251
Pixel	μ	80.4002	61.2208	10.9075	80.7681	78.3894	18.8555
	σ	6.3510	12.7537	7.3611	8.3143	12.3689	14.0679
	min	73.9414	39.1371	1.6369	67.7299	54.0676	1.8238
	max	88.8983	75.1827	19.7646	89.7851	90.8537	38.7609

All numbers are in percentage.

The classification results for color-GLCM feature shows a lower recall and good false positive rates as in Table 7.3. A lower false positive rate is desirable for a task such as terrain classification. Increasing the threshold for GMM increases both the recall and false positive rates. In this experiment the threshold for Mahalanobis distance criterion is increased from 10 to 15 and the results for color-Gabor and color-GLCM are given in Table 7.6.

Table 7.6: Classification results for increased threshold

Block Size		Color- Gabor			Color- GLCM		
		Accuracy	Recall	FPR	Accuracy	Recall	FPR
10 x 10	μ	79.5582	65.5204	15.0111	86.8116	72.6700	6.5163
	σ	12.5499	19.7457	14.8121	8.6218	26.9959	6.7531
	min	55.5556	34.6154	1.5625	76.4045	30.7692	0.0000
	max	93.1034	86.9565	41.6667	97.6744	100.0000	18.3333
40 x 40	μ	78.6008	77.0787	21.4496	83.4352	73.4159	11.9123
	σ	11.0622	11.9677	17.0602	5.4912	15.0019	8.4207
	min	57.5581	52.6786	1.1505	76.8088	47.9911	2.3969
	max	91.1962	89.0845	51.3972	87.4172	2.3969	23.4531
Pixel	μ	77.1001	77.0522	23.8762	81.7202	65.3631	10.9113
	σ	11.5217	12.8727	19.1443	4.9140	13.3952	7.2478
	min	54.9756	53.0083	1.5011	76.3978	40.8846	2.4691
	max	89.5867	93.7303	57.7929	88.4561	78.0086	21.5302

All numbers are in percentage.

7.2.2 Classification results of Multi Layer perceptron

Three architectures were used for this experiment. A maximum of 200 epochs were used to train the networks. The classification results are given in Table 7.7. None of the color only features and texture only features converged (marked '*'). And only three out of the 5 color-texture features converged to a minimum validation error of 0.095.

Different architectures and epoch limits were experimented with (eHSV - 5 hidden layers, with 100 neurons each for 500 iterations, CLTP, single hidden layer, with 1000 neurons, GLCM - 5 hidden layers, with 100 neurons each for 500 iterations, color-edge – single hidden layer, with 50 neurons for 1000 iterations). None of them converged or gave better classification results compared to the architecture resported in Table 7.7. The classified images of St Lucia dataset using MLP are shown as in Figure 7.2.

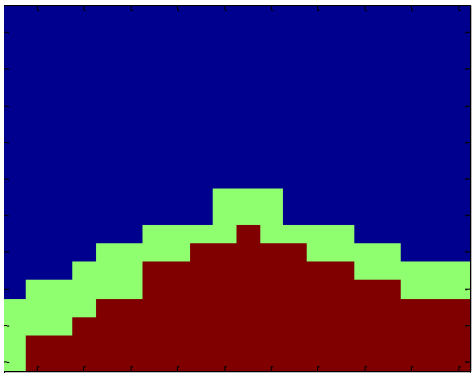
Table 7.7: MLP - Classification results on St Lucia dataset

Feature	Hidden Layers = 1, Neurons = 10			Hidden Layers = 2, Neurons = 50			Hidden Layers = 3, Neurons = 50		
	Accuracy	Recall	FPR	Accuracy	Recall	FPR	Accuracy	Recall	FPR
	Color Only Features								
RGB	71.83*	0*	0*	71.83*	0*	0*	71.83*	0*	0*
rgb-chroma	62.03*	0*	0*	62.03*	0*	0*	62.03*	0*	0*
Opponent	71.83*	0*	0*	71.83*	0*	0*	71.83*	0*	0*
eHSV	72.15*	10.11*	3.52*	72.15*	10.11*	3.52*	72.15*	10.11*	3.52*
LAB	71.83*	0*	0*	71.83*	0*	0*	71.83*	0*	0*
	Texture Only Features								
LTP	71.83*	0*	0*	71.83*	0*	0*	71.83*	0*	0*
CLTP	71.83*	0*	0*	71.83*	0*	0*	71.83*	0*	0*
GLCM	71.83*	0*	0*	71.83*	0*	0*	71.83*	0*	0*
Gabor	71.83*	0*	0*	71.83*	0*	0*	71.83*	0*	0*
Edge	71.83*	0*	0*	71.83*	0*	0*	71.83*	0*	0*
	Color and Texture Features								
C-LTP	90.58	84.55	5.72	92.74	87.39	3.98	91.97	85.77	4.22
C-CLTP	88.89	79.67	5.47	90.27	84.14	5.97	90.43	80.08	3.23
C-GLCM	90.58	82.11	4.22	91.04	83.73	4.47	91.66	88.21	6.21
C-Gabor	71.83*	0*	0*	71.83*	0*	0*	71.83*	0*	0*
C-Edge	73.10*	10.11*	2.20*	73.10*	10.11*	2.20*	73.10*	10.11*	2.20*

All numbers are in percentage.

Recall and false positive rate with zero entries, implies that all the outputs were mapped to negative i.e., not navigable. It is observed form the experiments that, increasing the number of layers, yielded a small gain in classification results. Color – LTP feature trained on a network with two hidden layers with 50 neurons each gave the best classification result for MLP. Local texture features, LTP, CLTP and, GLCM have performed better than filer bank texture approaches, Gabor and MPEG7.

(a) RGB 1

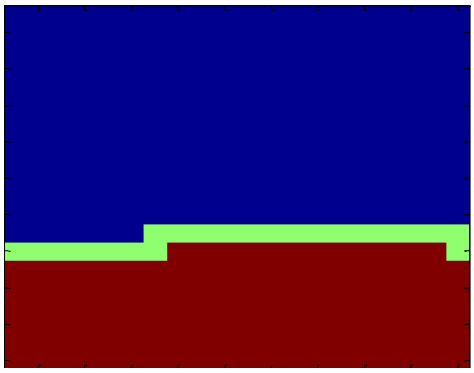


(b) Label 1

(c) Color-GLCM result 1

(d) Color-LTP result 1

(e) RGB 2



(f) Label 2

(g) Color-GLCM result 2

(h) Color-LTP result 2

Figure 7.2: Classification results, St Lucia – MLP. (a) and (e) RGB images. (b) and (f) Ground truth labels respectively. (c) and (g) Classified images using MLP with color-GLCM respectively. (d) and (h) Classified images using MLP with color-LTP respectively

7.3 Experimentations on LAGR dataset

7.3.1 Classification results of GMM

Table 7.8: Color only features GMM - classification results for LAGR dataset

Features		Pixel based			Blocks (10, 10)			Blocks (40, 40)		
		Accuracy	Recall	FPR	Accuracy	Recall	FPR	Accuracy	Recall	FPR
RGB	μ	**76.26**	**88.34**	**33.51**	**70.54**	**99.75**	**53.21**	**73.38**	**98.69**	**47.70**
	σ	8.21	5.26	18.05	18.75	0.72	31.90	14.54	2.28	26.16
	min	65.47	76.31	0.75	45.97	97.82	0	54.75	92.69	0.29
	max	89.95	93.13	48.79	100	100	83.92	96.59	99.86	71.38
rgb-chroma	μ	**75.14**	**90.12**	**37.05**	**61.91**	**100**	**69.56**	**65.10**	**98.44**	**62.95**
	σ	6.18	4.76	13.70	21.02	0	35.54	16.57	2.02	29.91
	min	67.29	82.71	11.78	37.34	100	5	45.80	94.35	7.13
	max	85.26	95.53	52.00	97.67	100	100	93.66	94.35	7.13
Opponent	μ	**75.62**	**91.21**	**37.20**	**69.76**	**99.75**	**54.84**	**72.49**	**99.07**	**49.83**
	σ	8.47	5.80	18.55	17.40	0.72	29.94	14.11	1.88	25.50
	min	65.17	76.55	1.95	47.12	97.82	0	54.75	94.10	1.18
	max	90.42	95.21	51.16	98.83	100	82.14	96.28	100	71.49
eHSV	μ	**82.41**	**81.70**	**16.14**	**87.44**	**84.75**	**9.90**	**86.09**	**85.37**	**12.79**
	σ	3.66	7.88	9.25	9.03	17.96	8.68	6.98	13.86	9.02
	min	76.74	63.12	0.74	74.41	52.17	8.68	75.37	61.02	0.14
	max	87.43	87.89	27.29	100	100	23.07	95.93	99.85	26.54
LAB	μ	**77.29**	**88.21**	**31.45**	**72.34**	**99.75**	**49.85**	**74.73**	**98.90**	**45.36**
	σ	7.62	4.70	16.75	17.17	0.72	28.87	13.62	1.81	24.11
	min	67.34	77.01	0.85	48.27	97.82	0	56.54	94.10	0.29
	max	90.11	91.69	46.02	98.83	100	80.35	96.69	99.73	68.44

All numbers are in percentage.

The classification results of GMM for LAGR dataset is summarized in Table 7.8 - color only feature classification, Table 7.9 for GMM - Texture only feature classification, Table 7.10 for GMM - color and texture feature classification and MLP – classification as in Table 7.11.

For experimentation, the LAGR dataset DS3A is used. For classification the threshold for Mahalanobis distance criterion was set to 5. It is observed that for color only feature classification, eHSV performs the best with lower false positive rates. The feature rgb-chroma was the worst performing color feature.

Table 7.9: Texture only features GMM - classification results for LAGR dataset

Features		Pixel based			Blocks (10, 10)			Blocks (40, 40)		
		Accuracy	Recall	FPR	Accuracy	Recall	FPR	Accuracy	Recall	FPR
LTP	μ	**54.50**	**88.60**	**75.36**	**46.49**	**100**	**99.72**	**47.83**	**99.54**	**97.63**
	σ	5.16	2.72	2.72	6.63	0	0.83	5.94	1.31	1.22
	min	45.93	83.93	71.49	35.63	100	97.50	37.46	96.02	95.98
	max	60.05	92.40	71.49	54.64	100	100	53.55	100	99.65
CLTP	μ	**52.72**	**79.91**	**71.18**	**46.36**	**100**	**100**	**46.85**	**100**	**100**
	σ	4.20	3.90	3.17	6.46	0	0	5.92	0	0
	min	46.19	72.33	65.26	35.63	100	100	36.70	100	100
	max	57.48	83.74	76.88	53.48	100	100	53.68	100	100
GLCM	μ	**68.18**	**41.47**	**7.95**	**77.90**	**66.96**	**12.13**	**74.17**	**62.18**	**14.60**
	σ	4.22	8.38	4.71	5.74	11.08	8.82	3.74	8.00	8.91
	Min	58.92	23.36	0.006	68.96	47.82	0	69.78	43.71	0
	max	72.15	50.91	12.88	85.39	82.92	28.57	81.57	70.77	26.55
Gabor	μ	**81.32**	**67.42**	**5.82**	**87.41**	**93.50**	**16.76**	**87.55**	**90.46**	**14.34**
	σ	4.93	9.91	3.42	6.17	14.15	10.75	4.44	12.93	8.50
	min	70.93	45.83	0.07	76,74	56.52	0	77.28	57.69	0
	max	86.49	77.75	9.48	96.42	100	30.23	92.98	97.23	23.24
Edge	μ	**45.12**	**79.45**	**85.31**	**46.36**	**100**	**100**	**46.84**	**99.84**	**99.87**
	σ	3.25	4.63	5.51	6.46	0	0	5.92	0.17	0.26
	min	40.40	72.43	76.006	35.63	100	100	36.70	99.48	99.27
	max	49.95	85.68	91.11	53.48	100	100	53.75	100	100

All numbers are in percentage.

Local texture patterns LTP and CLTP and filter bank approach edge texture, fail as they classify everything as navigable consequently resulting in very high recall and false positive rate.

Compared to the St Lucia dataset, Textures GLCM and Gabor independently achieve comparably good classification results. Filter bank texture response, Gabor performs the best in the texture only feature classification category.

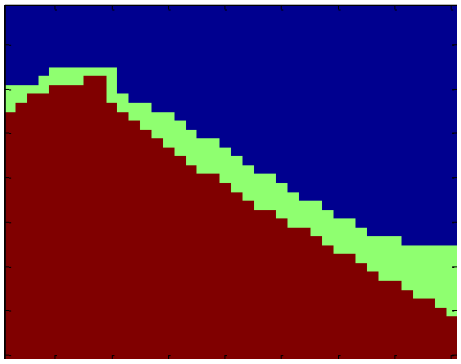
Table 7.10: Color and texture features GMM - classification results for LAGR dataset

Features		Pixel based			Blocks (10, 10)			Blocks (40, 40)		
		Accuracy	Recall	FPR	Accuracy	Recall	FPR	Accuracy	Recall	FPR
Color-LTP	μ	**80.84**	**62.39**	**2.47**	**96.64**	**94.69**	**1.13**	**93.71**	**89.32**	**1.76**
	σ	4.72	7.89	1.75	4,17	8.03	1.98	5.18	10.32	1.76
	min	70.38	44.97	0.06	86.04	73.91	0	80.59	63.84	0
	max	86.79	70.29	6.19	100	100	5.76	90.00	98.49	5.47
Color -CLTP	μ	**66.87**	**30.26**	**0.59**	**96.78**	**94.78**	**0.90**	**92.85**	**87.06**	**1.51**
	σ	5.15	3.61	0.55	4.56	8.87	2.10	4.55	8.86	1.95
	min	58.33	22.26	0.006	84.88	71.73	0	81.48	65.51	0
	max	75.13	34.12	1.95	100	100	6.25	97.86	96.43	6.29
Color -GLCM	μ	**60.91**	**18.12**	**1.24**	**78.14**	**58.93**	**4.92**	**74.24**	**51.27**	**4.99**
	σ	4.90	2.68	0.81	3.50	7.03	4.66	3.90	6.90	4.14
	min	53.05	12.41	0	72.09	47.82	0	65.86	36.41	0
	max	68.45	21.42	2.27	82.02	70.73	13.46	78.03	57.50	12.57
Color -Gabor	μ	**74.38**	**47.18**	**1.21**	**90.68**	**89.05**	**7.01**	**83.63**	**71.68**	**5.49**
	σ	5.63	7.03	0.82	8.03	14.68	6.97	5.24	11.22	4.14
	min	64.41	33.60	0	74.41	52.17	0	72.19	48.20	0
	max	82.16	58.99	2.51	98.83	100	21.15	89.27	83.57	13.15
Color -Edge	μ	**72.36**	**42.39**	**0.98**	**97.56**	**96.22**	**0.90**	**93.91**	**88.78**	**1.06**
	σ	4.98	6.09	0.67	2.47	4.57	2.10	3.67	6.92	1.08
	min	61.92	28.96	0	91.86	84.78	0	84.72	71.53	0
	max	78.93	49.19	2.24	100	100	6.25	97.65	96.24	3.40

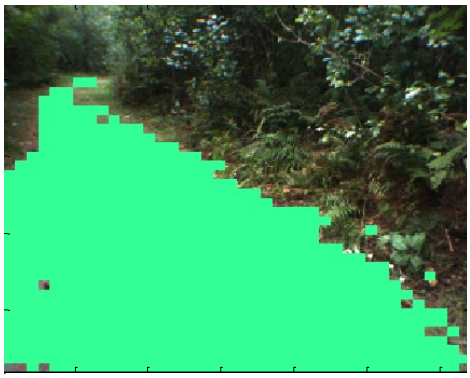
All numbers are in percentage.

- Color and texture features gave better classification results compared to color only and texture only classification results as seen in Table 7.10.
- The color-edge feature performed the best, with high recall and low false positive rate.
- Block based image sizes gave better performances than pixel based classification. Block image of size 10 x 10 gave the best classification performance for LAGR dataset.
- The classified images of the LAGR dataset, using GMM, are shown as in Figure 7.3.

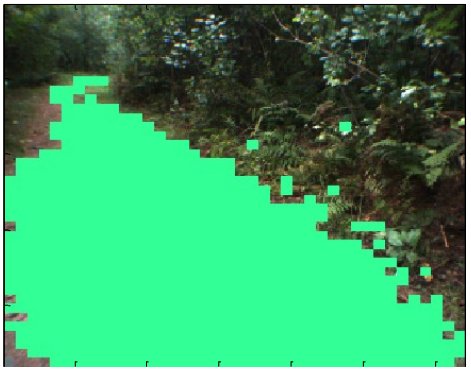
(a) RGB 1



(b) Label 1

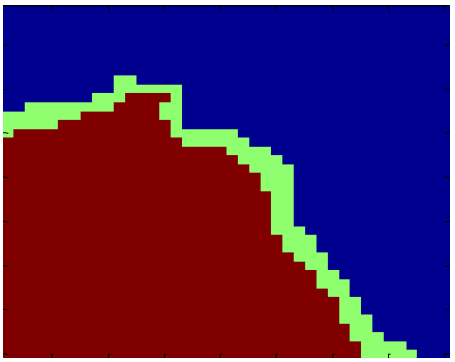


(c) Color-Edge result 1

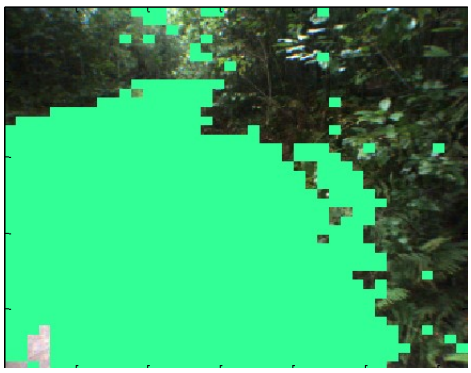


(d) Color-LTP result 1

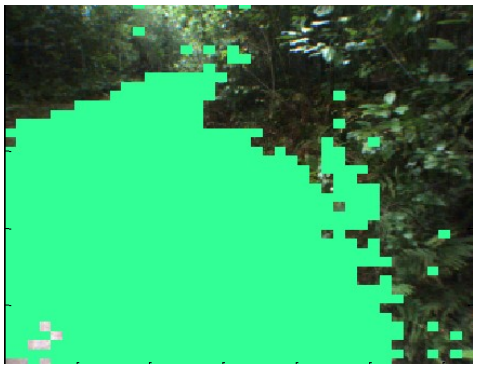
(e) RGB 2



(f) Label 2



(g) Color-Edge result 2



(h) Color-LTP result 2

Figure 7.3: Classification results, LAGR DS3A – GMM. (a) and (e) RGB images. (b) and (f) Ground truth labels respectively. (c) and (g) Classified images using GMM with color-Edge respectively. (d) and (h) Classified images using GMM with color-LTP respectively

7.3.2 Classification results of MLP

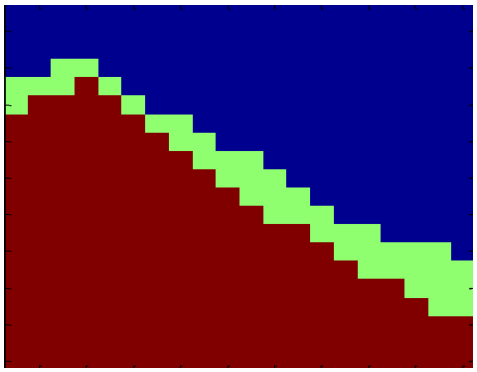
Table 7.11: MLP - Classification results on LAGR dataset

Features	Hidden Layers = 2, Neurons = 10			Hidden Layers = 2, Neurons = 25			Hidden Layers = 2, Neurons = 50		
	Accuracy	Recall	FPR	Accuracy	Recall	FPR	Accuracy	Recall	FPR
	Color Only Features								
RGB	94.69	91.19	1.08	94.69	91.19	1.08	95.43	93.45	2.17
rgb-chroma	88.90*	84.65*	5.97*	89.27*	87.35*	8.42*	88.90*	86.68*	8.42*
Opponent	94.94	92.77	2.44	94.69	91.64	1.63	94.82	92.55	2.44
eHSV	97.41	95.71	0.54	97.04	94.80	0.27	94.82	90.52	0
LAB	95.19	95.03	4.61	95.19	93.67	2.98	95.06	93.43	2.98
	Texture Only Features								
LTP	66.21*	52.82*	17.66*	66.21*	52.82*	17.66*	66.21*	52.82*	17.66*
CLTP	60.04*	42.21	18.47*	60.04*	42.21	18.47*	60.04*	42.21	18.47*
GLCM	94.57	91.87	2.17	93.71	90.06	1.90	95.68	94.58	2.98
Gabor	95.43	95.03	4.07	95.68	95.93	4.61	96.17	96.61	4.34
Edge	48.58*	9.02*	3.80*	48.58*	9.02*	3.80*	48.58*	9.02*	3.80*
	Color and Texture Features								
C-LTP	96.42	93.90	0.54	94.82	98.19	9.29	95.93	92.77	0.27
C-CLTP	95.43	97.51	7.06	96.30	94.13	1.08	95.93	94.58	2.44
C-GLCM	96.42	93.45	0	96.67	93.90	0	97.28	95.48	0.54
C-Gabor	96.55	96.89	4.61	97.16	96.61	2.17	95.43	98.41	8.15
C-Edge	95.56	91.87	0	95.93	92.77	0.27	95.93	92.55	0

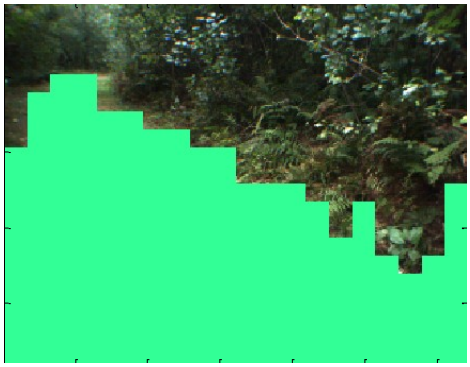
All numbers are in percentage.

Three architectures were used for this experiment. All the architecture for this experiment had 2 hidden layers with different number of neurons in each layer (10, 25 and 50). A maximum of 100 epochs were used to train the networks or till the validation error dropped to 0.095 or below. The classification results are summarized in Table 7.11. The color feature eHSV gave the best performance. It can be seen that color-texture features were more robust to false positive rates compared to color only features and texture only features. Gabor feature performed the best among texture only features and color-Gabor feature among the color-texture feature. The classified images of the LAGR dataset, using MLP, are shown as in Figure 7.4.

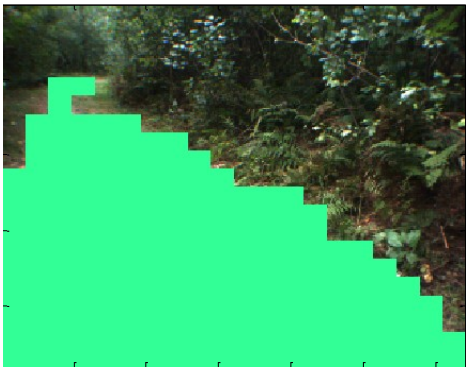
(a) RGB 1



(b) Label 1

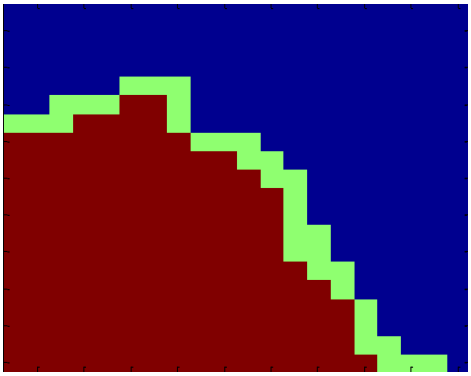


(c) eHSV result 1

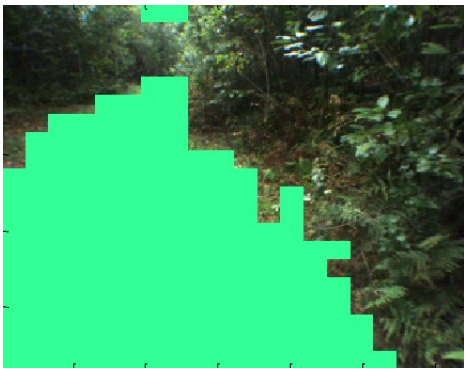


(d) Color-GLCM result 1

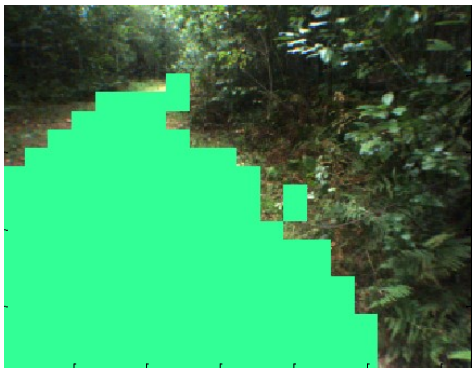
(e) RGB 2



(f) Label 2



(g) eHSV result 2



(h) Color-GLCM result 2

Figure 7.4: Classification results, LAGR DS3A – MLP. (a) and (e) RGB images. (b) and (f) Ground truth labels respectively. (c) and (g) Classified images using MLP with eHSV respectively. (d) and (h) Classified images using MLP with color-GLCM respectively

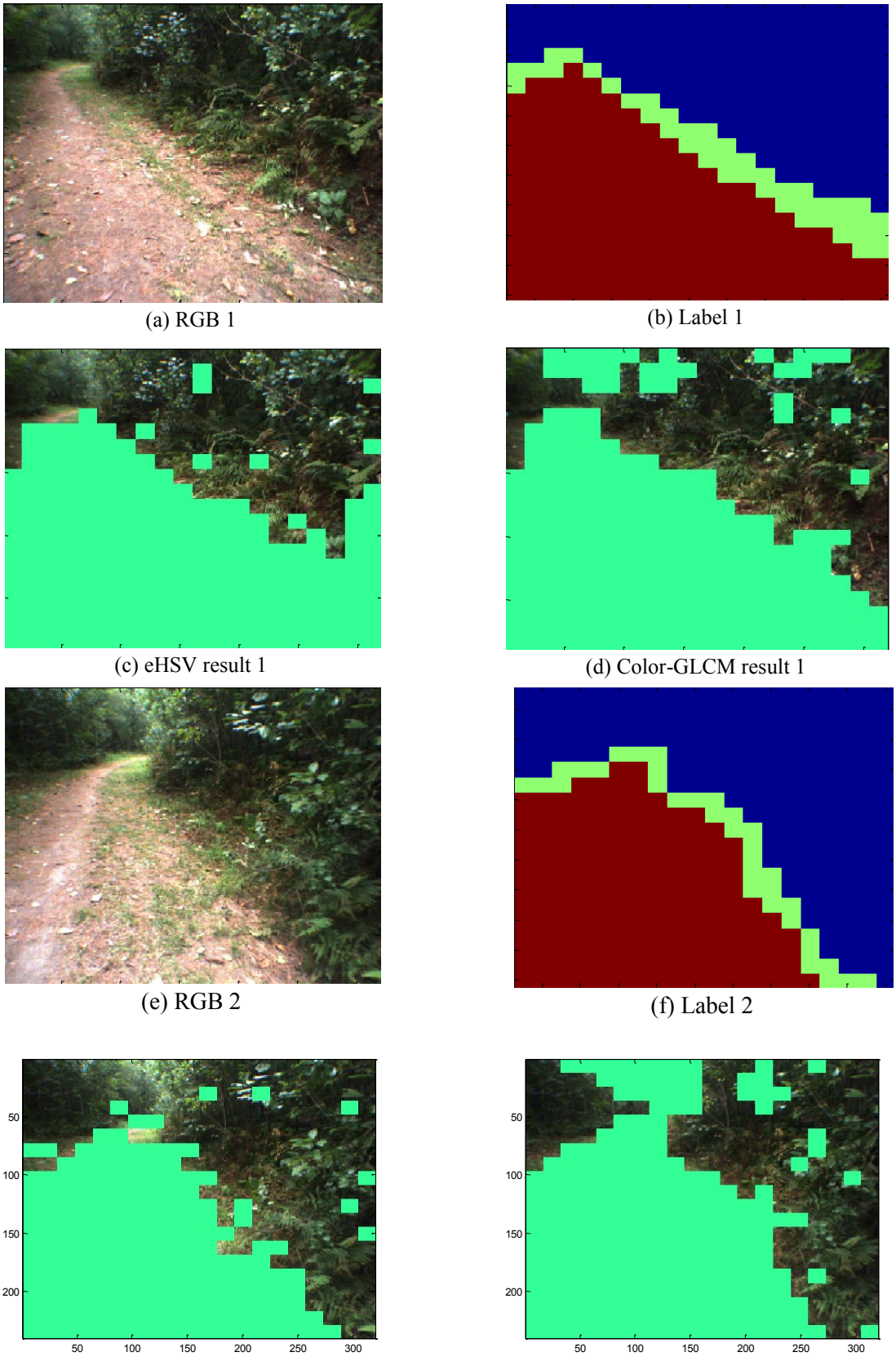
7.3.3 Cross dataset training and validation using LAGR datasets DS3A and DS3B

In real word scenarios, a neural network is trained offline and the classification task is performed online. Classification tasks employing outdoor images suffer from the problem of noise due to changing illumination conditions.

Table 7.12: MLP - Classification results with cross dataset training and validation on LAGR dataset

Features	Hidden Layers = 2, Neurons = 50		
	Accuracy	Recall	False positive
	Color Only Features		
RGB	22.01*	44.55*	55.01*
rgb-chroma	22.01*	44.55*	55.01*
Opponent	22.01*	44.55*	55.01*
eHSV	66.04*	31.66*	0.36*
LAB	22.01*	44.55*	55.01*
	Texture Only Features		
LTP	22.01*	44.55*	55.01*
CLTP	22.01*	44.55*	55.01*
GLCM	22.01*	44.55*	55.01*
Gabor	22.01*	44.55*	55.01*
Edge	22.01*	44.55*	55.01*
	Color and Texture Features		
C-LTP	92.82*	93.24*	7.57*
C-CLTP	22.01*	44.55*	55.01*
C-GLCM	89.79*	98.99*	19.19*
C-Gabor	41.00*	44.55*	17.48*
C-Edge	41.66*	44.30*	15.89*

In this experiment only one network is used with architecture of two hidden layers with 50 neurons in each hidden layer. The dataset DS3B is used to train the network and the dataset DS3B is used to validate and test the trained network. The datasets were constructed in different illumination conditions. The summary of the classification results are given in Table 7.12. None of the features trained converged hence all features used trained the network for 100 epochs without converging to the validation error limit 0.095. Only two of the Color-Texture features give acceptable classification results; eHSV color feature gave the best performance otherwise. The color-texture features color-LTP and color-GLCM give good classification results and color-LTP yielded the best performance. The classified images for this experiment are shown in Figure 7.5.



(a) RGB 1 (b) Label 1

(c) eHSV result 1 (d) Color-GLCM result 1

(e) RGB 2 (f) Label 2

Figure 7.5: Classification results, cross DS 3A and DS3B– MLP. (a) and (e) RGB images. (b) and (f) Ground truth labels respectively. (c) and (g) Classified images using MLP with eHSV respectively. (d) and (h) Classified images using MLP with color-GLCM respectively

Chapter 8
Appearance based Metric Map Generation

Relating image pixels to metric space on the ground plane is a radial/tangential distortion (taking the camera image into the equivalent pin-hole camera model), followed by a plane-induced homography which relates these pixels to the ground plane. Relating pixels to physical locations can be written as a linear mapping in homogeneous coordinates, once the camera is calibrated. Using this knowledge, a standard model can be used to calibrate the camera and estimate the pixel to floor mapping. Rectifying the image, however, is computationally expensive. In this specific case, the correspondence between pixels and the physical space is established at system initialization by learning structure from raw-sensor depth using a Kinect sensor as in [59].

The sensor space consists of 480 X 640 pixels. The sensor space corresponds to a 2D representation of the real world. However, the map represents only the plan (Top View) of the environment. To transfer points from the sensor space to the map, we consider only those parts of the sensor space which correspond to the floor or the surface (Terrain) in the real world. The terrain learning and classification algorithm discussed in the thesis gives the pixels corresponding to navigable terrain. For pixels which map to the floor or surface, one pixel in the image corresponds to a trapezium in the real world, as shown in the Figure 1

The set of points on a line segment (X_1, X_2) is given by the equation (8.1) as in [60].

$$Y = X_2 + \theta(X_1 - X_2) \tag{8.1}$$

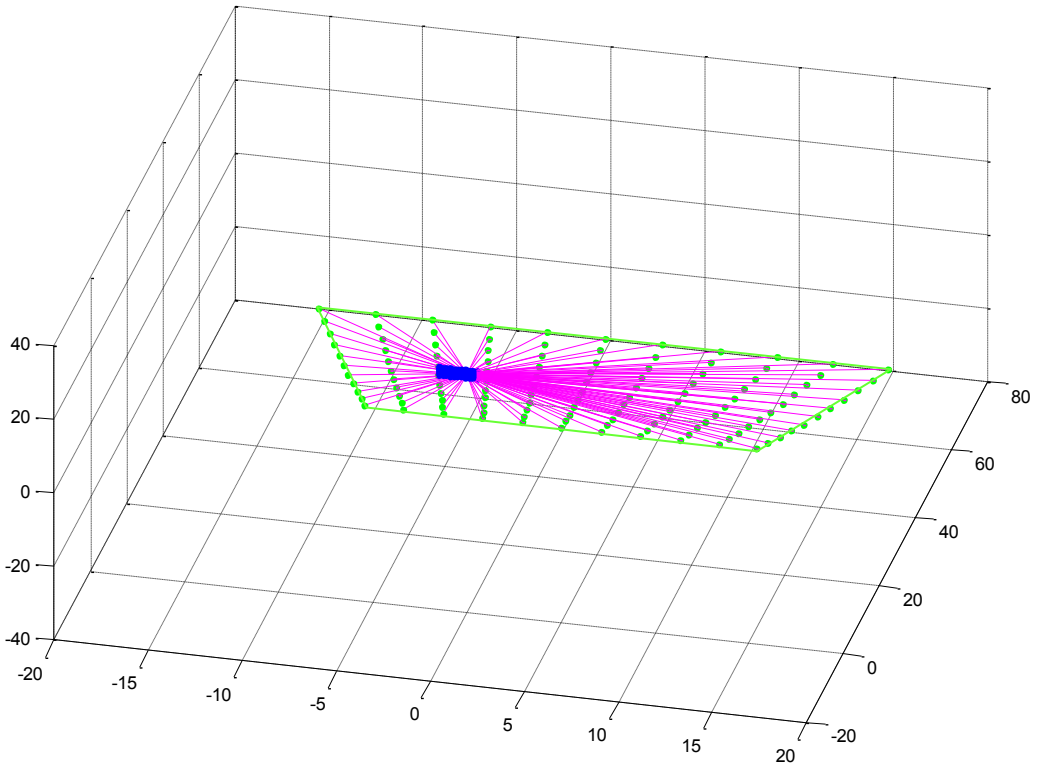
To project the pixel space to physical space the following transformations are used as given in (8.2) and (8.3).

$$t = \sqrt{\frac{distance^2}{\|f - p\|_2^2}} \tag{8.2}$$

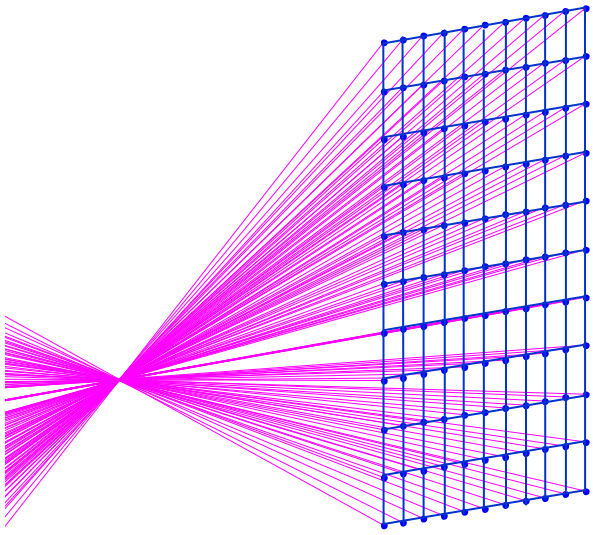
Where, f is the real word coordinates of the focal point of the Kinect camera(f_x, f_y, f_z), p are the four corners of the pixel (p_1, p_2, p_3 and p_4), which is projected to the physical space on a metric grid to form a trapezoid.

$$\begin{pmatrix} PJ_x \\ PJ_y \\ PJ_z \end{pmatrix} = \begin{pmatrix} p_{ix} \\ p_{iy} \\ p_{iz} \end{pmatrix} + t .* \begin{pmatrix} f_x - p_{ix} \\ f_y - p_{iy} \\ f_z - p_{iz} \end{pmatrix} \tag{8.3}$$

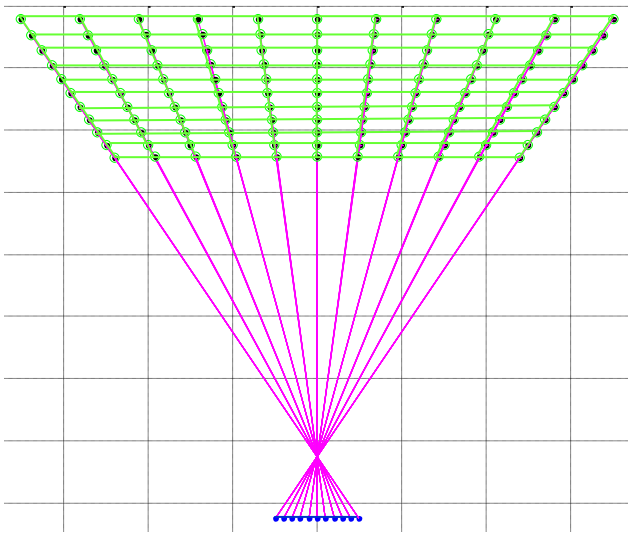
Where, PJ (PJ_x, PJ_y, PJ_z) are the four corners of the trapezium projected to the physical space. The projection of a typical pixel array through the focal point and to the physical space is depicted as in Figure 8.1. All the navigable pixels are projected on a 2D rover centric metric map as shown in Figure 8.2.



(a) Trapezium formed by a (n, n) pixel array



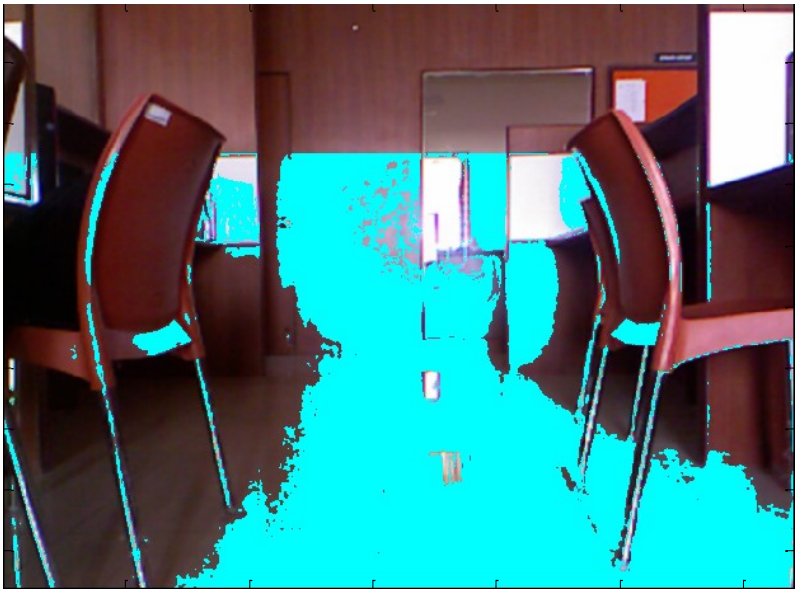
(b) Pixel array and the focal point



(c) Top view of the trapezium formed.

Figure 8.1: Perception space to physical space projection of a pixel array

(a) Depth frame extracted from the Kinect sensor



(b) Classified Image

(c) LocalRover centric metric map

Figure 8.2: Appearance based rover centric metric map generation

Chapter 9
Conclusions and Future scope

In this thesis a self-supervised terrain learning framework, GMM and a fully supervised neural network framework, MLP were employed for learning terrain in outdoor environments for autonomous mobile robot navigation. GMM was employed for online terrain learning for classification and MLP was trained offline using hand labelled images for online classification.

In order to find the most appropriate image feature descriptor and classifier for appearance based terrain learning and classification for outdoor autonomous mobile robots, fifteen features were investigated (five colors, five textures and five color-textures), with two different classifiers. Two different datasets were used for validation purposes and 3 architectures for each classifier were employed for evaluation. Among the color features eHSV had better performances across datasets and classifiers and among texture features LTP and GLCM were consistent accordingly.

The dataset St Lucia was set in a structured environment. In this dataset the terrain is non-uniformly illuminated with shadows of vegetation and buildings and reflections from the setting sun. It is observed from the experiments conducted that using the first order statistic, mean on larger block sizes (in this case (10, 10)) resulted in better performance of terrain classification compared to smaller block sizes and pixel based terrain classification approaches for the self supervised GMM classifier. For GMM classifier, color-Gabor, color-GLCM followed by color-LTP and color-CLTP performed best. In the case of MLP classifier only color-texture features yielded acceptable classification performances. To make the features amenable to real-time implementation experiments were conducted on reduced features of GLCM and Gabor respectively and it is observed that they perform better than their original counterparts.

The classification results for the unstructured off-road dataset LAGR were relatively better in comparison to the structured St Lucia dataset as the terrain was uniformly illuminated. For GMM classifier, color-edge followed by the local texture features color-LTP and color-CLTP performed the best accordingly. In case of the MLP classifier, the color feature eHSV gave the best performance followed by color-LTP and color-GLCM accordingly. In the experiment using cross dataset training and validation color-LTP performed the best followed by color-GLCM.

In summary, among the color features investigated eHSV feature gave the best classification results in both uniformly illuminated, unstructured LAGR dataset and non-uniformly illuminated, structured St Lucia dataset.

For the GMM classifier, among the texture features, GLCM and Gabor texture features gave the best performance independently. Gabor magnitude response texture feature gave good classification results for the structured dataset LAGR. When, color and texture features were combined, color-LTP and color-CLTP features gave good classification results, but the performance of color-GLCM and color-Gabor at small block sizes gave the best classification results. For the MLP classifier, the feature color-LTP performed the best.

It is observed from Table 7.3 and Table 7.7 that MLP classifier gave better classification results than the GMM classifier. Consequently it is concluded that when the environment is known apriori, MLP is more suitable for the terrain classification application.

The MLP classifier, for terrain classification in outdoor scenario is suitable when the environment is known and there is enough dataset and label to train the network. With new developments in hardware architectures [61], where real-time online training is feasible, pattern recognition for terrain classification using an online trained MLP architecture is desirable for outdoor autonomous vehicles in unknown environments as the space complexity of an MLP $(\boldsymbol{\theta}(1))$ remains constant across features and is lighter compared to the LUT used for GMM $(\boldsymbol{\theta}(N^2))$. But, the online learning framework of GMM is suitable in environments where the terrain model is dynamic and the mission is exploratory in nature.

Future Scope

In this Thesis, it is assumed that observations are spatially uncorrelated. The classification performance of both the classifiers can be improved by relaxing the independence of pixels in the local neighborhoods and by modeling correlations between pixels in the submodular MRF framework as in [25] and modeling observations in super pixels instead of blocks as in [24].

References

[1] Goldberg, S.B.; Maimone, M.W.; Matthies, L., "Stereo vision and rover navigation software for planetary exploration," *Aerospace Conference Proceedings, 2002. IEEE* , vol.5, no., pp.5-2025,5-2036 vol.5, 2002. doi: 10.1109/AERO.2002.1035370

[2] Poppinga, Jann, Andreas Birk, and Kaustubh Pathak. "Hough based terrain classification for realtime detection of drivable ground." *Journal of field Robotics* 25, no. 1-2 (2008): 67-88.

[3] Habib, Maki K. "Humanitarian demining: Reality and the challenge of technology-the state of the arts." *International Journal of Advanced Robotic Systems* 4, no. 2 (2007): 151-172.

[4] Raibert, Marc, et al. "Bigdog, the rough-terrain quadruped robot." *Proceedings of the 17th World Congress*. 2008.

[5] Astuti, G., Gaetano Giudice, Domenico Longo, C. Donato Melita, Giovanni Muscato, and Angelo Orlando. "An overview of the "Volcan Project": An UAS for exploration of volcanic environments." *Journal of Intelligent and Robotic systems* 54, no. 1-3 (2009): 471-494.

[6] Thrun, Sebastian, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong et al. "Stanley: The robot that won the DARPA Grand Challenge." *Journal of field Robotics* 23, no. 9 (2006): 661-692.

[7] Song, Dezhen, Hyun Nam Lee, Jingang Yi, and Anthony Levandowski. "Vision-based motion planning for an autonomous motorcycle on ill-structured roads."*Autonomous Robots* 23, no. 3 (2007): 197-212.

[8] Hadsell, Raia, Pierre Sermanet, Jan Ben, Ayse Erkan, Marco Scoffier, Koray Kavukcuoglu, Urs Muller, and Yann LeCun. "Learning long-range vision for autonomous off-road driving." *Journal of Field Robotics* 26, no. 2 (2009): 120-144.

[9] Dahlkamp, Hendrik, Adrian Kaehler, David Stavens, Sebastian Thrun, and Gary R. Bradski. "Self-supervised Monocular Road Detection in Desert Terrain." In*Robotics: science and systems*. 2006.

[10] Mishra, Prabhakar, and Anirudh Viswanathan. "Computationally Inexpensive Appearance Based Terrain Learning in Unknown Environments." *Journal of Artificial Intelligence and Soft Computing Research* 3, no. 3 (2013): 201-213.

[11] Milella, Annalisa, Giulio Reina, and James Underwood. "A Self-learning Framework for Statistical Ground Classification using Radar and Monocular Vision." *Journal of Field Robotics* (2014).

[12] Christopher A. Brooks*, "*Learning to visually predict terrain properties for planetary robots*",* PhD Thesis, Masachuesetts Institute of Technology (2009).

[13] Vandapel, Nicolas, Raghavendra Rao Donamukkala, and Martial Hebert. "Unmanned ground vehicle navigation using aerial ladar data." *The International Journal of Robotics Research* 25.1 (2006): 31-51.

[14] Lalonde, Jean-François, Nicolas Vandapel, Daniel F. Huber, and Martial Hebert. "Natural terrain classification using three-dimensional ladar data for ground robot mobility." *Journal of field robotics* 23, no. 10 (2006): 839-861.

[15] Singh, Sanjiv, Reid Simmons, Trey Smith, Anthony Stentz, Vandi Verma, Alex Yahja, and Kurt Schwehr. "Recent progress in local and global traversability for planetary rovers." In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2, pp. 1194-1200. IEEE, 2000.

[16] Wettergreen, David, Paul Tompkins, Chris Urmson, Michael Wagner, and William Whittaker. "Sun-synchronous robotic exploration: Technical description and field experimentation." *The International Journal of Robotics Research* 24, no. 1 (2005): 3-30.

[17] Larson, Jacoby, and Mohan Trivedi. "Lidar based off-road negative obstacle detection and analysis." In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 192-197. IEEE, 2011.

[18] Ishigami, Genya, Keiji Nagatani, and Kazuya Yoshida. "Path planning and evaluation for planetary rovers based on dynamic mobility index." In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 601-606. IEEE, 2011.

[19] Angelova, Anelia, Larry Matthies, Daniel Helmick, and Pietro Perona. "Learning and prediction of slip from visual information." *Journal of Field Robotics* 24, no. 3 (2007): 205-231.

[20] Karlsen, Robert E., and Gary Witus. "Terrain understanding for robot navigation." In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 895-900. IEEE, 2007.

[21] Helmick, Daniel, Anelia Angelova, and Larry Matthies. "Terrain adaptive navigation for planetary rovers." *Journal of Field Robotics* 26, no. 4 (2009): 391-410.

[22] Manduchi, Roberto, Andres Castano, Ashit Talukder, and Larry Matthies. "Obstacle detection and terrain classification for autonomous off-road navigation." *Autonomous robots* 18, no. 1 (2005): 81-102.

[23] Manduchi, Roberto. "Learning outdoor color classification." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28, no. 11 (2006): 1713-1723.

[24] Kim, Dongshin, Sang Min Oh, and James M. Rehg. "Traversability classification for ugv navigation: A comparison of patch and superpixel representations." In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 3166-3173. IEEE, 2007.

[25] Vernaza, Paul, Ben Taskar, and Daniel D. Lee. "Online, self-supervised terrain classification via discriminatively trained submodular Markov random fields." In*Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 2750-2757. IEEE, 2008.

[26] Khan, Yasir Niaz, Philippe Komma, and Andreas Zell. "High resolution visual terrain classification for outdoor robots." In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 1014-1021. IEEE, 2011.

[27] Shirkhodaie, Amir, Rachida Amrani, and Edward Tunstel. "Soft computing for visual terrain perception and traversability assessment by planetary robotic systems." In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, vol. 2, pp. 1848-1855. IEEE, 2005.

[28] Sung, Gi-Yeul, Dong-Min Kwak, and Joon Lyou. "Neural network based terrain classification using wavelet features." *Journal of Intelligent & Robotic Systems*59, no. 3-4 (2010): 269-281.

[29] Filitchkin, Paul, and Katie Byl. "Feature-based terrain classification for littledog." In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 1387-1392. IEEE, 2012.

[30] Zou, Yuhua, Weihai Chen, Lihua Xie, Pumit Su and Xingming Wu. "Comparison of different approaches to visual terrain classification for outdoor mobile robots."*Pattern Recognition Letters* 38 (2014): 54-62.

[31] Wietrzykowski, Jan, and Dominik Belter. "Boosting support vector machines for RGB-D based terrain classification." *Journal of Automation Mobile Robotics and Intelligent Systems* 8 (2014).

[32] Paul Jansen, "Color based terrain cover classification for off-road autonomous navigation", MS Thesis, University of Amsterdam (2004).

[33] Gevers, Theo, and Arnold WM Smeulders. "Color-based object recognition."*Pattern recognition* 32, no. 3 (1999): 453-464.

[34] Vertan, Constantin, and Nozha Boujemaa. "Color texture classification by normalized color space representation." In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 3, pp. 580-583. IEEE, 2000.

[35] Lee, Te-Won, Thomas Wachtler, and Terrence J. Sejnowski. "Color opponency is an efficient representation of spectral properties in natural scenes." *Vision research* 42, no. 17 (2002): 2095-2103.

[36] Todt, Eduardo, and Carme Torras Genís. "Color constancy for landmark detection in outdoor environments." (2001).

[37] Sofman, Boris, Ellie Lin, J. Andrew Bagnell, John Cole, Nicolas Vandapel, and Anthony Stentz. "Improving robot navigation through self-supervised online learning." *Journal of Field Robotics* 23, no. 11-12 (2006): 1059-1075.

[38] Tomasi, Carlo, and Roberto Manduchi. "Bilateral filtering for gray and color images." In *Computer Vision, 1998. Sixth International Conference on*, pp. 839-846. IEEE, 1998.

[39] Tan, Xiaoyang, and Bill Triggs. "Enhanced local texture feature sets for face recognition under difficult lighting conditions." In *Analysis and Modeling of Faces and Gestures*, pp. 168-182. Springer Berlin Heidelberg, 2007.

[40] Rassem, Taha H., and Bee Ee Khoo. "Completed Local Ternary Pattern for Rotation Invariant Texture Classification." *The Scientific World Journal* 2014 (2014).

[41] Guo, Zhenhua, and David Zhang. "A completed modeling of local binary pattern operator for texture classification." *Image Processing, IEEE Transactions on*19, no. 6 (2010): 1657-1663.

[42] Haralick, Robert M., Karthikeyan Shanmugam, and Its' Hak Dinstein. "Textural features for image classification." *Systems, Man and Cybernetics, IEEE Transactions on* 6 (1973): 610-621.

[43] Arvis, Vincent, Christophe Debain, Michel Berducat, and Albert Benassi. "Generalization of the cooccurrence matrix for colour images: application to colour texture classification." *Image Analysis & Stereology* 23, no. 1 (2011): 63-72.

[44] Burt, Peter J., and Edward H. Adelson. "The Laplacian pyramid as a compact image code." *Communications, IEEE Transactions on* 31, no. 4 (1983): 532-540.

[45] Bolcskei, Helmut, Franz Hlawatsch, and Hans G. Feichtinger. "Equivalence of DFT filter banks and Gabor expansions." In *SPIE's 1995 International Symposium on Optical*

Science, Engineering, and Instrumentation, pp. 128-139. International Society for Optics and Photonics, 1995.

[46] Won, Chee Sun, Dong Kwon Park, and Soo-Jun Park. "Efficient use of MPEG-7 edge histogram descriptor." *Etri Journal* 24, no. 1 (2002): 23-30.

[47] Chatzichristofis, Savvas A., and Yiannis S. Boutalis. "CEDD: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval." In*Computer Vision Systems*, pp. 312-322. Springer Berlin Heidelberg, 2008.

[48] Michael J. Procopio, "Long-term terrain learning using multiple models for outdoor autonomous robot navigation", PhD Thesis, University of Coloroado (2007).

[49] Howard, Ayanna, and Homayoun Seraji. "Vision-based terrain characterization and traversability assessment." *Journal of Robotic Systems* 18, no. 10 (2001): 577-587.

[50] Szeliski, Richard. *Computer vision: algorithms and applications*. Springer, 2010.

[51] Duda, Richard O., Peter E. Hart, and David G. Stork. "Pattern classification. 2nd." Edition. New York (2001).

[52] Thrun, Sebastian, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong et al. "Stanley: The robot that won the DARPA Grand Challenge." *Journal of field Robotics* 23, no. 9 (2006): 661-692.

[53] Markoff, John. "Google cars drive themselves, in traffic." *The New York Times*10 (2010): A1.

[54] Haykin, Simon, and Neural Network. "A comprehensive foundation." *Neural Networks* 2, no. 2008.

[55] Milford, Michael J., and Gordon F. Wyeth. "Mapping a suburb with a single camera using a biologically inspired SLAM system." *Robotics, IEEE Transactions on* 24, no. 5 (2008): 1038-1053.

[56] Procopio, M. J. "Hand-labeled DARPA LAGR data sets." (2007): 2006-2007.

[57] LeCun, Yann A., Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. "Efficient backprop." In *Neural networks: Tricks of the trade*, pp. 9-48. Springer Berlin Heidelberg, 2012.

[58] Tang, Isabelle, and Toby P. Breckon. "Automatic road environment classification." *Intelligent Transportation Systems, IEEE Transactions on* 12, no. 2 (2011): 476-484.

[59] Mishra P, Viswanathan A, Srinivasan A. "A supervised learning approach to far range depth estimation using a consumer-grade rgb-d camera." In Electronics, Computing and Communication Technologies (CONECCT), 2013 IEEE International Conference, pp. 1-6, 2013.

[60] Boyd, Stephen, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2009.

[61] Omondi, Amos R., and Jagath Chandana Rajapakse, eds. *FPGA implementations of neural networks*. Vol. 365. New York, NY, USA:: Springer, 2006.

[62] Urmson, Chris, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, M. N. Clark, John Dolan et al. "Autonomous driving in urban environments: Boss and the urban challenge." *Journal of Field Robotics* 25, no. 8 (2008): 425-466.