

Tree Classification Using Aerial Imagery



By

Talha Abubakr

**SUBMITTED TO THE FACULTY OF ELECTRICAL ENGINEERING
MILITARY COLLEGE OF SIGNALS,
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,
RAWALPINDI, PAKISTAN**

Nov 2019

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by Mr **Talha Abubakr of MSEE-23 Course**, Registration No **NUST00000203924** of **Military College of Signals** has been vetted by undersigned, found complete in all respect as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial, fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members of the student have been also incorporated in the said thesis.

Signature: _____

Name of Supervisor: Lt. Col Hasnat Khurshid, PhD

Date: _____

Signature (HoD): _____

Date: _____

Signature (Dean): _____

Date: _____

Tree Classification using Aerial Imagery

National University of Science & Technology (NUST)

By

Talha Abubakr

November 10, 2019

Declaration

I hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification either at this institution or elsewhere.

Dedication

I dedicate this thesis to my PARENTS and TEACHERS for their love, endless support and encouragement. I also want to offer special thanks to my parents, who instilled in me the love of learning from an early age. My parents have been constant cheerleaders through every academic and personal endeavor in my life. Thanks mom and dad for always believing in me and for encouraging me to strive for my dreams.

Acknowledgement

All praise to Allah, the Almighty, for blessing me and providing me the strength to complete this thesis. I would like to convey my gratitude to my supervisor Dr. Hasnat Khurshid for his supervision, guidance and continuous support. His invaluable help of constructive comments and suggestions throughout the experimental and thesis works are major contributions to the success of this research. I also want to express my sincere gratitude to Mr. Omer Latif who gathered and provided dataset for my thesis. Last, but not the least, I am highly thankful to my parents. They have always stood by my dreams and aspirations and have been a great source of inspiration for me. I would like to thank them for all their care, love and support through my times of stress and excitement.

Abstract

Type of a tree or species of a tree is its most extraneous property. Remote sensing has served as great solution to the problems related to manually acquiring data for tree species classification. Most of the studies in this area of research use Lidar & satellite imagery or combination of data sources. But these techniques have some disadvantages like Lidar is not cost effective in smaller areas and ineffective in rainfall. Hyper-spectral data is not feasible and Multi-spectral data are prone to factors such as viewing angle, sun angle, day / year time (seasons). Some studies have used optical imagery with the combination of other data sources. The optical imagery that have been used does not provide aerial view but side view of the tree which is not feasible to acquire for large areas. Therefore, there is a need for the solution which is effective, economical and scalable. A little research has been done on the classification, with over head view, using low cost commercial drones with optical sensors.

In this research optical aerial imagery has been used for tree species classification. Combination of statistical and spatial features is used as input to the two classifiers artificial neural network (ANN) and support vector machine (SVM). The results show that 97.4% and 96.26% accuracy, respectively, has been achieved.

Contents

1	Introduction	10
1.1	Overview	10
1.2	Problem Statement	11
1.3	Aims and Objectives	12
2	Literature Review	13
2.1	LiDAR	13
2.2	Multispectral & Hyper-spectral	16
2.3	Optical Imagery	19
2.4	Combination of data sources	21
3	Methodology	24
3.1	Study Area and Data Set	25
3.2	Pre-Processing	31
3.3	Feature Extraction	33
3.3.1	Colour Images	33
3.3.2	DCT	35
3.3.3	GLCM	37
3.3.4	Intensity	44
3.3.5	LBP	48
3.3.6	Gabor Filter	52
4	Classifier	56
4.1	Artificial Neural Network	56
4.1.1	Training a neural network	61

CONTENTS

4.1.2	Types of neural networks	64
4.1.3	Learning Techniques used in Neural Networks:	66
4.1.4	Advantages and applications of neural network	67
4.2	SVM	68
4.2.1	Tuning Parameters of SVM	71
4.2.2	Cost Function and Gradient Update	74
5	Performance Evaluation	77
5.1	Experiments	78
5.1.1	Experiment 1	79
5.1.2	Experiment 2	81
5.1.3	Experiment 3	83
5.1.4	Experiment 4	84
6	Conclusion & Future Work	85

List of Figures

3.1	Method adopted for Tree Species Classification	24
3.2	Orthomap of the orchards. Musammi(left), Guava(bottom), Fruiter(right)	25
3.3	DJI Phantom 4 Pro	26
3.4	Types of outputs by Pix4D Mapper	28
3.5	Number of overlapping images computed for each pixel of the ortho- mosaic	29
3.6	Drone trajectory to capture pictures at different position starting at big blue dot	30
3.7	Tree images extracted from orthomap	31
3.8	RGB Image decomposed into 3 channels	33
3.9	HSV color model	34
3.10	(a) Original Picture (b) Frequency Domain (c)DCT Coefficients Low, Medium and High	36
3.11	Original Image(Left) & its GLCM(Right)	37
3.12	Different angles at which GLCM can be calculated	38
3.13	Positively skewed, Normally Distributed and negatively skewed His- tograms	46
3.14	Platykurtic & Leptokutic	47
3.15	Grayscale image divided into cells	48
3.16	Thresholding to make binary pattern	48
3.17	Converting binary code to decimal value	49
3.18	Orientations of gabor filters	52
4.1	Human biological neuron	56

LIST OF FIGURES

4.2	Artificial neuron	57
4.3	Types of activation function	59
4.4	Architecture of a neural network	60
4.5	A simple neural network	61
4.6	Error calculation process	63
4.7	Types of neural networks	65
4.8	Different hyperplanes	68
4.9	Different hyper-planes	69
4.10	Good and bad margins	70
4.11	Classes distributed with simple separation	70
4.12	Classes distribution with complex boundary	71
4.13	Transition data into 3-D	71
4.14	Transition back to 2-D after drawing hyper-plane	72
4.15	Overlapping distribution and possible hyper-planes	72
4.16	Left: Low regularization value, Right: High regularization value	73
4.17	Top: High gamma value, Bottom: Low gamma value	74
4.18	Top: Good margin, Bottom: Bad Margin	75
5.1	Confusion Matrix	77
5.2	Hidden Layers with equal number of neurons	79
5.3	Accuracy results	80
5.4	Confusion Matrix	81
5.5	Converging Hidden Layer	81
5.6	Accuracy results	82
5.7	Confusion Matrix	83
5.8	Confusion Matrix	84

List of Tables

3.1	Specifications of DJI Phantom 4 Pro	27
3.2	Orthomap Details generated by Pix4d Mapper	29
3.3	Number of Overlapping Images	30
3.4	Types of LBP algorithms	51
3.5	Values of Gabor parameters in this research study	54
3.6	Feature Matrix size for each class	55
5.1	Parameter Details	79
5.2	Parameter Details	82
5.3	SVM Parameter Details	83
5.4	Accuracy comparison of individual features	84

Chapter 1

Introduction

1.1 Overview

Trees play a fundamental role in the ecology. Apart from providing renewable resources for human activities, they contribute to ecosystem, environment, economy and society. As a most important natural resource, trees need to be preserved for clear economical and environmental reasons. Efficient forest management is the answer to preservation of the forest and vegetation[5]. For better understanding of forest ecosystems and management it is very important to know the tree species composition of a forest. It would produce valuable information which would be of great help in estimation of the forest's economic value. In order to make efficient decisions, forest species composition is most sought pieces of information for forest preservation, planning and management, it is essential to obtain timely, accurate and periodic information regarding tree species[6]. Determining the tree species is an important topic in forest investigation, management, planning, inventory, environmental protection and statistics for relating forest resources[7],[8],[9] to mitigate problems like deforestation, response to global climate changes and extinction of certain species. Tree species classification is also an important component of vegetation resource mapping and wild life habitat mapping. Not only in forest but also in urban areas vegetation has very vital role in land surface temperature, urban climate, thermal environment, air quality and noise levels[10],[11],[12],[13],[14]. Detailed and up-to-date information regarding urban vegetation is required for long-term plan-

ning purposes.

Government and companies conduct the detailed classification of urban trees via ground surveys, manual photo interpretation, aerial photography, or remote sensing interpretation. However, conventional ground surveys and manual photo interpretation can be cost- and time intensive due to the urban scene complexity, landscape dynamics, and accessibility constraints for private areas. Although field surveys and manual visual interpretation has been a common practice for decades but it is not a viable option especially if species mapping and classification is to be done on a large areas and such information, e.g. vegetation mapping and specie distribution, was not available for most previous urban centers [15],[16],[17]. Remote sensing serves as great solution to the problems related to manually acquiring data. New possibilities and opportunities are offered by innovative methods, rapid evolution of technology and latest sensors to carry out remote sensing tasks. Advancement in image capturing and image processing technique has allowed image analysis at large scale and at fine level. Remote sensing is usually carried out using aircraft or satellite. Useful information can be extracted and employed for tree species related applications.

1.2 Problem Statement

Classification of tree species using spectral and spatial features with advanced machine learning algorithms.

1.3 Aims and Objectives

The main objectives of thesis are:-

- Collection of data-set using aerial optical imagery.
- Spectral and spatial feature extraction for species classification.
- Classify tree species using machine learning algorithms

Chapter 2

Literature Review

Tree detection and tree species classification requires remote sensing data with high spectral information and detailed information of the geometry in order to extract useful features for classification.

The types of data that are used in tree species classification:

- LiDAR
- Multispectral & Hyper-spectral
- Optical Imagery

2.1 LiDAR

LiDAR is an effective remote sensing method used to examine earth's surface. It is a safe and fast method for surveying. It measures the distance of the object by illuminating light pulses on the target object using pulsed laser light. Reflected light is sensed by the sensors. Sensors measure the time taken for the pulses to return. These pulses with other airborne data are used to generate the precise 3-Dimensional information of earth's surface. It is a very popular and most preferred technique in remote sensing to collect data for tree species classification because of its precise data collection and accuracy. Latest development & technological advancements has allowed LiDAR systems to capture data from 2500m away with accuracy 5mm

Apart from being precise and accurate, it can be used at any time of the day. It is not affected by the light variations and can be used at night time. LiDAR provides a much higher surface density. LiDAR system is equipped with sensors that are not affected by the geometric distortions such as angular landscapes. It makes complex data analysis easier as it can be integrated with other data sources. It is a versatile technology as it can operate in extreme weather conditions and has minimum human dependency as most processes are automated. Another great feature of LiDAR is that it can be used on the surfaces that are featureless and inaccessible.

LiDAR is the most popular tool in tree species classification because canopy height models (CHMs) can be obtained by using pulses from above tree crown and surface of the terrain. CHMs are used to acquire information regarding tree height, crown area and shape of an individual tree [18]. The approach proposed by Bin Wu *et al.* [19] derives CHMs from airborne LiDAR data and then contours on the basis of CHMs. Furthermore, Hierarchical structures of tree crowns are extracted followed by tree crown delineation accurately 94.21% and 75.07% in two forests. Lin Cao *et al.* [20] used small footprint full waveform LiDAR data and obtained accuracy of 68.6% for six, 75.8% for 4 species classes, and 86.2% for species classification between conifers and other broad leaf trees. First a digital terrain model (DTM) was created from point cloud data. For tree detection canopy height model (CHM) based algorithm was applied. Full waveform matrix were extracted from voxel based waveform approach which fed to Random Forest (RF) to get classification results. Daniele Marinelli *et al.* [21] exploited the fusion of multi-temporal LiDAR data. Their method was effective and produced confirmed results by first accurately characterizing crown shapes from high density data and later obtaining tree parameters from low density point data. Approach proposed by L.I. Duncanson *et al.* [22] uses watershed algorithm based delineation of CHM as several other developed crown delineation algorithms are good at extracting individual tree information from LiDAR point cloud data but have difficulty in discriminating the overlap crowns and also may fail to detect under-story trees. Algorithm used by them correctly identified 70%, 50%, 35% and 21% dominant, co-dominant, intermediate and

suppressed trees respectively. C.Vega *et al.*[23] introduced a framework known as PTrees. It is a point based segmentation approach for extraction of tree crowns from LiDAR point cloud data. Their research introduces new normalization approach for crown identification and characterization as they used absolute maxima rather than local maxima in order to locate the top peaks in point cloud data. Their framework was tested in 3 forest areas where they detected 82% of trees correctly and less than 10% false detection rate. Victor *et al.*[24] also proposed a segmentation algorithm to extract tree crowns using airborne LiDAR data and achieved 98.98%, 92.25% and 74.75% accuracy in three forest areas of 1 ha. Two Step approach was introduced by Antonio Ferraz *et al.*[25] which extracted individual tree crowns over tropical forest using LiDAR data. They used adaptive mean shift (ADM3D)[26] technique to properly extract individual tree.

Like every other technology LiDAR has limitations and cons. It is cost effective when dealing with the vast areas but has a high operational cost when used in applications where surveying is required on smaller area. It works great in almost all weather conditions but can be ineffective in heavy rainfall and low hanging clouds. Data quality is degraded in situations where sun angles are high and huge reflections as LiDAR works on the phenomenon of reflections. It works great on uniform surfaces but when used on water or on non-uniform surfaces it may not be reliable as depth of the water may affect the reflections. In addition it may not work as reliably on dense forest canopies and thick vegetation as pulses of LiDAR may not be able to penetrate. LiDAR collects huge amount of data points which help in creating precise and accurate 3-Dimensional view of the surface but analysis and interpretation is time consuming and increases the overall cost. There are no international protocols that guide the collection of LiDAR data. In instances where the beam is strong it can be harmful for human eye. LiDAR technology loses its accuracy and precision on altitude higher than 2500m because pulses will not be effective at this height.

2.2 Multispectral & Hyper-spectral

Images of earth and other planets obtained from satellites are known as satellite imagery. Among many uses and applications of satellite imagery, multispectral and hyper-spectral, its use in forestry has been in high demand. In terms of resolution, satellite images have three main types of resolution:

- **Spatial resolution:** is what size of surface area an image pixel represents.
- **Spectral resolution:** is defined by the interval size of the wavelength and the intervals that were measured by the sensors.
- **Temporal resolution:** is amount of time lapsed to capture same surface area.

Apart from LiDAR, multi-spectral and hyper-spectral data provided by satellites are prevalently used tool for tree classification[27][28] because of improved results in classification[29]. Unique opportunities for resource mapping at tree species level are brought by the arrival of latest satellites such as Sentinel-2, WorldView-3, WorldView-2, RapidEye and Pleiades[30]. Hyper-spectral data provide high spatial resolution combined with very fine spectral resolution which results in better tree species classification. Yuanyong Dian *et al.*[31] tested the effectiveness of combining spectral and spatial features for tree species classification. In order to reduce the dimensionality of the hyper-spectral image and highlight the variations minimum noise fraction(MNF) was used. to extract the features of the canopy GLCM was used and support vector machine SVM was used as classifier. Results of their experiment showed 85.92% of accuracy in classification of forest trees. Among 64 bands of AisaEAGLE-II Paras *et al.*[32] exploited 8-11 selected hyper-spectral bands on pixel level and plot level classification. Their research obtained accuracy and kappa of 93.50% 0.90 respectively. Their results show that, compared to pixel level, spatially aggregated plot level classification produce better results.

For identification of land use[33][34] and other features like soil[35], vegetation type[36] etc, hyper-spectral imagery provide better results due to high spectral res-

olution and large number of spectral bands. It is also very productive and effective in mapping diverse environments like estuaries[37] tropical rain forest[38] and marshlands[39]. Prem Chandra *et al.* [40] divided the hyper-spectral data to spectral ranges and performed principle component analysis (PCA). First three principle components of each spectral range were fused to make new data set. Their result shows that it contained 99.42% of the information of the original hyper-spectral data. Data set made from principle components was fed into maximum likelihood classification (MLC) and obtained the overall accuracy of 96.38% which is more than the accuracy of MLC applied on hyper-spectral data, 89.67%.SVM was used for classification of hyper-spectral data by Moses *et al.*[41]. Their method accurately mapped (overall accuracy = 89.3%) three dominant species found in Dukuduku forest in South Africa.For the development of the environmental support and decision systems tree species distribution and land use/land cover classes is important[42]. Akhtar Jamil and Bulent Bayram used an ensemble classifiers approach. They used classifiers different from different families to classify data into four classes. Output obtained from each classifier was then combined to get final results[42].

Jili Li *et al.*[6] classified three tree species using multispectral data. They developed local binary pattern(LBP) based index that proved to be very effective and over all accuracy of 77% was obtained. Classification was performed on pixel level and object level. But they classified the trees in generic categories i.e. High Value Species(HS), Medium Value Species(MS) and Low Value Species. Dawei Wen *et al.* [43] semantically classified urban tree species but into generic categories i.e. Roadside, Residential etc.

Weijia Li *et al.*[44] designed and implemented a deep convolution neural network based framework for tree detection using multispectral data. They achieved 92%-97% accuracy in the areas which is crowded and crowns overlap. Xiaojing Huang *et al.*[45] also detected trees using multispectral data with total accuray of 93%. They proposed a method multiresolution object based tree mapping method that includes

gaps, crown size and position of the trees.

However, the main disadvantage of hyper-spectral data is its cost and complexity. Large storage area is required to store hyper-spectral data. In addition fast computers are required to process the hyper-spectral data. Which if used for small area could be quite costly. Another hurdle faced by the researchers is programming of hyper-spectral satellite to only transmit important images, as storage and transmission of all data could prove to be difficult and expensive[46][47][48]. Multi-spectral data are prone to factors such as viewing angle, sun angle, day / year time (seasons). In addition to the above mentioned disadvantages of satellite imagery, trade off between spectral resolution and SNR is required as high spectral resolution reduces SNR[49].

2.3 Optical Imagery

Optical images are acquired using the optical part of the spectrum which encompasses the visible, infrared and ultraviolet bands of the electromagnetic spectrum. Optical images can be displayed in a way that is visible to human eye, such as natural colors. They can also be represented using false color, as in infrared. Leaves are the main objects that absorb light, in the visible part of the spectrum. They are able to do so because of the presence of foliar photosynthetic chlorophyll a and b and carotenoids[50].

Generally optical remote sensing imagery has been used for mapping vegetation, land-cover and land-use in many areas for past several decades. The first attempt to use remote sensing data for classification purposes was applied on natural forests using satellite images (such as Landsat TM and later ETM+)[51]. Hongyu Huang *et al.*[52] used optical imagery using un-manned aerial vehicle (UAV) for individual tree crown delineation (ITDC). Optical images are often combined with other data sources e.g. laser point cloud to include spectral information[53],[54]. Wen Zang *et al.*[55] used optical imagery with Lidar for automatic forest species classification. Their study had two stages. First segmentation of the trees using novel algorithm and second stage was tree species classification. Clark *et al.* used optical imagery with hyper-spectral data to achieve 86% of over-all accuracy.

Techniques that are traditionally used in optical remote sensing suffer from a lack of the ability to capture three-dimensional forest structural details, particularly in areas where trees are unevenly-aged, forests with multiple species canopy layers[56].

Among many significant advantages that optical remote sensing has over SAR remote sensing is that many optical remote sensing data sets with medium spatial resolution are available online for free, which reduces the cost of analysis considerably. At any given time, optical time series have better temporal resolution than SAR, when they are not obscured by clouds[57] as there are more optical remote

sensing systems. Additionally, optical imagery also is not affected by speckle and is quite efficient for land cover classification such as distinguishing among urban areas, vegetative surfaces, and water[58]

Advancement in technology and technical development has led to an alternate to above mentioned data sources for remote sensing. 'Unmanned Aerial Vehicles' (UAVs) has made the acquisition of image data efficient and economical. UAV also known as drone and 'Low Altitude Remote Sensing (LARS)[59] is an important development because of its low cost and practicality. Apart from providing its amazing maneuverability, UAV systems also provides high resolution colored images acquired by low cost cameras that can be integrated with the UAVs. These images can be processed as per desired purposes[60]. Optical imagery using UAVs can serve as an alternate to all the disadvantages and cons of other data sources as UAVs are portable weigh less and can be operated with low cost cameras to get images. Hung *et al.*[61] used RGB imagery, vision band only, for tree crown detection. Geometry, appearance and color segmentation was used for template matching. Malek *et al.*[62] also used vision spectrum to detect palm trees. Ramesh Kestur *et al.*[63] proposed a neural network based approach for tree crown detection, delineation, and counting using high resolution RGB images for spectral-spatial classification. Some areas that were not trees were classified as trees due to spectral intensity similarities. In order to overcome this problem spatial feature classification was done. Results were compared with K-Means spectral-spatial clustering. Object based analysis of RGB optical images was used to detect trees by Serdar Selim *et al.*[64].

Very less work has been done on the classification, with over head view, using low cost commercial drones with optical sensors.

2.4 Combination of data sources

Use of individual data sources i.e. LiDAR, ALS, Satellite Imagery and to some extent Optical Imagery has been investigated in many studies to improve the tree species classification, forest inventory, forest management and other applications. To make reliable, proper and effective decision, a single data source might not be adequate enough. For example, hyper-spectral optical data cannot provide the 3-dimensional data required like plant height and structure of the canopy. LiDAR data can provide the 3-D information required as mentioned but it will not work in the classification and differentiation of the tree species with the same height but this problem is solved by the hyper-spectral data. But once the complementary information of the two sources are combined, new information provides better and comprehensive interpretation for tree species classification.

Koetz *et al.* [65] fused LiDAR and HS bands using support vector machines for classification of fuel composition. Their results reported better classification performances in comparison to the results of the individual sensors. Dalponte *et al.* [66] -combined remote sensing data, LiDAR and Hyper-spectral data, for the classification of complex areas of the forest. In order to integrate acquired multi-sensor information, appropriately, multiple classifiers were used. [67] presented a model based on Random Forest for the fusion, automatically, Hyper-spectral and LiDAR data. This model was used to classify eight tree species of African savanna. Their experiment showed that once some features such as height, extracted from one data source, are combined with the complementary attributes like spectral information, extracted from the other data source, can considerably improve the performance of the classification model. Reference [68] proposed and presented model based on kernel learning that could manage fusion of heterogeneous features that were acquired from Hyper-spectral and LiDAR data sources. Gaussian kernels were used to model the similarity of feature between two different data sources. A framework was proposed by Yokoya *et al.* [69] to investigate the visual quality of landscape. The framework compared the physical features mastered from inte-

grated use of data sources, Hyper-spectral and LiDAR, with the expertise based on human understanding. The approach utilized in [70] fuses together spectral features, in this case, first few principal components extracted from the pre-processed hyper-spectral data with sizes and shapes of individual trees derived from the LiDAR data after individual tree-crown were delineated. Coillie *et al.* [71] combined PCA features in augmentation from two sources, Hyper-Spectral and LiDAR, by transferring the both sources to PCA domain. Optimal feature set was selected on the basis accuracy, for classification. Buddenbaum *et al.* [72] also fused images from Hyper-spectral and LiDAR data sources for classification of tree species on the basis of their age. Using image segmentation techniques, using LiDAR data as source, geometric information was derived [73]. This information was integrated with the spectral information extracted from hyper-spectral data to classify urban areas. Usually multi-sensor data fusion is to stack multiple feature sources together and use these as the input of a classifier, a simple and directional method. This is commonly used in remote sensing. Wenzhi Liao *et al.* [74] proposed framework for deep fusion. It integrates the complementary information derived from hyper-spectral and LiDAR data to map tree species. In addition to the above, the fusion of 'single-band' or 'multi-band' LiDAR with Hyper-spectral data, to map tree species, has been investigated. A solution was proposed for estimating crown size of tree species by the integration of multi-sensor data. Results of their experiments on fusing real APEX hyper-spectral and LiDAR data proved to be effective. Proposed method gives improvements in comparison to use of single data source or current deep fusion architecture. Classification accuracies ranging from 82.21% to 87.10% and 76.71% to 83.45%, respectively, were achieved. Luxia *et al.* [75] used the fusion of hyper-spectral data and remote sensing data, LiDAR, for mapping urban tree species. They evaluated the potential of the technologies using random forest classifier. Crown structural information, derived from LiDAR data, was combined with spectral indices (vegetation indices) and achieved overall accuracy of 51.1%, 61.0%, and 70.0% using hyper-spectral, LiDAR and the combined data respectively.

Matheus Pinheiro Ferreira *et al.* [76] discriminated and mapped tree species in tropical seasonal semi-deciduous by using airborne hyper-spectral and simulated multispectral data. Crown eight tree species were quantified on the basis of spectral variability and three supervised machine learning classifiers were applied to discriminate the species at the pixel level. For the classification of tree species, Holmgren *et al.* [54] exploited multispectral imagery fusion and LiDAR data.

Combining two resources may be fruitful in terms of increase in the classification results but the computational complexity would also increase. In addition, so will the cost of acquiring and processing of the data especially if used in smaller areas.

Chapter 3

Methodology

The proposed methodology for tree species classification comprises of four main blocks as shown in the Figure 3.1. The first step towards the objectives is acquisition of data which involves extensive UAV flight operations on target study area. The acquired data is then pre-processed to create an orthomap. Features were extracted from the data and were fed to the classifiers for the final classification

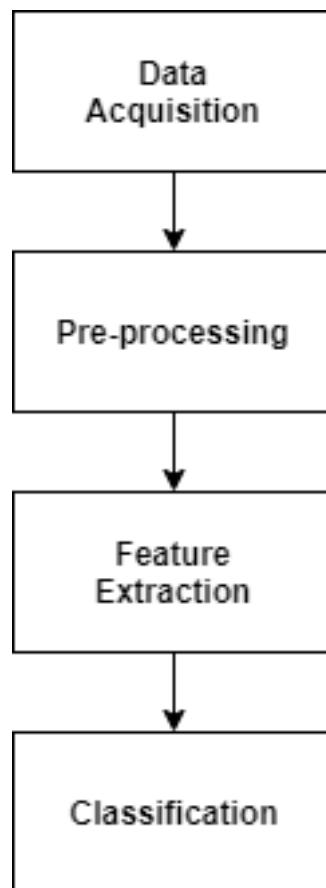


Figure 3.1: Method adopted for Tree Species Classification

3.1 Study Area and Data Set

Data set utilized in this study comes from the Kalowal area in the province of Punjab, Pakistan. Its coordinates are 31.85°N and 72.97°E . Prevailing climate of this area is known as local steppe climate. Temperature here averages 23.9° . The average rainfall in this area is 403mm. June is the warmest month and the average temperature in Kalowal is 34.1° . It is an agricultural area and its main vegetation are sugarcane, wheat, corn and seasonal fruits (Guava and citrus fruits).

The area was surveyed and selected on 9th February 2019. Area chosen was a privately owned land. 3 sites were chosen. They were orchards of guava, Fruiter & Musammi(types of oranges). Among total plantations of fruits, Guava is percentage of Guava and Citrus (Musammi & Fruiter) is 30-35 and 65-70 respectively.

Total 15 acres of study area was used of which around 3 acres were covered with guava trees, Musammi trees occupied 3.5 acres of the area and around 6 acres was covered by Fruiter trees. The area is shown in the Figure 3.2 below.

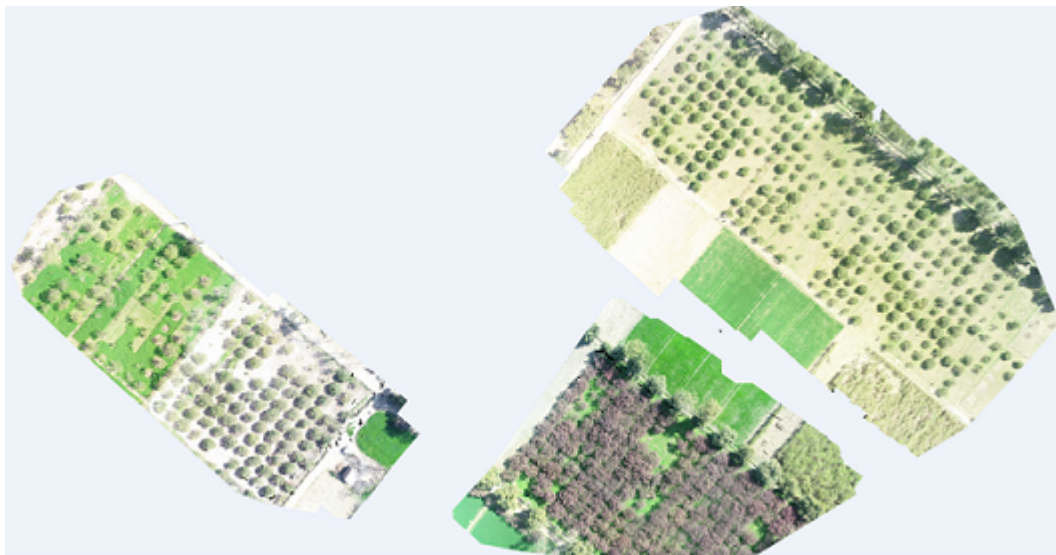


Figure 3.2: Orthomosaic of the orchards. Musammi(left), Guava(bottom), Fruiter(right)

Optical Imagery was captured on 16th of February 2019 at 11 AM using UAV drone. To capture the optical imagery (only RGB) DJI Phantom 4 Pro was utilized shown in the Figure 3.3 below.



Figure 3.3: DJI Phantom 4 Pro

Its is like an entry level DSLR for sky that is designed for cinematography and photography to provide live HD view. It is one of the most intelligent and easy to operate drone available. It is ready to be sent up in the sky , out of the box.

This drone has a 4k (4096 x 2160) camera & ultrasonic sensor that is required for calculating range and distances of the objects from the height of the drone. It has a 3 axis gimbal which is very active in making photos and videos. The remote operates on high frequency using 2.4 GHz which can provides control for up to 7km of range and about 33 feet height. Table 3.1 shows specifications of DJI Phantom 4 Pro.

Parameter	Detail
Max Ascent Speed	S-mode: 6 m/s, P-mode: 5 m/s
Max Descent Speed	S-mode: 4 m/s, P-mode: 3 m/s
Max Flying Speed	modes (S,A,P): 45 mph, 36mph, 31mph
Operating Temperature	32°to 104°F (0°to 40°C)
Sensing System	Infrared
Positioning System	GPS/GLONASS
Operating Frequency	2.400 - 2.483 GHz and 5.725 - 5.825 GHz
Altitude Range	0 - 33 feet (0 - 10 m)
Camera Sensor	1" CMOS, Effective pixels: 20M
Modes	Still photography and Video Recording
Picture Format	JPEG, DNG (RAW), JPEG + DNG
Video Format	MP4/MOV (AVC/H.264; HEVC/H.265)

Table 3.1: Specifications of DJI Phantom 4 Pro

It comes with the inbuilt capability of obstacle detection and collision avoidance during the flight. If it detects an object in its path, either it will stop or automatically change the course around the object. For object detection and collision avoidance, this machine uses front, backward, and rear looking optical sensors to detect objects to 50 feet in front of and 30 feet beneath.

DJI go 4 app was used to fly the drone using android mobile by mounting the mobile device on the remote controller of the DJI drone. It is the base app for any DJI drones like Mavic, Mavic pro and Phantom. It controls the drone as well as takes the imagery and video. In addition to this it also provides calibration options for the gimbal.

In order to generate orthomaps of the desired orchards, 'Pix4d Mapper' software was used. It is a photogrammetry software, used by professionals, that uses images to generate textured models, digital surfaces & terrain models, point clouds, orthomosaics etc shown in the Figure 3.4.

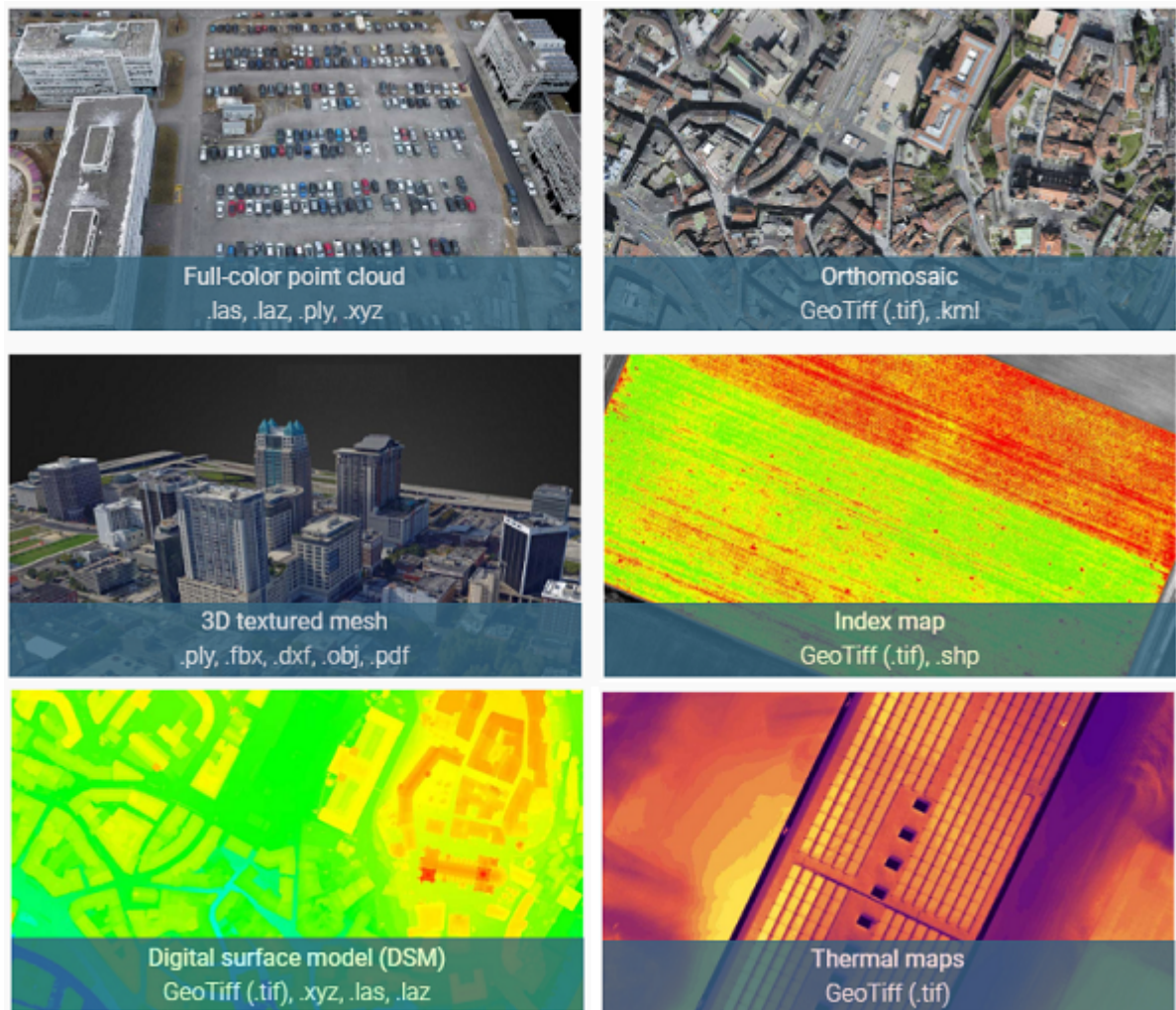


Figure 3.4: Types of outputs by Pix4D Mapper

In this research, orthomosaic option was utilized to get the desired orthomosaic of the study area as shown Figure 3.2. The mobile application of this software was connected with the drone for automated flights of drone and taking automated imagery of the orchids by giving it the proper constraints and location. You have to give the area to be covered like 50x30 and the height like 50 meters and then you would randomly select the lines of the drone flight by adjusting them with your fingers according to your requirement. After the images were taken they were fed to pix4d mapper windows software for the stitching and creation of orthomosaic as shown in the Figure 3.2. This app not only stitches the pictures but also provides some pre-processing like noise removal, image sharpening and geo-referencing. In addition to the above it generates detailed report for the project. Some details have been shown below in Table 3.2

Parameter	Detail
Project Name	Orchards
Date	S-mode: 2019-02-16
Camera/Imagery Type	FC6310'8.8'5472x3078 (RGB)
GSD	1.41 cm / 0.55 in
Area Covered	0.041 km ² / 4.0800 ha / 0.02 sq. mi. / 10.0872 acres
No. of Images	78
No. of calibrated Images	78
No. of Geolocated Images	78

Table 3.2: Orthomap Details generated by Pix4d Mapper

Images were captured at 60m altitude with 1.41cm ground sampling distance (GSD). Total 78 images of the area were captured and stitched together with 80% overlap for the creation of ortho-map. Overlap information is shown in the Figure 3.5, generated by the software Pix4D Mapper.

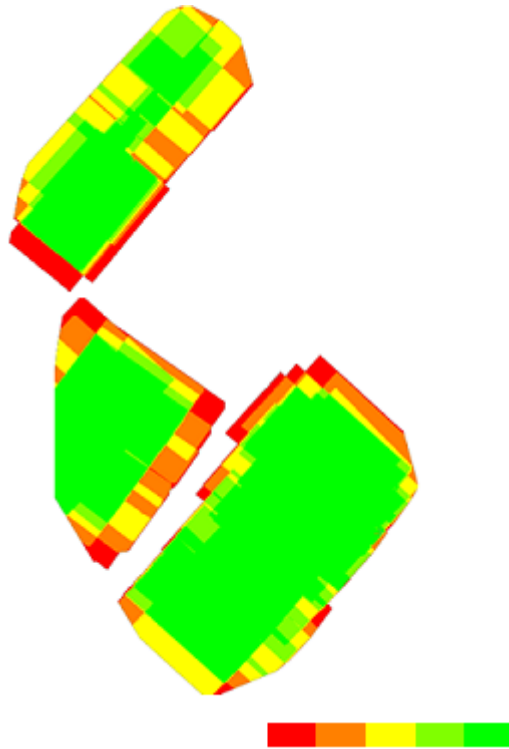


Figure 3.5: Number of overlapping images computed for each pixel of the orthomosaic

Color	No. of Images
Red	1
Orange	2
Yello	3
Light Green	4
Green	5+

Table 3.3: Number of Overlapping Images

Red and yellow areas indicate low overlap for which poor results may be generated. Green areas indicate an overlap of over 5 images for every pixel as shown in Table 3.3. Good quality results will be generated as long as the number of keypoint matches is also sufficient for these areas. Small blue dots shown in the Figure 3.6 below are pin locations at which picture were taken.

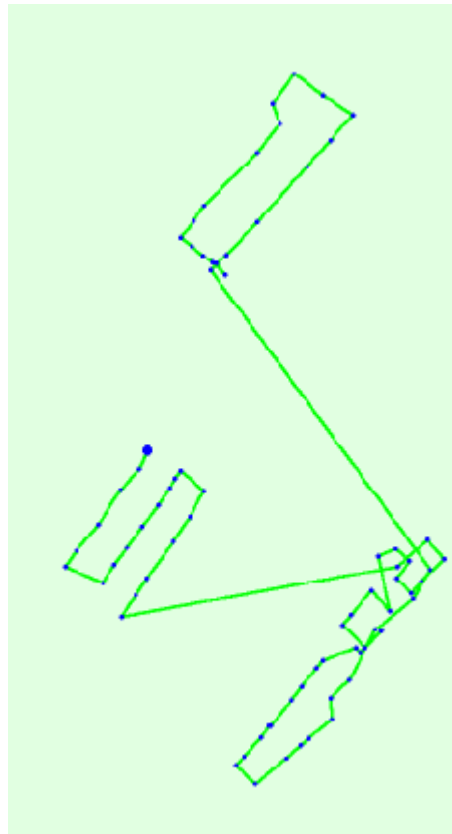


Figure 3.6: Drone trajectory to capture pictures at different position starting at big blue dot

3.2 Pre-Processing

Orthomap cannot be used as an input for extraction of required features as orthomaps are an aerial view of the area. It contains images/information of areas that are not of interest like grass, soil, roads etc. Area of interest is only trees. Only the images of trees are required.

There are two ways to detect trees out of a picture to be fed for feature extraction: 1) Manually or 2) Automatically. Tree detection actually itself is an area of research which is out of the scope of this research study. This study solely focuses on the classification of tree species. Therefore, trees were extracted manually by using GIMP software. GIMP is a free cross-platform image/graphic editor. The best feature of this software is that we can crop out the area of interest from center to outwards in multiple shapes. All you need is to define the size of the area that needs to be cropped and multiple scale levels are available. For example, area can be cropped in inches, percentage, pixels, points, picas, centimeter, millimeter and many more. This study uses rectangular shaped 200 x 200 in pixel scale, as our minimum grain level for feature extraction is pixel. All that is needed to be done is that you have to pinpoint to the center of the tree canopy, a rectangular block will be formed and ready to be cropped. Using the above mentioned procedure, 533 tree images were extracted in total as shown in the Figure 3.7 below. In order to carry out proper classification all the images should be standardized. So, all 533 images are of the same size.



Figure 3.7: Tree images extracted from orthomap

It may seem that Figure 3.7 does not show complete canopy of the tree but it contains all the required matter for feature extraction. As there were 3 types of trees, their respective images were stored in separate folders.

3.3 Feature Extraction

The aim for this research is to use spectral and spatial feature for tree species classification using optical imagery using RGB channels.

Spectral features uses the value of single pixel while spatial features uses the values of neighboring pixels into consideration as well.

3.3.1 Colour Images

A RGB image also known as colour image is made from the amalgamation of some basic colours. Every single pixel of an image can be broken down into red, green and blue values. If we break an image ($M \times N$) into 3 RGB channels, as shown in Figure 3.8 below, we would get 3 matrices of dimensions $M \times N$ arranged sequentially. [77]

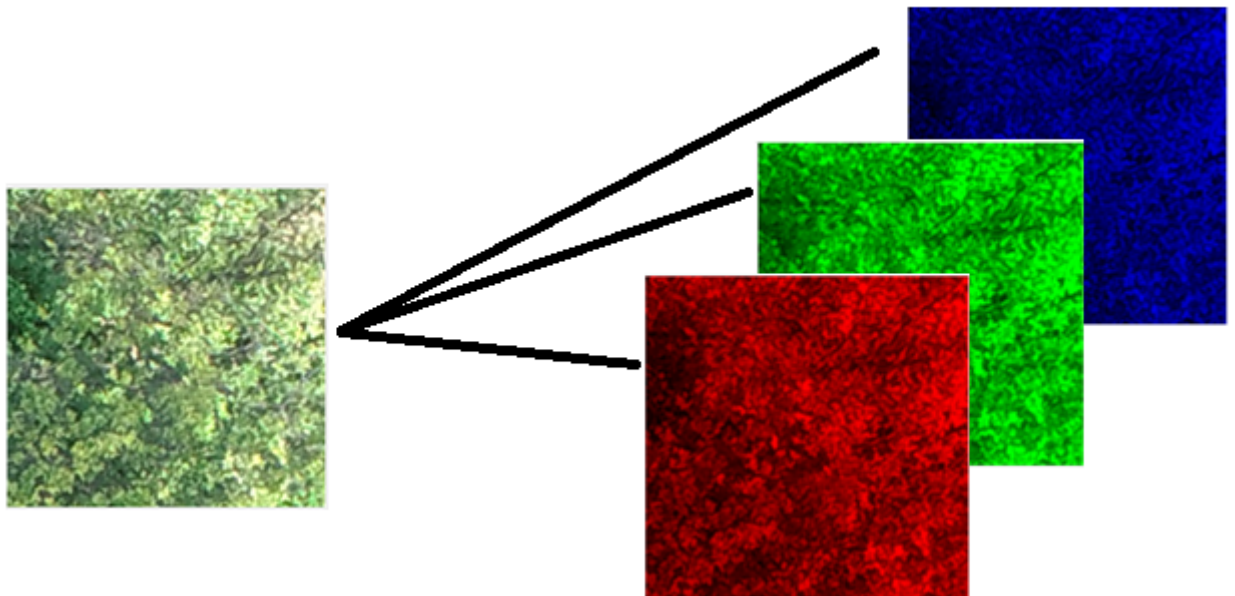


Figure 3.8: RGB Image decomposed into 3 channels

The three images formed from each channel can be used as an input and other features like mean, SD, variance, Entropy etc. can be calculated from it.

Although RGB colour space is most widely used. But it is not accurate in accordance to human visual perception and statistical analysis. So, HSV colour space is used for to get accurate colour statistics for features extraction. HSV is a non-linear transform of RGB colour space that it describes colour to perception relationship more accurately than RGB. Figure 3.9 shows HSV model. HSV separates colour into hue, saturation and value which means it can define colour into colour, intensity and brightness. Hue represents the property of the colour like blue, yellow, red etc. Saturation represents the intensity perceived and ‘Value’ represents brightness perceived. [78]

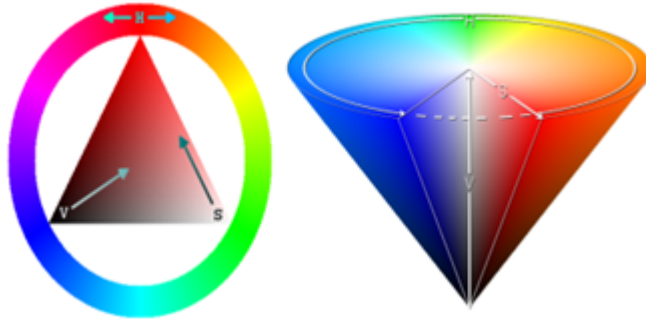


Figure 3.9: HSV color model

RGB colour space can be converted to HSV colour space by following formulas:

$$h = \begin{cases} 0, & \text{if max=min} \\ (60^\circ \times \frac{g-b}{max-min} + 360^\circ) \bmod 360^\circ, & \text{if max=r} \\ (60^\circ \times \frac{b-r}{max-min} + 120^\circ), & \text{if max=g} \\ (60^\circ \times \frac{r-g}{max-min} + 240^\circ), & \text{if max=b} \end{cases} \quad (3.1)$$

$$s = \begin{cases} 0, & \text{if } \max=0 \\ \frac{\max - \min}{\max} = 1 - \frac{\min}{\max}, & \text{otherwise} \end{cases} \quad (3.2)$$

$$v = \max, \quad (3.3)$$

RGB values used in above formulas are normalized from 0-1. Many statistical feature such as standard deviation etc are calculated for all 6 channels and used as a feature for classification. Explanation of features used in this study are as follows.

3.3.2 DCT

Discrete cosine transform (DCT) is a Fourier related transform which was proposed by Ahmed *et al.*[79] in 1974. It is similar to Discrete Fourier Transform (DFT). It represents an image in terms of sum of cosine functions but only using real numbers[80]. It is almost twice the length equivalent to DFT. Several variants of DCT have been proposed[81]. DCT has been widely used in face recognition and compression algorithms. It is used for extracting texture features of an image[80]. It does not reduce the data dimension but stores most of the information in few numbers of coefficients in compressed form[3]. DCT coefficients of M x N image can be calculated by the following equations:

$$F(u, v) = \frac{1}{\sqrt{MN}} \alpha(u) \alpha(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y) \cos \frac{(2x+1)u\pi}{2M} \cos \frac{(2y+1)v\pi}{2M} \quad (3.4)$$

where $\alpha(\omega)$ is defined by:

$$\alpha(\omega) = \begin{cases} \frac{1}{\sqrt{2}} & \omega = 0, \\ 1 & \text{otherwise} \end{cases} \quad (3.5)$$

$F(u, v)$ is matrix (2D) of coefficients, in frequency domain, and $I(x, y)$ are intensity value at respective position in an image matrix (spatial domain).

DCT converts image from spatial domain to frequency domain. DCT coefficients can be divided into three sub-bands: Low, Medium & High frequency. As shown in Figure 3.10 below

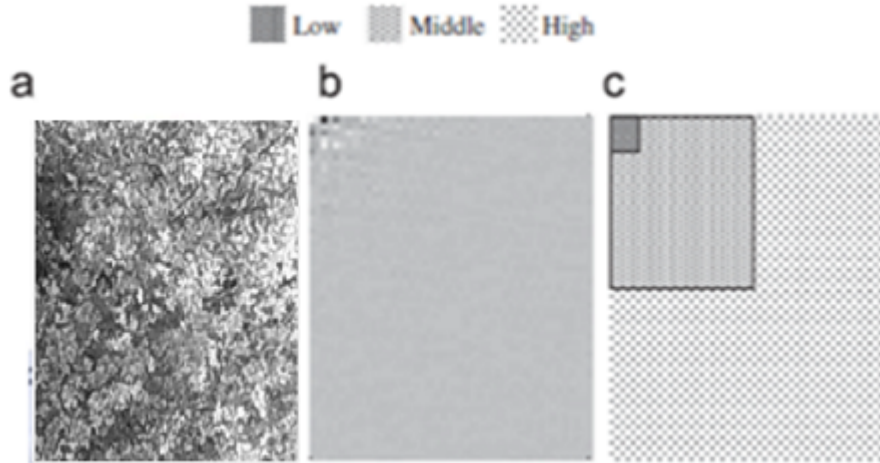


Figure 3.10: (a) Original Picture (b) Frequency Domain (c) DCT Coefficients Low, Medium and High

Low frequencies correlate with illumination while high frequencies show noise and small details. Middle frequencies actually are those which contain the useful information regarding the image structure.

The inverse of the DCT can be calculated by the following equation:

$$u(n) = \sum_{k=0}^{N-1} \alpha(k)v(k) \cos \frac{(2n+1)k\pi}{2N} \quad (3.6)$$

3.3.3 GLCM

Gray Level Co-occurrence Matrix, in short GLCM, is a method used to extract textural features from statistical distribution of intensities in an image at specified positions relative to each other for statistical texture analysis. It is also known as Gray Level Dependency Matrix and first introduced by Harlick *et al.*[4] It is an image analysis technique. It is a tabulation of how different combinations gray level intensities occur in an image. In texture analysis order of the feature means number pixel used to check combination. GLCM is a method to extract second order statistical texture features. As shown in the Figure 3.11, GLCM is showing

1	1	5	6	8
2	3	5	7	1
4	5	7	1	2
8	5	1	2	5

	1	2	3	4	5	6	7	8
1	1	2	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

Figure 3.11: Original Image(Left) & its GLCM(Right)

the occurrence of combinations adjacently on the right of the pixel of interest (i.e. ‘11’ has occurred once and ‘12’ has occurred twice). Pixel can be chosen at different angles and at different offset too as shown in the Figure 3.12 . It is not necessary that pixels have to be adjacent.

	135° [-D-D]				90° [-D1]				45° [-D 0]			
	1	1	2	0	0	1	0	0	0	0	0	0
	0	0	0	1	0	1	0	0	0	0	0	0
	0	0	0	0	0	1	0	0	0	0	0	0
	0	0	0	0	0	1	0	0	0	0	0	0
	1	1	0	0	0	0	1	2	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	
	2	2	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	1	0	0	0	0	0	

↑
0° [0 0]

Figure 3.12: Different angles at which GLCM can be calculated

Properties of GLCM:

- Matrix is square in shape.
- Number of rows and columns are equal to the quantization levels of an image.
- It is symmetrical about diagonal.

Work of Harlick *et al.*[4] and Connors *et al.*[82] show that number of texture features can be extracted from GLCM. Few are discussed below.

To explain various textural features, following notations are used.

$$g_{ij} = (i, j)^{th} \text{ entry in GLCM}$$

$$g_x(ij) = i_{th} \text{ marginal probability matrix entry, obtained by summing rows of } g_{ij} = \sum_{j=1}^{N_g} g(i,j)$$

N_g = Number of gray level distinct levels in image

$$\sum_i = \sum_{i=1}^{N_g}$$

$$\sum_j = \sum_{j=1}^{N_g}$$

$$g_y(i) = \sum_{i=1}^{N_g} g(i, j)$$

$$g_{x+y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} g(i, j) \quad \text{where } i+j=k=2,3,\dots,2N_g$$

$$g_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} g(i, j) \quad \text{where } i-j=k=0,1,\dots,N_g - 1$$

- **Energy**

$$\text{Energy} = \sum_i \sum_j (g_{ij})^2 \quad (3.7)$$

This feature is also known by the name of uniformity or angular second moment. It measures pixel pair repetitions. Its basic functionality is to detect disorders in texture.[4][83]

- **Entropy**

$$\text{Entropy} = - \sum_i \sum_j g_{ij} \log_2(g_{ij}) \quad (3.8)$$

It measures complexity of an image. More non-uniformity the image has, in terms of texture, more entropy it has. Entropy is inversely but strongly correlated to energy.[4][83]

- **Contrast**

$$\text{Contrast} = \sum_i \sum_j (i - j)^2 g_{ij} \quad (3.9)$$

Contrast measure the spatial frequency of an image. It is the actually difference between maximum and minimum values or local variation of neighboring set of pixels. [83][84]

- **Variance**

$$\text{Variance} = \sum_i \sum_j (i - \mu)^2 g_{ij} \quad (3.10)$$

Variance, statistical measure associated with GLCM, is a measure of heterogeneity and is correlated strongly with standard deviation (first order statistical measure). [4] [83][84]

- **Inverse Difference Moment(IDM)**

$$\text{IDM} = \sum_i \sum_j \frac{1}{1 + (i - j)^2} g_{ij} \quad (3.11)$$

It is also known as homogeneity. When all the elements in an image have same value then the homogeneity value is maximum. If energy is kept constant, contrast and IDM are strongly but inversely correlated.[83]

- **Correlation**

$$\text{Correlation} = \frac{\sum_i \sum_j (ig)g_{ij} - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (3.12)$$

where μ_x , μ_y , σ_x and σ_y are the means and standard deviation of g_x and g_y

It is a measure of the gray tone linear dependencies of pixels at positions specific positions relative to each other.[83][84]

- **Sum Average**

$$\text{Sum Average} = \sum_{i=2}^{2N_g} ig_{x+y}(i) \quad (3.13)$$

- **Sum Entropy**

$$\text{Sum Entropy} = - \sum_{i=2}^{2N_g} g_{x+y}(i) \log g_{x+y}(i) \quad (3.14)$$

- **Sum Variance**

$$\text{Sum Variance} = - \sum_{i=2}^{2N_g} (i - sa)^2 g_{x+y}(i) \quad (3.15)$$

- **Difference Variance**

$$\text{Difference Variance} = \text{vairance of } g_{x-y} \quad (3.16)$$

- **Difference Entropy**

$$\text{Difference Entropy} = - \sum_{i=0}^{N_g-1} g_{x-y}(i) \log g_{x-y}(i) \quad (3.17)$$

- **Maximum Correlation Coefficient**

$$\text{MCC} = (\text{Second largest eigen value of } Q)^{0.5} \quad (3.18)$$

$$\text{where } Q(i, j) = \sum_k \frac{g(i, k)g(j, k)}{g_x(i)g_y(k)} \quad (3.19)$$

- **Information Measures of Correlation**

$$\text{IMC1} = \frac{HXY - HXY1}{\max(HX, HY)} \quad (3.20)$$

$$\text{IMC2} = \sqrt{1 - \exp(-2.0(HXY2 - HXY))} \quad (3.21)$$

$$HXY = - \sum_i \sum_j g_{ij} \log_2 g_{ij} \quad (3.22)$$

where HX and HY are the entropies of g_i and g_j

$$HXY1 = - \sum_i \sum_j g_{ij} \log_2(g_x(i)g_y(j)) \quad (3.23)$$

$$HXY2 = - \sum_i \sum_j g_x(i)g_y(j) \log_2 g_x(i)g_y(j) \quad (3.24)$$

3.3.4 Intensity

There are several basic features (statistical or spectral) that can be calculated by intensities of an image. [85]

- **Intensity Mean**

$$\text{Intensity Mean} = \frac{\sum x}{N} \quad (3.25)$$

It simple an average of all the intensity values of an image.

- **Intensity Median**

It is the middle value if all the intensity values of an image are arranged in ascending order.

Median = centre value when n is a odd number.

Median = two central values when n is a even number.

- **Intensity Mode**

Most frequently occurring intensity value is its mode.

- **Intensity Lit Mean**

The mean intensity of set of pixels whose intensity value is greater than the mean intensity of all pixels.

- **Max/Top Intensity**

The maximum value of pixel intensity.

- **Min Intensity**

The minimum pixel intensity.

- **Intensity Standard Deviation**

$$\text{ISD} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (3.26)$$

x_i means pixel under consideration, μ represents mean of the pixel values. N represents total number of pixels.

- **Intensity Entropy**

Entropy of pixel intensities means average information or degree of randomness in an image.

$$\text{IE} = - \sum_{i,j} p_{ij} \log_2(p_{ij}) \quad (3.27)$$

P_{ij} represents probability of occurrence of intensity value under consideration.

- **Intensity Skewness**

Skewness also known as standardized third moment of intensity distribution.

It basically represents imbalance distribution from the mean value.

$$\text{IS} = \frac{m_3}{m_2^{\frac{3}{2}}} \quad (3.28)$$

$$\text{IS} = \frac{m_3}{\sigma^3}$$

$$m_3 = \frac{\sum_{i=1}^N (x_i - x_{avg})^3}{N}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{N}}$$

$$\text{IS} = \frac{1}{N} \frac{\sum_{i=1}^N (x_i - \mu)k^3}{\sigma^3}$$

$$\text{IS} = \frac{1}{N} \frac{\sum_{i=1}^N (x_i - \mu)^3}{\frac{(\sum_{i=1}^N (x_i - \mu)^2)^{\frac{3}{2}}}{N^{\frac{3}{2}}}}$$

$$\text{IS} = \sqrt{N} \frac{\sum_{i=1}^N (x_i - \mu)^3}{(\sum_{i=1}^N (x_i - \mu)^2)^{\frac{3}{2}}}$$

m_3 represents third moment and σ is standard deviation. If the tail of the distribution curve is towards left then it is call negatively skewed and if tail lies at the positive side or right side of the mean value than then it is known as positively skewed. Furthermore, if value of skewness is 0, it means data is evenly distributed[86] as shown in Figure 3.13 below.

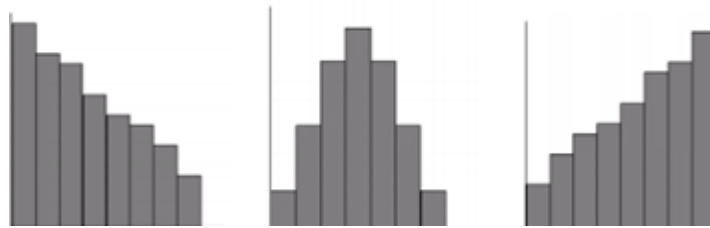


Figure 3.13: Positively skewed, Normally Distributed and negatively skewed Histograms

- **Intensity Kurtosis**

$$\text{IK} = \frac{\sum (x_i - \mu)^4}{N\sigma^4} \quad (3.29)$$

Kurtosis characterizes the flatness or peakedness of a distribution in relation with normal distribution. It is also known as the fourth moment. If the kurtosis value is less than 0, platykurtic, it means that the frequencies throughout the curve are closer to be equal (i.e., the curve is more flat and wide) relatively flat distribution as shown in Figure 3.14 and vice versa for leptokurtic.[86]

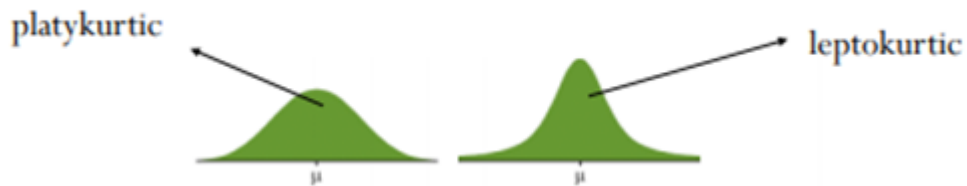


Figure 3.14: Platykurtic & Leptokurtic

3.3.5 LBP

LBP stands for ‘Local Binary Patterns’. They describe texture and shape of a digital image. The concept of LBP was introduced in 1993 but it became popular in the 1996 by the work of Ojala et al. [87]. Local representation of texture in an image is computed by LBP by comparing each pixel with its surrounding pixels. LBP works on grayscale image. So, image of interest is to be converted in grayscale. Divide the image into grid of cells (usually 16 x 16 grids) as shown in the Figure 3.15 below.

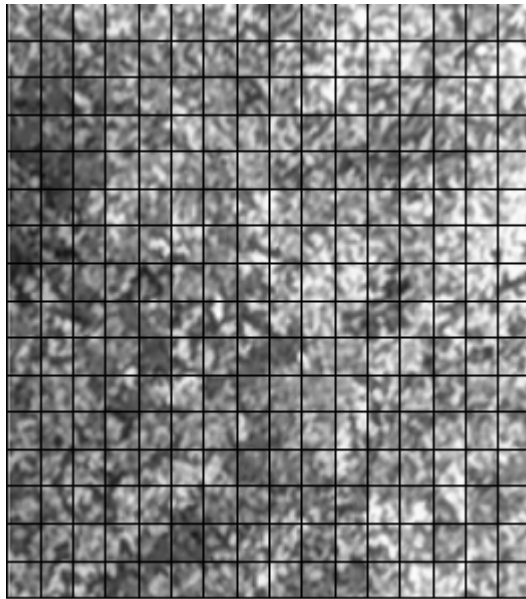


Figure 3.15: Grayscale image divided into cells

Each box represents a cell and each cell contains equal number of pixels. For each cell of grayscale image we chose a neighbourhood of size r that surrounds the center pixel. The original version of LBP uses 3 x 3 kernel window to compute LBP of a central pixel.



Figure 3.16: Thresholding to make binary pattern

Let us take an example of 3 x 3 window, used to calculate the LBP of a pixel in a cell of a grayscale image shown in Figure 3.16. Central pixel, in red, is compared with its 8 neighbouring pixel and threshold. If value of neighbouring pixel is greater or equal than the central value the, allocate '0' to that pixel. If neighbouring pixel's intensity value is less than the central pixel then allocate '1' to that pixel. Allocation of '0' and '1' can be used the other way around as well. To calculate LBP value for the central pixel, we can start from any neighbouring pixel and work our way anti-clock or clockwise. But make sure that pattern, clock or anti-clock, is consistent for all the pixels in the image. Starting from top right would make the binary code 11101000 if we move clockwise. This binary sequence is to be converted to decimal as shown in the Figure 3.17 below.



Figure 3.17: Converting binary code to decimal value

Decimal value for the combination turns out to be 23. Value 23 will be stored in central pixel of interest for which we have calculated the LBP. With 8 surrounding pixels total 256 (2^8) combinations of LBPs can be created. This process is to be created for each pixel in the cell. Last step is to calculate the histogram of the cell which shows the frequency of occurrence of decimal values stored in the cell. Histogram actually shows how many times a LBP has occurred. Histograms of all the cells are then concatenated to produce the final image. Histogram can be treated as a feature vector.

The primary benefit of the aforementioned basic LBP is that we can capture fine grained details. LBP can be represented by the following equation:

$$LBP(x_c, y_c) = \sum_{n=0}^7 2^n g(I_n - I(x_c, y_c)) \quad (3.30)$$

$I(n)$ is intensity value of neighbouring pixel and $I(x_c, y_c)$ is the value of central pixel.

Drawback of LBP is that small scale textures that may have negligible or no contribution at all are also calculated. In addition the basic LBP comes with the following drawbacks: long histograms, sensitive to image rotation, small spatial area of support (we cannot capture details at varying scales but only for fixed 3 x 3 windows), a loss of local textural information, and sensitive to noise. Researchers have made efforts, listed in the Table 3.4, to overcome the problems related to the conventional LBP.

In this research study conventional LBP with 3 x 3 kernel has been used.

Variations	Properties	Ref #
Enhancing the Discriminative Ability		
Mean LBP	Consider the effects of central pixel; present complete structure pattern	[88, 89, 90]
Hamming LBP	Incorporate non-uniform patterns into uniform patterns	[91]
Extended LBP	Discriminate the same local binary patter; cause high dimensionality	[92, 93]
Completed LBP	Include both sign and magnitude information of the local region	[94]
Improving Robustness		
Local Ternary pattern	Bring in new threshold; no longer strictly invariant	[95]
Soft LBP	Not invariant to monotonic grayscale changes; cause high computational complexity	[96]
Choosing the neighborhood		
Elongated LBP	Extract anisotropic information and lose anisotropic information; not invariant to rotation	[97]
Multi-Block LBP	Capture micro and macro -structure information	[98, 99]
Three/Four Patch LBP	Encode patch type of information	[100]
Extending to 3D		
3D LBP	Extended LBP to 3D volume of data	[101, 102]
Volume LBP	Describe dynamic texture; cause high dimensionality	[103, 104]
Combining with other features		
LBP & Gabor Wavelet	Combine the advantages of Gabor & LBP; increase time cost & high dimensionality	[105, 106, 107, 108, 109]
LBP & SIFT	Bring in the advantages of SIFT; reduces feature vector length	[110, 111, 112]
LBP Histogram Fourier	Obtain rotation invariance globally for whole region	[113]

Table 3.4: Types of LBP algorithms

3.3.6 Gabor Filter

Gabor filter was named after Dennis Gabor. It is linear filter that is used in the field of image processing for edge detection, feature extraction, texture analysis etc. These filters are well suited for applications like texture segmentation spatial and frequency domain as they have the optimal localization property[114]. Basically, they allow certain band of frequencies to pass and block others that is why they are also known as special classes of band pass filter. In simple terms, you can think of a gabor filter as a sinusoidal waves signals of a particular frequency and orientation that is modulated by a Gaussian wave.[115]

In order to analyze the texture of an image or to extract features from an image, a bank of gabor filters are used at different orientations/angles as shown in the Figure 3.18 below. In this research, orientation at angles 0° , 45° , 90° , 135° have been used.

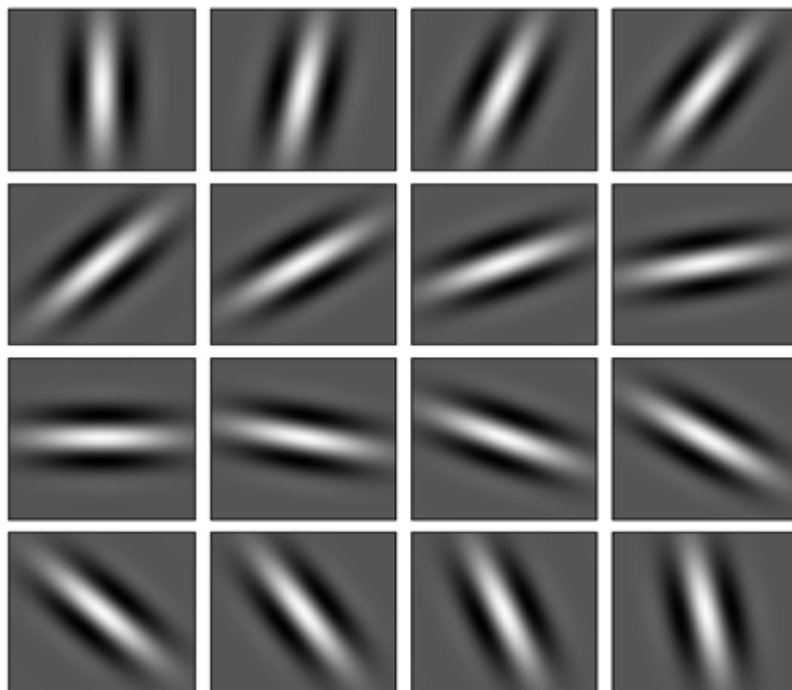


Figure 3.18: Orientations of gabor filters

When we convolve the input image with the with all the gabor filters, patterns in the image are highlighted. When the image is passed through the filter bank, edges that are oriented as the orientation of the filter gets detected. The highest

response of a gabor filter is at edges and at points in the image where there is a change in texture.[116]

Gabor function:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi \frac{x'}{\lambda} + \psi\right)\right) \quad (3.31)$$

Real Component:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right) \quad (3.32)$$

Imaginary Component:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \sin\left(2\pi \frac{x'}{\lambda} + \psi\right) \quad (3.33)$$

Where

$$x' = x \cos(\theta) + y \sin(\theta)$$

$$y' = -x \sin(\theta) + y \cos(\theta)$$

Gabor function is controlled by the following five parameters:

λ –**Lambda** is wave length of the sinusoidal component. It is responsible for controlling the stripes of gabor function. Increase in wavelength would result in thicker stripes.

θ –**theta** represents the orientation of the normal to the parallel stripes of gabor filter. $\theta = 0^\circ$ represents vertical orientation/position of gabor filter.

ψ –**psi** represents the phase offset of the sinusoidal function.

γ –**gamma** or aspect ration actually controls the height of thr hgabor function. Height of the gabor function and value of gamma are inversely proportional. More the aspect ratio less the the height of the gabor function.

σ – **sigma** or bandwidth governs the overall size of the envelope of gabor function. Value of σ is directly proportional to the stripes of the gabor function. if the value of σ is increased from 30 to 45, the number of stripes in the gabor function increases.

Stripes basically are the frequencies as mentioned before gabor filter bank acts as a classes of band pass filter.[116]

Parameter	Value
Gamma	0.3
Lambda	3.51 - 22.57
Sigma	2.81 - 18.01
Theta	$0^\circ, 45^\circ, 90^\circ, 135^\circ$

Table 3.5: Values of Gabor parameters in this research study

For **image decomposition, feature extraction** and creation of the **feature matrix** Matlab R2015a was used. All the images were imported into Matlab one by one. Each of the images were then decomposed into 6 channels R, G, B, H, S, & V. Then features were extracted for each image.

Total feature matrix had a size of 533 x 200100 as shown in the Table 3.6. 533 represents number of rows and 200100 represents columns. Each row represents a image and column represents features. Number of columns appear to be more than the number of features. That is because some features i.e. DCT produce an image of 200 x 200 that were used as it is. These 2D matrices are converted into row matrix and concatenated with rest of the features.

Class Name	Size
Musammi	141 x 200100
Guava	183 x 200100
Fruiter	209 x 200100
Total	533 x 200100

Table 3.6: Feature Matrix size for each class

Chapter 4

Classifier

After detailed literature review of the classifiers used for tree species classification, ANN and SVM were chosen as base classifiers for this research.

4.1 Artificial Neural Network

Neural nets, more precisely referred as ‘Artificial Neural Network’ are actually coded implementation of how human brain (neural structure) works or human brain/biologically inspired set of algorithms. It is also known as ‘artificial neural systems’, ‘parallel distributed processing system’ and ‘connectionist system’ [117] which are designed to recognize patterns.

Our brain is made up of millions of cells known as neurons that process the information in form of electrical signals. Figure 4.1 shows a single biological neuron.

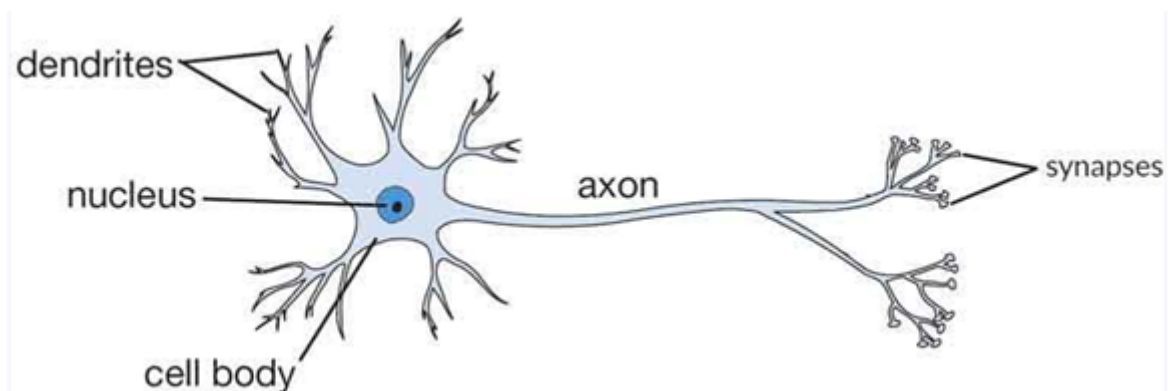


Figure 4.1: Human biological neuron

The functionality of dendrites is that it receives the signal from other neurons. The signal is processed by the ‘cell body’ also known as ‘soma’. All the received signals are summed up to generate an input. When a certain defined threshold level is reached by an input, a neuron is fired and the output signal flows to axon and onto other neurons. ‘Synapses’ is basically point that connects neurons. The next neuron either accepts or rejects the output of the previous neuron based on the signal strength.[118]

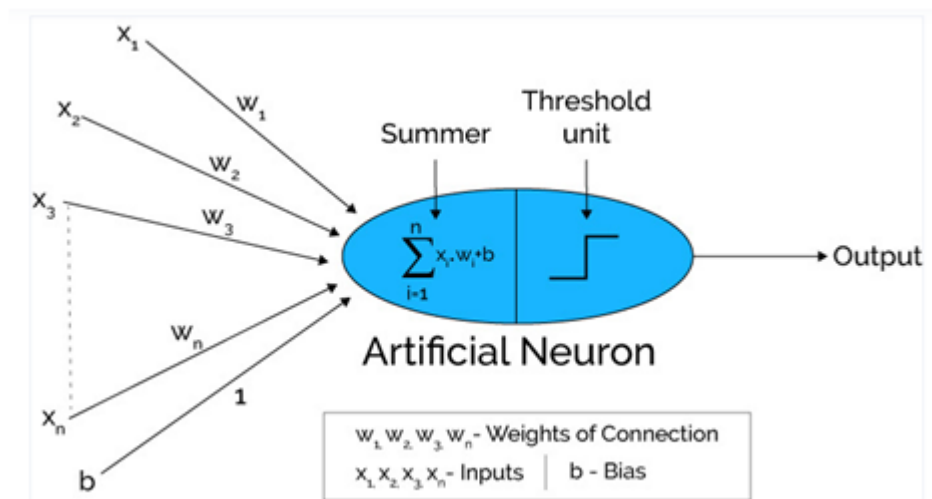


Figure 4.2: Artificial neuron

Figure 4.2 shows the working of artificial neuron. Artificial neuron or a single node works in the same way as human neuron. x_1, x_2, x_3 are the input signals that are multiplied by the weights (w_1, w_2, \dots) and are fed into artificial neuron which acts like cell body in human neural system. All the inputs are summed in the artificial computing unit. The sum of all inputs is passed through the node's activation function. An activation function is critical and important part of a neural network. The function of the activation function is to determine whether the signal should progress through the network and to what extent or whether a neuron should be fired or not (in terms of human neuron). It applies transformation on the weighted sum of inputs and thresholds the output to show whether or not the information, neuron has received, is relevant for the task at hand.

$$= \sum (\text{weight} \times \text{inputs}) + \text{bias} \quad (4.1)$$

Main utility of 'bias' is to make a zero weighted sum non zero or scale up the response of the system. If we do not apply activation function, then the output would be a linear output/function which would simply be a 'Linear Regression Model'. We need non-linear functions, so that ANN can be applied to any complex function and any complicated data. Another feature of activation function is that it should be differentiable because when we want to update the curve, we should know how much change and in which direction the change is occurring. This process is called back propagation. According to the current and desired output error is computed and then weights are adjusted so that the desired output can be produced.

Activation functions can be divided into two types:

- Linear activation function
- Non-linear activation function

The Figure 4.3 shows the list of common activation functions used in ANN:







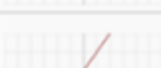


Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

Figure 4.3: Types of activation function

The highest block of machine learning is called a layer. Architecture of neural network contains input, hidden (can be more than one) and an output layer as shown in the figure 4.4 below.

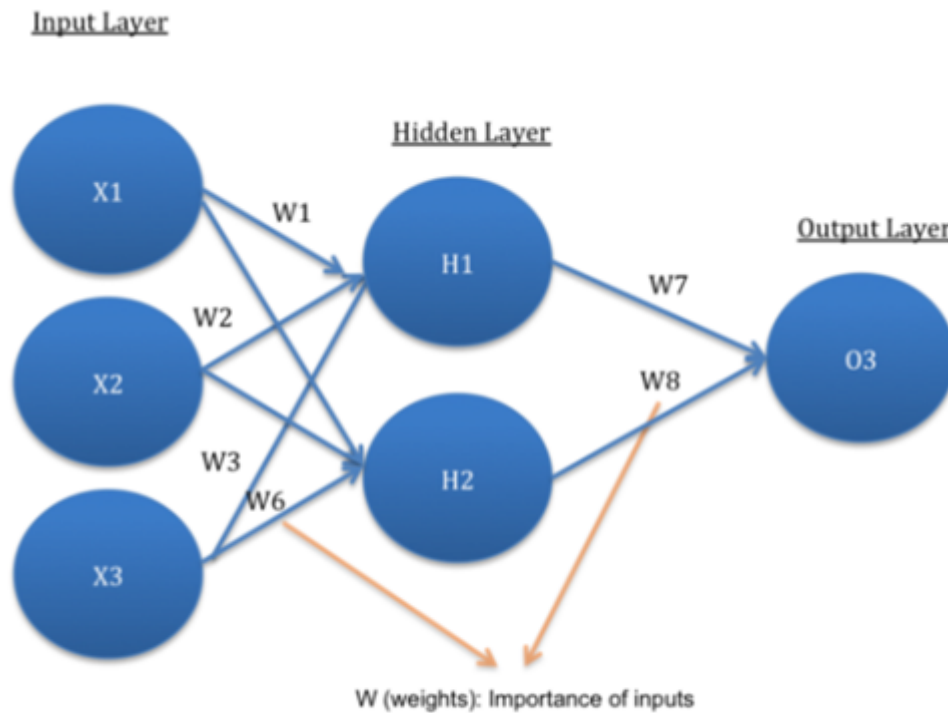


Figure 4.4: Architecture of a neural network

Middle layer is called hidden because its output is hidden or not visible. Each layer contains multiple neurons/nodes and is connected to the nodes of the next layer. Dependency of the relationship between nodes is weights that are associated with the connection between the nodes. The weights are adjusted by calculating error so that cost function can be reduced. Cost function is basically the measure of how close actual output is from the expected. Common optimization algorithms that are used to adjust weights are stochastic gradient descent, batch gradient descent, or mini-batch gradient descent algorithms etc. In short, a layer receives weighted inputs; transform it with an activation/transfer function and pass on the output to the next layer. A layer is usually uniform which means that it usually contains one type of activation function.

4.1.1 Training a neural network

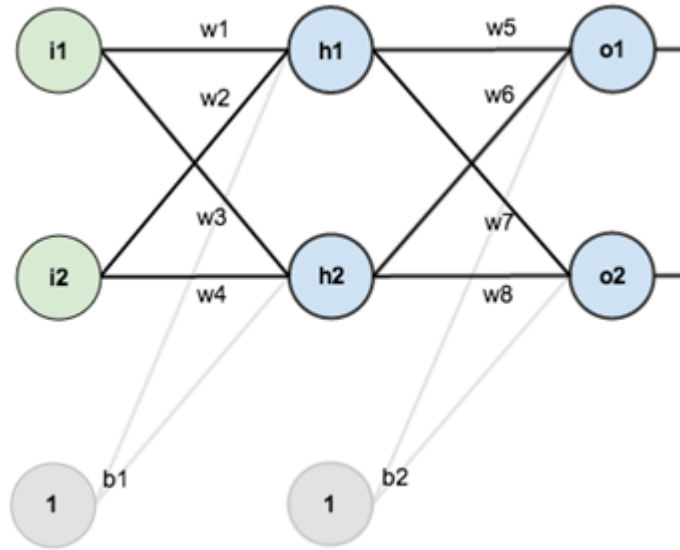


Figure 4.5: A simple neural network

Consider the architecture of simple neural network shown in the Figure 4.5 above. i_1 and i_2 are inputs 1 and 2, w_1 to w_8 are the weights associated with respective connection shown in the picture above. o_1 and o_2 are outputs of the network. b_1 and b_2 are biases used in each layer to make a zero summed input to non-zero value. Input received by h_1 and h_2 nodes are calculated by the following formula

$$net_{h1} = (\omega_1 \times i_1) + (\omega_2 \times i_2) + (b_1) \quad (4.2)$$

$$net_{h2} = (\omega_3 \times i_1) + (\omega_4 \times i_2) + (b_1)$$

Outputs of h_1 and h_2 are calculated by the following formula (depends upon the activation used by the layer)

$$out_{h1} = \frac{1}{1 + e^{-net_{h1}}} \quad (4.3)$$

$$out_{h2} = \frac{1}{1 + e^{-net_{h2}}}$$

Network outputs o1 and o2 are calculated by

$$net_{o1} = (\omega_5 \times out_{h1}) + (\omega_6 \times out_{h2}) + (b_2 \times 1) \quad (4.4)$$

$$net_{o2} = (\omega_7 \times out_{h1}) + (\omega_8 \times out_{h2}) + (b_2 \times 1)$$

Error can be calculated of each neuron and to get total error summation is used

$$E_{Total} = \sum \frac{1}{2} (target - output)^2 \quad (4.5)$$

Let us say we want to calculate the error of o1 and o2

$$E_{o1} = \frac{1}{2} (target_{o1} - output_{o1})^2 \quad (4.6)$$

$$E_{o2} = \frac{1}{2} (target_{o2} - output_{o2})^2$$

In order to adjust weights, we have to do the process shown in the Figure 4.6.

We need to calculate each piece of the equation

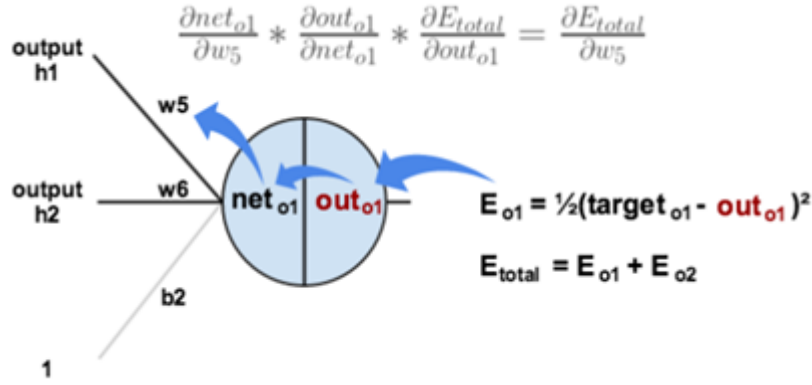


Figure 4.6: Error calculation process

$$E_{Total} = \frac{1}{2}(\text{target}_{o1} - \text{output}_{o1})^2 + \frac{1}{2}(\text{target}_{o2} - \text{output}_{o2})^2 \quad (4.7)$$

$$\frac{\partial E_{Total}}{\partial out_{o1}} = 2 \times \frac{1}{2}(\text{target}_{o1} - \text{output}_{o1})^{2-1} \times -1 + 0 \quad (4.8)$$

$$\text{out}_{o1} = \frac{1}{1 + e^{-net_{o1}}} \quad (4.9)$$

$$\frac{\partial out_{o1}}{\partial net_{o1}} = \text{out}_{o1}(1 - \text{out}_{o1}) \quad (4.10)$$

$$net_{o1} = (\omega_5 \times \text{out}_{h1}) + (\omega_6 \times \text{out}_{h2}) + (b_2 \times 1)$$

$$\frac{\partial net_{o1}}{\partial \omega_5} = 1 \times \text{out}_{h1} \times \omega_5^{1-1} + 0 + 0 \quad (4.11)$$

Putting it all together

$$\frac{\partial E_{Total}}{\partial \omega_{05}} = \frac{\partial E_{Total}}{\partial out_{01}} \times \frac{\partial out_{01}}{\partial net_{01}} \times \frac{\partial net_{01}}{\partial \omega_{05}} \quad (4.12)$$

To decrease the error we have to adjust the weight

$$\omega_5^+ = \omega_5 - \eta \times \frac{\partial E_{Total}}{\partial \omega_5} \quad (4.13)$$

Repeat this process to get the new weights ω_6 , ω_7 and ω_8 .

In the same way perform the back propagation on the hidden layer as done on output layer to adjust ω_1 , ω_2 , ω_3 and ω_4 .

4.1.2 Types of neural networks

There are multiple type neural networks each with different use case and level of complexity. Figure 4.7 shows types of neural networks and their architectures.

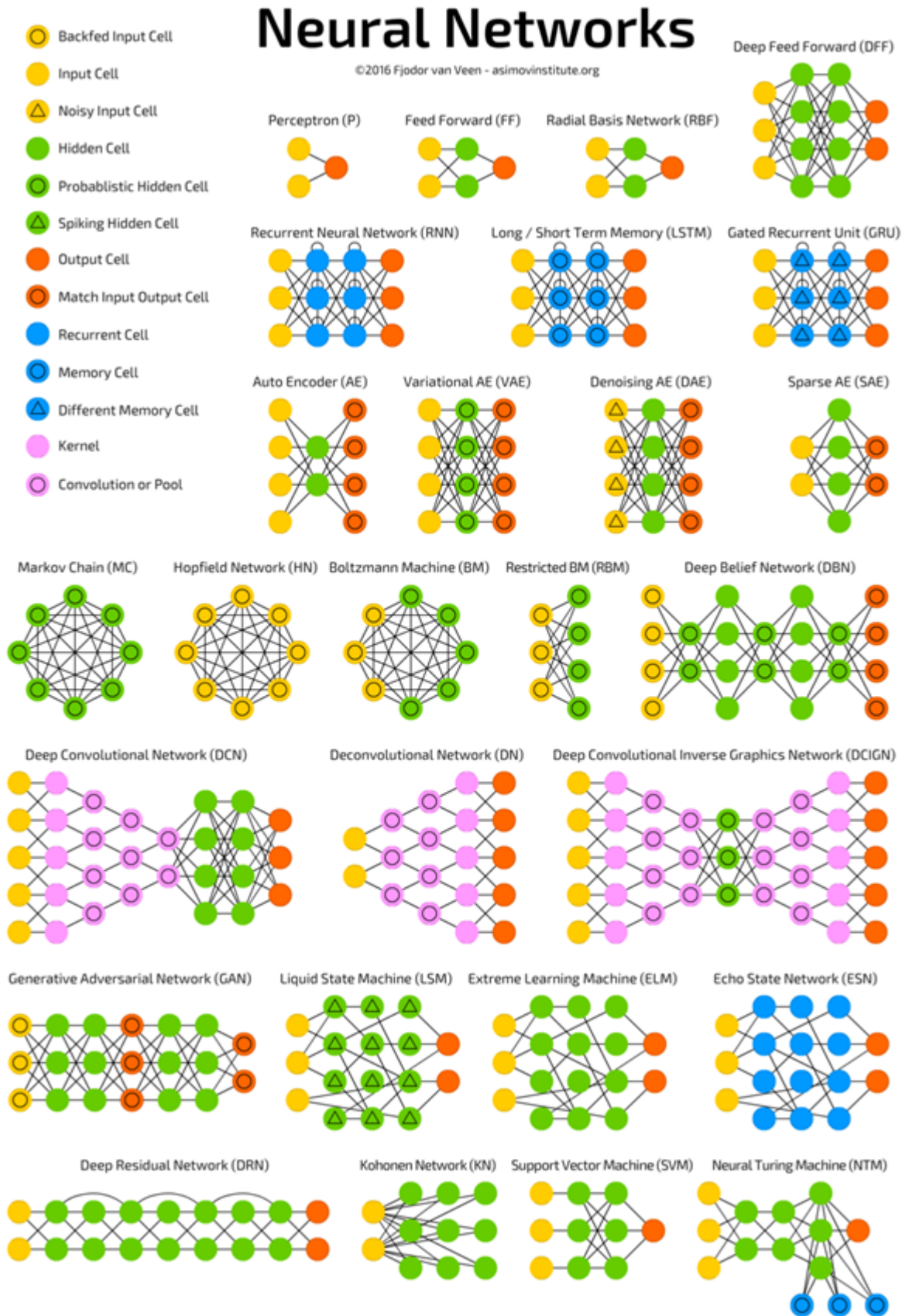


Figure 4.7: Types of neural networks

4.1.3 Learning Techniques used in Neural Networks:

There three main types of learning techniques used in machine learning:

- **Supervised Learning:**

Supervised learning is a machine learning method that uses labelled and known data to train the neural networks or produce specific outputs by adjustment of the system [119]. Following is the list classification and regression algorithms.

- Logistic Regression
- Decision Trees
- Support vector machine(SVM)
- k-Nearest Neighbor(KNN)
- Naive Bayes
- Random Forest(RF)
- Linear Regression
- Polynomial Regression
- SVM for regression

These algorithms are categorized under supervised learning

- **Unsupervised Learning:**

It is a second method associated with machine learning where un-labelled data is used to draw inferences. All clustering algorithms come under unsupervised learning algorithms.

- K – means clustering
- Hierarchical clustering
- Hidden Markov models

- **Reinforcement Learning:**

It is a type of learning where software agents are rewarded with feedback signal or reinforcement signal to learn its behaviour in a specific context.

4.1.4 Advantages and applications of neural network

- A neural network can perform tasks that a linear program cannot.
- When an element of the neural network fails, it can continue without any because of parallel processing nature.
- A neural network learns does not need to be reprogrammed.
- It can be implemented in any application.
- It can be performed without any problem.

Few applications of neural networks are listed below:

- Process modelling and control
- Machine Diagnostics
- Portfolio Management
- Target Recognition
- Medical Diagnosis
- Credit Rating
- Targeted Marketing
- Voice recognition
- Financial Forecasting
- Financial Forecasting
- Fraud detection etc

4.2 SVM

SVM is an abbreviation of Support Vector Machine. It is an algorithm used in supervised machine learning mostly for classification. SVM was invented by Vapnik & Chervonenkis initially. At that time, the algorithm was in early stages as hyper-planes were used to classify linearly [120]. Later in 1992 Vapnik, Boser & Guyon suggested a way for building a non-linear classifier by suggesting the use kernel trick in SVM latest paper [121]. It has been treated as one of the dominant classification algorithms since then. It has been used for classification in multiple areas like gene selection [122] cancer classification [123], text classification [124][125], classification of remote sensing images [126][127], tree species mapping [128] and many others.

For a dataset consisting of features set and labels set, an SVM classifier builds a model to predict classes for new examples. It assigns new example/data points to one of the classes. If there are only 2 classes then it can be called as a Binary SVM Classifier.

Objective of SVM is basically to separate classes by finding the hyper-plane (for N dimensional spaces) or a line that separates two classes distinctly. Hyper-planes are basically boundaries that decide that from which class a data point belongs to.

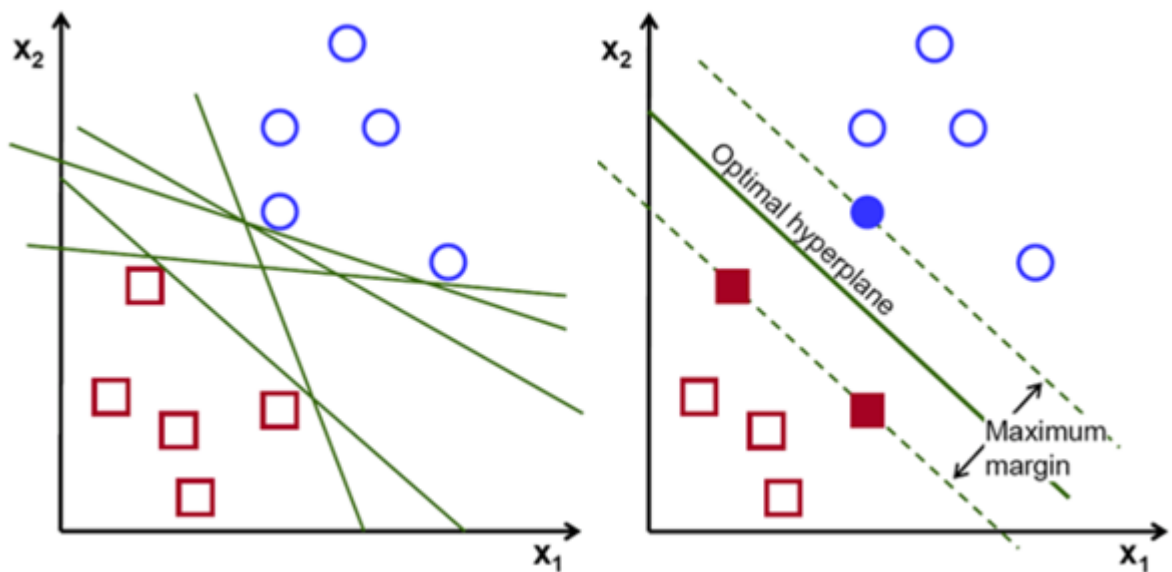


Figure 4.8: Different hyperplanes

In order to separate two classes, there may be many possible options of hyper-plane that can separate the classes as shown in the Figure 4.8 above. But the aim is to find that hyper-plane (for more than two classes) or a line (in case of two classes) that has the maximum margin. A margin is actually is a line/hyper-plane of separation to the closest data points also known as support vectors, Figure 4.9, of all classes that are in consideration.

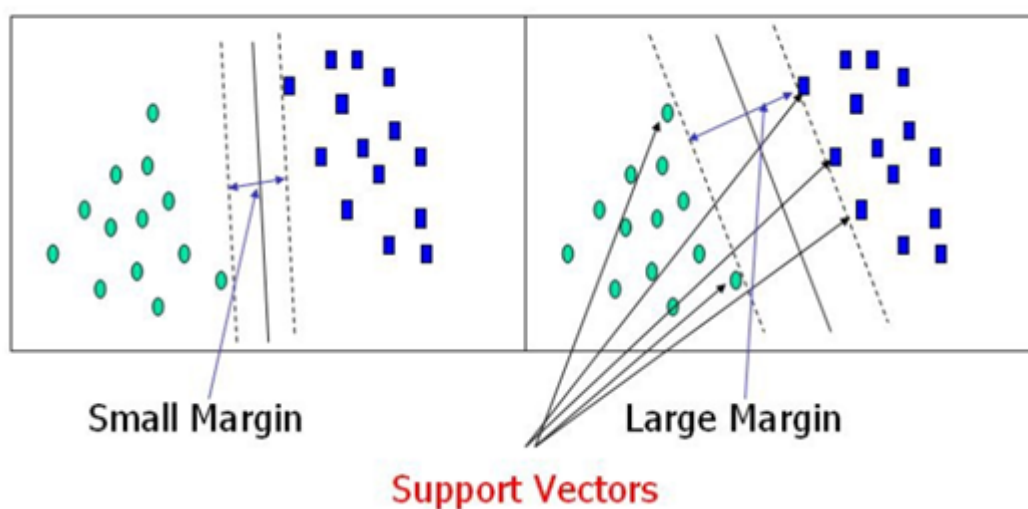


Figure 4.9: Different hyper-planes

A good margin is where the distance of line/hyper-plane to support vectors of each class/feature is maximum as shown in the Figure 4.10 below.

Dimension of data plane is dependent on the number of features being used. Consider the data points distribution the two classes shown in the Figure 4.11 below. Data point could easily be separated with a line. But what if the distribution of the two classes was as shown in Figure 4.12?

In this case we need to incorporate the z-axis as shown in the Figure 4.13 and a clear separation can be seen.

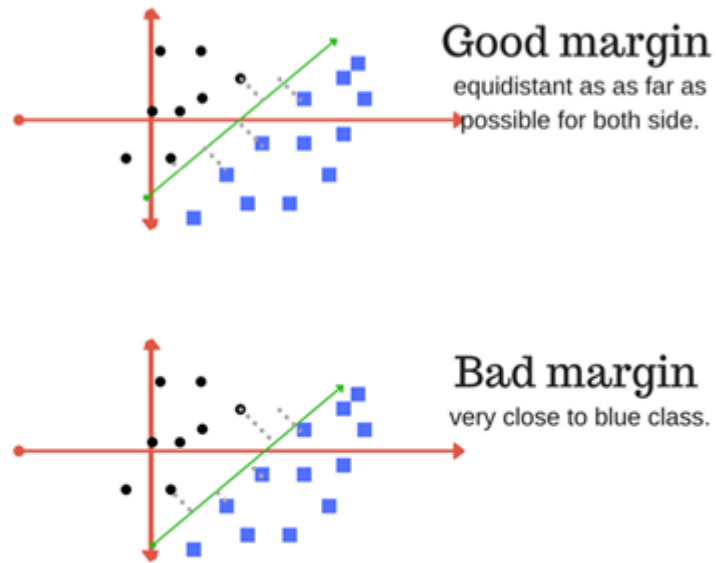


Figure 4.10: Good and bad margins

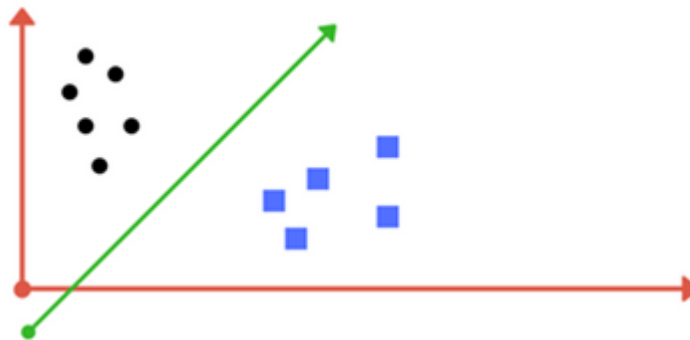


Figure 4.11: Classes distributed with simple separation

When we transform into original space form, a circular boundary is mapped 4.14. In addition what if the situation is bit more complex. What if data points overlap as shown in Figure 4.15 below?

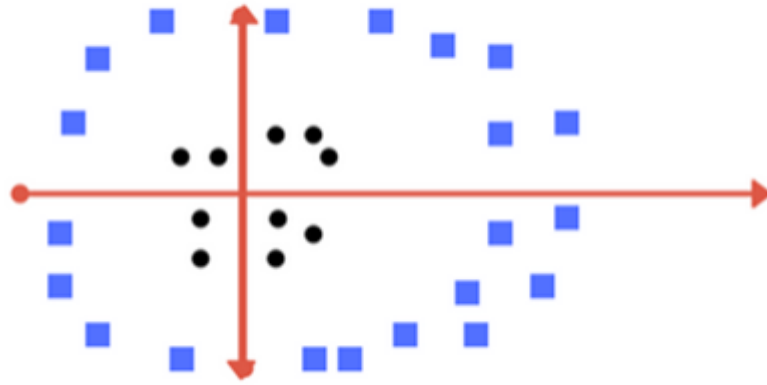


Figure 4.12: Classes distribution with complex boundary

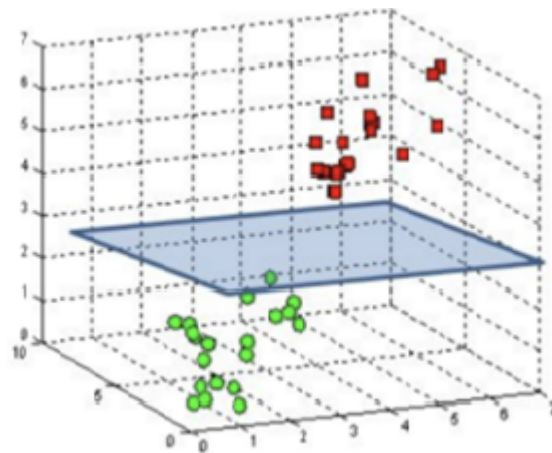


Figure 4.13: Transition data into 3-D

Possible solutions to the aforementioned problem are shown in the Figure 4.15 above. In Image1 hyper plane has excluded two black dots considering them to be outliers or exceptions. Image 2 has achieved perfect separation without any tolerance. In real life examples we need a trade-off because for training millions of data set requires huge amount of time in order to find the perfect answer.

4.2.1 Tuning Parameters of SVM

1. Kernel

Linear SVM learns about hyper-plane by transforming the problem using linear algebra. SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different

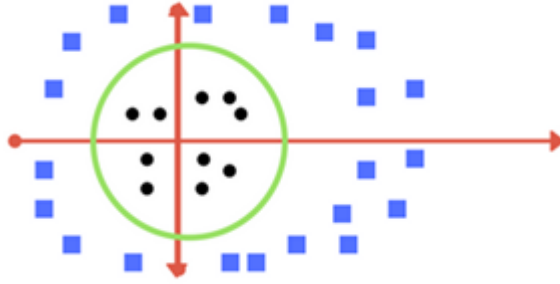


Figure 4.14: Transition back to 2-D after drawing hyper-plane

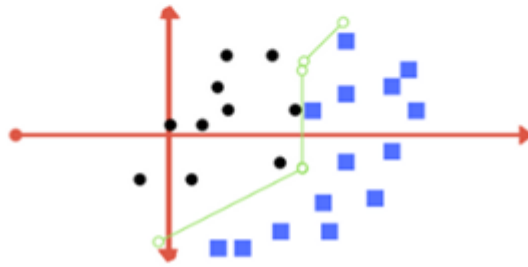
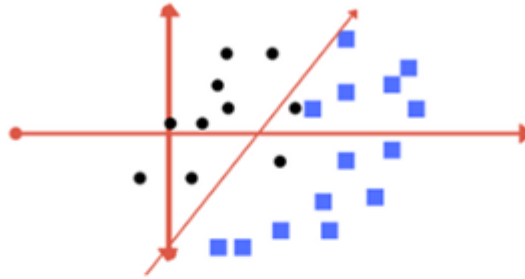


Figure 4.15: Overlapping distribution and possible hyper-planes

types of kernel functions for transformation such as linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.[129]

In order to predict a new input for a linear kernel equation is as follows:

$$f(x) = B(0) + \sum (a_i * (I(x), x_i)) \quad (4.14)$$

This is an equation calculates a new input vector y taking inner product $I(x)$ with all support vectors in training data. The learning algorithms must learn and estimate coefficients $B(0)$ and a_i (for each input) from the training data.

The polynomial kernel can be written as

$$K(I(x), x_i) = 1 + \sum (I(x) * x_i)^d \quad (4.15)$$

and exponential as

$$K(I(x), x_i) = e^{-\gamma * \sum (I(x) - x_i^2)} \quad (4.16)$$

In short, kernels (exponential or polynomial) determine the separation line in higher dimensions. It is also known as ‘kernel trick’.

2. Regularization

It is a parameter that tells how much misclassification needs to be avoided.

Look at the example in Figure 4.16:



Figure 4.16: Left: Low regularization value, Right: High regularization value

Image on the left has lower regularization value as it contains some misclassification. While on the other hand, image in the right side has high regularization as it has 0 tolerances for misclassification. The objective of the regularization parameter is to balance the margin maximization and loss.

3. **Gamma** Gamma tells us about the support vectors that influence in the calculation. If gamma is high that means data points that are closer to the line are under consideration and vice versa as clearly illustrated in the Figure 4.17.

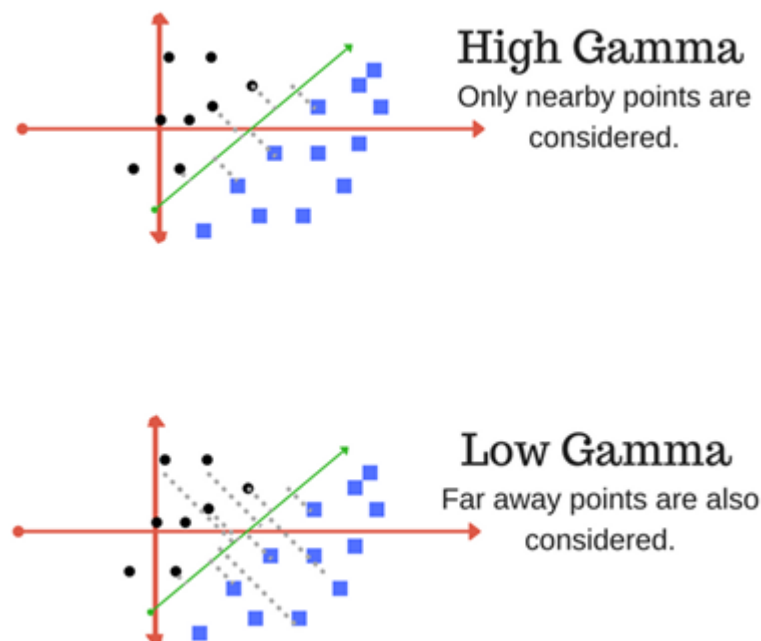


Figure 4.17: Top: High gamma value, Bottom: Low gamma value

4. **Margin** A margin is a separation of line to the closest class points. A good margin is one where this separation is larger for both the classes. Figure 4.18 below gives to visual example of good and bad margin.

4.2.2 Cost Function and Gradient Update

As mentioned earlier the objective of the SVM is to find the line or hyperplane that maximizes the margin. Loss function is used to reduce the cost and maximize the margin.

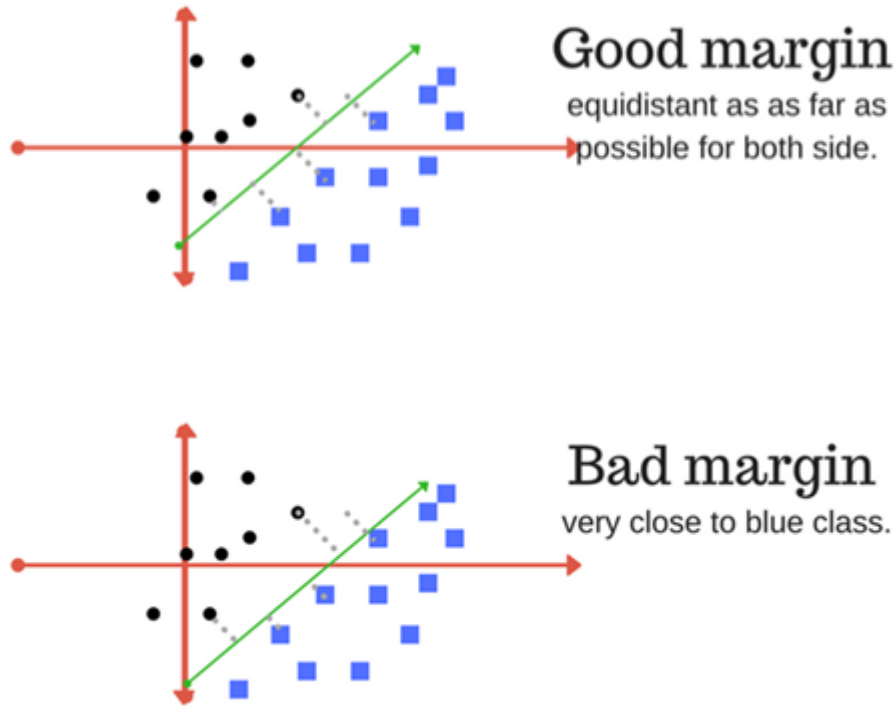


Figure 4.18: Top: Good margin, Bottom: Bad Margin

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases} \quad (4.17)$$

Initially feature matrix for each class was made separately and using Matlab command '.csv' files were generated. For each class an extra file created that contained class codes that represented the target. For example target value for Musammi was '100', for Guava value was '010' and for Fruiter, value was '001'. From these .csv files of each class 100 rows, total 300, were used to make another .csv file that would be used to train the classifier models. Equal number of data rows were selected from each class so that the model performance is not towards any class. Rest 232 rows were used for testing the accuracy of the model.

These 3 files 'train.csv', 'test.csv' and 'target.csv' were fed to the classifiers for classification. Classification models of both classifiers, Artificial Neural Network (ANN) & Support Vector Machine, were created using python language. 'Google

Colaboratory', an environment for python coding that runs entirely on the cloud and is free was used for processing. The best part of this environment is that there is no need to install modules and libraries as we have to do for the python environments, e.g. jupyter notebook, on our personal computer.

Chapter 5

Performance Evaluation

The metric used to evaluate the output of neural network & SVM model is 'Accuracy'. In classification problems, to find the accuracy of the model a confusion matrix is used. A confusion matrix, shown in Figure 5.1 provides detail about the performance of the classification model.

		Predicted Values	
		Good	Bad
Actual Values	Good	True Positive (TP)	False Negative (FN)
	Bad	False Positive (FP)	True Negative (TN)

Figure 5.1: Confusion Matrix

The words 'Good' and 'Bad' in Figure 5.1 actually represent the classes which in the case of this research study are 'Musammi', 'Fruiter' and 'Guava'.

True Positive (TP) is the case where actual value is true and the model also predicted the value to be true.

False Negative (FN) is the case where actual value is true and the model predicted the value to be false.

False Positive (FP) is the case where actual value is false and the model predicted the value to be true.

True Negative (TN) is the case where actual value is false and the model also predicted the value to be false.

Accuracy of a model is calculated by the following equation

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.1)$$

5.1 Experiments

In order to get the best results from the ANN model, several experiments were performed. For ANN classification I used Multi-layer Perceptron(MLP). MLP has a non-linear activation function which can distinguish the data, that is not linearly separable, easily. In this study multi-layer refers to multiple hidden layers. I used MLP neural network model because single hidden layer produced 100% accuracy which shows that the model is over-fitted.

5.1.1 Experiment 1

Initially, the number of neurons in each hidden layer were kept same as shown in Figure 5.2. This was done to check what effect number of hidden layers has on the accuracy of the model and what number of hidden layers are suitable for this research study.

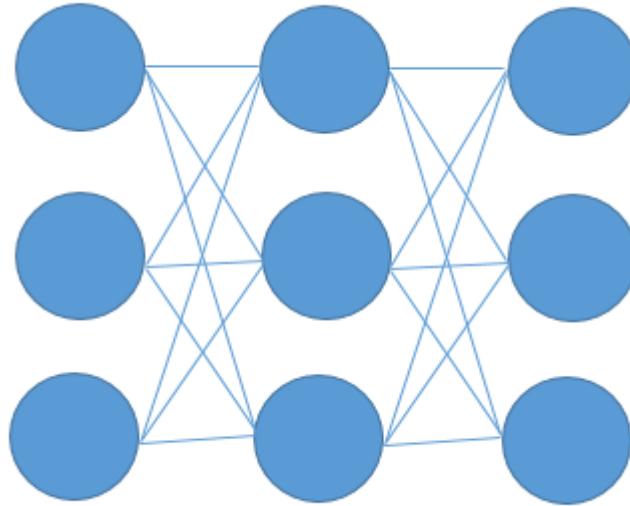


Figure 5.2: Hidden Layers with equal number of neurons

Number of Layers	Accuracy	Epochs
1	100%	2500
2	100%	2500
3	86.15%	2500
4	80.52%	2500
5	74.03%	2500
6	93.51%	2500
7	38.53%	2500
8	63.64%	2500
9	86.58%	2500
10	73.16%	2500

Table 5.1: Parameter Details

Epoch shows the number of times data set passes through the model. Figure 5.3 shows the accuracy results achieved.

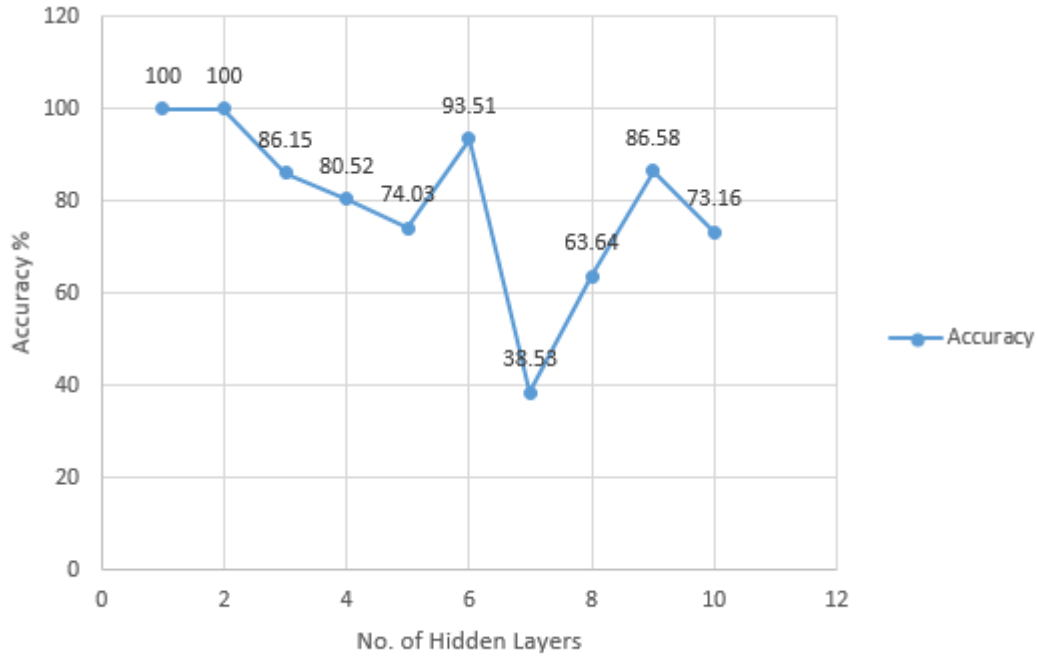


Figure 5.3: Accuracy results

It can be seen from the Figure 5.3 that the best result in terms of accuracy, 93.51%, was achieved using 6 hidden layers, excluding the results of hidden layer 1 & 2 as they depict the over-fitted model. Over-fitted models are those that have mastered training data but have not learned to make the condition generalized. Confusion matrix of experiment using 6 hidden layers is shown in the Figure 5.4.

Referring to Table 3.6, 'Musammi' had 41 test images, out of which 36 were accurately classified. 'Fruiter' had 109 test images, out of which 100 images were accurately classified. Lastly, 'Guava' had 83 total images for testing the classification model and 82 of them were accurately classified by the neural network model with 6 hidden layers.

		Predicted		
		Musammi	Fruiter	Guava
Actual	Musammi	36	4	1
	Fruiter	6	100	3
	Guava	0	1	82

Figure 5.4: Confusion Matrix

5.1.2 Experiment 2

In 2nd experiment I used different numbers of hidden layers with different number of neurons in it. First hidden layer had most neurons with decreasing number of neurons in the following layers. Shape of the hidden layer is converging towards the output layer in this experiment as shown in the Figure 5.5.

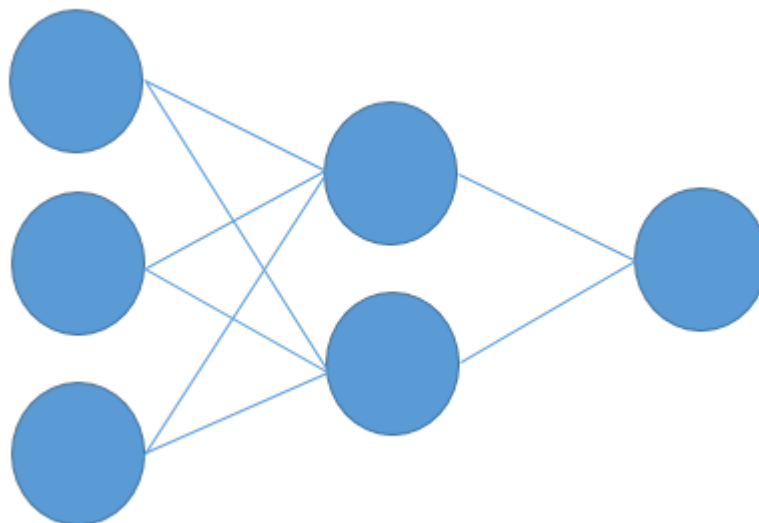


Figure 5.5: Converging Hidden Layer

Number of Layers	Accuracy	Epochs
1	100%	2500
2	100%	2500
3	72.82%	2500
4	74.03%	2500
5	67.13%	2500
6	79.22%	2500
7	84.85%	2500
8	87.53%	2500
9	71.68%	2500
10	97.4%	2500

Table 5.2: Parameter Details

In this experiment, 10 hidden with converging shape produced the best result by giving 97.4% of accuracy as shown in the Table 5.2 & Figure 5.6. In both experiments, 1 & 2, both models proved to be over-fitted when 1 or 2 hidden layers were used by providing 100% accuracy.

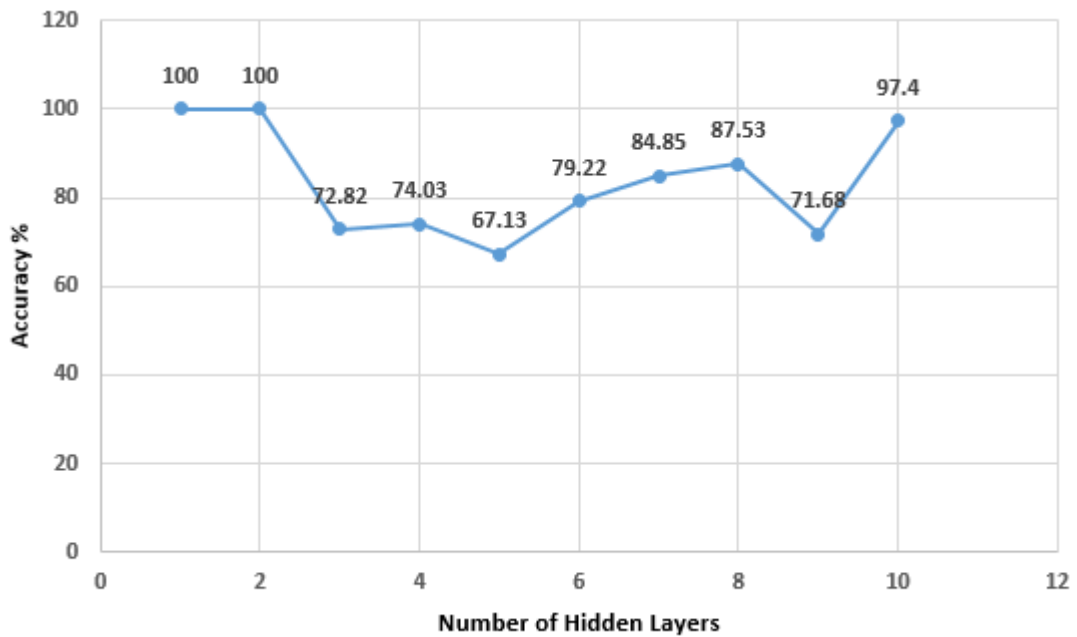


Figure 5.6: Accuracy results

Referring to Table 3.6, 'Musammi' had 41 test images, out of which 40 were accurately classified. 'Fruiter' had 109 test images, out of which 105 images were accurately classified. Lastly, 'Guava' had 83 total images for testing the classification model and 82 of them were accurately classified by the neural network model with 10 hidden layers.

		Predicted		
		Musammi	Fruiter	Guava
Actual	Musammi	40	0	1
	Fruiter	2	105	2
	Guava	1	0	82

Figure 5.7: Confusion Matrix

5.1.3 Experiment 3

After experimenting on different hidden layers in ANN model in this research study I used SVM as a second classifier on my dataset. Table 5.3 show the parameter details of support vector classifier (SVC) and produced accuracy of 96.26%.

Tuning Parameters	Values
Kernel	Linear
C (Penalty Parameter)/ Regularization	1
Gamma	Auto
Margin	Auto

Table 5.3: SVM Parameter Details

Referring to Table 3.6, 'Musammi' had 41 test images, out which 39 were accurately classified. 'Fruiter' had 109 test images, out of which 106 images were accurately classified. Lastly, 'Guava' had 83 total images for testing the classification model and 79 of them were accurately classified by the SVM.

		Predicted		
		Musammi	Fruiter	Guava
Actual	Musammi	38	2	1
	Fruiter	1	106	2
	Guava	2	2	79

Figure 5.8: Confusion Matrix

5.1.4 Experiment 4

Feature	Accuracy (ANN)	Accuracy (SVM)
DCT	76.10%	75.70%
Gabor	82.13%	81.31%
LBP	30.58%	34.58%
GLCM	16.88%	47.66%
Mean	90.04%	88.79%
MaxMean	76.5%	90.65%
MinMean	91.77%	87.85%
Median	89.95%	85.02%
Mode	91.02%	86.25%
Max	97.4%	89.72%
Min	78.05%	86.92%
SD	100%	88.78%
Skewness	93.94%	90.65%
Kurtosis	99.13%	85.05%

Table 5.4: Accuracy comparison of individual features

Accuracy of individual classifiers has been calculated for each feature in the Table 5.4. The results show that performance of ANN is better than SVM over all except for few features where SVM performed better.

Chapter 6

Conclusion & Future Work

This research study classified tree species using optical aerial imagery, extracted spatial and spectral features and use machine learning algorithms for classification. Aerial imagery of three species of trees, Musammi, Guava, Fruiter, were obtained and used as a data set in this research study. Spatial features,i.e. LBP, DCT, Gabor and other intensity based features like Mean, Median, Standard Deviation etc, were extracted which contributed significantly for high value of classification. Detailed analysis was done to get the best approach for tree species classification. After studying around 50 research papers and doing literature review two machine learning techniques were, Artificial Neural Network (ANN) and Support Vector Machine(SVM), for tree species classification as these two techniques are mostly used. These methods, ANN and SVM, turned out to be very useful and gave good results by producing 97.4% and 96.26% accuracy respectively for three classes of trees.

In order to use aerial imagery at larger scale, the next step is to extract other features i.e. morphological features (branchstarness, granulometry), gradient based feature (Forstner, gradstarness, gradanistropy, gradientropy), wavelet based features, geometric features i.e. features that describe the shape, area or boundary on aerial imagery and use them for tree species classification as these aforementioned classes of features were not used. Feature matrix utilized in this research was large and effect of over fitting was quite prominent, may be because of small data-set, Dimensionality reduction (decreasing the number of dimensions in the feature vector)

could be another way forward to reduce the size of the feature matrix. In addition, feature selection could also be incorporated and use only that feature contribute the most, produce optimum results and make the classification model more robust. Furthermore, it is claimed that spatial features are more robust to factors i.e. sun-illumination and viewing angle must be confirmed by experiment. Another topic of future work that can be added to this research is tree detection.

Bibliography

- [1] Dji phantom 4 pro – photography drone – dji. <https://www.dji.com/phantom-4-pro>.
- [2] Jun-Dong Chang, Shyr-Shen Yu, Hong-Hao Chen, and Chwei-Shyong Tsai. Hsv-based color texture image classification using wavelet transform and motif patterns. *Journal of Computers*, 20(4):63–69, 2010.
- [3] Saeed Dabbaghchian, Masoumeh P Ghaemmaghami, and Ali Aghagolzadeh. Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology. *Pattern Recognition*, 43(4):1431–1440, 2010.
- [4] Robert M Haralick, Karthikeyan Shanmugam, et al. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.
- [5] Steven E Taylor, Timothy P McDonald, Matthew W Veal, and Ton E Grift. Using gps to evaluate productivity and performance of forest machine systems. pages 151–55, 2001.
- [6] Jili Li, Baoxin Hu, and Murray Woods. A two-level approach for species identification of coniferous trees in central ontario forests based on multispectral images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(4):1487–1497, 2015.
- [7] Xiaowei Yu, Paula Litkey, Juha Hyypä, Markus Holopainen, and Mikko Vastaranta. Assessment of low density full-waveform airborne laser scanning for individual tree detection and tree species classification. *Forests*, 5(5):1011–1031, 2014.

- [8] Jili Li, Baoxin Hu, and Thomas L Noland. Classification of tree species based on structural features derived from high density lidar data. *Agricultural and forest meteorology*, 171:104–114, 2013.
- [9] G Xu, Y Pang, Z Li, D Zhao, and L Liu. Individual trees species classification using relative calibrated fullwaveform lidar data. In *2012 Silvilaser Int. Conf. on Lidar Applications for Assessing Forest Ecosystems*, pages SL2012–SL2084, 2012.
- [10] Haider Taha, Sharon Douglas, and Jay Haney. Mesoscale meteorological and air quality impacts of increased urban albedo and vegetation. *Energy and Buildings*, 25(2):169–177, 1997.
- [11] Argiro Dimoudi and Marialena Nikolopoulou. Vegetation in the urban environment: microclimatic analysis and benefits. *Energy and buildings*, 35(1):69–76, 2003.
- [12] Jinyu Sun, Xuhui Wang, Anping Chen, Yuecun Ma, Mengdi Cui, and Shilong Piao. Ndvi indicated characteristics of vegetation cover change in china’s metropolises over the last three decades. *Environmental monitoring and assessment*, 179(1-4):1–14, 2011.
- [13] Juanjuan Zhao, Shengbin Chen, Bo Jiang, Yin Ren, Hua Wang, Jonathan Vause, and Haidong Yu. Temporal trend of green space coverage in china and its relationship with urbanization over the last two decades. *Science of the Total Environment*, 442:455–465, 2013.
- [14] Xia Zhao, Daojing Zhou, and Jingyun Fang. Satellite-based studies on large-scale vegetation changes in china f. *Journal of integrative plant biology*, 54(10):713–728, 2012.
- [15] Xiaohua Tong, Xiaoichun Li, Xiong Xu, Huan Xie, Tiantian Feng, Tong Sun, Yanmin Jin, and Xiangfeng Liu. A two-phase classification of urban vegetation using airborne lidar data and aerial photography. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(10):4153–4166, 2014.
- [16] Renaud Mathieu, Jagannath Aryal, Albert Chong, et al. Object-based classification of ikonos imagery for mapping large-scale vegetation communities in urban areas. *Sensors*, 7(11):2860–2880, 2007.

- [17] Q Zhou, M Robson, and P Pilesjo. On the ground estimation of vegetation cover in australian rangelands. *International Journal of Remote Sensing*, 19(9):1815–1820, 1998.
- [18] Juha Hyypya, Olavi Kelle, Mikko Lehtikainen, and Mikko Inkinen. A segmentation-based method to retrieve stem volume estimates from 3-d tree height models produced by laser scanners. *IEEE Transactions on Geoscience and remote Sensing*, 39(5):969–975, 2001.
- [19] Bin Wu, Bailang Yu, Qiusheng Wu, Yan Huang, Zuoqi Chen, and Jianping Wu. Individual tree crown delineation using localized contour tree method and airborne lidar data in coniferous forests. *International journal of applied earth observation and geoinformation*, 52:82–94, 2016.
- [20] Lin Cao, Nicholas C Coops, John L Innes, Jinsong Dai, Honghua Ruan, and Guanghui She. Tree species classification in subtropical forests using small-footprint full-waveform lidar data. *International journal of applied earth observation and geoinformation*, 49:39–51, 2016.
- [21] Daniele Marinelli, Claudia Paris, and Lorenzo Bruzzone. Fusion of multitemporal lidar data for individual tree crown parameter estimation on low density point clouds. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 3999–4002. IEEE, 2018.
- [22] LI Duncanson, BD Cook, GC Hurtt, and RO Dubayah. An efficient, multi-layered crown delineation algorithm for mapping individual tree structure across multiple ecosystems. *Remote Sensing of Environment*, 154:378–386, 2014.
- [23] Cédric Vége, A Hamrouni, S El Mokhtari, Jules Morel, J Bock, J-P Renaud, M Bouvier, and Sylvie Durrieu. Ptrees: A point-based approach to forest tree extraction from lidar data. *International Journal of Applied Earth Observation and Geoinformation*, 33:98–108, 2014.
- [24] Victor F Strîmbu and Bogdan M Strîmbu. A graph-based segmentation algorithm for tree crown extraction using airborne lidar data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104:30–43, 2015.

- [25] António Ferraz, Sassan Saatchi, Clément Mallet, and Victoria Meyer. Lidar detection of individual tree size in tropical forests. *Remote Sensing of Environment*, 183:318–333, 2016.
- [26] António Ferraz, Frédéric Bretar, Stéphane Jacquemoud, Gil Gonçalves, Luisa Pereira, Margarida Tomé, and Paula Soares. 3-d mapping of a multi-layered mediterranean forest using als data. *Remote Sensing of Environment*, 121:210–223, 2012.
- [27] Elhadi Adam, Onesimo Mutanga, and Denis Rugege. Multispectral and hyperspectral remote sensing for identification and mapping of wetland vegetation: a review. *Wetlands Ecology and Management*, 18(3):281–296, 2010.
- [28] Enrica Belluco, Monica Camuffo, Sergio Ferrari, Lorenza Modenese, Sonia Silvestri, Alessandro Marani, and Marco Marani. Mapping salt-marsh vegetation by multispectral and hyperspectral remote sensing. *Remote sensing of environment*, 105(1):54–67, 2006.
- [29] Susan L Ustin, Dar A Roberts, John A Gamon, Gregory P Asner, and Robert O Green. Using imaging spectroscopy to study ecosystem processes and properties. *BioScience*, 54(6):523–534, 2004.
- [30] Galal Omer, Onesimo Mutanga, Elfatih M Abdel-Rahman, and Elhadi Adam. Performance of support vector machines and artificial neural network for mapping endangered tree species using worldview-2 data in dukuduku forest, south africa. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(10):4825–4840, 2015.
- [31] Yuanyong Dian, Zengyuan Li, and Yong Pang. Spectral and texture features combined for forest tree species classification with airborne hyperspectral imagery. *Journal of the Indian Society of Remote Sensing*, 43(1):101–107, 2015.
- [32] Paras Pant, Ville Heikkinen, Ilkka Korpela, Markku Hauta-Kasari, and Timo Tokola. Logistic regression-based spectral band selection for tree species classification: Effects of spatial scale and balance in training samples. *IEEE Geoscience and Remote Sensing Letters*, 11(9):1604–1608, 2014.
- [33] Kamaruzaman Jusoff. Land use and cover mapping with airborne hyperspectral imager in setiu, malaysia. *Journal of Agricultural Science*, 1(2):120, 2009.

- [34] George P Petropoulos, Kostas Arvanitis, and Nick Sigrimis. Hyperion hyperspectral imagery analysis combined with machine learning classifiers for land use/cover mapping. *Expert systems with Applications*, 39(3):3800–3809, 2012.
- [35] Alex Okiemute Onojeghuo and George Alan Blackburn. Optimising the use of hyperspectral and lidar data for mapping reedbed habitats. *Remote Sensing of Environment*, 115(8):2025–2034, 2011.
- [36] Peng Gong, Ruiliang Pu, and John R Miller. Correlating leaf area index of ponderosa pine with hyperspectral data. *Canadian Journal of Remote Sensing*, 18(4):275–282, 1992.
- [37] Shachak Pe’eri, J Ru Morrison, Fred Short, Arthur Mathieson, Anna Brook, and Philip Trowbridge. Macroalgae and eelgrass mapping in great bay estuary using aisa hyperspectral imagery. 2008.
- [38] Matthew L Clark, Dar A Roberts, and David B Clark. Hyperspectral discrimination of tropical rain forest tree species at leaf to crown scales. *Remote sensing of environment*, 96(3-4):375–398, 2005.
- [39] PH Rosso, SL Ustin, and A Hastings. Mapping marshland vegetation of san francisco bay, california, using hyperspectral data. *International Journal of Remote Sensing*, 26(23):5169–5191, 2005.
- [40] P. C. Pandey, N. J. Tate, and H. Balzter. Mapping tree species in coastal portugal using statistically segmented principal component analysis and other methods. *IEEE Sensors Journal*, 14(12):4434–4441, Dec 2014.
- [41] Moses Azong Cho, Oupa Malahlela, and Abel Ramoelo. Assessing the utility worldview-2 imagery for tree species mapping in south african subtropical humid forest and the conservation implications: Dukuduku forest patch as case study. *International Journal of Applied Earth Observation and Geoinformation*, 38:349–357, 2015.
- [42] A. Jamil and B. Bayram. Tree species extraction and land use/cover classification from high-resolution digital orthophoto maps. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(1):89–94, Jan 2018.

- [43] Dawei Wen, Xin Huang, Hui Liu, Wenzhi Liao, and Liangpei Zhang. Semantic classification of urban trees using very high resolution satellite imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(4):1413–1424, 2017.
- [44] W. Li, H. Fu, and L. Yu. Deep convolutional neural network based large-scale oil palm tree detection for high-resolution remote sensing images. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 846–849, July 2017.
- [45] X. Huang, C. Shi, and S. C. Liew. Tree crown detection and delineation using optical satellite imagery. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 2944–2947, July 2018.
- [46] F Tsai, E-K Lin, and K Yoshino. Spectrally segmented principal component analysis of hyperspectral imagery for mapping invasive plant species. *International Journal of Remote Sensing*, 28(5):1023–1039, 2007.
- [47] Andrew A Green, Mark Berman, Paul Switzer, and Maurice D Craig. A transformation for ordering multispectral data in terms of image quality with implications for noise removal. *IEEE Transactions on geoscience and remote sensing*, 26(1):65–74, 1988.
- [48] Lori Mann Bruce, Cliff H Koger, and Jiang Li. Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction. *IEEE Transactions on geoscience and remote sensing*, 40(10):2331–2338, 2002.
- [49] Jan J Nossin. Pm mather, computer processing of remotely sensed images, john wiley & sons, price gb£ 37.50, 324 pages, 9 chapters, with a cd-rom with exercises contributed by magaly koch, isbn 0-470-84919-3., 2005.
- [50] Matthew L Clark and Dar A Roberts. Species-level differences in hyperspectral metrics among tropical rainforest trees as determined by a tree-based classifier. *Remote Sensing*, 4(6):1820–1855, 2012.
- [51] JA Brockhaus and S Khorram. A comparison of spot and landsat-tm data for use in conducting inventories of forest resources. *International Journal of Remote Sensing*, 13(16):3035–3043, 1992.

- [52] Hongyu Huang, Xu Li, and Chongcheng Chen. Individual tree crown detection and delineation from very-high-resolution uav images based on bias field and marker-controlled watershed segmentation algorithms. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(7):2253–2262, 2018.
- [53] Jeanne E Anderson, Lucie C Plourde, Mary E Martin, Bobby H Braswell, Marie-Louise Smith, Ralph O Dubayah, Michelle A Hofton, and J Bryan Blair. Integrating waveform lidar with hyperspectral imagery for inventory of a northern temperate forest. *Remote Sensing of Environment*, 112(4):1856–1870, 2008.
- [54] Johan Holmgren, Å Persson, and U Söderman. Species identification of individual trees by combining high resolution lidar data with multi-spectral images. *International Journal of Remote Sensing*, 29(5):1537–1552, 2008.
- [55] Wen Zhang, Baoxin Hu, Linhai Jing, Murray E Woods, and Paul Courville. Automatic forest species classification using combined lidar data and optical imagery. In *IGARSS 2008-2008 IEEE International Geoscience and Remote Sensing Symposium*, volume 3, pages III–134. IEEE, 2008.
- [56] Michael A Wulder, Joanne C White, Ross F Nelson, Erik Næsset, Hans Ole Ørka, Nicholas C Coops, Thomas Hilker, Christopher W Bater, and Terje Gobakken. Lidar sampling for large-area forest characterization: A review. *Remote Sensing of Environment*, 121:196–209, 2012.
- [57] Jean-Paul Donnay, Mike J Barnsley, and Paul A Longley. *Remote sensing and urban analysis: GISDATA 9*. CRC Press, 2000.
- [58] Fabio Pacifici, Fabio Del Frate, William J Emery, Paolo Gamba, and Jocelyn Chanussot. Urban mapping using coarse sar and optical data: Outcome of the 2007 grss data fusion contest. *IEEE Geoscience and Remote Sensing Letters*, 5(3):331–335, 2008.
- [59] Kishore C Swain, Steven J Thomson, and Hemantha PW Jayasuriya. Adoption of an unmanned helicopter for low-altitude remote sensing to estimate yield and total biomass of a rice crop. *Transactions of the ASABE*, 53(1):21–27, 2010.
- [60] Ye Seul Lim, Phu Hien La, Jong Soo Park, Mi Hee Lee, Mu Wook Pyeon, and Jee-In Kim. Calculation of tree height and canopy crown from drone images using segmen-

- tation. *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography*, 33(6):605–613, 2015.
- [61] Calvin Hung, Mitch Bryson, and Salah Sukkarieh. Multi-class predictive template for tree crown detection. *ISPRS journal of photogrammetry and remote sensing*, 68:170–183, 2012.
- [62] Salim Malek, Yakoub Bazi, Naif Alajlan, Haikel AlHichri, and Farid Melgani. Efficient framework for palm tree detection in uav images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(12):4692–4703, 2014.
- [63] Ramesh Kestur, Akanksha Angural, Bazila Bashir, SN Omkar, Gautham Anand, and MB Meenavathi. Tree crown detection, delineation and counting in uav remote sensed images: A neural network based spectral–spatial method. *Journal of the Indian Society of Remote Sensing*, 46(6):991–1004, 2018.
- [64] Serdar Selim, Namik Kemal Sonmez, Mesut Coslu, and Isin Onur. Semi-automatic tree detection from images of unmanned aerial vehicle using object-based image analysis method. *Journal of the Indian Society of Remote Sensing*, pages 1–8.
- [65] Benjamin Koetz, Guoqing Sun, Felix Morsdorf, KJ Ranson, Mathias Kneubühler, Klaus Itten, and Britta Allgöwer. Fusion of imaging spectrometer and lidar data over combined radiative transfer models for forest canopy characterization. *Remote Sensing of Environment*, 106(4):449–459, 2007.
- [66] Michele Dalponte, Lorenzo Bruzzone, and Damiano Gianelle. Fusion of hyperspectral and lidar remote sensing data for classification of complex forest areas. *IEEE Transactions on Geoscience and Remote Sensing*, 46(5):1416–1427, 2008.
- [67] L Naidoo, Moses A Cho, R Mathieu, and G Asner. Classification of savanna tree species, in the greater kruger national park region, by integrating hyperspectral and lidar data in a random forest data mining environment. *ISPRS journal of Photogrammetry and Remote Sensing*, 69:167–179, 2012.
- [68] Yanfeng Gu, Qingwang Wang, Xiuping Jia, and Jón Atli Benediktsson. A novel mkl model of integrating lidar data and msi for urban area classification. *IEEE transactions on geoscience and remote sensing*, 53(10):5312–5326, 2015.

- [69] Naoto Yokoya, Shinji Nakazawa, Tomohiro Matsuki, and Akira Iwasaki. Fusion of hyperspectral and lidar data for landscape visual quality assessment. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2419–2425, 2014.
- [70] Tomohiro Matsuki, Naoto Yokoya, and Akira Iwasaki. Hyperspectral tree species classification of japanese complex mixed forest with the aid of lidar data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(5):2177–2187, 2015.
- [71] Frieke MB Van Coillie, Wenzhi Liao, P Kempeneers, K Vandekerkhove, Sidharta Gautama, Wilfried Philips, and Robert R De Wulf. Optimized feature fusion of lidar and hyperspectral data for tree species mapping in closed forest canopies. In *2015 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4. IEEE, 2015.
- [72] Henning Buddenbaum, Stephan Seeling, and Joachim Hill. Fusion of full-waveform lidar and imaging spectroscopy remote sensing data for the characterization of forest stands. *International Journal of Remote Sensing*, 34(13):4511–4524, 2013.
- [73] Dirk Lemp and Uwe Weidner. Improvements of roof surface classification using hyperspectral and laser scanning data. In *Proc. ISPRS Joint Conf.: 3rd Int. Symp. Remote Sens. Data Fusion Over Urban Areas (URBAN), 5th Int. Symp. Remote Sens. Urban Areas (URS)*, pages 14–16, 2005.
- [74] Wenzhi Liao, Frieke Van Coillie, Lianru Gao, Liwei Li, Bing Zhang, and Jocelyn Chanussot. Deep learning for fusion of apex hyperspectral and full-waveform lidar remote sensing data for tree species mapping. *IEEE Access*, 6:68716–68729, 2018.
- [75] Luxia Liu, Nicholas C Coops, Neal W Aven, and Yong Pang. Mapping urban tree species using integrated airborne hyperspectral and lidar remote sensing data. *Remote Sensing of Environment*, 200:170–182, 2017.
- [76] Matheus Pinheiro Ferreira, Maciel Zortea, Daniel Capella Zanotta, Yosio Edemir Shimabukuro, and Carlos Roberto de Souza Filho. Mapping tree species in tropical seasonal semi-deciduous forests with hyperspectral and multispectral data. *Remote Sensing of Environment*, 179:66–78, 2016.

- [77] R Chary, D Rajya Lakshmi, and KVN Sunitha. Feature extraction methods for color image similarity. *arXiv preprint arXiv:1204.2336*, 2012.
- [78] Jun-Dong Chang, Shyr-Shen Yu, Hong-Hao Chen, and Chwei-Shyong Tsai. Hsv-based color texture image classification using wavelet transform and motif patterns. *Journal of Computers*, 20(4):63–69, 2010.
- [79] T Natarajan, KR Rao, and N Ahmed. Discrete cosine transform. *IEEE Trans. Computers*, 23(1):90–93, 1974.
- [80] Chi-Man Pun and Hong-Min Zhu. Textural image segmentation using discrete cosine transform. *Rn*, 1:1, 2009.
- [81] Ziad M Hafed and Martin D Levine. Face recognition using the discrete cosine transform. *International journal of computer vision*, 43(3):167–188, 2001.
- [82] Richard W Connors, Mohan M Trivedi, and Charles A Harlow. Segmentation of a high-resolution urban scene using texture operators. *Computer vision, graphics, and image processing*, 25(3):273–310, 1984.
- [83] Dhanashree Gadkari. Image quality analysis using glcm. 2004.
- [84] Fritz Albrechtsen et al. Statistical texture measures computed from gray level cooccurrence matrices. *Image processing laboratory, department of informatics, university of oslo*, 5, 2008.
- [85] Linus Närvä. Spatial features for tree-crown classification from 3-cm resolution cir imagery. 2015.
- [86] Vijay Kumar and Priyanka Gupta. Importance of statistical measures in digital image processing. *International Journal of emerging technology and advanced engineering*, 2(8):56–62, 2012.
- [87] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.
- [88] Hongliang Jin, Qingshan Liu, Hanqing Lu, and Xiaofeng Tong. Face detection using improved lbp under bayesian framework. In *Third International Conference on Image and Graphics (ICIG'04)*, pages 306–309. IEEE, 2004.

- [89] Javier Ruiz-del Solar and Julio Quinteros. Illumination compensation and normalization in eigenspace-based face recognition: A comparative study of different pre-processing approaches. *Pattern Recognition Letters*, 29(14):1966–1979, 2008.
- [90] Gang Bai, Yi Zhu, and Zongyao Ding. A hierarchical face recognition method based on local binary pattern. In *2008 congress on image and signal processing*, volume 2, pages 610–614. IEEE, 2008.
- [91] Hong Yang and Yiding Wang. A lbp-based face recognition method with hamming distance constraint. In *Fourth international conference on image and graphics (ICIG 2007)*, pages 645–649. IEEE, 2007.
- [92] Di Huang, Yunhong Wang, and Yiding Wang. A robust method for near infrared face recognition based on extended local binary pattern. In *International Symposium on Visual Computing*, pages 437–446. Springer, 2007.
- [93] Yonggang Huang, Yunhong Wang, Tieniu Tan, et al. Combining statistics of geometrical and correlative features for 3d face recognition. In *BMVC*, pages 879–888. Citeseer, 2006.
- [94] Zhenhua Guo, Lei Zhang, and David Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, 19(6):1657–1663, 2010.
- [95] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *International workshop on analysis and modeling of faces and gestures*, pages 168–182. Springer, 2007.
- [96] Timo Ahonen and Matti Pietikäinen. Soft histograms for local binary patterns. In *Proceedings of the Finnish signal processing symposium, FINSIG*, volume 5, page 1, 2007.
- [97] Shu Liao and Albert CS Chung. Face recognition by using elongated local binary patterns with average maximum distance gradient magnitude. In *Asian conference on computer vision*, pages 672–679. Springer, 2007.
- [98] Shengcai Liao, Xiangxin Zhu, Zhen Lei, Lun Zhang, and Stan Z Li. Learning multi-scale block local binary patterns for face recognition. In *International Conference on Biometrics*, pages 828–837. Springer, 2007.

- [99] Lior Wolf, Tal Hassner, and Yaniv Taigman. Descriptor based methods in the wild. In *Workshop on faces in real-life images: Detection, alignment, and recognition*, 2008.
- [100] Janis Fehr. Rotational invariant uniform local binary patterns for full 3d volume texture analysis. In *Finnish signal processing symposium (FINSIG)*, page 34, 2007.
- [101] Ludovic Paulhac, Pascal Makris, and Jean-Yves Ramel. Comparison between 2d and 3d local binary pattern methods for characterisation of three-dimensional textures. In *International Conference Image Analysis and Recognition*, pages 670–679. Springer, 2008.
- [102] Guoying Zhao and Matti Pietikainen. Experiments with facial expression recognition using spatiotemporal local binary patterns. In *2007 IEEE International Conference on Multimedia and Expo*, pages 1091–1094. IEEE, 2007.
- [103] Guoying Zhao and Matti Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):915–928, 2007.
- [104] Wenchao Zhang, Shiguang Shan, Wen Gao, Xilin Chen, and Hongming Zhang. Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 786–791. IEEE, 2005.
- [105] Lianghua He, Cairong Zou, Li Zhao, and Die Hu. An enhanced lbp feature based on facial expression recognition. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 3300–3303. IEEE, 2006.
- [106] Shiguang Shan, Wenchao Zhang, Yu Su, Xilin Chen, and Wen Gao. Ensemble of piecewise fda based on spatial histograms of local (gabor) binary patterns for face recognition. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3. IEEE, 2006.
- [107] S. Yan, H. Wang, X. Tang, and T. Huang. Exploring feature descriptors for face recognition. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 1, pages I-629–I-632, April 2007.

- [108] Xiaoyang Tan and Bill Triggs. Fusing gabor and lbp feature sets for kernel-based face recognition. In *International Workshop on Analysis and Modeling of Faces and Gestures*, pages 235–249. Springer, 2007.
- [109] Richa Singh, Mayank Vatsa, and Afzel Noore. Integrated multilevel image fusion and match score fusion of visible and infrared face images for robust face recognition. *Pattern Recognition*, 41(3):880–893, 2008.
- [110] Marko Heikkilä, Matti Pietikäinen, and Cordelia Schmid. Description of interest regions with center-symmetric local binary patterns. In *Computer vision, graphics and image processing*, pages 58–69. Springer, 2006.
- [111] Marko Heikkilä, Matti Pietikäinen, and Cordelia Schmid. Description of interest regions with local binary patterns. *Pattern Recogn.*, 42(3):425–436, March 2009.
- [112] Di Huang, Guangpeng Zhang, Mohsen Ardabilian, Yunhong Wang, and Liming Chen. 3d face recognition using distinctiveness enhanced facial representations and local feature hybrid matching. In *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–7. IEEE, 2010.
- [113] Timo Ahonen, Jiří Matas, Chu He, and Matti Pietikäinen. Rotation invariant image description with local binary pattern histogram fourier features. In *Scandinavian conference on image analysis*, pages 61–70. Springer, 2009.
- [114] Wai Kin Kong, David Zhang, and Wenxin Li. Palmprint feature extraction using 2-d gabor filters. *Pattern recognition*, 36(10):2339–2347, 2003.
- [115] Dennis Dunn and William E Higgins. Optimal gabor filters for texture segmentation. *IEEE Transactions on image processing*, 4(7):947–964, 1995.
- [116] Simona E Grigorescu, Nicolai Petkov, and Peter Kruizinga. Comparison of texture features based on gabor filters. *IEEE Transactions on Image processing*, 11(10):1160–1167, 2002.
- [117] Horst Bischof, Werner Schneider, and Axel J Pinz. Multispectral classification of landsat-images using neural networks. *IEEE transactions on Geoscience and Remote Sensing*, 30(3):482–490, 1992.

- [118] Kishan Mehrotra, Chilukuri K Mohan, and Sanjay Ranka. *Elements of artificial neural networks*. MIT press, 1997.
- [119] Russell Reed and Robert J MarksII. *Neural smithing: supervised learning in feed-forward artificial neural networks*. Mit Press, 1999.
- [120] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.
- [121] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.
- [122] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines, 2002.
- [123] Terrence S Furey, Nello Cristianini, Nigel Duffy, David W Bednarski, Michel Schummer, and David Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.
- [124] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Icml*, volume 99, pages 200–209, 1999.
- [125] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.
- [126] Farid Melgani and Lorenzo Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on geoscience and remote sensing*, 42(8):1778–1790, 2004.
- [127] Giorgos Mountrakis, Jungho Im, and Caesar Ogole. Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3):247–259, 2011.
- [128] Matthew Colgan, Claire Baldeck, Jean-Baptiste Féret, and Gregory Asner. Mapping savanna tree species at ecosystem scales using support vector machine classification

BIBLIOGRAPHY

and brdf correction on airborne hyperspectral and lidar data. *Remote Sensing*, 4(11):3462–3480, 2012.

- [129] Mariette Awad and Rahul Khanna. *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. Apress, 2015.