CSC2005Z Report

Classification of Crop Trees Images by Clustering on Pixel-Based Features

Catherine Li

University of Cape Town

Department of Computer Science

Supervised by Professor Patrick Marais

October 2024

Abstract

Crop trees have high economic and even nutritional value. In a commercial context, it is increasingly important to employ technology in its management. An important part of the management chain is to filter out irrelevant trees when analysing a particular type of crop tree. Most works in crop tree classification base their research on data delineated within complex polygons, which entails complications. Alternatively, object detection is more widely accessible with architectures such as the You Only Look Once models. These models have high accuracy in identifying objects and capturing their spatial location in a bounding box. This report aims to explore classification techniques and algorithms by using an existing dataset that captures oil palms and non-palm tree species within bounding boxes. Feature extractors such as the Histogram of Oriented Gradients, Local Binary Pattern, Grey level Co-occurrence Matrix and Greyscale Histogram are applied to each tree instance to derive feature vectors. These feature vectors are inputted into three clustering algorithms to classify the palm and the non-palm trees. The clustering algorithm investigated includes K-Means, DBSCAN and Spectral clustering. Additional investigations include the effect of z-normalisation on images and PCA, which aims to reduce the feature vector dimensionality to assist the clustering algorithms in forming meaningful clusters. The experiments reveal K-Means and Local Binary Pattern as the best-performing clustering algorithm and feature extractor, respectively. Moreover, z-normalisation significantly enhanced clustering results, while PCA improved clustering run time and marginally optimised the F1 score achieved by clustering algorithms.

1. Introduction

Crop trees not only contribute to regulating climate but also are a critical source of economic contribution. When managing commercial crop tree farms, tree surveying is important for ensuring production quality and crop yield prediction. Examples of specific tasks include counting, locating and analysing the state of the tree. However, these tasks are labour-intensive, time-consuming and vulnerable to human error if done manually [1]; Integrating technology can help achieve more efficient, reliable and timely results. The technology used for tree surveying can be split into three categories: data collection, data processing and data analysis. This report focuses on classification in the data processing stage.

Data processing transforms raw data into information, which enables subsequent analysis. Common tasks include object detection, object segmentation and object classification. Object detection and object segmentation are both methods of extracting trees. Object detection includes identifying entities and encapsulating them in a bounding box. Object segmentation entails the delineation of an object on an image. This task entails describing the shape and position of the object with polygons in a nuanced manner. In comparison, object classification involves categorising the objects identified. Clustering algorithms are unsupervised learning algorithms that can be used for classification. In crop tree management, classification helps filter out unwanted objects for more accurate downstream analysis.

Popular classification approaches include the object-based approach and the pixel-based approach [2]. The pixel-based approach uses information derived from image pixels to categorise objects. Feature extractors are algorithms used to extract information in the form of feature vectors from the raw image pixels. They help classification algorithms interpret the image in a more meaningful manner. The pixel-based approach also requires basic object detection so the feature vector derived is localised to the object of interest. Alternatively, many papers have been published on object-based classification where trees are identified based on information extracted from segmented trees, like shape characteristics [2]. Often, these results assume perfect accuracy in segmentation, which is still a challenge due to the irregularity of a tree's shape [3].

1.1 Problem Statement

This report aims to explore classification based on a tree's pixel-based features, where the trees are localized within a bounding box and not delineated. These feature extractors will output feature vectors that describe aspects of the raw image data such as texture or its pixel distribution for z-normalised and unnormalized images. The feature vectors will be fed into clustering algorithms that define clusters through different similarity and dissimilarity criteria. However, some of these feature vectors exhibit high dimensionality, which can render clustering algorithms ineffective. Therefore, the effect of dimensionality reduction through Principle Component Analysis (PCA) will also be explored in conjunction with the clustering algorithms, normalisation and pixel-based feature extractors.

The experiments used an existing dataset consisting of images taken with unmanned aerial vehicles (UAV). The images contain palm oil trees and trees of other species that are demarcated in a bounding box and given a ground truth label.

The research aims to answer the following questions:

1. To what extent does normalisation improve the results obtained from clustering?
2. To what extent does PCA reduce the quality of feature-based classification for the chosen clustering methods?
3. Which individual feature extractors and clustering algorithms yield the best performance overall?
4. Which combination of features and clustering algorithms yield the best performance?

The remainder of this report is laid out as follows. Section 2 covers the relevant background for this report. Section 3 details the methodologies used during the experiments and, if applicable, how their parameters were chosen. Section 4 covers the experiments by describing the dataset, experimental setup, results and findings. The conclusion, future work and bibliography are laid out in sections 6, 7 and 8 respectively. Appendixes are also included at the end for additional experiment results not covered in the experiments section.

2. Background

The following section describes definitions and related works with regards to tree extraction techniques, the feature extractors used, principle component analysis, the clustering algorithms applied and the evaluation metrics employed.

2.1 Methods to Extract Individual Trees in Plantations

Trees can be extracted via object detection and object segmentation.

Object detection focuses on identifying a specific object within a localised region. There are many deep learning architectures dedicated to object detection, such as convolutional neural network (CNN), regions with convolutional neural network (R-CNN), faster R-CNN and the many versions of the You Only Look Once (YOLO) detectors. These detection models can identify trees in orchards and encapsulate the tree in a bounding box. One study proposed an optimised YOLO-v4 method to detect bayberry trees for counting tasks that yield up to 97.45% accuracy and 98.16% recall [4].

Compared to object detection, object segmentation involves precise localisation of objects by drawing complex polygons to describe their shape. Recently, the above-mentioned deep learning architectures employed for object detection have also been utilised for tree crown delineation. In a tree detection and delineation review, CNN models are found to outperformed other methods [5]. Nonetheless, these results depend on data efficiency, model choice and training strategy [3]. Moreover, trees are naturally irregular in shape, which makes segmentation a challenge despite recent developments [3].

2.2 Features to Describe Images

After extracting the trees from the image, feature extractors can be applied to obtain more useful information. The information can be local descriptors that focus on extracting local regions on an image or global features extracted from the entire image. Local descriptors include Scale-Invariant Feature Transform (SIFT), Daisy, and Oriented FAST and Rotated BRIEF (ORB), where feature extraction is centred around key points only [6]. In contrast, this report explores global descriptors, such as Grey Level Co-occurrence Matrix (GLCM), Histogram of Oriented Gradients (HOG), Local Binary Pattern (LBP) and greyscale histograms [6].

GLCM [7] analyses the spatial distribution of pixels in a greyscale image and is often used in texture analysis. It records the frequency for which each pair of pixel values appear together at the specified offset and direction of the pixels. The frequencies are stored in a two-dimensional NxN matrix, where N is the number of grey levels. The resulting co-occurrence matrix can be used to calculate texture measures that fall into three strongly associated categories: contrast-based, which includes contract, dissimilarity and homogeneity; orderliness-based, which includes angular second moment (ASM) and entropy; as well as statically based such as mean, variance and correlation [8].

HOG [9] describes the structure of an image by calculating the gradient of pixel intensity within a cell. The cells have a predetermined size. For each cell, there is a histogram of gradient orientations that reflects the distribution of edge directions within the cell. The final feature vector is the combination of all the histograms derived from each cell and it captures edges as well as contours (refer to Figure 1).

LBP [10, 11, 12] is based on local spatial patterns and greyscale contrasts to formulate feature vectors. It is often used in texture analysis. The LBP operator thresholds a neighbourhood of pixels with the centre value to assign binary labels to each surrounding pixel and results in a binary number. The histogram of all the binary labels can then be used as a feature descriptor. Additionally, it is robust to lighting discrepancies and noise [13].

Finally, greyscale histogram organises different pixel intensities into their respective bins and are usually normalised into probability distributions to make them more comparable across different images. Coloured histograms can be effective in distinguishing colour textures in ideal conditions such as lighting consistency [14]. However, due to sensitivity of clustering algorithms to high dimensional features, discriminating greyscale textures may be more appropriate.

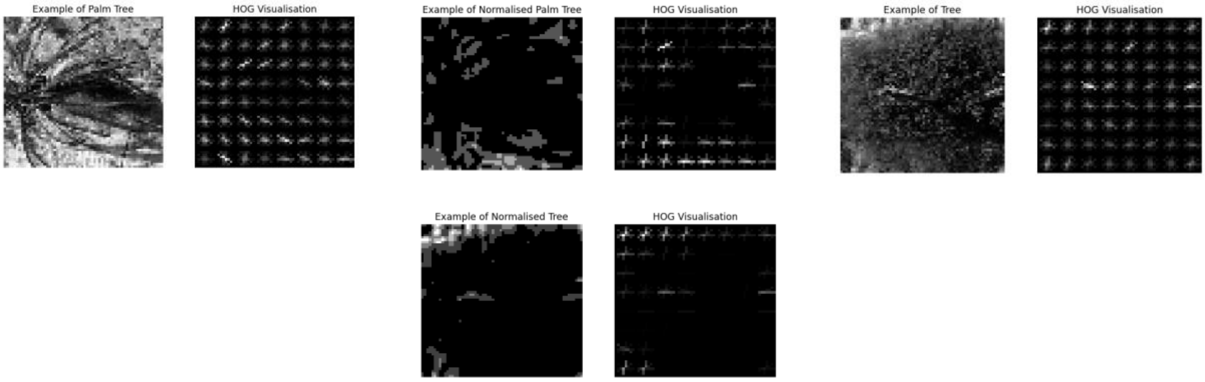

Figure 1 Visual Examples of HOG Description for Different Tree Instances

2.3 Clustering

Clustering algorithms are considered the most important question in unsupervised learning. It deals with partitioning unknown areas to reveal the data structure [15]. A widely accepted clustering definition is that instances in the same cluster must be the most similar, instances in different clusters must be the most different, and the measures for similarity as well as differences must be practically meaningful and clear [16]. The following clustering algorithms are chosen based on their vastly different strengths and clustering approaches.

K-Means [17] is a simplistic, partition-based clustering algorithm and is widely used to classify pixels [2]. Hence, it is also used as benchmark. This type of clustering algorithm assumes the centre of data points as the centre of the corresponding clusters. K-Means clustering is an iterative process where the centre of clusters, i.e. the centre of data points, are updated until the criteria for convergence is met. It generally has high computational efficiency and a relatively low time complexity. However, it is not a robust algorithm because it is sensitive to outliers. Furthermore, it is only suitable for convex data and clustering results are dependent on the preset number of clusters [15].

On the other hand, DBSCAN [18] is a density-based clustering algorithm, where regions with high data density are classified into the same cluster. This algorithm is efficient and applies to arbitrarily shaped data [15]. Additionally, the algorithm can also demarcate noisy data that do not fit under the classification criteria. The algorithm doesn’t perform well when data density is uneven [15]. Furthermore, it is notoriously sensitive to its two parameters, the radius of the neighbourhood, epsilon, and the minimum number of points in the neighbourhood.

Spectral clustering is based on spectral graph theory. First, a similarity matrix is constructed where each data point is seen as a vertex and the similarities amongst each vertex become a weighted edge. From there, a Laplacian matrix is calculated to reflect the data structure. Then, eigenvalue decomposition is performed on the Laplacian matrix to reduce the dimensionality of the data and make clustering more effective. The data is projected into a new space and clustered using clustering algorithms such as K-Means. Spectral clustering can handle high-dimensional data at a large scale [19]. Disadvantages include higher computational complexity and unclear construction of the similarity matrix.

2.4 PCA

A critical problem that clustering algorithms experience is the curse of dimensionality. In this context, as the dimensionality of the feature vectors increases, the differences and similarities between each data point become less meaningful. PCA is commonly used to reduce dimensionality to improve computational efficiency and clustering performance. It achieves this by applying a vector space transformation that only stores the most significant values from the original vector **[20]**. The**se** significant values are returned **in** a reduced vector **containing** Principle Components (PC). Though PCA can reduce dimensionality, **the** information **lost increases as** the number of PCs included in the final vector decreases. Hence choosing the amount of PCs to keep is an important consideration.

3. Methodology

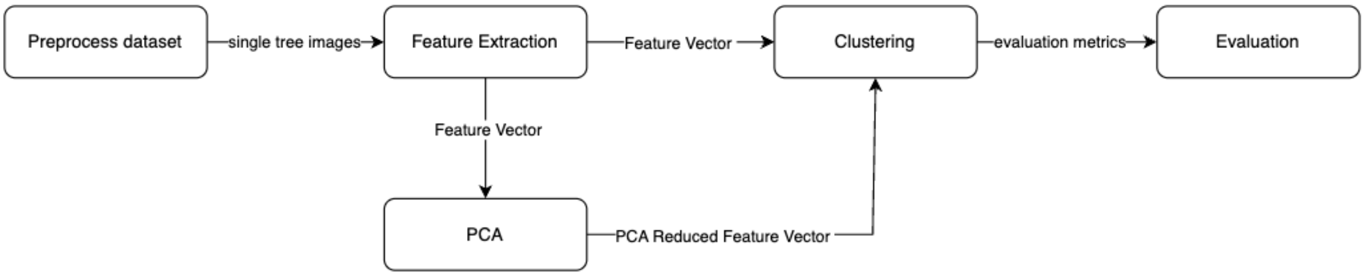

Figure 2 Architecture of the Research Process

The research will follow the flow depicted in Figure 2, where each step is dependent on the previous. First, the dataset is pre-processed by converting images to greyscale, applying histogram equalisation, extracting the individual trees, resizing the individual trees to a uniform shape, and applying normalisation to create an additional dataset of tree images. The preprocessing is meant to prepare the images for better feature extraction, allow comparison between normalised and unnormalized data, as well as make the experiment more valid. Then, the feature extractors are applied to each tree image to create the feature vectors on which the clustering algorithms will operate. PCA is applied to the feature vectors to create an additional set of feature vectors with reduced dimensionality (where its effect will be examined). Each experiment with a unique combination of normalisation application, feature extractor, clustering algorithm, and PCA application will be given a precision score, a recall score and an F1 score to evaluate performance.

3.1 Preprocessing

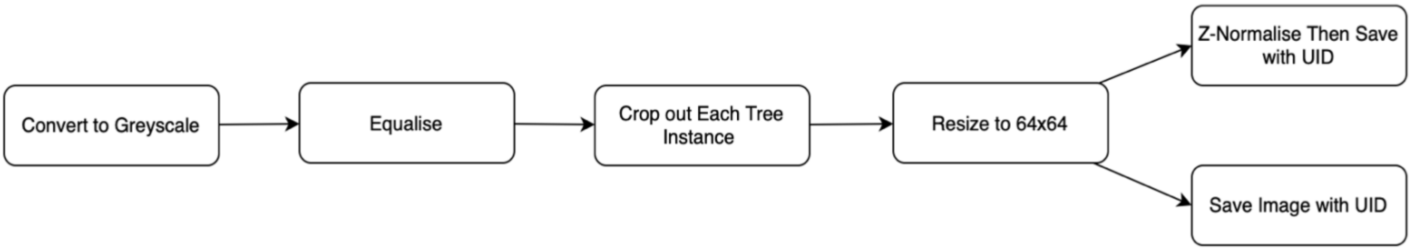

Figure 3 Task Flow for Dataset Preprocessing

The photographs were first converted from RGB to a greyscale image using OpenCV because LBP, GLCM and the greyscale histogram expect to operate on a greyscale image.

The grey image is then equalised using OpenCV's histogram equalisation. This step will improve the global contrast in the image. It enhances the different features of the image that may have been previously hard to detect, which aids the feature extraction process later [21].

Next, the bounding box coordinates of each tree instance are extracted and used to crop out the trees. The extracted single trees are resized to a 64x64 pixel image for consistency and comparability purposes in the evaluation. These cropped images are saved with a unique identifier (UID) that includes their ground truth label.

Additionally, z-normalisation is applied to the cropped images and saved separately with a UID that also includes the ground truth label. This step aims to standardise the different scales of the image to make the images more comparable and achieve better generalisability of the results.

It is important to include the ground truth label so that numerical ground truth sets can be derived later on to evaluate results.

3.2 Feature Extraction

Scikit Image's GLCM, LBP and HOG implementation was used and the greyscale intensities were summarised on a histogram using NumPy. The feature extractors are applied both to the normalised and unnormalized data. For each feature extractor, the resulting sample of feature vectors was first stored in a dictionary that mapped it to its respective UID. Then the dictionary was saved **as** a .npy **file** to be accessed in the clustering stage.

GLCM is versatile, concise and efficient in analysing texture and was revolutionary for greyscale analysis methods [22]. Thus, it is investigated in this report. It needs the parameters distance and angles to be preset. Distance represents the offset between pixels and angles represent the angle, in radians, of the pixels in the pair. One study highlights that an offset of 1 is most frequently used [8]. Hence, the distance will be set to 1 and 2. The angles are set at 0, 0.785398 and 1.5708 radians, representing 0, 45 and 90 degrees. The number of grey levels counted is set to 258, because the pre-processed dataset contains 8-bit greyscale images. The GLCM from each distance and angle pair is also normalised into a probability distribution, where the sum of all the elements in the matrix is one. For each combination of distance and angles, dissimilarity, contrast, homogeneity, ASM, energy and correlation were computed and appended to the feature vector of the same image.

LBP is chosen because it has low computational demand, a simplistic implementation and excels in differentiating textures [22]. It requires the parameters P, the number of circularly symmetric neighbouring pixels to be assigned a binary code, and R, the radius of the circle, to be preset. P and R are set to 8 and 3 respectively. The method is set to 'uniform' [11] so the LBP image returned is rotation and greyscale invariant with a finer quantization of the angular space. The LBP image is then turned into a discrete probability distribution with probabilities defined from integers 0 to 10.

HOG has been revealed to capture the unique texture of palm trees well [23] and will be appropriate for this study. The HOG in the Scikit Image has a few optional parameters including pixels_per_cell, orientation bins, cells_per_block and block normalisation. For good results, HOGs need fine-scaled gradients, fine orientation voting, coarser spatial quantization and good local contrast normalisation [9]. Hence, the number of orientation bins is set to 9, the pixels_per_cell is set to 8x8 for a coarse spatial quantization, the number of cells per block to 3x3 and block normalisation to L1-normalisation. A 1D HOG descriptor for the image is returned and stored.

Greyscale histograms reflects the intensity distribution of the images, which is appropriate for the report's research. To obtain the histogram, the greyscale image is simply passed to NumPy's histogram function. The number of bins is set to 32 and the density is set to true with the range set to the image's min and max+1 pixel intensity. The 1D array returned for the histogram is turned into a probability distribution for ease of comparability.

3.3 PCA

Scikit Learn was used to perform PCA. The amount of PCs to keep will be determined by applying PCA to the training dataset first (refer to section 4.1). A Scree plot of components to keep versus explained variance is plotted. From the plots, it is evident that 99% of the variance can be explained by less than ten components for most of the feature vectors, where only the HOG, normalised HOG and greyscale histogram vectors are the exceptions. Hence, an explained variance of 0.99 is chosen to be applied to all the feature vectors. Refer to Table 1 for the amount components within the feature before and after PCA projection.

Table 1
The amount of components within a feature vector before and after PCA projection

NUMBER OF PCS	GLCM	GLCM NORMALISED	LBP	LBP NORMALISED	HOG	HOG NORMALISED	HISTOGRAM	HISTOGRAM NORMALISED
BEFORE PCA	36	36	10	10	2916	2916	32	32
AFTER PCA	3	5	9	6	574	979	6	18

Additionally, PCA will be used to reduce the normalised and unnormalized datasets to three PCs. This will be used for visual depiction of the data's density and analysed in conjunction with DBSCAN results.

3.4 Clustering

K-Means and Spectral clustering requires the amount of clusters formed in the end (n_clusters) to be defined. The number of clusters is already known to be two. Hence, n_clusters is set to two. Furthermore, the two clustering algorithms initialise with a random centroid seed. Thus, clustering results can be non-deterministic. To optimise results, the amount of consecutive runs (n_init) is set to ten in both cases, where the returned predicted labels have the best inertia score of the ten consecutive runs.

DBSCAN is notorious for the difficulty in finding the right epsilon and minimum points per sample to use as its parameter. Different parameters will be tested on the training dataset (refer to section 4.1) and extrapolated to the entire dataset. To narrow down the range of epsilon being tested, a kth-nearest neighbour (k-NN) plot is used, where k is always set to one less than the minimum point being tested.

The minimum point is influenced by sample size. If the minimum point for the sample is set too high, meaningful clusters may be ignored; if it is set too low, the algorithm may form smaller insignificant clusters. Therefore, a percentage of the dataset will be chosen for extrapolation instead of a fixed value. From the ground truth dataset, it is found that the overall dataset and the portion of the dataset used for training is roughly 14.7% and 12.2% non-palm tree, respectively. As a result, the values tested include 10%, 5%, 2.5% and 1% of dataset size, which is truncated to an integer. These percentages will be tested at different epsilon values at a range and scale similar to the y-axis of the k-NN plot. At 10% and 5%, the results only reflected one cluster and data points classified as noise. Out of all the values tested, the minimum distance that yields the highest F1 score is 2.5% when the unique labels are only zero and one, except for noise classified as -1. Therefore, the minimum point is set to 2.5%, which is 326, for all the DBSCAN experiments conducted later.

The epsilons are chosen based on the highest F1 score that only contains the unique labels zero and one, except for noise classified as -1. The results are as follows:

Table 2
The different epsilon values used for each variation of feature vector during DBSCAN clustering

EPSILON	GLCM	GLCM NORMALISED	LBP	LBP NORMALISED	HOG	HOG NORMALISED	HISTOGRAM	HISTOGRAM NORMALISED
WITHOUT PCA	294	0.076	0.01	0.00975	3.259	4.81	0.042	1.34
PCA	270	0.074	0.01	6	3.243	4.78	461	78

3.5 Evaluation Metrics

The quality of the clustering is evaluated using the F1 score found in the Scikit Learn library. The F1 score will also be further scrutinised using the precision and recall scores metric found in the same library. The three metrics give a value between 0 and 1, where 1 is a perfect score.

The precision score reflects the proportion of the trees predicted as palms that were correct. It can be calculated using the following formula:

$$\textbf{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall describes the proportion of true palms that the clustering algorithm correctly labelled. It can be calculated using the following formula:

$$\textbf{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

The F1 score is a metric that provides an evaluation that balances precision and recall. It can be seen as the harmonic mean between the two. It is calculated as follows:

$$\textbf{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Though the above three metrics are mostly used to evaluate binary classification, the experiments conducted in this report are interested in multiclass classification. Thus, a weighted F1 score is chosen because it not only assesses the accuracy of predicted labels but accounts for the unbalanced nature of the dataset, where there are significantly more palm trees than non-palm trees. Scikit Learn calculates the weighted score by taking the weighted average of the palm's and the non-palm's F1 score. Hence, for all three scores the parameters 'labels' are set to 0,1, and 'average' is set to 'weighted'. The calculations are as follows:

$$\textbf{Weighted F1} = \sum_{i=0}^{1} \left(\text{F1 for label } i \times \frac{\text{Number of true instances of label } i}{\text{Total number of instances}} \right)$$

The experiments were conducted in Visual Studio Code's Jupyter notebooks and libraries such as Scikit Learn, Scikit Image, NumPy, and OpenCV were extensively used.

4. Experiments

4.1 The Dataset

The experiments are based on an oil palm tree dataset with 349 images, where 258 are taken from the Kharj region and 91 are taken from the Prince Sultan University campus. Within the dataset, there are 13 067 instances of trees that were manually labelled in a bounding box using Labelbox. Each tree's annotation is stored in both CSV and .xml files along with a ground truth label of either 'palm' or 'tree'. The palm trees have distinctly jagged leaves and appear as a muted cool-toned green(refer to Figures 4 and 5). In contrast, the other tree species appear more vibrant and warm in colour and the texture appears wispier and spotty (refer to Figure 4). After applying feature extractors and normalisation, the data distribution is roughly visually described in Figure 6.

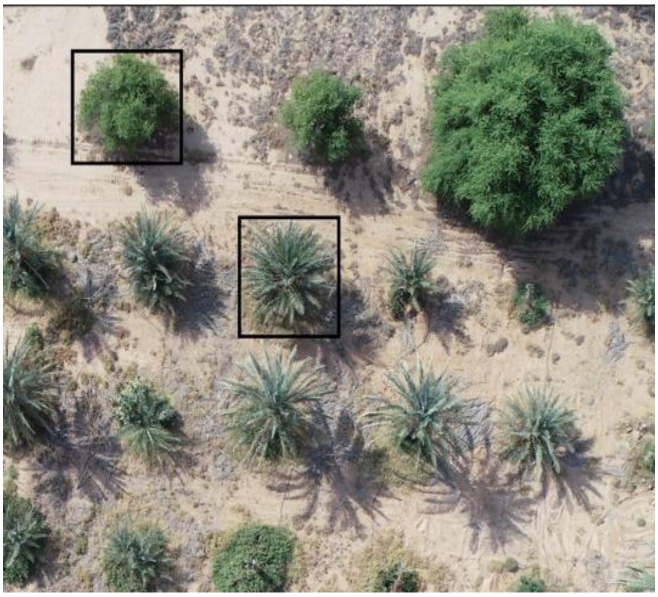

Figure 4 Palm Tree and Other Tree from Kharj Region with Examples of the Bounding Box

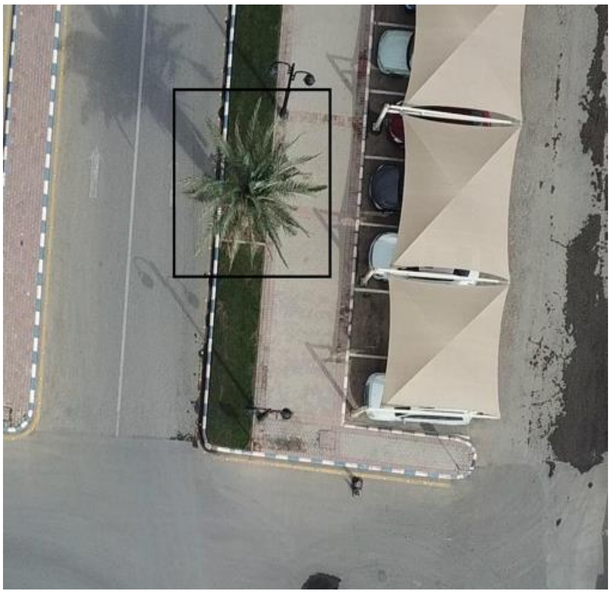

Figure 5 Palm Tree from Prince Sultan University Campus with an Example of the Bounding Box

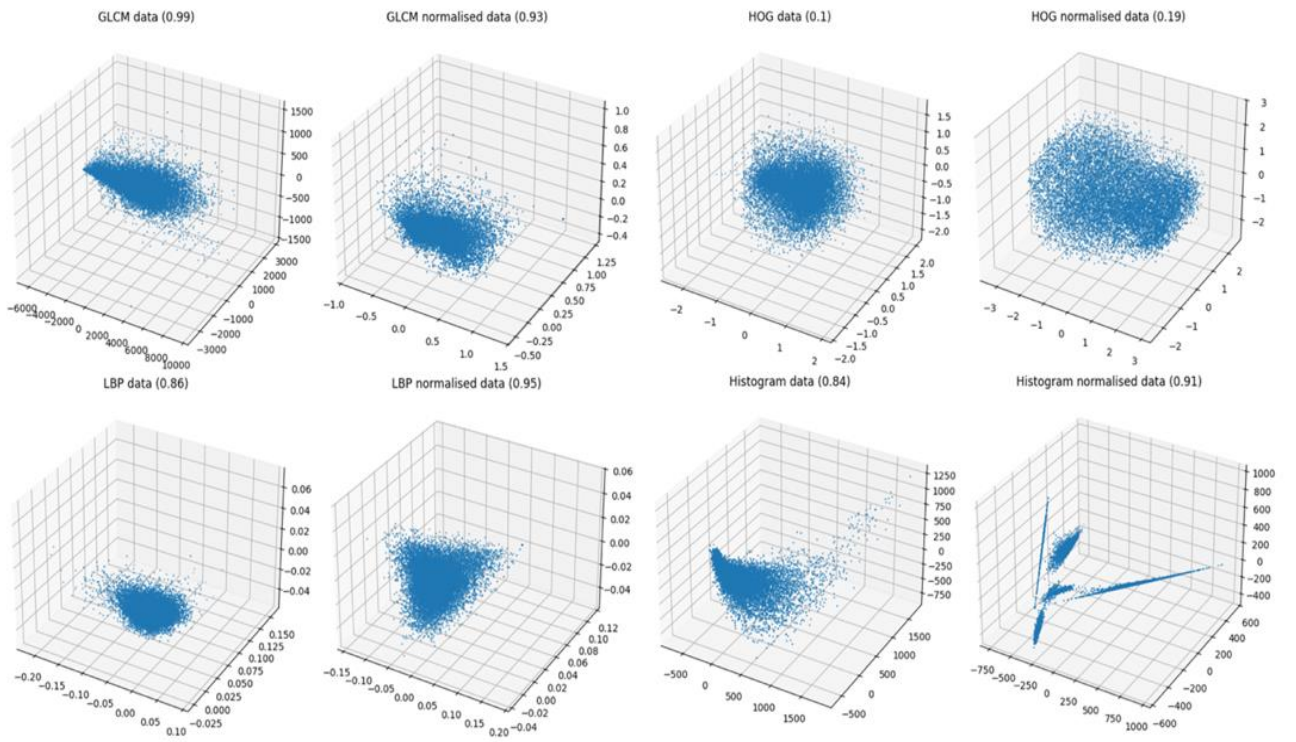

Figure 6 Graphical Depiction of the Distribution of Data Samples where Its Respective Feature Vector is PCA Reduced to 3 PCs. The number next to the plot title represents the total explained variance ratio of the 3 PCs.

4.2 The Experimental Setup

There are 48 experiments to run, where each experiment is a unique combination of whether or not the data was normalised, one of the four feature vectors, whether the dimensionality of feature vectors was reduced by PCA and one of three clustering algorithms. All the experiments will run on the entire dataset and results will all be rounded to four decimal places.

Although 2669 of the 13067 instances were used in finding appropriate PCA projections as well as DBSCAN's parameters, the risk of data leakage is small. This can be attributed to the nature of the algorithms, where both algorithms are meant for unsupervised learning, which do not rely on training sets to find parameters. In other words, both algorithms focus on the structure of the data, instead of relying on labelled data, to produce results.

DBSCAN will only be run once and the results will be recorded because the algorithm is deterministic. For K-Means and Spectral clustering, the results fluctuate due to the random initialisation. Therefore, experiments involving K-Means and Spectral clustering will be run 10 times. The highest modal F1 score will be taken as the final performance score. The mode is chosen, because clustering algorithms are meant to run under unsupervised circumstances where there the ground truth is not a given. In that scenario, it would be impossible to decipher the highest scoring set of predicted labels in terms of the F1 metric. If the highest F1 score differs from the mode, it will be noted in the results section and displayed in the appendix. The experiments performed as follows:

Table 3
Summary of the modal precision, recall and F1 scores obtained from running K-Means clustering experiments ten consecutive times. The frequency of the modal value is noted and starred values indicate that a higher F1 score was achieved (refer to Appendix A).

RESULTS		**K-MEANS**				**K-MEANS + PCA**		
METRICS	Precision	Recall	**F1**	Frequency (out of 10)	Precision	Recall	**F1**	Frequency (out of 10)
GLCM	0.6269	0.3576	**0.4455***	7	0.8592	0.6415	**0.6937**	5
NORMALISED GLCM	0.8312	0.6894	**0.7328**	3	0.8313	0.6893	**0.7327**	4

LBP	0.7368	0.4064	**0.4759***	2	0.7367	0.4061	**0.4756***	4
NORMALISED LBP	0.8218	0.6743	**0.7201***	3	0.8218	0.6744	**0.7202**	4
HOG	0.5387	0.2496	**0.3398***	2	0.5387	0.2496	**0.3399***	3
NORMALISED HOG	0.8097	0.5458	**0.6094**	4	0.8097	0.5460	**0.6095**	2
HISTOGRAM	0.5626	0.2250	**0.2798***	6	0.5911	0.2702	**0.3447***	4
NORMALISED HISTOGRAM	0.8439	0.6891	**0.7334**	5	0.6322	0.3422	**0.4267***	8

Table 4

Summary of the modal precision, recall and F1 scores obtained from running Spectral clustering experiments ten consecutive times. The frequency of the modal value is noted and starred values indicate that a higher F1 score was achieved (refer to Appendix B).

RESULTS			**SPECTRAL**				**SPECTRAL + PCA**	
METRICS	Precision	Recall	**F1**	Frequency (out of 10)	Precision	Recall	**F1**	Frequency (out of 10)
GLCM	0.0216	0.1469	**0.0376**	10	0.0216	0.1469	**0.0376**	10
NORMALISED GLCM	0.6094	0.2757	**0.3446***	7	0.6094	0.2757	**0.3446***	8
LBP	0.7575	0.5906	**0.6490***	4	0.7380	0.4094	**0.4792***	3
NORMALISED LBP	0.8209	0.6735	**0.7194***	3	0.8209	0.6735	**0.7194***	2
HOG	0.0216	0.1467	**0.0376**	10	0.0213	0.1446	**0.0371**	10
NORMALISED HOG	0.6511	0.3294	**0.4048***	6	0.6528	0.3321	**0.4077**	10
HISTOGRAM	0.5639	0.2258	**0.2805***	4	0.7447	0.8485	**0.7844**	1
NORMALISED HISTOGRAM	0.8747	0.1484	**0.0406**	10	-	-	**-**	-

Table 5

Summary of the precision, recall and F1 scores obtained from running Spectral DBSCAN. The proportion of the data classified as noisy, and the unique labels predicted by DBSCAN are listed.

RESULTS			**DBSCAN**					**DBSCAN + PCA**		
METRICS	Precision	Recall	F1	% Noisy data	Unique labels	Precision	Recall	F1	% Noisy Samples	Unique labels
GLCM	0.8774	0.0882	0.1562	85.92	-1 0 1 2	0.5588	0.0299	0.0496	79.41	-1 0 1
NORMALISED GLCM	0	0	0	100	-1	0.00090144	0.0002	0.0003	97.51	-1 0
LBP	0	0	0	100	-1	0	0	0	100	-1
NORMALISED LBP	0.0216	0.1469	0.0376	97.45	-1 0	0.0216	0.1469	0.0376	0	0
HOG	0.0007	0.0015	0.0009	67.51	-1 0	0.0006	0.0011	0.0008	69.86	-1 0
NORMALISED HOG	0.4135	0.0585	0.0605	47.05	-1 0 1	0.4205	0.0588	0.0606	47.07	-1 0 1
HISTOGRAM	0.0216	0.1466	0.0376	0.10	-1 0	0.0208	0.1400	0.0362	0.93	-1 0
NORMALISED HISTOGRAM	0.8130	0.1415	0.1216	0.32	-1 0 1 2	0.8341	0.5693	0.6713	15.94	-1 0 1 2

4.3 Results and Discussion

This section will first discuss the average performance according to the following aspects: normalisation's effect, PCA's effect, performance of the feature extractors, and performance of the clustering algorithms. This will be followed by a granular display and discussion of the results.

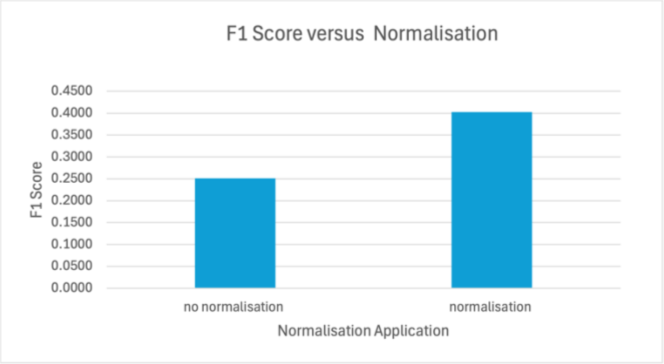

Figure 7 Graph Depicting Effect of Normalisation on Clustering Results

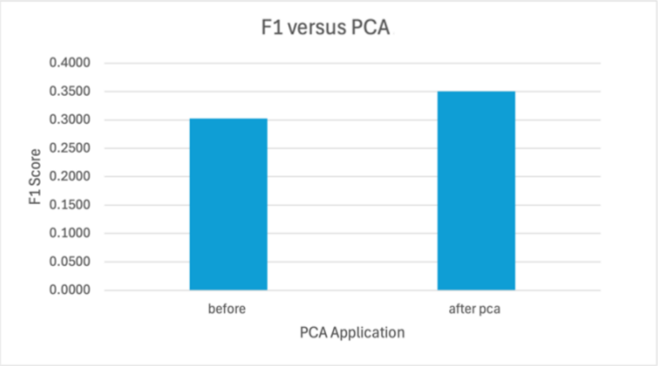

Figure 8 Graph Depicting Effect of PCA on Clustering Results

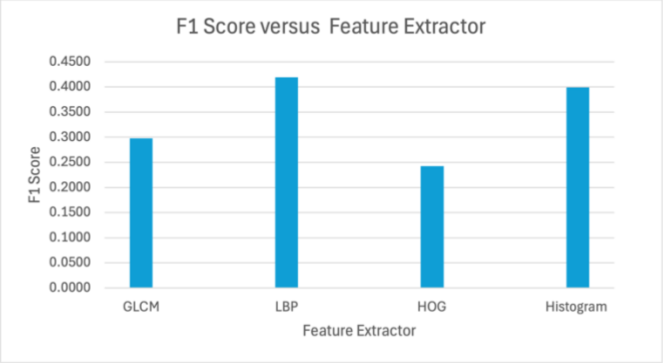

Figure 9 Graph Depicting Summary of Feature Extractor Results

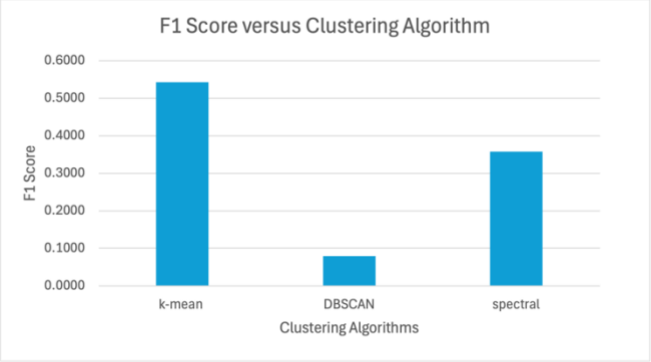

Figure 10 Graph Depicting Summary of Clustering Algorithm Results

Effect of Normalisation

The z-normalisation applied to the greyscale images before feature extraction significantly improved clustering quality by roughly 15% (Figure 7). This result aligns with the established importance of normalisation in the field of computer vision.

Effect of PCA

PCA has roughly improved the F1 score of predicted labels by 5% (Figure 8). One possible explanation is that the total explained variance ratio for the feature vectors was 99%. PCA has not significantly degraded the information available to the clustering algorithms and by reducing the dimensions of features, similarity and dissimilarity measures calculated by the clustering algorithm became more meaningful in cluster formation.

Feature Extractor Performance

In Figure 9, LBP feature possesses the highest F1 score, and the histogram feature is closely ranked second. Notably, both features demonstrate rotation invariance. It is plausible that rotation invariance contributed to its performance. HOGs performed the worst out of the four feature extractors. It is also the feature that is both sensitive to object orientation (refer to Figure 1) and has the highest dimensionality, which may explain its poor performance.

Clustering Performance

In Figure 10, K-Means clustering performed the best at an F1 score of 0.5425, which is 1.5 times the score of Spectral clustering. DBSCAN's sub-optimal performance aligns with expectations after examining the data's density distribution in Figure 6. All the feature vectors, except for the normalised histogram, appear as one big cluster of data points. There are no distinct regions of high and low density, which DBSCAN requires to define its clusters.

Due to K-Mean's simplicity, the clustering algorithm also serves as a benchmark for the other clustering algorithms. Neither Spectral nor DBSCAN clustering exhibited superior performance in comparison to K-Means. Consequently, they are deemed unsuitable for the characteristics of this data.

Overall Results

According to Tables 3, 4, and 5, Spectral clustering that classified dimension-reduced histogram feature vectors obtained the highest accuracy at 0.7844. The normalized histogram fed into K-Means clustering achieved the second-best overall result. Although 15 out of the 16 DBSCAN experiments achieved less than 0.16 in their F1 scores, the best-performing DBSCAN algorithm was able to achieve 0.6713.

Additionally, DBSCAN provided consistent results, but the parameter tuning was time-consuming. Deciding on its two parameters, minimum points and epsilon, presented a considerable challenge, because they had significant mutual influence on each other. Furthermore, most results scored below 0.1 for the F1 metric. The algorithm struggled to find meaningful clusters as there were several instances where all the data points were classified as noisy. Alternatively, only 1 cluster along with noisy data was identified, which aligns with the visual representation of data density in Figure 3.

Though this report does not focus on analysing the time complexities of the algorithms, it is worth noting that Spectral clustering took the longest to run. All the experiments for DBSCAN (excluding parameter tuning) and K-Means were completed within ten minutes, while Spectral Clustering required hours of run time. This may be attributed to the fact that Spectral clustering first forms a graph from the data and subsequently performs several mathematical calculations before clustering. Furthermore, Spectral Clustering fails to complete any runs with the dimension-reduced normalised histogram even after ten hours.

Moreover, only 9 of the 48 experiments had a higher recall score than their respective precision score, which is summarised in Table 6. This implies that most of the experiments yielded fewer incorrect predictions for palm and non-palms, but are more inclined to miss true instances of each class. All K-Means experiments exhibited this phenomenon. In the context of crop tree surveying, this outcome is preferable to the alternative. If the targeted crop trees are taken into further analysis, having a smaller sample with higher precision results in more accurate information that can be extrapolated to the population.

Table 6
Breakdown of all the experiments that had a higher recall score than precision score

NUMBER OF INSTANCES	GLCM	HOG	NORMALISED LBP	HISTOGRAM	TOTAL
SPECTRAL	0	2	2	2	**6**
DBSCAN	2	2	0	1	**5**
TOTAL	**2**	**4**	**2**	**3**	**9**

5. Conclusions

In essence, this report investigated three clustering algorithms and four feature extractors in conjunction with the effect of normalisation and PCA. K-Means clustering emerged as the most suitable out of the three for classifying palm trees because it yielded the highest F1 score and is more conservative in its predictions (i.e. more likely to predict a false negative than a false positive). It is plausible to suggest that LBP and the greyscale histogram outperform the other feature extractors in this scenario due to their invariance to the image's rotation. Normalisation has improved performance significantly and PCA slightly optimised clustering results. Nonetheless, the highest F1 score achieved, which was by applying Spectral clustering on the PCA-reduced histogram feature vector, was still less than 80%. These results are mediocre compared to other classification methods used in remote sensing crop tree identification can yield up to 99% accuracy [3].

6. Future Work

There are many limitations of this experiment. In the future, clustering algorithms should be chosen after charactering the dataset and not from inference based on the strength of the algorithm. Since the current dataset was taken with two vastly different datasets, one can also explore how having a uniform background could potentially improve clustering results. This report only briefly touches on the time and resource implications of the various algorithms, which is an important consideration when dealing with large-scale data. Different parameters for feature extractors and clustering algorithms can be investigated more rigorously in the future to see if results improve.

Despite the subpar results, there are many other aspects of the topic that can be investigated to improve performance and to deepen the understanding of pixel-based clustering. Multiple normalisation and equalization techniques can be compared and analysed for result optimisation. The influence of PCA at different explained variance ratios can be compared and the trade-off between runtime efficiency and information retention can be scrutinized in more detail. The extent in which different combination of feature extractors improve or deteriorate results could be investigated. The report only provided a rough overview of these algorithms and techniques.

7. Acknowledgement

Many thanks to Professor Patrick Marais for assisting in the research process and answering the many emails this author sent.

8. Bibliography

[1] Hela Jemaa, Wassim Bouachir, B. Leblon, A. LaRocque, Ata Haddadi, and Nizar Bouguila, “UAV-Based Computer Vision System for Orchard Apple Tree Detection and Health Assessment,” *Remote Sensing*, vol. 15, no. 14, pp. 3558–3558, Jul. 2023, doi: https://doi.org/10.3390/rs15143558.

[2] N. Zerrouki and D. Bouchaffra, “Pixel-based or Object-based: Which Approach Is More Appropriate for Remote Sensing Image classification?,” in *014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2014, pp. 864–869. doi: https://doi.org/10.1109/SMC.2014.6974020.

[3] Asli Ozdarici-Ok and Ali Özgün Ok, “Using remote sensing to identify individual tree species in orchards: A review,” *Scientia Horticulturae*, vol. 321, no. 112333, pp. 112333–112333, Nov. 2023, doi: https://doi.org/10.1016/j.scienta.2023.112333.

[4] Y. Chen, H. Xu, X. Zhang, P. Gao, Z. Xu, and X. Huang, “An Object Detection Method for Bayberry Trees Based on an Improved YOLO Algorithm,” *International Journal of Digital Earth*, vol. 16, no. 1, pp. 781–805, Mar. 2023, doi: https://doi.org/10.1080/17538947.2023.2173318.

[5] H. Zhao, J. Morgenroth, G. Pearse, and J. Schindler, “A Systematic Review of Individual Tree Crown Detection and Delineation with Convolutional Neural Networks (CNN),” *Current Forestry Reports*, vol. 9, pp. 149–170, Apr. 2023, doi: https://doi.org/10.1007/s40725-023-00184-3.

[6] Chao Zhu, Charles-Edmond Bichot, and Liming Chen, “Visual Object Recognition Using DAISY Descriptor,” in *IEEE International Conference on Multimedia and Expo*, Jul. 2011, pp. 1–6. doi: https://doi.org/10.1109/icme.2011.6011957.

[7] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural Features for Image Classification,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973, doi: https://doi.org/10.1109/tsmc.1973.4309314.

[8] H. Murray, A. Lucieer, and R. Williams, “Texture-based Classification of sub-Antarctic Vegetation Communities on Heard Island,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 12, no. 3, pp. 138–149, Jun. 2010, doi: https://doi.org/10.1016/j.jag.2010.01.006.

[9] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, San Diego, CA, USA, 2005, pp. 886–893. doi: https://doi.org/10.1109/cvpr.2005.177.

[10] T. Ojala, M. Pietikäinen, and D. Harwood, “A Comparative Study of Texture Measures with Classification Based on Featured Distributions,” *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, Jan. 1996, doi: https://doi.org/10.1016/0031-3203(95)00067-4.

[11] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, Jul. 2002, doi: https://doi.org/10.1109/tpami.2002.1017623.

[12] T. Ojala and M. Pietikäinen, “Unsupervised Texture Segmentation Using Feature Distributions,” *Pattern Recognition*, vol. 32, no. 3, pp. 477–486, Mar. 1999, doi: https://doi.org/10.1016/s0031-3203(98)00038-7.

[13] M. Pietikäinen, “Local Binary Patterns,” *Scholarpedia*, vol. 5, no. 3, p. 9775, 2010, doi: https://doi.org/10.4249/scholarpedia.9775.

[14] T. Mäenpää and M. Pietikäinen, “Classification with Color and texture: Jointly or separately?,” *Pattern Recognition*, vol. 37, no. 8, pp. 1629–1640, Aug. 2004, doi: https://doi.org/10.1016/j.patcog.2003.11.011.

[15] D. Xu and Y. Tian, “A Comprehensive Survey of Clustering Algorithms,” *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, Jul. 2015, doi: https://doi.org/10.1007/s40745-015-0040-1.

[16] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Englewood Cliffs, New Jersey: Prentice-Hall, 1988.

[17] J. Macqueen, “Some Methods for Classification and Analysis of Multivariate Observations,” *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, 1967, Available: https://digitalassets.lib.berkeley.edu/math/ucb/text/math_s5_v1_article-17.pdf

[18] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1996, pp. 226–231. Available: https://cdn.aaai.org/KDD/1996/KDD96-037.pdf

[19] Y. Zhao, Y. Yuan, F. Nie, and Q. Wang, “Spectral Clustering Based on Iterative Optimization for large-scale and high-dimensional Data,” *Neurocomputing*, vol. 318, no. 0925-2312, pp. 227–235, Nov. 2018, doi: https://doi.org/10.1016/j.neucom.2018.08.059.

[20] S. Ranjan, D. R. Nayak, K. S. Kumar, R. Dash, and B. Majhi, “Hyperspectral Image Classification: a K-Means Clustering Based Approach,” in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2017, pp. 1–7. doi: https://doi.org/10.1109/icaccs.2017.8014707.

[21] Vij Komal and Singh Yaduvir, “Enhancement of Images Using Histogram Processing Techniques,” *International Journal of Computer Technology and Applications*, vol. 02, no. 02, Mar. 2011, Accessed: Oct. 28, 2024. [Online]. Available: https://www.researchgate.net/publication/50346270_Enhancement_of_Images_using_Histogram_Processing_Techniques

[22] F. Bianconi, A. Fernández, F. Smeraldi, and G. Pascoletti, “Colour and Texture Descriptors for Visual Recognition: A Historical Overview,” *Journal of Imaging*, vol. 7, no. 11, p. 245, Nov. 2021, doi: https://doi.org/10.3390/jimaging7110245.

[23] Y. Wang, X. Zhu, and B. Wu, “Automatic Detection of Individual Oil Palm Trees from UAV Images Using HOG Features and an SVM Classifier,” *International Journal of Remote Sensing*, vol. 40, no. 19, pp. 7356–7370, Sep. 2018, doi: https://doi.org/10.1080/01431161.2018.1513669.

Appendix A: K-Means Clustering’s Highest Scores

Table 7
Summary of the highest precision, recall and F1 scores obtained from running K-Means clustering experiments ten consecutive times. The values starred in the F1 column represent the values that differ from the modal values listen in table 3.

RESULTS	K-MEANS			K-MEANS + PCA		
METRICS	Precision	Recall	**F1**	Precision	Recall	**F1**
GLCM	0.8593	0.6424	**0.6944***	0.8592	0.6415	**0.6937**
NORMALISED GLCM	0.8312	0.6894	**0.7328**	0.8313	0.6893	**0.7327**
LBP	0.75823482	0.59363282	**0.65144813***	0.7575	0.5909	**0.6492***
NORMALISED LBP	0.8219	0.6747	**0.7204***	0.8218	0.6744	**0.7202**
HOG	0.9044	0.7511	**0.7868***	0.9043	0.7507	**0.7865***
NORMALISED HOG	0.8097	0.5458	**0.6094**	0.8097	0.5460	**0.6095**
HISTOGRAM	0.8432	0.7750	**0.7990***	0.8485	0.7307	**0.7667***
NORMALISED HISTOGRAM	0.8439	0.6891	**0.7334**	0.8433	0.6578	**0.7076***

Appendix B: Spectral Clustering’s Highest Scores

Table 8
Summary of the highest precision, recall and F1 scores obtained from running Spectral clustering experiments ten consecutive times. The values starred in the F1 column represent the values that differ from the modal values listen in Table 4

RESULTS	SPECTRAL			SPECTRAL + PCA		
METRICS	Precision	Recall	**F1**	Precision	Recall	**F1**
GLCM	0.0216	0.1469	**0.0376**	0.0216	0.1469	**0.0376**
NORMALISED GLCM	0.8324	0.7243	**0.7598***	0.8324	0.7243	**0.7598***
LBP	0.7584	0.5942	**0.6519***	0.7586	0.5949	**0.6525***
NORMALISED LBP	0.82097	0.67368	**0.71954***	0.8210	0.6737	**0.7195***
HOG	0.0216	0.1467	**0.0376**	0.0213	0.1446	**0.0371**
NORMALISED HOG	0.8165	0.6706	**0.7168***	0.6528	0.3321	**0.4077**
HISTOGRAM	0.8425	0.7742	**0.7983***	0.7447	0.8485	**0.7844**
NORMALISED HISTOGRAM	0.8747	0.1484	**0.0406**	-	-	**-**

Appendix C: Alternative Graph Results

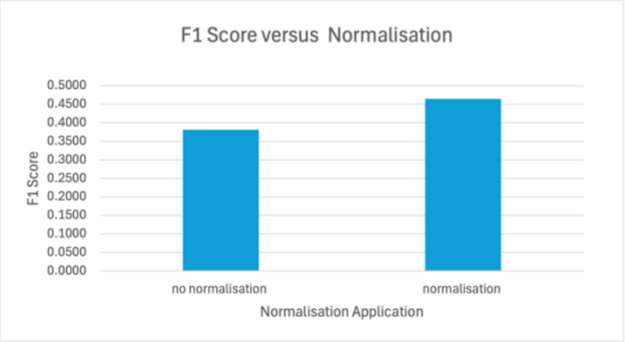

Figure 11 Graph Depicting the Effect of Normalisation on Clustering Results Using the Best predictions From Each Experiment

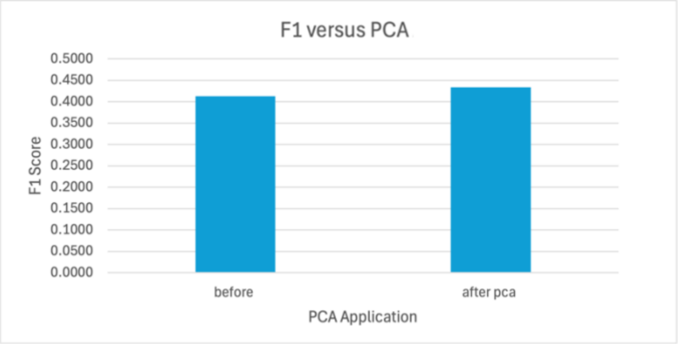

Figure 12 Graph Depicting the Effect of PCA on Clustering Results Using the Best predictions From Each Experiment

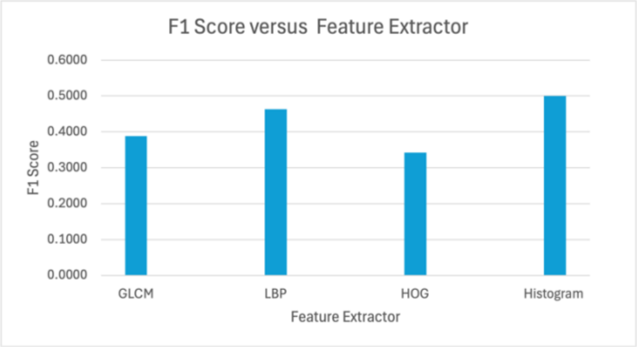

Figure 13 Graph Depicting Summary of Feature Extractor Results Using the Best predictions From Each Experiment

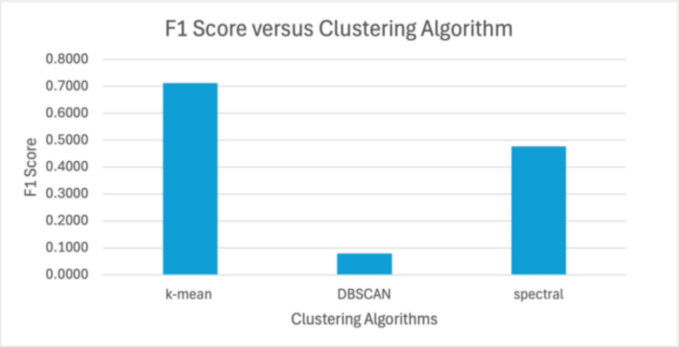

Figure 14 Graph Depicting the Summary of Clustering Algorithm Results Using the Best predictions From Each Experiment

As expected, every bar, except for the average of DBSCAN in Figure 14, has increased. K-Means is still the best-performing algorithm with an average of 0.7121. The conclusion for PCA and Normalization remains the same, though their effects are more subtle compared to Figures 7 and 8. The difference in performance of the feature extractors (Figure 13) is also less drastic compared to Figure 9 with the greyscale histogram performing the best. The best-performing experiment is the K-Means clustering experiment with the greyscale histogram feature vector (without PCA or normalization applied), which obtained an F1 score of 0.7990. Overall, the general conclusion remains similar with a few changes.