

Near Sets in Set Pattern Classification

by

Chidoteremndu Chinonyelum Uchime

A Thesis

submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg, Manitoba, R3T 5V6, Canada

Copyright © January, 2015 by Chidoteremndu Chinonyelum Uchime

Abstract

This research is focused on the extraction of visual set patterns in digital images, using relational properties like nearness and similarity measures, as well as descriptive properties such as texture, colour and image gradient directions. The problem considered in this thesis is application of topology in visual set pattern discovery, and consequently pattern generation. A visual set pattern is a collection of motif patterns generated from different unique points called seed motifs in the set. Each motif pattern is a descriptive neighbourhood of a seed motif. Such a neighbourhood is a set of points that are descriptively near a seed motif. A new similarity distance measure based on dot product between image feature vectors was introduced in this research, for image classification with the generated visual set patterns. An application of this approach to pattern generation can be useful in content based image retrieval and image classification.

Keywords: Pattern, pattern generation, pattern recognition, proximity space, visual set pattern, motif pattern, seed motif, description, perception, probe function, feature vector, binary classification, K-means, CBIR, salient, similarity measure, sensitivity, specificity.

Acknowledgements

I will like to thank the following people, for their role in the completion of this thesis.

- First, I thank God for preserving me in good health.
- My advisor, Professor James Peters for accepting me as his graduate student. His support, advice and leadership made the completion of this thesis possible.
- My examination committee members, Professor Attahiru Alfa and Professor Carson Leung, for accepting to serve on my committee.
- In a special way, I will like to thank Professor Attahiru Alfa, for believing in me, especially in the early stages of my program. Thanks for supporting me like a father, throughout my M.Sc. program.
- University of Manitoba, Faculty of Graduate Studies, for all the financial awards.
- Department of Electrical and Computer Engineering, for recommending me as a recipient for University of Manitoba Graduate Fellowship (UMGF) award. This gave me the financial stability that I needed, to help me focus on my research.
- My endearing mother and siblings, for supporting and believing in me.
- My lovely sister Onyinye Uchime, for all the encouragement, tips, teachings and support (both financial and emotional) throughout this journey.
- My lovely friend Okechukwu Efobi, for being there for me, every step of the way, towards the completion of this thesis and beyond.
- Special thanks to Mr. and Mrs. Anyadike, for helping me to settle down in Winnipeg, from the first day I landed, and for making me a part of their family.
- My friends in Winnipeg and lab mates, especially Randima, Ernest, David, Charles, Enoch, Mauricio, Matthew, Mrs. Bernice Ezirim, Emeka, Muyiwa, Dan, Saad, Ada, Charith, Kamal and Mita, for all their encouragements and support throughout my M.Sc. program.

Dedication

I dedicate this work to my sister Onyinye Uchime, my friend Okechukwu Efobi
and
to the lovely memory of my late aunty, Uchenna Flora Dita.

Contents

| | |
|---|------------|
| Abstract | iii |
| Acknowledgements | iv |
| Dedication | v |
| 1 Introduction | 1 |
| 2 State of the Art and Literature Review | 4 |
| 2.1 Computer Vision | 4 |
| 2.2 Probe Functions | 5 |
| 2.2.1 Gradient Direction | 5 |
| 2.2.2 Colour | 6 |
| 2.2.3 Texture | 6 |
| 2.3 Near Sets | 8 |
| 2.4 Neighbourhood of a Point | 9 |
| 2.4.1 Spherical Neighbourhood of a Point | 10 |
| 2.4.2 Unbounded Descriptive Neighbourhood of an Image Point | 11 |
| 2.4.3 Bounded Descriptive Neighbourhood of an Image Point | 12 |
| 2.4.4 Indistinguishable Bounded Descriptive Neighbourhood of an Image Point | 13 |
| 2.4.5 Indistinguishable Unbounded Descriptive Neighbourhood of an Image Point | 14 |
| 2.5 K-means Clustering | 15 |

| | | |
|----------|---|-----------|
| 2.6 | Binary Classification | 16 |
| 2.7 | Similarity Distance Measure | 16 |
| 2.7.1 | Čech Distance | 17 |
| 2.7.2 | Hausdorff Distance | 17 |
| 2.8 | Data set | 18 |
| 3 | Methodology | 19 |
| 3.1 | Image Pre-processing Phase | 20 |
| 3.1.1 | Digital Image Resizing | 20 |
| 3.1.2 | Edge Detection | 20 |
| 3.2 | Visual Set Pattern Discovery Phase | 22 |
| 3.2.1 | Seed Motif Selection | 22 |
| 3.2.2 | Motif Pattern Generation | 25 |
| 3.2.3 | Set Pattern Generation | 26 |
| 3.3 | Feature Extraction Phase | 28 |
| 3.4 | Similarity Distance Calculation Phase | 28 |
| 3.4.1 | Angle Between Vectors Using their Dot Product | 29 |
| 3.5 | Similarity Measure Based Classification Phase | 31 |
| 4 | Experiment and Results | 32 |
| 4.1 | Input Data | 32 |
| 4.2 | Experimental Setup | 33 |
| 4.3 | Testing and Image Classification | 35 |
| 4.3.1 | Classification with Similarity Distance Measure | 35 |
| 4.4 | Results and Analysis | 41 |
| 4.4.1 | Sensitivity and Specificity Analysis of Result | 43 |
| 4.4.2 | Correctly Classified Test Images | 50 |
| 4.4.3 | Average Time of Classification of a Test Image | 54 |
| 5 | Conclusion and Future Work | 62 |
| 5.1 | Conclusion | 62 |

| | |
|---|-----------|
| 5.2 Recommendation and Future Work | 63 |
| A Threshold Selection | 64 |
| B User Manual for Set_Pattern_Classification GUI | 66 |
| References | 71 |

List of Tables

| | | |
|-----|--|----|
| 4.1 | Sensitivity and specificity measures for set pattern method of classification testing using dinosaur query images. | 45 |
| 4.2 | Sensitivity and specificity measures for K-means method of classification testing using dinosaur query images. | 46 |
| 4.3 | Sensitivity and specificity measures for set pattern method of classification testing using horse query images. | 48 |
| 4.4 | Sensitivity and specificity measures for K-means method of classification testing using horse query images. | 49 |

List of Figures

| | | |
|-----|---|----|
| 2.1 | Popular Lena image and a point x in the image. | 9 |
| 2.2 | Spherical neighbourhood of point x ($\epsilon = 60$). | 10 |
| 2.3 | Unbounded Descriptive Neighbourhood of Point x ($\epsilon = 10$). | 12 |
| 2.4 | Bounded descriptive Neighbourhood of Point x ($\epsilon = 10; r = 60$). | 13 |
| 2.5 | Indistinguishable Bounded Descriptive Neighbourhood of Point x ($r = 60$). | 14 |
| 2.6 | Indistinguishable Unbounded Descriptive Neighbourhood of Point x | 15 |
| 3.1 | Methodology flow chart. | 19 |
| 3.2 | Edge maps of some sample images from SIMPLIcity database [50] [51]. | 21 |
| 3.3 | <i>473.jpg</i> digital image as a proximity space. | 23 |
| 3.4 | Edge gradient direction of <i>473.jpg</i> digital image. | 24 |
| 3.5 | Three modal seed motif points. | 24 |
| 3.6 | Motif patterns of the three modal seed motifs. | 26 |
| 3.7 | Visual Set pattern in <i>473.jpg</i> digital image. | 27 |
| 4.1 | Sample images from horse image class (SIMPLIcity database [50] [51]). | 32 |
| 4.2 | Sample images from dinosaur image class (SIMPLIcity database [50] [51]). | 33 |
| 4.3 | Set_Pattern_Classification graphic user interface (GUI). | 34 |
| 4.4 | Similarity distance between <i>464.jpg</i> and 50 dinosaur test images. | 37 |
| 4.5 | Similarity distance between <i>464.jpg</i> and 50 horse test images. | 38 |
| 4.6 | Similarity distance between <i>464.jpg</i> and 100 test images with respect to ϵ_1 | 39 |
| 4.7 | Similarity distance between <i>464.jpg</i> and 100 test images with respect to ϵ_2 | 40 |
| 4.8 | Patterns generated on edge maps using, both set pattern & K-means method of pattern generation. | 42 |

| | | |
|------|---|----|
| 4.9 | Percentage of correctly classified test images using 30 random dinosaur query images. | 50 |
| 4.10 | Percentage of correctly classified test images using 30 random horse query images. | 52 |
| 4.11 | Average classification time dinosaur query image vs. dinosaur test images. . | 55 |
| 4.12 | Average classification time dinosaur query image vs. horse test images. . . | 56 |
| 4.13 | Average classification time horse query Image vs. dinosaur test images. . . | 57 |
| 4.14 | Average classification time horse query image vs. horse test images. | 58 |
| 4.15 | Average classification time dinosaur query image vs. dinosaur & horse test images. | 59 |
| 4.16 | Average classification time horse query image vs. dinosaur & horse test images. | 61 |
| B.1 | Query Image. | 66 |
| B.2 | Test Image. | 67 |
| B.3 | Method of Classification. | 67 |
| B.4 | Set Pattern Parameters. | 68 |
| B.5 | K-means Parameters. | 68 |
| B.6 | Patterns in Query Image. | 69 |
| B.7 | Patterns in Test Image. | 69 |
| B.8 | Similarity Threshold Range. | 70 |
| B.9 | Output Result. | 70 |

Chapter 1

Introduction

Human perception to the similarities between objects depends on signals received from our senses and is highly feature based [1] [2]. These signals received are then mapped to values, and the values are used as the basis of comparison to tell if objects are similar or not. Henry in [3], described the human sense to be modelled as a probe, measuring the physical characteristics of objects in our environment. These physical characteristics of objects in our environment can be identified with the features of the objects. In [4], it is stated that the perception of sensed objects are formed by our mind identifying relationships between object feature values mapped from signals received by our senses. Thus objects that have similar descriptions are said to be perceptually near each other [3].

This way of human perception has been emulated and thus forms the basis for the near set theory. The main idea of near set theory is that near sets are disjoint sets of objects that resemble each other. Resemblance here, is determined by considering set descriptions defined by feature vectors, where feature vectors are n-dimensional vectors of numerical features that represent characteristics of points in the set [5]. This concept can thus be employed in creating perception based computer systems for solving problems in engineering and science, e.g., image analysis, classification and so on.

Near set theory has been applied to various works such as: identification of features in an automated facial feature extraction procedure [6]; image correspondence [7] [8] [9] [10];

adaptive learning [11]; image processing problems, like image morphology and segmentation evaluation [12] [13]; classification of digital image by introduction of wavelets in near sets approach [14]; study of region of interest in paintings [15] and so on.

K-means clustering as explained in [16], is an unsupervised learning clustering process. Clustering here is a form of pattern generation in a set, which initializes by selecting ‘K’ random number of centroids. Data points in the set are then assigned to these centroids, based on their proximity to the centroids. New centroids are recalculated and the process iterates by assigning data points in the set to the new centroids, till convergence is reached. Because of this random initializations, patterns generated in this process are random and changes depending on the initial centroids randomly selected. It also involves reassignment of points to different clusters in every iteration till convergence is reached and all the points in the set make up the resulting pattern. Thus, when applying this process to classification, computational times are not so fast and classification results are unpredictable since the initialization of the process is random in nature.

Recently, near sets were employed in generating neighbourhoods of points that are useful in pattern recognition [17]. Thus, in this research, using this concept of neighbourhood of a point, visual set patterns, which form a basis for image classifications were generated. In our work, visual set patterns in a set are generated from collections of motif patterns generated from different unique points called seed motifs, where a motif pattern is a descriptive neighbourhood of a seed motif. Neighbourhoods in this case, are a set of points that are descriptively near the seed motif. In effect, a visual set pattern is an example of a hierarchy of information granules, ranging from unique seed motifs points, to descriptive neighbourhoods of these points, to collections of all these neighbourhoods generated. These set patterns generated in this process are then used in image binary classification, using a new similarity distance measure, based on dot product between the set pattern’s feature vectors. It is important to note that set patterns generated in images using this method are not random. Finally, set patterns generated using this proposed method, generally gave a high percentage of successful classification, high sensitivity and high specificity

at a very fast computational time, when applied to binary classification.

Scope and Arrangement of the Thesis

This thesis is organized as follows: chapter 1 presents the introduction and arrangement of the thesis; chapter 2 summarizes the state of the art and review of related literature, thus building the background of the work reported in the thesis; chapter 3 presents the methodology used in generating visual set patterns and performing similarity distance based classification; chapter 4 discusses the experiment setup, data analysis and results obtained in this thesis, using a real world data set; finally, chapter 5 concludes the thesis, and suggests some future work.

Chapter 2

State of the Art and Literature Review

This chapter introduces techniques and developments in image analysis, as well as commonly employed methodologies that leads to successful pattern recognition in digital images. It also presents some literature review of related researches on topological approaches to pattern generation and recognition, with respect to digital images.

2.1 Computer Vision

Computer vision encompasses various ways of acquiring, processing and analyzing images from the real world in order to produce numerical or symbolic information for proper decision making. As humans, with the help of our sensory organs, we can identify an object and where it is, as we interact with our environment. On the other hand, computer vision systems relies on sensor measurements from reflected brightness from an image, taking into consideration some complex interactions of many other parameters like the position and angle of the object, the light sources with respect to the sensors, etc.

Thus, computer vision can be described as a representation of vision perception, since it involves electronically perceiving and understanding an image [18] [19] [20] [21] [22]. This gives rise to the major task of computer vision, which is deciphering what scene or object generated a particular digital image, after undergoing many complex interactions to be digitized, especially since image brightness is not generally independent. Thus the goal

of computer vision is to determine sufficient additional constraints to invert brightness into scene parameters [23].

Some of the applications of computer vision mentioned in [23] include:

- Visual inspection.
- Limited accurate and reliable detection and classification in Automatic Target Recognition (ATR).
- Assembly and material handling.
- Photo interpretation.
- Content-based image retrieval (CBIR), which is the application of computer vision techniques in the search for digital images in large databases, thus solving the problem of image retrieval [24] [25] [26] [27].

2.2 Probe Functions

According to Peters in [28] [29], a probe function is defined as a real value function, representing a feature of a perceptual object. Where a perceptual object is something that is perceivable that has its origin in the physical world. According to Henry, the idea of probe functions is the heart of near sets, since they are used to measure characteristics of visual objects and similarities among perceptual objects [3]. Some of the probe functions representing descriptive features in digital images used in this research include:

2.2.1 Gradient Direction

The gradient direction of a point in a digital image is computed from its directional change in intensity, with respect to the x-axis and y-axis (G_x and G_y) called *image gradient*. These image gradients extract useful information from the images and are thus used for the calculation of gradient directions as follows. Gradient direction of a pixel is given by

$$\theta(x, y) = \arctan \left[\frac{G_x}{G_y} \right]. \quad (2.2.1)$$

2.2.2 Colour

Colour as a probe function provides perceptually relevant representation of the colour content of an image. It is invariant to image size and orientation [30] [31]. Colour being one of the most useful features that helps humans recognize images, is used in this research for pattern recognition and classification. Colour can be defined in different n-dimensional colour spaces. Some of the colour spaces used in this work include:

- **RGB colour space** : The RGB colour space is an additive colour space based on the RGB colour model, in which red, green, and blue light are added together in various ways to reproduce a broad array of colours [32].
- **HSV colour space** : HSV stands for hue, saturation and value. HSV colour space is a cylindrical coordinate representation of the RGB colour model. Humans tend to focus on the hue, saturation and intensity (brightness) of an image, to be able to describe the image correctly. Thus the HSV colour model is best for colour description. In this work the saturation of an image, which measures how much of pure colour is diluted with white light, was used as a probe function for classification.

2.2.3 Texture

Texture is a visual feature that refers to the fundamental surface properties of an object and its relationship to the surrounding environment [33]. Haralick described texture as one of the most important characteristics in identifying region of interests in digital images [31]. Generally, there are statistical and structural approaches to identifying texture [34]. These statistical measures are generated from the grey level co-occurrence matrix.

Grey level co-occurrence matrices is one of the most widely used and efficient approaches in extracting texture features in grey level images. Where the co-occurrence probabilities provide a second order method for generating texture features [31]. The steps involved in calculating the grey level co-occurrence probabilities include [31] [33]:

1. Converting the RGB values of the colour image to greyscale Y, by calculating a

weighted sum of the red (R), green (G), and blue (B) components as in

$$Y = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B. \quad (2.2.2)$$

2. Calculating the co-occurrence probabilities between grey levels i and j , present in the grey level region or grey level image Y . Given two parameters namely inter pixel distance (δ) and orientation (θ), these probabilities representing the joint probabilities of all pair wise combinations of grey levels in the image of interest are calculated. Thus the co-occurrence probability between grey levels i and j is given by

$$C_{ij} = \frac{P_{ij}}{\sum_{i,j=0}^G P_{ij}}. \quad (2.2.3)$$

Where, P_{ij} is the number of occurrences of grey levels i and j given a certain (δ, θ) pair within the image. G is the quantized number of grey levels. $\sum_{i,j=0}^G P_{ij}$ is the total number of grey level pairs (i, j) in the region. These grey co-occurrence probabilities make up the grey level co-occurrence matrix.

Some of the textural properties, generated from statistical measures applied to the grey co-occurrence probabilities, used in this research as explained in [35] include:

- **Contrast** : This returns a measure of the intensity contrast between a pixel and its neighbour over the whole image, and is defined by

$$Contrast = \sum C_{ij} |i - j|^2. \quad (2.2.4)$$

- **Correlation** : This returns a measure of how correlated a pixel is to its neighbour over the whole image, and is defined by

$$Correlation = \sum_{i,j} \frac{(i - \mu_i)(j - \mu_j)C_{ij}}{\sigma_i \sigma_j}. \quad (2.2.5)$$

- **Energy** : This returns the sum of squared elements in the grey level co-occurrence matrix, and is defined by

$$Energy = \sum_{i,j} (C_{ij})^2. \quad (2.2.6)$$

- **Homogeneity** : This measures the closeness of the distribution of elements in the grey level co-occurrence matrix to the grey level co-occurrence matrix diagonal, and is defined by

$$Homogeneity = \sum_{i,j} \frac{C_{ij}}{1 + |i - j|}. \quad (2.2.7)$$

2.3 Near Sets

As humans, we constantly employ the concept of nearness, as we intuitively look at objects in our environment, and quickly assess the degree of their similarity with each other. This led to the birth of the idea of nearness in science. In 1908, Frigyes Riesz first published a paper on the nearness of sets, pioneering the field of study that is now known as proximity spaces [36] [37]. Proximity spaces here, describes a proximity relation that provides a framework for identifying how near a point is to a set and how near a set is to a set. Nearness here is completely based on the spatial relationship between the objects in question and not based on the descriptions of the objects.

Near set theory was discovered after collaboration in 2002 by Z. Pawlak and J. F. Peters [3], on a poem entitled ‘How Near’ [38]. The poem’s theme focused on the human perception of nearness. In 2007, this idea led to the publication of two papers on near set theory by Peters [39] [29]. These two papers introduced the fundamentals of near set theory and definitions associated with it. The papers showed how near set theory was influenced by Pawlak’s work on rough set theory [40] and Orłowska’s work on approximation spaces [41] [42], and how nearness of disjoint sets, based on objects’ descriptions was discovered. It is important to note that, near sets are discovered without the approximation of sets.

Thus near sets are defined as disjoint sets of objects that resemble each other, resemblance is determined by considering set descriptions that are defined by feature vectors. Feature vectors here, are n-dimensional vectors of numerical features that represent characteristics of points in the set [5]. Thus, near set theory emulates human perception in identifying

object nearness, based on their descriptions.

Near set theory has been applied to many applications and many works based on near set theory has been published over the years. Some of these works include: identification of features in an automated facial feature extraction procedure [6]; image correspondence [7] [8] [9] [10]; adaptive learning [11]; image processing problems like image morphology and segmentation evaluation [12] [13]; classification of digital image by introduction of wavelets in near set approach [14]; study of region of interest in paintings [15].

2.4 Neighbourhood of a Point

In 2014, Peters in [17] introduced the use of near set theory in finding neighbourhood of points. These were subsequently applied towards generating unique patterns in sets in this research. Peters defined a neighbourhood of a point x in a non-empty set X , as a set of points that are sufficiently near x . A point y belongs to a neighbourhood of point x , provided y satisfies the criteria to be near x . That is, the distance between x and y is less than some number $\epsilon \in (0, \infty]$. Here, distance is measured by some distance function. Neighbourhood can be either a descriptive neighbourhood or a spatial (spherical) neighbourhood. Some of these neighbourhoods as explained by Peters in [17] include the following:



Figure 2.1: Popular Lena image and a point x in the image.

2.4.1 Spherical Neighbourhood of a Point

According to Peters in [17], let X be a metric space; let $\epsilon \in (0, \infty]$ be a bound on neighbourhood distances of points from a centre $x \in X$; and let d be a metric. A spherical neighbourhood of a point x (denoted by N_x) is defined by

$$N_x = \{y \in X : d(x, y) < \epsilon\} \quad (\text{open neighbourhood of } x), \quad (2.4.1)$$

$$N_x = \{y \in X : d(x, y) \leq \epsilon\} \quad (\text{closed neighbourhood of } x). \quad (2.4.2)$$

Let X be a subset of the real number \mathbb{R} and let $x, y \in X$. Then consider the metric $d : X \times X \rightarrow [0, \infty]$ to be defined by

$$d(x, y) = |x - y|. \quad (2.4.3)$$

That is, $d(x, y)$ is the standard distance between x and y and ϵ is the threshold indicating the boundary of the neighbourhood.



Figure 2.2: Spherical neighbourhood of point x ($\epsilon = 60$).

This neighbourhood can be open, thus not including the boundary points as in equation 2.4.1 or closed, including both the interior points and the boundary points as in equation 2.4.2.

Figure 2.2 shows an example of a closed spherical neighbourhood of a point x , where the points inside the neighbourhood are highlighted with green.

Descriptive Neighbourhood of an Image Point

A descriptive neighbourhood of an image point is a set of points in the image, each with a description that matches the description of the neighbourhood seed point [17].

Let X be a set of points in a digital image and let $x \in X$. Choose Φ , a set of probe functions representing features of picture points in X . Then a descriptive neighbourhood of x , denoted by $N_{\Phi(x)}$, stands for the descriptive neighbourhood of seed point x , using a set of probe functions Φ . Four different types of descriptive neighbourhood of image points introduced by Peters in [17] are:

2.4.2 Unbounded Descriptive Neighbourhood of an Image Point

This is a descriptive neighbourhood in which there is no restriction on the location of a picture point in relation to the unbounded descriptive neighbourhood centre. That is, if $N_{\Phi(x)}$ is an unbounded descriptive neighbourhood of a picture point in X , then every $y \in X$ with a description close to the description of x by a certain threshold ϵ belongs to $N_{\Phi(x)}$. Thus, it is defined by

$$N_{\Phi}(x) = \{y \in X : d(\Phi(x), \Phi(y)) < \epsilon\}. \quad (2.4.4)$$

Choosing a probe function Φ to compute the grey level intensity of each point y in a digital image X , figure 2.3 shows an example of an unbounded descriptive neighbourhood of a point x . Where the grey level intensity of point x is 31 and ϵ is 10.



Figure 2.3: Unbounded Descriptive Neighbourhood of Point x ($\epsilon = 10$).

2.4.3 Bounded Descriptive Neighbourhood of an Image Point

This is a descriptive neighbourhood in which there is a restriction on the location of a picture point in relation to the neighbourhood centre. That is, if $N_{\Phi(x)}$ is a bounded descriptive neighbourhood of a picture point in X , then every $y \in X$ within a fixed distance r from x and with a description close to the description of x by a certain threshold ϵ belongs to $N_{\Phi(x)}$. Thus, it is defined by

$$N_{\Phi(x)} = \{y \in X : d(\Phi(x), \Phi(y)) < \epsilon \ \& \ d(x, y) < r\}. \quad (2.4.5)$$

Choosing a probe function Φ to compute the grey level intensity of each point y in a digital image X , figure 2.4 shows an example of a bounded descriptive neighbourhood of a point x . Where the grey level intensity of point x is 31, ϵ is 10 and r is 60.



Figure 2.4: Bounded descriptive Neighbourhood of Point x ($\epsilon = 10; r = 60$).

2.4.4 Indistinguishable Bounded Descriptive Neighbourhood of an Image Point

This is a descriptive neighbourhood in which there is a restriction on the location of a picture point in relation to the neighbourhood centre. That is, if $N_{\Phi(x)}$ is an indistinguishable bounded descriptive neighbourhood of a picture point in X , then every $y \in X$ within a fixed distance r from x and with a description that is exactly the same as the description of x belongs to $N_{\Phi(x)}$. Thus, it is defined by

$$N_{\Phi(x)} = \{y \in X : d(\Phi(x), \Phi(y)) = 0 \ \& \ d(x, y) < r\}. \quad (2.4.6)$$

Choosing a probe function Φ to compute the grey level intensity of each point y in a digital image X , figure 2.5 shows an example of an indistinguishable bounded descriptive neighbourhood of a point x . Where the grey level intensity of point x is 31 and r is 60.



Figure 2.5: Indistinguishable Bounded Descriptive Neighbourhood of Point x ($r = 60$).

2.4.5 Indistinguishable Unbounded Descriptive Neighbourhood of an Image Point

This is a descriptive neighbourhood in which there is no restriction on the location of a picture point with a description that matches the description of the neighbourhood centre. That is, if $N_{\Phi(x)}$ is an indistinguishable unbounded descriptive neighbourhood of a picture point in X , then every $y \in X$ with a description that is exactly the same as the description of x belongs to $N_{\Phi(x)}$. Thus, it is defined by

$$N_{\Phi}(x) = \{y \in X : d(\Phi(x), \Phi(y)) = 0\}. \quad (2.4.7)$$

Choosing a probe function Φ to compute the grey level intensity of each point y in a digital image X , figure 2.6 shows an example of an indistinguishable unbounded descriptive neighbourhood of a point x . Where the grey level intensity of point x is 31.



Figure 2.6: Indistinguishable Unbounded Descriptive Neighbourhood of Point x .

2.5 K-means Clustering

A form of pattern generation is K-means clustering as explained in [16]. K-means is an unsupervised learning clustering process, which aims to cluster a set of ‘d’ dimensional observation into ‘K’ number of clusters that are fixed a priori, where each observation or data point belongs to the cluster with the nearest mean. The clustering process is initialized by randomly selecting ‘K’ centroids (one for each cluster) and assigning data points to these ‘K’ clusters. At each iteration, the data points are placed in the cluster having a centroid nearest to the data point. The proximity to the centroid is calculated using a metric, for example, the Euclidean distance metric. After placement of points to one of these clusters, the cluster centroids are recomputed. The algorithm iterates until it converges. Convergence is guaranteed for K-means clustering and is reached when the assignment of data points to the cluster remains constant in two consecutive iterations. As a result of this re-computation of centroids and re-assignment of points to clusters till convergence is reached, ‘K’ centroids changes on every iteration until convergence is reached. This algorithm aims at minimizing a squared error function, defined by

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_j^{(i)} - c_j\|^2. \quad (2.5.1)$$

Where $\|x_j^{(j)} - c_j\|^2$ is a chosen distance measure between a data point $x_j^{(j)}$ and the cluster centroid c_j . Thus indicates the distance of the n data points from their respective cluster centroids.

2.6 Binary Classification

Binary classification, also known as binomial classification involves classifying test elements into two groups using a classification criterion. Some of the classic binary classification applications include: medical testing [43], quality control [44], information retrieval [45], and so on.

Sensitivity and specificity are statistical measures used to measure the performance of a typical binary classification test. In [46], Nickerson discusses recent binary classification experiments, in which the primary performance measure was the response time.

In this work, test elements are digital images and they are classified into two groups namely *same class* or *different class* with respect to a particular query image. Sensitivity and specificity measures, as well as response time performance measure, were used to analyze the results from this research.

2.7 Similarity Distance Measure

Similarity distance measures are used to calculate the degree at which sets resemble each other, by measuring the distance between the sets. These measures are highly useful when comparing patterns for successful image classification. In the case of classification of digital images, the descriptive distance between sets are calculated, thus revealing the degree at which the patterns are similar. Some of the existing similarity measures include the following:

2.7.1 Čech Distance

The descriptive Čech distance between sets A and B as described in [47] is given by

$$D(\mathcal{Q}(A), \mathcal{Q}(B)) = \inf \{d(\Phi(a), \Phi(b)) : \Phi(a) \in \mathcal{Q}(A), \Phi(b) \in \mathcal{Q}(B)\}. \quad (2.7.1)$$

When calculating the distance between patterns $P1$ and $P2$ which are collections of sets in digital images, Čech similarity distance between $P1$ and $P2$ described by Peters in [17] is defined as

$$D_{\Phi}(P1, P2) = \inf \{D(\mathcal{Q}(A), \mathcal{Q}(B)) : A \in P1, B \in P2\}. \quad (2.7.2)$$

Where,

$$\mathcal{Q}(A) = \{\Phi(a) : a \in A\}, \quad (2.7.3a)$$

$$\mathcal{Q}(B) = \{\Phi(b) : b \in B\}, \quad (2.7.3b)$$

$$D(\mathcal{Q}(A), \mathcal{Q}(B)) = \inf \{d(\Phi(a), \Phi(b)) : \Phi(a) \in \mathcal{Q}(A), \Phi(b) \in \mathcal{Q}(B)\}, \quad (2.7.3c)$$

$$d(\Phi(a), \Phi(b)) = \sum_{i=1}^{|\Phi|} |\phi_i(a) - \phi_i(b)|. \quad (2.7.3d)$$

Equation 2.7.3a & 2.7.3b are the set of descriptions of picture points in a set A and B respectively. Equation 2.7.3c is the descriptive Čech distance between sets in A and B as described in [47].

2.7.2 Hausdorff Distance

The Hausdorff distance as described in [48] [49], is used to measure the distance between sets in a metric space. Thus, the Hausdorff distance between sets X and Y is given by

$$d_H(X, Y) = \max \left\{ \sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \right\}. \quad (2.7.4)$$

2.8 Data set

For testing the validity of the methodology of this work, two classes of digital images from SIMPLicity [50] [51] database library were used. The two classes chosen are *Dinosaurs* and *Horses* classes of digital images. Each class contains 100 images. That is, 100 images of *Dinosaurs* and 100 images of *Horses*. This data set has been used as a bench mark, in various scientific articles for content-based image retrieval and classification.

Chapter 3

Methodology

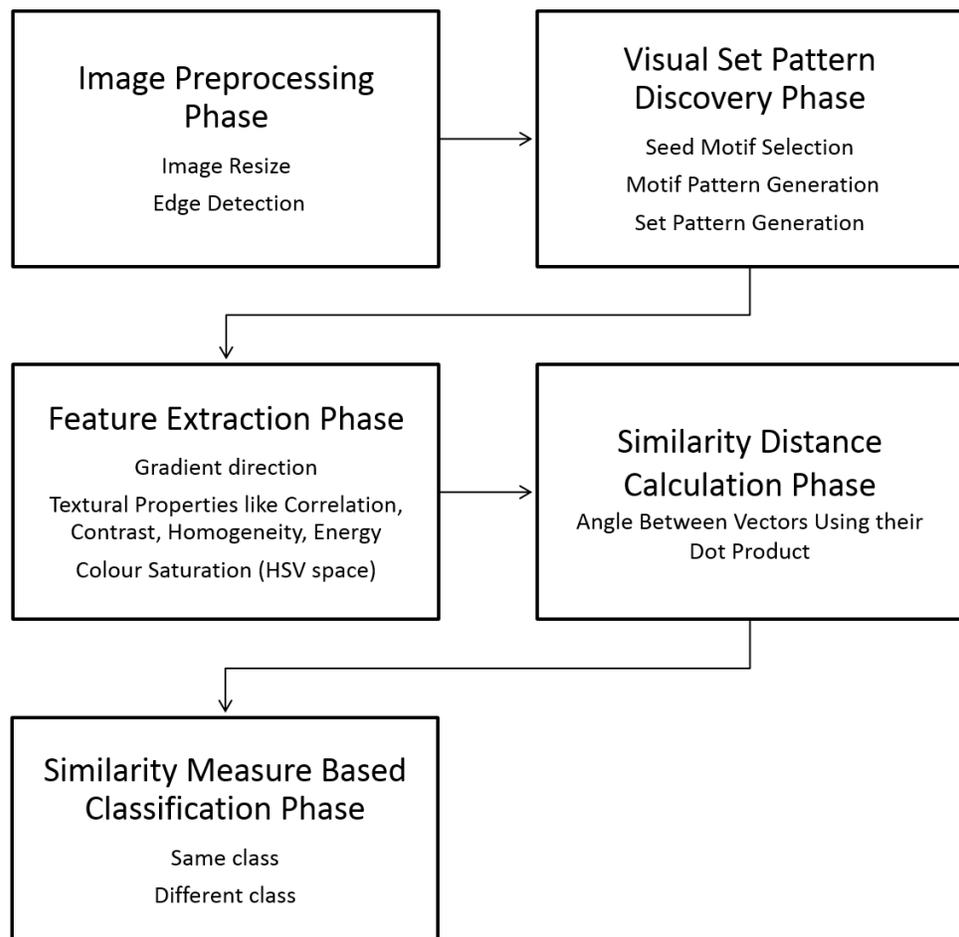


Figure 3.1: Methodology flow chart.

This chapter describes the methodology used to generate set pattern, using relational properties like nearness and similarity measures, as well as descriptive properties. It also explores the application of the proposed methodology in binary image classification. Figure 3.1 shows the flow chart of the methodology and its application in similarity based binary image classification.

3.1 Image Pre-processing Phase

3.1.1 Digital Image Resizing

This is a digital image pre-processing technique, that helps improve the computational speed of the algorithm. Some of the known methods of image resizing include nearest-neighbour interpolation, bilinear interpolation, bicubic interpolation and so on. Bicubic interpolation in which the output pixel value is a weighted average of the pixels in the nearest 4-by-4 neighbourhood was chosen in this work. The digital images were re-sized by 0.5, thus increasing its computational speed of algorithm.

3.1.2 Edge Detection

This is a digital image processing technique that detects a significant local change in the image intensity, usually associated with a discontinuity in either the image intensity or the first derivative of the image intensity [52]. There exists different methods of edge detection techniques. Some of the existing methods include: Canny's method; Sobel method; Prewitt method; Roberts method; Laplacian of Gaussian method; Zero-Cross method and so on.

According to Canny, 'The edge detection process serves to simplify the analysis of images by drastically reducing the amount of data to be processed, while at the same time preserving useful structural information about object boundaries' [53]. The steps involved in detection of edges in digital images, using the Canny's method, as explained in [53] include:

1. Smoothing of the image with a two dimensional Gaussian filter.

2. Finding the gradient of the image in both the x direction and y direction. These gradients show changes in intensity, which indicates the presence of edges.
3. Non-maximal suppression using the magnitude and direction of the gradient. This involves suppressing points where the gradient is not at maximum and letting edges occur only at points where the gradient is at a maximum.
4. Edge thresholding using ‘hysteresis’.

Canny’s optimal edge detection, which is one of the widely used edge detection methods, was used in this research because it produces thick continuous edges. Some of the resulting edges of the images used in this research, using the canny method of detection are shown in figure 3.2.

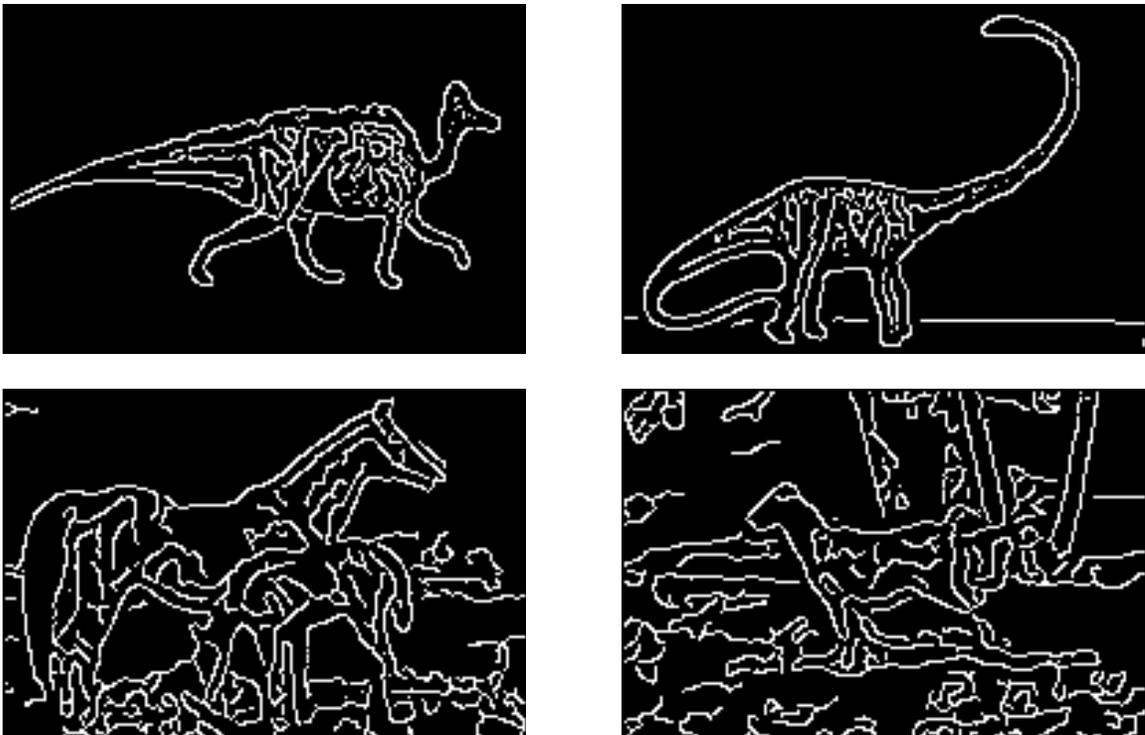


Figure 3.2: Edge maps of some sample images from SIMPLIcity database [50] [51].

3.2 Visual Set Pattern Discovery Phase

In 1960s, Pavlidis and Grenandar published papers on set patterns, where a set pattern is defined as a finite set of repeated elements such as linear segments or polygons [54] [55]. A visual set pattern in an image is a collection of sets that are near a given set, called a pattern generator, which serves as a building block in a pattern [56]. According to Peters in [17], set pattern generator is a neighbourhood of a distinguished point.

However, in this work, visual set pattern in an image is a collection of motif patterns generated in the image. Motif patterns are sets that are near a unique pattern generator, called a given seed motif, which serves as a building block in a motif pattern generation. The proposed proximity space approach to pattern recognition and pattern-based classification carries forward work on visual information granule extraction [57]. Thus it can be observed that visual set patterns typically emerge naturally in proximity spaces.

The steps in the visual set pattern discovery are explained in details below.

3.2.1 Seed Motif Selection

Seed motifs, also known as the pattern generators are the building blocks in motif pattern generation. Therefore selection of a unique seed motif is very important in the generation of unique motif pattern. Uniqueness here depends on the type of pattern to be extracted from the digital image, and this is determined by the application of the pattern recognized and subsequent pattern classification. Some of the ways to generate seed motif for pattern generation include:

1. **Adherent Point:**

An adherent point is an isolated point in a set [58]. A point $a \in A$ is descriptively adherent, provided no other point in A has the same description (based on some predefined salient features) as a . In this case, A can be the whole image or a subset of the image, depending on the application.

2. **Modal Point:** A point $a \in A$ is a modal point if its description (based on some pre-defined probe function) appears most often in A . This means that there are a lot of $a \in A$ with the same description. In this work, modal points were used as seed motifs. If more than one modal point is to be selected, to ensure that the points are unique, a value ‘mode deviation’ is used to make sure that the mode points selected are apart by ‘mode deviation’. This ensures that the subsequent motif patterns generated are unique and non-overlapping.

In this work, modal points were used in the selection of seed motifs. Both the query and test digital image sets are considered as proximity spaces. For example, let’s consider the *473.jpg* image shown in figure 3.3, from the SIMPLicity database [50] [51] as a proximity space.



Figure 3.3: *473.jpg* digital image as a proximity space.

To be able to extract seed points in a proximity space, the proximity space is endowed with a descriptive proximity relation δ_Φ . Image edge gradient direction which was discussed in section 2.2.1 was chosen as the probe function Φ as shown in figure 3.4.

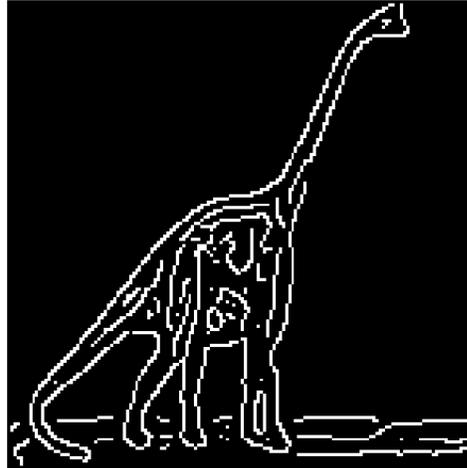


Figure 3.4: Edge gradient direction of 473.jpg digital image.

K Modal seed motifs were extracted as explained in section 3.2.1, by finding K number of modal points (mode deviation of 30) with respect to the image edge gradient direction. Three seed motifs which represents the first three modal points (with a mode deviation of 30) in the image, are shown in figure 3.5.

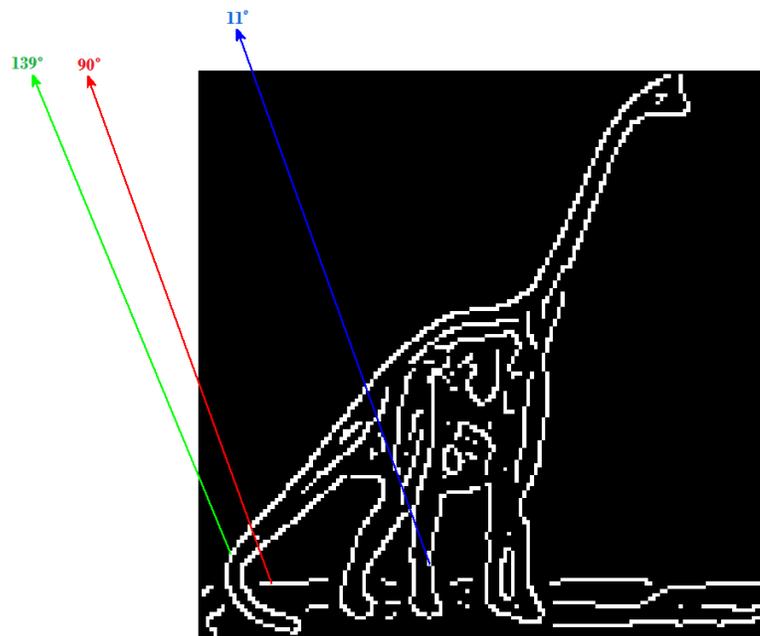


Figure 3.5: Three modal seed motif points.

3.2.2 Motif Pattern Generation

Pattern recognition in a data involves detection and extraction of patterns that make the data unique from the data set. These patterns are sometimes called salient patterns [59]. A salient pattern in a set is a subset in the set that is unique among a collection of other subsets in the set. It represents knowledge that characterizes the set completely. To be able to extract this salient pattern in a specified data, it has to be described by some well-defined set of rules. Our goal in this work, is to extract these salient patterns in digital image data, for image classification.

These seed motifs extracted in section 3.2.1, are then used as pattern generators to generate motif patterns that characterize the digital image. These motif patterns were generated by obtaining the unbounded descriptive neighbourhoods (as explained in 2.4.2) of each of these seed motifs. A threshold ϵ that describes the points in the neighbourhood is chosen for the generation of these neighbourhood. Figure 3.6 shows the unbounded neighbourhoods of the three seed motifs extracted earlier in figure 3.5. For example, consider the seed motif (red), with the gradient direction of 90° . To be able to generate a neighbourhood of this seed motif, using a threshold ϵ of 3, the points that will be part of this unbounded descriptive neighbourhood of the seed motif will be points that have a gradient direction in the range $[90-3 \ 90+3]=$ range of $[87 \ 93]$. These points can be anywhere in the image, since we are considering unbounded descriptive neighbourhoods. The same are done to generate the motif patterns for the other seed motifs. These three motif patterns (clusters) are coloured in red, green and blue as shown in figure 3.6.

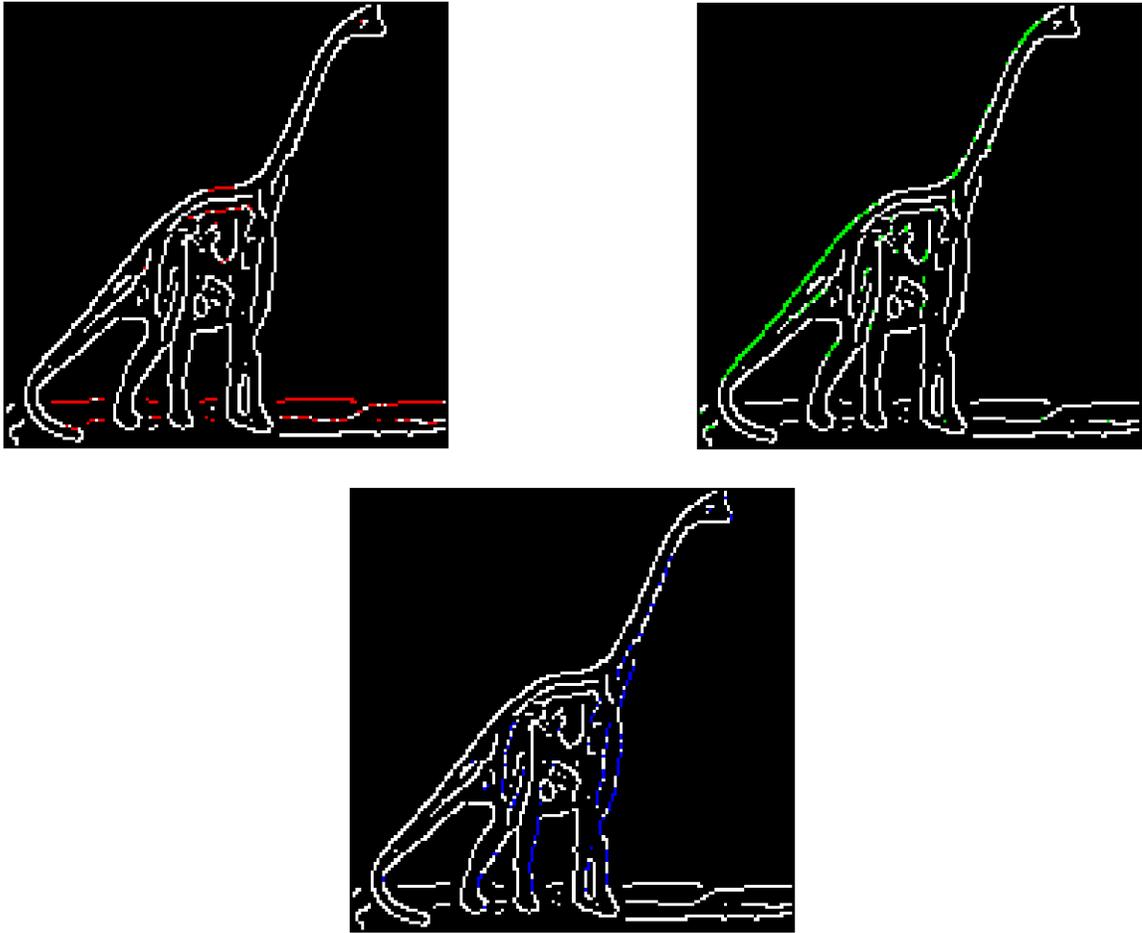


Figure 3.6: Motif patterns of the three modal seed motifs.

3.2.3 Set Pattern Generation

The union of all the motif patterns generated, gives the visual set pattern which contains points, whose features characterize the digital image. Figure 3.7 shows the visual set pattern of the *473.jpg* proximity space. It is important to note that only the points in red, green and blue are points that make up the set pattern. Points in white are not part of the set pattern.

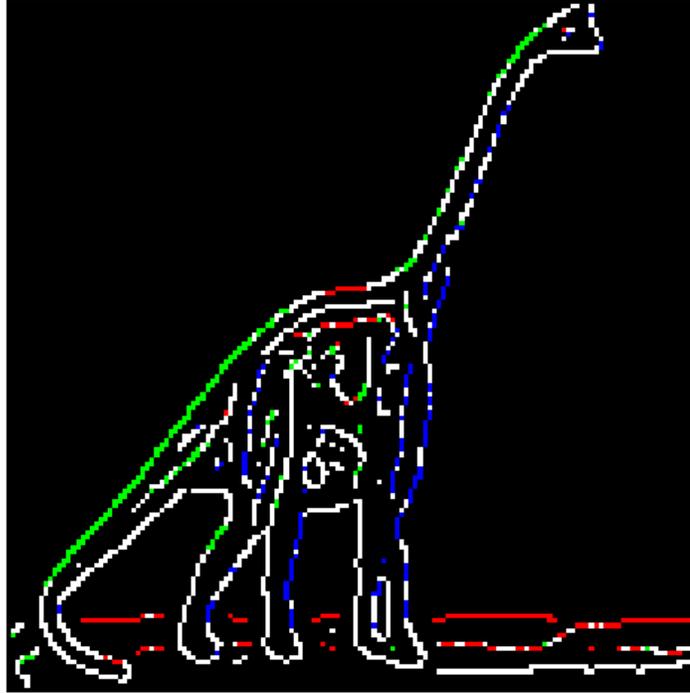


Figure 3.7: Visual Set pattern in 473.jpg digital image.

It is also important to note that the way of pattern generation is highly dependent on the application of the pattern generated and the data set involved. That is, a combination of the different types of neighbourhoods described in 2.4 can be used in the pattern generation.

Theorem 1. *The union of all the motif patterns generated in a digital image \mathcal{I} , known as the set pattern, is equal to or a subset of the image \mathcal{I} .*

Proof. Let \mathcal{I} be a digital image,

Let mP_i be motif patterns generated in \mathcal{I} ,

\therefore Set pattern $P = \bigcup mP_i \quad \forall mP_i \subset \mathcal{I}$,

If $P^c \cap \mathcal{I} \neq \emptyset \Rightarrow P \subset \mathcal{I}$.

Figure 3.7 is a picture proof of a case, where the union of the motif patterns generated is a subset of the image. Where the green, red and blue neighbourhoods are three motif patterns that make up the set pattern. Note that the remaining white points are not part of

the set pattern, although they are part of the image. Thus the union of all the motif patterns generated in the '473.jpg' image is a subset of the image.

Also, Let \mathcal{I} be a digital image,

Let mP_i be motif patterns generated in \mathcal{I} ,

\therefore Set pattern $P = \bigcup mP_i \quad \forall mP_i \subset \mathcal{I}$,

If $P^c \cap \mathcal{I} = \emptyset \Rightarrow P = \mathcal{I}$.

An example will be a uniform digital image, where all the points in the image are very near and belong to one motif pattern. □

3.3 Feature Extraction Phase

After the discovery of the visual set pattern, various features are then extracted from the points that make up this set pattern. For the subsequent similarity measure based classification, the values from these features are used in calculating the similarity distance between patterns generated in both the query image and test image.

Some of the probe functions defining the features extracted and used for classification in this work are explained in section 2.2. They include colour saturation, gradient direction and textural properties like contrast, homogeneity, correlation and energy.

3.4 Similarity Distance Calculation Phase

Once these unique patterns in these images are generated, and a set of features are extracted from the points in these patterns, a similarity distance (measure) is used to measure how the different patterns of these query and test images are related. These measurements are with respect to the description of selected unique features of points in the pattern. Thus this is a basis for membership of an image in a class. That is, it is used as a criteria to tell if a test image is in the same class as a query image or not, with respect to a predetermined

threshold. This similarity measure should be chosen, based on the type of data set, the features of the points in the patterns generated and the application of the classification. In this work, a new similarity measure was developed, employing the concept of the angle between vectors using their dot product.

3.4.1 Angle Between Vectors Using their Dot Product

If X and Y are the query image and test image respectively. We define the similarity distance between the two set patterns generated in X and Y which tells how similar X and Y as follows

Let X, Y be descriptive proximity spaces endowed with descriptive proximity δ_Φ , a metric d , an l_2 norm $\|\cdot\|_2$ and a Euclidean inner product $\langle \cdot, \cdot \rangle$.

Let $P1, P2$ be visual set patterns in X, Y , respectively, generated as explained in section 3.2.

Let A be one of the motif patterns (clusters) that make up the pattern $P1$, that is $A \in P1$

Let B be one of the motif patterns (clusters) that make up the pattern $P2$, that is $B \in P2$

Let a be points in motif pattern A , i.e., $a \in A$.

Let b be points in motif pattern B , i.e., $b \in B$.

Let $\Phi(a)$ = set of probe functions (vector of features) of point a .

Let $\Phi(b)$ = set of probe functions (vector of features) of point b .

The similarity distance $\mathbb{D}_\Phi(P1, P2)$ between patterns $P1$ and $P2$ in X and Y respectively is given as

$$\mathbb{D}_\Phi(P1, P2) = \inf \left\{ D(\widehat{\angle(A, B)}, \epsilon_1), D(\widehat{\angle(A, B)}, \epsilon_2) : A \in P1, B \in P2 \right\}. \quad (3.4.1)$$

Where,

$$\begin{aligned}
D(\widehat{\angle(A, B)}, \epsilon_1) &= \sqrt{\sum_{i=1}^k \sum_{j=1}^k |\widehat{\angle(A_i, B_j)} - \epsilon_1|^2}, \\
D(\widehat{\angle(A, B)}, \epsilon_2) &= \sqrt{\sum_{i=1}^k \sum_{j=1}^k |\widehat{\angle(A_i, B_j)} - \epsilon_2|^2}, \\
\widehat{\angle(A, B)} &= \{\widehat{\angle(A_i, B_j)} : A_i \in P1, B_j \in P2\}, \\
\widehat{\angle(A_i, B_j)} &= \frac{\angle(A_i, B_j)}{90}, \\
\angle(A_i, B_j) &= \text{Average}\{\angle(\Phi(a), \Phi(b)) : a \in A_i, b \in B_j\}, \\
\angle(\Phi(a), \Phi(b)) &= \arccos\left(\frac{a \cdot b}{\|a\|_2 \|b\|_2}\right) \\
&= \arccos\left(\frac{\sum_{i=1}^{|\Phi|} a_i b_i}{\sqrt{\sum_{i=1}^{|\Phi|} |a_i|^2} \sqrt{\sum_{i=1}^{|\Phi|} |b_i|^2}}\right).
\end{aligned}$$

$|\Phi|$ is the number of probe functions defining different features extracted from each point in the image set pattern. k is the number of motif patterns (clusters) in each of the set patterns $P1$ and $P2$, generated in both the query image and test image respectively. $\angle(A_i, B_j)$ is the average of the angles between points in each of the motif pattern pair A_i, B_j in $P1$ and $P2$. $\widehat{\angle(A_i, B_j)}$ is the normalized average angle between cluster pair A_i, B_j in $P1$ and $P2$. $\widehat{\angle(A, B)}$ is a vector containing all the normalized average angles between all possible motif pattern combinations in $P1$ and $P2$. This vector is called the **similarity vector** of the test image with respect to the query image. $D(\widehat{\angle(A, B)}, \epsilon_1)$ is the distance between a predefined threshold ϵ_1 and the similarity vector. $D(\widehat{\angle(A, B)}, \epsilon_2)$ is the distance between a predefined threshold ϵ_2 and the similarity vector. ϵ_1 is a predefined threshold that represents a lower angle and ϵ_2 is a predefined threshold that represents a higher angle. For this work, ϵ_1 and ϵ_2 were chosen, with respect to a benchmark pattern in a query image representing an image class and the extent at which images of the same class varies. Appendix A breaks down how these values for ϵ_1 and ϵ_2 were chosen for this experiment.

3.5 Similarity Measure Based Classification Phase

In this work, to classify test images with respect to a query image, the similarity measure that was used is explained in 3.4.1. Recall that the threshold ϵ_1 represents a lower angle. Therefore, if the similarity vector of a test image with respect to the query image is closer to ϵ_1 than it is to ϵ_2 as in equation 3.5.1, then it signifies that the points in the set patterns generated in both the query image and test image are closely related. Thus the query image is of the same class with the test image.

$$\mathbb{D}_\Phi(P1, P2) = D(\widehat{\angle(A, B)}, \epsilon_1). \quad (3.5.1)$$

Also recall that the threshold ϵ_2 represents a higher angle. Thus, if the similarity vector of a test image with respect to the query image is closer to ϵ_2 than it is to ϵ_1 as in equation 3.5.2, then it signifies that the points in the set patterns generated in both the query image and test image are not related. Thus the query image is of a different class with the test image.

$$\mathbb{D}_\Phi(P1, P2) = D(\widehat{\angle(A, B)}, \epsilon_2). \quad (3.5.2)$$

Chapter 4

Experiment and Results

This section explains the experiments carried out in this research and the results obtained.

4.1 Input Data

Experiments were carried out on two classes of digital images namely *dinosaurs* and *horses* from the SIMPLIcity data set [50] [51]. Some of the digital images from these classes of images used for this experiment are shown below in figures 4.1 & 4.2.



Figure 4.1: Sample images from horse image class (SIMPLIcity database [50] [51]).



Figure 4.2: Sample images from dinosaur image class (SIMPLIcity database [50] [51]).

4.2 Experimental Setup

An application named *Set Pattern Classification*, which is a MATLAB graphical user interface (GUI) was developed as a part of this research, for the purpose of testing test images with a particular query image. The application was developed using MATLAB 2011b. Figure 4.3 shows a snapshot of the GUI developed in MATLAB. Appendix B contains the user manual for using this application.

As seen in the figure 4.3, the graphical user interface has nine panels. The *QUERY IMAGE* panel lets the user select a query image, and *TEST IMAGE* panel lets the user select a test image. The *Method of Classification* panel lets the user choose a method of classification and the method's parameters. The *Set Pattern Parameters* panel lets the user select parameters for generating patterns in both the query image and test image, if *Set Pattern Classification* is selected. The *K-means Parameters* panel lets the user select parameters for generating patterns in both the query image and test image, if *K-means Classification* is selected. The *Patterns in Query Image* panel shows the resulting pattern generated in the query image, with respect to the method of classification selected. The *Patterns in Test Image* panel shows the resulting pattern generated in the test image, with respect to the method of classification selected. The *Similarity Threshold Range* panel lets the user input the ϵ_1 and ϵ_2 for the similarity measurement. The resulting distances of the test image's

similarity vector (with respect to the query image) and ϵ_1 and ϵ_2 are displayed in ‘Output Result’ panel. Also the classification result, which tells whether the test image is in the same class as the query image or in a different class, is displayed in **Output Result** panel.

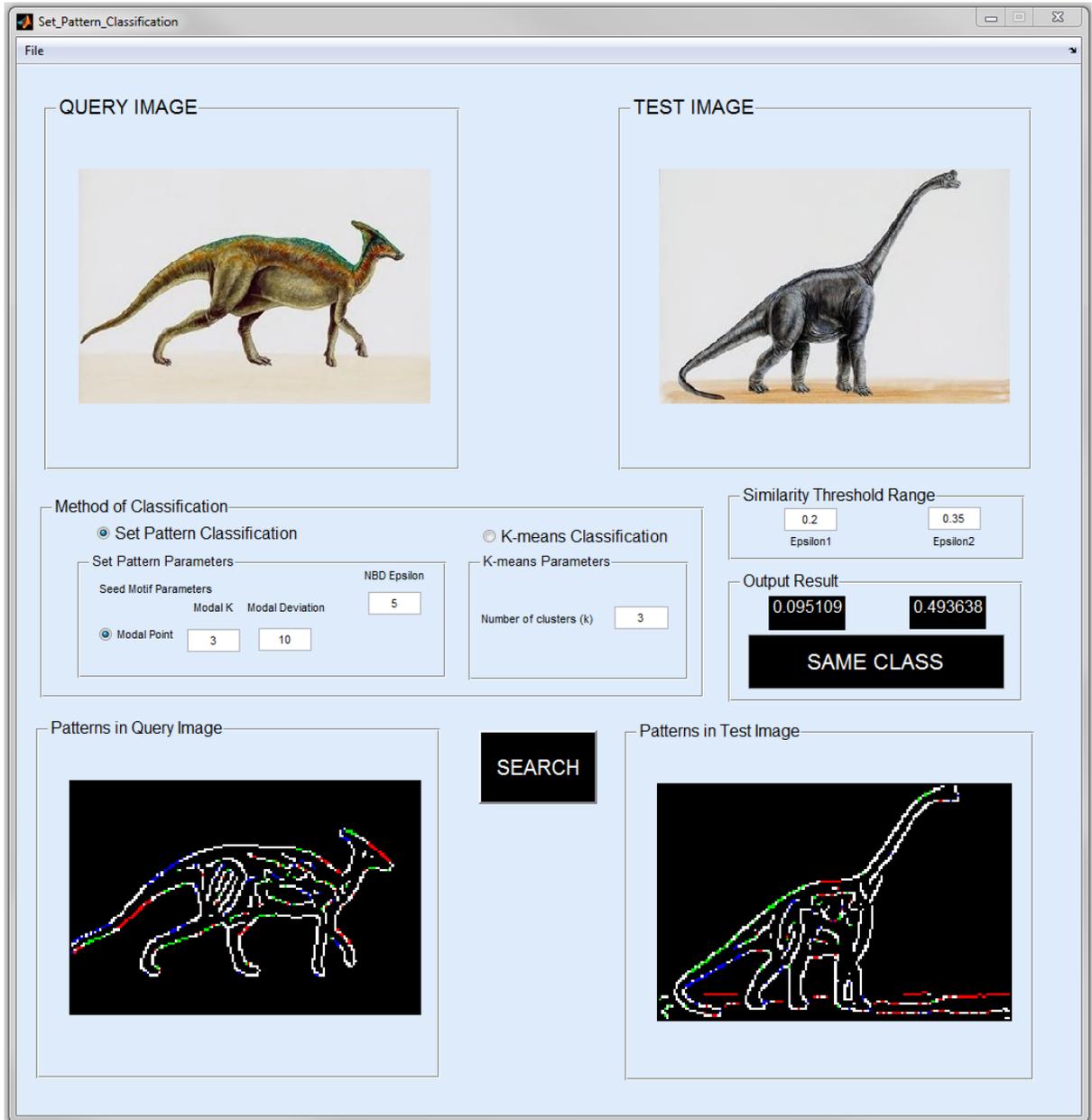


Figure 4.3: Set_Pattern_Classification graphic user interface (GUI).

A click on the **SEARCH** push button activates the classification process, once all necessary parameters has been filled.

4.3 Testing and Image Classification

This application was tested on an Intel(R) Core(TM) i7-3930K CPU @3.20GHz, 16GB RAM desktop PC using MATLAB R2011b. A query image is first selected via the **File** drop down menu. Then a test image is then selected likewise. To be able to start the comparison and classify the test image as being in the same class or different class with the query image, the method of classification and the parameters for that method of classification has to be selected. Also, the threshold ϵ_1 and ϵ_2 for the similarity measurement has to be selected. Then the **SEARCH** push button will be pressed. After processing the data and comparing the test image with the query image, using the method of classification selected, the result of the comparison will be outputted on the **Output Result** panel. This result will either be **SAME CLASS** or **DIFFERENT CLASS**.

4.3.1 Classification with Similarity Distance Measure

This work was tested by selecting 30 random query images from each of the two classes being considered. For each of the randomly selected query images, 100 randomly selected test images (50 from each class) from the database were compared to this query image. This can be seen as one round of classification, with respect to the query image.

For example, on selecting *464.jpg* from the dinosaur class of images as a query image and randomly selecting 50 test images from dinosaur class and 50 test images from horse class. These test images were compared to this query image *464.jpg*.

Let the normalized average angles between each motif pattern in the *464.jpg* dinosaur query image, and all motif patterns in each of the test images (pairwise combination), be called the ‘similarity vector’ for that test image, with respect to the query image.

Figure 4.4 shows the distances between the similarity vector for each dinosaur test image and predefined threshold ϵ_1 and ϵ_2 . It can be seen that for most of the test images, since the

query image is from the dinosaur class (same class), the distance between these similarity vector and ϵ_1 are lower than that of ϵ_2 . This is expected since it signifies that the angle between the points in the set pattern of both the dinosaur query image and dinosaur test images are very low, hence the similarity vector is tending towards a lower angle represented by ϵ_1 . Thus they are of the same class and as such, these test images were correctly classified. This can be appreciated in figure 4.4, where 48 of the test images were correctly classified.

However, if the distance between the similarity vector and ϵ_2 for any of the dinosaur test images, comes out to be lower than the distance between the similarity vector and ϵ_1 , it signifies that the angle between the points in the set pattern of both the dinosaur test image and dinosaur query image are high, and thus the test image is of a different class. In such a case, the test image will be falsely classified. This can be seen for test images 29 and 36 in the graph shown in figure 4.4.

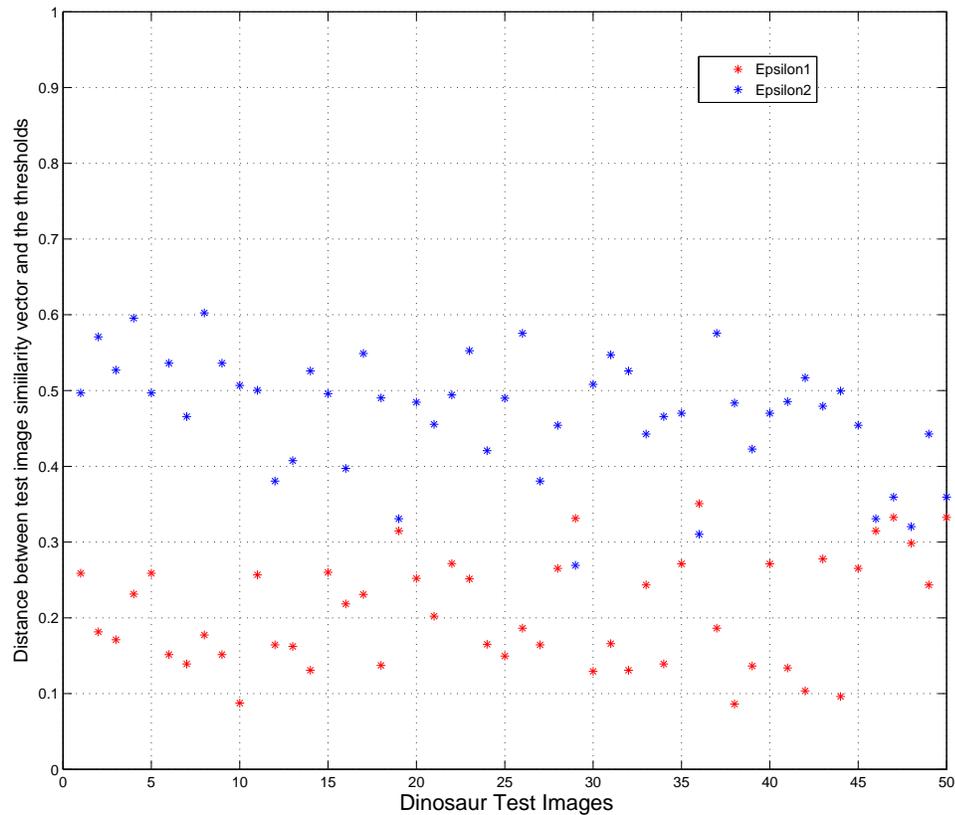


Figure 4.4: Similarity distance between 464.jpg and 50 dinosaur test images.

Figure 4.5 shows the distances between the similarity vector for each horse test image and predefined threshold ϵ_1 and ϵ_2 . It can be seen that for most of the test images, since the query image is from the dinosaur class (different class), the distance between these similarity vector and ϵ_2 are lower than that of ϵ_1 . This is expected since it signifies that the angle between the points in the set pattern of both the dinosaur query image and horse test images are very high, hence the similarity vector is tending towards a higher angle represented by ϵ_2 . Thus, they are of a different class and as such, these test images were correctly classified. This can be appreciated in figure 4.5, where 45 of the test images were correctly classified.

However, if the distance between the similarity vector and ϵ_1 for any of the horse test

images, comes out to be lower than the distance between the similarity vector and ϵ_2 , it signifies that the angle between the points in the set pattern of both the horse test image and dinosaur query image are low, and thus the test image is of the same class. In such a case, the test image will be falsely classified. This can be seen for test images 9,13,21,37 and 46 in the graph shown in figure 4.5.

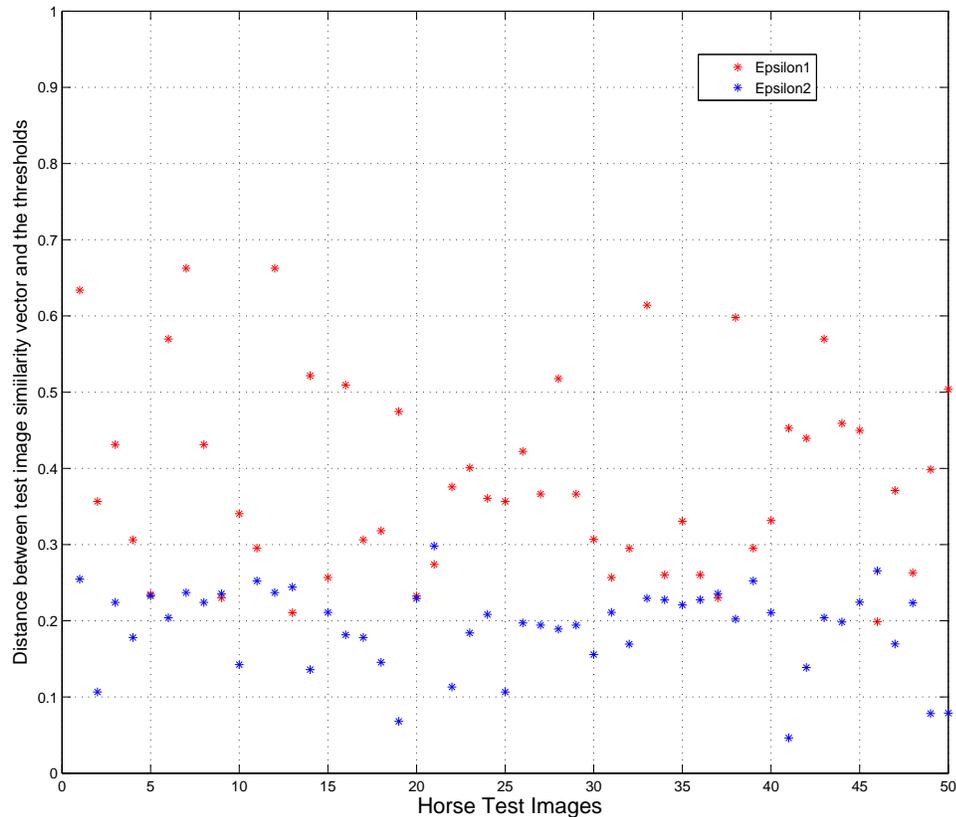


Figure 4.5: Similarity distance between 464.jpg and 50 horse test images.

For better illustration, figure 4.6 shows how the distances between ϵ_1 , and the similarity vector of these 100 test images with respect to the query image, vary from the 50 test images of the same class with the query (the first 50 test images), to the 50 test images of different class with the query (the last 50 images). It can be observed that for the first 50 images which were of the same class, most of the test images showed lower distances to ϵ_1 (which represents lower angles between points in the set pattern), and are thus similar to

the query image. However, for the next 50 test images which were of different class, most of the test images showed a considerable higher distances from ϵ_1 and are thus different from the query image.

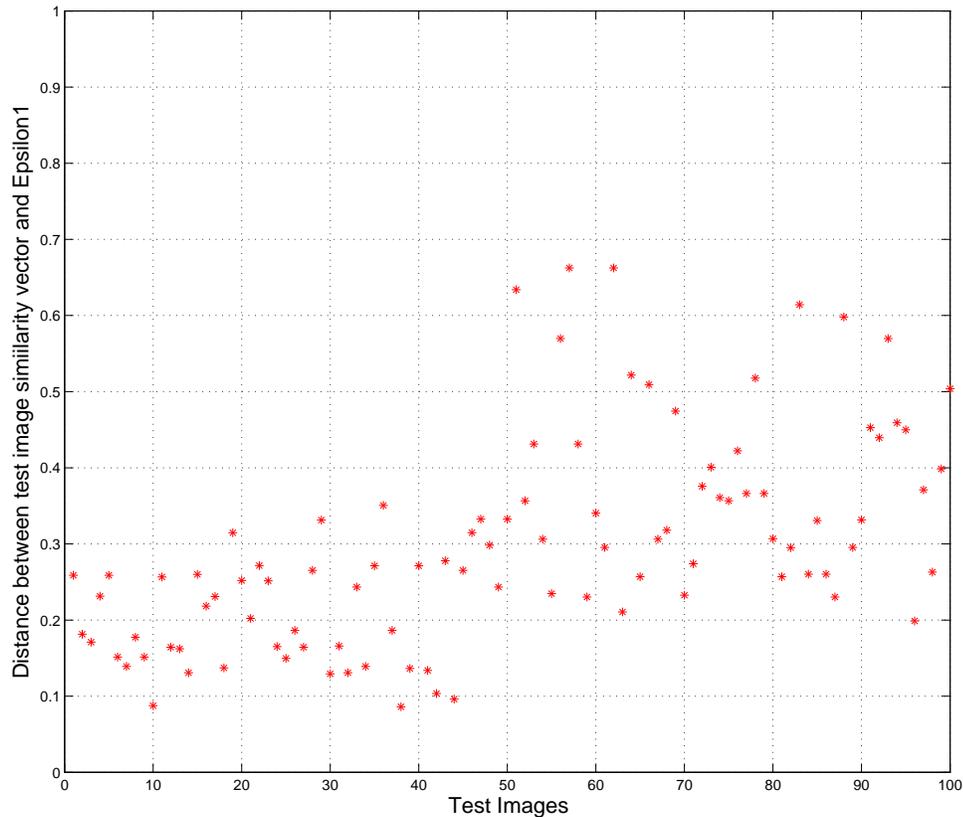


Figure 4.6: Similarity distance between *464.jpg* and 100 test images with respect to ϵ_1 .

In the same light, figure 4.7 shows how the distances between ϵ_2 , and the similarity vector of these 100 test images with respect to the query image, vary from the 50 test images of the same class with the query (the first 50 test images), to the 50 test images of different class with the query (the last 50 images). It can be observed that for the first 50 images which were of the same class, most of the test images showed higher distances to ϵ_2 (which represents higher angles between points in the set pattern) and are thus similar to the query image. However, for the next 50 test images which were of different class, most of the test images showed a considerable lower distances from ϵ_2 and are thus different from the

query image.

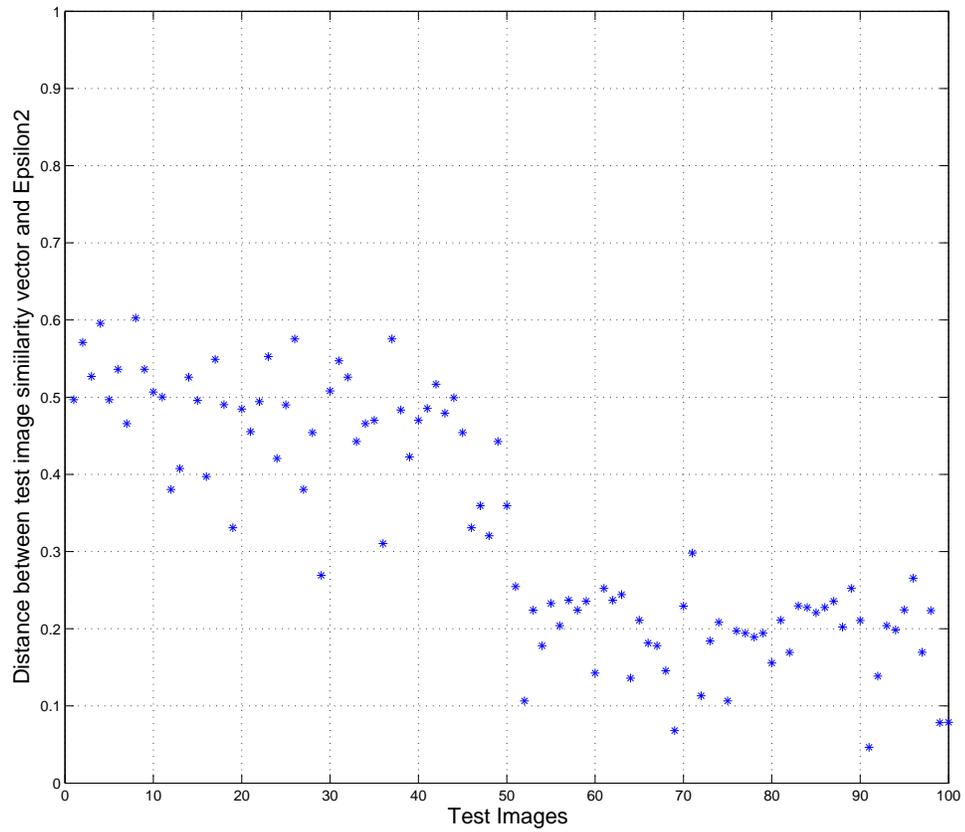


Figure 4.7: Similarity distance between *464.jpg* and 100 test images with respect to ϵ_2 .

4.4 Results and Analysis

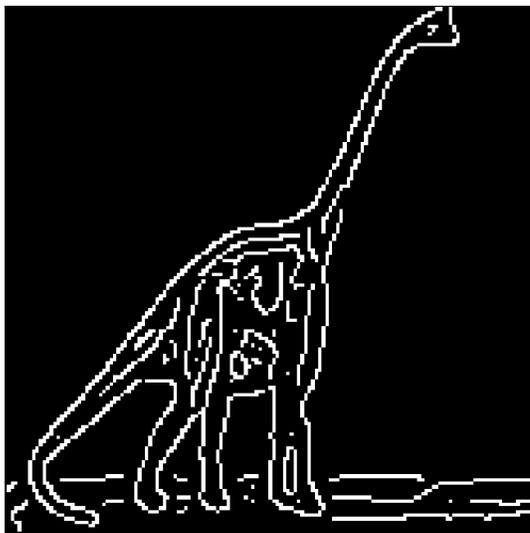
This work was tested using images from the dinosaur class and horse class of images in the SIMPLIcity data set [50] [51]. Using the same features, both set pattern method of classification and K-means method of classification, using the similarity measure discussed in section 3.4.1 were used for this experiment.

One of the most common application of K-means clustering process described in section 2.5 is similarity grouping. Thus, it will be ideal to compare our work with this well-known process. Clusters here are considered as patterns in the digital image data and thus are used for subsequent pattern recognition and classification. Figure 4.8 shows the edge map of '473.jpg' image using canny's method of edge detection [53] as in figure 4.8(a), and patterns generated in '473.jpg' image, using the set pattern generation method and the K-means clustering method. Also edge gradient directions were used as a probe function, for the pattern generation using both of the methods.

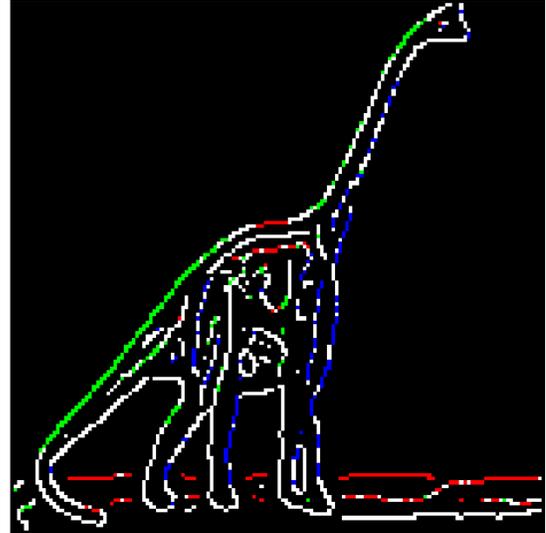
Figure 4.8(b) shows the resulting pattern generated, using the set pattern discovery method discussed in chapter 3. Here, three seed motifs were used in the pattern generation, and modal points were used as a basis of selection of seed motifs. It can be seen that the pattern generated here are the blue, green and red edge points only. The remaining white edge points are not part of the set pattern generated.

Figure 4.8(c) shows the resulting pattern generated using the K-means method. Here, three clusters were generated ($k=3$) and Euclidean distance was used for calculating the distance in the description of the points in the image. It can be seen that the pattern generated here are the yellow, blue and red edge points. All the edge points make up the pattern.

These patterns generated are used as inputs for the image classification, using a similarity measure as discussed in section 3.4.1 and illustrated in 4.3.1.



(a) Edge map



(b) Set pattern



(c) K-means pattern

Figure 4.8: Patterns generated on edge maps using, both set pattern & K-means method of pattern generation.

For the two methods, in each round, 100 randomly selected test images (50 from each class) were compared and classified as similar to or different from a randomly selected query image. 30 randomly selected query images from each class (30 dinosaur query images and 30 horse query images) were used for this experiment. Both the set pattern method of classification and K-means method of classification were used for the experiment. The sensitivity and specificity analysis of the result obtained; results in terms of correctly classified test

images and the average time taken to classify a test image are discussed below.

4.4.1 Sensitivity and Specificity Analysis of Result

Sensitivity and specificity are statistical measures of a binary classification test. In this work, classification can be seen as a binary classification test, since the test images are either of the same class or different class from the query image.

In classifying a test image, each test image can either be classified as in the same class or in a different class with the query image. If the outcome of the classification is positive, it implies that the test image is classified as in the same class with the query image, however this result can be false or true. If the outcome of the classification is negative, it implies that the test image is classified as in a different class with the query image, however this result can also be false or true.

Thus sensitivity which refers to the ability for the classification algorithm to correctly classify test images of the same class with the query image is given by

$$\text{Sensitivity} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}}. \quad (4.4.1)$$

A classification with 100% sensitivity correctly classifies all test images, in the same class with the query image.

While specificity which refers to the ability of the classification algorithm to correctly classify test image in a different class with the query image is given by

$$\text{Specificity} = \frac{\text{True negative}}{\text{True negative} + \text{False positive}}. \quad (4.4.2)$$

A classification with 100% specificity correctly classifies all test images that are not in the same class with the query image.

When testing images from both dinosaur and horse classes, using an image from the dinosaur class of images as a query image, the resulting terms that characterize sensitivity and specificity measures can be defined as follows.

1. True positive: A dinosaur test image is correctly classified as in the same class with the dinosaur query image.
2. False positive: A horse test image is incorrectly classified as in the same class with the dinosaur query image.
3. True negative: A horse test image is correctly classified as in a different class with the dinosaur query image.
4. False negative: A dinosaur test image is incorrectly classified as in a different class with the dinosaur query image.

A table showing these terms as well as the resulting sensitivity and specificity for classification of 30 rounds of 100 test images (50 of each class), using different randomly selected dinosaur query images is shown in table 4.1 and 4.2 for the set pattern method of classification and K-means method of classification respectively.

Table 4.1: Sensitivity and specificity measures for set pattern method of classification testing using dinosaur query images.

| Query Images | True Positive | False Negative | False Positive | True Negative | Total correct | Sensitivity | Specificity |
|--------------|---------------|----------------|----------------|---------------|---------------|-------------|-------------|
| 468.jpg | 49 | 1 | 0 | 50 | 99 | 98% | 100% |
| 485.jpg | 33 | 17 | 36 | 14 | 47 | 66% | 28% |
| 413.jpg | 41 | 9 | 0 | 50 | 91 | 82% | 100% |
| 465.jpg | 43 | 7 | 3 | 47 | 90 | 86% | 94% |
| 489.jpg | 48 | 2 | 0 | 50 | 98 | 96% | 100% |
| 495.jpg | 49 | 1 | 0 | 50 | 99 | 98% | 100% |
| 452.jpg | 40 | 10 | 6 | 44 | 84 | 80% | 88% |
| 460.jpg | 49 | 1 | 0 | 50 | 99 | 98% | 100% |
| 493.jpg | 45 | 5 | 0 | 50 | 95 | 90% | 100% |
| 441.jpg | 50 | 0 | 0 | 50 | 100 | 100% | 100% |
| 409.jpg | 47 | 3 | 0 | 50 | 97 | 94% | 100% |
| 447.jpg | 49 | 1 | 0 | 50 | 99 | 98% | 100% |
| 407.jpg | 42 | 8 | 0 | 50 | 92 | 84% | 100% |
| 439.jpg | 42 | 8 | 0 | 50 | 92 | 84% | 100% |
| 434.jpg | 42 | 8 | 0 | 50 | 92 | 84% | 100% |
| 459.jpg | 43 | 7 | 32 | 18 | 61 | 86% | 36% |
| 440.jpg | 50 | 0 | 0 | 50 | 100 | 100% | 100% |
| 442.jpg | 49 | 1 | 0 | 50 | 99 | 98% | 100% |
| 437.jpg | 49 | 1 | 0 | 50 | 99 | 98% | 100% |
| 473.jpg | 48 | 2 | 0 | 50 | 98 | 96% | 100% |
| 453.jpg | 50 | 0 | 0 | 50 | 100 | 100% | 100% |
| 490.jpg | 50 | 0 | 0 | 50 | 100 | 100% | 100% |
| 456.jpg | 46 | 4 | 0 | 50 | 96 | 92% | 100% |
| 466.jpg | 50 | 0 | 0 | 50 | 100 | 100% | 100% |
| 405.jpg | 50 | 0 | 0 | 50 | 100 | 100% | 100% |
| 461.jpg | 39 | 11 | 14 | 36 | 75 | 78% | 72% |
| 435.jpg | 45 | 5 | 0 | 50 | 95 | 90% | 100% |
| 419.jpg | 48 | 2 | 0 | 50 | 98 | 96% | 100% |
| 444.jpg | 48 | 2 | 11 | 39 | 87 | 96% | 78% |
| 462.jpg | 46 | 4 | 43 | 7 | 53 | 92% | 14% |

Table 4.2: Sensitivity and specificity measures for K-means method of classification testing using dinosaur query images.

| Query Images | True Positive | False Negative | False Positive | True Negative | Total correct | Sensitivity | Specificity |
|--------------|---------------|----------------|----------------|---------------|---------------|-------------|-------------|
| 468.jpg | 47 | 3 | 3 | 47 | 94 | 94% | 94% |
| 485.jpg | 24 | 26 | 38 | 12 | 36 | 48% | 24% |
| 413.jpg | 38 | 12 | 0 | 50 | 88 | 76% | 100% |
| 465.jpg | 37 | 13 | 0 | 50 | 87 | 74% | 100% |
| 489.jpg | 46 | 4 | 3 | 47 | 93 | 92% | 94% |
| 495.jpg | 47 | 3 | 1 | 49 | 96 | 94% | 98% |
| 452.jpg | 39 | 11 | 8 | 42 | 81 | 78% | 84% |
| 460.jpg | 48 | 2 | 2 | 48 | 96 | 96% | 96% |
| 493.jpg | 41 | 9 | 1 | 49 | 90 | 82% | 98% |
| 441.jpg | 50 | 0 | 1 | 49 | 99 | 100% | 98% |
| 409.jpg | 46 | 4 | 2 | 48 | 94 | 92% | 96% |
| 447.jpg | 49 | 1 | 0 | 50 | 99 | 98% | 100% |
| 407.jpg | 40 | 10 | 0 | 50 | 90 | 80% | 100% |
| 439.jpg | 39 | 11 | 2 | 48 | 87 | 78% | 96% |
| 434.jpg | 40 | 10 | 2 | 48 | 88 | 80% | 96% |
| 459.jpg | 29 | 21 | 28 | 22 | 51 | 58% | 44% |
| 440.jpg | 47 | 3 | 0 | 50 | 97 | 94% | 100% |
| 442.jpg | 47 | 3 | 1 | 49 | 96 | 94% | 98% |
| 437.jpg | 48 | 2 | 1 | 49 | 97 | 96% | 98% |
| 473.jpg | 44 | 6 | 1 | 49 | 93 | 88% | 98% |
| 453.jpg | 45 | 5 | 1 | 49 | 94 | 90% | 98% |
| 490.jpg | 50 | 0 | 2 | 48 | 98 | 100% | 96% |
| 456.jpg | 45 | 5 | 2 | 48 | 93 | 90% | 96% |
| 466.jpg | 46 | 4 | 3 | 47 | 93 | 92% | 94% |
| 405.jpg | 50 | 0 | 2 | 48 | 98 | 100% | 96% |
| 461.jpg | 36 | 14 | 21 | 29 | 65 | 72% | 58% |
| 435.jpg | 42 | 8 | 1 | 49 | 91 | 84% | 98% |
| 419.jpg | 46 | 4 | 3 | 47 | 93 | 92% | 94% |
| 444.jpg | 41 | 9 | 2 | 48 | 89 | 82% | 96% |
| 462.jpg | 38 | 12 | 28 | 22 | 60 | 76% | 44% |

It can be seen from tables 4.1 and 4.2, that the true positive rates (sensitivity) and true negative rates (specificity) of both methods are very high.

However set pattern method of classification with an average of 92% sensitivity outperformed the K-means method of classification with an average of 85.67% sensitivity. Thus

set pattern method of classification correctly classifies test images in the same class with the dinosaur query image, better than the K-means method of classification.

Also, set pattern method of classification with an average of 90.33% specificity outperformed the K-means method of classification with an average of 89.4% specificity. Thus set pattern method of classification correctly classifies test images in different class with the dinosaur query image, better than the K-means method of classification.

When testing images from both dinosaur and horse classes, using an image from the horse class of images as a query image, the resulting terms that characterize sensitivity and specificity measures can be defined as follows.

1. True positive: A horse test image is correctly classified as in the same class with the horse query image.
2. False positive: A dinosaur test image is incorrectly classified as in the same class with the horse query image.
3. True negative: A dinosaur test image is correctly classified as in a different class with the horse query image.
4. False negative: A horse test image is incorrectly classified as in a different class with the horse query image.

A table showing these terms as well as the resulting sensitivity and specificity for classification of 30 rounds of 100 test images (50 of each class), using different randomly selected horse query images is shown in table 4.3 and 4.4 for the set pattern method of classification and K-means method of classification respectively.

Table 4.3: Sensitivity and specificity measures for set pattern method of classification testing using horse query images.

| Query Images | True Positive | False Negative | False Positive | True Negative | Total correct | Sensitivity | Specificity |
|--------------|---------------|----------------|----------------|---------------|---------------|-------------|-------------|
| 711.jpg | 49 | 1 | 0 | 50 | 99 | 98% | 100% |
| 774.jpg | 36 | 14 | 0 | 50 | 86 | 72% | 100% |
| 739.jpg | 43 | 7 | 2 | 48 | 91 | 86% | 96% |
| 786.jpg | 35 | 15 | 0 | 50 | 85 | 70% | 100% |
| 721.jpg | 44 | 6 | 9 | 41 | 85 | 88% | 82% |
| 703.jpg | 45 | 5 | 8 | 42 | 87 | 90% | 84% |
| 787.jpg | 6 | 44 | 0 | 50 | 56 | 12% | 100% |
| 780.jpg | 34 | 16 | 3 | 47 | 81 | 68% | 94% |
| 797.jpg | 38 | 12 | 0 | 50 | 88 | 76% | 100% |
| 768.jpg | 39 | 11 | 1 | 49 | 88 | 78% | 98% |
| 754.jpg | 47 | 3 | 5 | 45 | 92 | 94% | 90% |
| 736.jpg | 47 | 3 | 5 | 45 | 92 | 94% | 90% |
| 747.jpg | 48 | 2 | 2 | 48 | 96 | 96% | 96% |
| 777.jpg | 46 | 4 | 0 | 50 | 96 | 92% | 100% |
| 713.jpg | 26 | 24 | 0 | 50 | 76 | 52% | 100% |
| 738.jpg | 34 | 16 | 3 | 47 | 81 | 68% | 94% |
| 735.jpg | 46 | 4 | 1 | 49 | 95 | 92% | 98% |
| 778.jpg | 46 | 4 | 5 | 45 | 91 | 92% | 90% |
| 743.jpg | 30 | 20 | 1 | 49 | 79 | 60% | 98% |
| 751.jpg | 47 | 3 | 7 | 43 | 90 | 94% | 86% |
| 716.jpg | 37 | 13 | 0 | 50 | 87 | 74% | 100% |
| 722.jpg | 15 | 35 | 1 | 49 | 64 | 30% | 98% |
| 773.jpg | 40 | 10 | 0 | 50 | 90 | 80% | 100% |
| 769.jpg | 38 | 12 | 0 | 50 | 88 | 76% | 100% |
| 745.jpg | 42 | 8 | 2 | 48 | 90 | 84% | 96% |
| 718.jpg | 48 | 2 | 6 | 44 | 92 | 96% | 88% |
| 789.jpg | 46 | 4 | 3 | 47 | 93 | 92% | 94% |
| 726.jpg | 39 | 11 | 0 | 50 | 89 | 78% | 100% |
| 764.jpg | 43 | 7 | 1 | 49 | 92 | 86% | 98% |
| 720.jpg | 49 | 1 | 7 | 43 | 92 | 98% | 86% |

Table 4.4: Sensitivity and specificity measures for K-means method of classification testing using horse query images.

| Query Images | True Positive | False Negative | False Positive | True Negative | Total correct | Sensitivity | Specificity |
|--------------|---------------|----------------|----------------|---------------|---------------|-------------|-------------|
| 711.jpg | 47 | 3 | 2 | 48 | 95 | 94% | 96% |
| 774.jpg | 32 | 18 | 1 | 49 | 81 | 64% | 98% |
| 739.jpg | 43 | 7 | 2 | 48 | 91 | 86% | 96% |
| 786.jpg | 39 | 11 | 3 | 47 | 86 | 78% | 94% |
| 721.jpg | 43 | 7 | 10 | 40 | 83 | 86% | 80% |
| 703.jpg | 47 | 3 | 10 | 40 | 87 | 94% | 80% |
| 787.jpg | 6 | 44 | 1 | 49 | 55 | 12% | 98% |
| 780.jpg | 18 | 32 | 3 | 47 | 65 | 36% | 94% |
| 797.jpg | 40 | 10 | 2 | 48 | 88 | 80% | 96% |
| 768.jpg | 42 | 8 | 3 | 47 | 89 | 84% | 94% |
| 754.jpg | 44 | 6 | 2 | 48 | 92 | 88% | 96% |
| 736.jpg | 47 | 3 | 6 | 44 | 91 | 94% | 88% |
| 747.jpg | 44 | 6 | 2 | 48 | 92 | 88% | 96% |
| 777.jpg | 43 | 7 | 0 | 50 | 93 | 86% | 100% |
| 713.jpg | 41 | 9 | 4 | 46 | 87 | 82% | 92% |
| 738.jpg | 40 | 10 | 4 | 46 | 86 | 80% | 92% |
| 735.jpg | 48 | 2 | 1 | 49 | 97 | 96% | 98% |
| 778.jpg | 46 | 4 | 4 | 46 | 92 | 92% | 92% |
| 743.jpg | 38 | 12 | 1 | 49 | 87 | 76% | 98% |
| 751.jpg | 44 | 6 | 7 | 43 | 87 | 88% | 86% |
| 716.jpg | 44 | 6 | 5 | 45 | 89 | 88% | 90% |
| 722.jpg | 9 | 41 | 1 | 49 | 58 | 18% | 98% |
| 773.jpg | 39 | 11 | 1 | 49 | 88 | 78% | 98% |
| 769.jpg | 41 | 9 | 0 | 50 | 91 | 82% | 100% |
| 745.jpg | 42 | 8 | 2 | 48 | 90 | 84% | 96% |
| 718.jpg | 49 | 1 | 7 | 43 | 92 | 98% | 86% |
| 789.jpg | 44 | 6 | 5 | 45 | 89 | 88% | 90% |
| 726.jpg | 44 | 6 | 1 | 49 | 93 | 88% | 98% |
| 764.jpg | 44 | 6 | 1 | 49 | 93 | 88% | 98% |
| 720.jpg | 46 | 4 | 4 | 46 | 92 | 92% | 92% |

It can be seen from tables 4.3 and 4.4, that the true positive rates (sensitivity) and true negative rates (specificity) of both methods are very high.

Set pattern method of classification with an average of 95% specificity outperformed the K-means method of classification with an average of 93.67% specificity. Thus set

pattern method of classification correctly classifies test images in different class with the horse query image, better than the K-means method of classification.

However K-means method of classification with an average of 79.6% sensitivity slightly outperformed the set pattern method of classification with an average of 78.87% sensitivity. Thus K-means method of classification correctly classifies test images in the same class with the horse query image, better than the set pattern method of classification.

4.4.2 Correctly Classified Test Images

The graphs showing the resulting total number of correctly classified test images (addition of true positive and true negative) for all the setups considered for both methods are shown below.

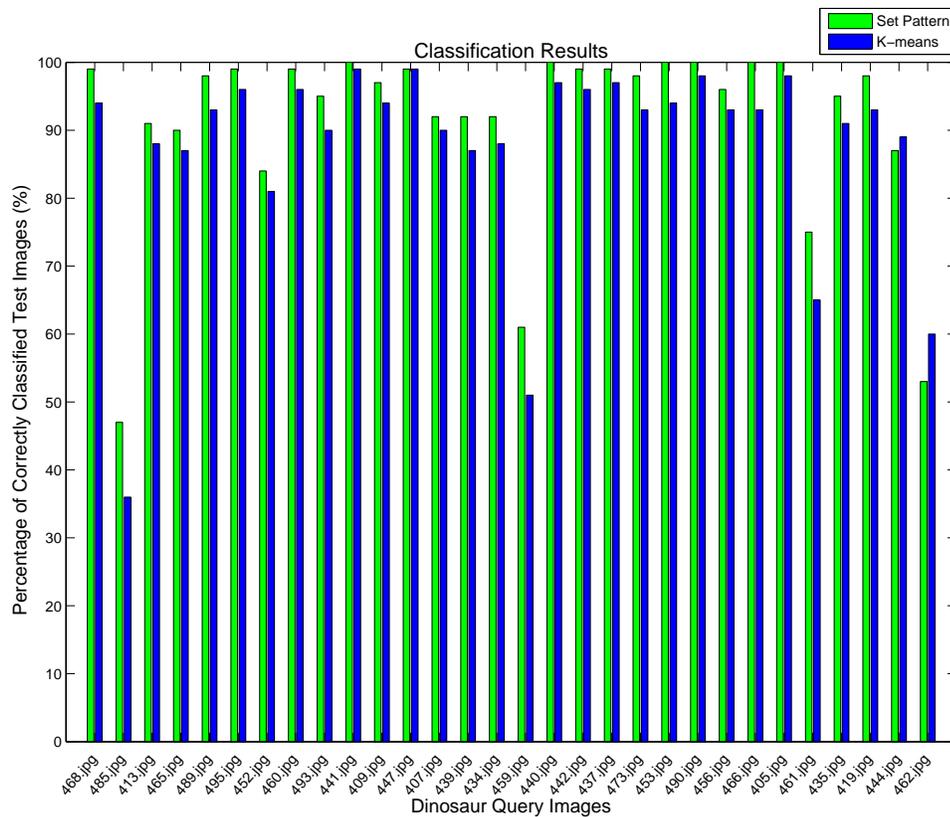


Figure 4.9: Percentage of correctly classified test images using 30 random dinosaur query images.

Figure 4.9 shows 30 rounds of classification of 100 randomly selected test images. In each round, a random query image from the dinosaur class of images in the data set is selected and 100 test images (50 from each of the classes under consideration) are classified with respect to this query image.

It can be seen in the figure that in 25 out of the 30 rounds, our set pattern method of classification outperformed the K-means method, in terms of number of correctly classified 100 test images. K-means method outperformed our set pattern method slightly in 2 rounds, while both methods gave the same result in 1 round.

In general, the set pattern method of classification correctly classified 91.1% of the test images correctly out of the 3000 test images. On the other hand, K-means method of classification correctly classified about 87.5% of the test images out of the 3000 test images.

This better performance in set pattern method of classification, is because set pattern generation uses only the salient edge points that characterize the image, in pattern generation, while K-means uses all the edge points to generate the image's pattern.

Also, it should be noted that patterns generated using the K-means method, changes by each run, since it assigns random points as its first centroids and as such can affect the classification results. Therefore, result of a particular classification might not be the same in each run for a particular test image, even when the same parameters are chosen for the classification. Whereas, in our set pattern method of classification, same results are obtained, given the same parameters for the classification, which include method of seed motif and other parameters.

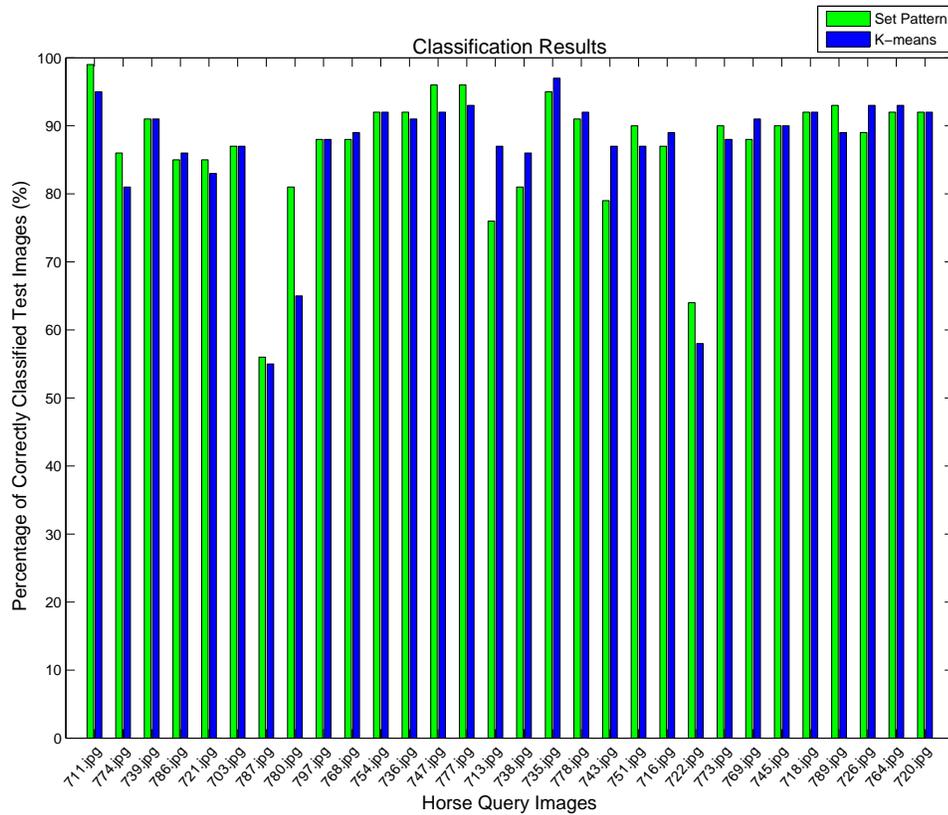


Figure 4.10: Percentage of correctly classified test images using 30 random horse query images.

Figure 4.10 shows 30 rounds of classification of 100 randomly selected test images. In each round, a random query image from the horse class of images in the data set is selected and 100 test images (50 from each of the classes under consideration) are classified with respect to this query image.

It can be seen in the figure that in 12 out of the 30 rounds, our set pattern method of classification outperformed the K-means method, in terms of number of correctly classified 100 test images. K-means method outperformed our set pattern method in 11 rounds, while both methods gave the same result in 7 rounds.

In general, the set pattern method of classification correctly classified 87% of the test images correctly out of the 3000 test images. On the other hand, K-means method of classifi-

cation correctly classified about 86.6% of the test images out of the 3000 test images.

Less rounds were won by set pattern method of classification in comparison with figure 4.9, because of the content of the horse images used as query images for the classification. The images in the horse class have different background like trees, bars, grasses, fences, as well as different number of horses in each of the images (ranges from 1 to 5). This makes it hard to successfully extract the useful foreground from the unwanted background in the image, without losing useful information in the foreground. Unlike the dinosaur images that have uniform backgrounds and only one dinosaur per image.

Therefore, in cases where set pattern method chooses seed motif points that are not part of the useful foreground, but that of the background, motif patterns generated would not characterize the image correctly, and using these patterns generated for classification will in turn give a false result.

Unlike the K-means method, where all the edge points (both useful foreground and useless background points) are used in the generation of the set pattern. Hence chances of better classification are better than using completely non useful points for the classification.

Therefore, having a uniform background while capturing the images used for this algorithm will improve the performance. This allows the foreground to be completely extracted from the background and only the useful foreground points that completely characterize the image will be used for extraction of seed motifs, generation of motif patterns and then subsequent classification.

Also the choice of the seed motif points greatly affects the overall performance of the algorithm. Seed motifs that characterize the image should be extracted and this is dependent on the application of the classification.

4.4.3 Average Time of Classification of a Test Image

One of the major advantages of set pattern method of classification is the fast classification time when compared to the K-means method. Since set pattern method does not necessarily use all the edge points in the image in its pattern generation, it means that less points are used for the calculation involved in the classification process. This gives rise to faster classification time.

Unlike the K-means method which uses the whole edge points for pattern generation and thus uses all the edge points for calculation that is involved in the classification. Also generation of patterns using the K-means method involves a lot of iterations till convergence is obtained. All these contributes to its slower classification time.

The average classification time for classifying a test image to a query image for all the setups considered are shown in the figures below.

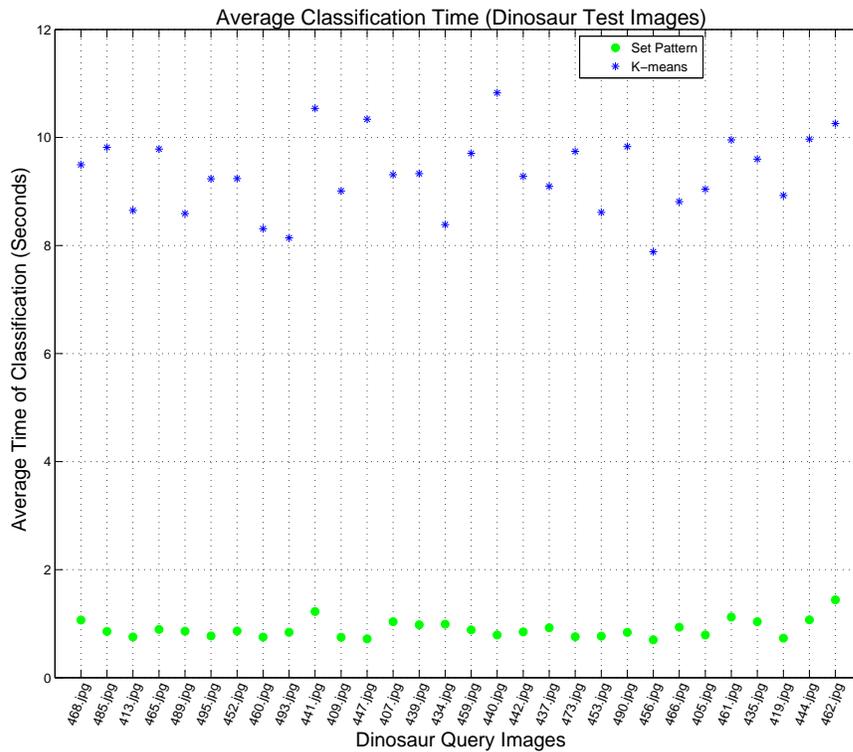


Figure 4.11: Average classification time dinosaur query image vs. dinosaur test images.

Figure 4.11 shows the average time of classification for each dinosaur test image, when classifying 50 dinosaur test images with respect to dinosaur query image. It shows the average time of classifying a test image in each of the 30 rounds of classification of randomly selected 50 dinosaur test images, against a randomly selected dinosaur query image. In each round, the average time of classification for each dinosaur test image, when classifying 50 dinosaur test images with respect to dinosaur query image is shown, for both set pattern and K-means method of classification.

It can be seen that, with set pattern’s method highest average classification time being 1.44 seconds, and that of K-means method being 10.82 seconds, set pattern method of classification is about 7.5 times faster than the K-means method.

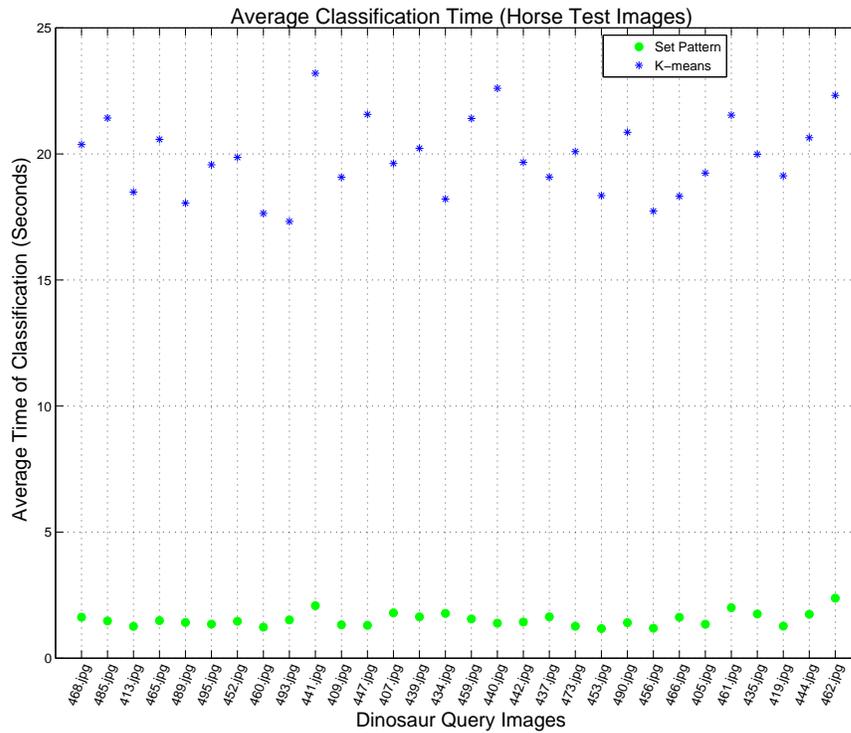


Figure 4.12: Average classification time dinosaur query image vs. horse test images.

Figure 4.12 shows the average time of classification for each horse test image, when classifying 50 horse test images with respect to dinosaur query image for both set pattern and K-means method of classification. It shows the average time of classifying a test image in each of the 30 rounds of classification of randomly selected 50 horse test images against a randomly selected dinosaur query image. In each round, the average time of classification for each horse test image, when classifying 50 horse test images with respect to dinosaur query image is shown, for both set pattern and K-means method of classification.

It can be seen that, with set pattern’s method highest average classification time being 2.38 seconds, and that of K-means method being 23.20 seconds, set pattern method of classification is about 9.7 times faster than the K-means method.

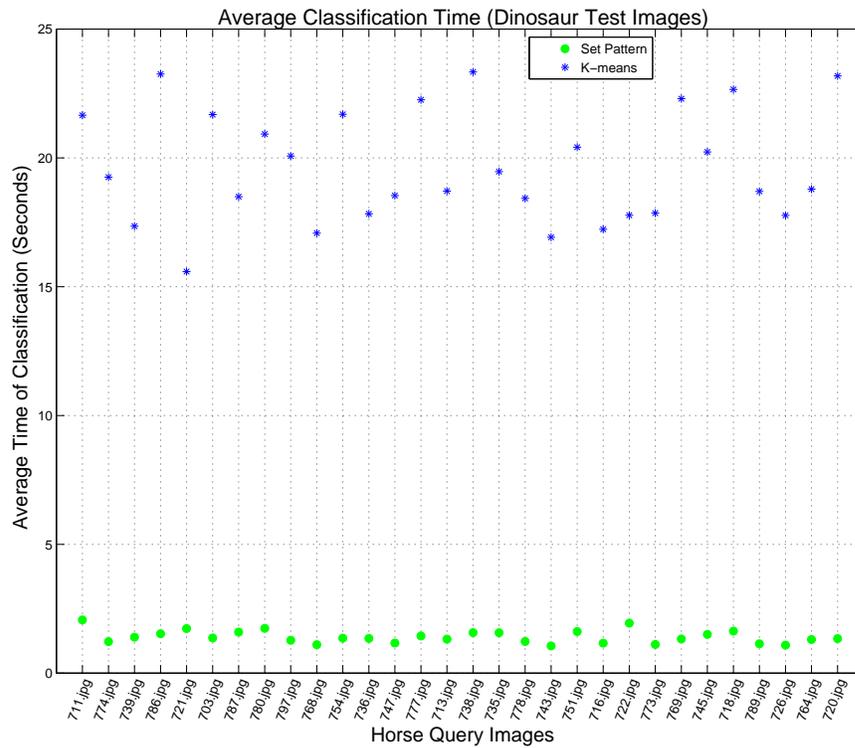


Figure 4.13: Average classification time horse query Image vs. dinosaur test images.

Figure 4.13 shows the average time of classification for each dinosaur test image, when classifying 50 dinosaur test images with respect to horse query image for both set pattern and K-means method of classification. It shows the average time of classifying a test image in each of the 30 rounds of classification of randomly selected 50 dinosaur test images against a randomly selected horse query image. In each round, the average time of classification for each dinosaur test image, when classifying 50 dinosaur test images with respect to horse query image is shown, for both set pattern and K-means method of classification.

It can be seen that, with set pattern's method highest average classification time being 2.07 seconds, and that of K-means method being 23.34 seconds, set pattern method of classification is about 11.3 times faster than the K-means method.

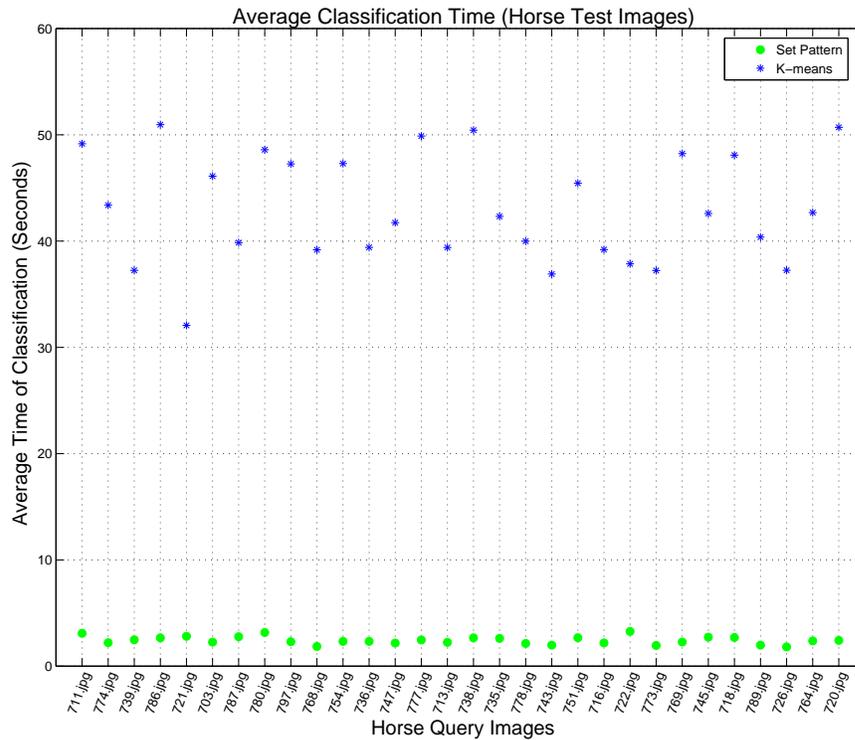


Figure 4.14: Average classification time horse query image vs. horse test images.

Figure 4.14 shows the average time of classification for each horse test image, when classifying 50 horse test images with respect to horse query image for both set pattern and K-means method of classification. It shows the average time of classifying a test image in each of the 30 rounds of classification of randomly selected 50 horse test images against a randomly selected horse query image. In each round, the average time of classification for each horse test image, when classifying 50 horse test images with respect to horse query image is shown, for both set pattern and K-means method of classification.

It can be seen that, with set pattern's method highest average classification time being 3.27 seconds, and that of K-means method being 50.96 seconds, set pattern method of classification is about 15.6 times faster than the K-means method.

For better illustration of the difference in classification time between these two methods

and also how the different classes of images affects the average classification time, the results are placed side by side.

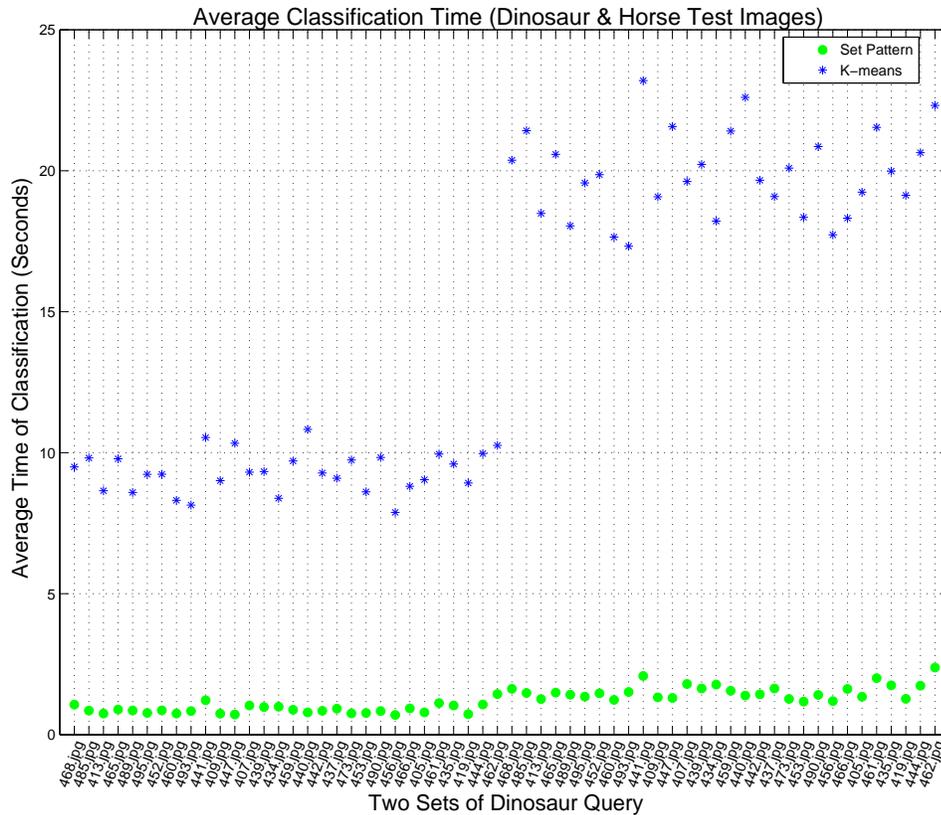


Figure 4.15: Average classification time dinosaur query image vs. dinosaur & horse test images.

Placing figures 4.11 and 4.12 side by side as seen in figure 4.15, shows how the average time varies, when testing images from the two classes with respect to the dinosaur query images, for both the set pattern and K-means method of classification.

It can be seen that for the set pattern method of classification, the average time of classifying dinosaur test images (first 30 rounds) and horse test images (last 30 rounds) with respect to the dinosaur query image are quite low. Also the time of classification for the test images of different classes are almost the same, with an insignificant increase when classifying test images from the horse class of images. This is because in set pattern method, salient points

extracted during pattern generation are used for the classification and although the edge points in the horse class of images are more, salient points are still extracted and used for classification.

However, it can be seen that for the K-means method of classification, the average time of classifying dinosaur test images (first 30 rounds) and horse test images (last 30 rounds) with respect to the dinosaur query image are quite high. Also the time of classification for the test images of different classes varies a lot, with a very significant 2.1 times increase in time, when classifying test images from the horse class of images. This is because there are a lot of edge points in the horse class of images, as a result of different objects in the background and the number of horses in the foreground (up to 5 horses in an image) and in K-means method of classification, all the edge points are used during pattern generation and subsequent classification, thus increasing the classification time.

Placing figures 4.13 and 4.14 side by side as seen in figure 4.16, shows how the average time varies, when testing images from the two classes with respect to the horse query images, for both the set pattern and K-means method of classification.

It can be seen that for the set pattern method of classification, the average time of classifying dinosaur test images (first 30 rounds) and horse test images (last 30 rounds) with respect to the horse query image are quite low. Also the time of classification for the test images of different classes are almost the same, with an insignificant increase when classifying test images from the horse class of images. This is because in set pattern method, salient points extracted during pattern generation are used for the classification and although the edge points in the horse class of images are more, salient points are still extracted and used for classification.

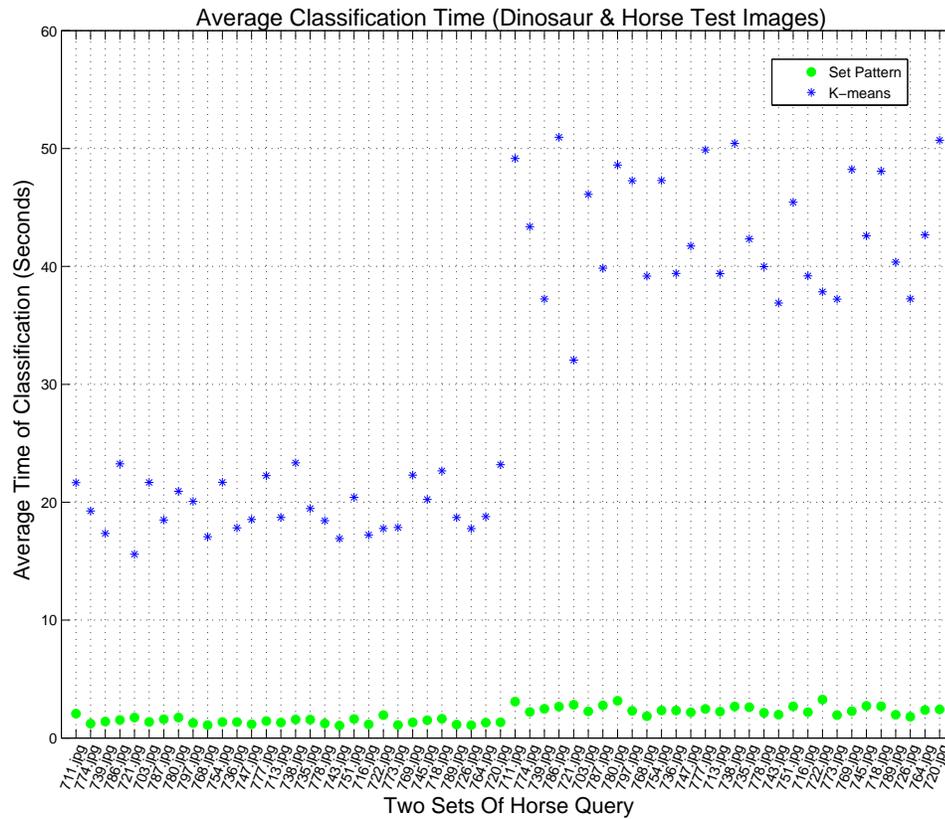


Figure 4.16: Average classification time horse query image vs. dinosaur & horse test images.

However, it can be seen that for the K-means method of classification, the average time of classifying dinosaur test images (first 30 rounds) and horse test images (last 30 rounds) with respect to the horse query image are very high. Also the time of classification for the test images of different classes varies a lot, with a very significant 2.2 times increase in time, when classifying test images from the horse class of images. This is because there are a lot of edge points in the horse class of images, as a result of different objects in the background and the number of horses in the foreground (up to 5 horses in an image). And in K-means method of classification, all the edge points are used during pattern generation and subsequent classification, thus increasing the classification time.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The research work reported in this thesis, focused on the extraction of visual set patterns in digital images, using nearness and similarity measures as well as descriptive properties, leading to unique pattern discoveries. These unique patterns discovered were then used for binary classification, using a unique similarity distance measure.

In set pattern method of pattern generation discussed in this thesis, only salient points that characterize the digital image, make up the image pattern. Hence, a new theorem was discovered and discussed in section 3.2.3. Also, given the same parameters, the patterns generated by the proposed method do not change, since it is not random in selecting seed points, as opposed to techniques such as K-means clustering.

Furthermore, new similarity distance measure for binary classification was introduced in this thesis, using the concept of the angle between vectors using their dot product.

The application of the unique patterns generated in this work to binary classification, showed 89.1% successful binary classification at a very fast computational speed and on the average, very high specificity and sensitivity.

Since classification is a supervised learning technique, thresholds were introduced in order to find a criteria for successful classification. These thresholds were chosen with respect to the images in the classes of images used for the classification. It is important to note that a change in these thresholds can either affect the sensitivity or (and) specificity of the classification results.

5.2 Recommendation and Future Work

This application of nearness measure in pattern generation can be applied to medical image analysis (medical testing). Depending on the application, unique methods of seed motif selection can be employed and a combination of neighbourhoods discussed in section 2.4 can be used for the motif generation. For example, adherent point method of seed motif selection can be used, where the spherical neighbourhood of an adherent point (seed motif) characterizes that digital medical image, i.e., contains salient points that distinguishes the image. For instance, a black spot with a surrounding redness. The black spot here, will be the adherent point and the surrounding redness will be the spherical neighbourhood which makes up the motif pattern.

Depending on the application of the patterns generated, the background of the images used for the experiment can be set up to be uniform and distinguished from the foreground. Thus reducing unnecessary points from the background (noise), which might be used for the pattern generation. Keeping in mind that some of the relevant points in the digital image can be lost in the bid to remove the background during image processing, it will be better to avoid this in the initial stages of image capturing if it is possible.

Also, other methods of seed motif generation, which leads to discovery of salient points that make up a pattern, and thus characterize an image, can be further explored.

Appendix A

Threshold Selection

The mathematical breakdown of how the threshold ϵ_1 and ϵ_2 , were selected for this experiment, are explained below.

Recall that orthogonal vectors are perpendicular non zero vectors, in which the angle between the two vectors is 90° , thus the cosine of the angle between them is 0. If the cosine of the angle θ between two vectors $V1$ and $V2$ is 0, that is $\cos\theta = 0$, it implies that vectors $V1$ and $V2$ are very different, therefore completely dissimilar.

While collinear vectors are parallel non zero vectors, in which the angle between the two vectors is 0° , thus the cosine of the angle between them is 1. If the cosine of angle θ between two vectors $V1$ and $V2$ is 1, that is $\cos\theta = 1$, it implies that vectors $V1$ and $V2$ are the same and parallel.

Based on this knowledge, ϵ_1 is chosen such that the $\cos\theta$ is 0.95, which is close to 1, thus $V1$ and $V2$ can be said to be similar. Therefore ϵ_1 was selected as shown below.

$$\cos \theta = 0.95$$

$$\theta = \arccos(0.95)$$

$$\theta = 18.19^\circ$$

$$\hat{\theta} = \frac{18.19}{90}$$

$$\hat{\theta} = 0.2$$

$$\rightarrow \quad \therefore \epsilon_1 = 0.2$$

Note that ϵ_1 was not chosen as 0, because images of the same class are not exactly the same. However, belonging to the same class, they are similar to a great extent, that is why ϵ_1 was chosen in such a way that $\theta = 18.19^\circ$.

To tighten the similarity measure and make sure that only test images that are similar to a great extent are classified as same class, ϵ_2 was selected as shown below.

$$\cos \theta = 0.85$$

$$\theta = \arccos(0.85)$$

$$\theta = 31.78^\circ$$

$$\hat{\theta} = \frac{31.78}{90}$$

$$\hat{\theta} = 0.35$$

$$\rightarrow \quad \therefore \epsilon_2 = 0.35$$

Thus, if the similarity vector of the test image with respect to the query image is closer to ϵ_1 (18.19°) than it is to ϵ_2 (31.78°), then the test image is similar enough to be of the same class with the query image.

However, if the similarity vector of the test image with respect to the query image is closer to ϵ_2 (31.78°) than it is to ϵ_1 (18.19°), then the test image is not similar enough to be of the same class with the query image, but rather, it is of a different class with the query image.

Appendix B

User Manual for

Set_Pattern_Classification GUI

The nine panels in the ‘Set_Pattern_Classification’ MATLAB graphical user interface include:

Query Image Panel

The query image is loaded in this panel, by choosing an image via the file drop down menu. Snapshot of the *Query Image* panel is shown in figure B.1.

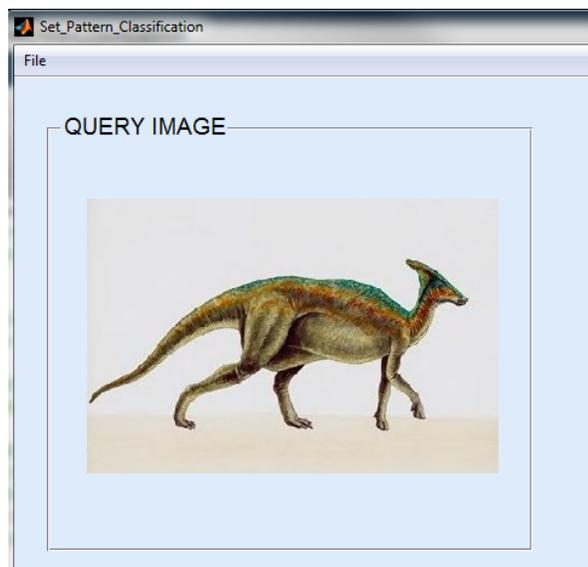


Figure B.1: Query Image.

Test Image Panel

The test image is loaded in this panel, by choosing an image via the file drop down menu. Snapshot of the *Test Image* panel is shown in figure B.2.

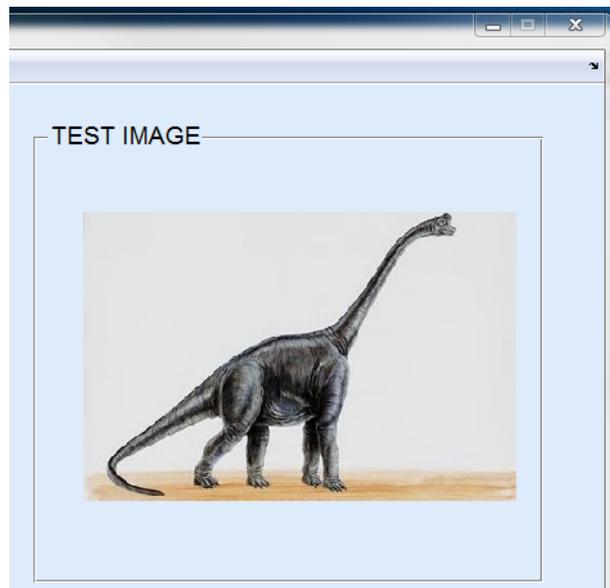


Figure B.2: Test Image.

Method of Classification Panel

The method of classification is chosen here. Either of the two methods, which include set pattern method of classification or K-means method of classification, is activated by clicking the radio buttons associated to them. Only one of the radio buttons must be chosen at a time for classification. Snapshot of the *Method of Classification* panel is shown in figure B.3.

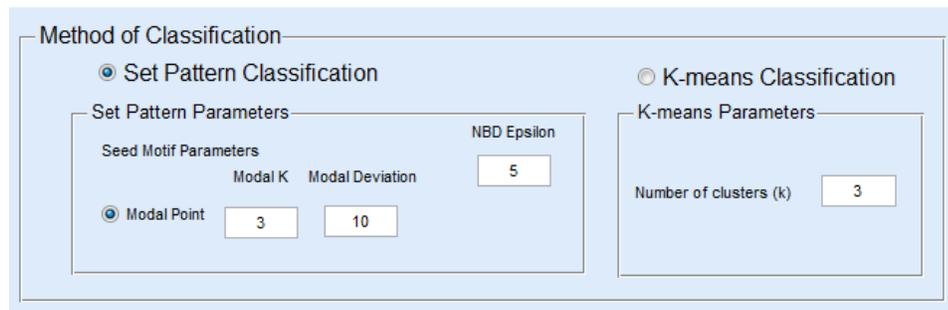
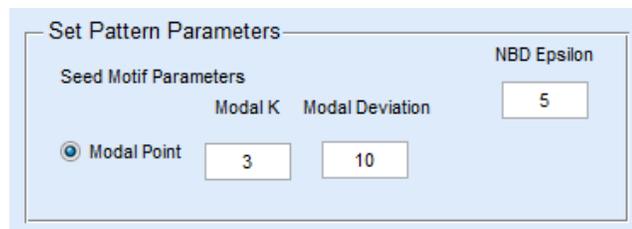
A screenshot of a software panel titled "Method of Classification". The panel is divided into two main sections. The left section is titled "Set Pattern Classification" and contains a sub-section "Seed Motif Parameters" with three input fields: "Modal K" (value 3), "Modal Deviation" (value 10), and "NBD Epsilon" (value 5). There is also a radio button labeled "Modal Point" which is selected. The right section is titled "K-means Classification" and contains a sub-section "K-means Parameters" with one input field: "Number of clusters (k)" (value 3). Both sections have radio buttons at the top, with "Set Pattern Classification" being selected.

Figure B.3: Method of Classification.

Set Pattern Parameters

The set pattern method of classification parameters are chosen here by first clicking the *Modal Point* radio button, which selects the modal point as a seed motif selection method. Then the next step is filling in values for the number of seed motifs expected in the *Modal K* edit text, the modal deviation in the *Modal Deviation* edit text and the motif pattern threshold in the *NBD Epsilon* edit text. To be able to input parameters in this panel, the *Set pattern Classification* radio button shown in B.3 should be clicked first. Snapshot of the *Set Pattern Parameters* panel is shown in figure B.4.

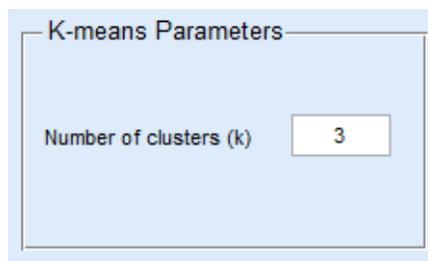


The screenshot shows a light blue panel titled "Set Pattern Parameters". Inside, there is a sub-section "Seed Motif Parameters". Below this, there are three radio buttons: "Modal Point" (selected), "Modal K", and "Modal Deviation". To the right of these radio buttons are three input fields: "Modal K" with the value "3", "Modal Deviation" with the value "10", and "NBD Epsilon" with the value "5".

Figure B.4: Set Pattern Parameters.

Parameters

The K-means method of classification parameters are chosen here, by filling in values for the number of patterns (clusters) expected in the *Number of clusters (k)* edit text. To be able to input parameters in this panel, the *K-means Classification* radio button shown in B.3 should be clicked first. Snapshot of the *K-means Parameters* panel is shown in figure B.5.



The screenshot shows a light blue panel titled "K-means Parameters". Inside, there is a single input field labeled "Number of clusters (k)" with the value "3".

Figure B.5: K-means Parameters.

Patterns in Query Image

The patterns generated in the query image are shown in this panel. Snapshot of the *Patterns in Query Image* panel is shown in figure B.6.

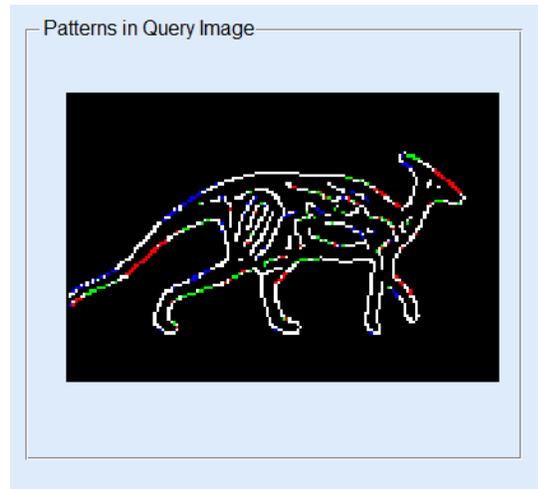


Figure B.6: Patterns in Query Image.

Patterns in Test Image

The patterns generated in the test image are shown in this panel. Snapshot of the *Patterns in Test Image* panel is shown in figure B.7.

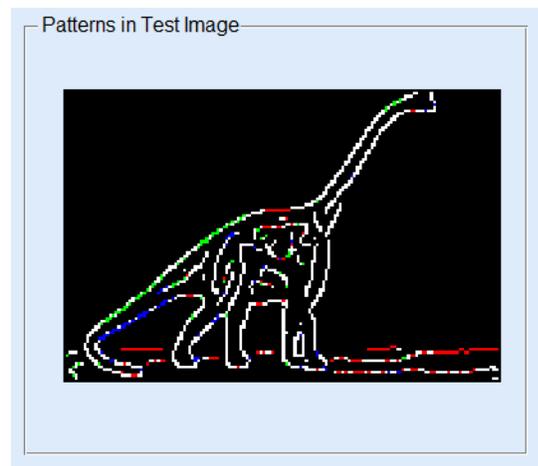


Figure B.7: Patterns in Test Image.

Similarity Threshold Range

Values for the two thresholds used for the similarity based classification are filled here. Value for the lower threshold ϵ_1 is filled in the *Epsilon 1* edit text and that of the higher threshold ϵ_2 is filled in the *Epsilon 2* edit text. Snapshot of the *Similarity Threshold Range* panel is shown in figure B.8.

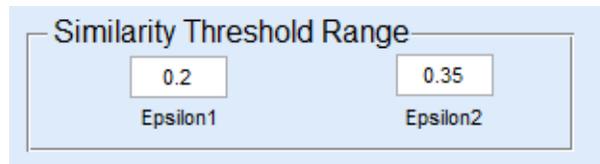


Figure B.8: Similarity Threshold Range.

Output Result

The result of the binary classification (**SAME CLASS** or **DIFFERENT CLASS**) is shown in this panel. Also the resulting distance of the thresholds and the similarity vector of the query image with respect to the test image is shown in this panel too. Snapshot of the *Output Result* panel is shown in figure B.9.

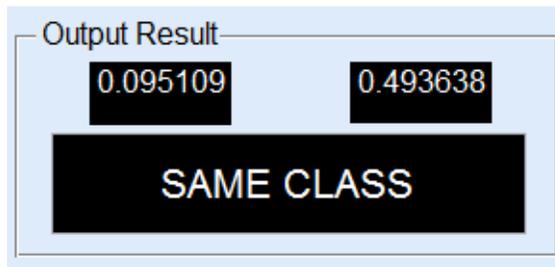


Figure B.9: Output Result.

References

- [1] M. Fahle and T. Poggio, *Perceptual Learning*. Cambridge, MA: The MIT Press, 2002.
- [2] D. Calitoui, B. J. Oommen, and D. Nussbaum, “Desynchronizing a chaotic pattern recognition neural network to model inaccurate perception,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 37, no. 3, pp. 692–704, 2007.
- [3] C. Henry, *Near Sets: Theory and Application*. Ph.d. thesis, University of Manitoba, 2010.
- [4] M. Merleau-Ponty, *Phenomenology of Perception*. Paris and New York: Smith, Calimard, Paris and Routledge & Kegan Paul, 1945.
- [5] J. Peters and S. Naimpally, “Applications of near sets,” *Notices Amer. Math. Soc.*, vol. 59, no. 4, pp. 536–542, 2012. Amer. Math. Soc. MR2951956.
- [6] S. Gupta and K. Patnaik, “Enhancing performance of face recognition systems by using near set approach for selecting facial features,” *Journal of Theoretical and Applied Information Technology*, vol. 4, no. 5, pp. 433–441, 2008.
- [7] A. E. Hassanien, A. Abraham, J. F. Peters, G. Schaefer, and C. Henry, “Rough sets and near sets in medical imaging: A review,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 3, no. 6, pp. 955–968, 2009.
- [8] J. F. Peters, “Tolerance near sets and image correspondence,” *International Journal of Bio-Inspired Computation*, vol. 1, no. 4, pp. 239–245, 2009.

- [9] A. H. Meghdadi, J. F. Peters, and S. Ramanna, “Tolerance classes in measuring image resemblance,” in *Knowledge-Based and Intelligent Information and Engineering Systems*, vol. 5712, (Santiago, Chile), pp. 127–134, 2009.
- [10] C. Henry and J. F. Peters, “Near set evaluation and recognition (near) system,” tech. rep., Computational Intelligence Laboratory, University of Manitoba, 2009. UM CI Laboratory Technical Report No. TR-2009-015.
- [11] J. F. Peters, S. Shahfar, S. Ramanna, and T. Szturm, “Biologically-inspired adaptive learning: A near set approach,” in *Frontiers in the Convergence of Bioscience and Information Technologies*, (Korea), 2007.
- [12] C. Henry and J. F. Peters, “Near set index in an objective image segmentation evaluation framework,” in *Proceedings of the GEOgraphic Object Based Image Analysis: Pixels, Objects, Intelligence*, (University of Calgary, Alberta), pp. 1–8, 2008.
- [13] C. Henry and J. F. Peters, “Perception image analysis,” *International Journal of Bio-Inspired Computation*, vol. 2, no. 2/3, p. *In Press*, 2010.
- [14] J. Peters and L. Puzio, “Image analysis with anisotropic wavelet-based nearness measures,” *Int. J. of Computational Intelligence Systems*, vol. 2, no. 3, pp. 168–183, 2009. doi 10.1016/j.ins.2009.04.018.
- [15] J. Peters, “How near are Zdzisław Pawlak’s paintings? study of merotopic distances between digital picture regions-of-interest,” in *Rough Sets and Intelligent Systems* (A. Skowron and Z. Suraj, eds.), pp. 89–114, Springer, 2012.
- [16] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, California, USA, 1967.
- [17] J. Peters, *Topology of Digital Images - Visual Pattern Discovery in Proximity Spaces*, vol. 63 of *Intelligent Systems Reference Library*. Springer, 2014. xv + 411pp, Zentralblatt MATH Zbl 1295 68010.

- [18] R. Klette, “Concise computer vision,” 2014.
- [19] L. Shapiro and G. C. Stockman, “Computer vision. 2001,” *ed: Prentice Hall*, 2001.
- [20] T. Morris, *Computer Vision and Image Processing (Cornerstones of Computing)*. Palgrave Macmillan, 2003.
- [21] D. H. Ballard and C. M. Brown, “Computer vision, 1982,” *Prentice-Hall, Englewood Cliffs, NJ*, 1982.
- [22] L. Barghout and J. Sheynin, “Real-world scene perception and perceptual organization: Lessons from computer vision,” *Journal of Vision*, vol. 13, no. 9, pp. 709–709, 2013.
- [23] W. E. L. Grimson and J. L. Mundy, “Computer vision applications,” *Communications of the ACM*, vol. 37, no. 3, pp. 44–51, 1994.
- [24] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, “Content-based multimedia information retrieval: State of the art and challenges,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, vol. 2, no. 1, pp. 1–19, 2006.
- [25] Z. Wang, Z. Chi, and D. Feng, “Fuzzy integral for leaf image retrieval,” in *Fuzzy Systems, 2002. FUZZ-IEEE’02. Proceedings of the 2002 IEEE International Conference on*, vol. 1, pp. 372–377, IEEE, 2002.
- [26] T. H. Painter, J. Dozier, D. A. Roberts, R. E. Davis, and R. O. Green, “Retrieval of subpixel snow-covered area and grain size from imaging spectrometer data,” *Remote Sensing of Environment*, vol. 85, no. 1, pp. 64–77, 2003.
- [27] M. Schroder, H. Rehrauer, K. Seidel, and M. Datcu, “Interactive learning and probabilistic retrieval in remote sensing image archives,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 38, no. 5, pp. 2288–2298, 2000.
- [28] J. F. Peters, “Classification of perceptual objects by means of features,” *International Journal of Information Technology & Intelligent Computing*, vol. 3, no. 2, pp. 1 – 35, 2008.

- [29] J. F. Peters, "Near sets. general theory about nearness of objects," *Applied Mathematical Sciences*, vol. 1, no. 53, pp. 2609–2629, 2007.
- [30] Y. Rui, T. S. Huang, and S.-F. Chang, "Image retrieval: Current techniques, promising directions, and open issues," *Journal of visual communication and image representation*, vol. 10, no. 1, pp. 39–62, 1999.
- [31] R. M. Haralick, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.
- [32] D. Pascale, "A review of rgb color spaces ...from xyy to r'g'b'," *The BabelColor Company*, pp. 1751–1759, 2003.
- [33] D. Chandrakala and S. Sumathi, "Application of artificial bee colony optimization algorithm for image classification using color and texture feature similarity fusion," *International Scholarly Research Notices*, vol. 2012, 2012.
- [34] H. Tamura, M. Shunji, and T. Yamawaki, "Textural features corresponding to visual perception," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, no. 6, pp. 460–473, 1978.
- [35] Mathworks, *Graycoprops*. Available at <http://www.mathworks.com/help/images/ref/graycoprops.html>.
- [36] S. A. Naimpally and B. D. Warrack, "Proximity spaces," in *Cambridge Tract in Mathematics No. 59*, Cambridge, UK: Cambridge University Press, 1970.
- [37] S. A. Naimpally, "Near and far. A centennial tribute to Frigyes Riesz," *Siberian Electronic Mathematical Reports*, vol. 6, pp. A.1–A.10, 2009.
- [38] Z. Pawlak and J. F. Peters, "Jak blisko (how near)," *Systemy Wspomagania Decyzji*, vol. I, pp. 57, 109., 2002.
- [39] J. F. Peters, "Near sets. special theory about nearness of objects," *Fundamenta Informaticae*, vol. 75, no. 1-4, pp. 407–433, 2007.

- [40] Z. Pawlak, "Classification of objects by means of attributes," Tech. Rep. PAS 429, Institute for Computer Science, Polish Academy of Sciences, 1981.
- [41] E. Orłowska, "Semantics of vague concepts. applications of rough sets," Tech. Rep. 469, Institute for Computer Science, Polish Academy of Sciences, 1982.
- [42] E. Orłowska, "Semantics of vague concepts," in *Foundations of Logic and Linguistics. Problems and Solutions* (G. Dorn and P. Weingartner, eds.), pp. 465–482, London/NY: Plenum Pres, 1985.
- [43] M. S. Pepe, *The statistical evaluation of medical tests for classification and prediction*. Oxford University Press, 2003.
- [44] C. A. Ball and A. E. Tschoegl, "The decision to establish a foreign bank branch or subsidiary: an application of binary classification procedures," *Journal of financial and Quantitative Analysis*, vol. 17, no. 03, pp. 411–424, 1982.
- [45] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*, vol. 1. Cambridge university press Cambridge, 2008.
- [46] R. S. Nickerson, "Binary-classification reaction time: A review of some studies of human information-processing capabilities," *Psychonomic Monograph Supplements*, 1972.
- [47] E. Čech, Z. Frolík, and M. Katetov, *Topological spaces*. Interscience Publishers, 1966.
- [48] F. Hausdorff, *Set theory*, vol. 119. American Mathematical Soc., 1957.
- [49] W. Rucklidge, *Efficient visual recognition using the Hausdorff distance*. Springer-Verlag New York, Inc., 1996.
- [50] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLicity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947–963, 2001.

- [51] J. Li and J. Z. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1075–1088, 2003.
- [52] R. Jain, R. Kasturi, and B. G. Schunck, *Machine vision*, vol. 5. McGraw-Hill New York, 1995.
- [53] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.
- [54] T. Pavlidis, "Analysis of set patterns," *Pattern Recognition*, vol. 1, no. 2, pp. 165–178, 1968.
- [55] U. Grenander, "Foundations of pattern analysis(pattern recognition and analysis, discussing definitions, rules of image transformation, pure and deformed patterns, etc)," *Quarterly of Applied Mathematics*, vol. 27, pp. 1–55, 1969.
- [56] U. Grenander, *General pattern theory-A mathematical study of regular structures*. Clarendon Press, 1993.
- [57] J. F. Peters, A. Skowron, and J. Stepaniuk, "Nearness of visual objects. application of rough sets in proximity spaces," *Fundamenta Informaticae*, vol. 128, no. 1, pp. 159–176, 2013.
- [58] W. H. Young and G. C. Young, *The theory of sets of points*, vol. 2. Cambridge, 1906.
- [59] F. W. Stentiford, "Saliency identified by absence of background structure," in *IS&T/SPIE Electronic Imaging*, pp. 86510V–86510V, International Society for Optics and Photonics, 2013.