Near Sets in Pattern Similarity Distance Based Classification

by

Diba Vafabakhsh

A Thesis

submitted to the Faculty of Graduate Studies of

The University of Manitoba

in Partial Fulfilment of the Requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg, Manitoba, R3T 5V6, Canada

Abstract

This research is focused on studying the nearness theory, various neighbourhoods of points in proximity spaces and similarity measures which leads to discovering the set patterns in digital images. The problem considered in this thesis is set pattern discovery using topology of digital images and nearness theory and then using the extracted set patterns as the basis for the classification method that is introduced.

Visual pattern is the repetition of some forms in a digital image. A visual set pattern is defined as a collection of sets that all of the members of the collection have common features and properties and are all descriptively near a given set called the pattern generator or motif set which itself is a bounded descriptive neighbourhood of a distinguished point of interest in the image. Using the generated set patterns, a classification method is introduced based on the set patterns similarity distance.

Keywords: nearness theory, neighbourhoods, proximity space, feature vector, probe function, spatial, descriptive, pre-processing, set pattern, set pattern generation, motif, extraction, pattern, pattern recognition, similarity measure, distance, pattern similarity distance, classification, saliency, k-means.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor Prof. James F. Peters for accepting me as his graduate student and introducing me to the topic of this thesis and also for his advice and guidance throughout all the stages of this research program which made the completion of this work possible.

I would like to thank my examination committee members, Dr. Carl Ho and Dr. Luming Wang for accepting to serve on my committee and for generously offering their time and support.

In addition, I would like to extend my sincere thanks to my friend, Mrigank Rochan whose advice and help was invaluable to me.

I wish to offer my heartfelt gratitude to my friend Shervin for his never ending encouragement, for inspiring me and for being so supportive.

Last but not the least; I would like to thank my parents for believing in me and their unconditional love, encouragement, constant support and countless sacrifices.

Table of Contents

List of Figures

Chapter 1

Introduction

Humans are gifted with eyes as sensors that collect 70% of our information through viewing the world and the output of multimedia systems around us [1]. In fact, visual perception is the ability to see and interpret the visual information in the environment by analysing the reflected visible light by the objects. This analysis is performed by human visual system which is like an optical system and can perform a number of image processing tasks even better than the way computers do [2]. Computer vision is the science that tries to give a similar capabilities as human vision to computers and machines by extracting, analysing and understanding the information from images. Computer vision systems emulate human visual perception and they focus on extracting useful information from images [3] [4]. Computer vision science is the study of digital image structures and patterns and it uses image processing as a tool for pursuing its goal [4].

In this research, computer vision science and image processing techniques have been used for studying the digital images and eventually classifying them using the proposed method. Nearness theory, various neighbourhoods of points in proximity spaces and similarity measures are studied which led to discovering the set patterns in digital images and eventually classifying the images.

The idea of computer vision emulating human perception arose the idea of nearnee theory and near sets which are disjoint sets that resemble each other [5]. The two types of near sets considered in this work are spatially near sets and descriptively near sets [6]. J. F. Peters in [7] has defined spatially near sets as sets that contain points identified by their location and have at least one point in common and descriptively near sets as sets that contain non-abstract points that have both locations and measurable features such as colour and gradient orientation. Connectedness, boundedness, convexity, shapes and shape theory are principal topics in the study of nearness and separation of physical as well as abstract sets. Near sets have a wide variety of applications from feature extraction and different classification methods to image morphology and segmentation methods [8] [9] [10] [11].

In this research, to achieve the goal which is the image classification, near sets played an important role. Visual set patterns were generated as the basis for the proposed classification method, using the nearness theory. According to Peters in [7], a visual set pattern is a collection of sets that all of the members of the collection have common features and properties and are all descriptively near a given set called the pattern generator or motif set. Motif set is a bounded descriptive neighbourhood of a distinguished point of interest in the image. Using the generated set patterns, a classification method is introduced based on the set patterns similarity distance. Set patterns generated in this work produced a successful classification result with a fast computational time.

1.1 Thesis Arrangement

This thesis is organized as follows:

- Chapter 1 contains the introduction of the thesis and also the chapters arrangement.
- Chapter 2 presents a review of related literature in this area of work and serves as the framework and guideline for what is done in the thesis.
- Chapter 3 explains the methodology used in this work for generating set patterns and performing the classification using pattern-similarity-distance.

- Chapter 4 presents the data set, the step-by-step experiments that were carried out in this research, results that were obtained in this work and data analysis.
- Chapter 5 discusses the thesis conclusion and suggests future works and directions.
- Appendix A offers a user manual for the developed GUI.

Chapter 2

State-of-the-Art and Literature Review

This chapter introduces some of the techniques of digital image processing, as well as images structures and various neighbourhoods of points in the topology of digital images and it explains how the discovery of these structures leads to pattern recognition in digital images. Some literature review and related researches on near sets and pattern recognition in digital images, using topological approaches are presented, as well.

2.1 Computer Vision

As humans, although most of us have little analytical understanding of visual cognition as a process, we still can solve many visual problems with apparent ease, since much of the human brain is dedicated to vision. Even though the richness of human imagination is not yet matched by engineering, there has been remarkable progress in the science of computer vision [12]. Grimson in [13] defined computer vision as a science that provides a method for understanding how to make intelligent decisions about an environment, on the basis of sensory inputs. the principal goal of computer vision is to reconstruct and interpret natural scenes based on the properties of the structures present in the scene or the content of images captured by digital cameras [4] [14] and also to understand the complex visual processes and to construct effective computer-based visual systems [15]. These systems usually receive a scene in terms of large arrays of digitized sensory information and after extracting the scene parameters, they use them to match against known-object models to support tasks such as recognition. They are expected to provide a description of the scene or the information relevant to the specific goals of the system as the output [16] [13]. computer vision employs image processing, pattern recognition and many other techniques to achieve its goals.

In recent years, computer vision became a key technology in many fields and today is being used in a wide variety of real-world applications, from applications for mobile phones or driver assistance for cars, to quality or process control applications [17].

Some of these applications mentioned in [14] include:

- **character recognition (OCR)**: reading handwritten postal codes on letters and automatic number plate recognition (ANPR) [18];
- **Machine inspection**: rapid parts inspection for quality assurance using stereo vision with specialized illumination to measure tolerances on aircraft wings or auto body parts or looking for defects in steel castings using X-ray vision;
- **Surveillance**: monitoring for intruders, analyzing highway traffic and monitoring pools for drowning victims;
- **Fingerprint recognition and biometrics**: for automatic access authentication as well as forensic applications.
- **Medical imaging**: registering pre-operative and intra-operative imagery or performing long-term studies of people's brain morphology as they age;

2.2 Euclidean Distance Metric

Euclidean distance is one of the most commonly used means of measuring distance. In Euclidean space $\mathbb{R}^n$, the straight line distance between two points is known as Euclidean distance. The elements of $\mathbb{R}^n$ are points (or also called vectors), each with n coordinates. According to J. F. Peters in [11], let $x, y \in \mathbb{R}^n$ with n coordinates and $x = (x_1, ..., x_n)$, $y = (y_1, ..., y_n)$. The norm of $x \in \mathbb{R}^n$ which is the vector length from the origin, is calculated by

$$\|x\| = \sqrt{x_1^2 + x_2^2 + ... + x_n^2}. \tag{2.1}$$

Euclidean norm $\|x - y\|$ is the distance between vector x and y, known as Euclidean distance [19], and is given by

$$\|x - y\| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}. \tag{2.2}$$

2.3 Neighbourhood of a Point

J.F Peters in [7] defined the neighbourhood of a point x in a non-empty set X as a set of points that are sufficiently near x. Based on this definition, point y belongs to a neighbourhood of point x, provided y is sufficiently near x. According to Peters's definition, sufficiently nearness of points x and y means that the distance between x and y is less than some number $\varepsilon \in (0, \infty]$, where distance is measured by some distance function [1]. Neighbourhoods can be either spatial (spherical) or descriptive. In [7], Peters defined these two types of neighbourhoods as follows:

2.3.1 Spherical Neighbourhood of a picture Point

A spherical neighbourhood of a point is defined in terms of the distance between points (pixels). The definition that Peters presented in [7] for spherical neighbourhood is as follows,

[1] See Section 2.2

Definition 1. *let X be a metric space, $\varepsilon \in (0, \infty]$ a bound on neighbourhood distances of points from a center $x \in X$ and d a metric. A spherical neighbourhood of a point x (N_x) is defined by*

$$N_x = \{y \in X : d(x, y) < \varepsilon\}. \tag{2.3}$$

where ε is a threshold that indicates the boundary of the neighbourhood.

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

$$d(x, y) = |x - y|. \tag{2.4}$$

That is, $d(x, y)$ is the standard distance between x and y.

Figure 2.1 shows an example of a spherical neighbourhood of a point x in an image[2] (the red circle).

Figure 2.1: Spherical neighbourhood of point $x = (194, 304)$, ($r = 80$).

[2]I took this photo in a restaurant in Montreal, Summer 2017.

2.3.2 Descriptive Neighbourhoods of picture Points

A descriptive neighbourhood of a point is defined in terms of the distance between the features of the points (pixels), e.g. the distance between pixel intensity values. The definition that Peters presented in [7] for descriptive neighbourhood is as follows,

Definition 2. *A descriptive neighbourhood of a picture point x is a set of picture points, each with a description that matches the description of the neighbourhood centre x. Let X be a set of picture 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, using a set of probe functions Φ is denoted as $N_{\Phi(x)}$.*

Peters in [7] [11] introduced four different types of descriptive neighbourhoods, namely,

2.3.2.1 Unbounded Descriptive Neighbourhood of a Picture 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 x. That is, if $N_{\Phi(x)}$ is an unbounded descriptive neighbourhood of a picture point x in X, then every $y \in X$ with a description close to the description of x belongs to $N_{\Phi(x)}$. $N_{\Phi(x)}$ is defined by

$$N_{\Phi(x)} = \{y \in X : d(\Phi_{(x)}, \Phi_{(y)}) < \varepsilon\}. \tag{2.5}$$

Figure 2.2 shows an example of an unbounded descriptive neighbourhood of a point x. In this example, the probe function Φ is a set of probe functions representing pixel colours and so identifies the colour intensity value of each point y in the image X. The green dots on the image show the unbounded descriptive neighbourhood.

Figure 2.2: Unbounded descriptive neighbourhood of point $x = (194, 304)$, ($\varepsilon = 15$).

2.3.2.2 Bounded Descriptive Neighbourhood of a Picture 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 x. That is, if $N_{\Phi(x)}$ is a bounded descriptive neighbourhood of a picture point x in X, then every $y \in X$ within a fixed distance ε from x and with a description close to the description of x belongs to $N_{\Phi(x)}$. $N_{\Phi(x)}$ is defined by

$$N_{\Phi(x)} = \{y \in X : d(\Phi_{(x)}, \Phi_{(y)}) < \varepsilon \quad \& \quad d(x, y) < r\}. \tag{2.6}$$

Figure 2.3 shows an example of a bounded descriptive neighbourhood of a point x. In this example, the probe function Φ is a set of probe functions representing pixel colours and so identifies the colour intensity value of each point y in the image X. The green dots in the red circle show the bounded descriptive neighbourhood in this example.

Figure 2.3: Bounded descriptive neighbourhood of point $x = (194, 304)$, $(\varepsilon = 15, r = 80)$.

2.3.2.3 Indistinguishable Bounded Descriptive Neighbourhood of a Picture 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 x. That is, if $N_{\Phi(x)}$ is an indistinguishable bounded descriptive neighbourhood of a picture point x in X, then every $y \in X$ within a fixed distance from x and with a description that matches the description of x belongs to $N_{\Phi(x)}$. $N_{\Phi(x)}$ is defined by

$$N_{\Phi(x)} = \{y \in X : d(\Phi_{(x)}, \Phi_{(y)}) = 0 \quad \& \quad d(x, y) < r\}. \tag{2.7}$$

Figure 2.4 shows an example of an indistinguishable bounded descriptive neighbourhood of a point x. In this example, the probe function Φ is a set of probe functions representing pixel colours and so identifies the colour intensity value of each point y in the image X. The green dots in the red circle show the indistinguishable bounded descriptive neighbourhood in this example.

Figure 2.4: Indistinguishable bounded descriptive neighbourhood of point $x = (194, 304)$, $(\varepsilon = 0, r = 80)$.

2.3.2.4 Indistinguishable Unbounded Descriptive Neighbourhood of a Picture 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 x. That is, if $N_{\Phi(x)}$ is an indistinguishable unbounded descriptive neighbourhood of a picture point x in X, then every $y \in X$ with a description that matches the description of x belongs to $N_{\Phi(x)}$. $N_{\Phi(x)}$ is defined by

$$N_{\Phi(x)} = \{y \in X : d(\Phi_{(x)}, \Phi_{(y)}) = 0\}. \tag{2.8}$$

Figure 2.5 shows an example of an indistinguishable unbounded descriptive neighbourhood of a point x. In this example, the probe function Φ is a set of probe functions representing pixel colours and so identifies the colour intensity value of each point y in the image X. The green dots on the image show the indistinguishable unbounded descriptive neighbourhood in this example.

Figure 2.5: Indistinguishable unbounded descriptive neighbourhood of point $x = (194, 304)$, $(\varepsilon = 0)$.

2.3.3 Open Sets and Closed Sets

Open sets and closed sets can be considered as different types of neighbourhoods and can be considered for either spatial or descriptive neighbourhoods of points. Peters in [7] introduced these notions for spatial neighborhoods:

- **Open Neighbourhood**: An open neighbourhood N_x of a point x is an open set N_x defined by

$$N_x = \{y \in X : d(x, y) < \varepsilon\}. \tag{2.9}$$

- **Closed Neighbourhood**: A closed neighbourhood N_x of a point x is the closed set N_x defined by

$$N_x = \{y \in X : d(x, y) \leq \varepsilon\}. \tag{2.10}$$

2.4 Near Sets History

In 1908, F. Riesz introduced the concept of nearness of pairs of sets as the pioneer of this field [20]. Based on the history that J.F. Peters presented in [11] on near sets, S.A. Naimpally and B.D. Warrack intoduced the proximity spaces in 1970 [21]. They defined spatially near sets in terms of the distance between sets. The concept of near sets offers a framework for comparing and classifying the elements in sets based on their nearness and has a wide variety of applications in areas such as image processing, pattern recognition, mathematics, engineering and science [8].

According to [21], let X be a metric normed topological space with proximity relation δ, non-empty subsets A, $B \subset X$ and $\parallel a - b \parallel$ as the distance between points $a \in A$ and $b \in B$. The Čech distance [22][3] between subsets A and B is defined by

$$D(A, B) = \inf\{\parallel a - b \parallel : a \in A, b \in B\}. \tag{2.11}$$

Based on the traditional approach for defining the nearness of sets, sets A and B are **spatially near** ($A\ \delta\ B$), provided $D(A, B) = 0$. Peters in [11] considered the descriptive nearness of disjoint sets. Disjoint sets are defined as sets that are not spatially near but can be descriptively near. That is, disjoint sets A and B are descriptively near, provided there is at least one pair $a \in A$, $b \in B$ such that the description of a matches the description of b. In other words, sets are descriptively near, provided $A \underset{\Phi}{\cap} B \neq \varnothing$, which means the descriptive intersection of A and B is nonempty [25].

According to Peters in [11], a **probe function** maps an object such as a digital image pixel to a real number that is a characteristic feature value of the object. Features can be colour components, circularity, length, area, gradient magnitude, gradient direction, or simply the gray-level intensity value, depending on the application. Point-based probe functions were introduced in [25] as a vital component of near sets study. A point-based

[3]This type of distance calculates the greatest lower bound of the standard distances between pairs of set elements [23] [24].

probe function $\varphi : X \to \mathbb{R}$, $x \in A$ returns a feature value of x. The description of a point $x \in A$ is a feature vector ($\Phi(x)$),

$$(\varphi_1(x), ..., \varphi_i(x), ..., \varphi_n(x)), \quad \text{with} \quad \varphi_i : X \to \mathbb{R}. \tag{2.12}$$

In 2013, region-based probe function was introduced by A. Di Concilio in [26], which was influenced by Di Concilio's earlier study on proximities and quasi-metrics in point-free geometry [27] as well as G. Gerla's earlier work in that area [28]. In the last few years, M. Wolski also dedicated a huge part of his research to near sets theory, its applications and distinction between near sets and rough sets [29] [30] [31].

Some of the probe functions representing descriptive features in digital images used in this research include:

- **Colour**: Colour as a probe function provides the colour content of the image and is one of the important and useful probe functions that can be used for pattern recognition and classification. In a colour image, $v = [R; G; B]$ is a feature vector containing colour components of a pixel in RGB colour space. In other words, R, G, B are the corresponding values of a particular pixel in three different planes which represents the colour feature. RGB is the most widely used colour space. It stands for red, green and blue and is defined by the three chromaticities of the red, green, and blue primaries, and can produce any chromaticity that is defined by those colours [32].

- **Gradient Direction**: The image gradients extract important information from the image and are widely used for pattern recognition and classification. They measure the changes in the image and provide the gradient magnitude and gradient direction of an image as a vector. The magnitude of the gradient (which is the length of the vector) shows how quickly the image is changing and the direction of the gradient (which is the direction of the vector) tells us the direction in which the image is changing [33]. Image gradient $\bigtriangledown f$ and gradient direction θ which is widely used in this research as one of the probe functions are calculated as follows [32]:

$$\triangledown f = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}. \tag{2.13}$$

$$\theta = \tan^{-1} \left[\frac{g_y}{g_x} \right]. \tag{2.14}$$

2.5 Metric Proximity

J. F. Peters in [7] defined proximity as a nearness relation between sets. In a metric space (X, d), a proximity relation between sets $A, B \subset X$ as described in [7] is given by

$$A \quad \text{is near} \quad B \iff D(A, B) = 0, \quad \text{where}$$

$$D(A, B) = \begin{cases} \inf\{d(a, b) : a \in A \text{ and } b \in B\}, & \text{if } A, B \neq \varnothing, \\ \infty, & \text{if } A \text{ or } B \text{ is empty.} \end{cases} \tag{2.15}$$

2.6 Spatial Nearness

Closure of a set is considered the basis for spatial nearness. The closure of a set A in a space X (denoted by clA) is the set of all points that are close to A [7]. In other words, clA contains all of the boundary points of A as well as all of the interior points of A.

According to Peters in [7] [8], let $\mathcal{P}(X)$ denote the collection of all subsets of X. Subsets $A, B \in \mathcal{P}(X)$ are spatially near, provided the intersection of closure of A (denoted by clA) and the closure of B (denoted by clB) is non-empty, which implies $A\delta B$. In other words, non-empty sets are spatially near, provided the sets have at least one point in common [34].

The concept of spatial nearness can be understood, considering the spatial closure of a subset $A \in \mathcal{P}(X)$ (denoted by clA), as follows

$$clA = \{x \in X : d(x, A) = 0\}, \quad \text{where}$$

$$d(x, A) = \inf \{d(x, a) : a \in A\}. \tag{2.16}$$

clA is the set of all points x in X that are close to A and $d(x, A)$ is the Hausdorff distance [35] between x and the set A. The Hausdorff Distance is a mathematical construct to measure the closeness of two sets of points that are subsets of a metric space. Two sets are close in the Hausdorff distance if every point of either set is close to some point of the other set [36]. The Hausdorff distance between sets X and Y is given by [37] [38]

$$d_H(X, Y) = \max\{\sup_{x\in X} \inf_{y\in Y} d(x, y)\, , \sup_{y\in Y} \inf_{x\in X} d(x, y)\}. \tag{2.17}$$

2.6.1 Closure of a set

According to Peters in [7], in a proximity space X, the closure of set A in X coincides with the intersection of all closed sets that contain set A. In 1964, J. M. Smirnov introduced the following theorem on closure of sets in his article on proximity spaces [39]:

Theorem 1. *The closure of any set A in the proximity space X is the set of points $x \in X$ that are close to A.*

Proof. Let $\bar{A}$ denote the set of all points $x \in X$ that are close to A. Then we show that $\bar{A} = clA$. First we show that $\bar{A} \subseteq clA$. Let $x \in \bar{A}$, i.e., x is close to A. Since $A \subseteq clA$, then A is close to clA. Since clA is closed so that $x \in clA$, then $\bar{A} \subseteq clA$. Next, we can show that $\bar{A} \supseteq clA$. $\square$

2.7 Descriptive Nearness

Descriptively near sets were introduced by J.F. Peters in 2007 for the first time [25] [40]. Descriptively near sets are sets that are disjoint and are spatially far from each other, but they resemble each other and can be used for solving classification and pattern recognition problems, as they are used in this research. In 2013, J. F. Peters and S. Naimpally presented the connection between spatially near sets and descriptively near sets in [41] [8].

According to Peters in [7] [42], let X be a non-empty set, x as a member of X, $\Phi = \{\phi_1, \dots, \phi_n\}$ a set of probe functions that represent features of each x. Let Φ_x denote a feature vector for the object x, i.e., a vector of feature values that describe x. To obtain a descriptive proximity relation, we need to choose a set of probe functions, which provides a basis for describing points in a set. Let $A, B \in \mathcal{P}(X)$. Let $\mathcal{Q}(A), \mathcal{Q}(B)$ denote sets of descriptions of points in A and B, respectively:

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

$$\mathcal{Q}(B) = \{\Phi(b) : b \in B\}. \tag{2.18}$$

The descriptive proximity relation of A and B (denoted by $A\,\delta_\Phi\,B$ and reads as A *is descriptively near* B) is defined by

$$A\,\delta_\Phi\,B \iff \mathcal{Q}(A) \cap \mathcal{Q}(B) \neq \varnothing. \tag{2.19}$$

The descriptive intersection of A and B is defined by

$$A \underset{\Phi}{\cap} B = \{x \in A \cup B : \Phi(x) \in \mathcal{Q}(A) \quad and \quad \Phi(x) \in \mathcal{Q}(B)\}. \tag{2.20}$$

That is, $x \in A \cup B$ is in $A \underset{\Phi}{\cap} B$, provided $a \in A$ and $b \in B$, such that $\Phi(x) = \Phi(a) = \Phi(b)$. Thus, A and B can be disjoint and yet $A \underset{\Phi}{\cap} B$ can be non-empty.

Now, the descriptive proximity relation δ_Φ is defined by

$$\delta_\Phi = \{(A, B) \in \mathcal{P}(X) \times \mathcal{P}(X) : \, clA \underset{\Phi}{\cap} clB \neq \varnothing\}. \tag{2.21}$$

Whenever sets A and B have no points with matching or almost near descriptions, the sets are descriptively far from each other.

2.8 Descriptive Similarity Distance

Descriptive similarity distance measures the descriptive distance between the pairs of collections of sets for finding similarities between disjoint sets and recognizing disjoint objects and set patterns that resemble each other. According to J. F. Peters in [11], let X be a descriptive proximity space and $\mathcal{A}, \mathcal{B} \in 2^X$ be collections containing sets A and B, respectively.

The descriptive distance D_Φ which was introduced by Čech [22] in 1966, is used to define the descriptive similarity distance $\mathbb{D}_\Phi$ between collections of sets. This type of distance calculates the greatest lower bound of the standard distances between pairs of set elements [23] [24].

The descriptive distance $\mathbb{D}_\Phi : \mathcal{P}^2(X) \times \mathcal{P}^2(X) \to \mathbb{R}$ that can be used to measure the distance between collections of non-empty sets $\mathcal{A}$ and $\mathcal{B}$ that are descriptively near each other, is defined by

$$\mathbb{D}_\Phi(\mathcal{A},\mathcal{B}) = \inf\{D_\Phi(A,B) : A \in \mathcal{A}, B \in \mathcal{B}\}, \text{ where}$$

$$D_\Phi(A,B) = \inf\{d(\Phi(a)),(\Phi(b)) : a \in A, b \in B\}. \tag{2.22}$$

If we consider $\mathbb{D}_\Phi$ as the descriptive similarity distance between collections of sets $\mathcal{A},\mathcal{B} \in 2^X$, $\varepsilon > 0$ and X as a non-empty set endowed with descriptive proximity δ_Φ, then the condition for sets $\mathcal{A}$ and $\mathcal{B}$ to be descriptively similar is

$$\mathcal{A}\,\delta_\Phi\,\mathcal{B} \iff \mathbb{D}_\Phi(\mathcal{A},\mathcal{B}) \leq \varepsilon. \tag{2.23}$$

2.9 Pattern Similarity Distance

Extending the idea of descriptive similarity distance measure discussed in 2.8 which calculates the similarity distance between sets, pattern similarity distance measure is introduced which calculates the similarity distance between patterns:

Let X and Y be metric descriptive proximity spaces endowed with metric d and descriptive proximity δ_Φ. Let $P1$ and $P2$ be set patterns in X and Y, respectively. The similarity distance between $P1$ and $P2$ (denoted by $\mathbb{D}_\Phi(P1,P2)$) is defined by

$$\mathbb{D}_\Phi(P1,P2) = \inf\{D(\mathcal{Q}(A),\mathcal{Q}(B)) :\ A \in P1\,,\ B \in P2\}. \tag{2.24}$$

Where $\mathcal{Q}(A)$ is the set of descriptions of picture points in the set A, i.e., $\mathcal{Q}(A) = \{\Phi(a) : a \in A\}$ and $D(\mathcal{Q}(A),\mathcal{Q}(B))$ is the descriptive Čech distance [22] between $\mathcal{Q}(A)$ and $\mathcal{Q}(B)$ defined as [7]

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

$$d(\Phi(a), \Phi(b)) = \sum_{i=1}^{|\Phi|} |\Phi_i(a) - \Phi_i(b)|. \tag{2.25}$$

In 1962, Franz L. Alta introduced the similarity distance between patterns for the first time [43]. His work and research in this area led to what is now known as pattern saliency.

2.9.1 Pattern Saliency

According to T. Kadir and M. Brady in [44], among a collection of patterns, a salient pattern is the one which is special and stands out. To find the salient pattern, the first step is to choose a pattern in a query image. Then, after finding the patterns in the test image, pattern similarity distance needs to be calculated between the pattern in query image and the pattern in test image. A test image pattern is salient, provided it is sufficiently close to the query image pattern [7].

J.F. Peters in [7] defined a saliency threshold as a number that separates salient picture patterns with similarity distances below the saliency threshold from non-salient picture patterns with similarity distances above the saliency threshold. According to Peters, let $P1$ and $P2$ be patterns in a query image X and test image Y and $\varepsilon > 0$ serve as a saliency threshold. Then $P2$ is considered salient, provided

$$\mathbb{D}_{\Phi}(P1, P2) < \varepsilon \tag{2.26}$$

Which means the set pattern[4] $P2$ is salient, provided the description of $P2$ is sufficiently close to the description of $P1$.

[4]A set pattern in a descriptive proximity space is a collection of sets descriptively near a motif set (or known as a pattern generator) [7]

2.10 Pattern-based Classification

Classification provides a basis for deciding which class a given pattern belongs to [45]. Pattern-based classification begins with identifying a motif (pattern generator) in a query image for each class. For the next step, a decision need to be made about the best fit between a query image motif and a test image motif. In the next step, a test image motif needs to be used in the generation of a test image pattern that represents that part of a test image that is close to a query image pattern [7].

Here is the basic approach for finding classes of similar images that was introduced by J.F. Peters [7]:

By finding those test images containing set patterns that are close to a set pattern in a query image, representing an image class, a class of mutually similar images can be constructed . By computing the similarity distance between a query image pattern and each test image pattern, closeness is determined. A test image belongs to the query image class, provided the similarity distance between query and test image patterns is less than some threshold.

2.11 K-means Clustering

In the simplest words, clustering is grouping similar objects together. Grouping the articles that are on similar subjects together. What is done by search engines or distinguishing between different stages of cancer tumour, based on provided information, are some examples that are all relied on clustering. K-means is one of the simplest unsupervised learning algorithms [46]. This algorithm can be used for creating algorithms that are computationally tractable and work with large data sets. K-means assumes Euclidean distance as the metric and it starts by randomly picking k as the number of clusters and then initializing k clusters by selecting one point per cluster. When the number of clusters is fixed to k, this algorithm gives a definition as an optimization problem: find the cluster centres and assign the objects to the nearest cluster center, such that the distances from the cluster are minimized.

After having the clusters populated with the k randomly picked points, each point in the data set is assigned in the cluster having a centroid nearest to that point. After all points are assigned to one of the clusters, the locations of the centroids of the k clusters are updated and all the points are re-assigned to their closest centroids. The algorithms iterates until it converges. Convergence happens when points do not move between clusters any more and the centroids stabilize. Since the centroids are updated and the points are re-assigned to clusters in every iteration, k centroids change until the convergence is reached. Best results are obtained with k-means when the data points are well separated and it is unable to handle noisy data and outliers. Its simplicity and its local-minimum convergence properties are often considered as its advantages.

This algorithm minimizes a squared error function, defined by

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \|x_i^{(j)} - c_j\|^2 \tag{2.27}$$

Where $\|x_i^{(j)} - c_j\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j and indicates the distance of the n data points from their respective cluster centres [47].

In [48], Hartigan and Wang introduced an efficient version of the classic K-means algorithm which was first proposed in 1975. The efficient version of the algorithm minimizes the sum of squares within the clusters, when trying to divide M points into K clusters and compared to the classic K-means, it is locally optimal and performs faster, especially with the larger number of clusters.

In 2002, Basu and Banerjee in [49] proposed some modifications to K-means in order to improve its performance by adding some labelled data to the data set to generate initial seed clusters and to bias the clustering process. This semi-supervised method is less sensitive to noise in the data.

Pelleg and Moore in [50] discussed the disadvantages of K-means: K as the number of clusters must be supplied by the user as a parameter and searching the space of the cluster locations that is performed in K-means is susceptible to local minima. The X-means algorithm introduced in [50] produces better clustering with respect to Bayesian Information Criterion (BIC) and runs much faster. X-means quickly estimates K and makes local decisions about which subset of the current centroids should split themselves in order to better fit the data, after each run of K-means. Ray and Turi in [51] also proposed a method which overcomes the limitations of K-means by allowing K as the number of clusters to be determined automatically by some measurements based on intra-cluster and inter-cluster distance measures, introduced in their work.

2.12 Data Set

For testing the validity of the methodology of this work, the pictures of children's left hand were used. I collected my data set at *University of Manitoba Campus Day Care Centre* by taking 146 pictures of the left hands of children between 3.5 to 5.5 years old in a course of 11 months.

Chapter 3

Methodology

The focus of this chapter is describing the methodology used in this research for classifying the images, from finding the spatial and descriptive neighbourhoods, selecting motif set, generating set patterns and calculating the similarity distances between set patterns to classifying images based on the calculated similarity distances. Figure 3.1 shows the flow chart of the methodology used in this work.

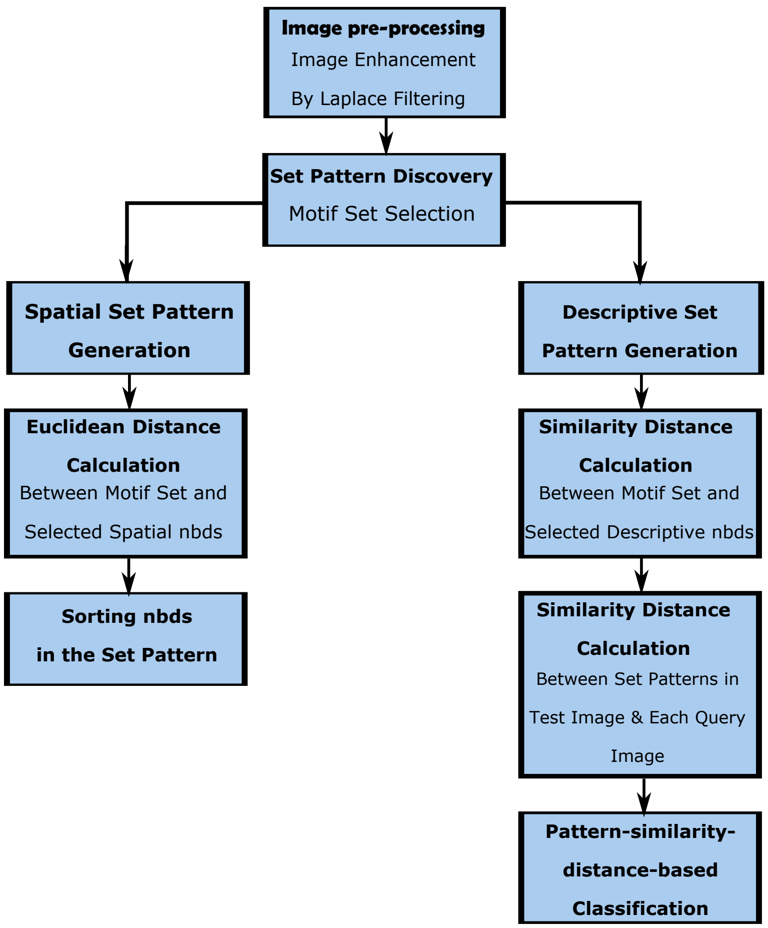


Figure 3.1: Methodology Flow Chart

3.1 Phase I - Image Pre-processing

Image pre-processing prepares the image for further analysis and it is useful for the study of near sets, visual patterns and different forms of topologies in digital images [7]. Image pre-processing can be used to enhance the appearance of the image, sharpen the image features, correct the image and detect the edges and corners in the image. Noise removal and colour correction are some of the candidates for pre-processing correction. Illumination enhancement and focus enhancements are some of the examples of pre-processing enhancement. Image filtering which is based on weighted sums of local neighbourhood pixel values [4] can also be used as part of the pre-processing process to improve the image, smooth the image or amplify the small differences in the image, depending on the application [52] [7].

3.1.1 Image Enhancement by Laplace Filtering

The purpose of image filtering is manipulating the image for smoothing, sharpening, restoration, compression, denoising, edge detection and shape morphology. According to [4], some filters such as Roberts, Prewitt and Sobel use the first order derivatives for image enhancement, but the second order derivative filters have stronger response to fine details and simpler implementation. In this research, Laplacian filter which is based on second spatial derivatives was used as a sharpening filter to highlight fine details and accentuate edges and also to remove blurring from images as a step toward image enhancement.

In the image g, the $2D$ Laplacian filter for a pixel $g(x, y)$ is given by

$$\nabla^2 g(x,y) = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}. \tag{3.1}$$

Where the partial second order derivatives in the x direction is defined as follows

$$\frac{\partial^2 g}{\partial x^2} = g(x+1,y) + g(x-1,y) - 2g(x,y). \tag{3.2}$$

And in y direction is calculated as follows

$$\frac{\partial^2 g}{\partial y^2} = g(x,y+1) + g(x,y-1) - 2g(x,y). \tag{3.3}$$

So, the discrete form of the Laplacian filter is given by

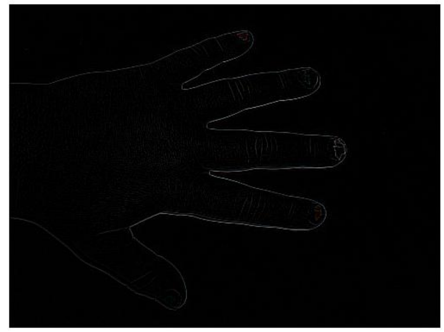
$$\nabla^2 g(x,y) = g(x+1,y) + g(x-1,y) + g(x,y+1) + g(x,y-1) - 4g(x,y). \tag{3.4}$$

Applying the Laplacian filter to an image, results in a new image with highlighted edges and other discontinuities, but it is not an enhanced image yet. To generate the final sharpened enhanced image h, the filtered image should be subtracted from the original image, in terms of a pixel value $g(x,y)$ [4],

$$h(x,y) = g(x,y) - \nabla^2 g(x,y). \tag{3.5}$$

The result of applying the Laplacian filter to the image *jasmine.jpg* from my data set that is used in this research is shown in figure 3.2. In the final sharpened image edges and fine detail are much more obvious.

(a) Original Image



(b) Laplacian Filtered Image

(c) Enhanced Image

Figure 3.2: Image Enhancement by Laplace Filtering

3.2 Phase II - Set Pattern Discovery

The word ”pattern” often means something capable of being repeated in a predictable manner into similar copies. Repetition of some form in parts of a digital image is called a visual pattern. A set pattern in an image is defined as a collection of sets (denoted by $\mathfrak{P}(M)$) that all of the members of the collection have common properties and are all near a given set called as a pattern generator or motif set M [53] [54] [7]. The nearness relation in set patterns definition can be specified as **spatial** or **descriptive** nearness [55]

- **Spatial Set Pattern** (denoted by $\mathfrak{P}(M)$) contains sets that are spatially near each other and a given set known as the pattern generator or the motif set M .

- **Descriptive Set Pattern** (denoted by $\mathfrak{P}_\Phi(M)$) contains sets that are descriptively near each other and a given set known as the pattern generator or the motif set M.

Peters in [7] defined a motif as a set $M \in \mathcal{P}$ with members which are near one or more members of other sets.

3.2.1 Spatial Motif Set Patterns

According to J. F. Peters in [7], let X be a set of picture points in a digital image and $\mathcal{P}^2(X)$ be the set of collections of subsets in X. Motif set M needs to be selected in X, in such a way that $M\delta A$ for each subset $A \in \mathfrak{P}(M)$. $\mathfrak{P}(M)$ is the collection of sets that are spatially near motif set M. If the following axioms are satisfied, then $\mathfrak{P}(M)$ is considered as a spatial motif set pattern [7]:

Axiom 1. *For each pair of sets $A, B \in \mathfrak{P}(M)$, A and B are disjoint.*

Axiom 2. *For each $A \in \mathfrak{P}(M)$, A is near motif set M.*

Axiom 3. *If $A, B \in \mathfrak{P}(M)$ are copies of the motif set M, then there is a descriptive isometry*[1] *$f : \mathbb{R}^n \to \mathbb{R}^n$, so that:*

$$d(f(\Phi(x)), f(\Phi(y))) = d(\Phi(x), \Phi(y)) \quad \textit{where,}$$

$$\Phi(x), \Phi(y) \in \mathcal{Q}(A) \quad \textit{and} \quad f(\Phi(x)), f(\Phi(y)) \in \mathcal{Q}(B).$$

3.2.1.1 Spatial Pattern Generation

The first step for generating the spatial proximity pattern in image X is choosing a region-of-interest Y in image X. The next step is selecting the motif set M in Y as a spatial neighbourhood of a point $x \in Y$. The last step in this process is determining all sets

[1] *isometry* map preserves the distance.

$A \in \mathcal{P}(X)$ that are spatially near the motif M. The resulting spatial pattern $\mathfrak{P}(M)$, based on what was discussed in section 2.6, is defined by

$$\mathfrak{P}(M) = \{A \in \mathcal{P}(X) : \ clA \cap clM \neq \varnothing\}. \tag{3.6}$$

3.2.2 Descriptive Motif Set Patterns

According to J. F. Peters in [7], let X be a set of picture points in a digital image and $\mathcal{P}^2(X)$ be the set of collections of subsets in X. Motif set M needs to be selected in X, in such a way that $M \in \mathcal{P}(X)$. Considering Φ as the set of probe functions that represent features of members of X, $\mathfrak{P}_\Phi(M)$ is the collection of all sets in X that are descriptively near motif set M. Then $\mathfrak{P}_\Phi(M) \in \mathcal{P}^2(X)$ is considered as a descriptive motif set pattern if the following axioms are satisfied [7]:

Axiom 4. *For each pair of sets $A, B \in \mathfrak{P}_\Phi(M)$, A and B are spatially and descriptively far from each other, provided $\mathcal{Q}(A) \cap \mathcal{Q}(B) = \varnothing$, which means for any pair of points $a \in A$ and $b \in B$, $\Phi(a) \neq \Phi(b)$. Otherwise, A and B can be spatially disjoint but descriptively near each other or they can be both spatially and descriptively near each other.*

Axiom 5. *For each $A \in \mathfrak{P}_\Phi(M)$, A is descriptively near motif set M.*

Axiom 6. *If the pairs $A, B \in \mathfrak{P}_\Phi(M)$ are descriptively near the motif set M, then a descriptive isometry maps the sets of descriptions of points in A into the set of descriptions of points in B.*

Figure 3.3 shows a simple visual example of a descriptive motif set pattern. X and Y represent two images. Φ as a set of probe functions, represents pixel colours. $M \in Y$ is a motif set and sets A, B and C in X have some pixels with descriptions that match some pixels in M, so $A\delta_\Phi M$, $B\delta_\Phi M$ and $C\delta_\Phi M$. A, B and C are spatially disjoint but descriptively near each other. In this example, the descriptive motif set pattern is defined as $\mathfrak{P}_\Phi(M) = \{A, B, C\}$.

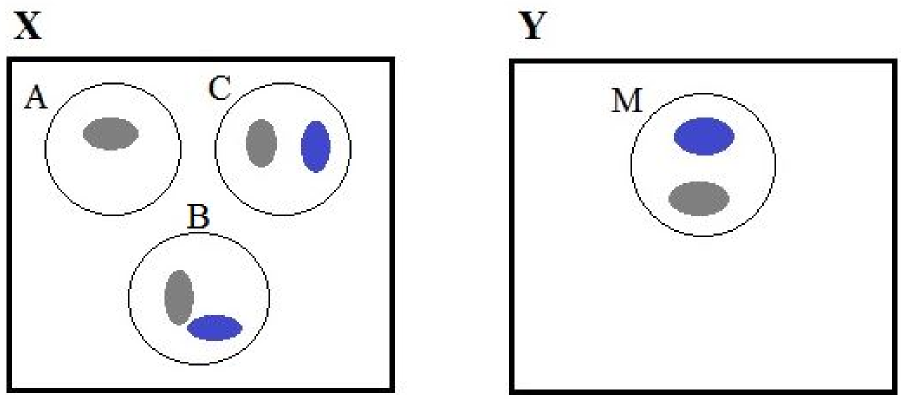


Figure 3.3: An example of a descriptive motif set pattern,
$\mathfrak{P}_{\Phi}(M) = \{A, B, C\}$

3.2.2.1 Descriptive Pattern Generation

The first step for generating the descriptive proximity pattern in image X is choosing a region-of-interest Y in image X. The next step is selecting the motif set M in Y as a descriptive neighbourhood of a distinguished point of interest $x \in Y$

$$M \doteqdot N_{\Phi(x)} = \{y \in Y : \Phi(y) = \Phi(x)\}. \tag{3.7}$$

The last step in this process is determining all sets $A \in \mathcal{P}(X)$ that are descriptively near the motif set M. The resulting descriptive pattern $\mathfrak{P}_{\Phi}(M)$, based on what was discussed in section 2.7, is defined by

$$\mathfrak{P}_{\Phi}(M) = \{A \in \mathcal{P}(X) : \ clA \underset{\Phi}{\cap} clM \neq \varnothing\}. \tag{3.8}$$

3.2.3 Set Pattern Generation Process

In this work, spatial and descriptive pattern generation is implemented.

- **In the spatial pattern generation** process, after generating the spatial set pattern $\mathfrak{P}(M)$ based on what was discussed in 3.2.1, the Euclidean distance[2] was calculated

[2]Euclidean distance was discussed in 2.2.

between the motif set M and all the other selected spatial neighbourhoods in the set pattern. Knowing the Euclidean distance distance between motif set M and each of the selected spatial neighbourhoods, then all of these neighbourhoods were sorted based on their spatial closeness to the motif set M.

- **In the descriptive pattern generation** process, after generating the descriptive set pattern $\mathfrak{P}_{\Phi}(M)$ based on what was discussed in 3.2.2, the descriptive similarity distance[3] was calculated between the motif set M and all the other selected bounded descriptive neighbourhoods in the set pattern. Knowing the descriptive similarity distance between motif set M and each of the selected descriptive neighbourhoods, then all of these neighbourhoods were sorted based on their closeness to the motif set M. This sorting process is based on the descriptively closeness of the selected neighbourhoods to the motif set M which improves the classification results. The 6 descriptively closest neighbourhoods to the motif set M stay in the set pattern to amplify the specific description of the set pattern, while the others can be removed as they are less effective in forming a strong set pattern. A threshold can be defined for clarifying the closeness, thus, the neighbourhoods which their similarity distance to the motif set M is less than the defined threshold would stay in the set pattern and the ones which their similarity distance to the motif set M is more than the defined threshold would be removed.

3.3 Phase III - Pattern-similarity-distance-based Classification

Specifying a criterion for deciding the membership of an image in a class is a necessary step for classifying digital images. In this research, pattern similarity is the basis selected for classifying the digital images. In this phase, pattern-similarity-distance-based classification

[3]Descriptive similarity distance was discussed in 2.8.

is conducted by selecting a motif set in each target image (or query image) as the generator for the set pattern which results in generating the set patterns. The generated motif set pattern is the representative of the class of images. Finding the set pattern is an important step towards classifying the test image. In fact, extraction of salient patterns[4], which are the representative of the characteristics of the selected region of the image [56], is one of the main parts of this work.

In this research, for each time that the program runs, there are two target hand images X and Y, known as *Target.Image1* and *Target.Image2* and one test hand image T, known as *Test.Image* involved in the program. *Target.Image1* and *Target.Image2* are representatives of two classes. The goal is to find out whether the test image T belongs to *Taget.Image1* class or *Target.Image2* class, based on the pattern-similarity-distance.

The steps that were followed for the pattern-similarity-distance-based classification in this research are :

1. *Target.Image1* and *Target.Image2* were selected as the target images (or can be called query images). These two images are representative of two different classes. *Test.Image* was selected as the test image which it needs to be found out that which class it belongs to.

2. In this step, a region-of-interest was selected in both target images and the test image, namely R_1, R_2 and R_3, respectively. Selecting the region-of-interest depends on the feature that is being used for classification. for example, if a shape-based classification is being conducted, then the selected region-of-interest should contain edge structures.

3. In this step, the feature that classification process is based on was selected. For this study, Φ contains two probe functions ϕ, one for extracting the colour of each picture point which was used for colour-based classification and one for extracting the gradient orientation of each picture point which was used for shape-based classification.

[4]Saliency was discussed in 2.9.1

4. The radius and epsilon value for each of the selected bounded descriptive neighbourhoods were selected in this step. Choosing the ε value is very important which can affect the classification result and its sensitivity, i.e. choosing a very high value for ε would reduce the accuracy of classification and selecting an unreasonably low value for ε reduces the classification performance.

5. A bounded descriptive neighbourhood of point a in *Target.Image1*, namely $N_{\Phi(a)}$ was constructed, which serves as the motif in *Target.Image1*.

6. $N_{\Phi(a)}$ was used as the set pattern generator to generate a set pattern $\mathfrak{P}(N_{\Phi(a)})$, which is the collection of sets in the selected region-of-interest R_1 in *Target.Image1* that are descriptively close to $N_{\Phi(a)}$

$$\mathfrak{P}(N_{\Phi(a)}) = \{A \in R_1 \ : \ A \,\delta_\Phi\, N_{\Phi(a)}\}.$$

7. A bounded descriptive neighbourhood of point b in *Target.Image2*, namely $N_{\Phi(b)}$ was constructed, which serves as the motif in *Target.Image2*.

8. $N_{\Phi(b)}$ was used as the set pattern generator to generate a set pattern $\mathfrak{P}(N_{\Phi(b)})$, which is the collection of sets in the selected region-of-interest R_2 in *Target.Image2* that are descriptively close to $N_{\Phi(b)}$

$$\mathfrak{P}(N_{\Phi(b)}) = \{B \in R_2 \ : \ B \,\delta_\Phi\, N_{\Phi(b)}\}.$$

the number of the bounded descriptive neighbourhoods in the generated set patterns in *Target.Image1* and *Target.Image2* should be equal.

9. A bounded descriptive neighbourhood of point c in *Test.Image*, namely $N_{\Phi(c)}$ was constructed, which serves as the motif in *Test.Image*.

10. $N_{\Phi(c)}$ was used as the set pattern generator to generate a set pattern $\mathfrak{P}(N_{\Phi(c)})$, which is the collection of sets in the selected region-of-interest R_3 in *Test.Image* that are descriptively close to $N_{\Phi(c)}$

$$\mathfrak{P}(N_{\Phi(c)}) = \{C \in R_3 \, : \, C \, \delta_\Phi \, N_{\Phi(c)}\}.$$

the number of the bounded descriptive neighbourhoods in the generated set patterns in *Test.Image*, *Target.Image2* and *Test.Image* should be equal.

11. Pattern similarity distance measure(2.9) was calculated between $\mathfrak{P}(N_{\Phi(c)})$ and $\mathfrak{P}(N_{\Phi(a)})$ set patterns and also $\mathfrak{P}(N_{\Phi(c)})$ and $\mathfrak{P}(N_{\Phi(b)})$ set patterns, which are defined by

$$\mathbb{D}_\Phi(\mathfrak{P}(N_{\Phi(a)}) \, , \, \mathfrak{P}(N_{\Phi(c)})) = \inf\{D_\Phi(A,C) \, : \, A \in \mathfrak{P}(N_{\Phi(a)}) \, , \, C \in \mathfrak{P}(N_{\Phi(c)})\}. \tag{3.9}$$

$$\mathbb{D}_\Phi(\mathfrak{P}(N_{\Phi(b)}) \, , \, \mathfrak{P}(N_{\Phi(c)})) = \inf\{D_\Phi(B,C) \, : \, B \in \mathfrak{P}(N_{\Phi(b)}) \, , \, C \in \mathfrak{P}(N_{\Phi(c)})\}. \tag{3.10}$$

12. If $\mathbb{D}_\Phi(\mathfrak{P}(N_{\Phi(a)}), \mathfrak{P}(N_{\Phi(c)})) \; < \; \mathbb{D}_\Phi(\mathfrak{P}(N_{\Phi(b)}), \mathfrak{P}(N_{\Phi(c)}))$, it means that the set pattern in *Test.Image* is descriptively closer to the set pattern in *Target.Image1*, so that the *Test.Image* classifies as a member of the *Target.Image1* class, otherwise if $\mathbb{D}_\Phi(\mathfrak{P}(N_{\Phi(a)}), \mathfrak{P}(N_{\Phi(c)})) \; > \; \mathbb{D}_\Phi(\mathfrak{P}(N_{\Phi(b)}), \mathfrak{P}(N_{\Phi(c)}))$, it means that the set pattern in *Test.Image* is descriptively closer to the set pattern in *Target.Image2*, therefore the *Test.Image* classifies as a member of the *Target.Image2* class.

Chapter 4

Experimental Results and Analysis

This chapter presents the step-by-step experiments that were carried out in this research and the obtained results.

4.1 Input Data Set

In this research work, the pictures of children's left hand were used as the input data set for testing the validity of the proposed classification method. The data set was collected at *University of Manitoba Campus Day Care Centre* by taking 146 pictures of the left hands of children between 3.5 to 5.5 years old in a course of 11 months. these pictures are different in colour and shape, i.e. some are lighter in colour and some are darker, some have longer fingers and some shorter, some have bigger gaps between the fingers which all make them classifiable based on colour and shape.

For testing the validity of the proposed method, these images have been visually classified based on two different features: colour and gradient orientation. Experiments were carried out on two classes of digital images, namely *Class1* and *Class2*. Figure 4.1 and 4.2 show some of the digital images in each of these two visually classified classes used for the experiments, when gradient orientation was the selected feature.

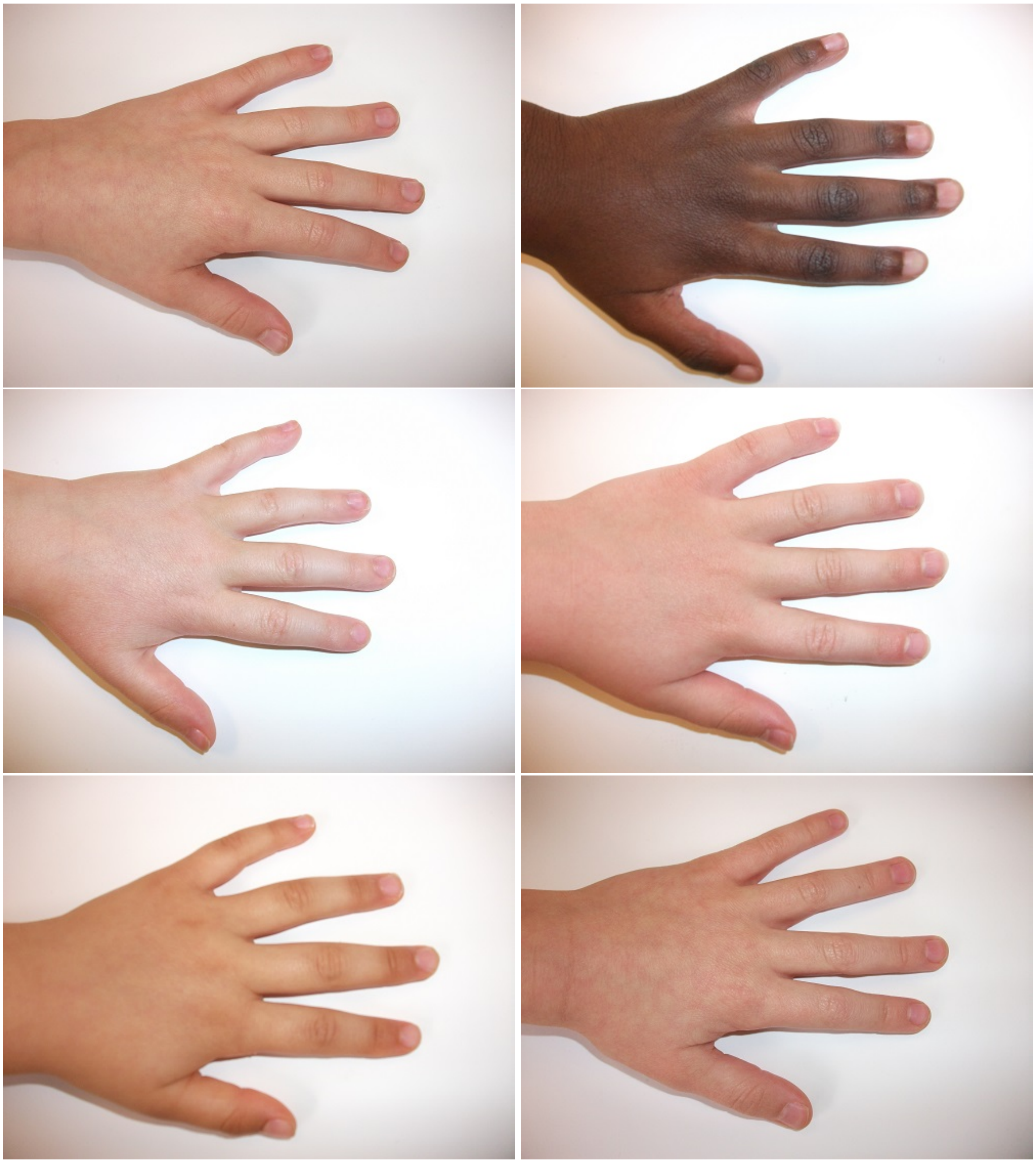


Figure 4.1: Sample images from hand images in class 1. Hands in this class are narrower with longer fingers.

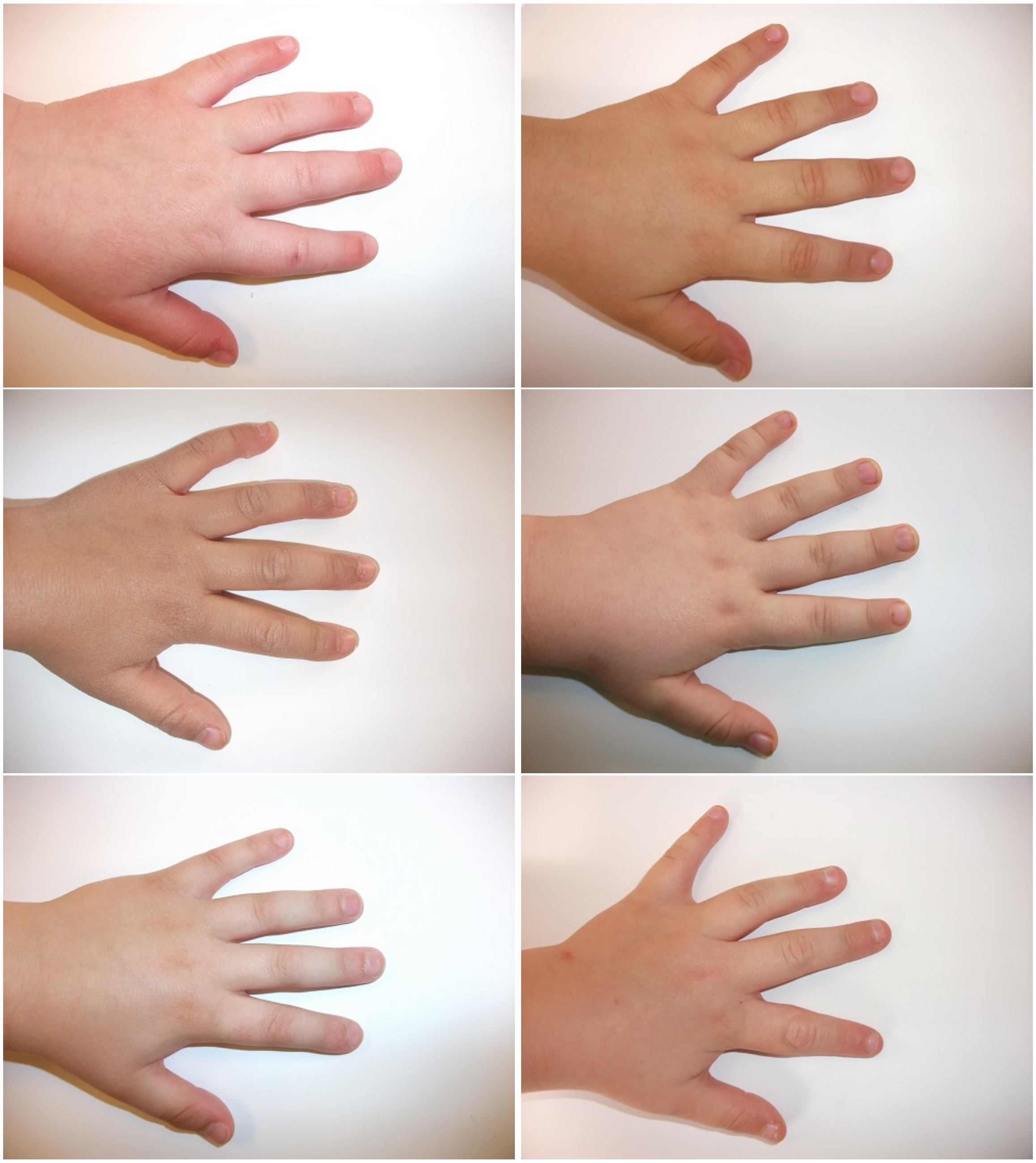


Figure 4.2: Sample images from hand images in class 2. Hands in this class are wider with shorter fingers.

4.2 Experimental Setup and Testing

An application named ***Start_Up*** was developed as the first step for this research work. This application, as well as all the other applications that were developed in this work are

MATLAB Graphical User Interface (GUI) which were developed using MATLAB R2013a. These applications were tested on an Inter(R) Core(TM) i5-2520M CPU @ 2.50 GHZ, 16GB RAM laptop computer.

The ***Start_Up*** application offers the option of choosing between *Descriptive*[1] and *Spatial* neighbourhood types for studying the features of each of these neighbourhoods. It also offers the *Classification* option which is the main focus in this work. Figure 4.3 shows a snapshot of this developed GUI in MATLAB.

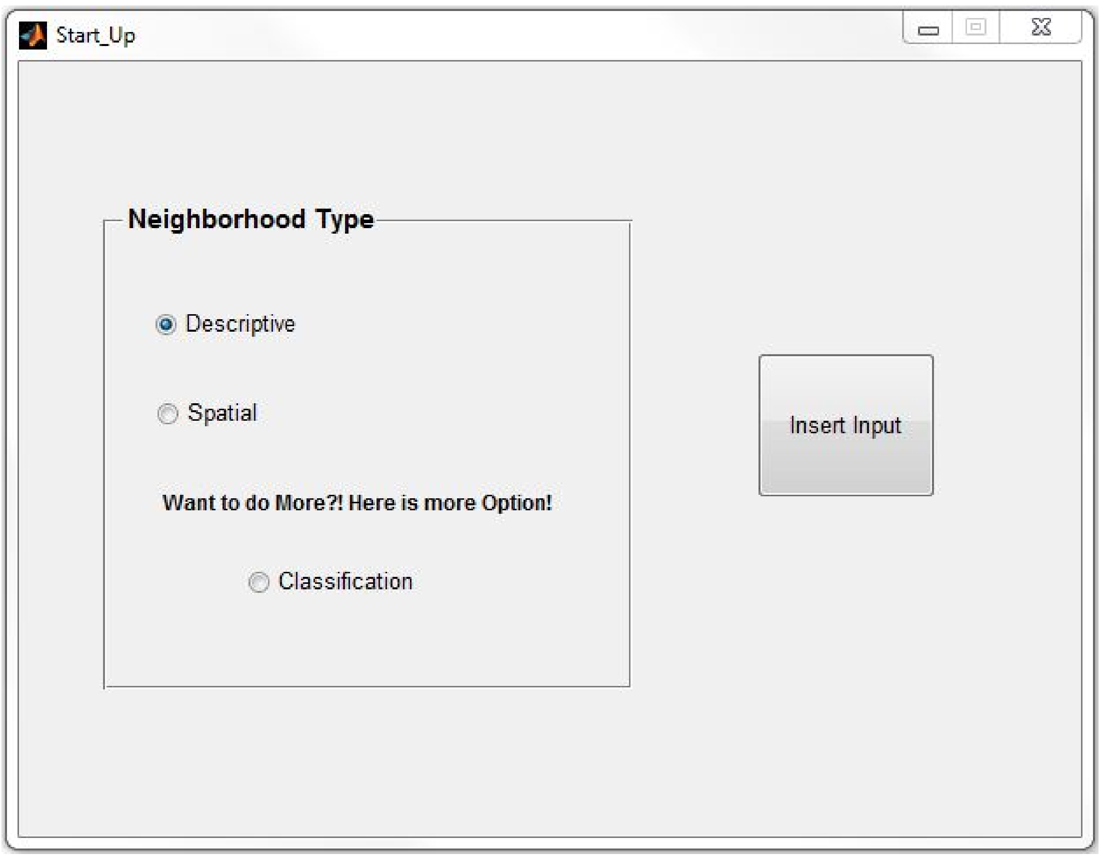


Figure 4.3: Start_Up Graphical User Interface (GUI).

4.2.1 Spatial Neighbourhood

Figure 4.4 shows a snapshot of the ***Spatial*** GUI that opens up after selecting it on ***Start_Up*** GUI. Spatial neighbourhood was previously discussed in 2.3.1. As seen in this figure, this

[1]Bounded descriptive neighbourhood that was discussed in 2.3.2.2

GUI lets the user load an RGB or gray scale image through the ***File*** drop down menu. As shown in figure 4.4, *"Class1-12.jpg"* was loaded as the input image for the presented example. The ***Input Data*** panel lets the user input the radius value for the spatial neighbourhood in the ***Radius*** edit box. After selecting the neighbourhoods on the image and all the processes, clicking on the ***Save*** push-button saves the image and the ***Reset*** push-button resets the GUI and the MATLAB workspace valuables. Figure 4.5 shows the centre pixel value of each of the selected neighbourhoods.

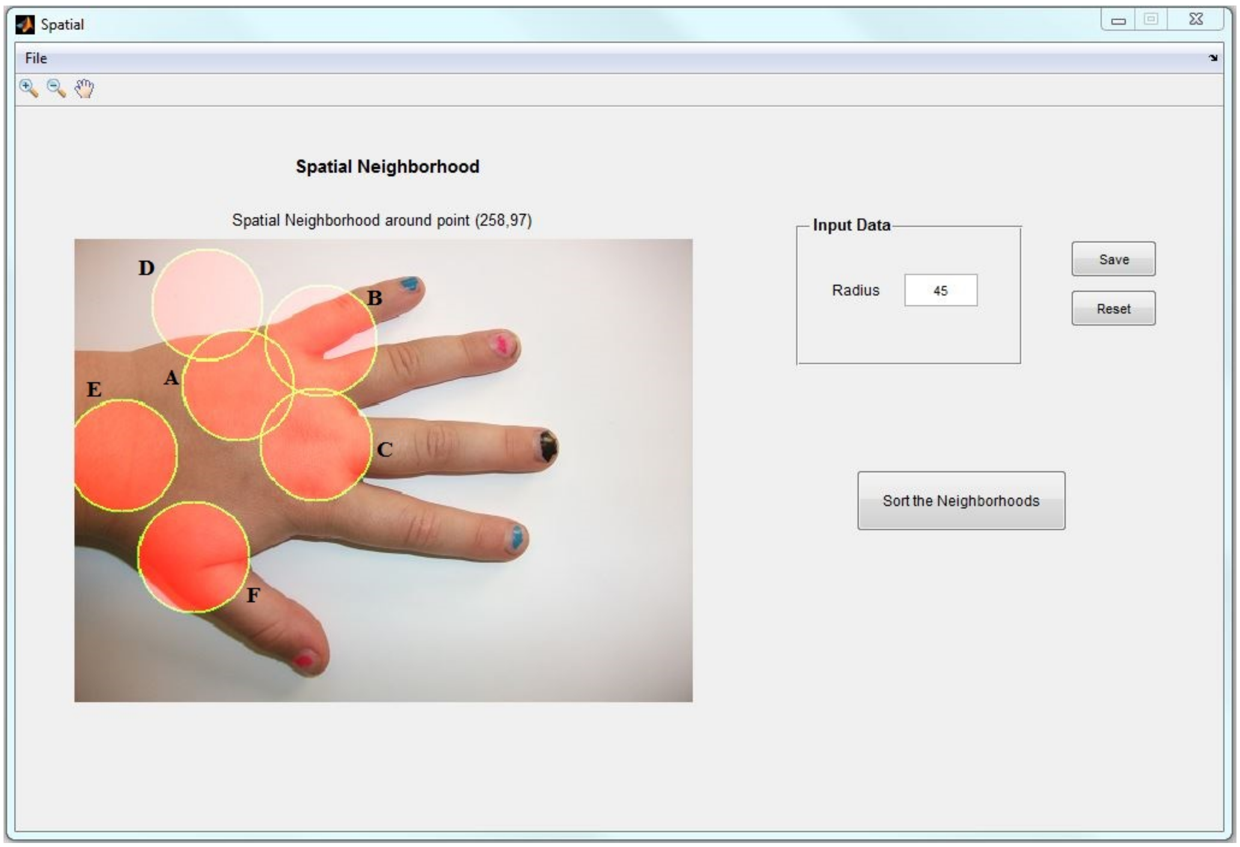


Figure 4.4: Spatial GUI - Radius: 45, Number of Neighbourhoods: 6.

```
>> Spatial
X(xpos,ypos):(119,133)
X(xpos,ypos):(83,200)
X(xpos,ypos):(167,196)
X(xpos,ypos):(54,108)
X(xpos,ypos):(176,39)
X(xpos,ypos):(258,97)
```

Figure 4.5: Centre Pixel Values for Neighbourhoods A, B, C, D, E, F in Order.

By clicking on the ***Sort the Neighbourhoods*** push-button, the selected neighbourhoods B, C, D, E and F are being sorted based on their closeness to neighbourhood A as the reference neighbourhood[2]. The result for the sorting process in this example is as follows:

Neighbourhood A, neighbourhood D, Neighbourhood B, Neighbourhood C.

Steps for spatially sorting neighbourhoods in terms of their closeness to a reference neighbourhood A are as follows,

0- Choose the neighbourhood radius.

1- Start with selecting a neighbourhood A on the image with radius $R1$.

2- Insert new neighbourhoods on the image. Let B be a new neighbourhood with radius $R2$ and let C be a new neighbourhood with radius $R3$. Consider only those neighbourhoods that overlap with neighbourhood A.

3- Compute

Dist (AB) = distance between centre of A and centre of B.

Dist (AC) = distance between centre of A and centre of C.

[2]Reference neighbourhood is the first selected neighbourhood and the Euclidean distance is calculated between this neighbourhood and each of the other selected neighbourhoods.

If Dist $(AB) < $ Dist (AC), then neighbourhood B is closest to neighbourhood A, else neighbourhood C is closest to neighbourhood A.

For observing and analysing the sorting process in this GUI, a structure named *store_ centres* is defined in this GUI which has different fields and it is accessible through the MATLAB workspace. It stores the centre pixel value for each of the selected spatial neighbourhoods, the Euclidean distance [3] between the centre pixel of the reference neighbourhood and each of the other selected neighbourhoods and also a field named *overlap* that specifies the neighbourhoods that overlap with the reference neighbourhood. This last field provides the required information for sorting the neighbourhoods. Figure 4.6 shows a snapshot of this structure in the MATLAB workspace for the ***Spatial*** GUI.

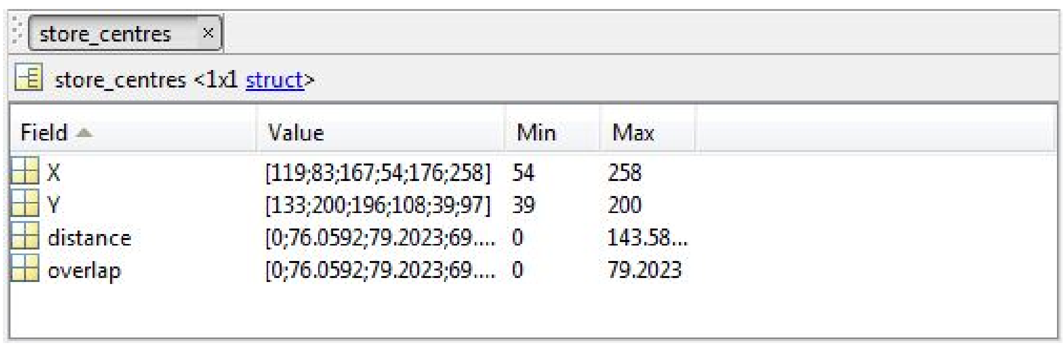


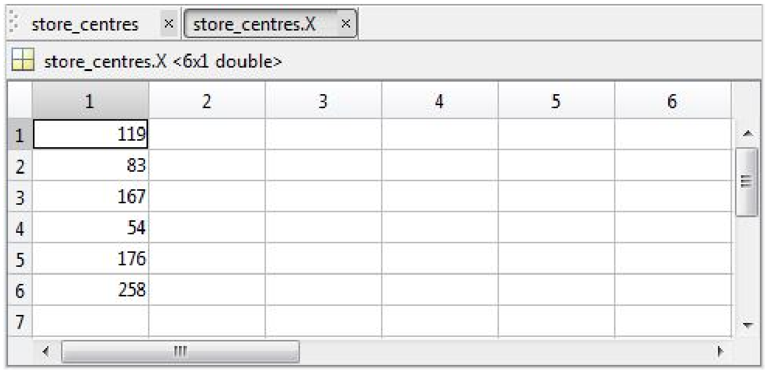
Figure 4.6: *"store_ centres"* Structure for *Spatial* GUI.

The *store_ centres* structure, as shown in the above figure, has four fields,

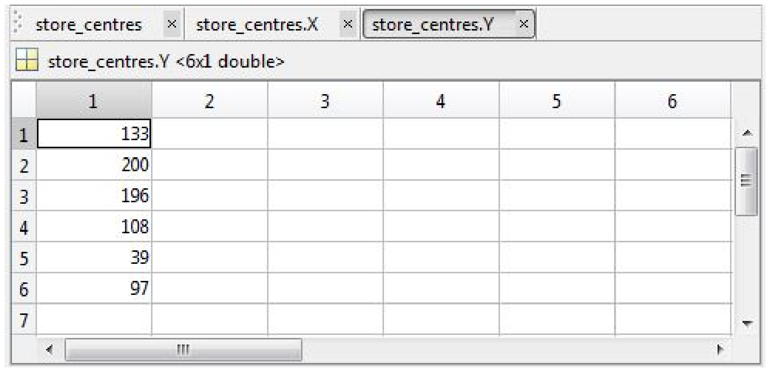
- Field X contains the x value of each of the centres of the selected neighbourhoods.
- Field Y contains the y value of each of the centres of the selected neighbourhoods.
- Field *distance* contains the Euclidean distance between neighbourhood A as the reference neighbourhood and each of the neighbourhoods B, C, D, E and F.
- Field *overlap* shows the neighbourhoods that overlap with neighbourhood A. Those are the neighbourhoods that their distance from A is less than $2 \times radius$. This field contains the information that is used for sorting the neighbourhoods.

[3] Euclidean distance was discussed in 2.2

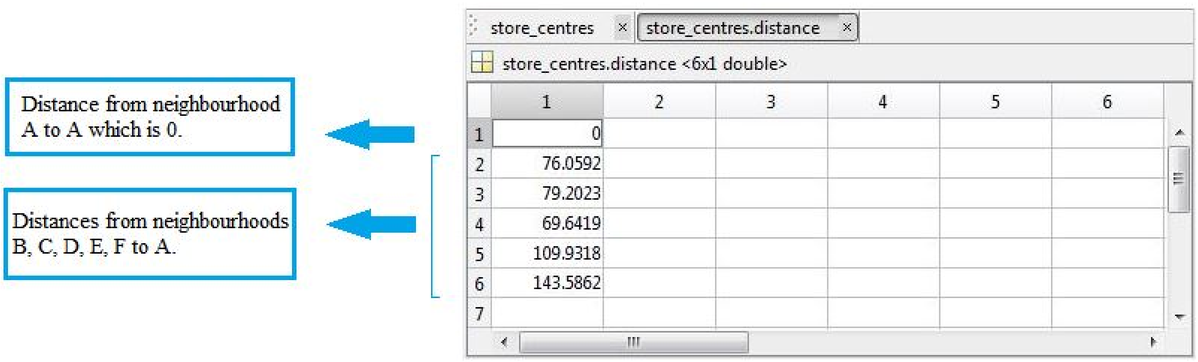
Figure 4.7 shows the different fields of the *store_ centres* structure which clarifies the neighbourhood sorting process.



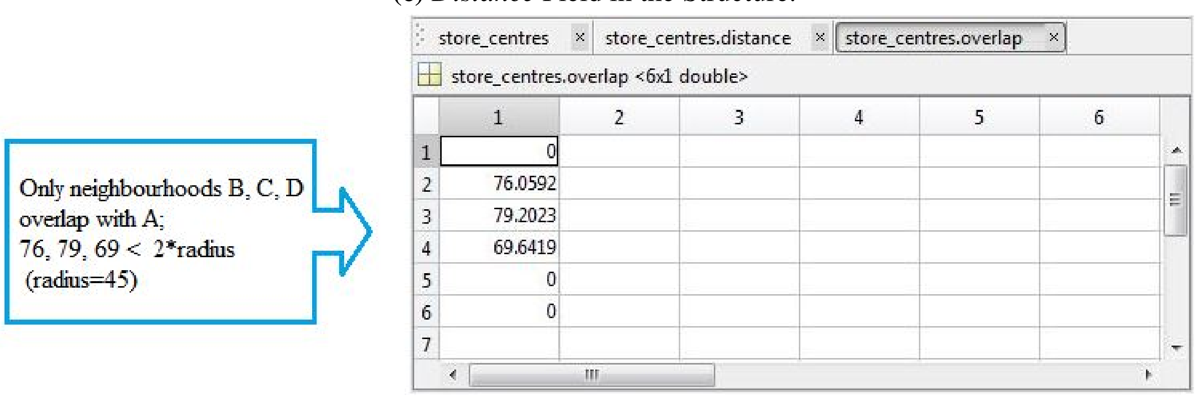
(a) X Values of the Centre Pixel of Each Neighbourhood.



(b) Y Values of the Centre Pixel of Each Neighbourhood.



(c) *Distance* Field in the Structure.



(d) *Overlap* Field in the Structure.

Figure 4.7: *"store_ centres"* Different Fields.

As mentioned earlier, clicking on the ***Sort the Neighbourhoods*** push-button activates the neighbourhood sorting process which results in the following structure that has the sorted neighbourhoods in it. In this structure, neighbourhoods are represented by their central pixel value.

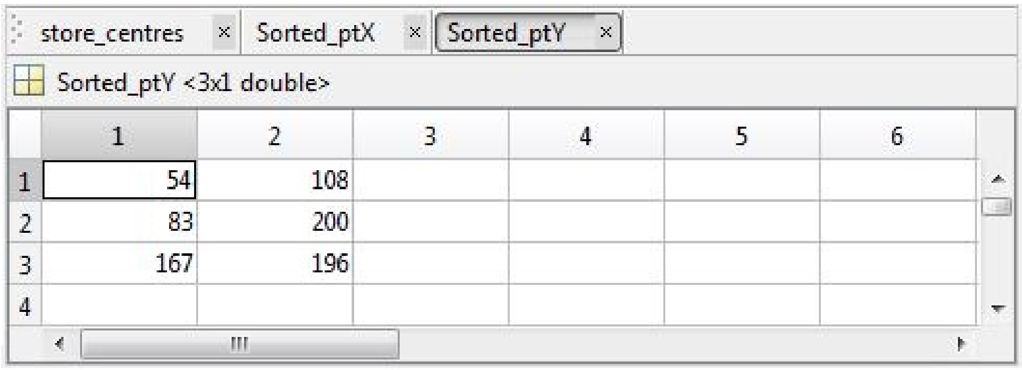


Figure 4.8: *Sorted Neighbourhoods - Neighbourhoods are Represented by their Central Pixel Value.*

4.2.2 Descriptive Neighbourhood

Figure 4.9 shows a snapshot of the ***Descriptive*** GUI that opens up after selecting it on ***Start_Up*** GUI. Bounded descriptive neighbourhood which was discussed in 2.3.2.2 was implemented in this GUI and it is the type of neighbourhood that *"Descriptive"* refers to, in this work.

As seen in this figure, the GUI lets the user load an RGB or gray scale image through the ***File*** drop down menu. As shown in figure 4.9, *"Class1-12.jpg"* was loaded as the input image for the presented example. Then Five different features namely colour, gradient magnitude, gradient orientation, vertical gradient (which finds vertical edges) and horizontal gradient (which finds horizontal edges) are offered to choose from through a pop-up menu. The ***Input Data*** panel lets the user input the radius value for the bounded descriptive neighbourhood in ***Radius*** edit box and the epsilon value in ***Epsilon*** edit box. By selecting each of the five features, the GUI shows the range of the required acceptable epsilon value for that feature. Clicking on the ***Save*** push-button saves the image after selecting the neighbourhoods on it and all the processes and the ***Reset*** push-button resets the GUI and the MATLAB workspace valuables. Figure 4.10 shows the centre pixel value of each of the selected neighbourhoods.

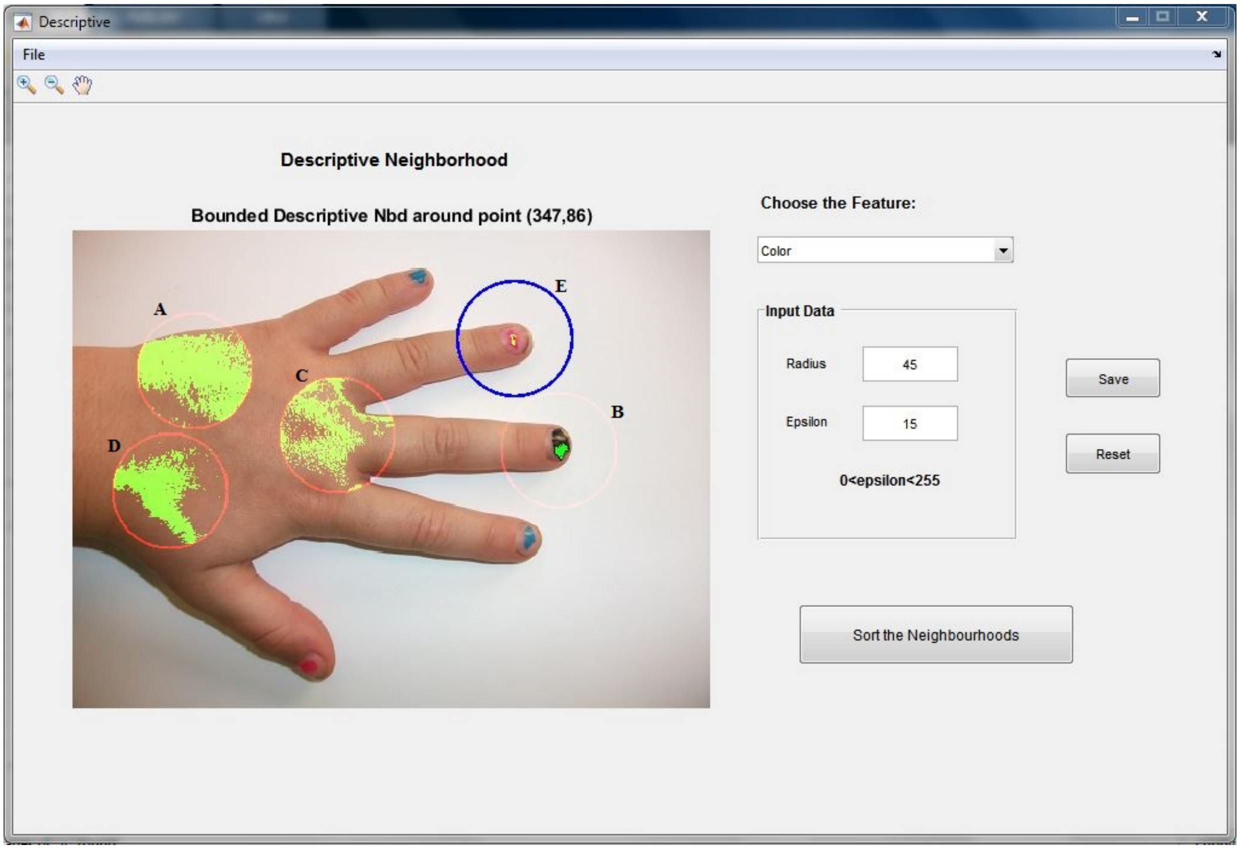


Figure 4.9: Descriptive GUI - Feature: Colour, Radius: 45, Epsilon: 15, Number of Neighbourhoods: 5.

```
>> Descriptive
X(xpos,ypos):(96,111)
X(xpos,ypos):(382,173)
X(xpos,ypos):(208,161)
X(xpos,ypos):(77,205)
X(xpos,ypos):(347,86)
```

Figure 4.10: Centre Pixel Values for Neighbourhoods A, B, C, D, E in Order.

By clicking on the ***Sort the Neighbourhoods*** push-button, the similarity distance between each of the neighbourhoods B, C, D ,E and the reference neighbourhood A[4] is being calculated and the selected neighbourhoods B, C, D and E are being sorted based on their descriptive closeness to neighbourhood A. The result for the sorting process in this example is as follows:

Neighbourhood A, neighbourhood D, Neighbourhood C.

Steps for descriptively sorting neighbourhoods in terms of their closeness to a reference neighbourhood A are as follows,

0- Choose the neighbourhood radius and epsilon value.

1- Start with selecting a descriptive neighbourhood A on the image with radius $R1$ and ε.

2- Insert new descriptive neighbourhoods on the image. Let B be a new descriptive neighbourhood with radius $R2$ and ε and let C be a new descriptive neighbourhood with radius $R3$ and ε.

3- Compute

Dist $(\Phi(A), \Phi(B))$ = distance between the description of centre of A and the description of centre of B.

Dist $(\Phi(A), \Phi(C))$ = distance between the description of centre of A and the description of centre of C.

If Dist $(\Phi(A), \Phi(B)) <$ Dist $(\Phi(A), \Phi(C))$, then neighbourhood B is descriptively closest to neighbourhood A, else neighbourhood C is descriptively closest to neighbourhood A.

[4] Reference neighbourhood is the first selected neighbourhood and the similarity distance is calculated between this neighbourhood and each of the other selected neighbourhoods.

The same as explained in the ***Spatial*** GUI, a structure named *store_ centres* is defined for observing and analysing the sorting process in this GUI, as well. It has different fields and it is accessible through the workspace in MATLAB. It stores the centre pixel value for each of the selected neighbourhoods, the Euclidean distance between the centre pixel of the reference neighbourhood and each of the other selected neighbourhoods, the similarity distance[5] value between the centre pixel of the reference neighbourhood and each of the other selected neighbourhoods and also an *overlap* named field that specifies the neighbourhoods that their description overlaps with the description of the reference neighbourhood. This last field provides the required information for sorting the neighbourhoods. The sorting results for the descriptive neighbourhoods in figure 4.9 were obtained based on the information in the *store_ centres* structure.

This sorting of the neighbourhoods process was used in the classification process for forming the set patterns, making sure that the neighbourhoods in the set pattern are all descriptively close enough to the motif set, which amplifies the classification results.

4.2.3 Classification

The application named ***Classification*** is the main part of this research work which does the pattern-similarity-distance-based classification. Figure 4.11 shows a snapshot of this GUI that opens up after selecting it on ***Start_Up*** GUI.

As seen in figure 4.11, the first query image namely ***Target Image1*** was selected through the ***Target_ image1*** drop down menu. This image represents the images in Class 1. The second query image namely ***Target Image2*** was also selected likewise through the ***Target_ image2*** drop down menu. This image is the representative of the images in Class 2. The test image namely ***Test Image*** which is the image that needs to be classified was also selected through the ***Test_ image*** drop down menu.

After visually classifying all the 146 images in the data set in to two classes based on

[5]Similarity distance was discussed in 2.8

their shape, *Class1* had 65 images and *Class2* had 81 images. This work was tested by selecting 50 random target images (or query images) from each of the two classes being considered. Then 60 test images were randomly selected from *Class1* and 60 test images were randomly selected from *Class2*. These test images were compared to the two target images.

In order to be able to start classifying the ***Test Image*** as being in the same class with ***Target Image1*** or ***Target Image2***, the parameters required for the classification needs to be selected and inserted by the user.

The classification could be done based on two different features which are colour and gradient orientation. These features can be selected through a pop-up menu. The ***Input Data*** panel lets the user input the radius value for the descriptive neighbourhoods in ***Radius*** edit box and the epsilon value in ***Epsilon*** edit box. By selecting each of the two features, the GUI shows the range of the required acceptable epsilon value for that feature. The ***Save*** push-button saves the GUI after all the processes and the ***Reset*** push-button resets the GUI and the MATLAB workspace valuables.

The ***Classify*** push-button in this GUI activates the classification process, once all the necessary parameters has been filled and after forming and selecting the set patterns in each of the images by the user. Finally the classification result, which tells whether the test image belongs to *Class1* or *Class2*, is displayed in ***Output Result*** panel.

Figure 4.11 is the first classification example presented, which is based on colour as the feature. In this example, *"Class1-23.jpg"* from *Class1* and *"Class2-42.jpg"* from *Class2* were selected as the two target images and *"Class2-18.jpg"* from *Class2* was selected as the test image. Set pattern in each of the target images and the test image contains three neighbourhoods. The similarity distance between the motif neighbourhood and all the other selected neighbourhoods in the set pattern in each image is calculated based on the explanation in the ***Descriptive*** GUI, making sure that the neighbourhoods in the set pattern are all descriptively close enough to the motif neighbourhood. Once all the necessary parameters are filled and after forming the set patterns, a click on ***Classify*** push-button activates the classification process by calculating the pattern similarity distance[6]. As seen in the ***Output Result*** panel, the ***Test Image*** belongs to *Class2*.

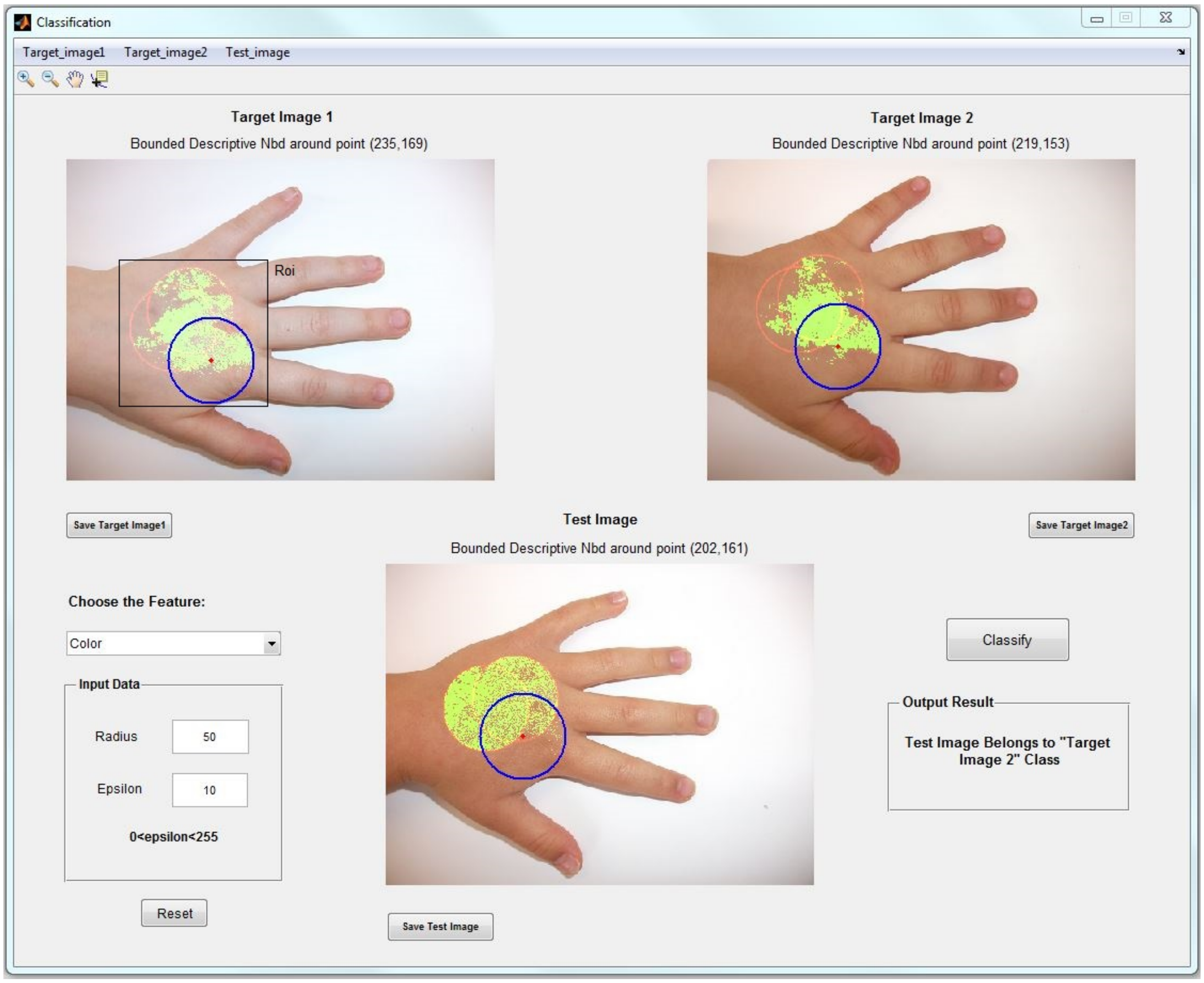
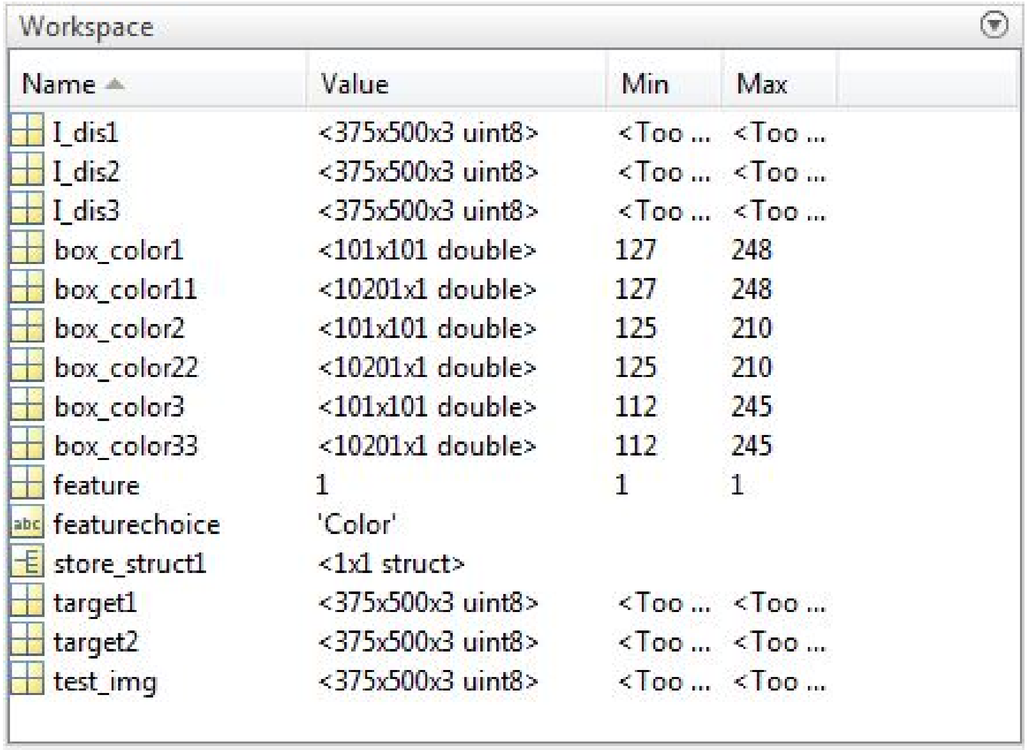


Figure 4.11: Classification GUI - Feature: Colour, Radius: 50, Epsilon: 10, Number of Neighbourhoods in Each Set Pattern: 3.

Figure 4.12 shows an overview of the MATLAB workspace for this example, which contains all the valuable and structures that are defined and used in this example.

[6]Pattern similarity distance was discussed in 2.9.

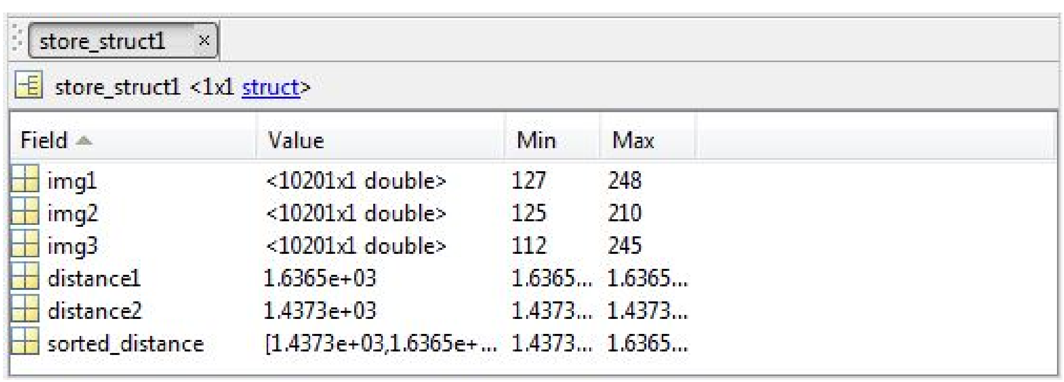


Workspace

Name	Value	Min	Max
I_dis1	<375x500x3 uint8>	<Too ...	<Too ...
I_dis2	<375x500x3 uint8>	<Too ...	<Too ...
I_dis3	<375x500x3 uint8>	<Too ...	<Too ...
box_color1	<101x101 double>	127	248
box_color11	<10201x1 double>	127	248
box_color2	<101x101 double>	125	210
box_color22	<10201x1 double>	125	210
box_color3	<101x101 double>	112	245
box_color33	<10201x1 double>	112	245
feature	1	1	1
featurechoice	'Color'		
store_struct1	<1x1 struct>		
target1	<375x500x3 uint8>	<Too ...	<Too ...
target2	<375x500x3 uint8>	<Too ...	<Too ...
test_img	<375x500x3 uint8>	<Too ...	<Too ...

Figure 4.12: An Overview of MATLAB Workspace for this Example.

For observing and analysing the classification process in this GUI, a structure named *store_ struct1* is defined in this GUI which has different fields and it is accessible through the MATLAB workspace. It stores each of the images, the calculated distance and the classification result. Figure 4.13 shows the different fields of the *store_ struct1* structure which clarifies the classification process.



store_struct1

store_struct1 <1x1 struct>

Field	Value	Min	Max
img1	<10201x1 double>	127	248
img2	<10201x1 double>	125	210
img3	<10201x1 double>	112	245
distance1	1.6365e+03	1.6365...	1.6365...
distance2	1.4373e+03	1.4373...	1.4373...
sorted_distance	[1.4373e+03,1.6365e+...	1.4373...	1.6365...

Figure 4.13: *"store_ struct1"* Structure for Classification GUI.

The *store_ struct1* structure, as shown in the above figure, has five fields,

- Since the selected feature for this example is colour, field $img1$ contains the colour value of all the pixels in the ***Target Image1*** (linearised).

- Since the selected feature for this example is colour, field $img2$ contains the colour value of all the pixels in the ***Target Image2*** (linearised).

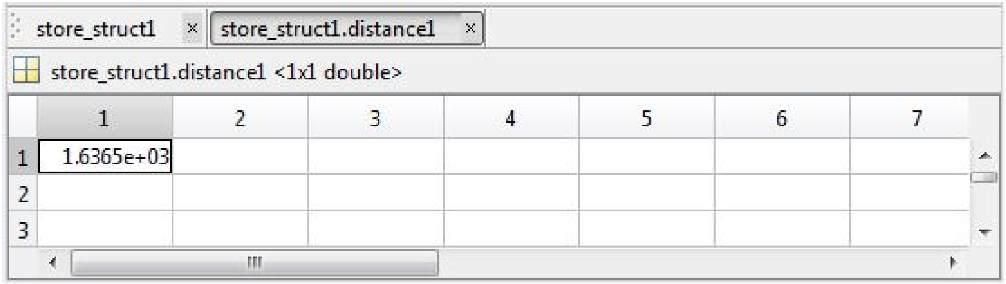
- Since the selected feature for this example is colour, field $img3$ contains the colour value of all the pixels in the ***Test Image*** (linearised).

- Field *distance1* contains the pattern similarity distance between the set pattern in ***Test Image*** and the set pattern in ***Target Image1***.

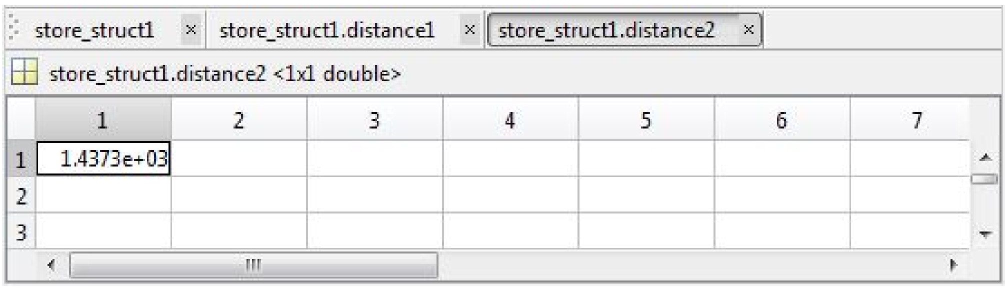
- Field *distance2* contains the pattern similarity distance between the set pattern in ***Test Image*** and the set pattern in ***Target Image2***.

- Field *sorted_ distance* contains the result of sorting the pattern similarity distances which is in fact the classification result and it is reflected in the ***Output Result*** panel in the GUI.

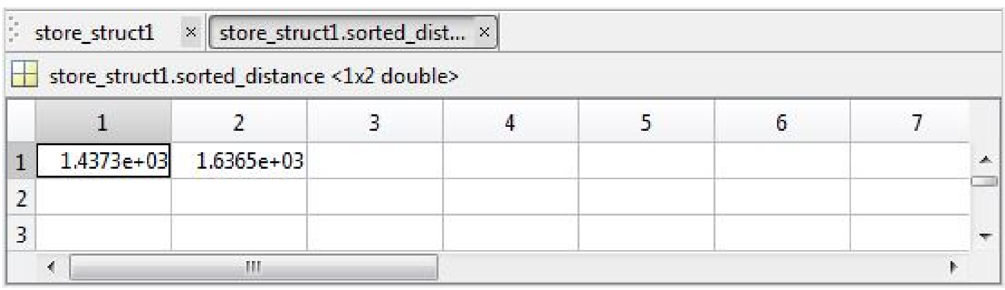
Figure 4.14 shows the different fields of the *store_ struct1* structure which clarifies the classification process.



(a) *distance1* Field in the Structure.



(b) *distance2* Field in the Structure.



(c) *sorted_ distance* Field in the Structure.

Figure 4.14: *"store_ struct1"* Different Fields.

Figure 4.15 presents another classification example, using the proposed method, which is done based on gradient orientation as the feature, trying to classify based on shape. In this example, *"Class1-33.jpg"* from *Class1* and *"Class2-38.jpg"* from *Class2* were selected as the two target images and *"Class1-27.jpg"* from *Class1* was selected as the test image. Set pattern in each of the target images and the test image contains three neighbourhoods. The selected region-of-interest in each of the images is also shown in the figure. As seen in the ***Output Result*** panel, the ***Test Image*** belongs to *Class1*, as expected.

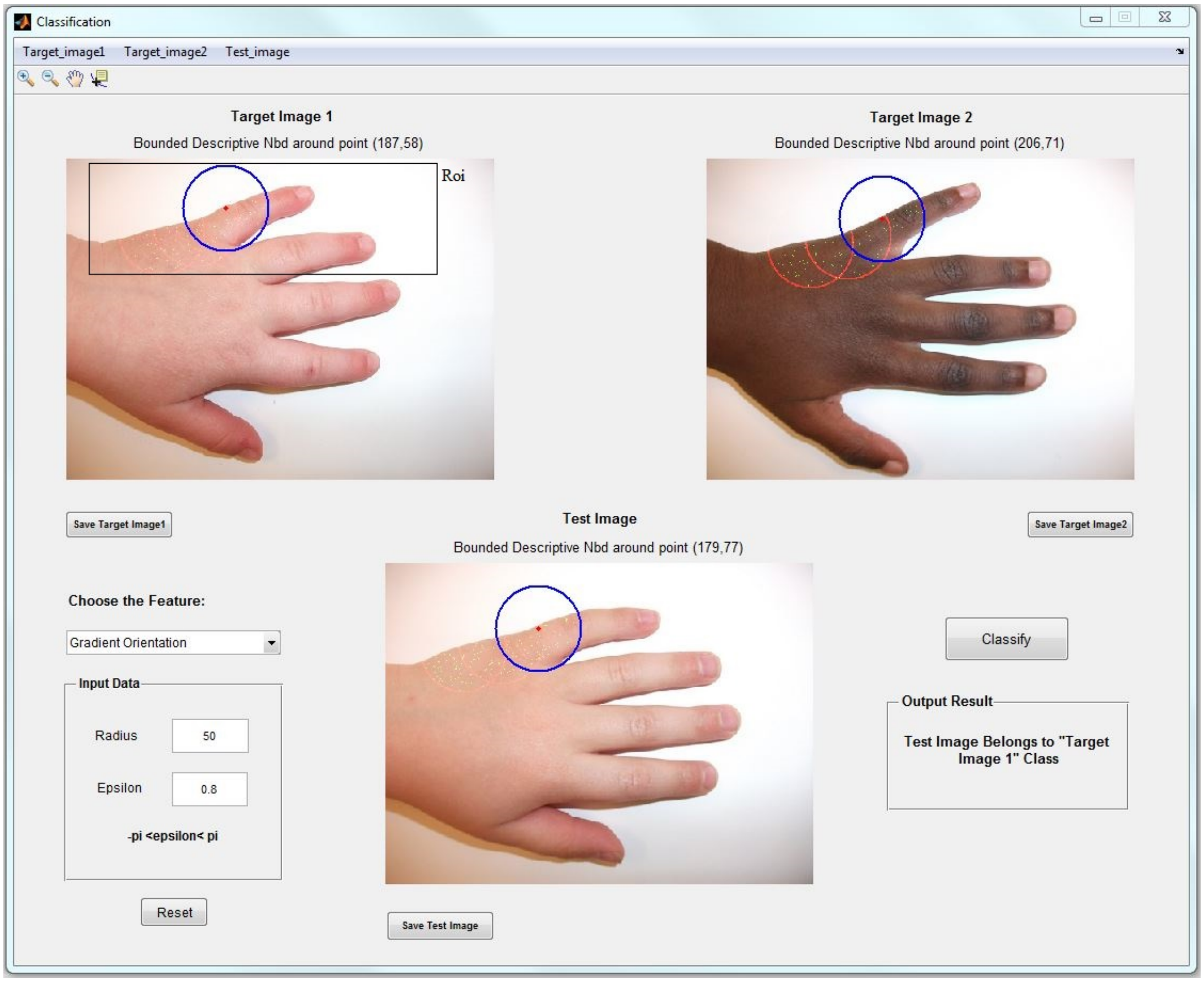


Figure 4.15: Classification GUI - Feature: Gradient Orientation, Radius: 50, Epsilon: 0.8, Number of Neighbourhoods in Each Set Pattern: 3.

For observing and analysing the classification process in this GUI, a structure named *store_ struct2* is defined in this GUI which has different fields and it is accessible through the MATLAB workspace. Figure 4.16 shows the different fields of the *store_ struct2* structure which clarifies the classification process.

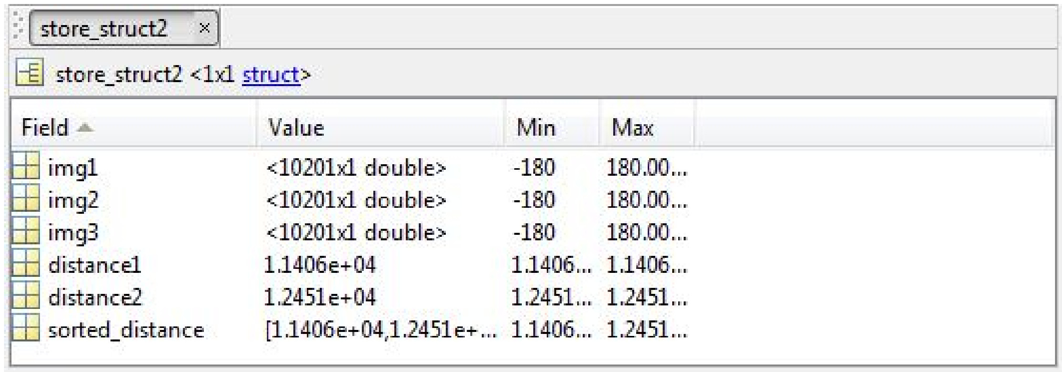
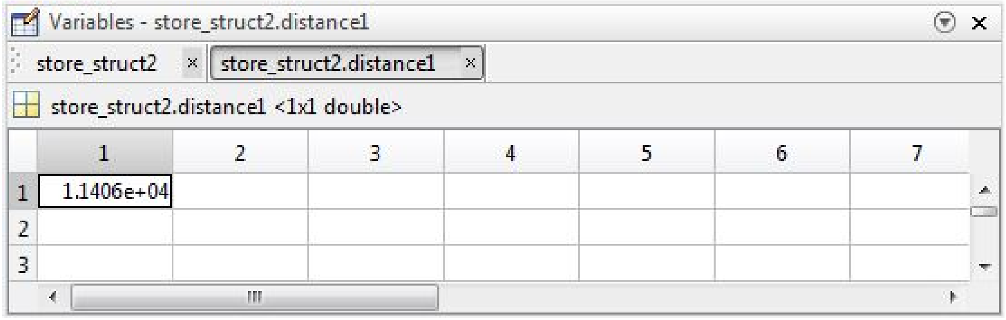
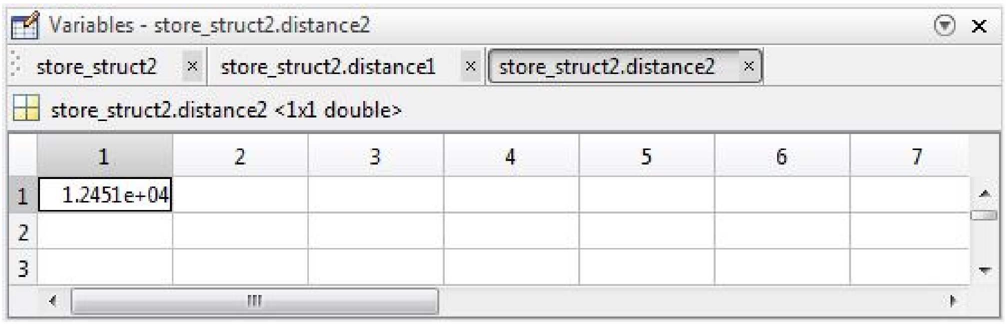


Figure 4.16: *"store_ struct2"* Structure for Classification GUI.

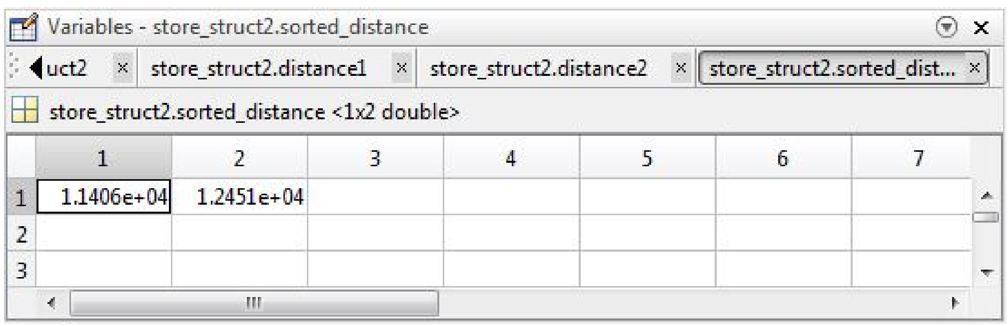
Figure 4.17 shows the different fields of the *store_ struct2* structure which clarifies the classification process.



(a) *distance1* Field in the Structure.



(b) *distance2* Field in the Structure.



(c) *sorted_ distance* Field in the Structure.

Figure 4.17: *"store_ struct2"* Different Fields.

Figure 4.18 and 4.19 show two more classification results, using the proposed pattern-similarity-distance-based method, when the gradient orientation is the selected classification probe function.

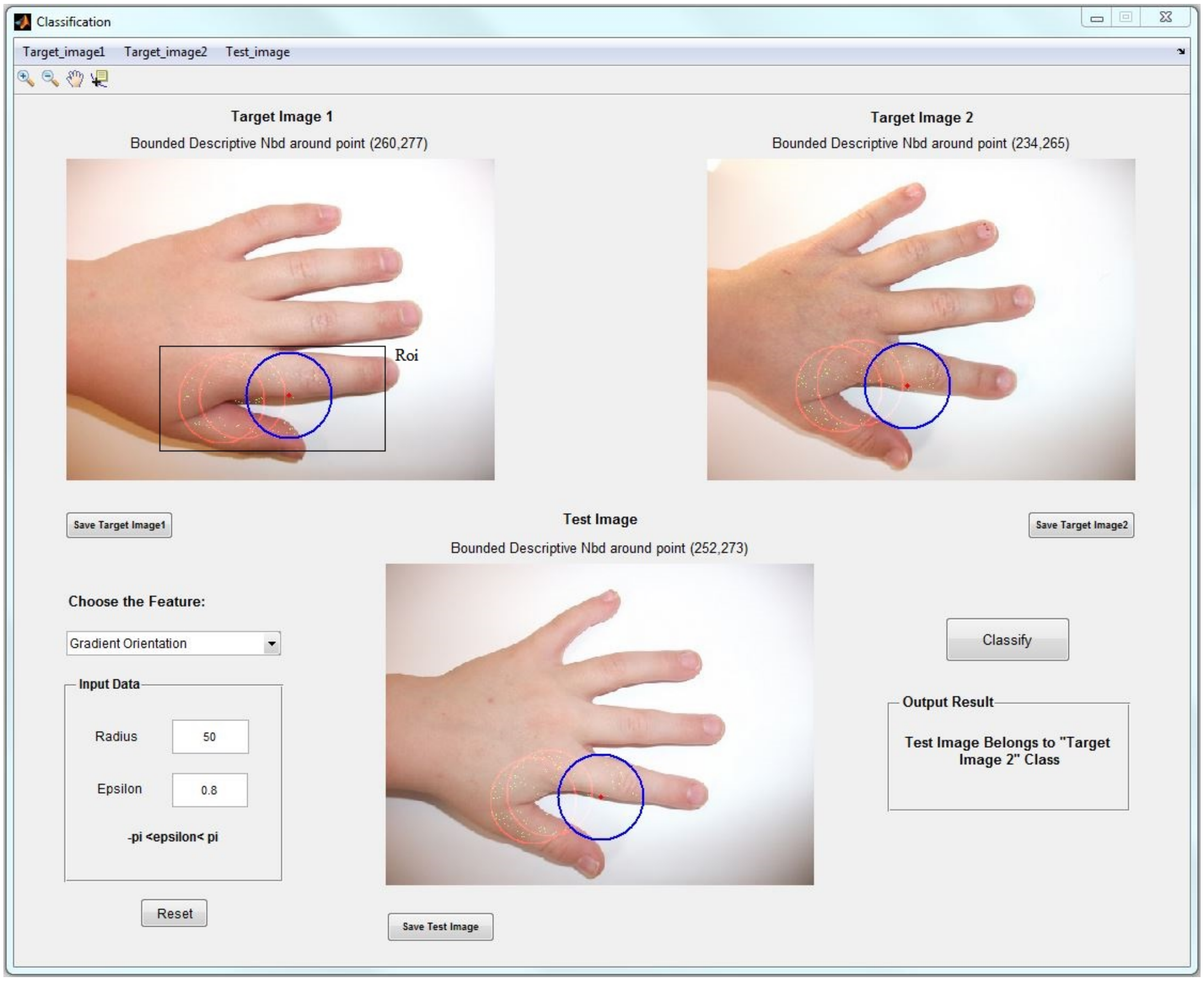


Figure 4.18: Classification GUI - Feature: Gradient Orientation, Radius: 50, Epsilon: 0.8, Number of Neighbourhoods in Each Set Pattern: 3.

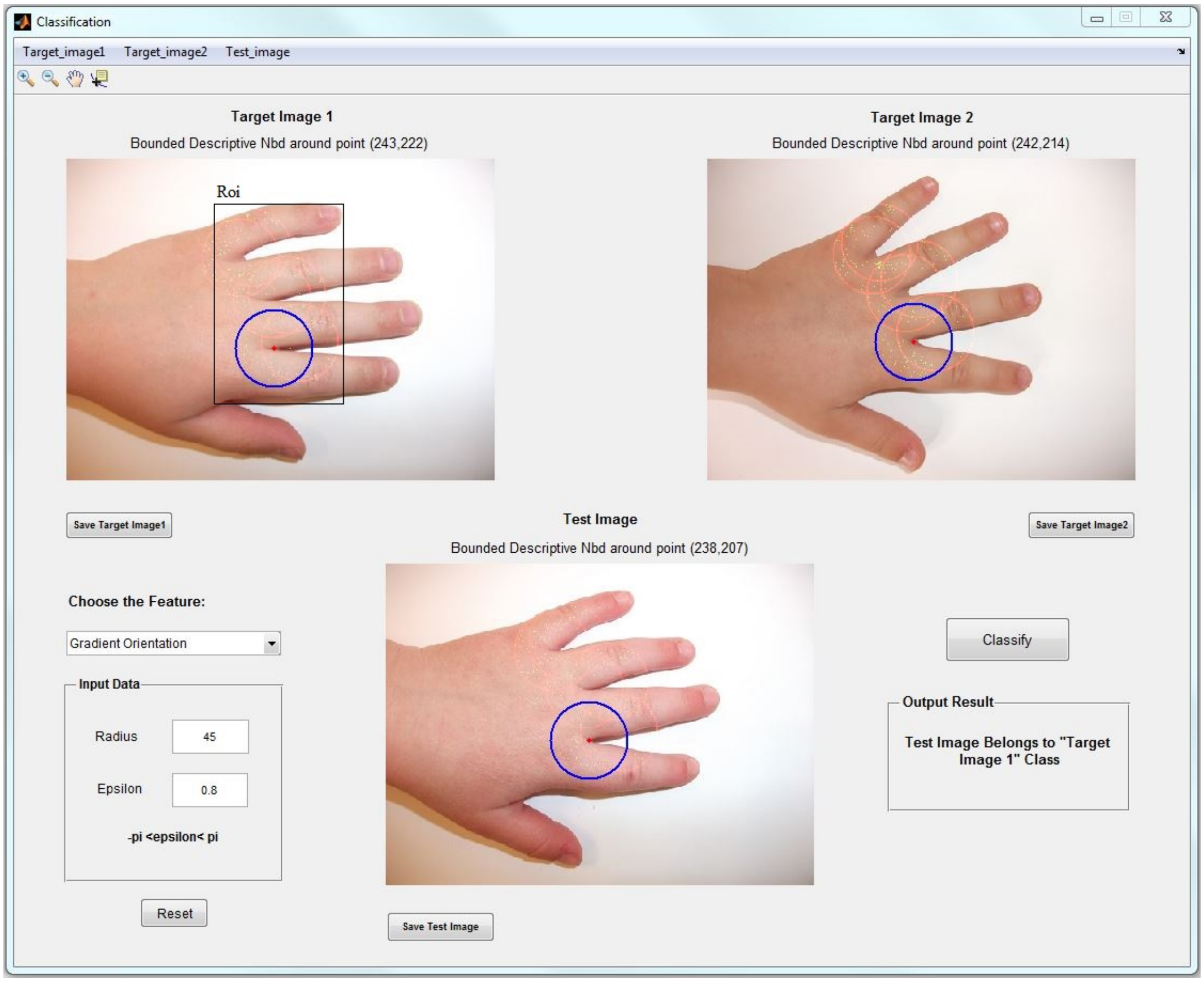


Figure 4.19: Classification GUI - Feature: Gradient Orientation, Radius: 45, Epsilon: 0.8, Number of Neighbourhoods in Each Set Pattern: 6.

4.3 Results and Analysis

This proposed classification method which is initially based on nearness theory, was tested on 146 pictures of the left hands of children between 3.5 to 5.5 years old. For testing the validity of the proposed method, these images have been visually classified based on two different features: colour and gradient orientation. Experiments were carried out on two classes of digital images, namely *Class1* and *Class2*.

When gradient orientation was the probe function used in the classification process, the 146 hand images were visually classified into two classes based on their shape. *Class1* had 65 images and *Class2* had 81 images. This work was tested by selecting 50 random target images (or query images) from each of the two classes being considered. Then 60 test images were randomly selected from *Class1* and 60 test images were randomly selected from *Class2*. These test images were compared to the two target images. Using the proposed method, out of 120 randomly selected test images from the two class, 98 images were classifies correctly. So in this case, the pattern-similarity-distance-based classification showed 81.6% successful classification.

When colour was the probe function used in the classification process, the 146 hand images were visually classified into two classes based on their colour. *Class1* had 79 images and *Class2* had 67 images. This work was tested by selecting 50 random target images (or query images) from each of the two classes being considered. Then 60 test images were randomly selected from *Class1* and 60 test images were randomly selected from *Class2*. These test images were compared to the two target images. The application of the proposed pattern-similarity-distance-based classification method, showed 90.8% successful classification which means 109 test images out of 120 images were classified correctly.

4.3.1 Accuracy, Sensitivity and Specificity Analysis of the Results

A binary classification as the one proposed in this research work classifies the elements of a given set into two groups, based on a classification method and produces output with two class values or labels [57] [58].

Accuracy, specificity and sensitivity that statistically measure the performance of a binary classification test are used in this research work to analyse the performance of the proposed classification method.

For a binary classifier that classifies instances into two classes, any single classification can fall into one of four buckets:

1. **True Positives(TP):** These are cases which were classified as *Class1* and they do belong to this class.

2. **True Negatives(TN):** These are cases which were classified as *Class2* and they do belong to this class.

3. **False Positives(FP):** These are cases which were classified as *Class1*, but they do not actually belong to that class.

4. **False Negatives(FN):** These are cases which were classified as *Class2*, but they do not actually belong to that class.

Accuracy is the proportion of the total number of classifications that were correct and it measures how well a binary classifier classifies the test images [59]. The best accuracy is 1.0, whereas the worst is 0.0,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Sensitivity measures the proportion of the test images from *Class1* that are correctly identified as such [59]. The best sensitivity is 1.0, whereas the worst is 0.0. Mathematically, sensitivity can be stated as:

$$Sensitivity = \frac{TP}{TP + FN} \tag{4.2}$$

Specificity measures the proportion of the test images from *Class2* that are correctly identified as such [59]. The best specificity is 1.0, whereas the worst is 0.0. Mathematically, specificity can be stated as:

$$Specificity = \frac{TN}{TN + FP} \tag{4.3}$$

A confusion matrix is a table that describes the performance of a classification method on a set of test data for which the true values are known. Figure 4.20 shows the confusion matrix for the proposed classification method, when gradient orientation was the selected probe function used in the classification process of 120 test images which 60 of them were randomly selected from *Class1* and 60 of them were randomly selected from *Class2*,

Total # of Test Images = 120		Actual Class	
		Class1	Class2
Predicted Class	Class1 (n=60)	TP = 51	FN = 9
	Class2 (n=60)	FP = 13	TN = 47

Figure 4.20: Confusion Matrix for the Proposed Classifier, When Gradient Orientation is the Probe Function

According to 4.20, accuracy, sensitivity and specificity measures for the proposed classification method, when gradient orientation is the selected feature are calculated as follows,

$$Accuracy = \frac{51}{51 + 47 + 13 + 9} = 0.82$$

$$Sensitivity = \frac{51}{51 + 9} = 0.85$$

$$Specificity = \frac{47}{47 + 13} = 0.78$$

Figure 4.21 shows the confusion matrix for the proposed classification method, when colour was the selected probe function used in the classification process of 120 test images which 60 of them were randomly selected from *Class1* and 60 of them were randomly selected from *Class2*,

Total # of Test Images = 120		Actual Class	
		Class1	Class2
Predicted Class	Class1 (n=60)	TP = 53	FN = 7
	Class2 (n=60)	FP = 4	TN = 56

Figure 4.21: Confusion Matrix for the Proposed Classifier, When Colour is the Probe Function

According to 4.21, accuracy, sensitivity and specificity measures for the proposed classification method, when colour is the selected feature are calculated as follows,

$$Accuracy = \frac{53}{53 + 56 + 4 + 7} = 0.91$$

$$Sensitivity = \frac{53}{53 + 7} = 0.88$$

$$Specificity = \frac{56}{56+4} = 0.93$$

In general, the high value for all the three factors, accuracy, sensitivity and specificity is the specification of a good binary classification test, whereas the low value for these factors is the specification of a poor binary classification test. According to the above calculations for the proposed classification method, the three factors, accuracy, sensitivity and specificity have a high value which confirms the good performance of the proposed method.

Chapter 5

Conclusion and Future Directions

5.1 Conclusion

The focus of the research work reported in this thesis is on studying the nearness theory, various neighbourhoods of points in proximity spaces and similarity measures which led to discovering the set patterns in digital images. Set pattern discovery and then calculating a unique similarity distance measure is the basis for the similarity-distance-based classification method, proposed in this research.

A set pattern in an image is defined as a collection of sets (denoted by $\mathfrak{P}(M)$) that all of the members of the collection have common properties and are all near a given set called as a pattern generator or motif set. Selecting the motif set and afterwards generating the salient set pattern in each of the target images and the test image paves the way for the proposed classification process. Extraction of the salient patterns in digital images, as a representative of the characteristics of the selected region in the image, is one of the main parts of this work. The similarity distance measure that is used in this research as a tool for classifying the images, uses the similarity distance concept that was introduced by J. F. Peters in [7].

In fact, in this research work, some of the concepts in topology of digital images such as nearness theory, neighbourhoods, similarity distance and set patterns, are gathered up and applied to introduce a classification method which does successful classification with high accuracy, sensitivity and specificity.

5.2 Future Directions

The proposed method which is based on nearness theory in digital images can be applied on many image analysis applications such as medical image analysis or microfossils images analysis. The motif set selection process and the type of selected neighbourhoods can be done differently, based on the specific application of the generated patterns.

Selecting a threshold value for the calculated similarity distances between the set patterns in the test image and each of the target images in the classification phase can be further explored. A threshold can be defined for clarifying the nearness between the set patterns in the test image and each of the target images, i.e. if the similarity distance between the set pattern in test image and the set pattern in the target image is less than the defined threshold, then the test image can be classified as the same class of the target image, otherwise it does not belong to the same class of the target image.

Appendix A

User Manual for Classification GUI

The four different Graphical user Interfaces (GUIs) which are developed in this work are as follows:

Start_ Up GUI

The developed MATLAB program starts by clicking on ***Start_ UP.m*** file in the main folder. Clicking on this file opens up the ***Start_ UP*** GUI which offers the user, three options to choose from:

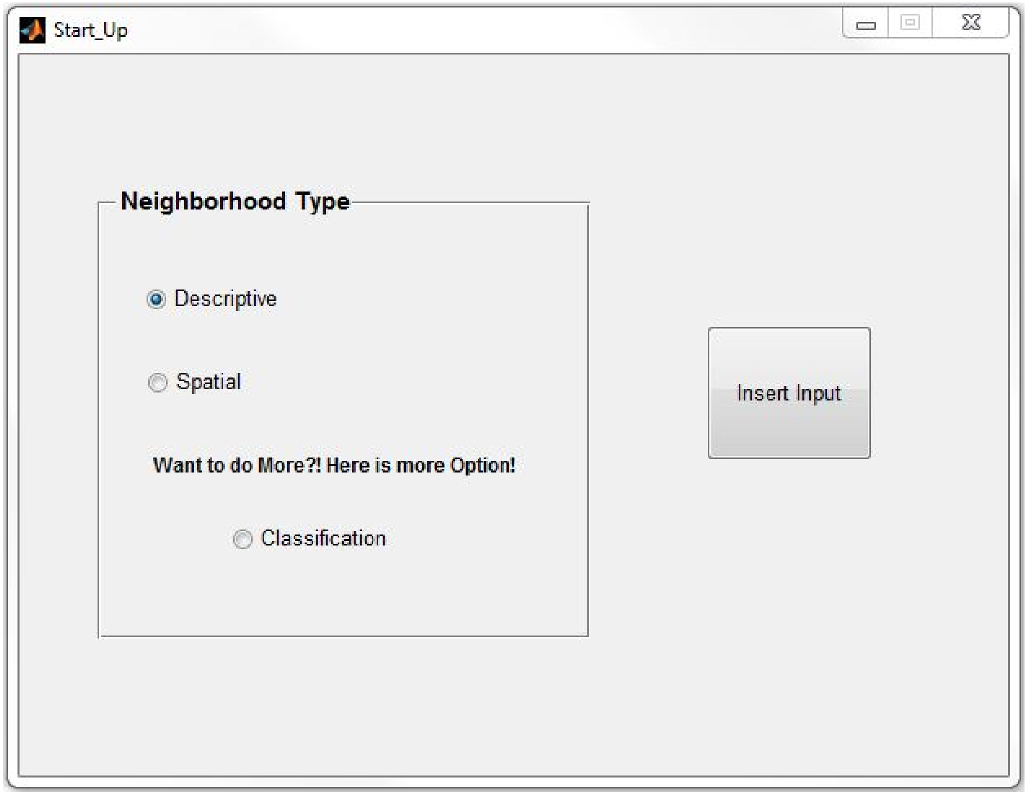


Figure A.1: Start_Up GUI.

Spatial GUI

If the ***Spatial Neighbourhood*** option is selected in the ***Start_ UP*** GUI, clicking on the ***Insert Input*** push-button, opens up the ***Spatial*** GUI as shown in figure A.2. An image is loaded in the GUI as shown in figure A.3, by browsing and choosing an image via the ***File*** drop down menu. After inserting the ***Radius*** value in the edit box, 6 neighbourhoods are selected on the image.

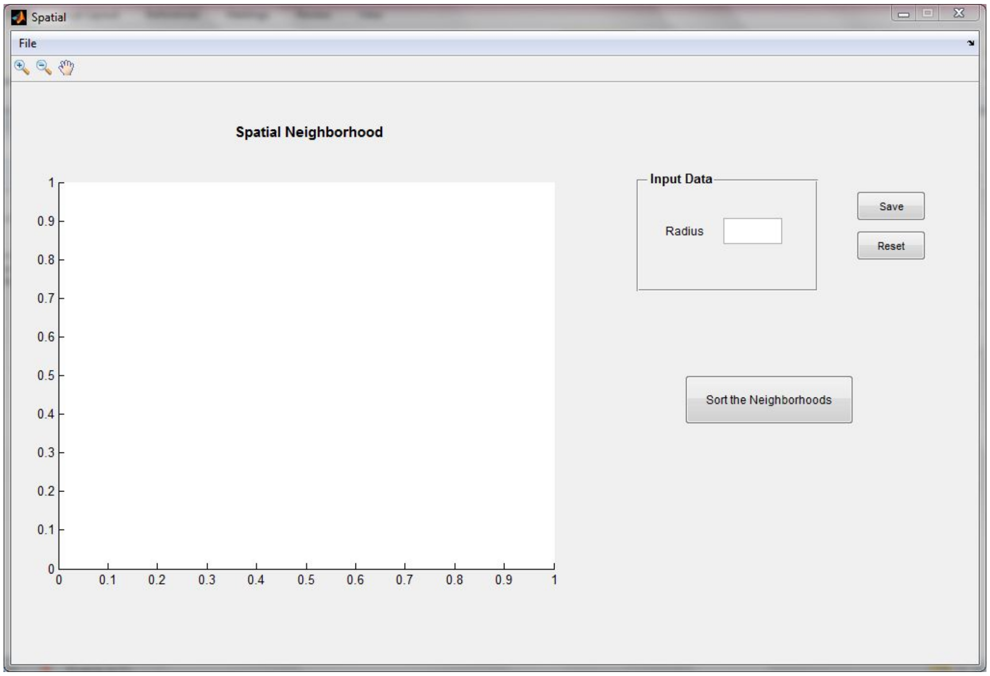


Figure A.2: Spatial GUI.

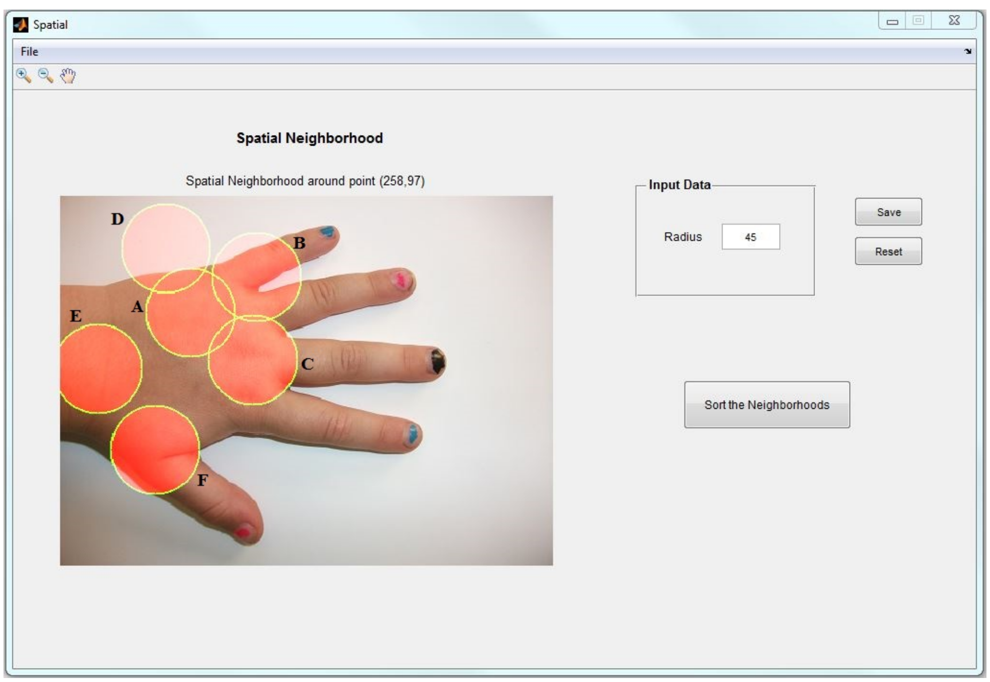


Figure A.3: Spatial Set Pattern in Spatial GUI (The set pattern includes neighbourhoods A, B, C, D.).

Descriptive GUI

If the ***Descriptive Neighbourhood*** option is selected in the ***Start_ UP*** GUI, clicking on the ***Insert Input*** push-button, opens up the ***Descriptive*** GUI as shown in figure A.4.

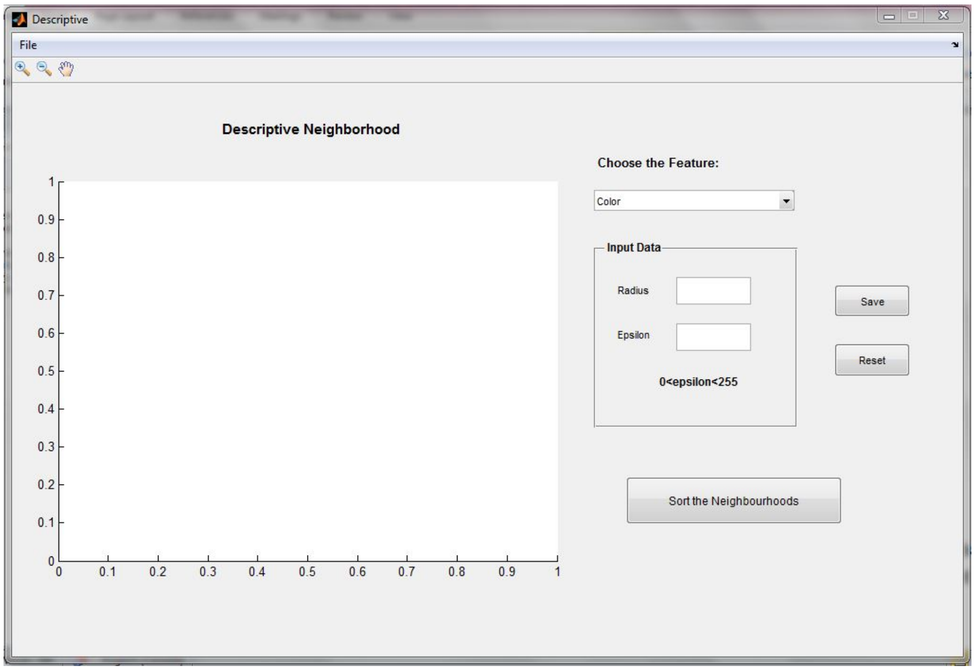


Figure A.4: Descriptive GUI.

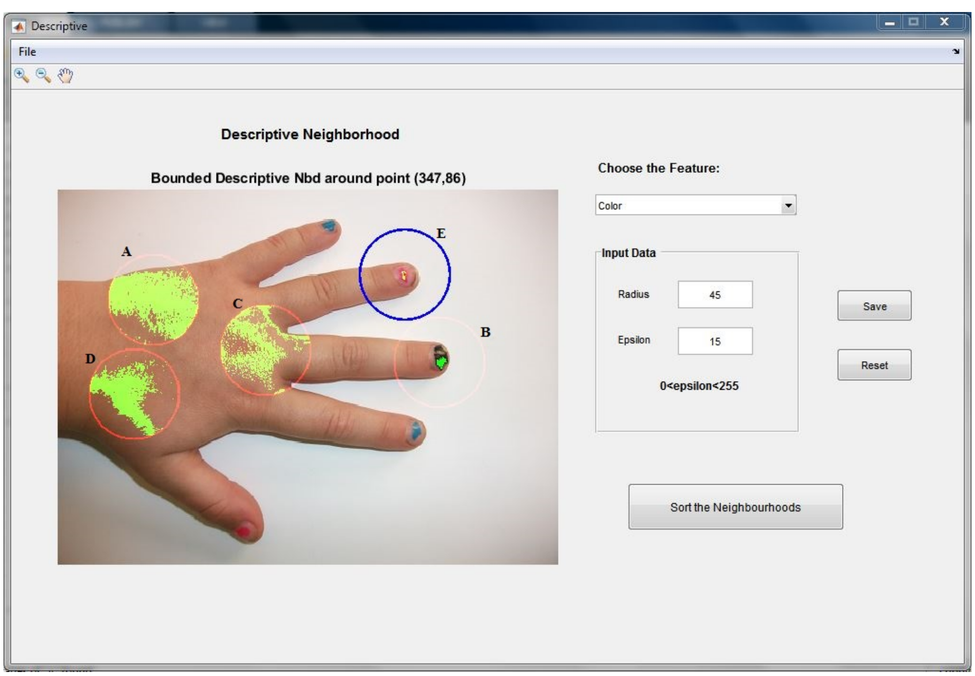


Figure A.5: Descriptive Set Pattern in Descriptive GUI (The set pattern includes neighbourhoods A, D, C.).

An image is loaded in the GUI as shown in figure A.5, by browsing and choosing an image via the ***File*** drop down menu. After choosing the desired ***Feature***, inserting the ***Radius*** value and ***Epsilon*** value in the edit boxes, 5 neighbourhoods are selected on the image.

Classification GUI

If ***Classification*** option is selected in the ***Start_UP*** GUI, clicking on the ***Insert Input*** push-button, opens up the ***Classification*** GUI as shown in figure A.6.

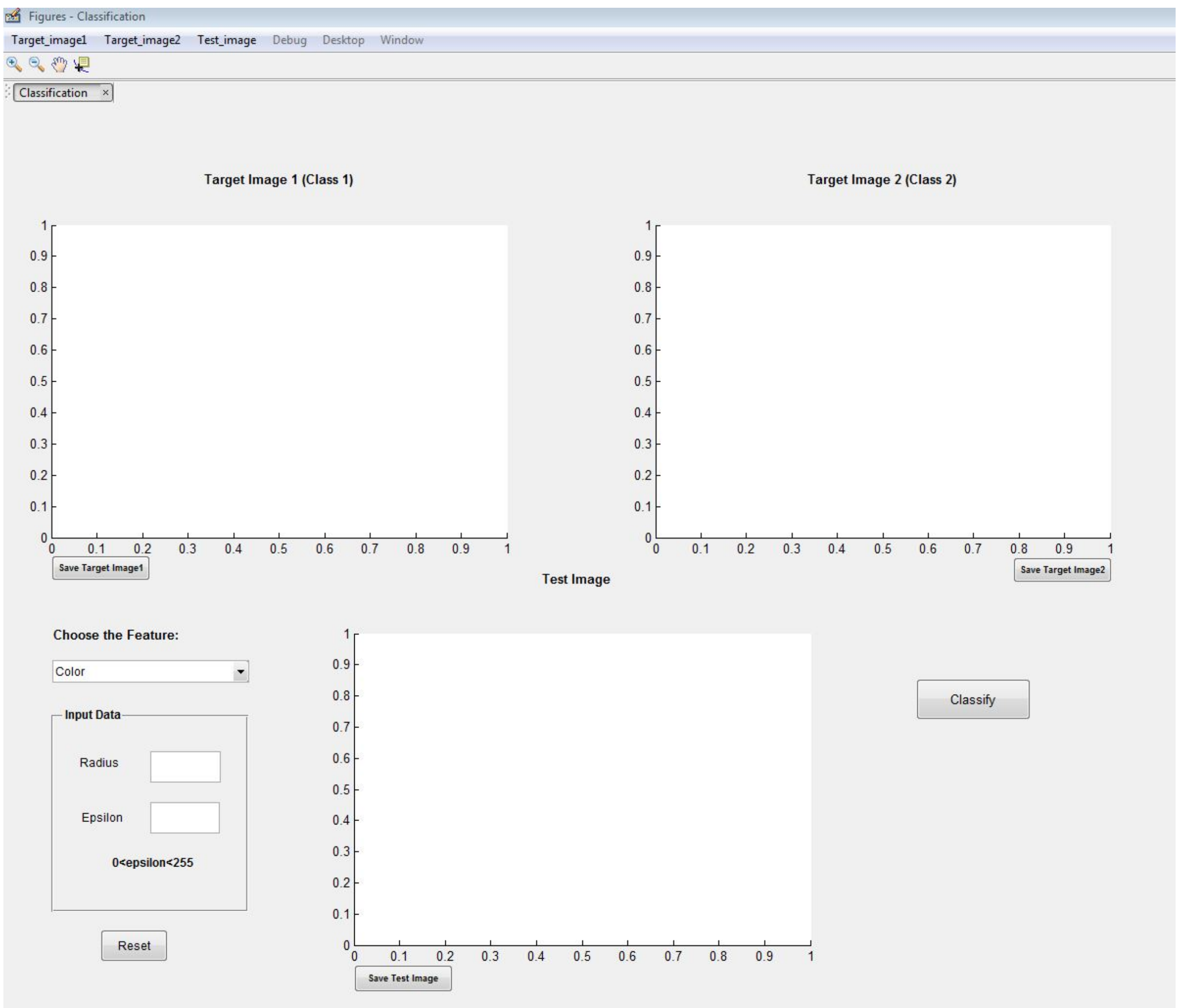


Figure A.6: Classification GUI.

Target Image1

The ***Target Image1*** is loaded as shown in figure A.7, by choosing an image via the ***File*** drop down menu.

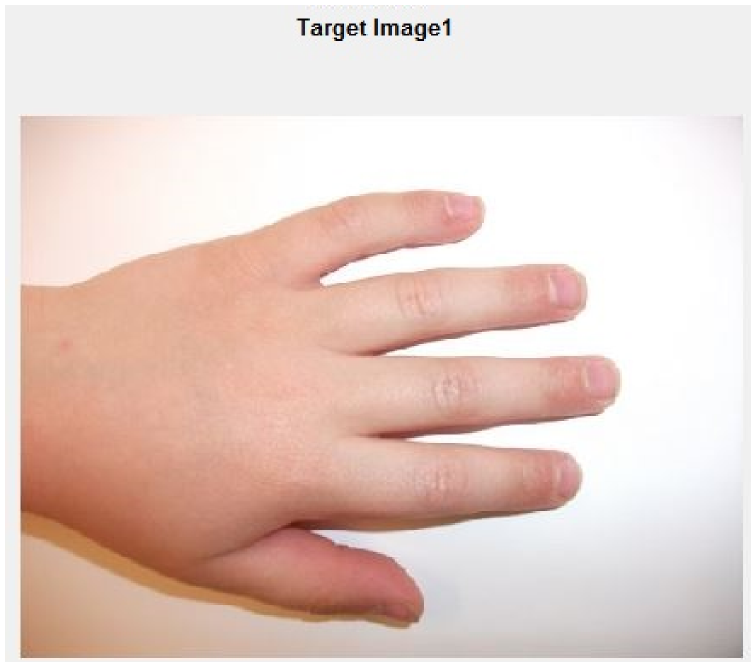


Figure A.7: Target Image1.

Target Image2

The ***Target Image2*** is loaded as shown in figure A.8, by choosing an image via the ***File*** drop down menu.

Figure A.8: Target Image2.

Test Image

The ***Test Image*** is loaded as shown in figure A.9, by choosing an image via the ***File*** drop down menu.

Figure A.9: Test Image.

Feature Selection

As shown in figure A.10, a feature or probe function is selected through a pop-up menu, so the classification process can be done based on that selected feature.

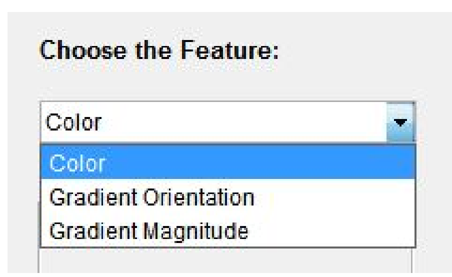


Figure A.10: The Pop-up Menu for Probe Function Selection.

Classification Parameters

The parameters for the proposed classification method are inserted in the ***Input Data*** panel. The ***Radius*** and ***Epsilon*** are the two parameters for this classification method.

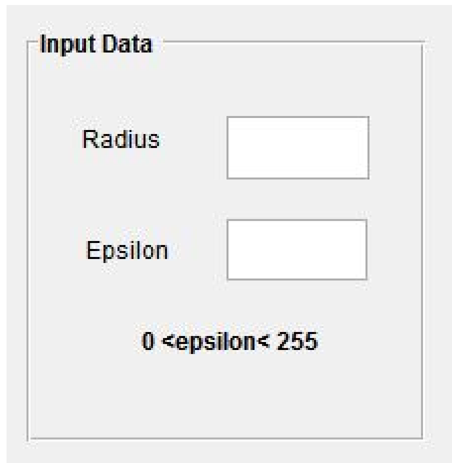


Figure A.11: Classification Parameters.

Set Pattern in Target Image1

The generated set pattern in the ***Target Image1*** is shown in figure A.12.

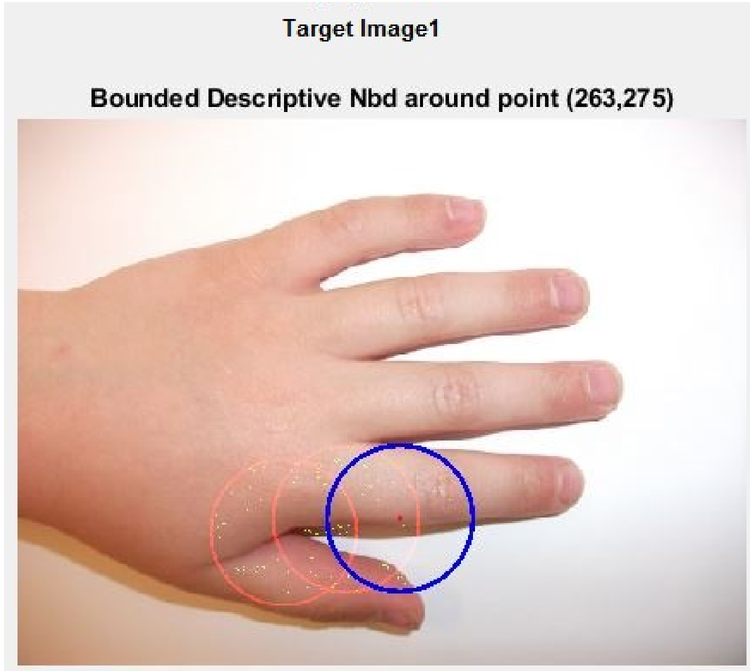


Figure A.12: Set Pattern in Target Image1.

Set Pattern in Target Image2

The generated set pattern in the ***Target Image2*** is shown in figure A.13.

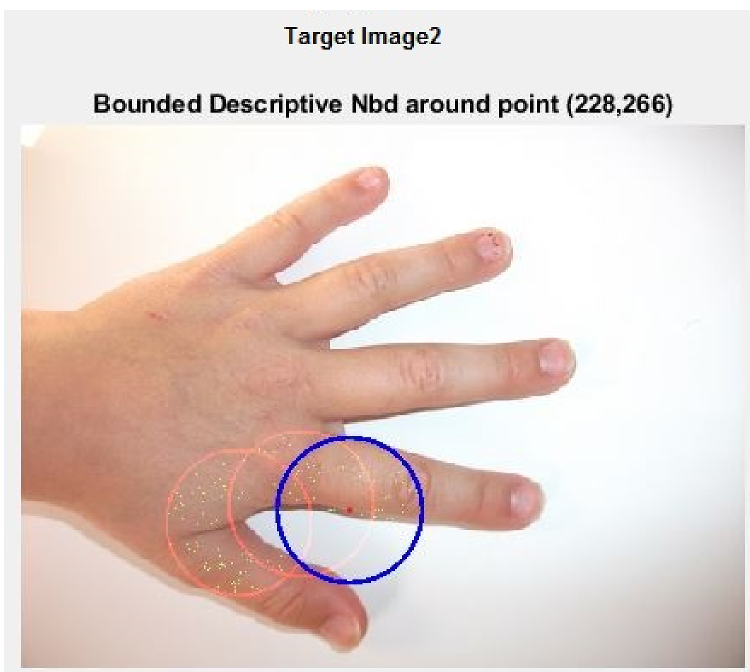


Figure A.13: Set Pattern in Target Image2.

Set Pattern in Test Image

The generated set pattern in the ***Test Image*** is shown in figure A.14.

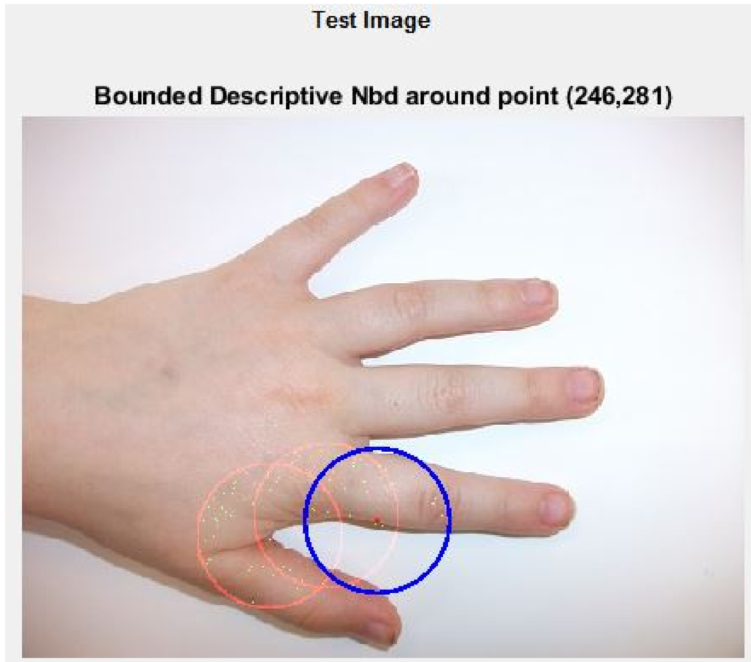


Figure A.14: Set Pattern in Test Image.

Output Result

The result of the pattern-similarity-distance-based classification is shown in this ***Output Result*** panel.

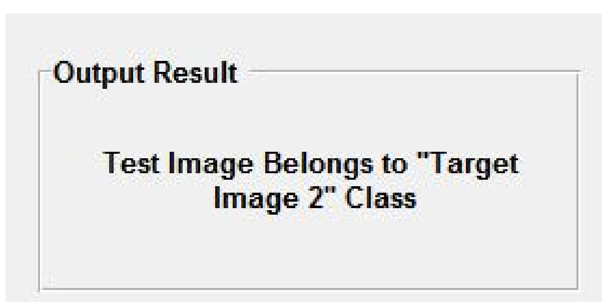


Figure A.15: Classification Result.

References

[1] M. K. Mandal, “The human visual system and perception,” in *Multimedia Signals and Systems*. Springer, 2003, pp. 33–56.

[2] T. Cornsweet, *Visual perception*. Academic press, 2012.

[3] S. J. Prince, *Computer vision: models, learning, and inference*. Cambridge University Press, 2012.

[4] J. F. Peters, *Foundations of Computer Vision: Computational Geometry, Visual Image Structures and Object Shape Detection*. Springer, 2017, vol. 124.

[5] J. Peters and S. Naimpally, “Approach spaces for near families,” *General Mathematics Notes*, vol. 1, no. 1, 2010.

[6] S. Naimpally, J. Peters, and M. Wolski, “Foreword,” *Mathematics in Computer Science*, vol. 7, no. 1, pp. 1–2, 2013.

[7] J. F. Peters, *Topology of digital images: visual pattern discovery in proximity spaces*. Springer Science & Business Media, 2014, vol. 63.

[8] J. Peters and S. Naimpally, “Applications of near sets,” *Amer. Math. Soc. Notices*, vol. 59, no. 4, 2012.

[9] S. Gupta and K. Patnaik, “Enhancing performance of face recognition system by using near set approach for selecting facial features.” *Journal of Theoretical & Applied Information Technology*, vol. 4, no. 5, 2008.

[10] C. J. Henry, *Near sets: theory and applications*. University of Manitoba (Canada), 2011.

[11] J. F. Peters, *Computational proximity*. Springer, 2016.

[12] L. G. Shapiro and G. Stockman, “Computer vision,” *Computer Vision, Prentice hall. ISBN 0-13-030796-3*, 2002.

[13] W. E. L. Grimson and J. L. Mundy, “Computer vision applications,” *Communications of the ACM*, vol. 37, no. 3, pp. 45–51, 1994.

[14] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[15] M. Rodd and Q. Wu, “Knowledge-based vision systems in real-time control,” *Annual Review in Automatic Programming*, vol. 15, pp. 13–18, 1989.

[16] A. Hanson, *Computer vision systems*. Elsevier, 1978.

[17] R. Klette, *Concise computer vision*. Springer, 2014.

[18] A. Pandey, V. Sharma, S. Paanchbhai, N. Hedaoo, and S. Zade, "Optical character recognition (ocr)," *International Journal of Engineering and Management Research*, vol. 7, 2017.

[19] M. M. Deza and E. Deza, "Encyclopedia of distances," in *Encyclopedia of Distances*. Springer, 2009, pp. 1–583.

[20] S. A. Naimpally, "Near and far. a centennial tribute to frigyes riesz," , vol. 6, no. 0, pp. 1–10, 2009.

[21] S. A. Naimpally and B. D. Warrack, *Proximity spaces*. Cambridge University Press Cambridge, 1970, vol. 69.

[22] E. Cech, *Topological Spaces*. Wiley-Interscience (London/New York), 1966.

[23] G. Peters, P. Lingras, D. Ślzak, and Y. Yao, *Rough sets: Selected methods and applications in management and engineering*. Springer Science & Business Media, 2012.

[24] S. Ramanna, A. H. Meghdadi, and J. F. Peters, "Nature-inspired framework for measuring visual image resemblance: A near rough set approach," *Theoretical Computer Science*, vol. 412, no. 42, pp. 5926–5938, 2011.

[25] J. F. Peters, "Near sets. special theory about nearness of objects," *Fundamenta Informaticae*, vol. 75, no. 1-4, pp. 407–433, 2007.

[26] A. Di Concilio, "Point-free geometries: proximities and quasi-metrics," *Mathematics in Computer Science*, vol. 7, pp. 31–42, 2013.

[27] ——, "Proximity: a powerful tool in extension theory, function spaces, hyperspaces, boolean algebras and point-free geometry," *Beyond Topology*, vol. 486, pp. 89–112, 2009.

[28] G. Gerla *et al.*, "Quasi-metric spaces and point-free geometry," *Mathematical Structures in Computer Science*, vol. 16, no. 1, pp. 115–137, 2006.

[29] M. Wolski, "Toward foundations of near sets:(pre-) sheaf theoretic approach," *Mathematics in Computer Science*, vol. 7, no. 1, pp. 125–136, 2013.

[30] Wolski, "Perception and classification. a note on near sets and rough sets," *Fundamenta Informaticae*, vol. 101, no. 1-2, pp. 143–155, 2010.

[31] M.Wolski, "Granular computing: Topological and categorical aspects of near and rough set approaches to granulation of knowledge." *Trans. Rough Sets*, vol. 16, pp. 34–52, 2013.

[32] R. Gonzalez and R. Woods, *Digital Image Processing*. Upper Saddle River, New Jersey: Pearson Education, Inc., 1977.

[33] Q. Guo, H. Liu, W. Chen, and I.-f. Shen, "Super resolution based on gradient field," pp. 164–168, 2008.

[34] D. Vakarelov, G. Dimov, I. Düntsch, and B. Bennett, "A proximity approach to some region-based theories of space," *Journal of applied non-classical logics*, vol. 12, no. 3-4, pp. 527–559, 2002.

[35] F. Hausdorff, *Grundzuge der Mengenlehre*. Veit & Company, Leipzig, 1914.

[36] Z. Danziger. (2013). [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/26738-hausdorff-distance

[37] F. Hausdorff, *Set theory*. American Mathematical Soc., 1962, vol. 119.

[38] W. Rucklidge, *Efficient visual recognition using the Hausdorff distance*. Springer Berlin, 1996, vol. 1173.

[39] J. M. Smirnov, "On proximity spaces," *Math. Soc. Trans. Ser*, vol. 2, no. 38, pp. 5–35, 1964.

[40] J. F. Peters, "Near sets. general theory about nearness of objects," *Applied Mathematical Sciences*, vol. 1, no. 53, pp. 2609–2029, 2007.

[41] S. A. Naimpally and J. F. Peters, *Topology with applications: topological spaces via near and far*. World Scientific, 2013.

[42] J. F. Peters, A. Skowron, and J. Stepaniuk, "Nearness of objects. approximation space model revisited." pp. 292–302, 2012.

[43] F. L. Alt, "Digital pattern recognition by moments," *Journal of the ACM (JACM)*, vol. 9, no. 2, pp. 240–258, 1962.

[44] T. Kadir and M. Brady, "Saliency, scale and image description," *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83–105, 2001.

[45] A. Rosenfeld, *Digital picture processing*. Academic press, 1976.

[46] S. Songma, W. Chimphlee, K. Maichalernnukul, and P. Sanguansat, "Classification via k-means clustering and distance-based outlier detection," pp. 125–128, 2012.

[47] A tutorial on clustering algorithms. [Online]. Available: https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html

[48] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[49] S. Basu, A. Banerjee, and R. Mooney, "Semi-supervised clustering by seeding," in *In Proceedings of 19th International Conference on Machine Learning (ICML-2002*. Citeseer, 2002.

[50] D. Pelleg, A. W. Moore *et al.*, "X-means: Extending k-means with efficient estimation of the number of clusters." in *ICML*, vol. 1, 2000, pp. 727–734.

[51] S. Ray and R. H. Turi, "Determination of number of clusters in k-means clustering and application in colour image segmentation," in *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques*. Calcutta, India, 1999, pp. 137–143.

[52] S. Krig, "Image pre-processing," in *Computer Vision Metrics*. Springer, 2016, pp. 35–74.

[53] U. Grenander, "Foundations of pattern analysis," *Quarterly of Applied Mathematics*, vol. 27, no. 1, pp. 1–55, 1969.

[54] Grenander, *General pattern theory-A mathematical study of regular structures*. Clarendon Press, 1993.

[55] T. Pavlidis, "Analysis of set patterns," *Pattern Recognition*, vol. 1, no. 2, pp. 165–178, 1968.

[56] F. W. Stentiford, "Saliency identified by absence of background structure." p. 86510V, 2013.

[57] Binary classification. [Online]. Available: https://en.wikipedia.org/wiki/Binary_classification

[58] Binary classification. [Online]. Available: https://classeval.wordpress.com/introduction/basic-evaluation-measures/

[59] W. Zhu, N. Zeng, N. Wang *et al.*, "Sensitivity, specificity, accuracy, associated confidence interval and roc analysis with practical sas implementations," *NESUG proceedings: health care and life sciences, Baltimore, Maryland*, vol. 19, 2010.