2018

Handwritten Digit Recognition and Classification Using Machine Learning

Ke Zhao
Technological University Dublin

Handwritten Digit Recognition and Classification using Machine Learning

Student Name

Ke Zhao

A dissertation submitted in partial fulfilment of the requirements of
Dublin Institute of Technology for the degree of
M.Sc. in Computing (Data Analytics)

2018

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Knowledge Management), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: **Ke Zhao**

Date: **01 Jan 2019**

ABSTRACT

In this paper, multiple learning techniques based on Optical character recognition (OCR) for the handwritten digit recognition are examined, and a new accuracy level for recognition of the MNIST dataset is reported. The proposed framework involves three primary parts, image pre-processing, feature extraction and classification. This study strives to improve the recognition accuracy by more than 99% in handwritten digit recognition. As will be seen, pre-processing and feature extraction play crucial roles in this experiment to reach the highest accuracy. Firstly, it was found that forms of image pre-processing such as normalization, slant correction or elastic distortion have a significant effect on the feature selection of the sample. In particular, slant correction is the central focus of this work because it can solve the problem whereby different people's handwriting is more or less tilted. In the feature extraction stage, Principal Component Analysis (PCA) and Histogram of Oriented Gradients (HOG) feature descriptor are presented to reduce the dimension of data and extract the relevant information. The classification task is performed by a number of classifiers namely, Support Vector Machine (SVM), Convolutional Neural Network (CNN), K-Nearest Neighbors (K-NN) and Random Forest (RF) to determine which classifier has the highest accuracy rate in this experiment. The experimental results indicated that the entire performance of CNN and K-NN models is superior to SVM and RF in the field of handwritten number recognition. Two combinations can improve the recognition accuracy to over 99% in this study, respectively Pre-processing + CNN and Pre-processing + PCA + K-NN. Moreover, four experimental results are analysed and evaluated by a series of tools such as the confusion matrix, 10-fold cross-validation, error rates, and classification reports. An interesting finding is that the level of accuracy achieved by using the HOG feature descriptor based on K-NN and RF was lower than the raw data. Notably, the combination of pre-processing and CNN reached the highest recognition rate of 99.44% in the experiment.

Key words: *OCR; Handwritten digit recognition; Slant Correction; PCA; Accuracy; Confusion Matrix*

ACKNOWLEDGEMENTS

In the process of studying for and preparing this dissertation, I was fortunate to obtain the assistance of many people, whom I would like to present my genuine thanks. My appreciation goes first to my supervisor, Dr. John Gilligan. He is an extremely knowledgeable, friendly and patient person who provided me with sufficient guidance and reminders to inspire me to stay focused and reflect deeply on the purpose of this study. His recommendations and inputs dramatically improved this project, and finally producing a dissertation of excellent quality dissertation.

I would also like to take this chance to thank all DIT module lecturers and staff members for their enthusiastic support and attention during my MSc study, especially Dr. Luca and Dr. Sarah. Dr. Sarah gave me some advice and helped me to analyze the pros and cons very patiently when I faced problems with the time conflict between job and dissertation. Studying at the DIT has been a fantastic experience and which has initiated a fresh beginning in my life.

This work would not have been possible without the encouragement and assistance of my dear family. They were always provided me with the unconditional help and valuable recommendations. I state my special gratitude to my mother in law because she came to Ireland to take care of my daughter to let me concentrate on the research. Last but not least, I would also like to thank all friends and classmates for their endless patience, generous support and total criticism which assisted me in improving my research.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 INTRODUCTION

1.1 Background

The rapid growth of new documents and multimedia news has created new challenges in pattern recognition and machine learning (Cecotti, 2016). Handwriting character recognition has become a standard research area due to advances in technologies such as the handwriting capture devices and powerful mobile computers (Elleuch, Maalej & Kherallah 2016). However, since handwriting very much depends on the writer, building a high-reliability recognition system that recognizes any handwritten character input to an application, is challenging.

This work considers the problem of recognizing handwritten digits, i.e. numbers from 0 to 9. Typically, handwritten digit recognition is an essential function in a variety of practical applications, for example in administration and finance (Niu & Suen, 2012). These industries require an excellent recognition rate with the highest reliability. Unconstrained handwritten number recognition has been applied with excellent results, to the amounts in written form on checks, to forms filled by hand such as tax forms or postal zip codes for a postcard (Lauer, Suen & Bloch, 2007). Constraint recognition refers to the extent to which individuals believe that factors beyond their control limit their behavior. By contrast, the unconstrained recognition system can be broken down into several parts: preprocessing, feature extraction, classification, evaluation and verification.

Optical character recognition (OCR) is one of a multitude of research fields in artificial intelligence and character recognition (Pramanik & Bag, 2018). OCR has developed many applications. For example, verification code images, automatic license plate recognition and text information extraction (Sarkhel *et al*., 2016). Besides, investigators working on the OCR systems have considered extensive features for handwriting digit recognition. While the majority of features are generic, several of them apply the particular attributes to enhance the function of the classifiers such as graph-theoretic methods, shadow based characteristics, gradient-based characteristics, etc. (Biswas *et al*., 2017).

Although many researchers have discussed images of isolated handwritten digits, only a few people mentioned pre-processing the image. For example, Niu and Sune (2012) proposed a hybrid model of combining the two superior classifiers: Convolutional Neural Network (CNN)

and the Support Vector Machine (SVM), which have been conducted on the non-preprocessing MNIST database and achieved the recognition rate of 94.4% with 5.6% rejection. Image pre-processing which includes filtering, segmentation, normalization, thinning, slant correction, etc. may deliver the dramatic positive effects on the characters and the results of image analysis. Most image preprocessing can reduce noise and reconstruct images so that operations can be easily performed on the image and further improve OCR accuracy. Moreover, different people's writings are more or less sloped. To correct this, elastic distortion is employed in the process of rotating an image that provides a method to increase the similarity between two samples representing the same digit.

In the domain of OCR handwriting digit recognition has been intensively researched for ten years in many systems and classification algorithms. These include, for example, the SVM, CNN and Random Forest (RF) algorithms. However, the recognition accuracy of the experiments is mostly around 95%. Since lots of classifiers cannot adequately handle the original images or data, feature extraction is one of the pretreatment steps that has the purpose of decreasing the dimension of data and abstracting the valid information (Lauer, Suen & Bloch, 2007).

Traditional manual design feature selection is a cumbersome and time-consuming mission that cannot process the original image, while an automatic extraction method by CNN can detect features directly from the original image (Bernard, Adam & Heutte, 2007). Lauer, Suen, and Bloch (2007) replaced the last layer of the LeNet4 network with the K-Nearest Neighbors (K-NN) classifier to process the abstracted features. A CNN is a feed-forward network that extracts topological attributes from images. It collects features from the original image in the first layer and uses its last layer to classify the pattern. At the classification stage, the SVM constructs the best separation hyperplane in the high dimensional characteristic space. Also, the k-NN algorithm is one of the most straightforward machine learning algorithms, and the input consists of the k nearest training samples in the feature space. RF build various decision trees and associate them to receive more accurate and stable predictions.

The Histogram of Oriented Gradient (HOG) is accessible for object detection that feature extraction needs to invert black/white pixels. In the HOG feature descriptor, the distribution of directions of gradients is used as a features. The gradient image removes a lot of non-essential information but highlights the outline. Moreover, Principal component analysis (PCA)

is an eigenvector-based multivariate analysis technique that usually extracts the best data variance. The other main benefit of PCA is that once the patterns are detected in the data, then the data is decreased without much loss of information.

1.2 Research Project

This research aims to recognize the handwritten digits by using tools from Machine Learning to train the classifiers, so it produces a high recognition performance. Furthermore, the use of tools from Computer Vision is explored to investigate the effect of the selection of classifiers, features, and image preprocessing on the entire error rate. The dataset used for the application is a MNIST dataset containing 60,000 training and 10,000 testing images originally, which are 28 x 28 grayscale (0 - 255), labelled and bitmap format. It is an excellent database for machine learning and pattern recognition methods while needing minimal efforts in preprocessing and formatting.

There are many features in this data, so it has many dimensions. PCA is a dimension-reduction tool that is applied to reduce the elements into a small but informative kind of set of characteristics before using the data in the machine learning models.

The research question addresses the following: *Can OCR use the combination of image pre-processing and classifiers to improve the accuracy of handwritten digit recognition to more than 99%?*

The OCR technique transforms the input graphics into a flexible format in the computer (Phangtriastu, Harefa & Tanoto, 2017). In OCR applications, the performance accuracy and speed of digital recognition is critical to the overall performance. In a handwriting recognition system, feature extraction is one of the vital factors for success. A good group of features ought to represent traits that are specific to one class (Lauer, Suen & Bloch, 2007). The commonly applied functions in character recognition are crossing points, structures, directions, intersections and contours (Niu & Suen, 2012). However, many classifiers such as SVM and RF cannot process raw images or data efficiently, because extracting appropriate structural features from complex shapes is a considerable challenge (Pramanik & Bag, 2018). While, the automatic extraction method by CNN can extract elements directly from the raw image (Bernard, Adam & Heutte, 2007), as well as the HOG feature vector is also very useful for

tasks like image recognition and object detection, as when it fesds into the classification algorithms like SVM or RF it produces good results. Besides, PCA can project digital images onto a low-dimensional interspace composed of few primary images for further feature extraction.

Some problems occur during the development of the OCR system. Firstly, raw image data may have a variety of issues such as blurring or skewing, hence it may not generate the most excellent computer vision results. That is why image pre-processing is considered in depth. Another problem is how to extract features with background noise. One clear example is the contrast between fonts and paper (Phangtriastu, Harefa & Tanoto, 2017). Furthermore, the performance of the classifier can depend on the feature quality of the classifier itself (Elleuch, Maalej & Kherallah 2016). Additionally, a common problem in the digital classification is considered. The similarity between numbers such as 1 and 7, 5 and 6, 3 and 8, 9, and 8 etc., makes recognition a difficult task. Because people write the same number in various ways, the uniqueness and variety in handwriting affect the structure and appearance of the digits. Therefore, how to use the combination of image pre-processing and classifier is the main problem of OCR in handwritten digit recognition.

In this digital epoch, many handwritten forms are still sent out through post. There is still a high number of people who can not access a personal computer or the internet, and in some cases where email addresses are not precise or don't even existed. Therefore, accurate handwritten character recognition is still a problem for a lot of businesses. The most distinct problem when identifying handwritten forms in the data capture procedure is poor quality or illegible handwriting. One clear example is where Educational Summit in 2012 found that 25-35% of pupils at a secondary school have not obtained competency in handwriting skills. That means the forms filled out by hand could produce an on-going challenge to the data gathering procedure.

1.3 Research Objectives

The objectives of this research are:

- To find out what opinions of the various image preprocessing techniques can be applied to this study;

- To identify whether can these image preprocessing methods have a significant impact on the error rate of selected classification models;
- To examine the potential of the HOG feature descriptor to predict the model in image recognition and object detection;
- To recommend which algorithms can improve the accuracy of handwritten digital recognition to up to 99% based on the evaluated findings;
- Multiple handwritten digital images are applied to the model with the highest performance for verification.

1.4 Research Methodologies

Hypothesis: The null hypothesis (H0) of this research is that the accuracy of handwritten digit recognition using the combination of image pre-processing and classifiers based on the OCR will be less than 95%, while the alternative hypothesis (H1) is that accuracy will be no less than 99%.

The objective of the research is to show the particularly template-based model, such as CNN and RF. However, the error rate increases when the number centered on the bounding box rather than the center of mass. So, feature extraction is one of the pretreatment steps, aimed at reducing the dimension of data and extracting the relevant information. On the other hand, image preprocessing such as sharpening, slant correction or elastic distortion is necessary because the oblique numbers and blurred images will affect the accuracy of feature extraction. Traditional manual design feature selection is a cumbersome and time-consuming mission that cannot process the original image, while an automatic extraction method by the LeNet5 CNN architecture can retrieve features directly from the original image and HOG is another feature descriptor.

The research methods used in this paper are quantitative. Specifically, quantitative methods emphasize objective measurements and manipulate pre-existing statistical data using software tools. The MNIST database files are online freely available train.csv of 60,000 examples and test.csv of 10,000 samples that contain images of handwritten English numerals. Also, the RF, CNN, K-NN and SVM in Python were used to study and build prediction models. The four preprocessed models will be evaluated and compared, and then the results will clearly show

the difference in performance for the classifiers. Moreover, the accuracy of the classifiers will be assessed to determine whether to reject or accept the null hypothesis.

1.5 Scope and Limitations

The scope of this paper is handwritten digit recognition regarding the application of machine learning algorithms based on image pre-processing and feature extraction. Additionally, the purposes are not only to improve the current recognition performance, but also to seek the highest reliability in the applications of handwritten digits.

This thesis has the following limitations:

- A handwritten digit dataset is vague in essence because there may not always be perfectly straight lines, and different people's writings are more or less sloped;
- The curves are not necessarily smooth like the printed characters;
- The recognition system sometimes shows inconsistent results due to the similarly shaped numerals;
- All handwritten digital images that are final tested do not automatically detect boundaries and cropping as well;
- The time assigned to this paper was five months. Due to the limited amount of time, the proposed model was not further optimized.

1.6 Document Outline

The rest of the paper consists of five chapters that are organized as follows:

Chapter Two: **Literature Review and related work:** The second chapter presents an in-depth overview of the domain of handwritten character recognition. It outlines the characteristics and the results of the different methods in machine learning and offers opinions about possible conclusions.

Chapter Three: **Design and methodology:** The third chapter delimits the research methodology applied involving the research design, data selection, the popular models, and research philosophy. It establishes a process of testing different methods of recognizing handwritten characters. It establishes a role for preprocessing images in these methods.

Chapter Four: **Implementation and results:** The fourth chapter exposes the actual work of the system implementation and experiment results. It will examine the impact of image pre-processing and different kinds of classifiers on recognition accuracy.

Chapter Five: **Analysis, evaluation and discussion:** The fifth chapter provides an analysis of the experimental results, the model evaluation and test, and the discussion in line with the literature review.

Chapter Six. **Conclusion:** The last chapter will present a short account of the work results, including the problems which were addressed, and the limitations of the study. This section will also outline suggestions for future research.

2 LITERATURE REVIEW AND RELATED WORK

2.1 Introduction

This chapter will introduce how the image preprocessing, feature selection and the relevant classification techniques contribute to handwritten digit recognition. Also, it provides an in-depth and detailed overview of the recent literature, corresponding to this study. Firstly, the part of this section presents an overview with references to the approaches to OCR and the template matching Machine Learning (ML) techniques. In the second part, an analysis of the factors which affect the recognition error rate is expressed. Furthermore, the applied classification techniques in ML and the evaluation of design will be reviewed in the third part. The final section will provide a summary of the next stage in the study and what will do next and what will come in the design of the experiment in.

2.2 The Importance of Handwritten Digit Recognition

More and more people are focusing on the use of the personal computer rather than acquiring excellent handwriting skills. The one reason is that the internet and applications are becoming more intelligent than before. Additionally, the poor quality or illegible handwriting is the main reason for inaccurate handwritten character recognition.

OCR refers to the recognition of characters on optical scanning and digital text pages by computer (Winkler, 1980). Although many systems are available for identifying printed text, identifying handwritten characters is still a challenge in the field of pattern recognition (Sarkhel *et al*., 2016). Despite its problems, it widely contributes to the progress of improving the interface between man and machine in a lot of applications (Sarkhel *et al*., 2017). Due to a variety of potential applications such as the reading of postal codes, medical prescription reading, interpreting handwritten addresses, processing bank checks, credit authentication, social welfare, forensic analysis of crime evidence which includes a handwritten note, etc., handwritten digital recognition is still an active area of research (Winkler, 1980). In recent years, the availability of devices has further broadened the range of applications for handwritten digital recognition for multiple personal uses such as note taking and extracting data from filling out forms, etc. (Das *et al.*, 2015).

The handwritten analysis is a cumbersome and organized process that relies on a broad knowledge of the way people form digits or letters, and which exploits the unique

characteristics of numerals and letters, for example, the shapes, sizes, and individual writing styles that people use (Winkler, 1980). Even personal writing styles may vary with the writing tools and environment and leave clues about the identity of the author. In the field of forensic analysis which includes crime scene investigation, DNA testing, fiber analysis, fingerprint analysis, to name but a few disciplines, the study of handwriting plays an important role. Questioned document examiners (QDEs) analyze files for signs of changes or forgery, and written comparison to identify or exclude authorship.

Typically, handwriting experts use sophisticated classification models to analyze printed or handwritten character images. As part of this process, they extract features from the samples which include slants, orientation and the center alignment of the letters. Offline digital recognition has many practical applications. For instance, the handwritten sample is analyzed and recognized by the handwriting expert to identify the zip code in an address written or printed on an envelope (Hanmandlu & Murthy, 2007). As a result, the benefits of applying this system at the post office are enormous. The system can realize the automatic sorting of millions of emails, thus reducing the human burden and speeding up the whole process (Mane & Kulkarni, 2018).

2.3 OCR

2.3.1 Approaches to OCR

OCR is a technique that recognizes printed text in scanned documents. But it serves many other purposes as well. For instance, the Google Translate application contains an OCR technique that works with the device's camera. It captures text from magazines, documents, and other handwritten characters and converts it to another language. OCR is a complex process which involves many steps. The steps involved in OCR are preprocessing, feature selection and classification. First capture the image of the digit is categorized in a standard image format such as JPEG, PNG or bitmap. Image formats are broadly categorized as lossy or non-lossy image formats which are used depending on the application. For example, it is usually a requirement that non-lossy image formats are used in medicine. Next, the image is preprocessed to standardize features such as size and resolution. From this, we extract features such as an edge outline or a chain code depending on the algorithm being used. Finally, these features will be passed on to the classification engine.

Pre-processing options consist of normalizing the size and aspect ratio of the image, elastic distortions and interpolation techniques for pixel values, and so on. The purpose of preprocessing is to remove noise, smooth and normalize the input data, which is essential for better differentiation of patterns in the feature space (Karimi et al., 2015). Liu et al. (2004) researched a method which is a combination of normalization, feature selection and classification to produce a very high accuracy on famous datasets. Additionally, Simard, Steinkraus and Platt (2003) extended the training set by increasing new forms of elastic distortion data to receive good results for CNN.

For feature extraction, there are multiple feature types, and extraction techniques can be used. The most common is the distribution of directional features because of their high performance and ease of implementation. Directional elements can be measured from the skeleton, chain code or gradient. Among them, the chain code features are extensively adapted, while gradient features are suitable for grayscale and binary images (Liu et al., 2004). Winkler (1980) applied a new feature combination, including of PCA/Modular PCA (MPCA) based statistical features and quad-tree found Longest-Run (QTLR) features for OCR. In general, another feature extraction approaches are mentioned to increase the recognition rate and shorten the time of recognition mode (Karimi *et al*., 2015), namely the scale invariant feature transform (SIFT) descriptor (Lowe, 2004) and HOG (Dalal & Triggs, 2005). Pramanik and Bag (2018) researched a method which is chain code histogram feature set with a multi-layered perceptron (MLP) based classifier provided good recognition accuracy compared with other methods (Pramanik & Bag, 2018).

In developing OCR systems, most classifiers can be used for classification: parametric and nonparametric statistical classifiers, K-NN, SVM, Neural Network (NN), CNN, RF and hybrid classifiers, etc. A multi-column multi-scale convolutional neural network (MMCNN) based structure has been adopted for faster recognition (Sarkhel *et al*., 2017). Also, Breiman (2001) introduced another class of methods called RF and provided parameter setting rules. One of the research study discussed identifying the mathematical symbols in the figure applying the SVM (Phangtriastu, Harefa & Tanoto, 2017). In 2012, Niu and Suen designed a hybrid CNN–SVM model for OCR which was based on the automatic retrieval feature of CNN architecture, where the unknown pattern is identified by SVM recognizer.

2.3.2 An Overview of Handwritten OCR

Handwritten OCR applications have been around for ten years, and are still one of the most area of research especially since recent technology, for example on mobile phones, makes it relatively easy to scan the text and capture the image of the digit in a standard image format. Early research involved letter recognition, and mostly it was based on template matching and achieved around 95% accuracy. In recent years, greater emphasis has been placed on the machine-learning technique.

Investigators working on the OCR systems have considered extensive features for handwriting digit recognition. Some features use script-specific attributes to improve the function of the underlying classifier — for instance, features based on syntax or formal grammar, graph theory methods, shadow-based characteristics and gradient-based characteristics (Sarkhel *et al*., 2017). In OCR applications, the accuracy and speed of digital recognition are critical to overall performance.

Normalization is regarded as the essential pre-processing factor for handwritten OCR (Liu *et al*., 2004). Moreover, the standardization of character images is provided with impact on the recognition performance and has advanced an aspect ratio adaptive normalization(ARAN) strategy to enhance the property. Elastic deformation is achieved by calculating a new target location relative to the original position for each pixel; some clear examples are translation, rotation, skewing, etc. (Simard, Steinkraus & Platt, 2003).

In a handwriting recognition system, feature extraction is one of the critical factors for success and has a significant impact on the classification. A good group of features should represent characteristics that are specific to one class. The commonly used functions in character recognition are partitions, structures, directions, intersections and contours (Niu & Suen, 2012). Fuzzy model-based recognition of a handwritten digit has been discussed by Hanmandlu and Murthy (2007). They proposed that each feature produces a fuzzy set when collected for all training samples.

Recently, a new method was proposed by Elleuch, Maalej and Kherallah (2016). They applied a CNN and SVM approach as an automatic feature extractor from raw images, and it let SVM classify by analyzing the error classification rate in the handwritten digit recognition. Drop-out

training is an effective method to control over-fitting by randomly omitting feature subsets in each iteration of the training process. Das *et al*. (2015) examined the innovative analysis of handwritten Bangla character recognition applying a soft computing paradigm embedded in the two-pass method. On experimentation, the proposed method showed a significant improvement of recognition rates compared to with the Single-pass approach. In 2018, Wu, Wei and Zhang compared the performance of several classifier algorithms on the MNIST database of handwritten digits. They found that that boosting gives a substantial improvement in accuracy, with a relatively small penalty in memory and computing expense.

2.4 Preprocessing and Feature Extraction

Since there is no standard large dataset available for handwritten Marathi numerals, Mane and Kulkarni (2018) have performed various transformations to add the size of the data set. For instance, scaling: the stochastic quantity mounted each image and shifted to a new random location; Horizontal and Vertical skewing: each image is tilted vertically and horizontally by a factor of 0.5, which is represented in Figure 2.1. These conversions have increased the dataset fourfold.

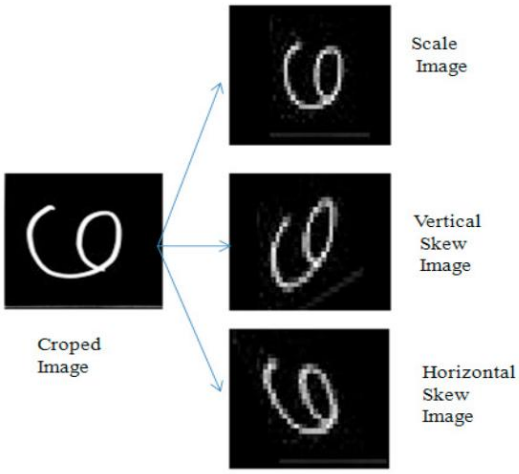


Figure 2.1: An example of the transformed images

A 2007 paper by Hanmandlu and Murthy proposed the distinct preprocessing techniques specifically slant correction, thinning and smoothing. In other words, the task of recognizing handwritten digits has been broken down into the following steps which are depicted in Fig.2.2. For handwritten characters, one of the first variances in ways of writing ways is caused by slope, which is defined as the slant of the writing trend relative to the vertical line. Besides that, slant correction must precede other pre-processing tasks, which is that correction operations

usually create a rough outline of the character and smoothing tends to change the image topology. Bilinear interpolation is useful for generating other distorted character images at the selected resolution (29x29). Furthermore, one method of achieving non-uniform thickness invariance includes determining a constant thickness "middle line" for each letter for identification purposes. Therefore, the process is called *Skeletonization*. Also, a new method was devised by Hanmandlu and Murthy (2007) to smoothing and removing the virtual slant of distorted numbers.

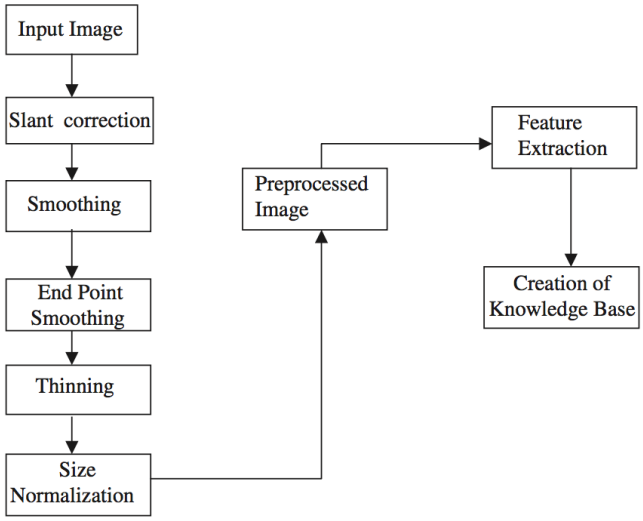


Figure 2.2: Block diagram describing system implementation

Sadri, Suen and Bui (2007) have located characteristic points on the string image according to the developed algorithm and produced possible segmentation hypotheses, as well as finding the group with the highest segmentation recognition reliability. Another group of researchers, Simard, Steinkraus and Platt (2003), mentioned that if the data is scarce and the distribution to be studied has transform-invariant attributes, applying transformations can generate additional data and even improve performance. In the case of handwriting recognition, elastic deformation which can be achieved by calculating a new target position relative to the original post for each pixel, is one of the techniques that can be used to extend the training set. In 2007, Lauer, Suen and Blochhave increased the MINST dataset size by four times by using affine transformations and the prior knowledge of transform invariant properties. In the raised method, the elastic distortions are applied to each sample of the training set to extend nine new samples. Keysers *et al.* (2007) further suggested that a local deformation technique defined that more complex models (e.g., 2-dimensional warping) do not certainly represent better models than simple image distortion models.

Some problems occur during the development of the OCR system. First, a computer can be complicated when identifying values that have similarities to other benefits. Another problem is when extracting features with background noise, such as the contrast of fonts and paper (Phangtriastu, Harefa & Tanoto, 2017). Liu and Suen (2009) applied the appropriate threshold and gray standard normalization technologies to standardize the grayscale levels of the background and foreground regions of the target image. In other words, a binary image or a background-eliminated gray-scale image can be obtained by thresholding the grayscale image.

Regarding the fact that the images on the database are not uniform in dimensional sizes, it is clear that the first step is to standardize them (Karimi *et al.*, 2015). The optimal target size for normalization is the all average resolution because of excessive size changes may result in data loss. Therefore, Karimi *et al*. (2015) have adjusted the TMU database to 40×40 pixel images in Matlab. A 2004 paper by Liu *et al.* concentrated on the diversity of performance for ARAN and orientation feature extraction. The property of ARAN is based on the aspect ratio mapping function. For this study, three databases namely CENPARMI, NIST and Hitachi have been selected. Notably, the researchers provided ten normalization functions including seven based dimensions; three found moments and eight feature vectors for comparison to discover the excellent choices (Karimi *et al.*, 2015).

The local gradient feature descriptors extract the feature vectors in the handwritten image and then submitted them to a machine learning algorithm for original classification (Surinta *et al.*, 2015). Recently, a new method was proposed by Phangtriastu, Harefa and Tanoto (2017) combining several feature extraction algorithms which are the projection histogram, HOG and zoning algorithm. In the HOG feature descriptor, the distribution of directions of gradients is used as features. Zoning is a way for partial information analysis on partitions of a specific pattern. Moreover, the projection profile that includes two types namely the vertical and horizontal is one of the feature extractions that cumulate black pixels along the rows and columns in the image. PCA is an eigenvector-based multivariate analysis technique that usually extracts the best data variance (Winkler, 1980). To facilitate calculations, the PCA reduces the dimension of the MNIST dataset from 784 to a lower value.

2.5 Classification Techniques in ML

Digital recognition is one of the most indispensable applications in pattern recognition. Many researchers have studied and identified different datasets. For example, the US Postal Code on the letter was collected into the CEDAR digital database and used as a standard database for researchers to analyze (Sarkhel *et al.,* 2016). Since there is no criterion database available at the moment for Marathi, a dataset of 2000 images containing Marathi numerals from 0-9, has been collected from different age groups (Mane& Kulkarni, 2018). Furthermore, the SD19 database provided by the National Institute of Standards and Technology (NIST) was also viral among researchers (Hochuli *et al.,* 2018). The CMATERdb 3.1.3.3 is a database including 171 unique categories of isolated grayscale images of Bangla compound characters. However, images in the dataset are neither of central nor uniform proportions, resulting in difficult pattern recognition problems (Roy *et al*., 2017). Table 2.1 gives a comparison of the classification techniques recognition rates (RR) for the handwritten characters or digits from the different databases.

Work reference	Techniques	Database	RR
Phangtriastu *et al*., 2017	SVM, ANN	Chars74K	94.43%
Mane & Kulkarni, 2018	CCNN	Self created	94.93%
Sadri *et al*., 2007	NN, SVM	NIST SD19	96.42%
Hochuli *et al*., 2018	CNN	NIST SD19	97%
Mahto *et al*., 2015	SVM, KNN	Self created	98.06%
Hanmandlu & Murthy, 2007	ID-3	CEDAR	98.4%
Roy *et al*., 2017	DCNN	CEDAR	90.33%
Bernard *et al*., 2007	RF	MNIST	93%
Cecotti, 2016	K-NN	MNIST	98.54%
Karimi *et al*., 2015	Bagging, Boosting	TMU	98.06%

Table 2.1: Comparative description of the classification techniques RR for the handwritten characters from the different databases.

The performance of a classifier can depend on the quality of the features of the classifier itself (Elleuch, Maalej & Kherallah 2016). However, many classifiers such as SVM and RF cannot

process raw images or data efficiently, because extracting appropriate structural features from complex shapes is a considerable challenge (Pramanik & Bag, 2018). Therefore, how to use the combination of sophisticated features extraction and classifier is the main problem of OCR in handwritten digit recognition. A 2017 paper by Phangtriastu, Harefa, and Tanoto compared the most commonly used classifiers SVM and ANN, while this experiment achieved the highest accuracy of 94.43% using the SVM classifier with the combination of feature extraction algorithms which are projection histogram and HOG.

In order to improve the high reliability in handwritten digit recognition, a new feature combination, including of MPCA based statistical features and QTLR features has been evaluated on handwritten digits of five prevalent scripts of Indian, viz., Arabic, Bangla, Devanagari, Latin, and Telugu with SVM based on OCR (Winkler, 1980). Winkler observed that only the features extracted by the PCA algorithm are not sufficient to solve the variability of the handwritten digit mode, while the QTLR-based topology features also have limitations in classifying digital patterns into individual scripts. Consequently, the combination of MPCA + QTLR is applied to increase the recognition accuracies significantly to 98.7%.

Many researches have been carried out on feature extraction and classifier algorithms for handwritten digit recognition. Most of them got good recognition accuracy. For instance, Mane and Kulkarni (2018) proposed a Customized Convolutional Neural Network (CCNN) that can automatically learn features and predict the categories of numerals in extensive ranged data set such as Marathi which is one of the most diffusely spoken regional languages in India. Besides, the CCNN's performance reached an average of 94.93% accuracy by using K-fold cross-validation. The proposed CCNN model does not impose any restrictions on the count of layers, but instead optimizes the number to satisfy the demand of the issue. Also, the different filter sizes have been applied for the intermediate convolutional layer.

In 2007, Sadri, Suen and Bui indicated that the correct use of context knowledge in segmentation, evaluation, and the search could remarkably improve the overall performance of the handwritten digit recognition system. As a result, the recognition system was able to get 95.28% and 96.42% recognition accuracy on handwritten numeric strings using NN and SVM classifiers, respectively. Hochuli *et al*. (2018) stated the CNN classifier could handle the complexities of touch numbers more efficiently than all the segmentation algorithms provided in the literature. The experiments on two famous databases consisting of Touching Pairs

Dataset and NIST SD19 highlight the proposed method by achieving an accuracy level of 97% recognition accuracy.

Recently, Mahto, Bahtia and Sharma (2015) designed a combination of horizontal and vertical projection feature extraction to identify Gurmukhi's handwritten characters. The experiment applied linear SVM and k-NN (k = 1, k = 3, k = 5, k = 7) to classify handwritten characters to a maximum accuracy of 98.06%. However, in some other reports, Lauer, Suen, and Bloch (2007) replaced the last layer of the LeNet4 network with a K-NN classifier to process the extracted features. But compared with the regular LeNet4 network, this approach does not improve the results.

A 2007 paper by Hanmandlu and Murthy proposed the recognition of handwritten Hindi and English digits in the form of exponential membership functions as fuzzy models. Furthermore, the defuzzification parameters have been optimized by the double layer perceptron, and the fuzzy rules have been generated based on the ID-3 method. This technique overcame the difficulties of traditional handwritten character recognition syntactic methods and achieved the accuracy of 95% for Hindi digits and 98.4% for English digits.

In the field of pattern recognition, researchers have paid more attention to multi-classifier systems in recent years, especially Bagging, Boosting. Bernard, Adam and Heutte (2007) researched a conventional feature extraction technique based on a greyscale multi-resolution pyramid to find out the effect of the parameter values on the performance of the RF. They have experimented with the Forest-RI algorithm, which is considered as the Random Forest reference method, on the MNIST handwritten digital database and reached a level of accuracy in handwritten digit recognition to greater than 93%.

A 2017 paper by Roy *et al.* studied the innovative analysis of handwritten Bangla that isolated compound character recognition using a novel deep learning technique. The researchers performed layered training on deep convolutional neural networks (DCNN) and used the RMSProp algorithm to enhance the training process to achieve faster convergence. On experimentation, the proposed DCNN showed significant improvement in recognition rates compared with the standard shallow learning model, reaching 90.33%.

In 2018, Shamim *et al.* completed the comparative analysis of the performance of SVM with polynomial kernel and radial basis function kernel (RBF) kernel for classifying students with or without handwriting difficulties. Also, cross-validation which is a statistical method for assessing and comparing ML algorithms has been widely used for evaluating the performance of NN and other applications such as SVM and K-NN. Shamim *et al.* (2018) adopted the ten cross-validation method to select the parameter to gain the highest recognition rates. While, this experiment illustrated that the performance of SVM with RBF is better than with polynomial kernel, reaches more than 93%.

Cecotti (2016) presented a novel active machine learning strategy for the classification of handwritten numerals. Dynamic learning methods solve the problem that large databases are not always immediately available by querying experts to set labels for specific instances. In this paper, Cecotti evaluated the performance of this method on four databases corresponding to distinct scripts (Latin, Bangla, Devanagari and Oriya) and received an accuracy of 98.54% on the MNIST training database.

Recently, Karimi *et al*. (2015) proposed a method for identifying Persian handwritten digits consisting of three main sections, preprocessing, feature extraction and classification. In the feature extraction phase, a set of appropriate and complementary features involves in 115 features abstracted from Persian handwritten digits. In the classification phase, the ensemble classifier algorithms such as Boosting and Bagging are used to separate the classes of samples from each other. Moreover, this experiment has been estimated on the Tarbiat Modares University (TMU) digital database, and the result, it was claimed give the highest recognition accuracy of 98.06% for Persian handwritten digits.

2.6 Summary

This chapter has reviewed the existing literature relevant to the research. Notably, it highlighted many operational techniques that have to be considered, namely image preprocessing, feature selection and pertinent classifiers of machine learning. These factors should be addressed in the acquisition of data resources and the preparation of the plan to achieve the highest handwritten recognition accuracy.

Firstly, image preprocessing such as normalization, slant correction or elastic distortion is necessary because the slanting numbers and blurred images will affect the accuracy of feature extraction. Besides that, if the data is scarce and the distribution to be studied has transform-invariant attributes, applying transformations can generate additional data and even improve performance, for example, scaling, horizontal and vertical skewing and the elastic distortions, etc. These conversions have expanded the dataset size by four to nine times.

Followed by the feature extraction step, aiming to reduce the dimension of data while extracting relevant information. In a handwriting recognition system, feature extraction is one of the critical factors for success and has a significant impact on the classification. In the HOG feature descriptor, the distribution of directions of gradients is used as features. Zoning is a means for partial information analysis on partitions of a specific pattern. Besides, the automatic extraction method by CNN can extract elements directly from the raw image. PCA can project digital images onto low-dimensional space composed of a small number of elemental pictures for further feature extraction.

The research will provide the summary of requirements which are an in-depth introduction to the principles and design of template-based models such as CNN, SVM, K-NN, and RF in the next chapter. To achieve higher accuracy, the sharpening, normalization, slant correction, and elastic distortion techniques will be more detailed research and discussion in handwritten digit recognition. To facilitate calculations, the PCA will be used to minimize the proportion of the MNIST dataset from 784 to a lower value. Additionally, HOG is a gradient feature descriptor that extracts the feature vectors in the handwritten image. The HOG feature vector is very useful for tasks like image recognition and object detection when it is fed into the classification algorithms like SVM or RF producing good results.

3 DESIGN AND METHODOLOGY

3.1 Introduction

The research question is the following: Can OCR use the combination of image pre-processing and classifiers to improve the accuracy of handwritten digit recognition to more than 99%? Here, an approach will be presented to address this question.

This chapter will outline and discuss the structure of the research including the data collection methods, sampling size, the principles and design of template-based models such as CNN, SVM, K-NN, and RF, the comparison of the technical details and the methodology adopted for designing and evaluating the solution. In particular, to achieve higher accuracy, the sharpening, grayscale normalization, slant correction and elastic distortion techniques will be applied to the image preprocessing stage. The PCA will be used to extract the best data variance, and the HOG feature vector will be applied to image recognition and object detection. Finally, the four previously mentioned classification techniques will be evaluated using K-fold cross-validation, error rates, accuracy, classification reports and confusion matrix.

3.2 Summary of Requirements

This study strives to improve the recognition accuracy to more than 99% in handwritten digit recognition. Because some numbers in the script are written in different people's handwriting styles, the machine may encounter difficulty identifying them. Fast and precise handwritten digit recognition is the most important aspect of finance and administration. The handwriting pattern recognition system consists of three main parts, namely preprocessing, feature extraction and classification. The MNIST dataset is an excellent database for machine learning and pattern recognition methods while involving minimal efforts in preprocessing and formatting. That is why it was selected for this study.

As mentioned in the literature review mentioned in the last chapter, some researchers have obtained some achievements in handwritten character recognition. For example, Hochuli *et al*. (2018) used the CNN classifier to perform experiments on two public databases consisting of Touching Pairs Dataset and NIST SD19, as well as highlighting the proposed method by achieving a 97% recognition accuracy. Recently, Mahto, Bahtia, and Sharma (2015) applied a linear K-NN to classify Gurmukhi handwritten characters with a maximum accuracy of 98.06%. Moreover, Bernard, Adam, and Heutte (2007) experimented with the Forest-RI

algorithm on the MNIST handwritten digital database, and the accuracy of handwritten digit recognition reached over 93%. Also, a 2017 paper by Phangtriastu, Harefa, and Tanoto achieved the highest accuracy, namely 94.43% using the SVM classifier. Overall, researchers have used these classifiers to get good results.

Since previous researchers have well verified the four classifiers mentioned above, CNN, SVM, K-NN and RF will be applied and compared in this experiment to determine which classifier delivers the highest performance. However, this research will pay more attention to preprocessing and feature extraction steps than in previous studies to reach the highest accuracy.

Firstly, raw image data may have a variety of issues such as blurring or skewing and thus are less likely to produce optimal computer vision results. That is why careful consideration of image preprocessing is fundamental. In particular, grayscale normalization will decrease the effect of an illumination's differences. Slant correction will be used to solve the problem that different people's handwritings are more or less skewed writing. Furthermore, only a handful of researchers have mentioned the technique of sharpening the blurry image in handwritten digit recognition. There are three primary reasons to sharpen the image, which are to overcome the blur introduced by the camera device, improve the legibility and contribute to the feature extraction in the next step. Also, some researchers have proposed the Elastic distortion technique to increase the training set. Therefore, this experiment will focus on the methods of normalization, slant correction, sharpening and elastic distortion in the preprocessing stage.

In handwriting recognition systems, feature extraction is one of the critical factors for success. However, extracting appropriate structural features from complex shapes is a considerable challenge. CNN will use the LeNet5 automatic extraction method to extract elements directly from the original image. Moreover, the HOG feature vector will be adopted in other classifiers, since it is useful for tasks such as image recognition and object detection. Besides, PCA can project digital images onto low-dimensional space composed of a small number of elemental images for further feature extraction. Therefore, HOG and PCA are the nuclear technologies in the feature extraction phase.

In the evaluation phase, when classifying data, k-fold cross-validation is applied to estimate the skill of the machine learning model for unseen data. Meanwhile, the accuracy of the

verification is observed for each epoch to ensure the correct training. After the model is optimized, it is then tested with unknown samples to find the test accuracy and represent it with a confusion matrix. Besides, how the loss changes are maximized and the differences between training accuracy and verification accuracy will be shown on the graph. Also, the error rates of the four models will be compared and analyzed based on the different techniques of pre-processing and feature extraction.

Finally, the handwritten numbers never seen by the systems will be applied to finalize the model. The resulting model with an accuracy of more than 99% will identify the handwritten digits and display the predicted numbers. Figure 3.1 presents some handwritten digits never seen by the systems.

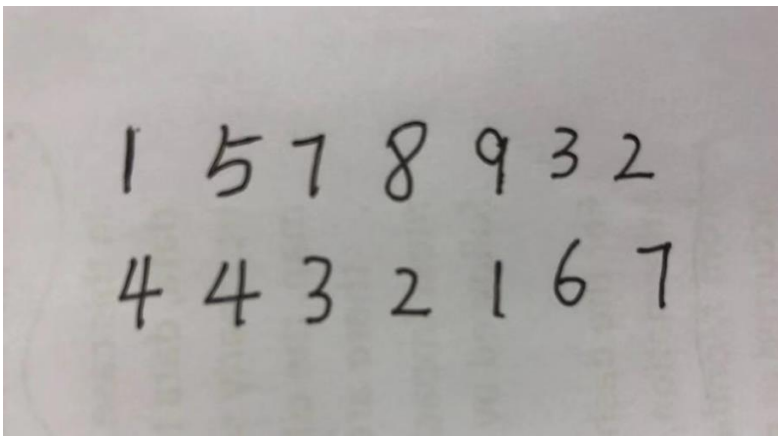


Figure 3.1: The several of handwritten digits never seen by the systems

3.3 Data Collection

A commonly applied dataset for handwritten digit recognition called MNIST can be searched on the Y. Lecun website. It is a gathering of 70,000 digits written by the different 750 Census Bureau employees and high school students. This dataset is a widely known benchmark that includes a training set with 60,000 images and testing set with 10,000 images. Numerals were size-normalized, centered and stored sequentially as 28 × 28 pixel images in gray-level bitmaps. The resulting datasets are provided with the labels, and each image includes a single digit. This ready-to-use database is the data was applied in the experiments below. Figure 3.2 displays some sample numbers in the training set.

Figure 3.2: MNIST database: the sample numbers in the training set

The MNIST dataset consists of NIST's unique database 3 and unique database 1, which involves binary images of handwritten numbers. In particular, the MNIST training set that contained samples from approximately 250 writers is constructed from 30,000 patterns in SD-3 and 30,000 patterns in SD-1. Similarly, the test set consists of 5,000 patterns in SD-3 and 5,000 patterns in SD-1. NIST formerly appointed SD-3 as the training set and SD-1 as the test set. However, compared with SD-1, SD-3 is more transparent and more accessible to identification. The reason for this is that SD-3 was collected from Census Bureau staff, while SD-1 was collected from high school students. Since it is essential to ensure that the writers of the training set and the test set are disjoint, a new database MNIST was built by mixing the NIST data sets.

Some researchers have used the database for analysis and have achieved beautiful results. For instance, Bernard and his team (2007) reached an average of 94.93% for handwritten digit recognition accuracy, and Cecotti (2016) gained the highest accuracy of 98.54%. In some experiments, to add the training set, the artificial distortions are applied to each sample to extend new samples.

In the field of handwritten character recognition, there are also other researchers who have used the different script datasets for research and most of them have achieved satisfactory results. One clear example is that Surinta *et al*. (2015) who collected a new Thai handwritten script database from 150 natives who were aged from 20 to 23 years and studied in the university. The Thai handwritten numeric dataset (THI-D10) has 9555 samples consisting of 8055 training

samples and 1500 test samples. Surintaet al. (2015) reached a better result of 97.87% on THI-D10 than the results reported in previous studies.

Another example is a dataset of 2000 images containing Marathi numerals from 0-9 which were collected from different age groups since there is no criterion database available at the moment for Marathi (Mane& Kulkarni, 2018). Mane and Kulkarni used CCNN and verified by K- fold cross validation and obtained an average 94.93% recognition accuracy for the testing dataset.

The Bangla language which includes 11 vowels, 39 consonants, 10 modifiers, and 334 compound characters is the sixth most universal language in the world. The CMATERdb 3.1.3.3 is a database including 171 unique categories of isolated grayscale images of Bangla compound characters that have been applied to handwritten Bangla composite character recognition. However, the images in the dataset are neither central nor uniform, resulting in problems with in pattern recognition problems. In contrast, the MNIST dataset is composed of only 10 categories and is less challenging. Roy *et al.* (2017) applied Deep Convolutional Neural Network (DCNN) on the CMATERdb 3.1.3.3 and achieved a recognition accuracy of 90.33%, representing an excellent result in handwritten Bangla compound character recognition.

3.4 CNN

A Convolutional Neural Network (CNN) is a multi-layer neural feed-forward network with deep supervised learning architecture, which can be regarded as a two-part combination: automatic feature extractor and trainable classifier. The classifier and weights of the back-propagation algorithm in the feature extractor are applied. Besides, CNN can also extract topology attributes from images. It abstracts features from the primary image in the first layer and classifies the pattern with the last layer. The best property on pattern recognition mission was implemented. For example, Hochuli *et al.* (2018) proposed that the CNN classifier can handle the complexities of touch numbers more efficiently than all the segmentation algorithms provided in the literature, and achieved a recognition accuracy of 97%.

CNN is utilized to learn complex, high-dimensional data, and differs in how the convolution and sub-sampling layers are queried. The difference is in their structure. Generally, the first

layer is an alternation of the convolutional layer and the sub-sampled layer or convolutional filtering and down-sampling. The convolutional layer is used to extract basic visual features from the local receptive domain. It is organized in a plane called a simple unit of neurons, also known as feature mapping. Each group has 25 inputs connected to the 5×5 area in the input image, which is the local receptive area. Furthermore, the down-sampling operation through convolution filtering has a ratio of 2.

Many CNN constructions are proposed for distinct problems such as object recognition and handwriting character recognition. Furthermore, to ensure some level of invariance of scope, shift and distortion, CNN mixes three primary hierarchical fields such as local receptive area, weight sharing and spatial sub-sampling. Trainable weights are assigned to each connection for the standard neural network, but all elements of a feature map share the equal weight. This characteristic is evidenced by the fact that the primary feature detectors useful on a portion of the image may be helpful throughout the image. Also, weight sharing techniques allow for a reduction in the number of trainable parameters. For instance, LeNet5 has only 60,000 trainable parameters out of 345 308 links.

Since the exact position of the abstracted features is insignificant, the spatial resolution of the feature map is reduced by the sub-sampling layer. Such a layer includes as many characteristic maps as the previous convolutional layer, while with half the amount of rows and columns. Specifically, each unit j is related to a 2x2 sensitive area, calculates the average of the four inputs *yi*, multiplies them with the trainable weight *wj* and adds the trainable deviation *bj* to get the activity level *vj*:

$$v_j = w_j \frac{\sum_{i=1}^{4} y_i}{4} + b_j.$$

(Equation 1)

The weight sharing technology can be used for the subsampling layer in each feature map, and the trainable parameters consist of the shared weight and the bias. The characterization of a specific convolutional neural network referred to as LeNet5 below and the architecture of LeNet5 as depicted in Figure 3.3.

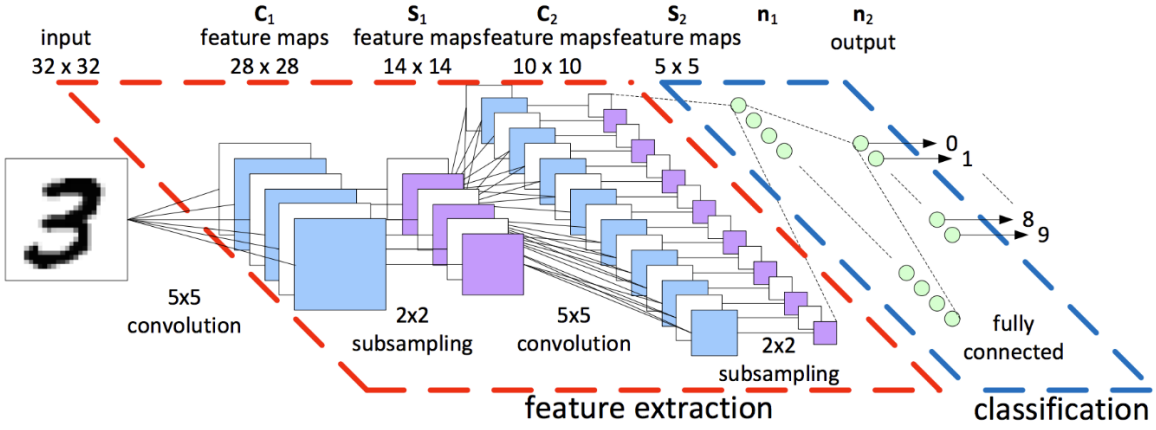


Figure 3.3: The architecture of LeNet5

LeNet5 adopts an original image of 32 × 32 pixels as input. It consists of two convolutional layers (C1 and C2), two sub-sampling layers (S1 and S2), one fully connected layer (N1) and an output layer (N2). It can be seen from Fig.3.3 that the convolution and sub-sampling layers are interlaced. In particular, the first convolutional layer C1is composed of six 28×28 units feature maps. The following S1 reduces the resolution by 2, while the next layer C2 expands the number of feature maps to 16. Here, each feature map of S2 is not connected to each feature map of C2. Each unit of C2 is connected to several receiving fields at the identical position in the subset of S1. These combinations are random, but they also decrease the number of free parameters and compel the different feature maps to draw different features when different inputs are obtained. The layer S2 is used as S1, and the size of the feature map is reduced to 5x5. Finally, the minimum output provides the class of the input mode.

After the model architecture is defined and designed, the model requires to be trained with the training dataset in order to be capable of recognizing the handwritten numbers. So, one epoch implies one forward and the backward pass of all training sets. If we visualize the whole training log, the model will be more stable with more sequences of epochs.

3.5 K-NN

K-NN is one of the most straightforward and well-known non-parametric algorithms which is suitable for large numbers of data. The K-NN algorithm has been applied for the statistical calculation, scene identification and also writer recognition systems. In some previous research studies, K-NN has been used for handwritten character recognition, and a high recognition

accuracy was obtained. In 2014, Babu *et al.* proposed four feature extraction techniques that are composed of water Reservoir principle based features, the number of loops in the image, maximum profile distances and fill hole density feature, and then experimented on the MNIST dataset. The recognition accuracy with this method is 96.94%. Rakesh *et al.* (2012) introduced feature mining algorithms to calculate the feature vector and experimented with the pattern on the Devanagari vowels dataset. Also, the KNN algorithm is adopted and achieved a recognition performance of 96.14%.

In K-NN, K represents the number of votes used for decision making. It is optimal to select an odd value of K, so it eliminates the connection between the two groups. The training set is a multidimensional array which includes the characteristic values of the training image and the class labels, while the test set includes unique values. There are three stages of implementing for the K-NN classifier:

First step: The distance metric is used to compare the input vector *x* with each training image sample *y* to find the most similar *k* neighbors. The effect of the K-NN algorithm relies on two fundamental factors: an appropriate distance function and the parameter *k*. In this field of study, the Euclidean Distance (ED) is the most widely used distance metric since it supplies the normalized value. Other distance metrics comprise Manhattan, Chebychev and Minkowski, (please consult Appendix A). The ED is calculated by the following equation where *N* is the number of dimensions of *x* and *y*. Then the distances between *x* and *y* are contrasted to recognize the closest neighbors to *x*.

$$d(x,y) = \sqrt{\sum_{i=1}^{N} (x^i - y^i)^2}$$ (Equation 2)

Second Step: Once the distances of the test objects are compared with known objects, they can be ranked accordingly. If 1000 known samples are given, the resulting gaps can be sorted from 1 to 1000. The value *K* indicates the number of levels to use, and it is a hyperparameter of the model. For instance, if *K* is equal to 31, then the first 31 distance vector values are considered. Generally, if *K* is too small, the model will not be well promoted due to high variance. In particular, it is highly sensitive to outliers. Conversely, if *K* is too large, the accuracy of the model will deteriorate.

Third Step: Considering that the case is true or false, the one with the most significant number among the 31 results is the value of the test case. Then, if 16 results point to true and 15 results indicate to false, the test sample is correct. That is why the unique value of *K* has been chosen. It should be noted that the K-nearest neighbor algorithm does not have a learning model, and the classifier only stores data points and compares with the new target points with them. That is a comparison with other classifiers, one clear example is the logistic regression model, which learns an uncomplicated mathematical model on the training set.

The K-NN method is not only applicable to complex classification problems with irregular decision boundaries but also regression problems. However, it may be computationally intensive for large training datasets because a large amount of distance must be calculated for testing. In general, a dedicated tree-based data structure can be used to speed up the search for the nearest neighbors. For regression problems, the model classifies the average of the objective value of the nearest neighbors. In both cases, distinct weighting strategies can be applied. To sum up, the principle of K-nearest neighbors is as follows: given the new points in the feature space, find K nearest points from the training set and assign labels for the majority of those points.

3.6 SVM

The Support Vector Machine (SVM) algorithm invented by Vapnik and Cortes (1998) is a powerful discriminant classifier that has been effectively applied to many pattern recognition or classification problems and has obtained positive results. Besides, due to its simplicity, flexibility, prediction capability and global optimality, it is considered to be the most advanced tool for solving linear and nonlinear classification problems. They are based on structural risk minimization instead of the empirical risk minimization that is traditionally used for artificial neural networks.

Some research institutes have proposed SVM as a learning classifier for capacity control with regression and binary classification problems. It has also been certified as being very excellent in many other applications such as face detection, text classification and handwritten number recognition. For instance, a 2017 paper by Phangtriastu, Harefa, and Tanoto achieved the highest handwritten character recognition accuracy of 94.43% using the SVM classifier with the combination of feature extraction algorithms which are projection histogram and HOG.

Shamim *et al*. (2018) adopted the SVM with RBF and ten cross-validation method to select the parameter to gain the highest recognition rates, achieving more than 93%. Besides, Sadri, Suen and Bui (2007) used the recognition system which contains segmentation and were able to get 96.42% recognition accuracy on handwritten numeric strings using SVM classifiers.

In particular, SVM is primarily used to determine the best separation hyperplane or decision surface by employing new techniques based on mapping sample points to high dimensional feature spaces although the first SVM is a linear binary classifier, which is useful for two-class classification tasks. However, it does not supply great separation for non-sparse complex data. For classification, the SVM attempts to search the best hyperplane by segregating the points of two classes to the greatest extent, which correctly classifies the data points. For linear separable tasks, the SVM algorithm merely seeks merely out the separating hyperplane with the highest margin. Moreover, to resolve the problem of non-linear data, SVM has the kernel trick that can acquire better accuracy. Fig.3.4: represents the working of the SVM classifier as depicted below:

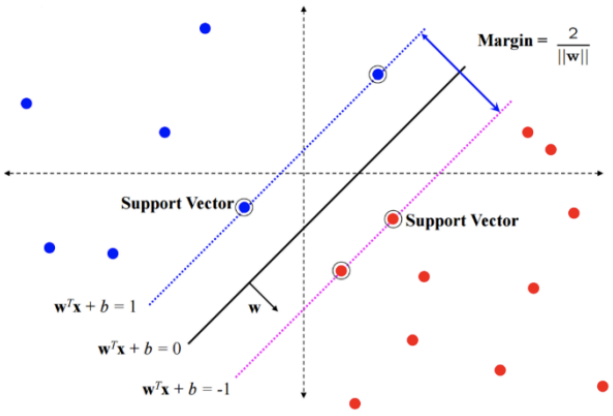
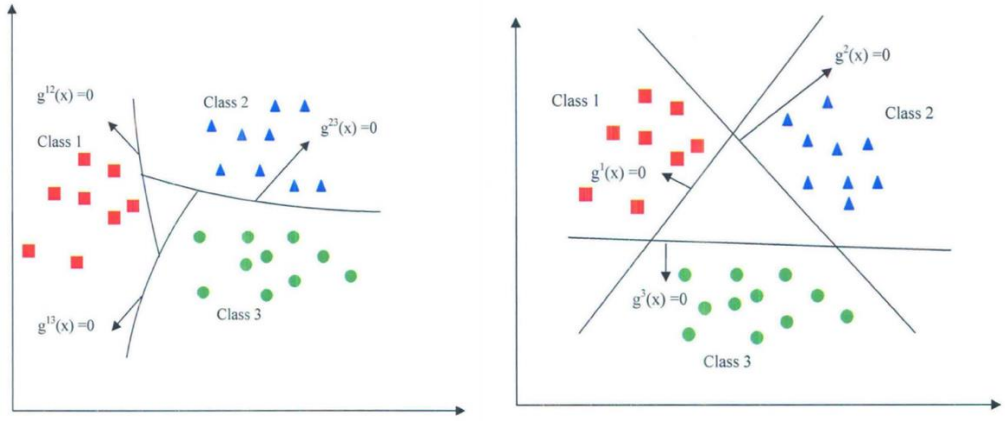


Figure 3.4: The working of the SVM classifier

Although SVM is primarily devised for binary mode classification, multiclass pattern identification problem may also to be resolved by associating several binary SVM classifiers. Two conventional methods are widely used to solve the multi-class problem of binary classifier SVM: "One Versus One" (OVO) and "One Versus All" (OVA), which are represented in Fig.3.5. For the OVO method, a classifier is set up for each pair of classes to segregate the classes two by two. By contrast, in the OVO scheme, a classifier is set up for each type and

assorted to the segmentation of this class from the others. OVA is generally used for identification because of its low complexity.



(a) (b)

Figure 3.5: Class boundaries of OVO (a) and OVA (b) SVM formulation for three-class problem

The first idea is to use the transformation function to change *xi* into a higher dimensional space so that the linear separation of the samples can be implemented in this new space to deal with the nonlinear decision boundary dimension. SVM has several of the most common kernels to solve these problems. Among them, the simplest kernel is the Linear kernel. The most popular kernel is the radial basis function (RBF) kernel. The kernels for normalizing data problems are the Polynomial kernel and the Sigmoid kernel.

- The linear kernel: $K(x, y) = x \times y$ (Equation 3)
- RBF kernel: $K(x, y) = \exp(-\gamma \|x - y\|^2)$ (Equation 4)
- The Sigmoid kernel: $K(x, y) = \tanh(\beta_0 \, x \, y + \beta_1)$ (Equation 5)
- The polynomial kernel: $K(x, y) = [(x \times y) + 1]^d$ (Equation 6)

In this work, the inner product of the vectors in the feature space is simply computed by the kernel function *K (·,·)* to map the data in the feature space. It allows constructing a decision function that is non-linear in the input space, but equal to a linear function in the feature space:

$$f(x) = \text{sign}\left(\sum_{support\ vectors} y_i \alpha_i K(x_i, x) + b\right) \quad \text{(Equation 7)}$$

Where $K(x_i, x) = exp(-\gamma||x_i - x||^2)$ is a kernel function established on the RBF.

3.7 RF

Random Forest (RF) is a collective term for a combination of classifiers using the L-tree classifier {h (x, Θk), k = 1,... L}, where Θk is an independent random vector of the same distribution and x is Input. It can be said that random forest is a series of methods which include several algorithms based on this definition. The concept of the random forest was introduced based on the bagging principle in 2001 by Breiman. A decision forest is a collection of some decision trees that act in a parallel pattern. It is distinct from the way that each tree attempts to classify the data completely independent of other trees. Furthermore, boosting is a collection classification method that means building a set of basic classifiers, each of which is trained on a guided replicate of the training set and then makes a decision based on the vote. In 2007, Bernard, Adam and Heutte experimented with the Forest-RI algorithm on the MNIST handwritten digital database, which is considered a random forest reference method, and the accuracy of handwritten digit recognition was over 93%.

Since not all features contribute to the recognition rate, some features may degrade the results. Therefore, Breiman recommended that randomness can be used in the choice of training samples and the group of elements applied to classify the data. The accuracy of RF depends on several factors:

- The power of each tree in the forest;
- The overall recognition accuracy of a forest varies with the intensity of a single tree;
- Correlation between each pair of trees in the forest;
- The error rate will be increased in model predictions when increasing the relationship or dependence in each couple of trees.

When a sufficient number of decision trees and different feature sets are adequately prepared, only the most persistent features in the data for classification are desirable and unrelated features have little effect on their efficiency. In addition, the random selection also assists the classifier to deal with missing values. Random forest classifiers can avoid over-fitting if the number of trees has been controlled.

In the Forest-RI algorithm, Bagging is used with the principle of random feature selection. The training phase of the method involves building multiple trees, each of this trained on the Bagging principle. A set of sample is randomly abstracted from the available training data for each tree. Then at each node, a random feature set involving in a regular number of feature variables is chosen from the feature vector. Some arbitrary functions and linear combinations are generated from the selected elements and trained to search the best linear combination of the characteristic variables. The RF applies the Gini standard derived from the classification and regression tree (CART) decision tree where pi represents the proportion of data samples from class i. The CART algorithm modifies the feature selection process at each node of the tree by adopting a random subspace principle.

$$Gini = \sum_{i=0,1} p_i (1 - p_i) . \quad \text{(Equation 8)}$$

The other significant criterion is the entropy function which is computed as follows:

$$Entropy = \sum_{i=0,1,2,\ldots,n} -p_i \log(p_i) . \quad \text{(Equation 9)}$$

In the past few years, variants of the Forest-RI algorithm have been proposed by several researchers. For example, Breiman (2001) developed another program for growing RF called Forest-RC, where each node's segmentation is based on a linear combination of features rather than a single function. That means that the processing to provide only a small amount of input, and the original Forest-RI method is difficult to handle. In 2004, Robnik attempted to enhance the combination process of the original Forest-RI by leading into a weighted voting method. The goal is to consider a limited subset of classifier outputs due to the individual accuracy of similar instances.

3.8 The Key Techniques in this Experiment

3.8.1 Normalization and Reshape Data

The MNIST handwritten digits have been size-normalized, centered and stored sequentially as 28 × 28 pixel images in gray-level bitmaps. The pixel values are grayscale between 0 and 255. The background is mostly close to 0 and those close to 255 represents the digit. Besides, the pixel values can be quickly normalized to the range of 0 and 1 by dividing each value by a maximum of 255. Some grayscale samples from the MNIST dataset are displayed in Fig. 3.6.

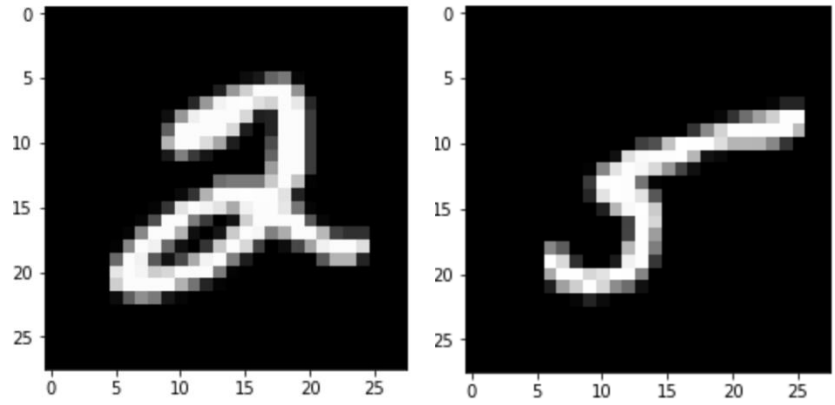


Figure 3.6: Some grayscale samples from the MNIST dataset

The training set is constructed as a 3-dimensional array of examples. For multi-layer perceptron models, it is necessary to reduce the image to a pixel vector. In this case, an image that is reshaped to a size of 28 x 28 x 1 will be a 784-pixel input value.

3.8.2 Slant Correction and Sharpening

Slant correction is very significant in the pre-processing stage of this study because handwritten digits with a pronounced slant present considerable difficulty for OCR. A slant in OCR is defined as the slope of rotation out of the reference plane. Since digital skewing can lead to inaccurate results in subsequent recognition processes, detecting and repairing images at oblique angles is a particularly important step, as well as the slant correction minimizing the error rate with in recognition. Moreover, a robust OCR must be able to hold high performance regardless of the position and slant of a given character or number.

The slants of the documents usually occur during image collection, and the characters in the document are also tilted during image acquisition. The characteristics of handwritten digits are primarily influenced by personal style and direction, to determine the position of the text, whether horizontal, vertical or forming a fixed angle. To be precise, the tilt is the angle formed by the near vertical stroke of the writing and the specific vertical direction. Meanwhile, it is one of the features that makes handwriting more challenging to process automatically than printed text. The slope of the text is the angle of contrast between the baseline and the horizontal of the sentence, which is not the same as the slant of the handwritten digits. Some instances illustrating these cases are displayed in Fig. 3.7.

Figure 3.7 Some slant samples from the MNIST dataset

In handwriting, slant estimation, detection, and removal are necessary components to perform a standardization process such as OCR, to optimize training procedures and reduce computational costs. After the slant removal process, the number should be a state in which the vertical stroke is parallel to the vertical axis of the page. Due to its significance, some researchers have already developed technologies for slant removal. Several instances of applying this technique for handwriting recognition will be introduced in detail below.

Kavallieratou *et al.* (2018) proposed a new technique for removing skew from the entire document page that prevents the process from becoming segmentation into text lines and words. The method mentioned first depends on tilt angle detection from proper segment selection. Then, a slant correction technique is applied. Additionally, the presented slant correction technology can be combined with another slant detection algorithm. However, it should be noted that this technique is only suitable if the tilt is uniform throughout the document image. As a result, this non-segmentation technique ensures the minimization of additional noise that may be introduced from the segmentation procedure. To examine the accuracy of the proposed technology, Kavallieratou, and his team performed experiments on a dataset of four document images, namely, TrigraphSlant dataset, two historical documents, and a printed dataset.

In 2006, Frank introduced an algorithm for automatically removing slant, which segments the text into words according to the principle of horizontal and vertical projection histograms. Then, each character is shared by angles that maximize the peak height in the vertical projection histogram of the role. The purpose of slant removing is to make the text more appropriate for digital processing of a system. Therefore, the best method to assess an algorithm could be to

measure the performance changes of the systems it contains. Frank (2006) reported that the character recognition system not only improved the recognition accuracy by 9% but also shortened the training time when adding the slant correction algorithm. However, it is still difficult to compare distinct algorithms when they have been applied and evaluated in different systems.

Rodiah *et al*. (2016) proposed a slant correction method for detecting handwritten images of geometric distortion. Firstly, the straight letter separation points are obtained by the segmentation of candidate oblique images. Then, the detected slant angles are received by applying the way of the method of the affine transformation 2D. Finally, the digital image is moved according to the slant of the character in the x or y-axis direction and a specific scaling factor is adopted. This method can rotate the angle of the image from - 45 to 45 degrees and then chooses the best corner. The results showed that the 2D affine transformation successfully detects and corrects the skew of letters, thus avoiding the excessive segmentation of candidate images. It improves the accuracy of handwritten character recognition.

In this study, slant detection and correction are essential components for performing the preprocessing phase of a digital identification system. In general, the slant detection process in handwritten digit recognition includes the following steps:

- Define the maximum angle of the raw image 45°.
- Rotate to deliver the pixel values of the starting writing location with the detected oblique from the horizontal image of handwriting by the following formulas：

$$x_2 = \cos(\theta) \times (x_1 - x_0) - \sin(\theta) \times (y_1 - y_0) + x_0$$
$$y_2 = \sin(\theta) \times (x_1 - x_0) + \cos(\theta) \times (y_1 - y_0) + y_0 \qquad \text{(Equation 10)}$$

where:

X_0, Y_0 = Central coordinates of the input image

θ = The axis of rotation

The slant correction algorithm is depicted as follows:

- The image matrix is partitioned into upper and lower halves. The cores of gravitation of the lower and upper halves are calculated and connected. The slant of the connecting line determines the slope β of the image matrix.

- The skew-corrected image is obtained by using the following transformation to pixel values with coordinate points *x*, *y* in the raw image:

$$x' = x - y\tan(\beta - def), \quad y' = y,$$

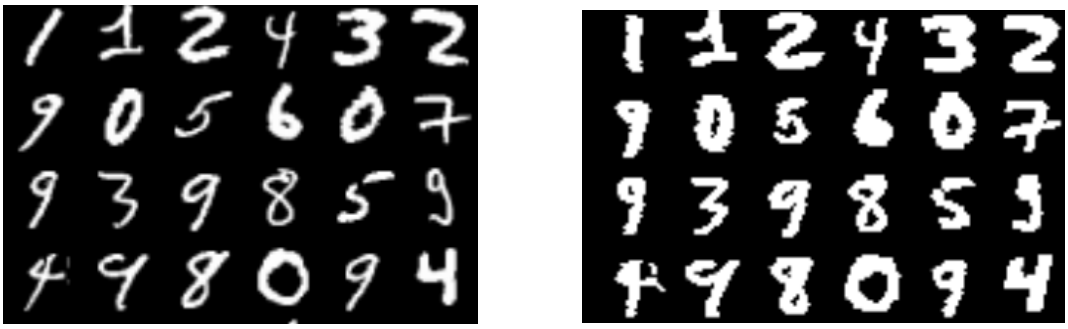
(Equation 11)

where:

x' , y' = The slant corrected coordinates

def = A parameter specifying the default oblique

Slant correction must precede another preprocessing task such as sharpening which applies a blurred negative vision to produce a mask of the raw image. Then, the unshaped mask is aggregated with the raw image, creating a picture that is sharper than the original. However, this operation needs to be done after the slant correction task since sharpening tends to change the image topology. The effect of skew correction and sharpening on the images can be seen in Fig. 3.8.



(a) (b)

Figure 3.8: Some digits in training set: original (a) and after deskewing and sharpening (b)

3.8.3 Elastic Distortion

Common distortions such as translations, rotations, and skewing can be generated by using affine displacement scenes to images. In 2003, Simard *et al.* found that elastic distortion is a form of image transformation which can simulate the variations of the handwriting to produce new data and improve the performance. The process of elastic deformation is depicted as follows.

Firstly, the image distortions were created by stochastic random displacement fields which are established from a uniform distribution between −1 and +1, that is Δx(x,y) = rand (-1,+1). Then, they are convolved with a Gaussian of standard bias σ. After standardization and multiplication by a scaling factor α which dominates the strength of the deformation, and they are used to the image. Mainly, σ represents the elastic coefficient. A small σ indicates more elastic distortion and the field seems like a completely random field after standardization. Conversely, the deformation is close to affine, and the displacements change into translations if it is tremendous. The process of elastic deformations on images for dataset expansion is shown in Fig. 3.9.

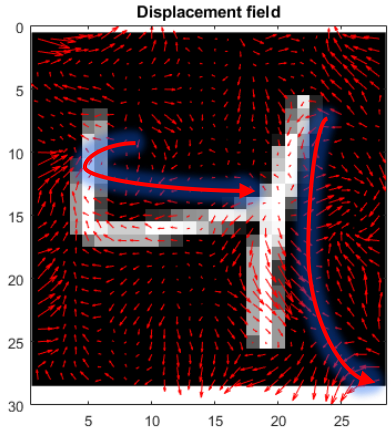


Figure 3.9: The process of elastic deformations on images

In the mentioned method, the elastic distortions are applied to each sample in the training set to produce several novel samples for each one. Fig.3.10 displays some examples created by elastic deformation.

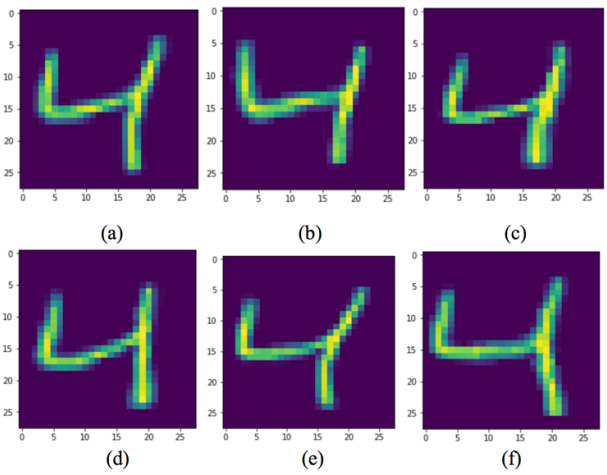


Figure 3.10: Some numbers (a) - (f) generated by elastic distortion

3.8.4 HOG

The HOG descriptor was first proposed by Dalal and Triggs (2005) for human body detection in an image. Recently, it is one of the most commonly and successfully used descriptors for computer vision and image recognition for object detection. The principle of the HOG descriptor is that the appearance and shape of the local object within an image can be explained by the arrangement of intensity gradients or edge directions. This technique divides the image into small connected regions and then calculates a histogram of the gradient direction or edge direction according to the mean differences. Moreover, HOG vectors are computed by taking direction histograms of edge intensity in a local area. In this research, each pixel is convolved with the common convolution kernel and is depicted as follows:

$$G_x = f(x+1, y) - f(x-1, y)$$

$$G_y = f(x, y+1) - f(x, y-1) \quad \text{(Equation 12)}$$

Where *Gx* and *Gy* represent the horizontal and vertical parts of the gradients, respectively, in this experiment, the HOG vector is computed over Rectangular-HOG in non-overlapping blocks.

The scope of gradient direction is limited among 0° and 180° to neglect negative gradient orientations, and the gradient magnitude M and the gradient orientation θ can be computed by:

$$M(x, y) = \sqrt{G_x^2 + G_y^2}$$

$$\theta(x, y) = \tan^{-1} \frac{G_y}{G_x} \quad \text{(Equation 13)}$$

After that, histograms are calculated from the occurrences of the local intensity gradients across large constructions of the image. The performance and vector size of the HOG descriptor depends on the number of blocks selected. Finally, the feature descriptors are normalized.

3.8.5 PCA

PCA is a powerful and extensive technology applied for data exploration and compression in neural networks and machine learning. It involves linearly converting a set of related variables into alternative representations that emphasize the variance between observations. Effectively, it reduces the dimensions of the observed data by eliminating redundancy. PCA can provide a

lower-dimensional representation if a multivariate data are visualized as a series of coordinates in the high-dimensional data space. However, Das (2012) mentioned that only the features extracted by the PCA algorithm are not sufficient to solve the variability of the handwritten digit mode. Furthermore, the QTLR-based topology features also have limitations in classifying digital patterns into individual scripts. Consequently, the combination of MPCA + QTLR is applied to increase the recognition accuracies significantly to 98.7%.

By decreasing the dimensionality of observed data, the low-dimensional database can be constructed with more presentation visualizations or reduce memory and processing demands in the high-dimensional database. In particular, dimensionality reduction may even increase the accuracy of the OCR system or predictive model.

PCA is an eigenvector-based multivariate analysis technique that usually extracts the best data variance. The other main benefit of PCA is that once the patterns are detected in the data and then the data are reduced without too much loss of information. In this experiment, the PCA will be used to minimize the proportion of the MNIST dataset from 784 to a lower value to facilitate calculations. The mathematical equations applied to implement PCA are depicted in detail below. Consider a group of *n* observations on the vector of *p* variables formed in a matrix *X (n x p)*

$$\{X_1, X_2, \cdots, X_n\} \in R^p$$

The PCA approach finds *p* principal components, and each one is a linear un-correlation of *X* matrix columns, in which the weights are factors of an eigenvector to the correlation or data covariance correlation matrix. The condition is that the data are concentrated and normalized. The first principal component of the linear transformation is:

$$Z_1 = a_1^T x_j = \sum_{i=1}^{p} a_{i1}\, x_{ij}\,, j = 1, \cdots, n \qquad \text{(Equation 14)}$$

where:

$$a_1 = (a_{11}, a_{21}, \cdots, a_{p1})$$
$$x_j = (x_{1j}, x_{2j}, \cdots, x_{pj})$$

If a_1 and x_j are chosen as such the variance of Z_1 is the maximum. Each principal component begins from the origin of the ordinate axes.

3.8.6 K-fold Cross-validation

Cross-validation is a technology to assess predictive models by dividing the original sample into a training dataset to train the model, and a validation dataset to evaluate it. In the experiment, the selected MNIST dataset was randomly divided into a training set and a small part of the validation set. At each training epoch, the model is provided with training data and permitted to update its weights, learning how to fit the training data more accurately. However, the performance of the validation model on the independent datasets which not applied to train – is essential and performed at the end of each epoch to guarantee that the model is capable of generalizing to new data.

The advantage of K-fold cross-validation is that all observations are applied to training and verification, and each representation is only adopted for verification once to avoid over-fitting of the error rate. Mane and Kulkarni (2018) proposed that the CCNN's performance reached an average of 94.93% accuracy by using K-fold cross-validation. Although having a separate training and validation set makes things simple, 10-fold cross-validation will be used in this research. A specific description of this technology now follows.

In k-fold cross-validation, the original samples are randomly divided into *k* equal-sized subsamples. Among the *k* subsamples, a single subsample is reserved as the validation data for testing model performance, and the other *k-1* subsamples are combined into a training set. The cross-validation procedure is then repeated *k* times or folds, with each of the *k* subsamples adopted precisely once as the validation set. The collapsed *k* results can then be averaged to produce a single estimate.

In other words, the dataset will be divided into *K* approximately equal parts, and for each k = 1,..., K, the classification using a set of *p* features Z= (Z1,Z2,...,Zp) is applied. Then, calculate the error rate for *k*-fold is calculated as:

$$E_k(Z) = \frac{n_{Ek}}{n_k}, \qquad \text{(Equation 15)}$$

where: n_{Ek} = the count of misclassified images in the *k* fold

n_k = the size of the *k* fold

Hence, the cross-validation rate is:

$$CV_E(Z) = \frac{1}{K}\sum_{k=1}^{K} E_k(Z).$$

(Equation 16)

The process repeats the cross-validation for the different set of features and chooses the best association of them to minimize the $CV_E(Z)$.

3.8.7 Confusion Matrix

In 1998, the confusion matrix was proposed by Kohavi and Provost, which contained the information about actual and predicted classifications performed by a classification system. It concentrates on the predictive ability of a model rather than the time and speeds the model takes to complete the designation.

The confusion matrix is indicated by a matrix where each row shows the examples in a predicted class, while each column shows them in an actual class. This performance assessment tool can not only determine whether the model confuses two classes, but also evaluate the overall or average accuracy of the classifier. Therefore, in this experiment, the confusion matrix will be used for the evaluation of the model.

A confusion matrix of size *n* x *n* related to a classifier illustrates the predicted and actual classification, where n represents the number of different classes. Table 3.1 displays a confusion matrix for n = 2.

	Predicted Negative	Predicted Positive
Actual Negative	a	b
Actual Positive	c	d

Table 3.1: A confusion matrix for two classes

Where:

a = the number of correct negative predictions

b = the number of incorrect positive predictions

c = the number of incorrect negative predictions

d = the number of correct positive predictions

The prediction accuracy and error rate of the classification system can be received from this matrix as follows:

$$Accuracy = \frac{a+d}{a+b+c+d}$$

$$Error = \frac{b+c}{a+b+c+d} \quad \text{(Equation 17)}$$

3.9 Flow Chart

This experiment will be completed according to the following flow chart is depicted in Fig.3.11.

1. Pre - Processing: Slant correction, Sharpening, and elastic distortion
2. Feature Extraction using PCA or HOG
3. Classification using the CNN, K-NN, RF and SVM

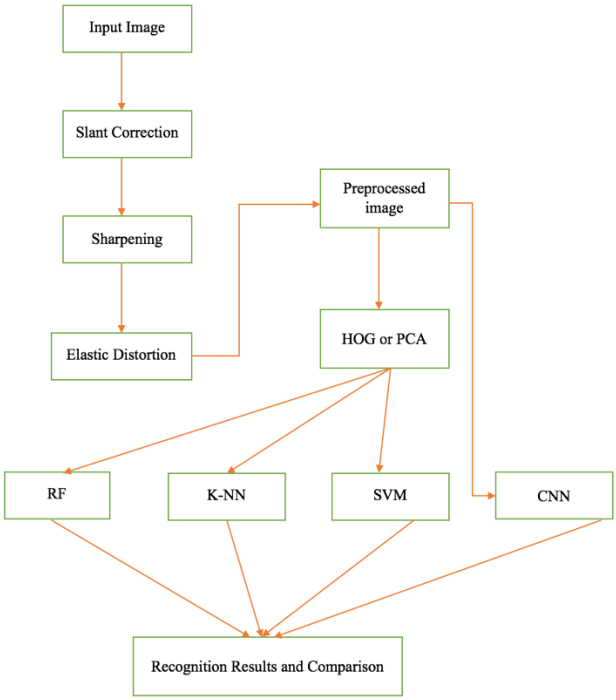


Figure 3.11: The flow chart for this experiment

3.10 Summary

This chapter has first outlined a summary of the research objectives and requirements in this experiment. Next, the research structure includes data collection methods, sample size, template-based model principles, and a variety of techniques used for design and evaluation and has been discussed in depth. The MNIST dataset is an excellent database for machine learning and pattern recognition methods while minimizing preprocessing and formatting. Therefore, this study used that for analysis.

There are several instances where some researchers have applied SVM, CNN, RF, and K-NN to obtain recognition accuracy of 94.43%, 97%, 93%, and 98.06%, respectively. Since previous researchers have fully validated the above four classifiers mentioned above, CNN, SVM, K-NN and RF will be used and compared in this experiment to determine which classifier has the highest performance. However, as this study strives to improve the recognition accuracy of more than 99% in handwritten digit recognition, preprocessing and feature extraction are the crucial roles of this experiment to reach the highest accuracy.

In the pre processing stage, slant correction and image sharpening are the focus of the design because they can solve the problem of different people using handwriting which is more or less oblique writing, improve legibility and help feature extraction in the next step. Also, HOG and PCA are the core technologies in the feature extraction phase. The PCA can be used to extract the best data variance, and the HOG feature vector can be applied to image recognition and object detection. In the evaluation phase, the four previously mentioned classification techniques will be evaluated using K-fold cross-validation, error rates, accuracy, classification reports and confusion matrix. Finally, the new handwritten numbers will be used to test the models. The resulting model with a recognition rate of more than 99% will identify the handwritten digits and show the predicted numbers.

The next chapter will expose the actual work of the system's implementation and experiment results. It will compare the differences in recognition accuracy between the four classifiers, as well as the effects of the image preprocessing techniques (including skew correction, sharpening, and elastic distortion) and feature extraction techniques (involving PCA and HOG) on accuracy.

4 IMPLEMENTATION AND RESULTS

4.1 Introduction

This chapter will describe the classification experiments in more detail and present the results from five combinations. These are as referenced in Fig.3.11:

1. Preprocessing +CNN
2. Preprocessing + PCA + K-NN
3. Preprocessing + PCA+ SVM
4. Preprocessing + HOG+ K-NN
5. Preprocessing + HOG+ RF

It is essential to revisit the data set used in these experiments to benchmark our comparisons. The processes and results of data pre-processing are introduced in depth. Besides, implementation details such as the package selection and argument set-up will be described. The final section of the chapter will compare the results of the five combinations and determine which combination can achieve an accuracy of more than 99%. Both the preparation and procedure of the experiment were accomplished using the programming language Python.

4.1.1 Data Set

The dataset used for the application is the MNIST dataset originally constituted of 60,000 training, and 10,000 testing images which are 28 x 28 grayscale (0 - 255) labeled and have bitmap format. Some numbers from the MNIST dataset are displayed in Fig. 4.1. The corrupted images such as missing values have been checked before application. There are no missing values in the dataset. It is an excellent database for machine learning and pattern recognition methods while taking minimal efforts in preprocessing and formatting. Furthermore, the similar counts for the digits, i.e., numbers from 0 to 9, are indicated by the histogram in Fig. 4.2.

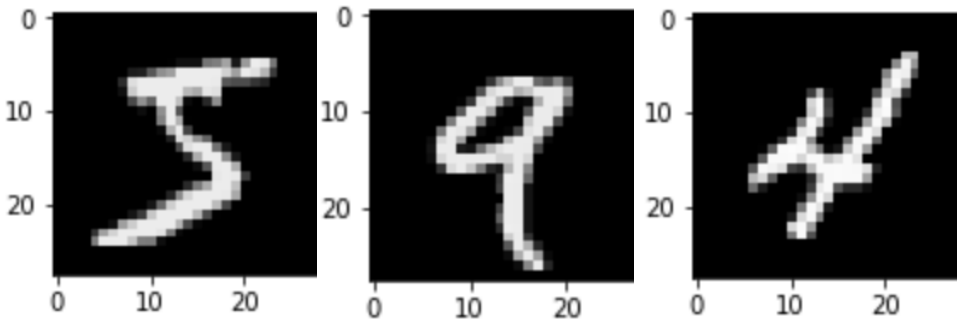


Figure 4.1: The numbers from the MNIST dataset

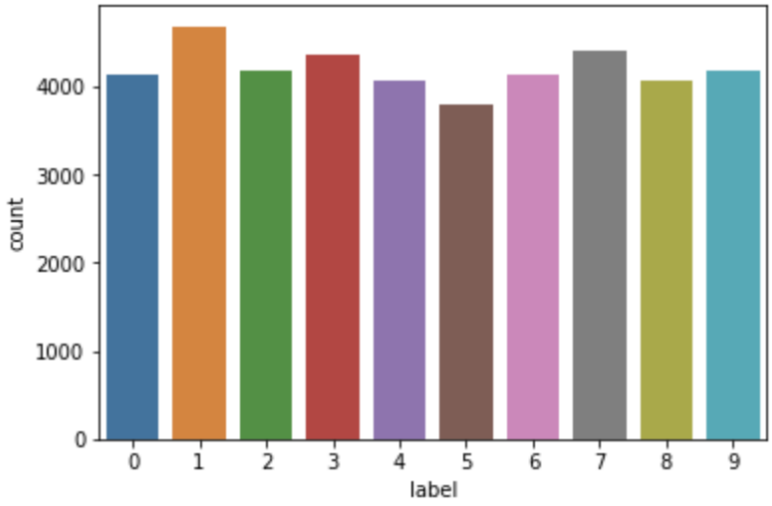


Figure 4.2 The similar counts for the digits

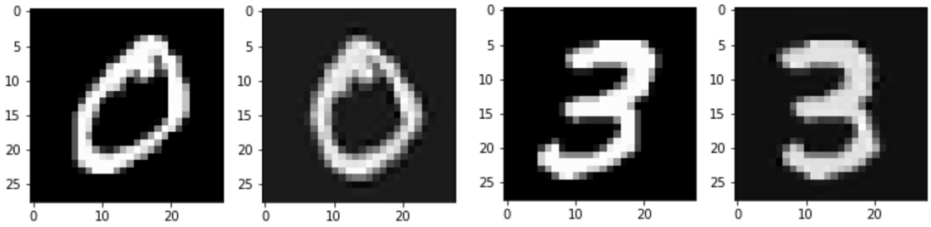
Although MNIST handwritten digits have been size-normalized and centred, it is necessary to reduce the effects of illumination differences. The background is mostly close to 0 and those close to 255 represent the digit. Hence, the pixel values can be quickly normalized to the range of 0 and 1 by dividing each value by a maximum of 255. Also, the training set is constructed as a 3-dimensional array of examples. For multi-layered perceptron models such as CNN, it is necessary to reduce the image to a pixel vector. In this case, an image that is reshaped to a size of 28 x 28 x 1 will be a 784-pixel input value.

4.1.2 Image Preprocessing

Slant correction and elastic distortion play very significant roles in the preprocessing stage of this study. The characteristics of handwritten digits are primarily influenced by personal style and direction to determine the position of the text, whether horizontal, vertical or forming a fixed angle. Since digital skewing can lead to inaccurate results in subsequent recognition processes, detecting and repairing images at oblique angles is a particularly important step, as well as the slant correction minimizing the errors in recognition. Moreover, the apparent distortions such as translations, rotations, and skewing can be generated by using affine displacement scenes to images.

For the slant detection and correction of the handwritten digits, the first step is to find the center of mass in the image to determine how much is needed to offset the image. Then, next step is to define the maximum angle of the raw image 45° and detect the covariance matrix of the

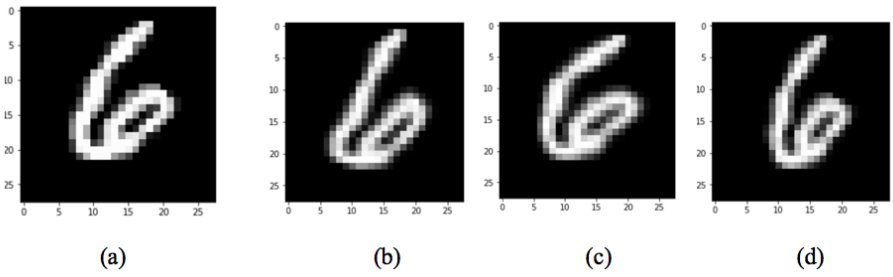
image pixel strengths. The slant of the connecting line determines the slope β of the image matrix, and the pixel values with coordinate points x, y in the raw image are transformed according to the formula of the slant correction algorithm described in Chapter Three. In this case, the **moments ()** function has been used to calculate the relevant quantities and the covariance matrix. Also, the **deskew ()** function has been applied to rotate the image according to the combinations of the appropriate quantities and patterns. Some examples of slant correction before and after are clearly shown in Fig. 4.3.



(a) (b) (c) (d)

Figure 4.3: Some examples before (a) & (c) and after (b) & (d) tilt correction

In the process of elastic distortion, the image distortions were created by stochastic random displacement fields which are established from a uniform distribution between −1 and +1, that is Δx(x,y) = rand (-1,+1). The entire process is implemented by the **elastic transform (image, alpha, sigma, random_state=None)** function and the present implementation support only gray-scale images. In particular, α represents the alpha value of the elastic transformation, and σ represents the elastic coefficient — the smaller the sigma, the more conversion. Finally, a processed image is returned by the **map coordinates ()** function that is an interpolation of the value of the original array as the coordinates have specified. In the proposed method, the elastic distortion has been applied to each sample of the training set to extend several new samples, and one clear example can be seen in Fig. 4.4.



(a) (b) (c) (d)

Figure 4.4: (a) represents the number before the elastic distortion, and (b)-(d) display the deformed numbers

4.2 The Combination of Preprocessing and CNN

CNN is a multi-layered neural feed-forward network with deep supervised learning architecture, which can be regarded as a two-part combination: automatic feature extractor and trainable classifier. In this section, the mixture of pre-processing and CNN has been applied to obtain a satisfactory result. Next, the creation of the model, the selection of parameters and the final results will be described in detail and the following flow chart in Fig. 4.5 will assist people in understanding the temporary structure of this section.

Figure 4.5: The flow chart for the combination of pre-processing and CNN

First of all, the **Keras Sequential () API** was adopted to create the CNN model and its architecture includes: **In -> [Conv2D-> relu -> MaxPool2D -> Dropout]*2 -> Flatten -> Dense -> Out** as well as it is shown in Table 4.1.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 24, 24)	832
max_pooling2d_1 (MaxPooling2	(None, 32, 12, 12)	0
conv2d_2 (Conv2D)	(None, 15, 10, 10)	4335
max_pooling2d_2 (MaxPooling2	(None, 15, 5, 5)	0
dropout_1 (Dropout)	(None, 15, 5, 5)	0
flatten_1 (Flatten)	(None, 375)	0
dense_1 (Dense)	(None, 128)	48128
dense_2 (Dense)	(None, 50)	6450
dense_3 (Dense)	(None, 10)	510

Table 4.1: The architecture of CNN model

The first layer is the **Conv2D** layer, which is like a series of learnable filters. Correctly, the first **conv2D** layer is set to 30 filters and 15 filters for the other one. A filter could be thought

of as a transformation of an image, and the kernel filter matrix is used for the entire picture. Therefore, each filter transforms a section of the image which is defined by the kernel size applying the kernel filter. The CNN can separate features that are useful anywhere from the feature maps.

The second central layer in CNN is the **MaxPool2D** layer that is simply used as a down-sampling filter. The purpose of viewing two adjacent pixels and selecting the maximum value is to reduce the computational cost and over-fitting to some extent. The choice of pool size is necessary because the more pooling dimensions that are higher, the more the down-sampling. According to the combination of **Conv2D** and **MaxPool2D** layers, CNN can combine partial features and obtain more global characteristics of the image.

Dropout is a regularization method, where the weight of each training sample is set to zero. That will randomly drop a portion of the nodes and forces the network to look at features in a distributed way. This function also ameliorates generalization and decreases the over-fitting. A **'relu'** is the rectifier to activate the function **max (0, x)**. Rectifier activation work can be applied to add nonlinearity to the network. Moreover, the role of the **flatten** layer is to transform the ultimate feature map into a single 1D vector. It integrates all the detected local features of the previous **Conv2D** layers. Finally, three **Dense** layers were added that is artificial neural networks (ANN) classifier. In the last layer of the model, **Dense (num_classes, activation='softmax')** was used to net out the probability distribution of each class.

After the model was created, the learning rate (LR) will be taken for the optimizer to converge faster and come closest to the minimum value of the loss function. Apparently, the bigger the LR the quicker is the convergence. The **ReduceLROnPlateau ()** function of **Keras.callbacks** is adopted to maintain the benefit of the fast computation period with a high LR. If the accuracy is not improved after three epochs, LR will be halved. Although the elastic distortion technique has been used to increase the data set during the preprocessing stage, the **ImageDataGenerator ()** function was still added here to create a compelling model to avoid over-fitting.

Finally, the original dataset and the preprocessed dataset were respectively applied to the previously created CNN model with 30 epochs, and the obtained result is shown in Table 4.2, where the handwritten digit recognition rate is 98.75% using the un-preprocessed dataset. In

contrast, the accuracy after preprocessing increased to 99.44%. However, the Training Time (TT) for both is the same for 3.5 hours.

	Non-preprocessing	preprocessing
RR	98.75%	99.44%
TT	3.5h	3.5h

Table 4.2: The recognition rate (RR) and Training Time (TT) based on CNN

Overall, the choice of parameters affects the performance of the CNN model. After many adjustments, the settings that can achieve the highest performance are Conv2D (30), Conv2D (15), Dropout (0.2), dense (128), dense (50) and dense (10). On the other hand, since the recognition rate after preprocessing has reached 99.44% from 98.75% at the same running time, this result indicates that the preprocessing technique plays a significant role in this combination for handwritten digit recognition. The question raised by this experiment can be addressed by this combination, which improved the accuracy by more than 99%.

4.3 The Combination of Preprocessing, PCA and K-NN

PCA is a powerful and extensive technology applied for data exploration and compression in neural networks and machine learning. K-NN is one of the simplest and best well-known non-parametric algorithms which is suitable for large numbers of data. In this part, the combination of preprocessing, PCA and K-NN has been used for an experiment in handwritten digit recognition experiment. Also, the selection of the number of components in the PCA, the adjustment of the parameters and the creation of the K-NN model will be explained in depth. The flow chart of this combination is illustrated in Fig. 4.6.

Figure 4.6: The flow chart for the combination of preprocessing, PCA and K-NN

PCA is an eigenvector-based multivariate analysis technique that usually extracts the best data variance. It uses correlations between specific dimensions and tries to provide a minimum number of variables to maintain information about the distribution of the original data. Since people typically use 2D or 3D plots to observe the data structure, the original 784 dimensions generated the first three principal components and applied a 2D scatterplot to find out how many changes were made from the total dataset. In this case, the function **ggplot ()** has been adopted to describe the information from the components, specifically for particular digits. However, as shown in Fig. 4.7, it is not sufficient to separate all the information.

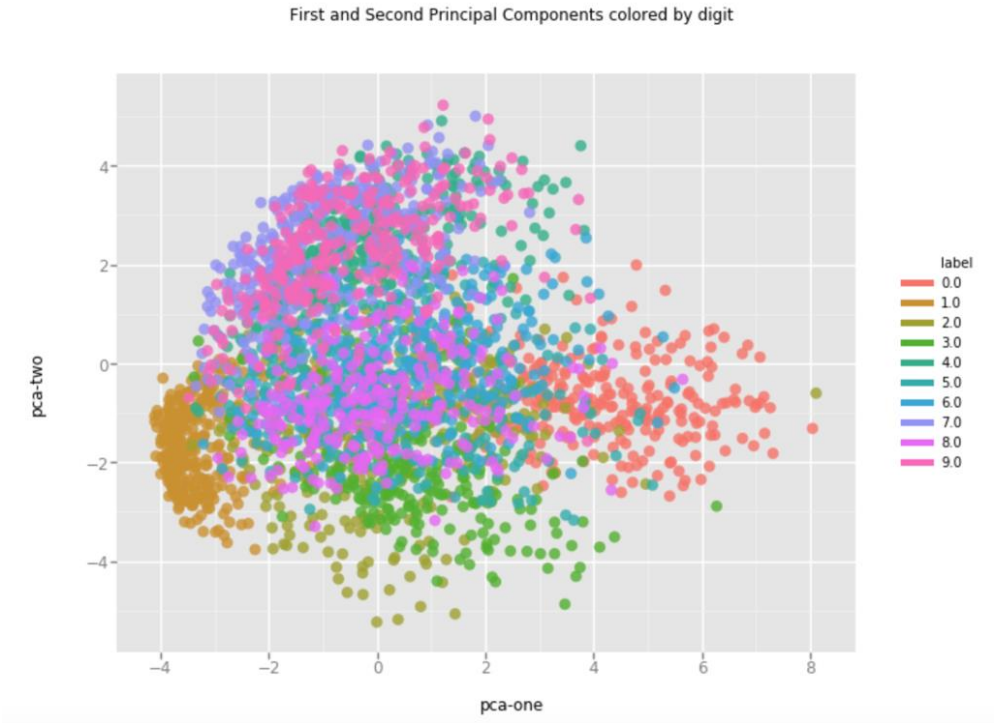


Figure 4.7: First and Second Principal Components colored by digit

The first step in the process of PCA is to generate a screen plot to offer a view of the cumulative sum of variance of the component. Fig.4.8 displays the screen plot that includes the explained variance ratio in numbers of components in the pre-processed MNIST data. As can be seen from the figure below, the scree plot intuitively indicates that the first 200 components explained the cumulative variance ratio related to the original variance, since there is not a significant change of cumulative explained variance after the 200 component occurs.

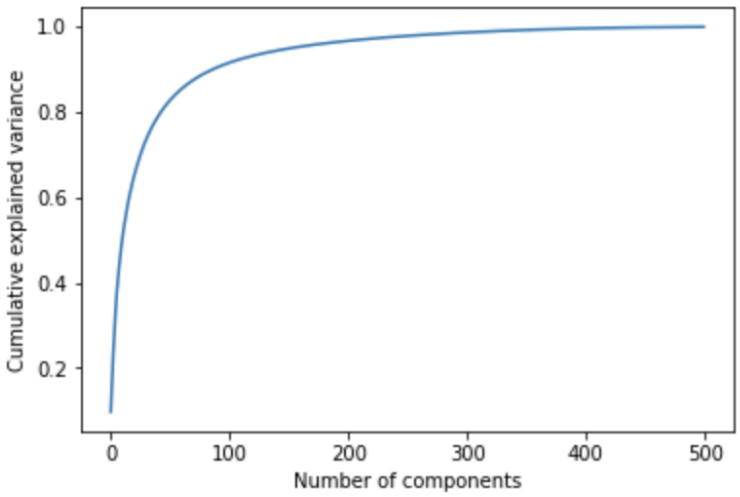


Figure 4.8: The Scree Plot of the Cumulative sum of Variance

Additionally, the first 200 eigenvalues correspond to approximately 98% of all variance meaning that only 2% of the information is lost. The preprocessed data was dimension reduced by applying the **PCA (n_components=200)** and **pca.fit_transform ()** functions, and some examples are shown in Fig.4.9.

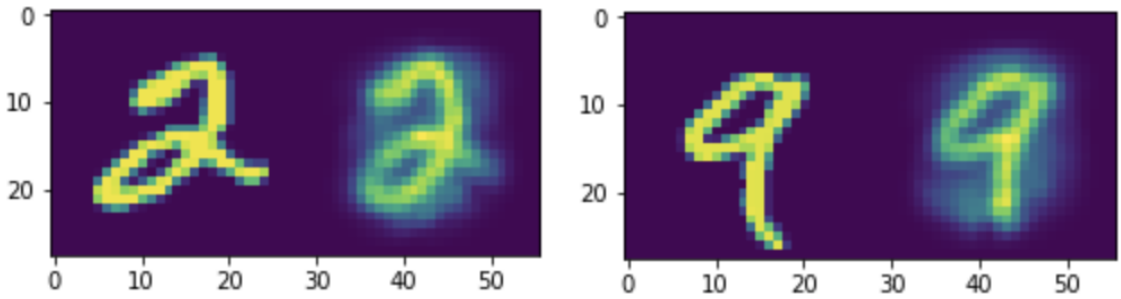


Figure 4.9: The numbers after the preprocessing and PCA

In K-NN, *k* represents the number of votes used for decision making. It is optimal to select an odd value of *k*, because it eliminates the connection between the two groups. Then, each of the *k* values was looped, and a **KNeighborsClassifier ()** was trained, offering the training and testing data to the fit method of the model. After the model is trained, the validation data will are evaluated. Finally, the odd numbers from 1-30 were used to the number of votes *k*, where **k=3** achieved the highest accuracy of 99.17% on the validation data after preprocessing and PCA in particular. Table 4.3 indicates the impact of using preprocessing and PCA on the accuracy of handwritten digit recognition based on the K-NN model.

	Non-preprocessing	Preprocessing	Preprocessing + PCA
RR	96.68%	98.14%	99.17%
TT	5.8 s	4.7 s	0.9 s

Table 4.3: The comparison of RR and TT using preprocessing or PCA based on the K-NN

To sum up, when **k=3**, the combination of preprocessing, PCA and K-NN can also reach the highest accuracy of more than 99%. Besides, the efficiency of using pre-processing technology has increased from 96.68% to 98.14%. On the other hand, the training time has been reduced from 5.8s to 4.7s. If the K-NN model only recognizes the preprocessed data, the recognition rate cannot be maximized. Feature extraction of PCA is an indispensable stage in the process.

4.4 The Combination of Preprocessing, PCA and SVM

The Support Vector Machine (SVM) algorithm is a powerful discriminant classifier that has been effectively applied to many pattern recognition or classification problems and has obtained favorable results. Shamim *et al.* (2018) adopted the SVM with RBF and the ten cross-validation method to select the parameter to gain the highest recognition rates, reaching more than 93%. This section introduces a multivariate analysis framework for feature detection in a recognition system, while the PCA and SVM based supervision scheme can determine patterns in the recognition system. The data set for this experiment was pre-processed, which is a combination of pre-processing, PCA and SVM. Fig.4.10 displays the flowchart representing this combination.

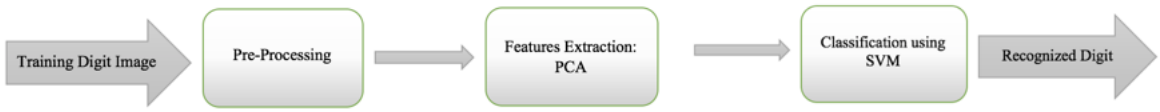


Figure 4.10: The flow chart of the preprocessing, PCA and SVM

As Fig.4.8 illustrates an asymptote at around 200, which is the optimal number of PCs to use, the preprocessed data was dimension reduced by applying the **PCA (n_components=200)**, and **pca.fit_transform ()** functions as well.

The next step is to create an SVM classification model by using the **LinearSVC () function**, where SVM comes with many built-in parameters. In order to maximize the performance of this model, the settings of these parameters include **C=1.0**, **cache_size=200**, **decision_function_shape='ovr'**, **gamma='auto'**, **kernel='rbf'** and so on. Moreover, an SVM can only be assorted into two categories. The method for differentiation of *N* (10 digits for this study) classes is to train *N x (N-1)/2* classifiers. Finally, the preprocessed and PCA-derived data is applied to the already-created SVM model for digital identification. The results obtained are shown in Table 4.4.

	Non-preprocessing	Preprocessing	Preprocessing + PCA
RR	91.17%	94.54%	94.90%
TT	34.5 s	87.7 s	21.0 s

Table 4.4: The comparison of RR and TT using preprocessing or PCA based on the SVM

Overall, preprocessing is one of the essential techniques to improve the recognition rate from 91.17% to 94.54%, and the combination of preprocessing and PCA reaches the highest accuracy of 94.90%. However, this combination is not able to solve the aim of this research which is to achieve a recognition rate for handwritten digits obtained of over 99%. Also, the running time is as long as 88s compared with other models.

4.5 The Combination of Preprocessing, HOG and K-NN

In recent times, the HOG descriptor has become one of the most common and successfully used descriptors for computer vision and image recognition for object detection. In the HOG feature descriptor, the distribution of directions of gradients is used as features. Moreover, HOG vectors are computed by taking direction histograms of edge intensity in a local area. A 2017 paper by Phangtriastu, Harefa, and Tanoto compared the most commonly classifiers SVM and ANN, while this experiment achieved the highest accuracy 94.43% using the SVM classifier with the combination of feature extraction algorithms which are a projection histogram and HOG. According to the above study, the combination of PCA and K-NN can achieve an accuracy of more than 99%. So in this section, the combination of preprocessing,

HOG and K-NN were applied to the exploration and analysis. The flow chart of this combination is introduced in Fig.4.11.

Figure 4.11: The flow chart of the preprocessing, HOG and K-NN

To calculate a HOG descriptor, the projection profile that includes two types namely the vertical and horizontal gradients has to be computed. In addition, the same results could be obtained by using the **cv2.Sobel ()** function. Next, the calculation of the gradient magnitude and direction can be done applying the function **cv2.cartToPolar(x, y, angleInDegrees=True)**. Then, the image gradients were pooled and normalized into orientation bins in a dense manner by **hog (orientations=9, pixels_per_cell=(14, 14), cells_per_block=(1, 1))**.

In K-NN, each of the *k* values was also looped, and a **KNeighborsClassifier ()** was trained, offering the training and testing data to the fit method of the model. Besides, the odd numbers from 1-30 were used to the number of votes *k*, where *k=5* achieved the highest accuracy of 95.39% on validation data after preprocessing and HOG in particular. Table 4.5 below displays the impact of using preprocessing and PCA or HOG on the accuracy of handwritten digit recognition based on the K-NN model.

	Raw Data	Preprocessing	Pre + PCA	Pre + HOG
RR	96.68%	98.14%	99.17%	95.39%
TT	5.8 s	4.7 s	0.9 s	13.8 s

Table 4.5: The summary of RR and TT with preprocessing, PCA or HOG based on the K-NN

To conclude, although the combination of preprocessing and PCA offers the recognition rate of K-NN to over 99%, an interesting finding is that the accuracy achieved by using the preprocessing and HOG feature descriptor is lower than with the raw data. Furthermore, the training time after using HOG is about 13.8 seconds longer than other combinations. So, this

implies that the combination of preprocessing, HOG and K-NN can not address the question raised by this research.

4.6 The Combination of Preprocessing, HOG and RF

In the field of pattern recognition, researchers have paid more attention to multi-classifier systems in recent years, especially Bagging, Boosting. The concept of RF was introduced based on the bagging principle by Breiman in 2001. In the Forest-RI algorithm, not all the features contribute to the recognition rate. In fact some features may degrade the results. Bernard, Adam and Heutte (2007) researched a conventional feature extraction technique based on a greyscale multi-resolution pyramid to find out the effect of the parameter values on the performance of the RF. They have experimented on the MNIST handwritten digital database and reached an accuracy level greater than 93%.

The combination of HOG and K-NN produced a very interesting result, namely that its accuracy is lower than the original data. Therefore, in this part, preprocessing, HOG and RF were explored as the last combination for this experiment. The following flow chart of this combination is depicted in Fig.4.12.

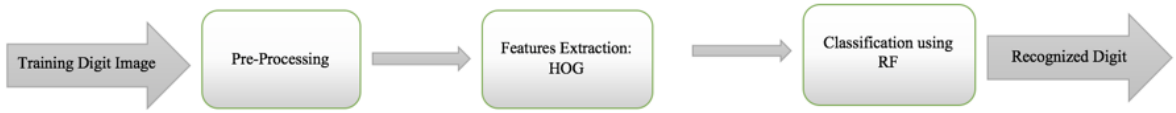


Figure 4.12: The flow chart of the preprocessing, HOG and RF

The calculation of vertical and horizontal gradients has been done by using the **cv2.Sobel ()** function to calculate the HOG descriptor with the same functions as mentioned in the last section. Then, the gradient magnitude and direction can be computed by applying the function **cv2.cartToPolar ()**. Next, the image gradients were normalized by the function **hog ()**. To create the RF model, this function **RandomForestClassifier ()** was adopted. In addition, the parameter is set as **n_estimators='warn', criterion='gini', min_samples_split=2, min_samples_leaf=1** etc. Table 4.6 describes the comparison of RR and TT with preprocessing, PCA or HOG based on the RF.

	Raw Data	Preprocessing	Pre + PCA	Pre + HOG
RR	93.56%	95.60%	90.89%	92.6%
TT	2.2 s	2.7s	4.8 s	2.7 s

Table 4.6: The comparison of RR and TT with preprocessing, PCA or HOG based on the RF

Similar to the K-NN model, the RF algorithm using the preprocessing and hog combination did not achieve a high level of accuracy which is lower than the classification based on raw data. Overall, the performance of RF and SVM is the same, as well as the recognition rate, is which around 95%. Conversely, the training time of RF is relatively shorter. However, this combination still cannot achieve a handwritten digit recognition rate of more than 99%.

4.7 Summary

This chapter has revisited the data sets adopted in this experiment at the start, and the techniques of slant correction and elastic distortion in the preprocessing stage have been described in depth. Then, according to the previous research on different types of preprocessing, feature extraction and classification technology, five combinations were selected for further exploration:

1. Preprocessing +CNN
2. Preprocessing + PCA + K-NN
3. Preprocessing + PCA+ SVM
4. Preprocessing + HOG+ K-NN
5. Preprocessing + HOG+ RF

Furthermore, the implementation details of the five combinations such as the selection of the package, and the adjustment of the parameters have been analyzed. Finally, the result of this were compared and it was determined which ones can answer the question of this study, namely, whether it is possible to improve the accuracy of handwritten digit recognition by more than 99%. Table 4.7 indicates the impact of using preprocessing, PCA or HOG on the accuracy of handwritten digit recognition based on four classifiers.

	Raw Data	Preprocessing	Pre + PCA	Pre + HOG
CNN	98.75%	99.44%	_____	_____
K-NN	96.68%	98.14%	99.17%	95.39%
RF	93.56%	95.60%	90.89%	92.6%
SVM	91.17%	94.54%	94.90%	93.01%

Table 4.7: The summary of handwritten digit RR based on four classifier models

As can be seen from Table 4.7, the combined performances of the CNN and K-NN models are higher than SVM and RF in the field of handwritten digit recognition. Furthermore, the performances of two combinations have successfully answered the challenge of this study and improved the accuracy to over 99%, respectively Preprocessing + CNN and Preprocessing + PCA + K-NN. Notably, the combination of pre-processing and CNN reached the highest efficiency of 99.44% throughout the experiment. However, an automatic extraction method LeNet5 by CNN can detect features directly from the original image, PCA and HOG technologies were not explored based on the CNN model. In contrast, most application SVM and RF models had recognition rates below 95% in general. An interesting finding is that the accuracy achieved from using the HOG feature descriptor based on K-NN and RF was lower than the raw data. One of the reasons may be that the two classification algorithms are not sensitive to the alignment of the intensity gradient of the image, and that will be the future research direction.

The next chapter will provide an analysis of the experimental results as depicted in Table 4.7. Moreover, the model evaluation and test will also be illustrated in detail. Finally, a comparison and discussion will be provided in line with the literature review and state what is new in the own work.

5 ANALYSIS, EVALUATION AND DISCUSSION

5.1 Introduction

In these experiments, five combinations were studied, and the performance of each combination was measured by using the recognition rate as an evaluation metric. The recognition rate is the accuracy of the classifier for image recognition. Table 4.7 displays a summary of the results in light of the basic techniques, pre-processing and two feature extraction techniques, namely PCA and the HOG descriptor.

This chapter will present a detailed analysis in each section based on four classification algorithms, namely CNN, K-NN, RF and SVM. In particular, K-fold cross-validation with k=10 was applied to training sets to avoid over-fitting due to large parameter values. Further, in order to evaluate the model, the confusion matrix, error rate, classification reports and some errors which are the difference between predicted labels and correct labels will be illustrated. Finally, handwritten digits never seen by the systems will be used to test the specific models and show the anticipated figures.

5.2 Initial Experiment

Firstly, the performance of the four classification models such as CNN, K-NN, RF and SVM was evaluated on the original MNIST data. Table 5.1 indicates a comparison of the four classifiers regarding error rates (ER) and training time (TT). It can be seen from this table that the ER of CNN in this experiment is the lowest at 1.25% compared with the other three classifiers, and the ER of SVM is up to 9%. On the other hand, the training time spent by CNN is 6,000 times higher than RF by 3.5 hours. The reason for this may be that CNN is well-suited for extensively used digital databases and images since they can recognize patterns with numerous features, namely pixels in 2D and characters in 1D. In contrast, RF, KNN and SVM demonstrate superiority in other kinds of challenges: mainly in the space of relatively few various features such as tens or hundreds. They will defeat CNN's easily there.

Cross-validation is usually not applied for evaluating deep learning models due to the high computational expense. However, when the total of data available is limited, the k-fold cross-validation estimator results in lower variance than a single estimator. In this project, k-fold cross-validation showing how k=10 was adopted when classifying data because it supplies a reliable estimate of the performance of a classification model on the unseen data. One clear

example is depicted in Table 5.2, 10-fold cross-validation was performed for the RF model and the individual of accuracy individually are shown below.

	CNN	K-NN	RF	SVM
ER	1.25%	3.3%	6.5%	8.9%
TT	3.5h	5.9 s	2.2 s	34.5s

Table 5.1: The comparison of four classifiers in terms of ER and TT

k-fold	1	2	3	4	5	6	7	8	9	10
Accuracy	93.3	93.6	93.1	93.6	94.3	93.3	93.1	93.1	93.1	93.2 (%)

Table 5.2: The accuracy of the 10-fold cross-validation based on RF algorithm

Confusion matrices focus on the predictive ability of a model instead of how fast the model performs the classification. One of the strengths of using this performance evaluation method is that a data mining analyzer may easily notice if the model is confusing multiple or two classes. As can be seen from Fig.5.1, CNN performs considerably well on the digits with only a few errors and the size of the validation database includes 10000 images. Nevertheless, it seems that the CNN presents some little problems with images of the number '9'. They are misclassified as '1' or '4'. According to the handwriting habits, it is difficult to capture the diversity between '1', '4' and '9' when the curves are smooth.

The classification report is often used to check the quality of classification model predictions. Fig. 5.2 show how the primary classification metrics consist of a precision, recall and f1-score on each class basis. These are defined as referenced below:

1. True Negative(TN): both case and predicted were negative
2. True Positive(TP): both case and predicted were positive
3. False Negative(FN): the case was positive but predicted negative
4. False Positive(FP): the case was negative but predicted positive

Precision represents the accuracy of positive predictions. The question that this metric respond is of all numbers that labeled as ‘0’, how many actually ‘0’ names there? High precision involves the low FP rate. Besides, recall is the rate of positives that were correctly identified. The F1 score is a weighted average of precision and recalls such that the best score is 1 and the worst is 0.

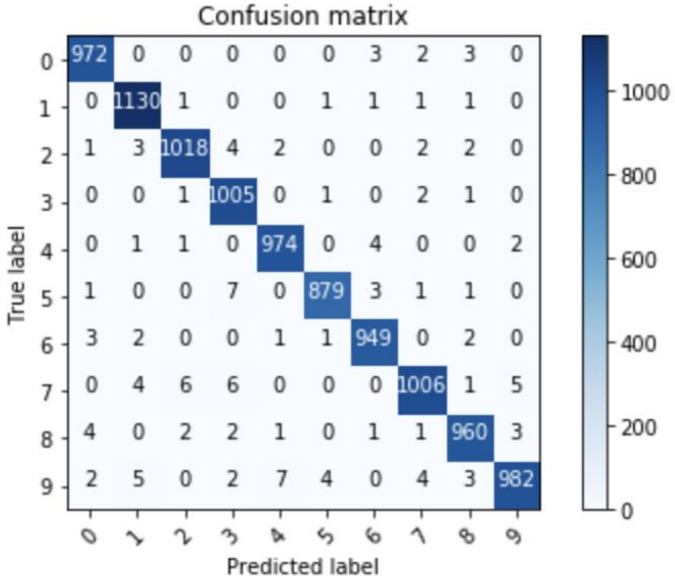


Figure 5.1: The confusion matrix for CNN

The final classification reports for the K-NN and SVM model are described in Fig.5.2. Regardless of which classification indicator, the classification performance of K-NN is higher than that of SVM classifier. In particular, the precision of the number ‘8’ reached 99% in the K-NN classification report, while it obtained only 88% for the SVM model.

	precision	recall	f1-score	support
0	0.97	0.99	0.98	1015
1	0.96	0.99	0.98	1190
2	0.98	0.96	0.97	1077
3	0.96	0.97	0.96	1070
4	0.98	0.96	0.97	1034
5	0.95	0.97	0.96	930
6	0.97	0.99	0.98	1044
7	0.96	0.97	0.96	1129
8	0.99	0.91	0.95	995
9	0.95	0.95	0.95	1016
avg / total	0.97	0.97	0.97	10500

	precision	recall	f1-score	support
0	0.95	0.96	0.96	1015
1	0.96	0.97	0.96	1190
2	0.91	0.90	0.90	1077
3	0.89	0.88	0.88	1070
4	0.91	0.91	0.91	1034
5	0.86	0.87	0.86	930
6	0.93	0.96	0.95	1044
7	0.93	0.92	0.93	1129
8	0.88	0.85	0.87	995
9	0.88	0.87	0.87	1016
avg / total	0.91	0.91	0.91	10500

Figure 5.2: The final classification reports for K-NN (left) and SVM (right) model

Given the high accuracy implemented by CNN model, this was the priority proposed algorithm for this project. Nevertheless, the CNN model can't achieve a recognition rate of more than 99% when it was explored in raw data, and the other three classification models presented the lower performance in this experiment. Therefore, image preprocessing and two feature extraction techniques such as PCA and HOG were applied to address the question raised in this research.

5.3 Experiments with Pre-processing Techniques

In this part of the experiment, an image preprocessing technique was adopted to reduce the error rate. Then, the performance of the four classification models was evaluated on the preprocessed data. Table 5.3 demonstrates the results of four classifiers with the pre-processing techniques. Compared with Table.10, the ER of the four models decreased dramatically, especially CNN was less than 1%. This demonstrates that image preprocessing technology is very significant for this research; it can improve the accuracy of the CNN model by more than 99%. Admittedly, the training time of the four models has increased, and it may be that the preprocessed data adds a burden to the model identification procedure. Even though the ER of the K-NN model was close to 1%, the use of preprocessing alone does not enable it to solve this research question.

	CNN	K-NN	RF	SVM
ER	0.56%	1.25%	4.4%	5.6%
TT	4h	4.6 s	2.7 s	87.7s

Table 5.3: The ER and TT of four classifiers using Pre-processing techniques

The loss and accuracy curves for training and validation from CNN are introduced in Fig.5.3. Since the implementation of the CNN algorithm is too complicated and the training time is exceptionally long, only epoch=30 (4h) was set in this experiment. An epoch is total images are processed for a time individually of forward and backward through the neural network only once. The model achieved nearly 99% (98.75%) accuracy on the validation set after three epochs. It can be seen from the figure below that the verification accuracy is almost higher

than the training accuracy during the training period. That implies that there is no over-fitting in this model.

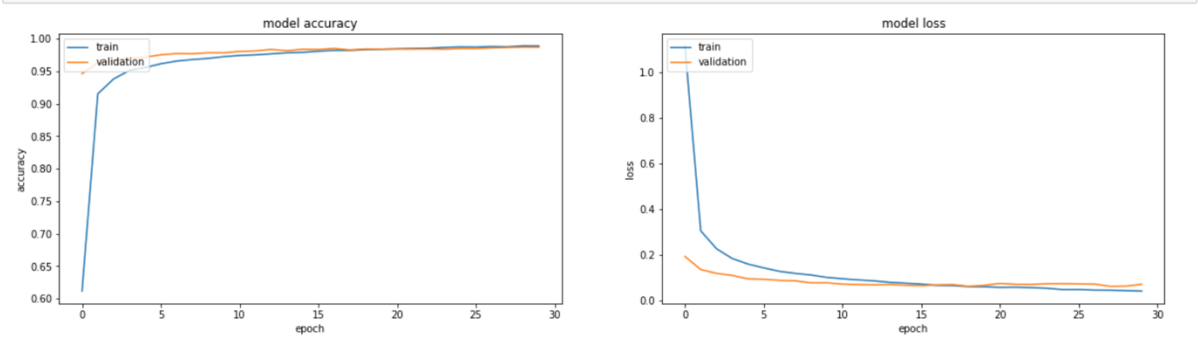


Figure 5.3: The loss and accuracy curves for training and validation from CNN

The two confusion matrices are compared in Fig.5.1 and Fig.5.4 respectively. A considerable part of the previous digitally identified errors was corrected after image preprocessing. For instance, the number '1' that was previously misclassified into the number '9' is successfully recognized after being preprocessed.

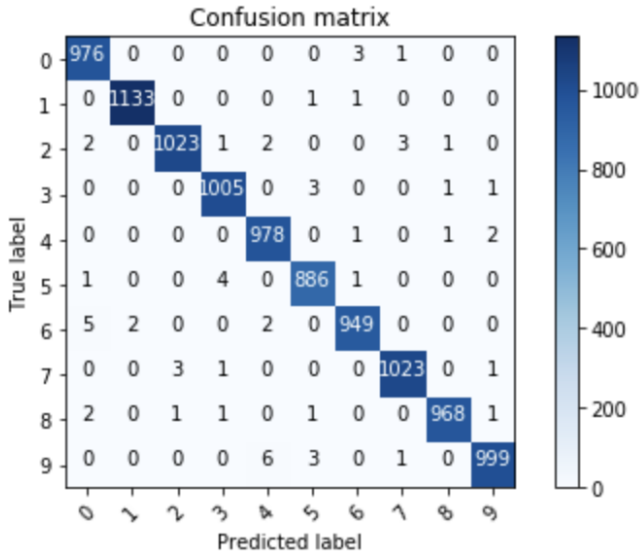


Figure 5.4: The confusion matrix for CNN applying Pre-processing

For those six error cases that are shown in Fig.5.5 this CNN model is not surprising because some of these errors can also be caused by human handwriting habits, especially for one digit '3' which is very close to digit '5'. Another representative example is the number '0', and a considerable number of people recognize it as the number '6'.

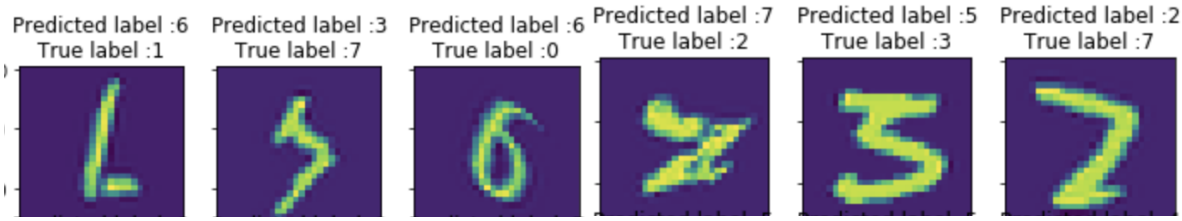


Figure 5.5: Some error results recognized by CNN model using Pre-processing

As shown in Table 5.4 the 10-fold cross-validation method was still adopted to evaluate the RF model. Compared with Table 5.2, the overall recognition rate of the RF model was increased by nearly 2% after pre-processing. Furthermore, Fig.5.6 shows the classification reports of the K-NN and SVM models after preprocessing. Although the classification performance of K-NN is still higher than that of the SVM classifier, all of the indicators of SVM have improved rapidly. The F1 scores for each class based on K-NN are close to 1, which represents an excellent performance of this model.

k-fold	1	2	3	4	5	6	7	8	9	10
Accuracy	95.1	94.6	94.4	94.3	94.4	95.4	95.1	94.7	94.7	95.2 (%)

Table 5.4: The accuracy of the 10-fold cross-validation based on RF using Pre-processing

	precision	recall	f1-score	support
0	0.98	0.99	0.99	1240
1	0.99	1.00	0.99	1405
2	0.98	0.98	0.98	1253
3	0.98	0.97	0.98	1305
4	0.99	0.98	0.98	1222
5	0.98	0.97	0.98	1139
6	0.98	0.99	0.99	1241
7	0.98	0.98	0.98	1320
8	0.98	0.96	0.97	1219
9	0.97	0.97	0.97	1256
avg / total	0.98	0.98	0.98	12600

	precision	recall	f1-score	support
0	0.97	0.97	0.97	1240
1	0.98	0.99	0.98	1405
2	0.94	0.94	0.94	1253
3	0.95	0.92	0.93	1305
4	0.94	0.96	0.95	1222
5	0.92	0.93	0.92	1139
6	0.96	0.97	0.97	1241
7	0.95	0.94	0.95	1320
8	0.91	0.93	0.92	1219
9	0.92	0.91	0.92	1256
avg / total	0.95	0.95	0.95	12600

Figure 5.6 The classification reports for K-NN (left) and SVM (right) using Pre-processing

The error rate of the four models was significantly reduced after using the pre-processing technique, especially the CNN model which increased the accuracy to 99.44%. Nevertheless,

the combination of this experiment with PCA or HOG is still necessary for the other three models such as K-NN in order to complete the research question.

5.4 Experiments with Pre-processing Plus PCA

In this section, the combination of the image preprocessing and PCA was applied to three classification algorithms to reduce the error rate. Since an automatic extraction method LeNet5 by CNN can detect features directly from the original image, PCA and HOG technologies were not explored based on the CNN model.

The error rate and training time of three classifiers applying preprocessing and PCA are indicated in Table 5.5. A satisfactory finding is that the ER of the K-NN model has dropped to 0.87% compared with the results in Table 5.3. This is the second combination that can improve the handwritten digit recognition rate by more than 99%. One of the reasons may be that PCA assists in the specification of training features before linear regression, which is highly desirable for sparse data sets. Additionally, the K-NN model asserts the average of the objective value of the nearest neighbors for regression problems. On the other hand, the TT of SVM was much shortened and ER was also reduced by 0.5%. However, the adoption of PCA technology has increased both the ER and TT of the RF model several times over the initial experiment.

	CNN	K-NN	RF	SVM
ER	——	0.83%	9.2%	5.1%
TT	——	0.9 s	4.8 s	20.8s

Table 5.5: The ER and TT of four classifiers using Pre-processing and PCA

Table 5.6 describes the result of the 10-fold cross-validation based on the RF model by using Pre-processing and PCA. Compared with the results in Table 5.4, the accuracy rate of the RF model was declined by nearly 4%. That could be caused by PCA assisting to standardize training features before linear regression, and RF itself has performed an extraordinary

regularization without assuming linearity. Hence, PCA before RF may not offer great benefit if any.

k-fold	1	2	3	4	5	6	7	8	9	10
Accuracy	90.1	90.3	90.0	90.5	90.7	90.9	90.0	90.1	90.4	90.6 (%)

Table 5.6: The10-fold cross-validation based on RF using Pre-processing and PCA

The line chart of the accuracy of the K-NN model is introduced in Fig.5.7. All odd numbers within 30 are evaluated as the value of k. The K-NN model achieved the highest accuracy of 99.17% on validation data when k=3. A K-NN model with k = 1 usually leads to over-fitting in most cases, and this is quite sensitive to the sort of distortions such as noise, outliers, missing data, and so on.

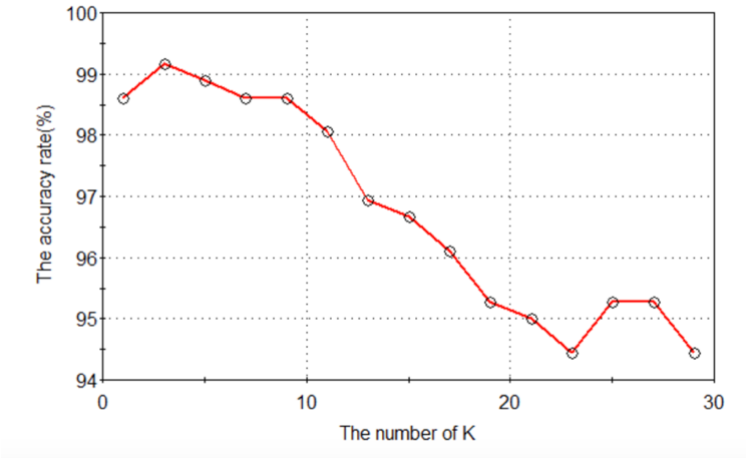


Figure 5.7: The effect of k on the accuracy of the K-NN model

	precision	recall	f1-score	support
0	1.00	1.00	1.00	33
1	0.97	1.00	0.98	28
2	1.00	1.00	1.00	33
3	1.00	1.00	1.00	34
4	0.98	1.00	0.99	46
5	0.98	0.98	0.98	47
6	1.00	1.00	1.00	35
7	1.00	0.97	0.99	34
8	1.00	0.97	0.98	30
9	0.95	0.95	0.95	40
avg / total	0.99	0.99	0.99	360

	precision	recall	f1-score	support
0	0.97	0.98	0.97	1240
1	0.98	0.99	0.99	1405
2	0.94	0.94	0.94	1253
3	0.95	0.92	0.94	1305
4	0.94	0.96	0.95	1222
5	0.92	0.93	0.93	1139
6	0.96	0.97	0.97	1241
7	0.95	0.95	0.95	1320
8	0.92	0.93	0.92	1219
9	0.94	0.91	0.92	1256
avg / total	0.95	0.95	0.95	12600

Figure 5.8: The classification reports for K-NN (left) and SVM (right) applying Pre-processing and PCA

Fig.5.8 displays the classification reports for K-NN and SVM applying preprocessing and PCA. In the process of exploring the K-NN model, 30% of the training data were allocated to from validation set, while the remaining 70% were reserved as the training data. It can be seen from the following figure that all of the indicators of K-NN have reached 99%, where the digits 0, 2, 6,7 and 8 were classified correctly at a rate of 100%. In contrast, the accuracy of SVM was not significantly improved, still around 95%.

5.5 Experiments with Pre-processing Plus HOG

In this part of the experiment, the image pixels and dimensions were changed from 28x28 to 36 after HOG processing. The error rate and training time of three classifiers applying preprocessing and HOG are demonstrated in Table 5.7. An interesting finding is that the error rates of the three models are higher than those after the pre-processing technique was used as compared with Table 5.3. The ER and TT of the KNN and RF models based on this combination are higher than the initial experiment.

	CNN	K-NN	RF	SVM
ER	——	4.6%	7.4%	6.9%
TT	——	13.8 s	2.7 s	1.5s

Table 5.7: The ER and TT of four classifiers using Pre-processing and HOG

HOG calculates the edge gradient of an entire image and detects the orientation of each pixel. The reason why the HOG technology did not work as well played in this experiment may be because the single HOG vector that extracted from an image does not contain accurate feature information, and is passed to the machine learning algorithms such as K-NN and RF. In particular, RF selects an arbitrary feature set in the feature vector at the node of each tree, and then some random functions and linear combinations are generated and trained to search for the best linear combination of feature variables. Additionally, the K-NN method is already good at solving complex classification problems with irregular decision boundaries. Consequently, the HOG feature vector is not as capable when used along side K-NN and RF.

It is recommended to predict some handwritten digits never seen by the systems using the classifier such as CNN to verify the performance of the model again. Fig.5.9 displays the recognition results consisting of the bounding box and predicted digits on the input image. It can be observed that the recognition result from the handwritten numbers is very satisfactory, which shows that CNN is delivering a high recognition ability in this study.

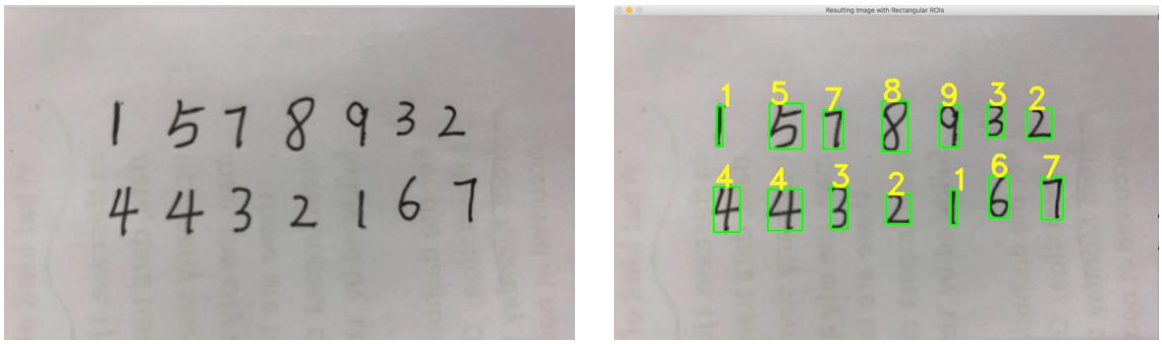


Figure 5.9: The recognition results of some new handwritten digits

Compared with other experiments in the literature(REFS), the focus of this experiment is to explore whether distinct techniques such as image pre-processing, PCA and HOG can improve accuracy by more than 99%. According to the above evaluation and analysis, the two innovative combinations can address the proposed research question, namely preprocessing +CNN and Pre-processing + PCA + K-NN. However, the HOG feature descriptor has not been applied to this handwritten digit recognition experiment effectively.

5.6 Summary

In summary, the results of four experiments have been analyzed. Each modification produced changes in the results mostly improved accuracy and widely varying performance times. The original research question was whether the accuracy could be improved and the answer has been confirmed as yes. So, the recognition rates of the classifiers were assessed to reject the null hypothesis.

This chapter has delivered an analysis of the experimental results in line with the pre-processing and two feature extraction techniques such as PCA and the HOG descriptor at first. Then, a detailed report in each section based on four classification algorithms consist of CNN, K-NN, RF, and SVM was presented. In addition, some evaluation techniques such as

confusion matrices, error rates, classification reports and error cases were adopted and illustrated. Finally, the chapter has displayed the discussion and comparison in light of the literature review and stated what is new in the present work.

Based on the above analysis and discussion, it can be concluded that the two innovative combinations can address the proposed research question, namely preprocessing + CNN and preprocessing + PCA + K-NN. Also, image preprocessing technology plays an indispensable role in handwritten digit recognition and PCA technology also assisted K-NN to achieve the purpose. However, the HOG feature descriptor be effectively applied to this handwritten digit recognition experiment.

6 CONCLUSION

This is the last chapter of the research and it that will present a short account of the experiments' results and stress what is new in the current study, including the problems that were addressed, and the limitations of the study. This section will also introduce some suggestions for future research.

6.1 Research Overview

This study attempted to recognize the handwritten digits by using tools from Machine Learning to train the classifier. Also, the use of techniques in Computer Vision was explored to investigate the effect of selection image preprocessing, feature extraction and classifiers on the overall accuracy. The dataset used for the experiment is MNIST dataset originally constituted of 60,000 training, and 10,000 testing images which are 28 x 28 grayscale (0 - 255) labeled and bitmap format. It is a brilliant database for machine learning and characters recognition methods while taking minimal efforts in preprocessing and formatting.

According to the literature analysis of the field of character recognition, there are some research studies which have made some achievements. For instance, Hochuli et al. (2018) used the CNN classifier to perform experiments on two public databases consisting of Touching Pairs Dataset and NIST SD19, as well as highlighting the proposed method by achieving a 97% recognition accuracy. Recently, Mahto, Bahtia, and Sharma (2015) applied a linear K-NN to classify Gurmukhi handwritten characters with a maximum accuracy of 98.06%. Moreover, Bernard, Adam, and Heutte (2007) experimented with the Forest-RI algorithm on the MNIST handwritten digital database, and the accuracy of handwritten digit recognition reached over 93%. Also, a 2017 paper by Phangtriastu, Harefa, and Tanoto achieved the highest accuracy of 94.43% by using the SVM classifier. Overall, the four classifiers mentioned above have been well verified by previous researchers and got good results. Consequently, CNN, SVM, K-NN, and RF were applied and compared in this experiment to determine which classifier delivers the highest performance.

Compared with other research, this study focused on exploring which image preprocessing and feature extraction techniques based on OCR can work for improving the accuracy of classification models by more than 99%. In the initial experiment, the CNN algorithm won with a recognition accuracy of 98.75%, followed by K-NN with 96.68%. The performance of

RF and SVM in this experiment is not outstanding because they are not good at pattern recognition, while they demonstrate superiority in other kinds of challenges: mainly in the space of relatively few different features such as tens or hundreds. After that, image preprocessing techniques (slant correction, sharpening and elastic deformation) and feature selection techniques (PCA and HOG) were applied to the experiment. Finally, CNN based on image preprocessing, and K-NN based on the combination of image preprocessing and PCA achieved the goal of successfully improving the accuracy to over 99%. In particular, slant correction played a significant role at the image preprocessing stage and the HOG feature descriptors did not perform well in image recognition and object detection.

Four experimental results were analyzed and evaluated by a series of tools such as confusion matrices, k-fold cross-validation, error rates, and classification reports. Each modification produced changes in the results mostly improved accuracy and widely varying performance times. The original objective of this study was could the handwritten digit recognition accuracy is improved by image preprocessing and feature extraction, and it was confirmed to say yes. At the end of the experiment, to verify the performance of the model again, some handwritten digits never seen by the systems were forecasted using the classifier and achieved satisfactory results.

6.2 Problem Definition

The problem of this research study was: *Can OCR use the combination of image pre-processing, feature extraction and classifiers to improve the accuracy of handwritten digit recognition to more than 99%?*

As described in the first chapter, the null hypothesis (H0) of this research is that the accuracy of handwritten digit recognition using the combination of image pre-processing, feature extraction and classifiers based on the OCR would be less than 99% while the alternative hypothesis (H1) is that accuracy will no less than 99%. Based on the evaluation and comparison of the four models, the results have clearly shown the difference in the performance of the classifier. Finally, the recognition rates of the classifiers were assessed to reject the null hypothesis.

OCR is a technique that recognizes printed text in scanned documents. In OCR applications, the function that involves in accuracy and speed of character recognition is critical to overall performance. OCR is a complex process which includes several steps, namely pre-processing, feature selection and classification. Further, some problems occur during the development of the OCR system. Blurred or skewed handwritten digit present difficulties for computer recognition. Another problem is that extracting features with background noise, such as the contrast of fonts and paper (Phangtriastu, Harefa & Tanoto, 2017). Most importantly, extracting appropriate structural features from complex shapes is also a considerable challenge (Pramanik & Bag, 2018).

There is evidence that image preprocessing technology can remove noise, smooth and normalize the input data, which is essential for better differentiation of patterns in the feature space (Karimi et al., 2015). A 2007 paper by Hanmandlu and Murthy proposed the distinct preprocessing techniques specifically slant correction, sharpening and smoothing. For handwritten numbers, one of the first variance in writing ways is caused by slope, which is defined as the slant of the writing trend relative to the vertical line. Also, a new method was devised Hanmandlu and Murthy (2007) to smoothing and removing the virtual slant of distorted numbers. One group of researchers, Simard, Steinkraus and Platt (2003), proposed that if the data are scarce and the distribution to be studied has transform-invariant attributes, applying elastic deformation can generate additional data and even improve performance. This was discussed in detail in Chapter Two.

Because of many classifiers cannot effectively process raw images or data, the purpose of the feature extraction is to reduce the dimension of data and extract significant information (Lauer, Suen & Bloch, 2007). Various feature extraction methods have been advanced for the character recognition system. As described in Chapter Three, PCA can provide a lower-dimensional representation if a multivariate data is visualized as a series of coordinates in the high-dimensional data space. Effectively, it reduces the dimensions of the observed data by eliminating redundancy. Nevertheless, Das (2012) mentioned that only the features extracted by the PCA algorithm are not sufficient to solve the variability of the handwritten digit mode. That is why another feature descriptor HOG was introduced. The HOG descriptor was first proposed by Dalal and Triggs (2005) for human body detection in an image. In recent years, it has become one of the most commonly and successfully used descriptors for computer vision and character recognition for object detection. A 2017 paper by Phangtriastu, Harefa, and

Tanoto achieved the highest accuracy of 94.43% by using SVM classifier with the combination of feature extraction algorithms which are a projection histogram and HOG.

6.3 Experimentation, Evaluation & Results

The process of this experiment has been displayed in detail in chapter four. The dataset adopted was revisited at the beginning to ensure the accuracy of the subsequent analyses. Then, the foremost techniques such as slant correction and elastic distortion in the preprocessing stage were illustrated in depth. Since an automatic extraction method LeNet5 by CNN can detect features directly from the original image, PCA and HOG technologies were not explored based on the CNN model. According to the previous research on different types of preprocessing, feature extraction and classifier technology, five combinations were the focal points for further exploration:

1. Preprocessing +CNN
2. Preprocessing + PCA + K-NN
3. Preprocessing + PCA+ SVM
4. Preprocessing + HOG+ K-NN
5. Preprocessing + HOG+ RF

Moreover, the implementation details of the five combinations such as the selection of the package, and the adjustment of the parameters have been analyzed. The final part of this chapter compared the results of the five combinations and determined which one can achieve an accuracy of more than 99%. As can be seen from the summary of handwritten digit recognition rates based on the four classifier models, the overall performances of CNN and K-NN models are higher than SVM and RF. In addition, two of the five combinations mentioned above have successfully addressed the question and improved the accuracy to over 99%, respectively pre-processing + CNN and preprocessing + PCA + K-NN.

Table 4.7 displays a summary of the results in light of the pre-processing, basic techniques and two feature extraction techniques, namely PCA and the HOG descriptor. In the fifth chapter, each part was analyzed based on four classification algorithms, namely CNN, K-NN, RF and SVM. Meanwhile, K-time cross-validation was applied with k = 10 on the training set to avoid over-fitting due to large parameter values. Then, four types of experiments were explained and the confusion matrices, error rates, and classification reports were evaluated.

1. Initial experiment
2. Experiments with Preprocessing techniques
3. Experiments with Preprocessing plus PCA
4. Experiments with Preprocessing HOG

Overall, based on the evaluation of the results of the four experiments above, each modification produced changes in the results mostly increased accuracy and had widely varying performance times. All of the obtained results show that the performance of most of the classifiers improved after applying the preprocessing technique in handwritten digit recognition. Specifically, the recognition rate of CNN after preprocessing has reached 99.44% from 98.75%.

Given the highest accuracy was implemented by CNN, this was the priority proposed algorithm for this study. In contrast, RF, KNN and SVM display superiority in the space of relatively few complex features such as tens or hundreds. They defeat CNN's performance easily there. On the other hand, preprocessing and PCA technology also assisted K-NN to improve the accuracy from 96.68% to 99.17%, as well as achieving the purpose. However, HOG technology did not perform well in this experiment because the single HOG vector extracted from the image may contain inaccurate feature information and was passed to machine learning algorithms such as K-NN and RF. In summary, OCR can use the combination of image pre-processing, feature extraction and classifiers to improve the accuracy of handwritten digit recognition to more than 99%.

6.4 Contributions and Impact

Intelligent image recognition and analysis is an entertaining research area in Artificial Intelligence, and also significant to a variety of present open research problems. Handwritten digit recognition is a well-researched subarea and vital benchmark task within the field due to its vast practical applications and financial implications. Because of a variety of potential applications such as the reading of postal codes, medical prescription reading, interpreting handwritten addresses, processing bank cheques, credit authentication, social welfare application forms, the forensic analysis of crime evidence which includes a handwritten note, etc., handwriting digital recognition is still an active area of research (Winkler, 1980). As mentioned in the first chapter, in modern times people are paying more attention to using of their personal computer rather than getting excellent handwriting skills. The reason is the

internet and applications are becoming more intelligent. Moreover, the poor quality or illegible handwriting are causing many problems in daily life, which is the main reason for this research.

In recent years, the availability of devices such as ultra-portable digital notebooks and mobile phones with cameras has further broadened the range of applications for the digital recognition of handwriting for multiple personal uses such as captcha images, note-taking and extracting data from filling out forms, etc. (Das *et al.*, 2015). In the financial industry, a satisfactory recognition rate with the highest reliability is demanded. The higher recognition rate on handwritten numbers improved the accuracy for handwritten digits, and the reliability is much more considerable than the accuracy in real-life systems.

Increasing the accuracy of handwritten digit recognition to over 99% is the primary purpose of this study. Therefore, some techniques such as slant correction, elastic deformation, PCA and HOG were explored and analyzed. Finally, OCR applied the combination of image pre-processing, feature extraction and classifiers to improve the accuracy of handwritten digit recognition to more than 99%.

Overall, the implementation and completion of this project have a series of advantages. One clear example is that the system can realize the automatic sorting of millions of emails, thus decreasing the human burden and speeding up the whole process. On the other hand, this application can also improve the accuracy of the review of social welfare application forms as the handwriting ability of some elderly or disabled people is not entirely reliable. In addition, the benefits of applying this system to the financial industry are substantial. It can reduce the enormous economic loss because of the small errors in reading cheques.

6.5 Future Work & Recommendations

In this paper, although the method of addressing the research question was found by training on the MNIST database, there are still some problems that need to be explored and solved in the future. For example, the accuracy of the KNN, SVM and RF models based on the combination of preprocessing and HOG are smaller than the initial experiment. Nevertheless, Ebrahimzadeh et al. (2014) employed the linear SVM as the classifier, and the HOG feature descriptor on the MNIST database and a 97.25% accuracy rate was obtained. So, the causes of these problems mentioned above should be analyzed and found to be resolved in the future.

There are also some natural expansions to this research that would assist extend and reinforcing the results. The benchmark database of MNIST was developed for this work, and it is an excellent database for machine learning and pattern recognition methods while making minimal efforts in preprocessing and formatting. However, not all handwritten digit sets are normalized in size, or centered and stored sequentially as 28x28 pixel images in grayscale in the actual cases. Hence, it would be necessary to add similar experiments with distinct databases regarding the features array dimension and various language scripts such as Chinese, Arabic, French, etc.

The complex recognition problem associated with handwriting is an interesting topic for future research areas. For instance, when some anonymous pieces of handwritten character are found at a crime site, and it is possible to automatically identify that the writer may be a "left-handed man," that would reduce the set of suspects to be investigated. In general, these classification problems are extremely complex, since it is quite hard to detect which handwriting features correctly characterize each involved class (Morera *et al.*, 2018). One clear example of this happens in the classification of gender. Even though the feminine writing is more circular and uniform than the masculine one, there are some examples which masculine writing may exist with a "feminine" appearance. This could be another exact topic in the field of handwritten digit recognition for future work.

BIBLIOGRAPHY

Bernard, S., Adam, S., & Heutte, L. (2007). Using Random Forests for Handwritten Digit Recognition. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*,1043-1047. doi:10.1109/icdar.2007.4377074

Biswas, M., Islam, R., Shom, G. K., Shopon, M., Mohammed, N., Momen, S., & Abedin, A. (2017). BanglaLekha-Isolated: A multi-purpose comprehensive dataset of Handwritten Bangla Isolated characters. *Data in Brief,12*, 103-107. doi:10.1016/j.dib.2017.03.035

Cecotti, H., & Belaid, A. (2005). Rejection strategy for convolutional neural network by adaptive topology applied to handwritten digits recognition. *Eighth International Conference on Document Analysis and Recognition (ICDAR05)*,765-769. doi:10.1109/icdar.2005.200

Cecotti, H. (2016). Active graph based semi-supervised learning using image matching: Application to handwritten digit recognition. *Pattern Recognition Letters,73*, 76-82. doi:10.1016/j.patrec.2016.01.016

Cireşan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). Deep, Big, Simple Neural Nets for Handwritten Digit Recognition. *Neural Computation,22*(12), 3207-3220. doi:10.1162/neco_a_00052

Dalal, N., & Triggs, B. (n.d.). Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05)*,886-893. doi:10.1109/cvpr.2005.177

Das, N., Sarkar, R., Basu, S., Saha, P. K., Kundu, M., & Nasipuri, M. (2015). Handwritten Bangla character recognition using a soft computing paradigm embedded in two pass approach. *Pattern Recognition,48*(6), 2054-2071. doi:10.1016/j.patcog.2014.12.011

Duerr, B., Haettich, W., Tropf, H., & Winkler, G. (1980). A combination of statistical and syntactical pattern recognition applied to classification of unconstrained handwritten numerals. *Pattern Recognition,12*(3), 189-199. doi:10.1016/0031-3203(80)90043-6

Dinov, I. D. (2018). Variable/Feature Selection. *Data Science and Predictive Analytics*,557-572. doi:10.1007/978-3-319-72347-1_17

Ebrahimzadeh, Reza, and Mahdi Jampour. "Efficient Handwritten Digit Recognition Based on Histogram of Oriented Gradients and SVM." *International Journal of Computer Applications*, vol. 104, no. 9, 2014, pp. 10–13., doi:10.5120/18229-9167.

Elleuch, M., Maalej, R., & Kherallah, M. (2016). A New Design Based-SVM of the CNN Classifier Architecture with Dropout for Offline Arabic Handwritten Recognition. *Procedia Computer Science,80*, 1712-1723. doi:10.1016/j.procs.2016.05.512

Hanmandlu, M., & Murthy, O. R. (2007). Fuzzy model based recognition of handwritten numerals. *Pattern Recognition,40*(6), 1840-1854. doi:10.1016/j.patcog.2006.08.014

Hochuli, A., Oliveira, L., Jr, A. B., & Sabourin, R. (2018). Handwritten digit segmentation: Is it still necessary? *Pattern Recognition,78*, 1-11. doi:10.1016/j.patcog.2018.01.004

Karimi, H., Esfahanimehr, A., Mosleh, M., Ghadam, F. M., Salehpour, S., & Medhati, O. (2015). Persian Handwritten Digit Recognition Using Ensemble Classifiers. *Procedia Computer Science,73*, 416-425. doi:10.1016/j.procs.2015.12.018

Kavallieratou, E., Likforman-Sulem, L., & Vasilopoulos, N. (2018). Slant Removal Technique for Historical Document Images. *Journal of Imaging,4*(6), 80. doi:10.3390/jimaging4060080

Keysers, D., Deselaers, T., Gollan, C., & Ney, H. (2007). Deformation Models for Image Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,29*(8), 1422-1435. doi:10.1109/tpami.2007.1153

Khosravi, H., & Kabir, E. (2007). Introducing a very large dataset of handwritten Farsi digits and a study on their varieties. *Pattern Recognition Letters,28*(10), 1133-1141. doi:10.1016/j.patrec.2006.12.022

Lauer, F., Suen, C. Y., & Bloch, G. (2007). A trainable feature extractor for handwritten digit recognition. *Pattern Recognition,40*(6), 1816-1824. doi:10.1016/j.patcog.2006.10.011

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision,60*(2), 91-110. doi:10.1023/b:visi.0000029664.99615.94

Li, F., & Gao, S. (2010). Character Recognition System Based on Back-Propagation Neural Network. *2010 International Conference on Machine Vision and Human-machine Interface,*393-396. doi:10.1109/mvhi.2010.185

Liu, C., Nakashima, K., Sako, H., & Fujisawa, H. (2004). Handwritten digit recognition: Investigation of normalization and feature extraction techniques. *Pattern Recognition,37*(2), 265-279. doi:10.1016/s0031-3203(03)00224-3

Marsico, M. D., Petrosino, A., & Ricciardi, S. (2016). Iris recognition through machine learning techniques: A survey. *Pattern Recognition Letters,82*, 106-115. doi:10.1016/j.patrec.2016.02.001

Mane, D., & Kulkarni, U. (2018). Visualizing and Understanding Customized Convolutional Neural Network for Recognition of Handwritten Marathi Numerals. *Procedia Computer Science,132*, 1123-1137. doi:10.1016/j.procs.2018.05.027

Mohebi, E., & Bagirov, A. (2014). A convolutional recursive modified Self Organizing Map for handwritten digits recognition. *Neural Networks,60*, 104-118. doi:10.1016/j.neunet.2014.08.001

Morera, Ángel, *et al.* "Gender and Handedness Prediction from Offline Handwriting Using Convolutional Neural Networks." *Complexity*, vol. 2018, 2018, pp. 1–14., doi:10.1155/2018/3891624.

Niu, X., & Suen, C. Y. (2012). A novel hybrid CNN–SVM classifier for recognizing handwritten digits. *Pattern Recognition,45*(4), 1318-1325. doi:10.1016/j.patcog.2011.09.021

Phangtriastu, M. R., Harefa, J., & Tanoto, D. F. (2017). Comparison Between Neural Network and Support Vector Machine in Optical Character Recognition. *Procedia Computer Science,116*, 351-357. doi:10.1016/j.procs.2017.10.061

Phon-Amnuaisuk, S. (2013). Applying Non-negative Matrix Factorization to Classify Superimposed Handwritten Digits. *Procedia Computer Science,24*, 261-267. doi:10.1016/j.procs.2013.10.049

Pramanik, R., & Bag, S. (2018). Shape decomposition-based handwritten compound character recognition for Bangla OCR. *Journal of Visual Communication and Image Representation,50*, 123-134. doi:10.1016/j.jvcir.2017.11.016

Rashad, M., Amin, K., Hadhoud, M., & Elkilani, W. (2012). Arabic character recognition using statistical and geometric moment features. *2012 Japan-Egypt Conference on Electronics, Communications and Computers*. doi:10.1109/jec-ecc.2012.6186959

Roy, S., Das, N., Kundu, M., & Nasipuri, M. (2017). Handwritten isolated Bangla compound character recognition: A new benchmark using a novel deep learning approach. *Pattern Recognition Letters,90*, 15-21. doi:10.1016/j.patrec.2017.03.004

Rusu, C. (2012). Fast design of efficient dictionaries for sparse representations. *2012 IEEE International Workshop on Machine Learning for Signal Processing*,1134-1144. doi:10.1109/mlsp.2012.6349795

Sadri, J., Suen, C. Y., & Bui, T. D. (2007). A genetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings. *Pattern Recognition,40*(3), 898-919. doi:10.1016/j.patcog.2006.08.002

Sarkhel, R., Das, N., Das, A., Kundu, M., & Nasipuri, M. (2017). A multi-scale deep quad tree based feature extraction method for the recognition of isolated handwritten characters of popular indic scripts. *Pattern Recognition,71*, 78-93. doi:10.1016/j.patcog.2017.05.022

Sarkhel, R., Das, N., Saha, A. K., & Nasipuri, M. (2016). A multi-objective approach towards cost effective isolated handwritten Bangla character and digit recognition. *Pattern Recognition,58*, 172-189. doi:10.1016/j.patcog.2016.04.010

Sethi, I. K., & Chatterjee, B. (1976). Machine Recognition of Hand-printed Devnagri Numerals. *IETE Journal of Research,22*(8), 532-535. doi:10.1080/03772063.1976.11451104

Suen, C. Y., & Tan, J. (2005). Analysis of errors of handwritten digits made by a multitude of classifiers. *Pattern Recognition Letters,26*(3), 369-379. doi:10.1016/j.patrec.2004.10.019

Surinta, O., Karaaba, M. F., Schomaker, L. R., & Wiering, M. A. (2015). Recognition of handwritten characters using local gradient feature descriptors. *Engineering Applications of Artificial Intelligence,45*, 405-414. doi:10.1016/j.engappai.2015.07.017

Wu, S., Wei, W., & Zhang, L. (2018). Comparison of Machine Learning Algorithms for Handwritten Digit Recognition. *Communications in Computer and Information Science Computational Intelligence and Intelligent Systems*,532-542. doi:10.1007/978-981-13-1651-7_47

Winkler, G. (1980). A combination of statistical and syntactical pattern recognition applied to classification of unconstrained handwritten numerals. *Pattern Recognition,12*(3), 189-199. doi:10.1016/0031-3203(80)90043-6

Zheng, S., Zeng, X., Lin, G., Zhao, C., Feng, Y., Tao, J., . . . Xiong, L. (2016). Sunspot drawings handwritten character recognition method based on deep learning. *New Astronomy,45*, 54-59. doi:10.1016/j.newast.2015.11.001

APPENDIX A

The summary of the classification report in the experiment.

Initial experiment:

	precision	recall	f1-score	support
0	0.95	0.97	0.96	1015
1	0.97	0.98	0.98	1190
2	0.92	0.95	0.93	1077
3	0.91	0.91	0.91	1070
4	0.93	0.94	0.94	1034
5	0.91	0.93	0.92	930
6	0.94	0.96	0.95	1044
7	0.96	0.93	0.95	1129
8	0.93	0.88	0.90	995
9	0.92	0.90	0.91	1016
avg / total	0.94	0.94	0.94	10500

Figure A.1: The classification reports for the RF algorithm

	precision	recall	f1-score	support
0	0.97	0.99	0.98	1015
1	0.96	0.99	0.98	1190
2	0.98	0.96	0.97	1077
3	0.96	0.97	0.96	1070
4	0.98	0.96	0.97	1034
5	0.95	0.97	0.96	930
6	0.97	0.99	0.98	1044
7	0.96	0.97	0.96	1129
8	0.99	0.91	0.95	995
9	0.95	0.95	0.95	1016
avg / total	0.97	0.97	0.97	10500

Figure A.2: The classification reports for K-NN

	precision	recall	f1-score	support
0	0.95	0.96	0.96	1015
1	0.96	0.97	0.96	1190
2	0.91	0.90	0.90	1077
3	0.89	0.88	0.88	1070
4	0.91	0.91	0.91	1034
5	0.86	0.87	0.86	930
6	0.93	0.96	0.95	1044
7	0.93	0.92	0.93	1129
8	0.88	0.85	0.87	995
9	0.88	0.87	0.87	1016
avg / total	0.91	0.91	0.91	10500

Figure A.3: The classification reports for the SVM algorithm

Experiments with Pre-processing techniques:

	precision	recall	f1-score	support
0	0.97	0.98	0.98	1240
1	0.99	0.99	0.99	1405
2	0.93	0.97	0.95	1253
3	0.93	0.93	0.93	1305
4	0.95	0.97	0.96	1222
5	0.94	0.93	0.94	1139
6	0.98	0.97	0.97	1241
7	0.97	0.96	0.96	1320
8	0.95	0.93	0.94	1219
9	0.95	0.92	0.94	1256
avg / total	0.96	0.96	0.96	12600

Figure A.4: The classification reports for the RF algorithm with Pre-processing

	precision	recall	f1-score	support
0	0.98	0.99	0.99	1240
1	0.99	1.00	0.99	1405
2	0.98	0.98	0.98	1253
3	0.98	0.97	0.98	1305
4	0.99	0.98	0.98	1222
5	0.98	0.97	0.98	1139
6	0.98	0.99	0.99	1241
7	0.98	0.98	0.98	1320
8	0.98	0.96	0.97	1219
9	0.97	0.97	0.97	1256
avg / total	0.98	0.98	0.98	12600

Figure A.5: The classification reports for the K-NN algorithm with Pre-processing

	precision	recall	f1-score	support
0	0.97	0.97	0.97	1240
1	0.98	0.99	0.98	1405
2	0.94	0.94	0.94	1253
3	0.95	0.92	0.93	1305
4	0.94	0.96	0.95	1222
5	0.92	0.93	0.92	1139
6	0.96	0.97	0.97	1241
7	0.95	0.94	0.95	1320
8	0.91	0.93	0.92	1219
9	0.92	0.91	0.92	1256
avg / total	0.95	0.95	0.95	12600

Figure A.6: The classification reports for the SVM algorithm with Pre-processing

Experiments with Pre-processing plus PCA:

	precision	recall	f1-score	support
0	0.89	0.97	0.93	1240
1	0.98	0.98	0.98	1405
2	0.87	0.91	0.89	1253
3	0.87	0.90	0.89	1305
4	0.86	0.91	0.89	1222
5	0.89	0.85	0.87	1139
6	0.96	0.94	0.95	1241
7	0.94	0.91	0.93	1320
8	0.90	0.84	0.87	1219
9	0.92	0.86	0.89	1256
avg / total	0.91	0.91	0.91	12600

Figure A.7: The classification reports for the RF algorithm with Pre-processing and PCA

	precision	recall	f1-score	support
0	1.00	1.00	1.00	33
1	0.97	1.00	0.98	28
2	1.00	1.00	1.00	33
3	1.00	1.00	1.00	34
4	0.98	1.00	0.99	46
5	0.98	0.98	0.98	47
6	1.00	1.00	1.00	35
7	1.00	0.97	0.99	34
8	1.00	0.97	0.98	30
9	0.95	0.95	0.95	40
avg / total	0.99	0.99	0.99	360

Figure A.8: The classification reports for the K-NN algorithm with Pre-processing and PCA

	precision	recall	f1-score	support
0	0.97	0.98	0.97	1240
1	0.98	0.99	0.99	1405
2	0.94	0.94	0.94	1253
3	0.95	0.92	0.94	1305
4	0.94	0.96	0.95	1222
5	0.92	0.93	0.93	1139
6	0.96	0.97	0.97	1241
7	0.95	0.95	0.95	1320
8	0.92	0.93	0.92	1219
9	0.94	0.91	0.92	1256
avg / total	0.95	0.95	0.95	12600

Figure A.9: The classification reports for the SVM algorithm with Pre-processing and PCA

Experiments with pre-processing plus HOG:

-	precision	recall	f1-score	support
0.0	0.95	0.97	0.96	803
1.0	0.98	0.96	0.97	936
2.0	0.88	0.93	0.91	858
3.0	0.89	0.91	0.90	881
4.0	0.92	0.92	0.92	867
5.0	0.94	0.94	0.94	734
6.0	0.96	0.95	0.96	815
7.0	0.93	0.90	0.91	850
8.0	0.92	0.90	0.91	814
9.0	0.90	0.88	0.89	842
avg / total	0.93	0.93	0.93	8400

Figure A.10: The classification reports for the RF algorithm with Pre-processing and HOG

	precision	recall	f1-score	support
0.0	0.96	0.99	0.97	818
1.0	0.99	0.98	0.98	947
2.0	0.95	0.95	0.95	866
3.0	0.92	0.95	0.94	834
4.0	0.96	0.93	0.94	818
5.0	0.99	0.96	0.97	769
6.0	0.94	0.98	0.96	817
7.0	0.97	0.92	0.94	879
8.0	0.94	0.96	0.95	836
9.0	0.92	0.93	0.92	816
avg / total	0.95	0.95	0.95	8400

Figure A.11: The classification reports for the K-NN algorithm with Pre-processing and HOG

	precision	recall	f1-score	support
0.0	0.95	0.98	0.97	803
1.0	0.99	0.96	0.97	936
2.0	0.90	0.91	0.91	858
3.0	0.90	0.92	0.91	881
4.0	0.93	0.92	0.92	867
5.0	0.95	0.95	0.95	734
6.0	0.96	0.97	0.96	815
7.0	0.92	0.90	0.91	850
8.0	0.90	0.91	0.91	814
9.0	0.90	0.88	0.89	842
avg / total	0.93	0.93	0.93	8400

Figure A.12: The classification reports for the K-NN algorithm with Pre-processing and HOG