HANDWRITTEN DIGIT RECONGNITION

A Project Report Submitted

In Partial Fulfilment of the Requirements

For the Degree of

MASTER OF COMPUTER APPLICATION

By

SHAMBHAVI MISHRA

University Roll No. 1900290149089

Under the Supervision of

Mr. Ankit Verma

Assistant Professor

KIET Group of Institutions

Submitted to

DEPARTMENT OF COMPUTER APPLICATION

Affiliated to

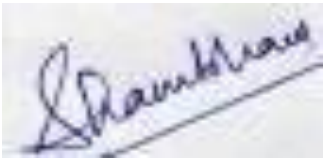
DR. A. P. J ABDUL KALAM TECHNICAL UNIVERSITY

LUCKNOW

JULY, 2021

DECLARATION

I hereby declare that the work presented in this report entitled **"Handwritten Digit Recognition"**, was carried out by US. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.
I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution.
I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.
I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, We shall be fully responsible and answerable.



SHAMBHAVI MISHRA
University Roll No.-1900290149089

CERTIFICATE

Certified that **SHAMBHAVI MISHRA (Univ. Roll No.-1900290149089)** have carried out the project work having **"Handwritten Digit Recognition"** for Master of Computer Application from Dr.A.P.J.Abdul Kalam Technical University (AKTU),Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Shambhavi Mishra (1900290149089)

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Date:

Mr. Ankit Verma
Assistant Professor
Department of Computer Application
KIET Group of Institutions, Ghaziabad

Signature of External Examiner **Signature of Internal Examiner**

Dr. Ajay Kumar Shrivastava
Head, Department of Computer Application
KIET Group of Institutions, Ghaziabad

ABSTRACT

Optical Character Recognition (OCR) is a subfield of Image Processing which is concerned with extracting text from images or scanned documents. In this project, we have chosen to focus on recognizing handwritten digits available in the MNIST database. The challenge in this project is to use basic Image Correlation, also known as Matrix Matching, techniques in order to maximize the accuracy of the handwritten digits recognizer without going through sophisticated techniques like machine learning.

ACKNOWLEDGEMENTS

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor **Mr. Ankit Verma**, for his guidance, help and encouragement throughout our project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava**, Professor and Head, Department of Computer Application, for his insightful comments and administrative help at various occasions.
Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Shambhavi Mishra

Univ. Roll No. 1900290149089

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROJECT DESCRIPTION

It is easy for the human brain to process images and analyze them. When the eye sees a certain image, the brain can easily segment it and recognize its different elements. The brain automatically goes through that process, which involves not only the analysis of this images, but also the comparison of their different characteristics with what it already knows in order to be able to recognize these elements. There is a field in computer science that tries to do the same thing for machines, which is Image Processing.

Image processing is the field that concerns analyzing images so as to extract some useful information from them. This method takes images and converts them into a digital form readable by computers, it applies certain algorithms on them, and results in a better quality images or with some of their characteristics that could be used in order to extract some important information from them.

Image processing is applied in several areas, especially nowadays, and several software has been developed that use this concept. Now we have self-driven cars which can detect other cars and human beings to avoid accidents. Also, some social media applications, like Facebook, can do facial recognition thanks to this technique. Furthermore, some software uses it in order to recognize the characters in some images, which is the concept of optical character recognition that we will be discussing and discovering in this project.

One of the narrow fields of image processing is recognizing characters from an image, which is referred to as Optical Character Recognition (OCR). This method is about reading an image containing one or more characters, or reading a scanned text of typed or handwritten characters and is able to recognize them. A lot of research has been done in this field in order to find optimal techniques with a high accuracy and correctness. The most used algorithms that proved a very high performance are machine learning algorithms like Neural Networks and Support Vector Machine.

One of the main applications of OCR is recognizing handwritten characters. In this project, we will focus on building a mechanism that will recognize handwritten digits. We will be reading images containing handwritten digits extracted from the MNIST database and try to recognize which digit is represented by that image. For that we will use basic Image Correlation techniques, also referred to as Matrix Matching. This approach is based on matrices manipulations, as it reads the images as matrices in which each element is a pixel. It overlaps the image with all the images in the reference set and find the correlation between them in order to be able to determine the digit it represents.

The goal of this project is to apply and manipulate the basic image correlation techniques to build program and keep polishing and enhancing in order to investigate to which extent it can get improved. This would allow us to see how far we can go, in terms of accuracy and performance, but using just the very simple and basic techniques of matrix matching and without going into complicated methods like machine learning

1.2 IMAGE PROCESSING

Image processing is a very wide field within computer science which deals mainly with analyzing images and trying to get some information out of them. The image to be processed is imported then analyzed using some computations, which, by the end, results either in an image with a better quality or some of the characteristics of this image depending on the purpose of this analysis. This is a very wide field within computer science, which also has several other subfields of which Optical Character Recognition that we will be mainly dealing with throughout this project.

1.3 OPTICAL CHARACTER RECOGNITION (OCR) - HISTORY

It is easy for the naked eye to recognize a character when spotted in any document; however, computers cannot identify the characters from an image or scanned document. In order to make this possible, a lot of research has been done, which resulted in the development of several algorithms that made this possible. One of the fields that specialize in character recognition under the light of Image Processing is Optical Character Recognition (OCR).

In Optical Character Recognition, a scanned document or an image is read and segmented in order to be able to decipher the characters it contains .The images are taken and are preprocessed so as to get rid of the noise and have unified colors and shades, then the characters are segmented and recognized one by one, to finally end up with a file containing encoded text containing these characters, which can be easily read by computers.

Optical Character Recognition dates back to the early 1900s, as it was developed in the United States in some reading aids for the blind. In 1914, Emanuel Goldberg was able to implement a machine able to convert characters into “standard telegraph code”. In the 1950s, David Shepard, who was at that time an engineer at the Department of Defense, developed a machine that he named Gismo, which is able to read characters and translate them into machine language. In 1974, Ray Kurzweil decided to develop a machine that would read text for blind and visually impaired people under his company, Kurzweil

Computer Products. There are several software and programs, nowadays, which use OCR in several different applications. In 1996, the United States Postal Services were able to develop a mechanism, HWAI, which recognizes handwritten mail addresses.

1.4 METHODS USED IN OCR

A lot of research has been done in the field of OCR, and still being done, which resulted in the development of several algorithms which enable computers to recognize characters from images or scanned texts. Many of these techniques have attained very high efficiency and a low error rate. However, these algorithms are still being investigated and improved for a better performance.

1.5 MACHINE LEARNING

Machine learning is a field that concerns making programs learn and knows how to behave in different situations using data. One of its applications is Optical Character Recognition.

1.6 ARTIFICIAL NEURAL NETWORK

An Artificial Neural Network (ANN) is a system that mimics the human's biological neural network in the brain. It is an algorithm used for machine learning, which means it uses data to learn how to respond to different inputs. The ANN can be seen as a box, which takes one or more inputs and gives one output. Inside the box, there exist several interconnected nodes. The input is fed into the program, which goes through the several layers and nodes of the ANN and gives an output using a transfer function.

Artificial Neural Networks are used for OCR and have proved a very high accuracy rate. In this case, the ANN would "recognize a character based on its topological features such as shape, symmetry, closed or open areas, and number of pixels". The high accuracy of this kind of algorithms is mainly thanks to its ability of learning from the training set, which would contain characters with similar features.

Some Neural Networks have proven a very high performance. An implementation of the ANN done by Simard, Steinkraus, and Platt has reduced the error rate of recognizing handwritten digits from the MNIST dataset to a percentage as low as 0.7%

1.7 SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is an algorithm that belongs to machine learning as well. SVMs are known as high performance pattern classifiers. While Neural Networks aim at minimizing the training error, SVMs have as goal to minimize the "upper bound of the generalization error". The learning algorithm in this technique is based on classification and regression analysis.

This kind of classifier has been used in the recognition of very complex characters like the Khmer language and has proved a very high performance.

1.8 IMAGE CORRELATION

Image Correlation is a technique used to recognize characters from images. This approach, also referred to as Matrix Matching, uses mathematical computations in order to analyze the images.
By using this technique, the images are read as matrices, where each element represents a pixel, which makes it easier to manipulate them using mathematical approaches. The image to be identified is loaded as a matrix and compared to the images in the reference set. The test image is overlapped with each image in the reference set to be able to see how it matches with each one of them so as to tell which one represents it the most. The decision can be made by seeing the pixels that match and the ones left out from either one of the two images. This technique has many challenges and limitations, as it only overlaps the images and tries to see how much they look alike. By using this method, problems arise when having characters of different sizes, or when one of them is rotated by a certain angle**.**

1.9 FEATURE EXTRACTION

Feature extraction is a technique based on pattern recognition. The main idea of feature extraction is analyzing the images and derives some characteristics from these images that identify each specific element. An example of these characteristics would be the curvatures, the holes, the edges, etc. In the case of digits recognition, these features could be the holes inside the digits (for example for the eight, the six, and maybe the two as well) as well as the angles between some straight lines (for example in the one, the four, and the seven). Whenever an unknown image is to be recognized, its features are compared to these so that it can be classified.

CHAPTER 2

LITERATURE REVIEW

2.1 ABSTRACT

Handwriting recognition has gained a lot of attention in the field of pattern recognition and machine learning due to its application in various fields. Optical Character Recognition (OCR) and Handwritten Character Recognition (HCR) has specific domain to apply. Various techniques have been proposed to for character recognition in handwriting recognition system. Even though, sufficient studies and papers describes the techniques for converting textual content from a paper document into machine readable form. In coming days, character recognition system might serve as a key factor to create a paperless environment by digitizing and processing existing paper documents. This paper presents a detailed review in the field of Handwritten Character Recognition.

2.2 KEYWORDS

Handwritten Character Recognition, Optical Character Recognition

2.3 INTRODUCTION

Character recognition is a fundamental, but most challenging in the field of pattern recognition with large number of useful applications. It has been an intense field of research since the early days of computer science due to it being a natural way of interactions between computers and humans. More precisely Character recognition is the process of detecting and recognizing characters from the input image and converts it into ASCII or other equivalent machine editable form. The technique by which a computer system can recognize characters and other symbols written by hand in natural handwriting is called handwriting recognition system. Handwriting recognition is classified into offline handwriting recognition and online handwriting recognition. If handwriting is scanned and then understood by the computer, it is called offline handwriting recognition. In case, the handwriting is recognized while writing through touch pad using stylus pen, it is called online handwriting recognition. From the classifier perspective, character recognition systems are classified into two main categories i.e. segmentation frees (global) and segmentation based (analytic). The segmentation frees also known as the holistic approach to recognize the character without segmenting it into subunits or characters. Each word is represented as a set of global features, e.g. ascender, loops, cusp, etc. Whereas segmentation based approach; each word/ligature is

segmented into subunits either uniform or non-uniform and subunits are considered independently. Handwritten character processing systems are domain and application specific, like it is not possible to design a generic system which can process all kinds of handwritten scripts and language. Lots of work has been done on European languages and Arabic (Urdu) language. Whereas domestic languages like Hindi, Punjabi, Bangla, Tamil, Gujarati etc. are very less explored due to limited usage. In this paper, the section II describes the basic working principle of character recognition followed by a detailed literature survey.

2.4 WORKING PRINCIPLE

Normally handwritten recognition is divided into six phases which are image acquisition, pre-processing, segmentation, feature extraction, classification and post processing. The block diagram of the basic character recognition is shown in fig1. Post Processing Classification Feature Extraction Segmentation Pre-processing Image Acquisition Figure 1: Block Diagram of Character Recognition S Ayush Purohit et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (1) , 2016, 1-5 www.ijcsit.com 1 A. Image Acquisition Digitized/Digital Image is initially taken as input. The most common of these devices is the electronic tablet or digitizer. These devices use a pen that is digital in nature. Input images for handwritten characters can also be taken by using other methods such as scanners, photographs or by directly writing in the computer by using a stylus.

B. Pre-processing Pre-processing is the basic phase of character recognition and it's crucial for good recognition rate. The main objective of pre-processing steps is to normalize strokes and remove variations that would otherwise complicate recognition and reduce the recognition rate. These variations or distortions include the irregular size of text, missing points during pen movement collections, jitter present in text, left or right bend in handwriting and uneven distances of points from neighbouring positions. Pre-processing includes five common steps, namely, size normalization and centring, interpolating missing points, smoothing, slant correction and resampling of points.

C. Segmentation is done by separation of the individual characters of an image. Generally document is processed in a hierarchical way. At first level lines are segmented using row histogram. From each row, words are extracted using column histogram and finally characters are extracted from words.

D. Feature Extraction The main aim of feature extraction phase is to extract that pattern which is most pertinent for classification. Feature extraction techniques like Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA), Chain Code (CC), Scale Invariant Feature Extraction (SIFT), zoning, Gradient based features, Histogram might be applied to extract the features of individual characters. These features are used to train the system.

E. Classification When input image is presented to HCR system, its features

are extracted and given as an input to the trained classifier like artificial neural network or support vector machine. Classifiers compare the input feature with stored pattern and find out the best matching class for input. F. Post Processing Post-processing refers to the procedure of correcting misclassified results by applying linguistic knowledge. Postprocessing is processing of the output from shape recognition. Language information can increase the accuracy obtained by pure shape recognition. For handwriting input, some shape recognizers yield a single string of characters, while others yield a number of alternatives for each character, often with a measure of confidence for each alternative

An early notable attempt in the area of character recognition research is by Grimsdale in 1959. The origin of a great deal of research work in the early sixties was based on an approach known as analysis-by-synthesis method suggested by Eden in 1968. The great importance of Eden's work was that he formally proved that all handwritten characters are formed by a finite number of schematic features, a point that was implicitly included in previous works. This notion was later used in all methods in syntactic (structural) approaches of character recognition. K. Gaurav, Bhatia P. K. Et al, this paper deals with the various pre-processing techniques involved in the character recognition with different kind of images ranges from a simple handwritten form based documents and documents containing colored and complex background and varied intensities. In this, different preprocessing techniques like skew detection and correction, image enhancement techniques of contrast stretching, binarization, noise removal techniques, normalization and segmentation, morphological processing techniques are discussed. It was concluded that using a single technique for preprocessing, we can't completely process the image. However, even after applying all the said techniques might not possible to achieve the full accuracy in a preprocessing system. Salvador España-Boquera et al, in this paper hybrid Hidden Markov Model (HMM) model is proposed for recognizing unconstrained offline handwritten texts. In this, the structural part of the optical model has been modelled with Markov chains, and a Multilayer Perceptron is used to estimate the emission probabilities. In this paper, different techniques are applied to remove slope and slant from handwritten text and to normalize the size of text images with supervised learning methods. The key features of this recognition system were to develop a system having high accuracy in preprocessing and recognition, which are both based on ANNs. In, a modified quadratic classifier based scheme to recognize the offline handwritten numerals of six popular Indian scripts is proposed. Multilayer perceptron has been used for recognizing Handwritten English characters. The features are extracted from Boundary tracing and their Fourier Descriptors. The character is identified by analysing its shape and comparing its features that distinguish each character. Also an analysis has been carried out to determine the number of hidden layer nodes to achieve high performance of the back propagation network. A

recognition accuracy of 94% has been reported for Handwritten English characters with less training time. In [9], diagonal feature extraction has been proposed for offline character recognition. It is based on ANN model. Two approaches using 54 features and 69 features are chosen to Ayush Purohit et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (1) , 2016, 1-5 www.ijcsit.com 2 build this Neural Network recognition system. To compare the recognition efficiency of the proposed diagonal method of feature extraction, the neural network recognition system is trained using the horizontal and vertical feature extraction methods. It is found that the diagonal method of feature extraction yields the recognition accuracy of 97.8 % for 54 features and 98.5% for 69 features. A. Brakensiek, J. Rottland, A. Kosmala, J. Rigoll et al, in this paper a system for off-line cursive handwriting recognition is described which is based on Hidden Markov Models (HMM) using discrete and hybrid modelling techniques. Handwriting recognition experiments using a discrete and two different hybrid approaches, which consist of a discrete and semi-continuous structures, are compared. A segmentation free approach is considered to develop the system. It is found that the recognition rate performance can be improved of a hybrid modelling technique for HMMs, which depends on a neural vector quantizer (hybrid MMI), compared to discrete and hybrid HMMs, based on tired mixture structure (hybrid - TP), which may be caused by a relative small data set. R. Bajaj, L. Dey, S. Chaudhari et al , employed three different kinds of features, namely, the density features, moment features and descriptive component features for classification of Devanagari Numerals. They proposed multi classifier connectionist architecture for increasing the recognition reliability and they obtained 89.6% accuracy for handwritten Devanagari numerals. Sandhya Arora in, used four feature extraction techniques namely, intersection, shadow feature, chain code histogram and straight line fitting features. Shadow features are computed globally for character image while intersection features, chain code histogram features and line fitting features are computed by dividing the character image into different segments. On experimentation with a dataset of 4900 samples the overall recognition rate observed was 92.80% for Devanagari characters. Mohammed Z. Khedher, Gheith A. Abandah, and Ahmed M. Al Khawaldeh et al, this paper describes that Recognition of characters greatly depends upon the features used. Several features of the handwritten Arabic characters are selected and discussed. An off-line recognition system based on the selected features was built. The system was trained and tested with realistic samples of handwritten Arabic characters. Evaluation of the importance and accuracy of the selected features is made. The recognition based on the selected features give average accuracies of 88% and 70% for the numbers and letters, respectively. Further improvements are achieved by using feature weights based on insights gained from the accuracies of individual features. Sushree Sangita Patnaik and

Anup Kumar Panda May 2011 et al, this paper proposes the implementation of particle swarm optimization (PSO) and bacterial foraging optimization (BFO) algorithms which are intended for optimal harmonic compensation by minimizing the undesirable losses occurring inside the APF itself. The efficiency and effectiveness of the implementation of two approaches are compared for two different conditions of supply. The total harmonic distortion (THD) in the source current which is a measure of APF performance is reduced drastically to nearly 1% by employing BFO. The results demonstrate that BFO outperforms the conventional and PSO based approaches by ensuring excellent functionality of APF and quick prevail over harmonics in the source current even under unbalanced supply. In literature, T. Som have discussed fuzzy membership function based approach for HCR. Character images are normalized to 20 X 10 pixels. Average image (fused image) is formed from 10 images of each character. Bonding box around character is determined by using vertical and horizontal projection of character. After cropping image to bounding box, it is resized to 10 X 10 pixels size. After that, thing is performed and thinned image is placed in one by one raw of 100 X 100 canvas. Similarity score of test image is matched with fusion image and characters are classified. In [16], Renata F. P. Neves have proposed SVM based offline handwritten digit recognition. Authors claim that SVM outperforms the Multilayer perceptron classifier. Experiment is carried out on NIST SD19 standard dataset. Advantage of MLP is that it is able to segment non-linearly separable classes. However, MLP can easily fall into a region of local minimum, where the training will stop assuming it has achieved an optimal point in the error surface. Another hindrance is defining the best network architecture to solve the problem, considering the number of layers and the number of perceptron in each hidden layer. Because of these disadvantages, a digit recognizer using the MLP structure may not produce the desired low error rate. G. Pirlo and D. Impedovo in his work on, presented a new class of membership functions, which are called Fuzzymembership functions (FMFs), for zoning-based classification. These FMFs can be easily adapted to the specific characteristics of a classification problem in order to maximize classification performance. In this research, a realcoded genetic algorithm is presented to find, in a single optimization procedure, the optimal FMF, together with the optimal zoning described by Voronoi tessellation. The experimental results, which are carried out in the field of handwritten digit and character recognition, indicate that optimal FMF performs better than other membership functions based on abstract level, ranked-level, and measurement-level weighting models, which can be found in the literature. Ayush Purohit et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (1) , 2016, 1-5 www.ijcsit.com 3 Yoshimasa Kimura presented [18] a work on how to select features for Character Recognition Using Genetic Algorithm. The author proposes a novel method of feature selection for character recognition using

genetic algorithms (GA). The proposed method selects only the genes for which the recognition rate of training samples exceeds than the predetermined threshold as a candidate of the parent gene and adopts a reduction ratio in the number of features used for recognition as the fitness value. Nafiz Arica at al. [19] proposed a method which avoids most of the pre-processing operations, which causes loss of important information. One of the major contributions of the method is to development of a powerful segmentation algorithm. Utilization of the character boundaries, local maxima and minima, slant angle, upper and lower baselines, stroke height and width, and ascenders and descenders improve the search algorithm of the optimal segmentation path, applied on a gray-scale image. This approach decreases the over-segmentation. Another contribution is the use of Hidden Markov Models (HMM) training, not only for the estimation of model parameters, but also for the estimation of some global and feature space parameters. Also, HMM probabilities are used to measure the shape information and rank the candidate character. One dimensional representation of a two dimensional character image increases the power of the HMM shape recognizer. M. Hanmandlu, O.V. Ramana Murthy have presented in their study the recognition of handwritten Hindi and English numerals by representing them in the form of exponential membership functions which serve as a fuzzy model. The recognition is carried out by modifying the exponential membership functions fitted to the fuzzy sets. These fuzzy sets are derived from features consisting of normalized distances obtained using the Box approach. The membership function is modified by two structural parameters that are estimated by optimizing the entropy subject to the attainment of membership function to unity. The overall recognition rate is found to be 95% for Hindi numerals and 98.4% for English numerals. In, a method to construct a handwritten Tamil character by executing a sequence of strokes is proposed. A structure or shape-based representation of a stroke was used in which a stroke was represented as a string of shape features. Using this string representation, an unknown stroke was identified by comparing it with a database of strokes using a flexible string matching procedure. A full character was recognized by identifying all the component strokes.

2.5 CONCLUSION

The paper discusses in detail all advances in the area of handwritten character recognition. The most accurate solution provided in this area directly or indirectly depends upon the quality as well as the nature of the material to be read. Various techniques have been described in this paper for character recognition in handwriting recognition system. A sort comparison is shown between the different methods proposed so far in table 1. From the study done so far, it is analysed that the selection of the classification as well as the feature extraction techniques needs to be proper in order to attain good rate in recognizing the character. Studies in the paper reveals that there is still scope

of enhancing the algorithms as well as enhancing the rate of recognition of characters.

2.6 REFRENCES

[1] Kai Ding, Zhibin Liu, Lianwen Jin, Xinghua Zhu, "A Comparative study of GABOR feature and gradient feature for handwritten chinese character recognition", International Conference on Wavelet Analysis and Pattern Recognition, pp. 1182-1186, Beijing, China, 2-4 Nov. 2007.

[2] Pranob K Charles, V.Harish, M.Swathi, CH. Deepthi, "A Review on the Various Techniques used for Optical Character Recognition", International Journal of Engineering Research and Applications, Vol. 2, Issue 1, pp. 659-662, Jan-Feb 2012.

[3] Bhatia Neetu, "Optical Character Recognition Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014.

[4] Liana M. Lorigo and Venu Govindaraju, "Offline Arabic Handwriting Recognition: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 28 Issue 5, May 2006.

[5] K. Gaurav and Bhatia P. K., "Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition", 2nd International Conference on Emerging Trends in Engineering & Management, ICETEM, 2013.

[6] Salvador España-Boquera, Maria J. C. B., Jorge G. M. and Francisco Z. M., "Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 4, April 2011.

[7] U. Pal, T. Wakabayashi and F. Kimura, "Handwritten numeral recognition of six popular scripts," Ninth International conference on Document Analysis and Recognition ICDAR 07, Vol.2, pp.749-753, 2007.

[8] Anita Pal & Dayashankar Singh, "Handwritten English Character Recognition Using Neural," Network International Journal of Computer Science & Communication. Vol. 1, No. 2, July-December 2010, pp. 141-144.

[9] J. Pradeep, E. Srinivasan and S. Himavathi, "Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Network", International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011.

[10] A. Brakensiek, J. Rottland, A. Kosmala and J. Rigoll, "Offline Handwriting Recognition using various Hybrid Modeling Techniques & Character N-Grams", Available at http://irs.ub.rug.nl/dbi/4357a84695495.

[11] Reena Bajaj, Lipika Dey, and S. Chaudhury, "Devnagari numeral recognition by combining decision of multiple connectionist classifiers", Sadhana, Vol.27, part. 1, pp.-59-72, 2002.

[12] Sandhya Arora, “Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition”, IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, INDIA, December 2008.

[13] Mohammed Z. Khedher, Gheith A. Abandah, and Ahmed M. AlKhawaldeh, “Optimizing Feature Selection for Recognizing Handwritten Arabic Characters”, proceedings of World Academy of Science Engineering and Technology, vol. 4, February 2005 ISSN 1307-6884.

[14] Sushree Sangita Patnaik and Anup Kumar Panda, “Particle Swarm Optimization and Bacterial Foraging Optimization Techniques for Optimal Current Harmonic Mitigation by Employing Active Power Filter Applied Computational Intelligence and Soft Computing”, Volume 2012, Article ID 897127.

[15] T.Som, Sumit Saha, "Handwritten Character Recognition Using Fuzzy Membership Function", International Journal of Emerging Technologies in Sciences and Engineering, Vol.5, No.2, pp. 11-15, Dec 2011.Lubomir prech,yuri y.,vladimir, “Overview of apex project results”,frontiers.published on 21 December 2018 .

[16] Renata F. P. Neves, Alberto N. G. Lopes Filho, Carlos A.B.Mello, CleberZanchettin, “A SVM Based Off-Line Handwritten Digit Recognizer”, International conference on Systems, Man and Cybernetics

[17] G. Pirlo and D. Impedovo, “Fuzzy Zoning Based Classification for Handwritten Characters”, IEEE Transaction on pattern Recognition and Machine Intelligence, vol.19, no. 04, pp.780-785, August 2011. [18] Yoshimasa Kimura, "Feature Selection for Character Recognition Using Genetic Algorithm", Fourth International Conference on Innovative Computing, Information and Control 978-0-7695-3873-0/09© 2009 IEEE, 2009. [19] Nafiz Arica, and Fatos T. Yarman-Vural, ―Optical Character Recognition for Cursive Handwriting,‖ IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.24, no.6, pp. 801-113, June 2002.

CHAPTER 3

FEASBILITY STUDY

3.1 INTRODUCTION

Feasibility of the system in an important aspect, which is to be considered. The system needs to satisfy the law of economic, which states that the maximum output should be yielded in minimum available resources.

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions. There are five types of feasibility study—separate areas that feasibility study examines, described below.

1. Technical Feasibility

From a technical perspective, since this project makes heavy use of numerical computations, using Octave is a wise choice as it will make the program more efficient. This software will also provide us with some libraries to read and manipulate the images that will make the implementation process easier.

As for the dataset to use in the testing of the project, we have chosen the MNIST Database. This database contains thousands of handwritten digits that have been used in the development of programs with a similar aim. This dataset is open for public use with no charges. It is also very convenient for our project and will help us reduce the time by using directly as a test set without having to make one ourselves.

Since all the tools to be used in this project are free of charge and very easy to use, we can conclude that this project is very feasible in terms of financial resources as well as effort and time.

Requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building—currently; this project is not technically feasible.

2. Economic Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and

enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

3. **Legal Feasibility**

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let's say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization's ideal location isn't zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

4. **Operational Feasibility**

This assessment involves undertaking a study to analyze and determine whether—and how well—the organization's needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

5. **Scheduling Feasibility**

This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

When these areas have all been examined, the feasibility analysis helps identify any constraints the proposed project may face, including:

- Internal Project Constraints: Technical, Technology, Budget, Resource, etc.
- Internal Corporate Constraints: Financial, Marketing, Export, etc.
- External Constraints: Logistics, Environment, Laws, and Regulations, etc.

3.2 MAIN ASPECTS

There are three aspects of feasibility to be considered namely.

1. Technical
2. Operational
3. Economical

TECHNICAL:

In the technical aspects one may consider the hardware equipment for the installation of the software. The system being centralized will required very little hardware appliances. Hence this helps the system to work smoothly with limited amount of working capitals.

OPERATIONAL:

In the operational aspects may think of the benefits of the workload that many a personal may have to share. This is eased out and the required

output may be retrieved in a very short time. Thus there is accuracy in the work on time is also saved there will be very little work that needs to be performed.

ECONOMICAL:

Economical system is definitely feasible because the hardware requirement is less and the operational working for the system requires less number of recruits. This help introduction over-staffing and wastage funds.

We studied on the position to evaluate solution. Most important factors in this study were tending to overlook the confusion inherent in system Development the constraints and the assumed studies. It can be started that it the feasibility study is to serve as a decision document it must answer three key questions.

1. Is there a new and better way to do the job that will benefit the user?
2. What are the costs and savings of the alternatives?
3. What is recommended?

On these questions it can be explained that feasibility study of the system includes following different angles.

3.2.1 Technical feasibility:

This centers on the existing computer system (hardware, software etc.) and to what extent it can support the proposed additional equipment .in this stage of study, we have collected information about technical tools available by which I could decide my system design as the technical requirements.

3.2.2 Operational Feasibility:

In this stage of study we have checked the staff availability. I concentrate on knowledge of end users that are going to use the system. This is also called as behavioral feasibility in which I have studied on following aspects; people are inherently resistant to change, and computers have been known to facilitate change .An estimate has been made to how strong a reaction the user staff is having toward the development of a computerized system. It is common knowledge that computer installations have something to do with turnover. I had explained that there is need to educate and train the staff on new ways of conducting business.

3.2.3 Economical feasibility:

Economic analysis is the most frequently used method for evaluating the effectiveness of candidate system. More commonly known as cost\benefit analysis, the procedure is to determine the benefits and savings that benefits outweigh costs. The decision was to design and implement system because it is for having chanced to be approved. This is an ongoing effort that improves the accuracy at each phase of the system life cycle.

In developing cost estimates for a system I need to consider several cost elements. Among these is hardware personal facility. Operating and supply costs.

3.3 BENEFITS

Benefits of conducting feasibility study:

- Improves project teams' focus
- Identifies new opportunities
- Provides valuable information for a "go/no-go" decision
- Narrows the business alternatives
- Identifies a valid reason to undertake the project
- Enhances the success rate by evaluating multiple parameters
- Aids decision-making on the project
- Identifies reasons not to proceed

3.4 SYSTEM REQUIREMENT SPECIFICATION

This project's main objective is to be able to read the images containing the handwritten digits and be able to identify those digits using basic image correlation techniques. These images are normally represented and read as matrices, in which every element portrays a pixel. The image correlation technique takes these matrices and compares them using some algorithms so as to identify the match that represents the digit we are trying to figure out. This project will be mainly using matrices and heavy numerical computations, that is why it is very important to consider the tools that would provide us with a suitable environment for performing these computation

3.4.1 Octave

Octave is free and open source software that uses a high-level programming language. It has the same functionalities as Matlab and is compatible with it. It offers a very simple and suitable interface to exert some mathematical computations. It provides some tools to solve mathematical problems like some common linear algebra problems [13]. It is also very efficient when it comes to the use of resources, i.e., time and memory, when it comes to these operations. Also, it is very easy to use it when dealing with matrices, as it provides with many functions and operations that make it less costly to manipulate them. In this project, we will deal with images as matrices, in which each element represents a pixel that is why it is very necessary for us to choose a tool that will make our computations easier and more efficient in terms of time and memory resources. Both Matlab and Octave are very easy to learn and work with and provide a suitable environment for this kind of projects. We have opted for Octave as it is free and open source.

3.4.2 MNIST Database

The MNIST database, which stands for the Modified National Institute of Standards and Technology database, is a very large dataset containing several thousands of handwritten digits. This dataset was created by mixing different sets inside the original National Institute of Standards and Technology (NIST) sets, so as to have a training set

containing several types and shapes of handwritten digits, as the NIST set was divided into those written by high school students and others written by the Census Bureau workers . The MNIST dataset has been the target of so many research done in recognizing handwritten digits. This allowed the development and improvements of many different algorithms with a very high performance, such as machine learning classifiers.

In order to be able to implement our recognizer and test its performance, it is necessary to have a suitable dataset which contains a large number of handwritten digits. This dataset should be able to allow us to discover the challenges and limitation of the image correlation technique and push us to look for ways and rules to enhance it and assess its accuracy. We have opted for this dataset to be used for testing our program since it has proved a great reliability and importance in the field.

CHAPTER 4

DESIGN

4.1 INTRODUCTION

The first step we had to go through while working on this project was getting familiar with the tools used, i.e., Octave and the MNIST dataset. After setting up the environment for Octave to work perfectly and downloading the dataset, I have started experimenting with both in order to get familiar with them and know how to use them easily in the future.

Since all the programming is mainly done in Octave, we had to download it along with its Graphical User Interface into the computer, and learn a little bit about its functions and how to use it. Octave is a free software which makes it very easy to work with matrices and vectors and is very efficient in performing calculations on them. I have started learning how to use it and looking for its main functions that I will be using in the implementation of the project. For that, I have used some random images of digits to see how they can be read and modified as well as how to apply some computations on them.

Moreover, I had to investigate the format of the MNIST dataset and get familiar with its representation.The MNIST dataset, which was used to create our test set, contains thousands of handwritten digits, represented as matrices. It has been used in the development of several programs and projects with the same aim as ours. After downloading the file which contains the handwritten digits, I have loaded it on Octave in order to visualize the images and figure out how to use and manipulate them.

4.2 SYSTEM DESIGN

System design is the process of planning a new system or to replace the existing system. Simply, system design is like the blueprint for building, it specifies all the features that are to be in the finished product.
System design phase follows system analysis phase. Design is concerned with identifying functions, data streams among those functions, maintaining a record of the design decisions and providing a blueprint the implementation phase.
Design is the bridge between system analysis and system implementation. Some of the essential fundamental concepts involved in the design of application software are:

- Abstraction

- Modularity
- Verification

Abstraction is used to construct solutions to problem without having to take account of the intricate details of the various component sub problems. Abstraction allows system designer to make step-wise refinement, which at each stage of the design may hide, unnecessary details associated with representation or implementation from the surrounding environment.

Modularity is concerned with decomposing of main module into well-defined manageable units with well-defined interfaces among the units. This enhances design clarity, which in turn eases implementation, Debugging, Testing, Documenting and Maintenance of the software product. Modularity viewed in this sense is a vital tool in the construction of large software projects.

Verification is fundamental concept in software design. A design is verifiable if it can be demonstrated that the design will result in implementation that satisfies the customer's requirements. Verification is of two types namely.

- Verification that the software requirements analysis satisfies the customer's needs.
- Verification that the design satisfies the requirement analysis.

Some of the important factors of quality that are to be considered in the design of application software are:

Reliability:
The software should behave strictly according to the original specification and should function smoothly under normal conditions.

Extensibility:
The software should be capable of adapting easily to changes in the specification.

Reusability:
The software should be developed using a modular approach, which permits modules to be reused by other application, if possible.
The System Design briefly describes the concept of system design and it contains four sections. The first section briefly describes the features that the system is going to provide to the user and the outputs that the proposed system is going to offer.
The second section namely Logical Design describes the Data Flow Diagrams, which show clearly the data movements, the processes and the data sources, and sinks, E-R diagrams which represent the overall logical design of the database, and high-level process structure of the system.

Preliminary Design:

Preliminary design is basically concerned with deriving an overall picture of the system. Deriving entire system into modules and sub-modules while keeping Cohesion and Coupling factors in mind. Tools, which assist in preliminary design process, are Data Flow Diagrams.

Code design:

The purpose of code is to facilitate the identification and retrieval for items of information. A code is an ordered collection of symbols designed to provide unique identification of an entity or attribute. To achieve unique identification there must be only one place where the identified entity or the attribute can be entered in the code; conversely there must be a place in the code for everything that is to be identified. This mutually exclusive feature must be built into any coding system.

The codes for this system are designed with two features in mind. Optimum human oriented use and machine efficiency They are also operable i.e., they are adequate for present and anticipate data processing both for machine and human use.

Input /Output design :

is a part of overall system design, which requires very careful attention. The main objectives of input design are:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable to and understood by the user staff.

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also to provide a permanent hard copy of these results for later consultation.

The various types of outputs are required by this system are given below:

- External outputs, whose destination is outside the concern and which require special attention because they, project the image of the concern.
- Internal outputs, whose destination is within the concern and which require careful design because they are the user's main interface within the computer.
- Operation outputs, whose use is purely within the computer department, E.g., program listings, usage statistics etc,

4.3 SDLC

Software Development Life Cycle (SDLC) is a framework that defines the steps involved in the development of software at each phase. It covers the detailed plan for building, deploying and maintaining the software.

SDLC defines the complete cycle of development i.e. all the tasks involved in planning, creating, testing, and deploying a Software Product.

Figure 4.1: Image depicting the planning step

SDLC Phases

Given below are the various phases:

- Requirement gathering and analysis
- Design
- Implementation or coding
- Testing
- Deployment
- Maintenance

Requirement Gathering and Analysis

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.

Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.

Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product. In case of any ambiguity, a call is set up for further discussion.

Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

Design

In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

Implementation or Coding

Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

Testing

Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

Deployment

Once the product is tested, it is deployed in the production environment or first UAT (User Acceptance testing) is done depending on the customer expectation.

Maintenance

After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

4.4 SOFTWARE ENGG. PARADIGM APPLIED

Software engineering is a layered technology. The foundation for software engineering is the process layer. Software engineering processes the glue that holds the technology layers together and enables ratios and timely development of computer software. Process defines a framework for a set of key process areas that must be established for effective delivery of software engineering technology.

Software engineering methods provide the technical how-to's for building software. Methods encompass a broad array of tasks that include requirements analysis, design, program construction, testing and support. Software engineering tools provide automated or semi-automated support for the process and the methods. When tools are integrated so that information created by one tool can be used by another tool, a system for the support of software development, called computer-aided software engineering is established.

The following paradigms are available:

1. The Waterfall Model

2. The Prototyping Model

3. The Spiral model

Etc.

4.4.1 The Prototype model

The prototype model requires that before carrying out the development of actual software, a working prototype of the system should be built. A prototype is a toy implementation of the system. A prototype usually turns out to be a very crude version of the actual system, possible exhibiting limited functional capabilities, low reliability, and inefficient performance as compared to actual software. In many instances, the client only has a general view of what is expected from the software product. In such a scenario where there is an absence of detailed information regarding the input to the system, the processing needs, and the output requirement, the prototyping model may be employed.

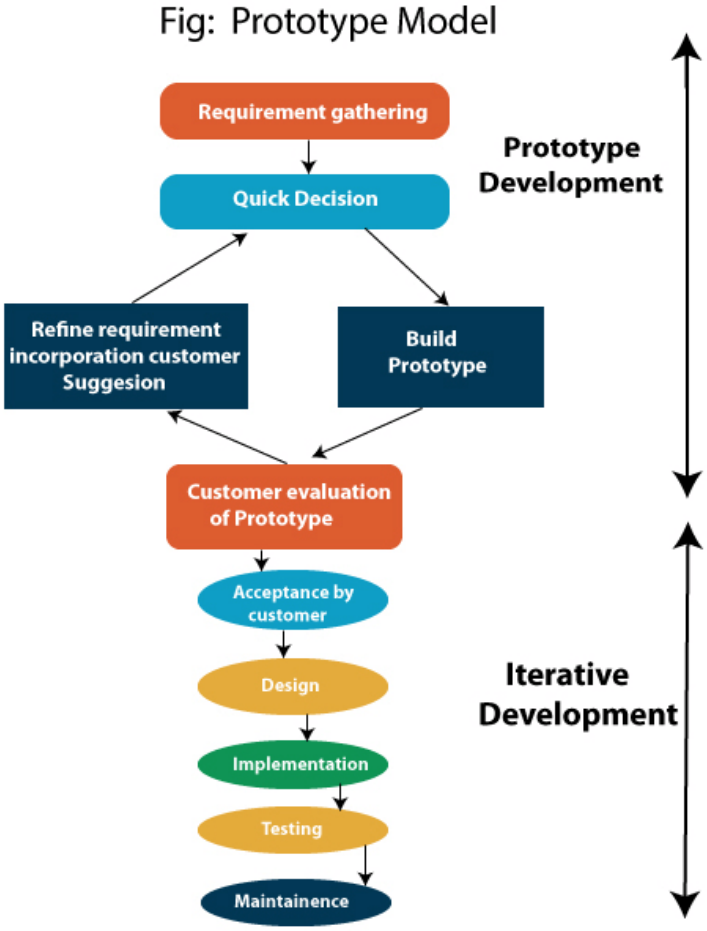


Figure 4.2: Image depicting the prototyping model

4.4.1.1Advantage of Prototype Model

1. Reduce the risk of incorrect user requirement
2. Good where requirement are changing/uncommitted
3. Regular visible process aids Presidency
4. Support early product marketing
5. Reduce Maintenance cost.
6. Errors can be detected much earlier as the system is made side by side.

4.4.1.2 Disadvantage of Prototype Model

1. An unstable/badly implemented prototype often becomes the final product.
2. Require extensive customer collaboration
    - Costs customer money
    - Needs committed customer
    - Difficult to finish if customer withdraw
    - May be too customer specific, no broad market
3. Difficult to know how long the project will last.
4. Easy to fall back into the code and fix without proper requirement analysis, design, customer evaluation, and feedback.
5. Prototyping tools are expensive.
6. Special tools & techniques are required to build a prototype.
7. It is a time-consuming process.

CHAPTER 5

DATA

5.1 GETTING FAMILIAR WITH THE TOOLS

The first step we had to go through while working on this project was getting familiar with the tools used, i.e., Octave and the MNIST dataset. After setting up the environment for Octave to work perfectly and downloading the dataset, I have started experimenting with both in order to get familiar with them and know how to use them easily in the future.

Since all the programming is mainly done in Octave, we had to download it along with its Graphical User Interface into the computer, and learn a little bit about its functions and how to use it. Octave is a free software which makes it very easy to work with matrices and vectors and is very efficient in performing calculations on them. I have started learning how to use it and looking for its main functions that I will be using in the implementation of the project. For that, I have used some random images of digits to see how they can be read and modified as well as how to apply some computations on them.

Moreover, I had to investigate the format of the MNIST dataset and get familiar with its representation.The MNIST dataset, which was used to create our test set, contains thousands of handwritten digits, represented as matrices. It has been used in the development of several programs and projects with the same aim as ours. After downloading the file which contains the handwritten digits, I have loaded it on Octave in order to visualize the images and figure out how to use and manipulate them.

5.2 CREATING THE REFERENCE SET

One of the main steps in the project is creating the reference and the test set that will both be used in the implementation phase.

The test set is to be used in order to assess the performance of the program and evaluate its success or error rate. It is to be taken from the MNIST dataset, since it contains the handwritten digits that we intend to recognize and identify.

As for the reference set, it is used to compare the test images and be able to identify the digit they represent. It is to be created using different fonts.

5.3 DIFFERENT VERSIONS

After having a look at the dataset and deciding on tools to be used for the implementation, we have started the development of our mechanism by developing a very basic version. After that, we have started identifying the challenges and problems

we have faced and kept enhancing it. We have ended up with several versions different one from another, in each new version we increase the accuracy of the program by improving the method and introducing some new rules.

5.4 DATASET FORMAT

The dataset that I have downloaded from the MNIST database contains 60,000 images of handwritten digits, from zero to nine, all grouped in one file. Each of the images is of size 28 by 28 pixels and represents a digit.I have noticed that there is no pattern or order to the way the images were organized in the file. The images are represented as matrices, of which the elements represent the pixels. Also, each image has a label that indicates the digit represented. This label was very helpful later on in order to be able to create the test set. Furthermore, the data did not contain noise or any major problems to deal with, that is why it was used without preprocessing it.

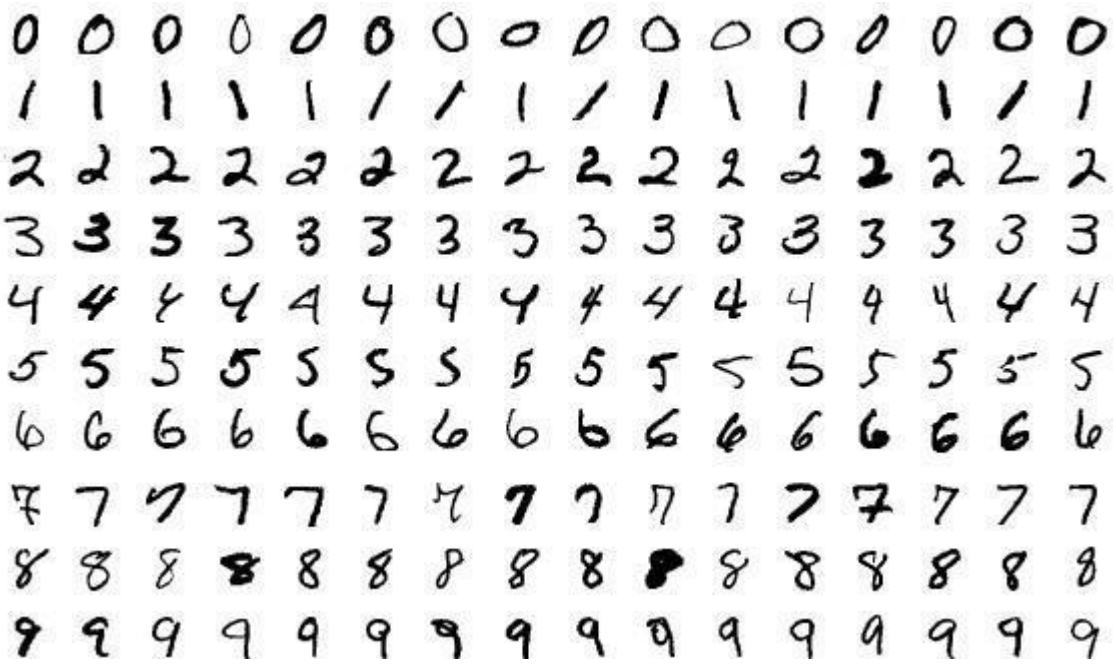


Figure 1.3: Image depicting the Dataset Format

5.5. REFERENCE SET

To be able to recognize the digit represented by a certain image, it is required to compare it with other images containing known digits to be able to make the decision. For that it is necessary to create a reference set which will contain all these images.

That is to say, each image we would want to recognize is to be compared to the images in the reference set. The image with the highest match is the one that represents the right number. Since handwritten digits differ from a person to another, the reference set needs to have digits with different fonts. That is why,

we have created six images of each digit using the online image editor *pixlr.com*, each one with a different font. The reference set contains images with the same dimensions as the ones in the MNIST dataset, i.e., 28 by 28 pixels. Furthermore, these images have a black background and a white font, which made it easier to use and manipulate them later on using Octave. Furthermore, to make the comparison easier, we have regrouped eachsix images representing the same digit under one file. So the resulting reference set was ten files, each one representing a digit from zero to nine, and containing six images of that digit in different fonts.
The pixels of these images are then changed into zeros and ones, which makes the overlapping of the images easier. The black background was initially represented as zeros, so it is left the same. As for the pixels of the white font, each one of them was represented with a different non zero value depending on the shade ofwhite. These non zero values are all converted into ones. The following image displays the digit "2" reference set. Rest of the reference sets are in Appendix A.

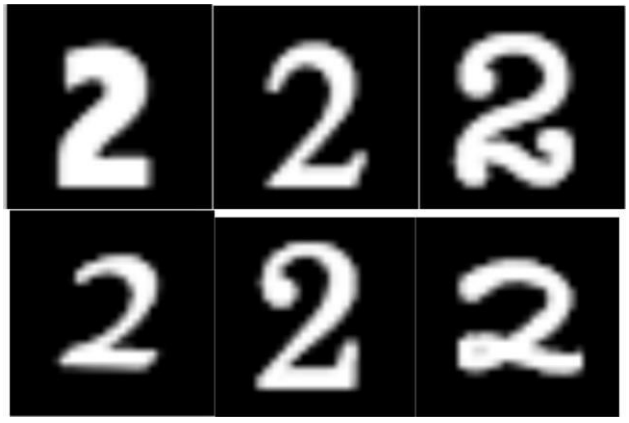


Figure 1.4: Image depicting the Reference Set

5.6. TEST SET

The program to be developed needs to be tested against some images that contain handwritten digits so as to be able to assess its performance and calculate its success rate. That is why it is very necessary to create a test set. This set was formed using the file from the MNIST database. The original file contained 60,000 images representing different digits. This made it difficult to look for each number using the label for the testing of the program. In order to make it easier to access each digit we want, we have decided to store a number of images from each digit in a separate file. The black pixels were originally represented as zeros, so they were left the same. As for the white ones, each of them had a different non zero number, so we turned them all into one

CHAPTER 6

REPORT

In this project we aim at building a mechanism that would recognize handwritten digits from the MNIST dataset. We have opted for the Image Correlation technique, also referred to as Matrix Matching. The goal of this project is to use the basic and simple concepts of this methods and see how good can we make the accuracy rate without using complex techniques such as machine learning.
We have started the implementation of the program using a very simple and basic method, which is explained in further details under the section Version 1, then we have calculated its error rate. Afterwards, we have tried to spot the problems in the mechanism and find the limitations of the technique in order to improve it, and that is how we ended up with a second version of the program. We kept doing the same thing, each time trying to improve the previous version, which enabled us to keep improving the program and reach a higher accuracy and performance.

6.1 VERSION 1 - FIRST MAXIMUM

The first version of our mechanism was very straightforward. We have applied the concept of matrix matching literally and just consider the digit with highest match as the one represented by the test image.
In this version, we have taken the test image and compare it with every image we have in the reference set. We have ten files as a reference set, each one of them contains six images that we have created before. Since the images are read as matrices, whose elements are zeros and ones, for each file the test image is overlapped with all six images, using the dot product. Therefore, by summing the result of the dot product of both the test and the reference image, we get the number of pixels in which they overlap. We, then, take the maximum overlap out of the six results, and store it in a different matrix M. Each column in the matrix M contains the maximum overlap with a certain digit, the first element being the maximum overlap with the zeros, the second one representing the overlap with the ones, and so on, until the overlap with the nine. Finally, the result is the digit corresponding to the maximum pixels overlap.
This technique, although it might seem logical, resulted in a very low accuracy rate, which is of 12.5%. This represents a reliability rate slightly higher than the random guess, which is 10%, since we have ten digits.
After testing this technique with the 200 images that we have in the test set and looking at the results, we were able to see the kinds of problems that caused the error rate to be very high. The first thing we have noticed is that the majority of the digits we are trying to identify match with the eight the most. This could be explained by the eight being the digit with the most number of pixels, which would make it have a very large overlap

with the test images. Also, the shape of this digit allows most of the test images to fit with it.
If the digits in the test images have a very large overlap with the eight but represent another digit, there is a high chance that the number of pixels left out of the reference image, representing the eight, is a very large number. This problem could occur in other cases with the opposite case, where the number of pixels left out of the test image is significantly higher than the one left out of the reference image. That is why we thought about not only considering the number of pixels that do overlap, but investigate the number of pixels left out of both the test and the reference image.

6.2 VERSION 2 - FIRST AND SECOND MAXIMUM WITH RULES

In the first version of the program, we have noticed that most of the unknown digits match with the eight. In order to avoid this issue, we have decided to consider the second maximum as well. By just looking at the second maximum and considering it as the number we are trying to identify, the accuracy rate is 22%. In this case we would not take into consideration all the numbers which had a correct match with the first maximum.
For this reason we have decided to introduce some rules that would help us distinguish if the digit in the test image should be mapped with its first or second maximum. In this technique we do not only look at the number of pixels that overlap, but we investigate the pixels left out from both the test and the reference image as well.
Introducing this rule helped us slightly increase the success rate of the program from 22% to 24.5%. This percentage is still very small and needs to be further improved. That is why, we thought that one way to reduce the margin of error in our program is to change the reference set, since we are relying on images that contain typed digits in order to recognize the handwritten ones.

6.3 VERSION 3 - UPPER LEFT ADJUSTED IMAGES

In this version of the program, we have tried to solve the issue related to the digits that are either shifted or rotated by an angle. The approach we have considered in order to overcome this challenge is to shift and adjust all the non-zero pixels to the upper left of the image (Appendix B). The figure that follows gives an example about this approach with the two different digits 1 and 4.

Figure 1.5: Image depicting Adjusted Images

This method did not give us much accuracy, and unfortunately, we did not investigate it more so as to find a way to improve it due to time constraints.

6.4 VERSION 4 - EXTRACTING THE REFERENCE SET FROM THE MNIST DATABASE

The initial reference set we have worked with contained images of types digits with different fonts. The decision of taking different fonts was mainly because to maximize the chances of a correct match, since the handwritten digits are written in different manners.

However, this was not a very accurate approach since the typed digits differ a lot from the handwritten ones. This approach resulted in a very low performance. That is why, we have decided to change the reference set, from the set of files containing the typed digits, into a set of images from the initial MNIST database, but not the same ones we have in the test set (Appendix C).

By following this change of reference set, and by taking just the first maximum without introducing any rules to it, we have an accuracy rate of 52.5%. We can see that the performance has changed drastically by changing the reference set.

That is why, the first step we did is to display matrices containing the maximum overlap with each digit along with the number of pixels left out from the test image and the number of pixels left out from the reference image (Appendix D). These results were used so as to be able to generate the rules.

After looking at the results of the first and second maximum, we have printed the pixels left out from both the test and the reference image, in order to be able to see the relationships between them and how they can help improve the efficiency of the program. We ended up extracting the following rules:

1. If the first maximum is 8, and the second maximum is 1, then we look at the number of pixels left out from reference image of the 8 after the overlap with the test image of 1. If this number is significantly high, then we consider that the unknown digit is a 1, otherwise it is considered to be 8.
2. If the first maximum is 6, and the second maximum is 3, then we look at the number of pixels left out from the reference image of the 6 after overlapping it with the test image of the 3. If this number is significantly high, then we consider the unknown digit to be a 3, otherwise we take it as a 6.
3. If the first maximum is 8, and the second maximum is 7, then we consider the number of pixels left out from the reference image representing the 8. If this number is significantly high, then we consider the unknown digit to be 7, otherwise, we consider it as 8.
4. If the first maximum is 7, and the second maximum is 6, then we there is a possibility it is either a 7 or 9. That is why we look at the number of pixels left out from both the reference and the test image of the 9.

CHAPTER 7

CODING

```
##########################################################################################
####################                LEFT ADJUSTED                 ########################
##########################################################################################


## Files

filename2 = '/Users/macbookair/Desktop/cap/left.csv';

## Loading test set

#test_0 = load('/Users/macbookair/Desktop/cap/0.mat');


v = ones(28, 1);
j = 0;
while(j < 10)
  x = num2str(j);
  file = strcat("/Users/macbookair/Desktop/cap/", x ,".mat");
  test_0 = load(file);
  cnt0 = 1;
  M0 = zeros(20, 10);
  L0 = zeros(20, 10);
  R0 = zeros(28, 28);
  F = zeros(28, 28);
  cnt0 = 1;
  cnt = 1;
  while(cnt0 <= 533)
    # maximum overlap w/ each number


    # Get test image
    test = test_0(cnt0:cnt0+27, 1:28);

    ## left-adjust the test image
    cnt2 = 1;
    cnt1 = 1;
    while(cnt2 < 29)
      s = test(cnt2, :) * v;   # try sum instead
      if(s > 0)
        F(cnt1,:) = [ones(1,s),zeros(1,28-s)];
        cnt1 = cnt1 + 1;
      endif
      cnt2 = cnt2 + 1;
    endwhile

```

```
    ## compare with 0's
    i = 0;
    while(i < 10)
        m0 = 0;
        y = num2str(i);
        #printf("%s %d\n", y, i);
        file2 = strcat("/Users/macbookair/Desktop/cap/", y ,"_r.mat");
        ref_0 = load(file2);
        c0 = 1;
        while(c0 < 142)
            # get ref image
            ref = ref_0(c0:c0+27, 1:28);

            # left-adjust the ref image
            c00 = 1;
            c01 = 1;
            while(c00<29)
                s = sum(ref(c00, :));
                if(s > 0)
                  R0(c01,:) = [ones(1,s),zeros(1,28-s)];
                  c01 = c01 + 1;
                endif
                c00 = c00 + 1;
            endwhile
            # compare left-adjusted test w/ left-adjusted ref
            m = R0 .* F;
            tot0 = sum(sum(m));
            if(tot0 > m0)
              m0 = tot0;
            endif
            c0 = c0 + 28;
        endwhile
        #printf("i = %d, cnt = %d, m0 = %d\n", i, cnt, m0);
        M0(cnt, i+1) = m0;
        i = i + 1;
    endwhile
    cnt = cnt + 1;
    cnt0 = cnt0 + 28;
  endwhile
  #cnt0 = cnt0 + 28;
  csvwrite(filename2, j, "-append");
  csvwrite(filename2, M0, "-append");
  csvwrite(filename2, "\n\n", "-append");
```

```
4
5   ## Max overlap
6
7   ## OPENING FILE
8
9   filename = "/Users/macbookair/Desktop/cap/v1.txt";
10  fid = fopen (filename, "w");
11  filename2 = "/Users/macbookair/Desktop/cap/rules.csv";
12  fid2 = fopen (filename2, "w");
13
14
15  ## Loading reference set
16
17  ref_0 = load('/Users/macbookair/Desktop/cap/reference/0.mat');
18  ref_1 = load('/Users/macbookair/Desktop/cap/reference/1.mat');
19  ref_2 = load('/Users/macbookair/Desktop/cap/reference/2.mat');
20  ref_3 = load('/Users/macbookair/Desktop/cap/reference/3.mat');
21  ref_4 = load('/Users/macbookair/Desktop/cap/reference/4.mat');
22  ref_5 = load('/Users/macbookair/Desktop/cap/reference/5.mat');
23  ref_6 = load('/Users/macbookair/Desktop/cap/reference/6.mat');
24  ref_7 = load('/Users/macbookair/Desktop/cap/reference/7.mat');
25  ref_8 = load('/Users/macbookair/Desktop/cap/reference/8.mat');
26  ref_9 = load('/Users/macbookair/Desktop/cap/reference/9.mat');
27
28
29  m0 = 0;
30  m1 = 0;
31  m2 = 0;
32  m3 = 0;
33  m4 = 0;
34  m5 = 0;
35  m6 = 0;
36  m7 = 0;
37  m8 = 0;
38  m9 = 0;
39  c1 = 1;
40
41  B = zeros(1, 2);
42
43  lt = 0;
44  lf = 0;
45
46  ## FILE OF ZEROS
47
48  #printf("before %d\n", cnt0);
49  #F = "FILE ZEROS";
50
```

```
51  i = 0;
52  while(i < 10)
53    R = zeros(20, 30);
54    #printf("i = %d\n", i);
55    M = zeros(20, 10);
56    cnt = 1;
57    x = num2str(i);
58    file = strcat("/Users/macbookair/Desktop/cap/", x ,".mat");
59    test_0 = load(file);
60    cnt0 = 1;
61    while(cnt0 <= 533)
62      #printf("i = %d, in\n", i);
63      m0 = 0;
64      m1 = 0;
65      m2 = 0;
66      m3 = 0;
67      m4 = 0;
68      m5 = 0;
69      m6 = 0;
70      m7 = 0;
71      m8 = 0;
72      m9 = 0;
73      c1 = 1;
74
75      test = test_0(cnt0:cnt0+27, 1:28);
76      c0 = 1;
77      while(c0 < 142)
78        ref = ref_0(c0:c0+27, 1:28);
79        im0 = ref .* test;
80        s0 = sum(sum(im0));
81        if(s0 > m0)
82          m0 = s0;
83          lt0 = sum(sum(test)) - s0;
84          lr0 = sum(sum(ref)) - s0;
85        endif
86        c0 = c0 + 28;
87      endwhile
88      M(cnt, 1) = m0;
89      R(cnt, 1) = lt0;
90      R(cnt, 2) = m0;
91      R(cnt, 3) = lr0;
92
93      c1 = 1;
94      while(c1 < 142)
95        ref = ref_1(c1:c1+27, 1:28);
96        im1 = ref .* test;
97        s1 = sum(sum(im1));
```

```
      lr1 = sum(sum(ref)) - s1;
    endif
    c1 = c1 + 28;
  endwhile
  M(cnt, 2) = m1;
  R(cnt, 4) = lt1;
  R(cnt, 5) = m1;
  R(cnt, 6) = lr1;

  #compare with the two's
  c2 = 1;
  while(c2 < 142)
    ref = ref_2(c2:c2+27, 1:28);
    im2 = ref .* test;
    s2 = sum(sum(im2));
    if(s2 > m2)
      m2 = s2;
      lt2 = sum(sum(test)) - s2;
      lr2 = sum(sum(ref)) - s2;
    endif
    c2 = c2 + 28;
  endwhile
  M(cnt, 3) = m2;
  R(cnt, 7) = lt2;
  R(cnt, 8) = m2;
  R(cnt, 9) = lr2;

  #compare with the three's
  c3 = 1;
  while(c3 < 142)
    ref = ref_3(c3:c3+27, 1:28);
    im3 = ref .* test;
    s3 = sum(sum(im3));
    if(s3 > m3)
      m3 = s3;
      lt3 = sum(sum(test)) - s3;
      lr3 = sum(sum(ref)) - s3;
    endif
    c3 = c3 + 28;
  endwhile
  M(cnt, 4) = m3;
  R(cnt, 10) = lt3;
  R(cnt, 11) = m3;
  R(cnt, 12) = lr3;

  #compare with the four's
  c4 = 1;
```

```

  #get first max
  [m1, ind1] = max(M(cnt,:), [], 2);
  #csvwrite(filename2, ind1, "-append");
  M(cnt, ind1) = 0;
  [m2, ind2] = max(M(cnt,:), [], 2);
  #csvwrite(filename2, ind1, "-append");
  ind1 = ind1 - 1;
  ind2 = ind2 - 1;
  if(ind1 == 8)
    if(ind2 == 1)
      if(lr8 > lr1)
        m = ind2;
      else
        m = ind1;
      endif
    elseif(ind2 == 7)
      if(lr8 > lr7)
        m = ind2;
      else
        m = ind1;
      endif
    else
      m = ind1;
    endif
  elseif(ind1 == 6)
    if(ind2 == 3)
      if(lr6 > lr3)
        m = ind2;
      else
        m = ind1;
      endif
    else
      m = ind1;
    endif
  elseif(ind1 == 7)
    if(ind2 == 6)
      if(lt9/lr9 >= 1.8)
        m = 9;
      else
        m = 7;
      endif
    else
      m = ind1;
    endif
  else
    m = ind1;
```

CHAPTER 8

TESTING

Implementing this technology on a large scale could evidently bring about several repercussions. If we were to begin with an impact of this approach, the most obvious one is the social effect. Recognizing handwritten digits, if not used with good intentions, could lead to the manifestation of several social issues.

First of all, this method could lead to the increase of theft in societies. That is to say, if someone was to take a picture of a code or pin number of a private account, which maybe written by the owner, or available in a scanned document, the image could be used, even if it is not of a high resolution as it could be taken by a camera located far away from the original document, in order to be able to segment it and extract the code from it. This problem could appear in several other cases. For example, thieves or other people with malicious incentives, could use this technique in order to get some important codes and keys of important ciphers, so as to serve political or personal needs.

This kind of images can also be extracted from social media accounts or from phones. Nowadays, with the increase of technological advancements, the use of social media became very popular, and people started sharing many information through private direct messages, which could also include this types of codes and ciphers. If someone is spying on these people or could penetrate their private accounts in a way or another, they could get them and extract the information they need in order to serve their personal motives.

In addition to that, recognizing handwritten digits could be used in a bad way so as to break the captcha code. This code is used in order to make sure that the entity who is accessing a certain website or is trying to get a benefit from a certain service is a human being and not a computer. The captcha is usually put as an image to make it hard for computers to read it, which will make sure that only human beings can be granted access. However, if digits recognition is used in this case by computers, then they would be able to appear as humans.

Furthermore, these issues could increase the lack of trust in societies, therefore, affect the social behavior of people. As a consequence of the bad uses and applications of this technique, people in different societies could become more suspicious and doubtful of everyone around them. If people notice that this kind of methods could be used in spying and theft, then there will no longer be trust inside societies and the problem of wariness will end up being elevated. This problem will affect the behaviors of these people towards each other.

Consequently, this would lead to the increase of the lack of communication as well as misunderstandings. If everybody could see how this technology could be used against their own benefit and used to serve some immoral motives this would lead to the creating an environment of lack of trust can have a psychological impact on human beings. As an example, this problem could lead to emotional distress as people would be worried, all the time , that someone is spying on them or trying to use their personal and private information for their own incentives.If this kind of behaviour is proven to be true, then it would raise some questions and concerns related to psychology. In other words, when people strive to look for other people's personal information in order to use it for their own benefits or to serve some other party's evil intentions this means that there is a whole psychological factor behind it that is pushing them towards this behavior. If these psychological problems could be traced, it could be possible to reduce them or prevent them in the future in order not to fall in this kind of problems to begin with.

CHAPTER 9

ANALYSIS

9.1 SOCIAL

Image processing in general can cause the opposition of some people from different societies. The reason behind this is because this field can be further developed for malicious intentions, perhaps in the military field to be able to target certain people, and many other similar possibilities. That is why if image processing is not well managed and used with a wise mentality and good intentions, it could represent a threat to societies.

9.2 TECHNOLOGICAL

The output of this project is a program that can read images of unknown handwritten digits and be able to identify. Therefore, it can be useful in the future to identify images containing handwritten digits. Moreover, it provides some techniques that could be further developed in order to be able to recognize all types of characters in images or scanned documents.

9.3 ECONOMIC

This kind of projects can be used by companies and could be implemented in some softwares, which could be able to reduce their costs. In 1997, HWAI, developed by the US Postal Services in order to recognize handwriting in letters was very successful in cost reduction, as they have reached eight billion dollars in cumulative

9.4 ENVIRONMENTAL

This project does not have any effect on the environment, therefore, does not need to take into consideration any environmental factors.

9.5 POLITICAL

This project does not have any direct impact on the political factors. However, if we take it into the broader context of image processing as a whole, then we can say that this field could be used by some political parties to reach their objectives, either with good or bad motives, as mentioned earlier in the societal section.

9.6 LEGAL

Image processing needs to be taken into consideration by all the authorities, as they need to enforce some laws on its uses and applications to ensure that none of them could be used against the well-being and serenity of humanity.

9.7 ETHICAL

This project may open the discussion to a very large ethical dilemma related to the reduction of human labor due to computerization. The wider field of image processing may appear as a risk of having a lower need for human beings in some fields. Going back to the example of HWAI by USPS, this technology was able to reduce 100 million dollars of labor costs, which means it has reduced the number of employees. Moreover, this field could raise some ethical questions about its applications. Since using image processing can enable us not only to develop characters recognition but facial recognition as well, in addition to multiple other applications, people could be worried about preserving their privacy and security.

CHAPTER 10

CHALLENGES AND LIMITATIONS

This project was my first encounter with Optical Character Recognition (OCR). That is why, while working on this project I was faced with many challenges and issues. First of all, it took me a long while to understand all the concepts and get familiar with them, from image processing to OCR, to all of the techniques and algorithms used in it. Furthermore, the data we were dealing with was very problematic in terms of the way the digits are written. Since some of the digits were rotated by an angle, some of them were thicker or thinner than the rest, and some of the digits were not well centered or were written in confusing ways. In addition to that, overcoming these challenges was not easy since we were only using basic image correlation techniques. We have tried to maximize the success rate, but we have only reached 57%, which is not a very high performance.

CHAPTER 11

CONCLUSION AND FUTURE WORK

Optical Character Recognition is a very broad field concerned with turning an image or a scanned document containing a set of characters into an encoded text that could be read by machines. In this project, we have attempted to build a recognizer for handwritten digits using the MNIST dataset. The challenge of this project was to be able to come up with some basic image correlation techniques, instead of some sophisticated algorithms, and see to what extent we can make this mechanism accurate. We have tried several versions and kept trying to improve each one in order to reach a higher performance rate. The last version has reached a rate of 57% accuracy. Unfortunately, we could not compare the performance of the mechanism we have built to some others that have already been designed and/or implemented before because we did not find any academic paper that tackles this method. The performance we have reached is far less than that of machine learning, which reaches a performance rate of 99.3%; however, it could be further improved and made into a better one. The goal of this project was to explore the field of OCR and try to come up with some techniques that could be used without going into deep computations, and even if the final result is not very reliable, it still provides an accuracy way better than random.

The future steps that to go for would be having a closer look at the results of all the versions in order to find new rules. By extracting and implementing them, we will be able to enhance the performance of these versions. Moreover, it would be good if we could make some modifications to both the reference set and the rules in order to make our program more general and able to identify both typed and handwritten digits.

Furthermore, in the future, we could make a great use of the matrices that indicate the first maximum overlap of each test image with the reference images, along with the number of pixels left out from both. These matrices could be used with some clustering algorithms to build a program able to recognize handwritten digits with a very high efficiency.

Last but not least, we thought about using linear or high level regression in the versions we have developed in order to create more rules. As regression could be used for binary classification and is not very suitable to classify a digit out of ten, this technique could be used in order to tell which digit is the most suitable, the first maximum or second maximum, which will enable us to generate more rules; thus, reach a higher efficiency.

CHAPTER 12

REFERENCES

[1] Kai Ding, Zhibin Liu, Lianwen Jin, Xinghua Zhu, "A Comparative study of GABOR feature and gradient feature for handwritten chinese character recognition", International Conference on Wavelet Analysis and Pattern Recognition, pp. 1182-1186, Beijing, China, 2-4 Nov. 2007.

[2] Pranob K Charles, V.Harish, M.Swathi, CH. Deepthi, "A Review on the Various Techniques used for Optical Character Recognition", International Journal of Engineering Research and Applications, Vol. 2, Issue 1, pp. 659-662, Jan-Feb 2012.

[3] Bhatia Neetu, "Optical Character Recognition Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014.

[4] Liana M. Lorigo and Venu Govindaraju, "Offline Arabic Handwriting Recognition: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 28 Issue 5, May 2006.

[5] K. Gaurav and Bhatia P. K., "Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition", 2nd International Conference on Emerging Trends in Engineering & Management, ICETEM, 2013.

[6] Salvador España-Boquera, Maria J. C. B., Jorge G. M. and Francisco Z. M., "Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 4, April 2011.

[7] U. Pal, T. Wakabayashi and F. Kimura, "Handwritten numeral recognition of six popular scripts," Ninth International conference on Document Analysis and Recognition ICDAR 07, Vol.2, pp.749-753, 2007.

[8] Anita Pal & Dayashankar Singh, "Handwritten English Character Recognition Using Neural," Network International Journal of Computer Science & Communication. Vol. 1, No. 2, July-December 2010, pp. 141-144.

[9] J. Pradeep, E. Srinivasan and S. Himavathi, "Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Network", International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011.

[10] A. Brakensiek, J. Rottland, A. Kosmala and J. Rigoll, "Offline Handwriting Recognition using various Hybrid Modeling Techniques & Character N-Grams", Available at http://irs.ub.rug.nl/dbi/4357a84695495.

[11] Reena Bajaj, Lipika Dey, and S. Chaudhury, “Devnagari numeral recognition by combining decision of multiple connectionist classifiers”, Sadhana, Vol.27, part. 1, pp.-59-72, 2002.

[12] Sandhya Arora, “Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition”, IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, INDIA, December 2008.

[13] Mohammed Z. Khedher, Gheith A. Abandah, and Ahmed M. AlKhawaldeh, “Optimizing Feature Selection for Recognizing Handwritten Arabic Characters”, proceedings of World Academy of Science Engineering and Technology, vol. 4, February 2005 ISSN 1307-6884.

[14] Sushree Sangita Patnaik and Anup Kumar Panda, “Particle Swarm Optimization and Bacterial Foraging Optimization Techniques for Optimal Current Harmonic Mitigation by Employing Active Power Filter Applied Computational Intelligence and Soft Computing”, Volume 2012, Article ID 897127.

[15] T.Som, Sumit Saha, "Handwritten Character Recognition Using Fuzzy Membership Function", International Journal of Emerging Technologies in Sciences and Engineering, Vol.5, No.2, pp. 11-15, Dec 2011.Lubomir prech,yuri y.,vladimir, “Overview of apex project results”,frontiers.published on 21 December 2018 .

[16] Renata F. P. Neves, Alberto N. G. Lopes Filho, Carlos A.B.Mello, CleberZanchettin, “A SVM Based Off-Line Handwritten Digit Recognizer”, International conference on Systems, Man and Cybernetics