Project on: Hand Writing Recognition

COMILLA UNIVERSITY

Course Title: Image Processing

Course Code: ICT 710

Group - F

Submitted To:

Alimul Rajee

Lecturer

Department of ICT

Comilla University

Submitted By:

Md. Rakibul Islam Riyad (ID-11909005)

Gargi Roy Tushi (ID-11909016)

Tamanna Akter (ID-11909020)

Arshadul Islam Sourav (ID-11909031)

Meheniger Alam (ID-11909035)

Department of ICT

Comilla University

Submission Date: 22th March, 2023

Declaration

"We Do hereby declare that this submission is our own work conformed to the norms and guidelines given by our Supervisor sir and that, to the best of our knowledge and belief, it contains no material previously written by another neither person which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text."

....

Meheniger Alam
ID-11909035, Session-(2018-19)
Dept. of ICT, Comilla University

....

Gargi Roy Tushi
ID-11909016, Session-(2018-19)
Dept. of ICT, Comilla University

....

Rakibul Islam Riyad
ID-11909005, Session- (2018-19)
Dept. of ICT, Comilla University

....

Arshadul Islam Sourav
ID-11909031, Session-(2018-19)
Dept. of ICT, Comilla University

....

Tamanna Akter
ID-11909020, Session-(2018-19)
Dept. of ICT, Comilla University

Date-22th March,2023

Certificate of Acceptance

This is to certify that the project report entitled “Hand Writing Recognition” by ID 11909035,11909005,11909016,11909020 & 11909031 has been carried out as a partial fulfillment of requirement for the Image Processing based lab. The dissertation has been carried out under my guidance and is a record of the authentic work carried out successfully. Their performance has been satisfactory during this project period.

I wish their every success in life.

Supervisor:

....

Alimul Rajee

Lecturer

Dept. of Information and Communication Technology

Comilla University, Cumilla.

Acknowledgement

This project report is the outcome of the contributions and sincere co-operation of different persons. For the fear of sounding like a vote of thanks speech, we could not possibly thank all of those marvelous people who have contributed something of them directly or indirectly in preparing this project report successfully. They are of course some very special people who cannot go without mention. It gives us immense thank them for their cordial cooperation and encouragement.

First of all, we would like to thank Almighty Allah and we express heartfelt gratitude to our respected supervisor sir Alimul Rajee (Lecturer, Dept. of ICT, Comilla University).

Specially, we are very much grateful to our teammates and classmates of ICT 10th batch for co-operating and helping us in collection of necessary data and in the preparation of this project report.

In spite of sincere and denoted efforts, there might be some mistakes in the report. We take the entire responsibility for such unintended errors and omissions.

Thanking you,

Meheniger Alam, Rakibul Islam Riyad

Tamanna Akter, Arshadul Islam Sourav

Gargi Roy Tushi

Session: 2018-19

Dept. Of ICT,Comilla University.

Abstract

Handwriting recognition is a field of research in computer science that deals with the automated recognition of handwritten text or symbols. It is a challenging problem due to the large variability in handwriting styles, the presence of noise and artifacts in the input data, and the need to handle cursive writing. There are two main approaches to handwriting recognition: online and offline. Online recognition involves capturing the pen strokes as they are written, while offline recognition involves analyzing a scanned image of the handwritten text. Recent advances in deep learning techniques have led to significant improvements in the accuracy of handwriting recognition systems. These systems typically use CNNs or RNNs to extract features from the input data and make predictions. Applications of handwriting recognition include OCR for digitizing documents, signature verification, and inputting text on mobile devices. In our proposed method we will use the MNIST dataset & CNN algorithm. The accuracy obtained by this project is about 98%.

List of Contents

List of Figures

List of Tables

List of Abbreviations

HWR	Handwriting Recognition
HTR	Handwriting Recognition Technology
MLP	Machine Learning Pattern
KNN	Knowledge-Based Network
RNF	Random Forest
NLP	Natural Language Processing
KNN	K-Nearest Neighbors
SVM	Support Vector Machine
CNN	Convolutional Neural Network
CPU	Central Processing Unit
GPU	Graphics Processing Unit
GUI	Graphical User Interface
MLP	Multi-Layer Perceptron
DP	Deep Learning
DLA	Diffusion Limited Aggregation
CTC	Connectionist Temporal Classification
IAM	Identity and Access Management

UCM	Universal Church of Man
MNIST	Modified National Institute of Standards and Technology Database
OPENCV	Open Source Computer Vision Library
RNN	Recurrent Neural Network

CHAPTER ONE

Introduction

Hand writing recognition is the technology that allows computers or digital devices to recognize and interpret handwritten text. It involves analyzing and interpreting handwritten text images, identifying the characters or words present in image, and converting them into machine-readable text. It can be used in a variety; of applications, such as digitizing handwritten notes or documents, recognizing handwritten addresses on envelopes, converting handwritten text into typed text in text editors or email clients, and even recognizing handwritten signature for authentication purposes. It has applications in many fields, including education, finance, healthcare, and law enforcement.

1.1 Introduction

Handwriting recognition, sometimes known as HWR or HTR, is the capacity of a computer to get and decipher coherently written input by which a computer receives and interprets intelligible handwritten data from a variety of sources, including paper documents, photographic images, touchscreen displays, and other electronic devices Formatting, splitting letters, and finding the most relevant words are all handled by the handwriting recognition system.

Automatic handwriting character recognition acknowledgment has numerous scholarly and marketable interface. Dealing with the enormous variety of writing styles of various authors presents the fundamental challenge in handwritten character identification. The most challenge in manually written character acknowledgment is to bargain with the colossal assortment of penmanship styles by diverse scholars. Moreover, a few complex penmanship scripts contain distinctive styles for composing words. Depending on the dialect, characters are composed disconnected from each other in a few cases (e.g., English, Bangla, and Arabic). In a few other cases, they are cursive and some of the time characters are related to each other. This challenge has been as of now recognized by numerous analysts within the field of NLP.

Besides, English comprises of numerous comparative molded characters like as Bangla. In a few cases, a character varies from its comparative one with a single dab or stamp. Besides, English too contains a few extraordinary characters with proportionate representation of vowels. Transcribed digit acknowledgment includes distinguishing 10 characters, i.e., 0-9, but the input is delicate to the natural clamor. Input is sensitive to environmental noise.

The most popular approach to constructing the handwritten digit recognizer is using a MLP, KNN or support vector machine SVM. Different other procedures were created utilizing diverse methods with MLP structure. Arbitrary Woodland, SVM, KNN, and other machine learning procedures have been created to recognize written by hand digits. Profound learning strategies like CNN have the most noteworthy precision when compared to the foremost commonly utilized machine learning calculations for manually written character acknowledgment. Design acknowledgment and large-scale image classification are both done with CNN. Penmanship character acknowledgment may be acquire about field in computer vision, manufactured insights, and design acknowledgment. It could be claimed that a computer application that conducts handwriting recognition has the capacity to procure and recognize characters in photos, paper documents, and other sources, and change over them to electronic or machine-encoded shape. Profound learning could be a well-known field of machine learning that employments progressive structures to memorize high-level deliberations from information. Concurring to references, the accessibility of innovation CPUs, GPUs, and difficult drives, among other things, machine learning calculations, and expansive information, such as MNIST manually written digit information sets are all variables in profound learning's success.

Deep learning is currently gaining popularity as a method for extracting and learning to recognize deep patterns. It can produce patterns from a given dataset at a deep learning level. It is a remarkable algorithm with a variety of libraries to recognize patterns in photographs and classify them. The CNN is one of the more effective deep learning algorithms, and it performs well at image classification, image recognition, pattern identification, feature extraction, and other tasks. However, in our project, we'll use the packages Tensorflow, Keras, Matplotlib, and Numpy to create our system.

1.2 Problem Statements

Character is difficult to define, and there is to universally accept single definition the words is used to refer to both the inner different writing method of a person researchers have developed various taxonomies of character related character trained by machine learning for use in different domains or contexts.

1.3 Objectives

1.3.1 Broad Objectives-

The main objective of this study is to evaluate the classification of the general regression neural network for character Recognition. Sub objectives contain select the ideal model. Training and testing the input data recognized the character.

1.3.2 Specific Objectives-

Hand written character acknowledgment is more challenging compared with the printed shapes of character due to the taking after reasons:

• Handwritten characters composed by distinctive journalists are not as it were no indistinguishable but too change completely different perspectives such as estimate and shape;

• Numerous varieties in composing styles of person character make the acknowledgment assignment troublesome;

• The similitudes of distinctive character in shapes, the covers, and the interconnects of the neighboring characters encourage complicate the character acknowledgment issue.

1.4 Findings

In our project on Handwriting Recognition, we have used the CNN with some python frameworks and also used the A-Z English alphabet consisting of 5 vowels and 21 consonant words as well as MNIST datasets. This study have reported high accuracy rates of up to 98% in recognizing isolated handwritten characters, while recognizing full sentences and paragraphs still presents challenges.

1.5 Chapter Summary

To sum up, the wide range of writing styles and unique characteristics of handwritten characters make it difficult to categorize them.

1.6 Keywords

Isolated hand writing recognition, MNIST, Python library, Tensorflow, Keras, Matplotlib, Numpy.

1.7 Paper Type

Project Report

1.8 Chapter Overview

This study has been documented in five chapters. Chapter one reviews the research problem and the goals of the study. Chapter tow reviews the definition of character pattern recognition and its kinds and its components and its applications. Chapter three on the study methodology review in. Chapter four on the artificial neural networks contained. Chapter five contain the results and conclusion and recommendations.

CHAPTER TWO

Literature Survey

In this literature review, we will explore the current state-of-the-art techniques and technologies for handwriting recognition. We will delve into the various approaches used, their strengths and limitations, and their applications in different domains. Additionally, we will examine the challenges that are associated with handwriting recognition, including variations in handwriting styles, character, segmentation, and noise in input data. We will also discuss about the datasets, results and other limitations from the previous experiments by different author. We hope to provide a comprehensive overview of the current landscape of handwriting recognition research, highlight the most promising approach, and identify areas that require further investigation.

2.1 Background of Study

Developing a handwriting recognition system requires a deep understanding of image processing, feature extraction, machine learning, language modeling, and testing and evaluation. With the right approach and expertise, it is possible to create an accurate and reliable system for recognizing handwritten text. The key areas must be studied: the dataset used for the model, there may be so many datasets such as Banglalekha-Isolated, MNIST, IAM, CVL, RIMES, NumtaDB, EMNIST used by different researchers on their projects. Again there different types of algorithms like as CNN, SVM, MLP, DL, KNN, DLA, CTC along with the frameworks known as Tensorflow, Keras, Matplotlib, Panda and Numpy were used in different researches and they gave different accuracy which were much better than the previous proposed methods.

2.2 Theoretical Background

This section provides state-of-the-art pertinent to character recognition based on machine learning Tandra et all used the CNN with the dataset of Banglalekha – Isolated dataset. They got training loss of 0.0108 and validation loss of 0.0153. Obtained training accuracy is 99.87% and validation accuracy is 99.50%. In this paper they intend to solve the problem of explicit feature extraction and proposed a method that will automatically select and extract feature from the character images irrespective of the individual writing style and spacing. We have also considered 50 classes of basic letters (11 vowel classes and 49 consonant classes) and 10 classes of digits and tried to propose a method that will have better overall accuracy than existing methods. [3]

According to I Khandokar et all, the average accuracy increases with higher number of training images. The accuracy obtained from 200 training images as 65.32% and the accuracy reaches to 92.91% with the 1000 training images. Where they have used the Convolutional Neural Network with the datasets of NIST. The handwritten characters in NIST was given as images. The images were split into training and testing sets. Training was carried out with various number of images; then testing was conducted to find the accuracy of the CNN. The main focus of their work was to investigate CNN capability to recognize the characters from NIST dataset with a high degree of accuracy. [4]

Yintong Wang, Wenjie Xiao and Shuo Li had proposed that the fundamental purpose of deep learning is to obtain equivalent or better learning effects based on less domain knowledge or expert experience. In this model they had used the CNN and RNN algorithm with the dataset of IAM, Bentham, BH2M, HIT-MW, HWDB1.0-1.1, HWDB2.0-2.2 . From authors view, this trend will continue for future work, better handwritten text feature extract and recognition will explore by researchers. [5]

Yasir Babiker Hamdan and Prof. Sathish presented on the Statitical Method, Template Maching Method, Structural Pattern Acknowledgement, proposed statistical SVM using MNIST, CENPARMI, UCOM, IAM datasets. Found that SVM-based HCR method gives 94% accuracy and a good recognition rate while compared to existing methods. The proposed model has tested on a training section that contained various stylish letters and digits to learn with a higher accuracy level. We obtained test results of 91% of accuracy to recognize the characters from documents. [6]

Omer Othman Ahmed analyzed on different types of handwritten characters and scripts including English,Arabic, and others. Each text contains a set of symbols, known as letters, which have basic shapes. Where he used the Tensor flow , Lightgbm, Numby , Theano , Keras , Pandas , Scipy , Scikit-Learn , Python , Machine Learning algorithms and found that After training for about 50 epochs the CER is 10.72% while the WER is 26.45% and hence Word Accuracy is 73.55% . [2]

According to Md. Ismail Hossain et all have proposed that Bangla Writing has significantly less no. of samples (21,234 words) with less data diversity (5,470 unique words).BN-HTRd has a large no. of samples (108,147) with large data diversity (23,115 unique words). On their paper A. Synthetically Generated Isolated Printed Characters Our teacher model is an isolated character prediction model. The training dataset for the teacher model is generated synthetically by using the Text Recognition Data Generator module from Python Package Index. A list of various Bangla fonts was fed to the package, along with a set of graphemes. For each grapheme in the grapheme set, arbitrarily 160 isolated character images were generated using random augmentations. B. Handwritten Word Datasets In our experiments, we have used two handwritten word level datasets to train our student model. The Bangla Writing dataset has 21,234 Bangla handwritten words. Additionally, the BN-HTRd dataset has 108,147 Bangla handwritten words. [7]

On the project of Handwriting Digit Recognition using Machine and Deep learning Algorithms , Ritik Dixit, Rishika Kushwah and Samay Pashine had mentioned that after implementing all the models, we found that SVM has the highest accuracy on training data while on testing dataset CNN accomplishes the utmost accuracy with the datasets of MNIST and NIST. They approach the novel task of nonnegative image generation by designing a framework that can take advantage of both existing image-to-image translation methods and human visual perception quirks. Specifically, they first incorporate existing image-to-image translation models to generate a proposal image that satisfies the task, but is not necessarily feasible in the non-negative generation setting. Then fine-tune the proposal image so that it can be produced by adding non-negative light to the input image while remaining perceptually unchanged. Their method is compatible with almost all existing image-to image translation methods, thereby permitting its application to a wide variety of tasks. They draw upon the strengths of state-of-the-art image generation methods while availing ourselves of degrees of freedom provided by the human visual system. [10]

Rabia KARAKAYA and Serap KAZAN proposed that the accuracy of SVM is 90%,Decision Tree making is 87%,Rndom forest is 97%,ANN is 97%,KNN is 98%,KMA is 98% while using the MNIST dataset with the algorithm of SVM, Decision Tree, Random forest, ANN, KNN and KMA .[8]

The boosted LeNet 4 method performs the best with the accuracy of 99.3% and is the best among the methods that have been studied in this paper. The operational/ actual recognition time is 0.05 ms. Using the MLP, SVM was proposed on Analysis of machine learning algorithms for character recognition: a case study on handwritten digit recognition a paper by Owais Mujtaba Khanday and Dr Samad Dadvandipour. A study was done in showed the bundle of feature extraction techniques and were evaluated using the benchmark datasets available publically; the methods and outperformed the other methods available in the literature by showing the accuracy of 99.03% and 98.75% respectively. [9]

Rouhan Noor et all, the author said that The challenges of the NumtaDB dataset are highly unprocessed and augmented images. So different kinds of preprocessing techniques are used for processing images and deep CNN is used as the classification model in this paper. The deep convolutional neural network model has shown an excellent performance, securing the 13 th position with 92.72% testing accuracy in the Bengali handwritten digit recognition challenge 2018 among 57 participating teams. A study of the network performance on the MNIST and EMNIST datasets were performed in order to bolster the analysis. [11]

Although Bangla the official language of Bangladesh and several Indian states with over 200 million native speakers Bangla handwritten character recognition is quite far behind S M Azizul Hakim and Asaduzzaman presented a 9 layer sequential Convolutional Neural Network model to recognize 60 (10 numerals+ 50 basic characters) Bangla handwritten characters. BanglaLekha-Isolated dataset is used as train-validation set. A new dataset of 6000 images is created for cross validation. The proposed model trained to recognize 60 characters achieves state-of-the-art 99.44% accuracy on BanglaLekha-Isolated dataset and 95.16% accuracy on prepared test set. Experiments on recognizing Bangla numerals separately also show state-of-the-art performance. [12]

Md Zahangir Alom et all showed that the performance of several popular DCNNs for handwritten Bangla character (e.g., digits, alphabets, and special characters) recognition. (e experimental results indicated that DenseNet is the best performer in classifying Bangla digits, alphabets, and special characters. Specifically, they achieved recognition rate of 99.13% for handwritten Bangla digits, 98.31% for handwritten Bangla alphabet, and 98.18% for special character recognition using DenseNet. [13]

Md Shopon, Nabeel Mohammed and Md Anowarul Abedin presented that the use of unsupervised pre-training using auto encoder with deep ConvNet in order to recognize handwritten Bangla digits, i.e., 0-9. The datasets that were used in this paper are CMATERDB 3.1.1 and a dataset published by the Indian Statistical Institute (ISI). This paper studies four different combinations of these two datasets-two experiments are done against their own training and testing images, other two experiments are done cross validating the datasets. In one of these four experiments, the proposed approach achieves 99.50% accuracy, which is so far the best for recognizing handwritten Bangla digits. The ConvNet model is trained with 19,313 images of ISI handwritten character dataset and tested with images of CMATERDB dataset. [14]

A database of 114988 samples of handwritten Bangla digits is utilized to do a test on Bangla Handwritten Numeral Recognition Using Deep Convolutional Neural Network by Md. Hadiuzzaman Bappy et all. The database is created by randomly picking samples from a bigger database of 85596 for NumtaDB, 23392 for ISI , and 6000 for CMATERdb for each of the 10 digit classes. The largest dataset collection for handwritten digits in Bangla is called NumtaDB, and it consists of six sizable datasets gathered from various sources. The samples that made up the datasets were compiled on a pre-defined database sheet from persons of various ages, sexes, and educational levels. As previously indicated, we employed three datasets for the training set and test set, with the NumtaDB dataset having 72044 samples and 13552 samples, the ISI dataset having 19337 samples and 3986 samples, and the CMATERdb dataset having 4000 samples and 2000 samples, respectively. [16]

CNN aims to reduce the images into a form without losing any features. In this paper, we introduce an Autoencoder with a Deep CNN, which we call DConvAENNet for recognizing Bangla Handwritten Character (BHC). A total of 22 experiments were performed on the three-character datasets (BanglaLekha-Isolated, CMATERdb 3.1, Ekush). All attempts acquire satisfying results up to 90% accuracy for the recognition of Bangla handwritten numerals, vowels, consonants, compound characters, modifiers, and all characters set unitedly. Using this supervised and unsupervised learning technique, our proposed DConvAENNet model achieved 95.21% on BanglaLekha-Isolated for 84 classes, 92.40% on CMATERdb 3.1 for 238 classes, and 95.53% on Ekush for 122 classes. Most errors in our model were caused due to the similarity and high curvature nature of the BHC sets is showed in a paper of Md Ali Azad et all. [15]

Shahrukh Ahsan et all choose ”BanglaLekha-Isolated” dataset to take the first handwritten samples. The entire set contained a significant number of samples ranging from simple to modern handwriting and cursive. Two hundred samples are taken for individual characters for implementation. Where the model had an accuracy of up to 94% for a single character and an average of 91% for all characters. This method provides good results compared to existing methods of Bengali handwriting recognition and is more efficient. On their study they developed a different approach using face mapping to improve the recognition of handwritten Bengali characters. It is quite effective in distinguishing different characters. The real highlight is that the recognition results are more efficient than expected with a simple machine learning technique. The proposed method uses the Python library Scikit-Learn, including NumPy, Pandas, Matplotlib, and SVM classifier. [17]

According to Aditya Srivastava and Pawan Singh on their study showed the discipline of AI is quickly extending, especially in the computer vision fields. In computer vision, the possibilities of a human making an error is about 3%. This recommends that PCs are now better compared to people at distinguishing and investigating photographs. What an incredible achievement it is that computers used to be large pieces of technology the size of a room; now, they can comprehend the world around us in ways we never imagined. Undoubtedly there’s huge scope of machine learning in the coming next years. [18]

After training for about 50 epochs Anil Chandra Naidu Matcha found that the CER is 10.72 % while the WER is 26.45 % and hence Word Accuracy is 73.55% on his study. The model is able to predict the characters accurately to a great extent but it suffers in few cases such as awfully is predicted as and fully, stories is predicted as stories. In the project he used CTC algorithm with the IAM, CVL, RIMES datasets. In the project he broadly classified the handwriting recognition method into the below two types- online methods and offline methods. Online methods involve a digital pen/stylus and have access to the stroke information, pen location while text is being written as the seen in the right figure above. Offline methods involve recognizing text once it's written down and hence won't have information to the strokes/directions involved during writing with a possible addition of some background noise from the source i.e paper. [19]

When running the application a window will pop up where one can write the digit. And next, when clicking on recognize button, it will recognize the digit you have written with the probability percentage showing how exactly the digit matches with the original one. Here the author have written digit 1 and it recognized it as 1 with 17% accuracy while using CNN algorithm .He also used MNIST (contains images of digits from 0 to 9,all these images are in greyscale, 60000 training images which are very large and about 10000 testing images. All these images are like small squares of 28 x 28 pixels in size) developed by Amrutha K. [23]

Tsige Tadesse Alemayoh ORCID et all developed a novel digital pen that embodies two main sensors: inertial and force sensors. Handwriting data for the 36 alphanumeric (10 numeral and 26 small Latin) characters were collected from six subjects. The segmented datasets were restructured into an 18 × 200 2D array of virtual images. As a validation method, the dataset was used to train four neural network models (ViT, CNN, LSTM, and DNN) using deep-learning methodologies. ViT performed better than the other three with a validation accuracy of 99.05%. While finding the accuracy they used LSTM , CNN , DNN , ViT algorithms. [24]

Saniya Firdous used different types of English handwritten words such as cursive or block writing. Along with this, Text-to-Speech is used to help people who have trouble reading on-screen text and KNN, SVM, Naïve Bayes algorithms but she found various feature based classification techniques for offline handwritten character recognition. After experimentation, it proposes an optimal character recognition technique. The proposed method involves segmentation of a handwritten word by using heuristics and artificial intelligence. The results obtained by using the proposed Character Recognition system are found to be satisfactory. [25]

It is important to not treat these DL models as black box, but to gain some intuition as to why predictions are being made in these ways. We already provided some explanations regarding failure spaces of the models, which motivated the use of ensemble learning. In this section, we use an interpretation of LIME for time series data, lime-for-time. In this interpretation, the time series is divided into 20 “slices” and the importance of each slice is determined by the LIME algorithm. Additionally, since we are working with MTS data, each of the 13 signals (channels) will be analyzed separately. The top 30 signals and slices, in terms of importance, will be examined. A green bar, indicating a score greater than 0, represents that this section of the data has a positive impact on the model output and a red bar, indicating a score less than 0, represents that this section of the data has a negative impact on the model output. The graphs below show all 13 channels of the raw data with an importance bar overplayed on top of it. The darkness of colour indicates how much influence that slice of the data has on the classification according to Hilda Azimi et all. They used L-WD, L-WI, U-WD, U-WI, U-WD, C-WD, and C-WI with different characteristics datasets and found the accuracy by using the DL algorithm. [26]

The teacher model is an isolated character prediction model. The training dataset for the teacher model is generated synthetically by using the Text Recognition Data Generator module from Python Package Index. A list of various Bangla fonts was fed to the package, along with a set of graphemes. For each grapheme in the grapheme set, arbitrarily 160 isolated character images were generated using random augmentations. They have used two handwritten word level datasets to train our student model. The Bangla Writing dataset has 21,234 Bangla handwritten words. Additionally, the BN-HTRd dataset has 108,147 Bangla handwritten words by Md. Ismail Hossain et all .They found Bangla Writing has significantly less no. of samples (21,234 words) with less data diversity (5,470 unique words).BN-HTRd has a large no. of samples (108,147) with large data diversity (23,115 unique words).And they used LILA- for Bangla Handwriting Recognition. [27]

IAM is a popular benchmark used for handwriting recognition. WikiText is a set of 103 images used to make synthetic images. Free Form Answers is a proprietary dataset with 13K samples, Answers2 is 16K words and lines taken from handwritten test responses, and Names is a private dataset with 22K names and identifiers. This dataset was used in the DNN algorithm by SumeetS. Singh and Sergei Karayev. When the model is trained on all the datasets and tested on the Free Form Answers dataset, it gets the error rate down to 7.6%, which is lower than the best cloud API's 14.4%. It only takes about 4 seconds to download a set of 2500x2200 pcs, 456 characters, and 1165 lines without any compression, which is when the model is pruned, distilled, or quantized.[32]

2.3 Literature Summary

Table I- Table on Literature Summary

No.	Dataset	Algorithm	Result
1	IAM	DNN	Error rate of 7.6% vs the best available cloud API's 14.4%. Inferencing takes an average of 4.6 seconds on a single CPU thread.[32]
2	BanglaLekha-Isolated	SVM	Accuracy of up to 94% for a single character and an average of 91% for all characters. [17]
3	IAM, CVL, RIMES	CTC	CER is 10.72 % while the WER is 26.45 % and hence Word Accuracy is 73.55%. [19]
4	MNIST	CNN	Here the author have written digit 1 and it recognized it as 1 with 17% accuracy. [23]
5	36 alphanumeric characters	LSTM, CNN, DNN, ViT	Validation accuracy of 99.05%. [24]
6	Different shaped latters of English , Arabic and others	Python, Tensorflow , MLs	After training for about 50 epochs the CER is 10.72% while the WER is 26.45% and hence Word Accuracy is 73.55% . [2]
7	Cursive or block writing English handwritten words	KNN,SVM	The results obtained by using the proposed Character Recognition system are found to be satisfactory. [25]
8	L-WD, L-WI, U-WD, U-WI,C-WD, and C-WI	DL	The MTS data, each of the 13 signals (channels) will be analyzed separately. [26]
9	Human handwriting style, Color.	DNN	Statistically, the frequency of numerical characters (including Roman numerical characters) in the UIT-HWDB-word trainset is approximately 0.065 and in the UIT-HWDB-word test set is approximately 0.0066.
10	IAM, RIMES	NN	Accuracy 12.7% in IAM and 6.6% in RIMES.
11	DW ,DS	SVM	The word error rate in the above example is WER = (2 + 1 + 1)/6 · 100 = 66%. One should notice that the WER can be larger than 100% because the hypothesis can be longer than the reference.
12	NumtaDB	CNN	Training accuracy 99.66%, Validation accuracy 98.96% ,Training loss 0.0163, Validation loss 0.0508
13	CMATERDB and ISI	Autoencoder , DCNN	For the CMATERDB, it achieves an accuracy rate of 99.50%, which is the best reported result on this dataset so far.
14	Banglalekha-Isolated, Prepared	DCNN	For 10 numerals achieving 99.6% accuracy on prepared dataset and 99.82% accuracy on BanglaLekha-Isolated dataset.
15	NumtaDB	DCNN	Achieved 92.72% testing accuracy which is a good result for large and unbiased NumtaDB dataset comparing to other biased datasets.
16	MNIST	CNN	The average accuracy increases with higher number of training images. The accuracy obtained from 200 training images as 65.32% and the accuracy reaches to 92.91%

			with the 1000 training images
17	MNIST, CENPARMI, UCOM, IAM	SVM	SVM-based HCR method gives 94% accuracy and a good recognition rate while compared to existing methods
18	IAM, Bentham, BH2M, HIT-MW	CNN	From authors view, this trend will continue for future work, better handwritten text feature extract and recognition will explore by researchers.
19	Density based clustering Features extraction	CNN	The results of the present process will transfer to the NN stage which generates a high level of correctness and accuracy by training.
20	OnHW -chars	KNN, SVM.	ML models improved the accuracy 23.56% and our DL models improved the accuracy of 7.01%.
21	EMNIST	CNN	The model was able to predict with an accuracy of 87.1 percent. The loss with keras model during training began at 0.68 with the first epoch and ended up at 0.31.
22	Real world dataset	Heuristic or greedy searching algorithm	It is possible to reduce the number of features up to 30%, without any loss in terms of recognition rate, while a higher reduction of the feature number (up to 70%) implies a reduction of the recognition rate less than 2%.

CHAPTER THREE

Methodology and Modeling

The methodology for a handwriting recognition project involves several stages, including data collection, preprocessing, feature extraction, and classification. The first step is to collect a dataset of handwritten images. The next step is to preprocess the dataset by removing any noise or artifacts that may interfere with the recognition process. Techniques like edge detection, contour analysis, and texture analysis can all be used in feature extraction. Techniques like neural networks, support vector machines, and decision trees can all be used in classification. The trained model can then be used to recognize new handwritten characters and texts. The modeling aspect of a handwriting recognition project involves selecting an appropriate algorithm or model architecture to perform the classification task. There are several popular models for handwriting recognition, including: CNN, KNN, SVM, DNN and so on.

3.1 Image Recognition Methods

3.1.1 Steps of Developing the Model

1. Read the data: The dataset is being read using **pd.read_csv ()**, and the first 10 images are being printed using **data.head (10)**.

2. Splitting into images and their labels: Split the read data into images and their corresponding labels. Since the column "0" contains the labels, we remove the '0' column from the read data frame and use it in y to form the label.

3. Reshaping the Data in the CSV file: In the segment above, we are using train_test_split() to divide the data into the training and testing datasets. Additionally, we are reshaping the train and test image data, which were initially presented in the CSV file as 784 columns of pixel data, so that they can be displayed as an image. We then change it to 2828 pixels. The labels are all present as floating point values. We transform all of the labels from their original form as floating point values to integer values before mapping the integer values to the characters in a dictionary called word_dict.

4. Plotting the numbers: Only the distribution of alphabets is described here. First, we convert the labels to integer values and add the list to the list according to the label. This list shows the number of images in the dataset for each alphabet. Now we create a list - an alphabet containing all the characters using the values () dictionary function. Now, using the number and alphabet lists, let's draw a horizontal line. Data shuffle: At this point, we have changed the order of some of the train set's images.

5. Data Shuffling: Now let's shuffle some images of the train set. The shuffling is done with the shuffle () function so that we can display random images. Then we create 9 graphics in 3×3 format and show threshold images of 9 alphabets.

3.1.2 Data Reshaping

We now modify the train and test image data sets to be incorporated into the model. New shape of the train data (297960, 28, 28, and 1), new shape of the train data (74490, 28, 28, and 1). It is now necessary to modify the train and test image datasets in order to incorporate them into the model. The train data has changed from a float to a categorical data. This is achieved by the CNN model, which takes the input of labels and converts the output into a probability vector.

3.1.3 CNN

One of the most used techniques for handwriting recognition is CNN. The image must first go through pre-processing before being fed into the convolutional neural network. The pre-processing steps are as follows:

1) Input the image you want to recognize

2) Do editing or twisting. The objective is that the image portion that does not need to be recognized is misplaced

3) Set the image size. All images must be the same size [20]

Figure 1 shows an illustration of Handwriting Recognition Diagram

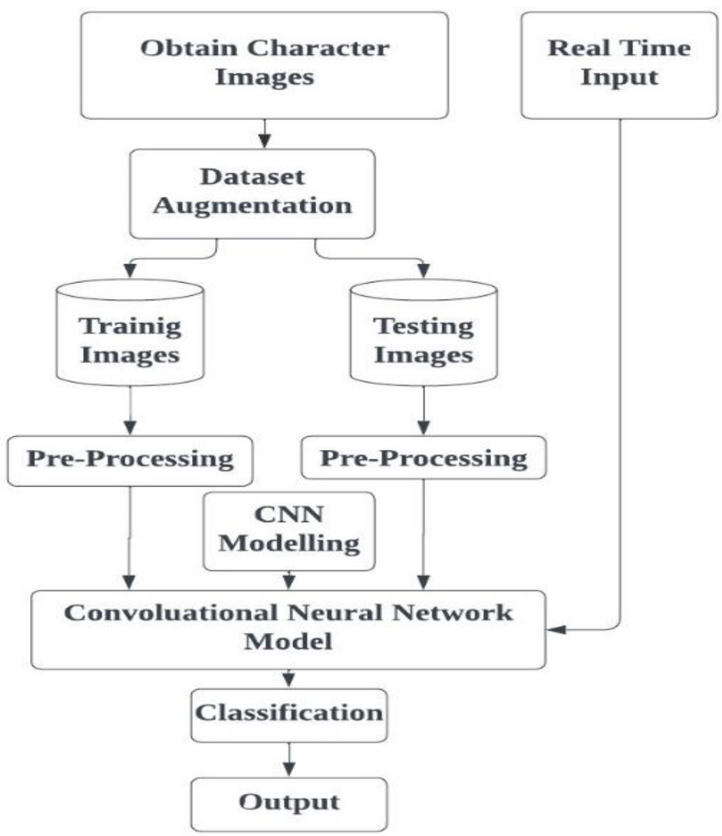

Figure 1: Diagram of Hand Writing Recognition

Incorporating an additional layer into a CNN architecture can have a significant impact on the accuracy of the image detection. Convolutional layer, sub-sampling layer, and fully connected layer are the three layers that make up CNN in general. But another layer, such as the soft max layer, can also be added. Each layer was connected to the one before it. The soft max level improves the accuracy of image recognition. A CNN architecture example without an additional layer is shown in Figure 2. Depending on the situation, CNN may or may not use the same number of layers. Differences in recognizable handwritten languages also affect the layers and number of layers used. Additional layers inserted into the CNN are optional. Adding an extra layer to the CNN has an effect. For example, adding a soft max layer to the CNN results in higher handwriting recognition accuracy than a CNN without the soft max layer. What layers are employed and how many layers are used are also influenced by variations in identifiable handwriting language. On CNN, further layers can be introduced at your discretion. There will be a result if a new layer is added to CNN. For instance, if the soft max layer is added to CNN, the handwriting will change.

A CNN is typically composed of three layers:

1) Convolutional Layer :

Convolutional layer is the introductory layer that builds a CNN. The convolution procedure is carried out in this layer. The input picture feature is extracted by this image's convolution procedure. The final convolution layer maintains the spatial location and grayscale information of the convolution feature map.

2) Sub-sampling layer :

Pooling layer (Subsampling). Used to transform an input feature into a statistically-resulted representation of its surrounding features, with the resulting feature size much smaller than the previous feature. Max pooling is mostly used in CNN's subsampling.

3) Fully-Connected Layer :

As a classifier on CNN, this layer may be a CNN design comprising of input layer, hidden layer, and output layer.

4) Soft-max Layer :

On CNN, the soft max layer comes last. A Soft Max Layer is used to represent the output in the form of probabilities. Quite helpful for classifying. The soft max layer is used to classify characters. The soft max function has a value between 0 and 1. The class with the maximum value will be selected as the class for the image, while the smaller value means not including the main image to be recognized.

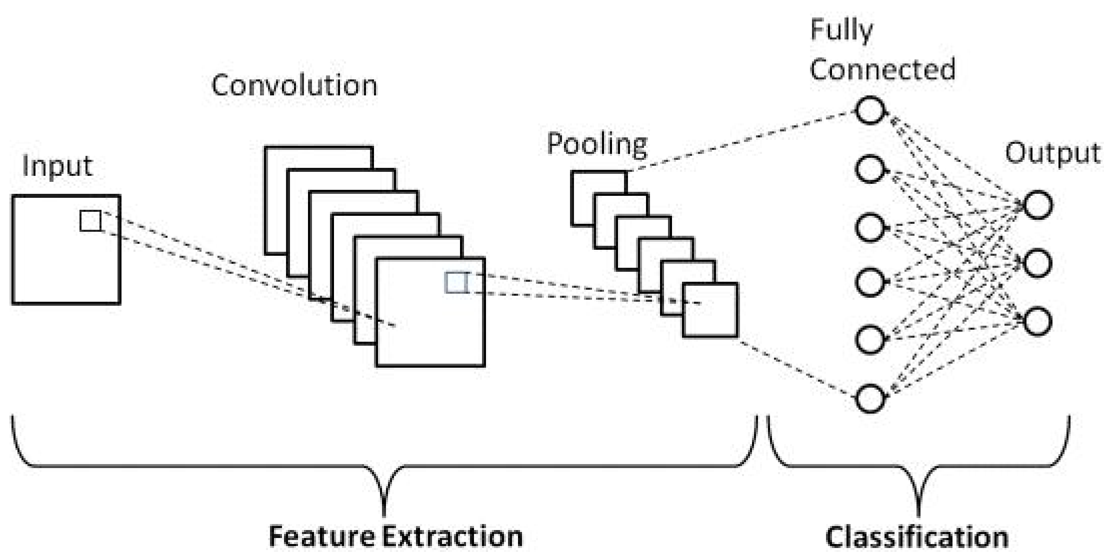

Figure 2: CNN Architecture

Source: Datafair.training

3.2 Project Perquisites

Following are the prerequisites for this project:

1. Python - version 3.7.4
2. IDE - Jupyter

Required frameworks used in the project-

1. Numpy -version 1.16.5
2. CV2 or openCV - version 3.4.2
3. Keras - version 2.3.1
4. Tensorflow (TensorFlow is used Keras in backend and for some image preprocessing) - version 2.0.0
5. Matplotlib - version 3.1.1
6. Pandas - version 0.25.1

3.3 Python Frameworks

Some of the majorly used libraries in python for machine learning are discussed below-

1. NumPy -

The NumPy library was developed shortly after Numeric, an older library, and it is used to handle multi-dimensional data and sophisticated mathematical operations. NumPy is an extremely quick computing library that can handle a wide range of tasks and features, from basic algebra to Fourier transforms, shape manipulations, and random simulations.

2. Pandas -

The complete name for Pandas is Python data analysis library. This allows us to create multidimensional structures by updating records for numerical data and time series. It uses data frames and collecting to simultaneously display 3-dimensional and 2-dimensional data. Additionally, it provides alternatives for quickly indexing huge statistics in large datasets. It is highly known for its skill in reshaping records, dealing with missing records, and merging data and filter data. Pandas could be veritably salutary and rapid-fire with huge datasets.

3. Keras -

Keras is an advanced deep learning API written in Python for neural networks. It helps in computing multiple background networks and makes the implementation of neural networks smooth. As a high-level TensorFlow API, Keras is a robust and easy-to-use open-source Python framework for creating deep learning models.

4. Matplotlib -

Matplotlib could be a library utilized in Python for graphical outline to recognize the records some time recently moving it to data-processing and preparing it for Machine learning capacities. It uses object-oriented APIs to obtain graphs and associated visualizations using Python GUI toolkits. A similar UI to MATLAB is also provided by Matplotlib so users can carry out operations similar to those found in MATLAB. This open-source, free package offers multiple extension interfaces that extend the matplotlib API to various other libraries. In machine learning, there are four essential sorts: administered learning, unsupervised learning, semi-supervised learning, and strengthening learning. [18]

5. OpenCV -

OpenCV is an open source computer vision and machine learning computer program library. OpenCV was erected to supply a common frame for computer vision operations and to quicken the use of machine recognition within the marketable particulars. [22]

6. TensorFlow -

TensorFlow is a comprehensive open source machine learning framework. TensorFlow is a versatile system for managing all aspects of a machine learning system. [21]

3.4 Project Modeling

3.4.1 Data Acquisition

A. Synthetically Generated Isolated Printed Characters

Our model is a specialized character prediction model. The training dataset for the model is synthetically generated using the Text Recognition Data Generator module from the Python Package Index. A list of various English fonts was entered into the package along with a series of numbers. For each grapheme in the grapheme set, randomly separated character images were generated using random additions.

B. Datasets of handwritten words

Two handwritten word-level datasets were used to train our model in our experiments: the MNIST dataset of 372450 Alphabet images of 28×2, all present in the form of a CSV train CSV format, and the A-Z English Character dataset. For orienteer-dataset evaluation protocol, we have picked this dataset for training, and the other for testing and also repeated the process by changing the places of the datasets. Random augmentations were used to create randomly isolated character images for every grapheme in the grapheme set. We chose this dataset for training and the other for testing in our inter-dataset assessment strategy.

3.4.2 MNIST Datasets

The MNIST dataset is utilized to construct this application. This dataset consists of images of digits ranging from 0 digits to 9 digits, all rendered in a greyscale format. Both training and testing images are included in the dataset, with the training images comprising approximately 60000 large-scale training images and the testing images comprising approximately 10000 smaller-scale testing images. Each training image is composed of a small square (28 x 28 pixels) and a testing image is a handwritten image of an individual digit. The following is how these datasets images appear.

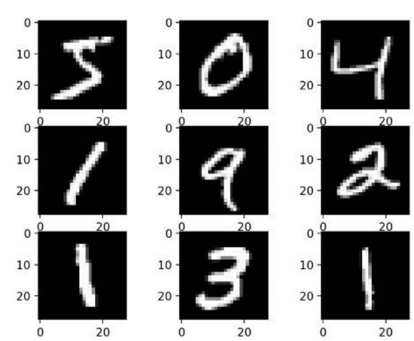

Figure 3: Image of MNIST dataset

Source: Machine Learning Mastery

3.5 Recognition System

The CNN method for recognizing images goes through the following stages:

1. Pre-processing: The image is resized during pre-processing; if it is too large, the calculation will be expensive, or if it is too small, it will be challenging to adapt to big networks. To get the standard size, larger images are cropped, and padding is added to smaller images.

2. Generating datasets: If an open source dataset is not available for handwriting character recognition, it must be built into a new dataset, but if a dataset is available, an existing dataset can be used.

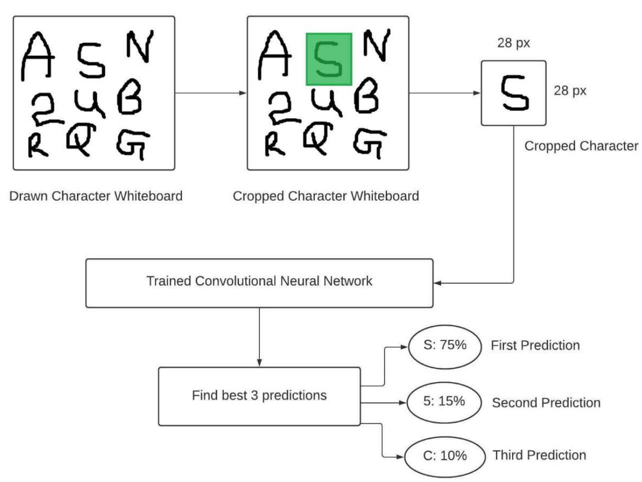

Figure 4: Procedure of the proposed project

3. Determine the final data: A large dataset is required to train a CNN. To achieve this, the resulting images are modified and altered to produce a large number of variations.

4. Classification: The given input image is classified using the Soft max layer, which is the CNN end layer.

5. Testing: The test module is associated with the test image. The test images were obtained by randomly dividing the augmented dataset. [20]

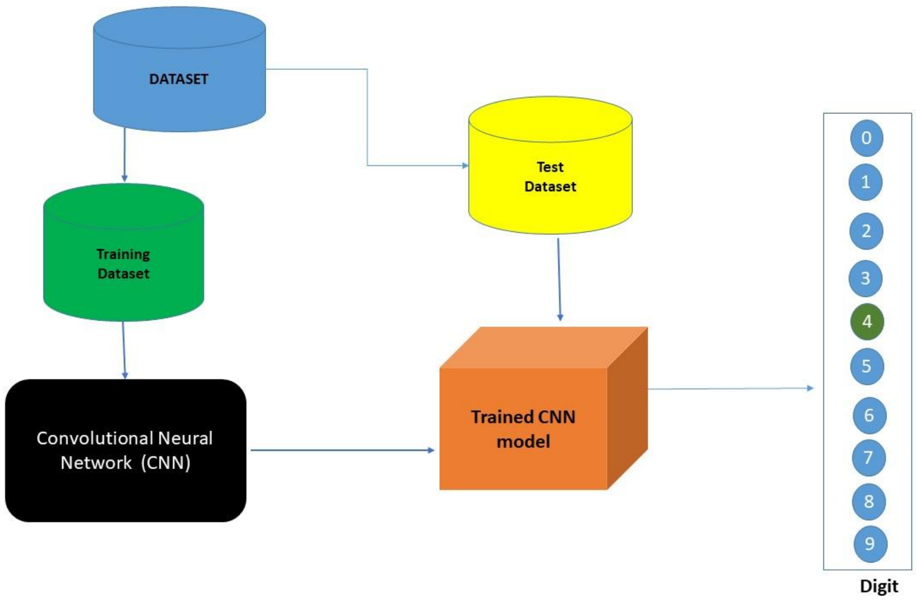

Figure 5: End-to-end CNN

3.6 Experiments

3.6.1 Pseudocode-

The pseudocode of our proposed model is given below-

1: Algorithm CNN

2: input: dataset, dataset true labels

3: output: score of Parallel-CNN trained model on

test dataset

4: let f be the featureset 3d matrix

5: for i in dataset do

6: let fi be the featureset matrix of sample i

7: for j in i do

8: vj ← vectorize (g, w)

9: append vj to fi

10: append fi to f

11: ftrain, ftest, ltrain, ltest— split feature set and labels

into train subset and test subset

12: M ← Parallel-CNN (ftrain, ltrain)

13: score ← evaluate (i, ltesi, M)

14: return score

CHAPTER FOUR

Results and Findings

Handwriting character recognition is the task of accurately identifying and converting handwritten characters into machine-readable text. This project typically involves training a machine learning algorithm on a large dataset of handwritten characters to enable it to recognize and classify new handwritten characters accurately. The results of a handwriting English character recognition project can be measured in terms of accuracy, precision, recall, and F1 score. The accuracy of the model indicates how often the algorithm correctly recognizes the handwritten characters. Precision measures the fraction of the correctly recognized handwritten characters out of all the characters identified by the model. Recall measures the fraction of correctly recognized handwritten characters out of all the actual handwritten characters in the dataset. The F1 score combines the precision and recall measures to give an overall measure of the model's performance. In recent years, handwriting character recognition has seen significant improvements due to the development of deep learning algorithms such as CNNs and RNNs. These models have been able to achieve state-of-the-art results on various handwriting recognition benchmarks, such as the MNIST datasets.

4.1 Experimental Results

CNN provided the best accuracy rates for both datasets, even when cross-testing, showing the usefulness of unsupervised training for this purpose. Tables I and II show previously published results for A-Z handwriting and MNIST digit data sets. Table III shows the accuracy rates for different experiments. For MNIST digit data set, our CNN configuration is pretty close to 100% accuracy at 98.87%. In fact, out of all the ConvNet data sets, our reported result comes in second. When you train a CNN model on the tiny A-Z handwriting dataset and test it on the larger MNIST dataset, you'll still get 98.55% accuracy, which is pretty close to some of the previously published results from Table II. To show how useful supervised pre-training is, we can also see that CNN gives 0.32% better results than SCMA, even with the same data augmentation.

CNN provided the highest accuracy rates for both datasets, including cross-data sets, demonstrating the effectiveness of unsupervised prior training for this purpose. Table I and II summaries the previously published results for A-Z hand written words and MNIST digit datasets. Table III summarizes the error rates obtained in various experiments. For the MNIST digit data set, our CNN configuration provides comparable results at 98.87%.In fact, among all, we also applied our proposed model for the previous dataset, which is used in some research papers. We have used the same preprocessing, and deep CNN model for both datasets. However, we get better results for MNIST than for A-Z English Alphabets. This is because MNIST contains more complex images, which are difficult to recognize. Last but not least, we applied the same procedure to MNIST and found that MNIST digit recognition is 99.70% accurate, while

MNIST state of the art is 99.79%. Finally, we found that EMNIST digit and letter recognition is also 99.79% accurate, with 47 balanced letter classes. Table IV illustrates the improved accuracy of our proposed CNN model compared to an earlier work on an EMNIST digit and balanced dataset using a linear and OPIUM classifier.

4.2 Evaluation

To figure out how well the spotting algorithm works, we look at how many times we've run it on each of the samples in the test data set with different window sizes and overlap. As we said in 3, the spotting algorithm is biased towards recognizing handwriting, so we expect it to have a high recall and a low precision. Since we have a lot more non-handwritten data than handwritten data, we also look at the specificity, which tells us how many non-handwritten samples we've classified correctly. This number is related to the number of times we've seen no handwriting and no handwriting has been spotted. We can get really high recall values up to 98%, which is possible with different combinations of window size and shift width. For example, if you have a small shift width between two windows, you can get the highest recall because the combined classifier, CCOMB, includes all the windows that a sensor sample is in. On the other hand, you can get a really small shift width and a really high recall if you have a really big shift between windows.

4.3 Combined Evaluation

Table II: Previous Works on A-Z Handwritten Alphabet

Work	Accuracy
AA[31]	98.47%
Dutt A [40]	98.7%
Ghosh AAM et al.[33]	98.08%
Ranjan Jana et al.[37]	98.85%

Table III: Previous Works on MNIST Dataset

Work	**Accuracy**
Nasir Uddin [29]	96.80%
Wen He[30]	96.91%
CNNAP [15]	98.18%

We observe mainly three results from our experiment. These are training accuracy, validation accuracy, and testing accuracy. After 10 epochs we get this result. Table IV shows our training, validation result as well as the testing result for un-weighted average accuracy.

Table IV: Average Accuracy

Algorithm	Training	Validation	Testing
CNN	99.47%	98.61%	98.97%
KNN, SVM	94.03%	92.65%	92.71%
DCNN	97.59%	97.05%	97.13%

The processing time of training is also measured. It takes 12 seconds per epoch. As there are a total of 10 epochs, the total training time is 10×12=120 seconds (about 2 minutes).

The accuracy vs. epoch graph in Figure 4 shows that the difference between train and validation accuracy is fairly small when trained with 50% dropout. The result is more generalized as a result.

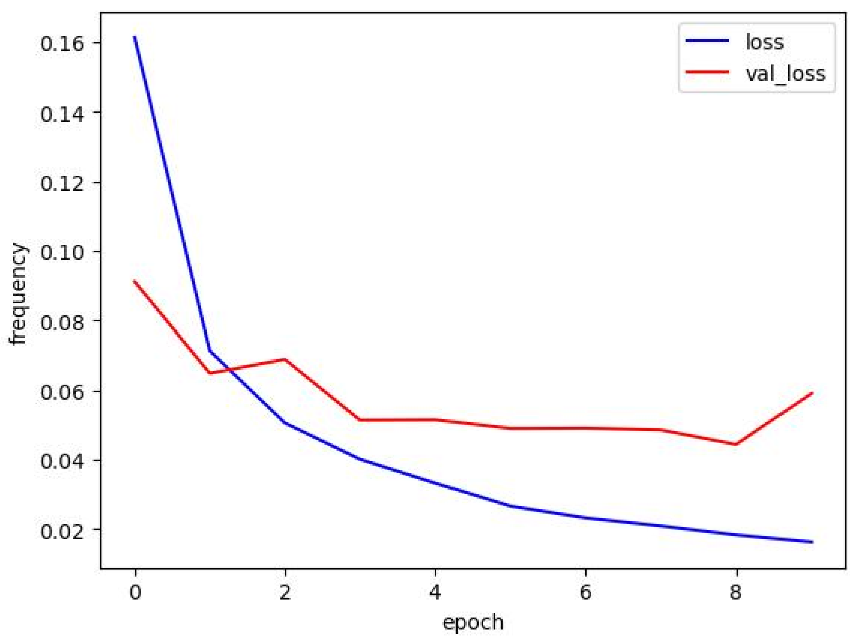

Figure 6: Frequency vs. epoch graph of CNN model on MNIST datasets

As illustrated in Figure 7, using 10% dropout, we can reduce overfitting and get 99.33 percent train accuracy and 98.92 percent validation accuracy.

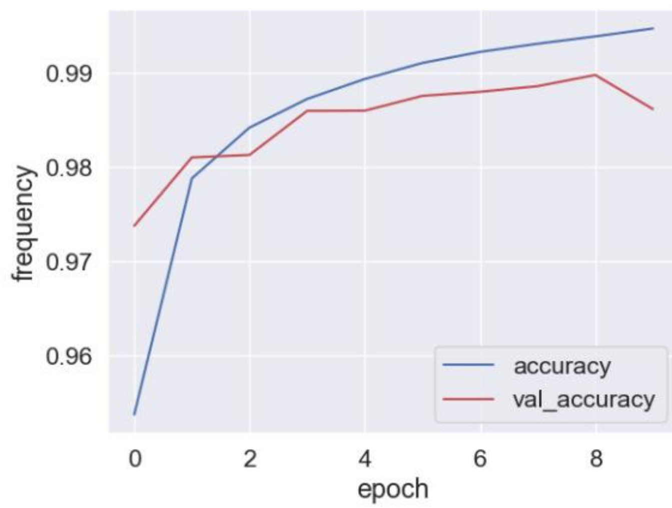

Figure 7: Accuracy vs. epoch graph on CNN model with two different merged datasets

CHAPTER FIVE

Conclusion and Future Work

5.1 Conclusion

In this paper, we show how unsupervised pre-training with an auto encoder can be used as a stepping stone to supervised training for English handwritten digit recognition. We evaluated this method using three distinct training settings on two separate datasets of Standard English characters to show its efficacy. The suggested approach yields accuracy rates for the test images of the A-Z Handwritten Alphabets dataset that are comparable to earlier research. For the A-Z Written by hand Letter set, it accomplishes a precision rate of 99.50%, which is the leading detailed result on this dataset so distant. Separated from demonstrating the utility of unsupervised pre-training within the setting of English digit acknowledgment, our project comes about too demonstrate that such pre-training can be valuable indeed when the information sets are autonomously and indiscriminately collected. In every experiment, the demonstrate with CNN gives superior exactness rate than the demonstrate with as it were ConvNet. The suggested method yields cutting-edge results for the MNIST dataset and excellent results for the A-Z Hand Written dataset also. It is worth saying that past considers on English manually written digit acknowledgment once in a while detailed on these two datasets together. Future research about this project can investigate whether pre-training on bigger datasets, which are not particular to English, can offer assistance accomplish way better comes about to be essentially valuable.

5.2 Future Works

Handwriting recognition technology has come a long way in recent years, but there is still room for improvement. Here are some potential areas for future work in handwriting recognition: improves accuracy, multilingual support, real-time recognition, and integration with other technologies, recognition of mathematical symbols and equations, support for low-resource languages.

References

1. Md Zahangir Alom,Paheding Sidike,Mahmudul Hasan,Tarek M. Taha,and Vijayan K. Asari,(Volume 2018, 27 Aug 2018); "Handwritten Bangla Character Recognition Using the State-of-the-Art Deep Convolutional Neural Networks”;
   DOI: https://doi.org/10.1155/2018/6747098
2. Omer Othman Ahmed Alsmany(Dec 2022);"Hand Written Recognition Characteristics”
3. Das, T.R., Hasan, S., Jani, Md.R., Tabassum, F. and Islam, Md.I. (2021) Bangla Handwritten Character Recognition Using Extended Convolutional Neural Network.

Journal of Computer and Communications, 9, 158-171;DOI: https://doi.org/10.4236/jcc.2021.93012

4. Khandokar, Md M Hasan, F Ernawan, Md S Islam, M N Kabir(2021 J. Phys.: Conf. Ser. 1918 042152);"Handwritten character recognition using convolutional neural network"; doi:10.1088/1742-6596/1918/4/0421526
5. Yintong Wang et al 2021 J. Phys.: Conf. Ser. 1848 0120156; "Offline Handwritten Text Recognition Using Deep Learning: A Review";doi:10.1088/1742-6596/1848/1/012015
6. Hamdan, Y. B. & Sathesh, A. (2021). Construction of Statistical SVM based Recognition Model for Handwritten Character Recognition. Journal of Information Technology and Digital World, 3(2), 92-107. doi:10.36548/jitdw.2021.2.003
7. Md. Ismail Hossain, Mohammed Rakib,Sabbir Mollah, Fuad Rahman and Nabeel Mohammed; "LILA-BOTI : Leveraging Isolated Letter Accumulations By Ordering Teacher Insights for Bangla Handwriting Recognition”
8. Katie Luo, Guandao Yang, Wenqi Xian(Feb 2022); "Stay Positive: Non-Negative Image Synthesis for Augmented Reality; arXiv:2202.00659v1
9. S. M. Azizul Hakim and Asaduzzaman, "Handwritten Bangla Numeral and Basic Character Recognition Using Deep Convolutional Neural Network; doi: 10.1109/ECACE.2019.8679243
10. Paheding Sidike,Mahmudul Hasan,Tarek M. Taha,and Vijayan K. Asari(27 Aug 2018); "Handwritten Bangla Character Recognition Using the State-of-the-Art Deep Convolutional Neural Networks";https://doi.org/10.1155/2018/6747098
11. M. Shopon, N. Mohammed and M. A. Abedin, "Bangla handwritten digit recognition using autoencoder and deep convolutional neural network," doi: 10.1109/IWCI.2016.7860340
12. M. A. Azad, H. S. Singha and M. M. H. Nahid, "Bangla Handwritten Character Recognition Using Deep Convolutional Autoencoder Neural Network," doi: 10.1109/ICAICT51780.2020.9333472
13. Md. Hadiuzzaman Bappy, Md. Siamul Haq, Kamrul Hasan Talukder (September 28, 2022);KUS: ICSTEM4IR-22/0166; "BANGLA HANDWRITTEN NUMERAL RECOGNITION USING DEEP CONVOLUTIONAL NEURAL NETWORK"; DOI: https://doi.org/10.53808

14. Shahrukh Ahsan, Shah Tarik Nawaz,Talha Bin Sarwar, M. Saef Ullah Miah, Abhijit Bhowmik(Jun 24, 2022);"A machine learning approach for Bengali handwritten vowel character recognition"; IAES International Journal of Artificial Intelligence (IJ-AI); DOI: 10.11591/ijai.v11i3.pp1143-1152
15. A. Srivastava and P. Singh,“Handwritten Digit Image Recognition Using Machine Learning,” Journal of Informatics Electrical and Electronics Engineering(JIEEE),Vol. 03, Iss. 02; Doi: https://doi.org/10.54060/JIEEE/003.02.003