HANDWRITTEN CHARACTER RECOGNITION USING DEEP LEARNING

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by:

PRAJJWAL KUMAR

2K20/CSE/15

Under the supervision of

Dr. Shailender Kumar

(Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY, 2022

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi - 110042

CANDIDATE'S DECLARATION

I, Prajjwal Kumar, Roll No. 2K20/CSE/15 student of M. Tech (Computer Science andEngineering), hereby declare that the project Dissertation titled **"**Handwritten Character Recognition**"** being submitted by me to the Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of and Degree, Diploma Associateship, Fellowship or any other kind of similarity in title or recognition.

Place: Delhi Prajjwal Kumar

Date: 2K20/CSE/15

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi - 110042

CERTIFICATE

I hereby certify that the Project Dissertation titled **"Handwritten Character Recognition"** which is submitted by Prajjwal Kumar, 2K20/CSE/15 Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carriedout by the students under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this Universityor elsewhere.

Place: Delhi

Date:

Dr. Shailender Kumar

Professor

Department of CSE,

DTU

ACKNOWLEDGMENT

The success of this project requires the assistance and input of numerous people and the organization. I am grateful to everyone who helped in shaping the result of the project.

I express my sincere thanks to **Dr. Shailender Kumar**, my project guide, for providingme with the opportunity to undertake this project under his guidance. His constantsupport and encouragement have made me realize that it is the process of learningwhich weighs more than the end result. I am highly indebted to the panel facultiesduring all the progress evaluations for their guidance, constant supervision and formotivating me to complete my work. They helped me throughout with new ideas,provided information necessary and pushed me to complete the work.

I also thank all my fellow students and my family for their continued support.

Prajjwal Kumar

2K20/CSE/15

ABSTRACT

Handwritten character identification is a topic that has been researched for years and is an area of interest for the community of Pattern recognition researchers since It may be put to use in a wide range of fascinating applications. all across the field. This subject is a difficult challenge as a task because each person has their own unique writing style. SVM, ANN, and CNN models are some of the available options for handling this problem's many different ways and approaches. HCR is a need in the modern world since it assists us in a variety of fields of public domain, which makes it all the more vital to study in depth. Off-line digit recognition and online digit recognition are both examples of the hybrid character recognition (HCR) category. In this study, we review the many existing algorithms that have been implemented to get the better knowledge of the course, and we will come to a conclusion on the best strategies that are currently being developed for HCR.

HCR for Devanagari is carried out by the performance of a computational device that accepts input from documents, screens, photos, and other responsive devices and believe to provides output by reading those images as an ASCII or UNICODE format. This theory is supported by the fact that computers have become increasingly powerful in recent years. Sanskrit, Nepali, Marathi, and Hindi are some of the languages that are represented in Devanagari. This script is a blend of numerous languages. This implementation is more important because the design of upper-case and lower-case characters in Devanagari are more complicated than in most other languages out there. Comparatively speaking, the set of characters and digits used in Devanagari is more complicated than the set of characters used in the English language. Character recognition has been hampered by the absence of verified datasets including Devanagari, which has made the task more difficult to do in the field.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

1. CNN: Convolutional Neural Network
2. DNN: Deep Neural Network
3. SVM: Support Vector Machine
4. FAQ: Frequently Asked Questions
5. Q-A: Question Answering
6. OCR: Optical Character Recognition
7. IP: Image Processing
8. DHCD: Devanagari Hindi Character Dataset
9. TF: Transfer Learning
10. KNN: K-Nearest Neighbor
11. MLPNN: Multi-Layer Perceptron Neural Network
12. MNIST: Modified National Institute of Standards and Technology

CHAPTER 1

INTRODUCTION

1.1 DEFINITION

HCR is the subfield of OCR that is optical character recognition, HCR is a current area of study that is being actively pursued, HCR is ever-growing field due to its immense and growing need. Digitally storing the handwritten papers or letters is a difficult task that takes a lot of time, and if the documents are in a large number, then it can take an extremely long time to store the data in hard drives or storages manually, which is where the need for handwritten character recognition comes into play. It is an area where one can work for the benefit of the public and the research community. Because there are not nearly as many datasets devoted to Devanagari characters as there are to other scripts, character recognition software designed to operate with them still has a long way to go. However, the language itself is quite difficult to read in comparison to other languages, such as English. HCR's primary purpose is to receive an input in the form of an image from a camera, scanner, file, or any other responsive device, analyses the picture to determine the character of a particular language, and then output the results in either the UNICODE or ASCII format [1].

The globe suffers from a shortage of datasets pertaining to Indic languages, which is one of the reasons why there has been so little progress made in improving the Devanagari script. As a result, this field is currently lagging behind the related issues of HCR. We utilized a dataset that is publicly accessible and called the Devanagari Handwritten Character Dataset (DHCD). This dataset contains around 92000 photos of 46 distinct Devanagari characters. In his study, we used this dataset. This dataset contains 2000 images for each character, which are further divided into two sets: training 1700 and testing 300 for each of the characters. In the beginning, MNIST served as the inspiration for this DHCD, but now this dataset is technically larger than MNIST. An example image of MNIST dataset given in 1.1 figure.

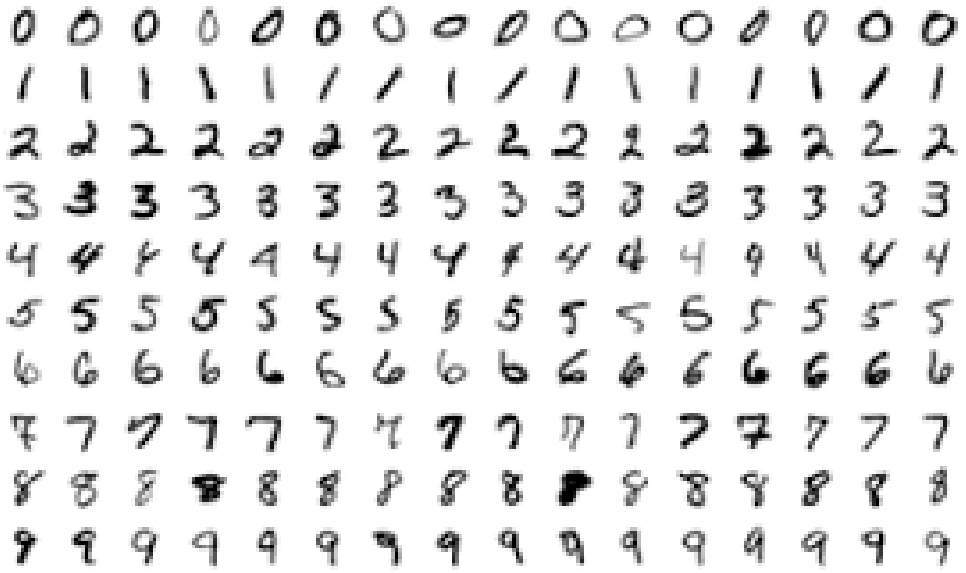


Figure 1.1: MNIST dataset character image

[2] All of the images got the magnitude of 32x32 pixels, whereas real digit is centered within a size of 28 x 28 pixels. All of the images got the magnitude of 32x32 pixels in order to achieve this progression in picture size, they padded the image with two more pixels on each side, giving them a value of 0. Figure 1.2 displays a selection of characters written in the Devanagari script.

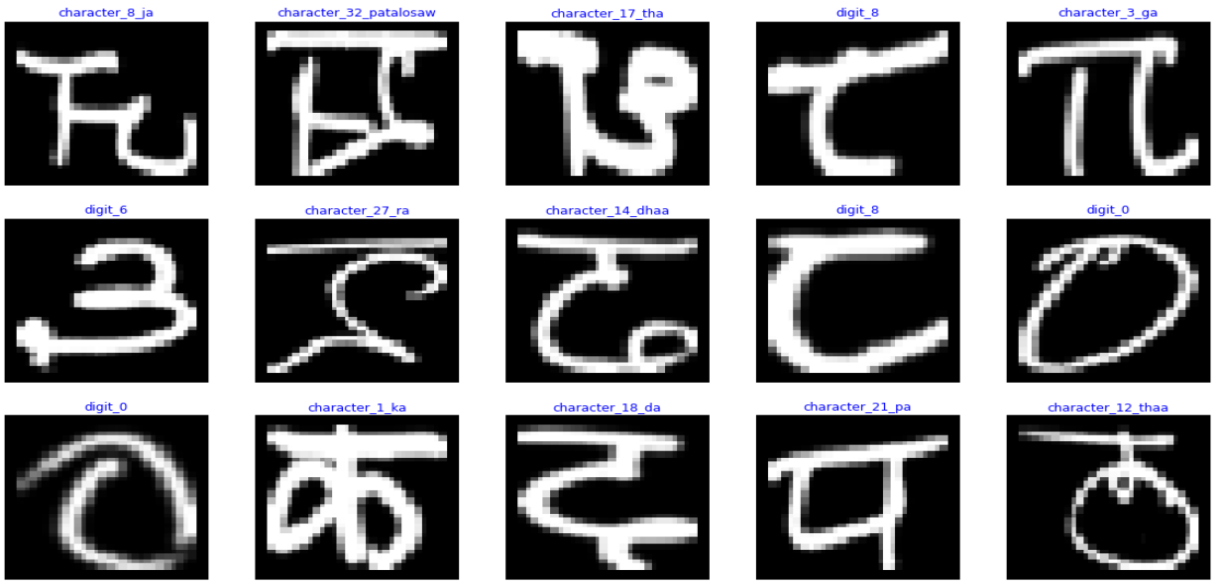


Figure 1.2: Images from DHCD [24]

Within the scope of this paper, we shall talk about a variety of approaches. One of them is called CNN, which is an abbreviation for Convolutional Neural Network and a category within DNN. DNN is fundamentally composed of a great number of nonlinear hidden(concealed)

layers. which makes it difficult to train because the trainable parameters are enormous and so is the number of connections in it. CNN, on the other hand, is relatively faster and cleaner than DNN and it takes less time in comparison to its set. In this article, we have utilized a class of KERAS known as Efficient Net B2 for the purpose of preprocessing the input; this is followed by the process of transfer learning, and then two dense layers are used to measure the output according to the specifications laid forth in the study. The primary objective of pattern recognition is to translate human perceptual processes into computer-based models. It is astounding how we can read any handwritten word or machine printed language; even now, artificial intelligence can scarcely make up for it. When we talk about the visual proficiency of humans, it is important to note that this ability is not shared by other animals. For the same reason, it is necessary for this subject to be researched extensively. However, as we are well aware, there are several commercial solutions available to address all of these OCR-related issues; this demonstrates that the technology has reached a very mature stage in this sector. However, due to the fact that many languages use a variety of special as well as lowercase and uppercase letters, not all methods can be utilized in order to understand the images. Because certain languages are difficult to recognize, such as Roman or Chinese, and even Arabic letters have too many dots, which makes it tough for mapping, there is a need for a systematic method to extract the characteristics and characters.

1.2 Relevant theory

The process of Hand Written Character Recognition (HWCR) often consists of many steps. Here, the first one is the Image Acquisition, and in this state, we take all of the possible images and obtain the input either by taking a picture with a camera or phone, or we can take it by drawing the image on a sheet of page, and after that, we can scan that using a scanner or the camera on our phone. Creating drawings with the light stylus is yet another type of image input that may potentially be used.

The subsequent step in the process is called pre-processing, and it is at this phase that the quality of the photos is enhanced and improved. After applying various procedures to the input photos, which may include thinning, skeletonization, normalizing, skew correction, noise removal, filtering, and binarization, During the process of preprocessing, we provide the photographs a higher standard of quality by improving and enhancing them.

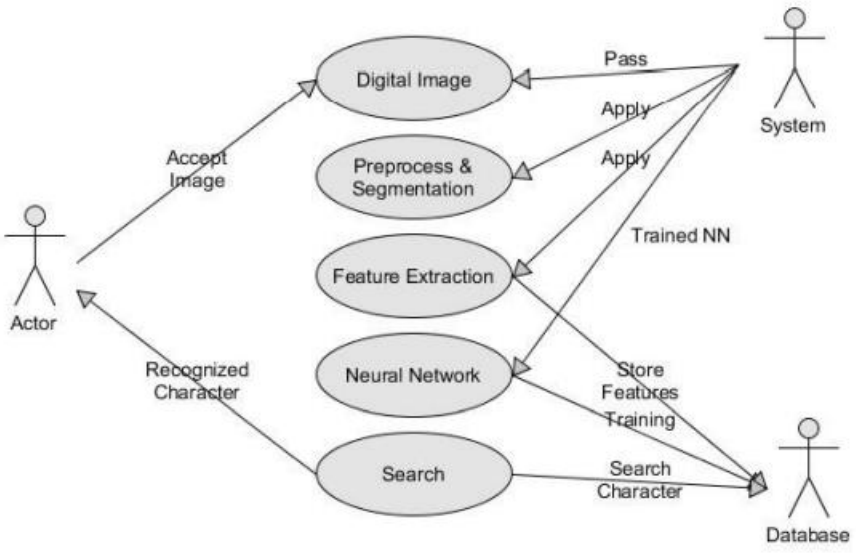


Figure 1.3: Use case Diagram

The third stage of the system is called segmentation, and it is important for decomposing the input picture into meaningful pieces. Additionally, this procedure helps to separate the many things that are displayed in the object. A basic function shown in figure 1.3. Therefore, we are able to define segmentation as the condition in which the input is broken down into subparts, and each subpart is defined as an object [3].

Throughout the subsequent step, we will be classifying the objects that were produced during the process of segmentation in order to make a determination regarding the category of item to which each individual object belongs [4]. Therefore, when we are in this stage, it is possible for us to have classes that are dependent on the object. This is because numerous classes may be developed for categorization, and as we obtain the properties of the entity, we will be able to assign a class to all of the objects.

For illustration's sake, let's say we have a class called "car" and another class called "plane." At this point, any item that exhibits vehicle-like qualities must be assigned to the "car" class owing to the traits [5] it possesses. In the same way as this system, this review paper investigates the many approaches to HCR that are already in use and discusses both the benefits and drawbacks associated with each of them.

The use of a deep neural network (DNN), which, in addition to being an efficient feature extractor and classifier [6], as well as one of the many approaches that can be used, also happens to be one of the many approaches that can be taken to address this issue, is one of the many ways that this problem can be solved.. However, there is a catch, and that is the necessity for an excessively long period of time to train the network. This is because the network has a significant number of nonlinear hidden layers, in addition to certain connections. While Deep

Neural Networks (DNN) were developed to solve these challenges, Convolutional Neural Networks (CNN) were developed to solve these problems by implementing nonlinear hidden layers in a smaller amount compared to DNN [7].

This is the key reason that we use CNN to extract the characteristics of position-invariant. CNN has a simpler structure compared to DNN, which is why we employ it. Because it provides the user with temporal subsampling and allows for a degree of rotation, shift invariance, and distortion [8], a map may be constructed between the input picture or dataset and the output dataset through the usage of a convolutional neural network with relative ease.

Fans of machine learning and data mining have already put in a significant amount of work to improve their chances of successfully approximating pattern recognition [9]. Back checks are a perfect illustration of how dependent we are on HCR as a medium, which takes the information and lets us communicate with others. HCR has a significant effect on the way we live today because of how dependent we are on it.

The same debate occurs whenever we make use of the distortion of the HCR. This is due to the fact that various locations house a variety of languages, and these locations will also provide a variety of handwriting styles and methods. As a result, it is somewhat challenging to maintain control and extract the character from the language that is provided [10].

1.3 LITERATURE SURVEY

According to the information provided in article [14], they have included transfer learning and VEGG19NET into their implementation. This combination has been tested using the dataset DHCD and Kannada characters, and it has enabled them to claim an accuracy of 87 percent.
As we learn about DNN, we also learn CNN with it because CNN is a subpart of DNN only, and paper [15] has used the Deep CNN technique to report their implementation work. They were also the first to use this publicly available dataset in 2015, and as a result, they registered an accuracy of 98.47 percent.
This [16] study investigates offline handwritten character recognition and uses the same database, which is DHCD. Their method was CNN with grey scale conversion employing overlaying of picture, and finally, they recognized it; the accuracy they attained was 98.94 percent.
The paper [17] also discusses the Deep Neural Network on DHCD. The authors structured four architectures for their implementation part, each of which contained CNN. One of the

architectures contained one CNN layer, while the other contained two CNN layers. The final two architectures each contained three CNN layers with a 3x3 filter size to achieve an accuracy of 97.86-99.29.

This study [18] makes use of a different approach; in fact, it has two techniques inside it. One of them is inceptionV3, which is a slow technique but offers an accuracy of 99; AlexNet, on the other hand, has an accuracy of 98 and is somewhat faster than the second technique.

This author [19] uses its CNN as a combination of ReLu conv1 with the character caps in its architecture; however, the amount of time required for its each epoch is longer than the typical amount of time required by the other algorithms; despite this, the accuracy was very high, coming in at 99.07 percent.

Now, the technique described in this study [20] makes use of SVM, which achieves an accuracy of 95.65 percent. In this particular application, they utilized an SVM classifier with three distinct features, which are referred as LBP, HOG, and LOOP. The most important thing I learned was that if we employ some of the region attributes of the digits in conjunction with the HOG feature, the accuracy may be enhanced much further.

This [21] author uses the techniques of random forest, KNN, and extra tree for its implementation on DHCD. He reported a score of 75.48 for KNN, 77.04 for random forest, and 90 for the extra tree methodology. These are some of the procedures that have been tried out on this dataset in the past by a number of different researchers.

In this section, you will find a concise description of the research or work which is being made over the years on HCR. M. Ramzan et al. [9] conducted an in-depth analysis and assessment of an HCR research that exclusively utilized neural networks. Their research offers a comprehensive analysis of neural networks, which explains information on HCR in great depth, including the benefits and drawbacks associated with each approach that uses it, as well as the overall advantages and disadvantages.

M. Agarwal et al [10] have conducted study on the numerous languages that are spoken in India with reference to the handwriting recognition of these languages. This research can be found here. They evaluated several datasets using various approaches that produce a range of findings and came to the conclusion that SVM delivers an accuracy of 99.6 percent, while CNN provides an accuracy of 98.47 percent.

The article by P. C. Vashist et al. [11] analyses and contrasts all of the available techniques for Handwritten Character Recognition. These techniques include SVM, KNN, MLP, Back propagation, and Distilling GRU. The authors present both the benefits of the approach and the drawbacks of the method.

This study by F. Siddique et al. [12] uses the MNIST dataset and attempts to install CNN on it using various hidden layers each time. This causes the study's findings to fluctuate and also alters the amount of time required for training. The iterations yielded the best result possible, which was an accuracy of 99.21 percent on the 15 epochs that were tested.

A. Garg et al. [13] developed a CNN model that is a class of DNN. This model has four convolutional layers in the first two levels, 32 filters with a size of 5X5 and 64 filters got the magnitude of 3X3 and 64 in numbers in the layers that follow. The accuracy of this model was determined to be 98.45 percent.

References	**Dataset**	**Classification**	**Evaluation Measurement**
[13]	MNIST digit Recognizer	CNN model using 4 convolutional layers	98.45
[27]	MNIST	SVM with RBF kernel	99.1
	Oriya numerals		97.2
	Devanagari numerals		98.54
	Telugu numerals		96.5
	Bangla Numerals		98.375
	Bangla basic characters		95.6
[30]	MNIST	SVD with LeNet-5 imp. model	99.03
[31]	IAM off-line HTR	Combination of CNN and RNN	99.67
[32]	MNIST	Decision tree classifier,	90.37
		SVM	95.88
		Random Forest Classifier	93.85
		Naïve Bayes	90.83
		KNN	86.58
[14]	DHCD	VGG19NET + transfer learning	87
[15]	DHCD	Deep CNN	98.47
[16]	DHCD	CNN	98.94
[17]	DHCD	CNN	97.86-99.29
[18]	DHCD	InceptionV3(slow)	99
	DHCD	AlexNet (fast) + transfer learning	98
[19]	DHCD	CapsNet	99.07
[20]	DHCD	SVM with LBP, HOG and LOOP	95.65
	Odia	SVM with LBP, HOG and LOOP	99
[22]	DHCD	KNN	75.48
	DHCD	RF	77.04
	DHCD	Extra Tree	90

Table 1.1: Comparison Between previously used techniques

R. Jayadevan and colleagues [29] present a comprehensive analysis of bank cheques, which they analyzed in great detail and for which they compiled all of the essential information in a variety of languages, including English, Hindi, Chinese, and French. They evaluate the precision of the extraction process across all of the languages by comparing the results from each language. as shown in table 1.1.

CHAPTER 2

RELATED WORK

2.1 Description of some publicly available Dataset we used

MNIST

MNIST is a large database that is made up of grayscale pictures of handwritten single digits ranging from 0 to 9 that are square in shape and have a pixel size of 28 by 28. This is comprised of a total of 70,000 handwritten drawings of digits, of which 60,000 are used in the training set and 10,000 are shown in the test set respectively. [23]. In the event that you become confused, there is a label next to each picture that explains which digit it represents. There are a total of 10 different digit groups that may be differentiated from one another (from 0 to 9).

DHCD

As our dataset, we have utilized DHCD, which stands for Devanagari Handwritten Character Data, and it has 92000 pictures. This research presents a comparison between the many current strategies and the transfer learning for Convolution Neural Network, which is the methodology that we, the authors of this dissertation, utilized. The DHCD dataset includes 46 classes of the Devanagari script (.+.), which break down as follows: 36-character classes and 10-digit classes [24]. Each class comprises a total of 2,000 photos, which are separated into two distinct sets: The training set has 1 700 different pictures, whereas the test set only has 300 different pictures. Therefore, technically speaking, this dataset has a bigger number of samples as well as classes than the well-known MNIST dataset, which served as the initial motivation for the construction of this dataset. This dataset was created using the MNIST dataset.

2.2 Review of Recent Techniques

The following is a rundown of the numerous approaches that might be taken in the quest for improved handwriting recognition:

2.2.1 Back Propagation on Handwritten Test Image Authentication

Because the input picture cannot be utilized by the NN, this approach involves transforming it into a certain matrix (or text), as the original image cannot be utilized. After then, the input is altered, and new values for RGB, often known as red pigmentation, green pigmentation, and blue pigmentation are provided, which is being depicted in figure 1. Then, we will apply a normalizing function to produce these values in normalize mode, which then displays the pixel's color when using only this approach, as defined by P.C. Vashist and his colleagues [11].

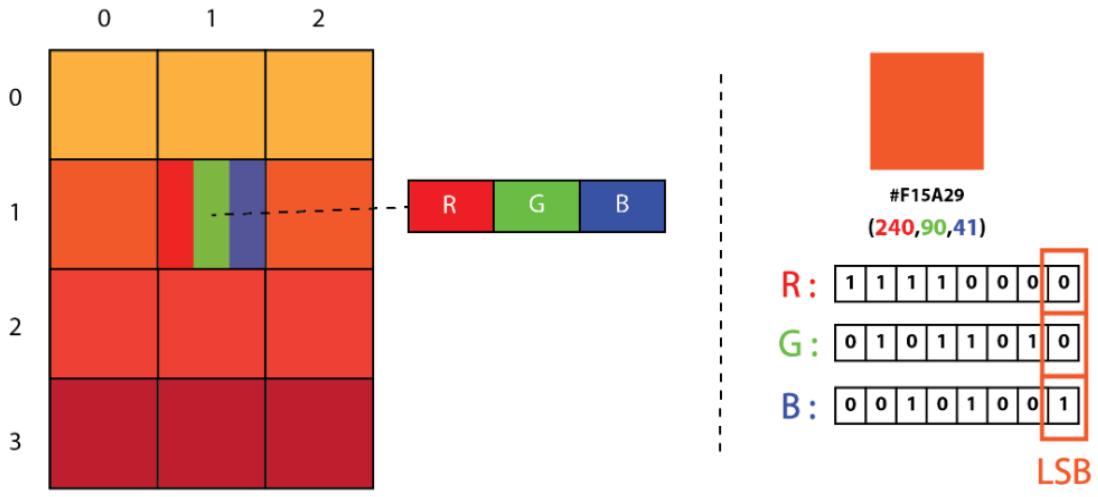


Figure 2.1: extracting pixel values

Following the completion of the conversion, the picture that was supplied will be converted into a matrix that contains numbers. This matrix may then be sent into a neural network, and the picture itself can also serve as a training example for the neural network to learn from. In order to demonstrate the work that has been done, we will develop an equation. 2.1

$$Cn = Ct/255 \quad (1)$$

Equation 2.1

Here, Cn is the result produced from the equation once it has been solved, and Ct is the value of the RGB components as this one function represents them.

2.2.2 Distilling Gated Recurrent Unit for handwritten text recognition with augmentation of data

The system must project a route or trajectory that is sequentially pen-tip, and it must highlight the charts by employing eight-dimensional and path signature, feature extraction in CNN, and LSTM in order to complete the processing. They are also able to record the sequential route of the pen tip, which may be dynamic and has also been utilized for character recognition or text recognition that does not require any pre-processing Over-segmentation and feature extraction are the two methods that are used to attain this goal. To put things another way, this method was only applied to the recognition of horizontal characters, and it did not take into account the characteristics of a typical manuscript, such as overlapping writing, vertical handwritings, multi-line handwritings, or right-to-left Handwritten text recognition (as shown in figure 2.2).

Horizontal ADIJSJJJFASD	Vertical N G L P O	Right-down S X L J
Screw-Rotation AB CD G F K	Multi-Line A B B C D DH J JD NK JS K G KF DS K J N L	Overlap

Figure 2.2: Unconstrained HCR which includes multi-line, horizontal, screw-rotation, vertical and right-down

A distillation with many layers The GRU system will be taken into consideration in order to provide the user with the following benefits, which will be made possible by the usage of

unrestricted handwritten characters. As also proposed by MLD Gated recurrent unit to get epitome model, the unconstrained pen-tip will move to state the input information consecutively. This will assist us in accelerating the convergence phase, and there will be no discernible loss of precision as a result of this change.

2.2.3 Recognition of handwritten text accomplished with the use of KNN, SVM, and neural networks

Python programming language, OpenCV vision library, and Sklearn library are some of the most up-to-date technologies that may be used to learn and build the dataset for these methods. And in order to do this, the training and classification are carried out based on the findings of MNIST. MNIST has compiled a dataset on the recognition of handwritten characters. In MNIST, which stands for "Modified National Institute of Standards and Technology" and is a small part of NIST, we have approximately 70000 total images of characters, of which 60000 are used for training and 10000 are used for testing. The format of each image is 28x28 pixels, and the classification algorithms that are provided are discussed in more detail below.

2.2.3.1 K-Nearest Neighbor algorithm (KNN)

When we talk about KNN, the first thing that stands out about it is that it does not have any particular phase of training. This is due to the fact that it works between the difference of two unidentified points by determining in the same class which instances are the closest ones. As can be seen in figure 2.3, a vote is taken in order to determine which data item should be considered the winner. KNN may be computationally expensive at times because if the data set is enormous, then it is necessary for KNN to store the data that is utilized for processing. This is not something that occurs with other algorithms, which is why it is essential to scale the data set before running KNN.

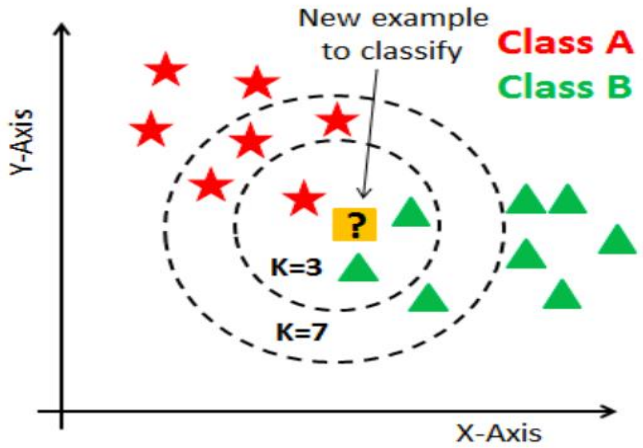


Figure 2.3: shows the categorization using KNN.

The MNIST dataset is utilized for the method in such a way that 70 percent of its entries go to the training phase and 30 percent go to the testing phase. After the training phase, the optimal value of the k is determined based on the dataset.

2.2.3.2 Support Vector Machine (SVM)

The context of distribution and regression issues, this model is a linear representation of the data. This may work for both linear and non-linear issues, and it works very well for both types of problems. Additionally, it works very well for practical situations and performs very well overall. This technique distributes lines, or what we may call a hyper plane, as the model classifies the data into contrasting respective classes. This function of the SVM is fundamental and straightforward. The data element is represented in N-size space by a specific dot (where

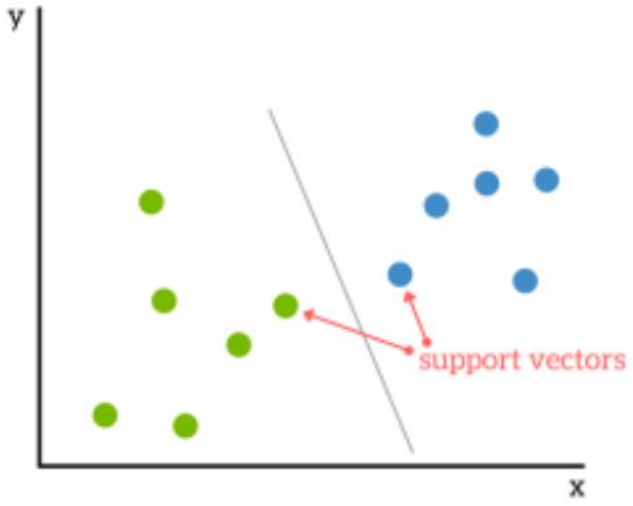


Figure 2.4: Classification using the SVM

n is the number of characteristics), and the values that are shown against it is the co-ordinates of the same. After then, it was discovered that there are two separate groups, which are separated by a hyperplane, as shown in figure 2.4.

After that, we used both sparse and dense sample vectors as input into the scikit-learn programme.

We are given with three different techniques inside scikit-learn that have the capability of determining numerous classes. The first is called SVC, the second is called LinearSVC, and the third is called datasets of NuSVC.

LinearSVC are typically utilized for the purpose of classification of MNIST datasets. This is due to the fact that LinearSVC is able to provide excellent accommodation for choosing penalties as well as the loss functions, and MNIST include a sufficient number of samples for

the algorithm to evaluate.

A download of the MNIST is performed, and the dataset is then extracted from the HOG, which stands for histogram for oriented Gradients (HOG). After that, we adjust the size of the cells to be 14 by 14. Additionally, it destroys the set for the 28x28, which is a vector of orientation with 9 elements. This results in a HOG vector that is 4 by 9 in size.

2.2.3.3 Neural networks with several layers of multi-layer perceptron (MLPNN)

It is an addition to FFNN and operates as an algorithm that is supervised. The input, the output, and the hidden layer, which is seen in figure 5, make up its three-layered structure. When it comes to training the neurons of MLP, the algorithm for back propagation learning is passed down here. It is designed to work for any continuous function and produce results that are non-linearly separable.

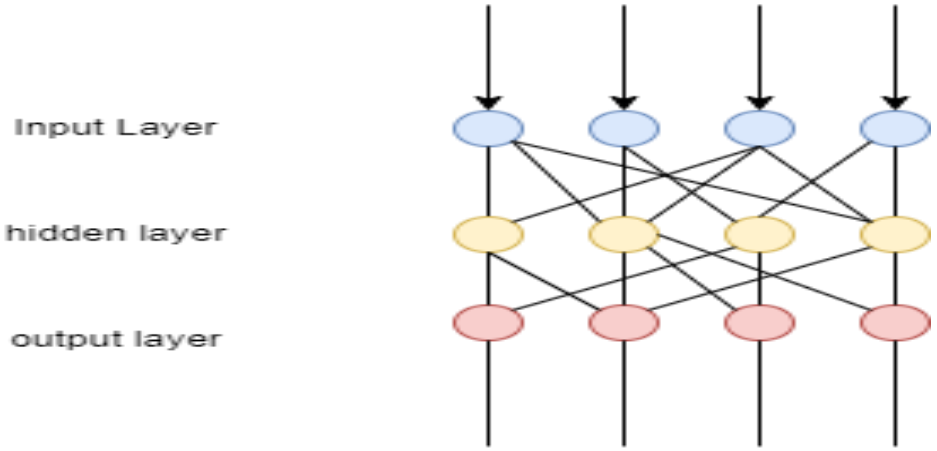


Figure 2.5: Categorization of MLPs.

It is distinct from logistic regression due to the fact that there may be one or more than one layer that is non-linear in between the source and feedback layers. The layers that is labelled as non-linear are referred as hidden layers.

Function has been updated to HOG in order to make it compatible with the MNIST dataset. The dataset together with the picture input that is included in it is stored in the NumPy array of characters. Both of the previously mentioned methods, KNN and SVM, will correctly address the dataset; however, when using an MLP neural network, it is an error to address 9. This is due to the fact that the function for MLP is non-linear, whereas in the previously discussed methods, the prediction is made based on the feature extraction itself. Therefore, it works well for non-linear models. In addition, MPLs include buried layers that contain more than a local level's worth of minimum contrast, which is a nonconvex loss characteristic [11].

CHAPTER 3

TECHNOLOGIES USED

3.1 DEEP LEARNING

It belongs to the field of machine learning that focuses on ANN and feature learning being its central techniques. Instead, then carrying out a series of orders that have been pre-programmed, DL algorithms could be able to learn from enormous volumes of data. DL stands for deep learning. Deep learning has been applied in a variety of frameworks, and these frameworks have shown competitive results in a variety of domains, including speech recognition, NLP, computer vision, and image processing of medical insurgence, amongst other domains. Increasingly, it is becoming clear that network traffic inspection and intrusion detection are two promising applications for deep learning.

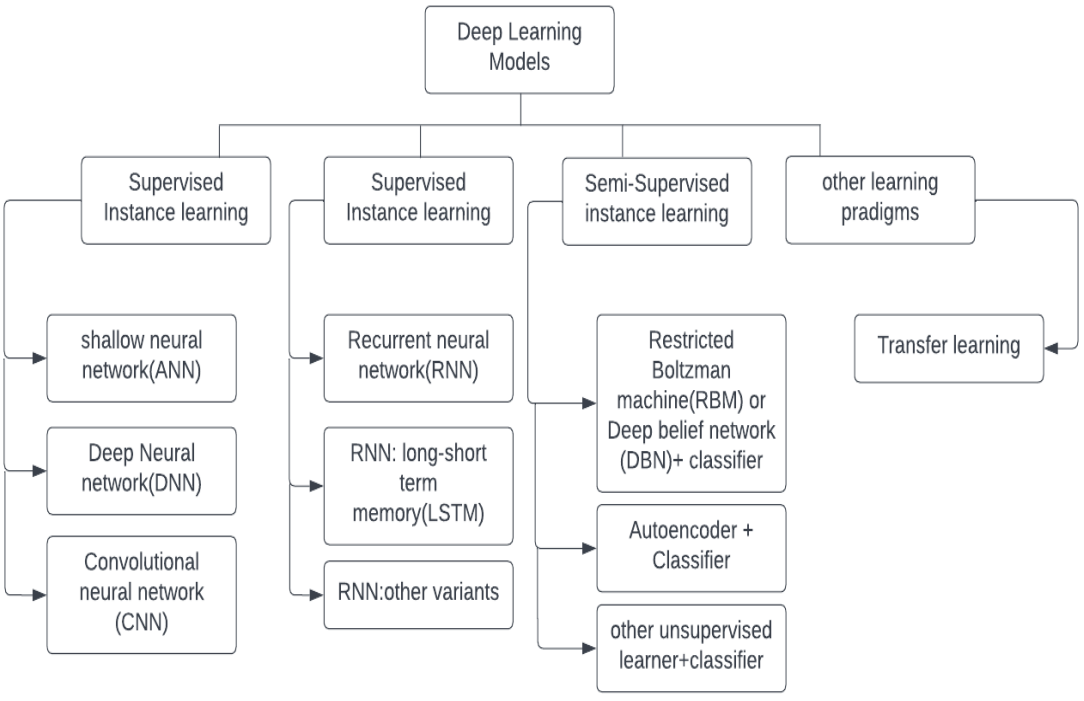


Figure 3.1: Various DL methods

Over the course of the last several ages, a great lot of progress has been made in the creation of DL models for the protection of network devices. This protection has been a primary focus of the development of DL models. The different DL Methods are depicted above in Figure3.1.

3.2 Convolutional Neural Network

CNN is yet another clever approach that is surrounded by the basics of deep learning, and it is utilized to develop a quick system for classifying pictures. CNN is a technique that is employed. In a similar vein, the CNN model can be of use in the development of secure systems. A conventional neural network is similar to the CNN approach in that it is composed of four layers: an input (source) layer, a convolutional layer, a poolinglayer and a Dense layer.

3.2.1 Convolutional layer:

The convolutional layer, which is comprised of several convolution kernels, is responsible for exploring and filtering the training sample. The "weighted summation kernel layer" is regenerated together with the "weight matrix" of the fed data set by the convolutional layer, which is responsible for producing the "weight matrix" of the fed data set.
In order to decrement the magnitude of the input, the filter makes use of integer values.

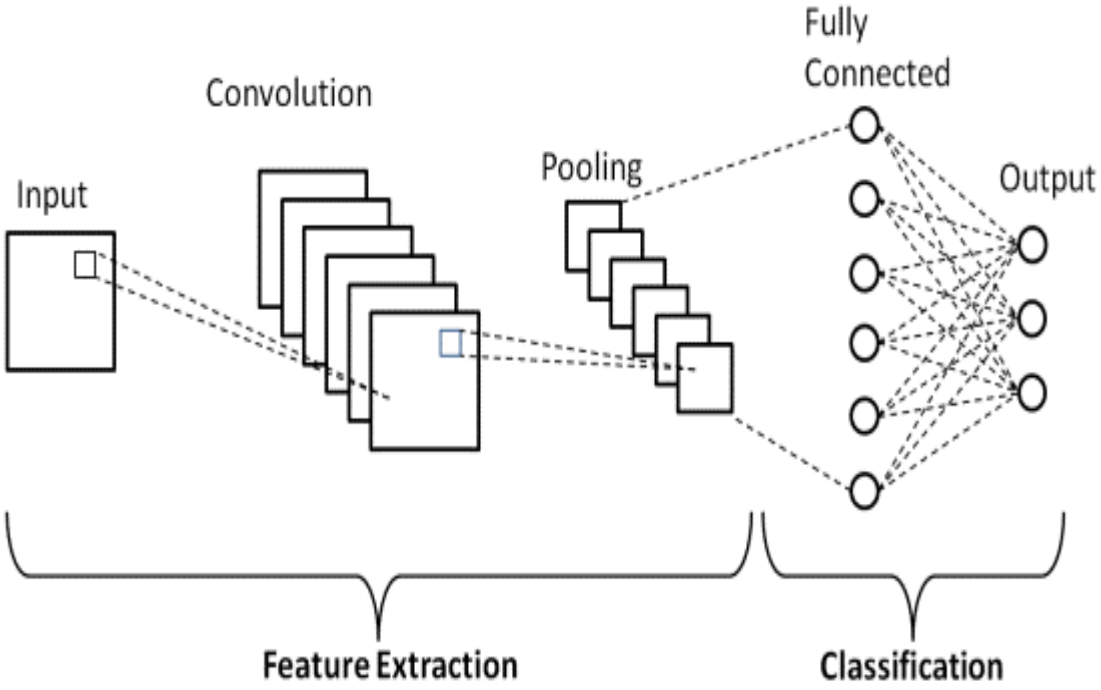


Figure 3.2: CNN layers

Figure 3.2 provides a more in-depth look into the CNN's layers. The performance of convolutional kernels may be improved by adjusting three crucial hyperparameters: the filter

size, the zero padding, and the stride. Choosing the values that are ideal can help reduce the amount of complexity that the network has, which in turn can improve its accuracy.

Equation 1 depicts the operation of convolutional layers, in this w is shorthand for the weights, x is shorthand for the input, b is shorthand of the basis and f is an activation function.

$$x_i = f\left(w_i \otimes x_{i-1} + b_i\right)$$

Equation 3.1: operation of convolutional layer

3.3 POLLING LAYER

This layer is utilized for only one purposes and that is to construct a fit matrix and decrementing the integer of characteristics contained within feature map. This is accomplished only if we select the peak value for each individual region. The subsequent layer will perform operations on this matrix. The size of the pooling area may be changed, and for the purpose of our research, we chose 5. The polling system may be implemented with either the max function or the average function. In order to reduce the number of features, maximum polling chooses the value that was highest, whereas average polling calculates the average of each region individually. Figure 3.3 is a depiction of the architecture of the polling layer.

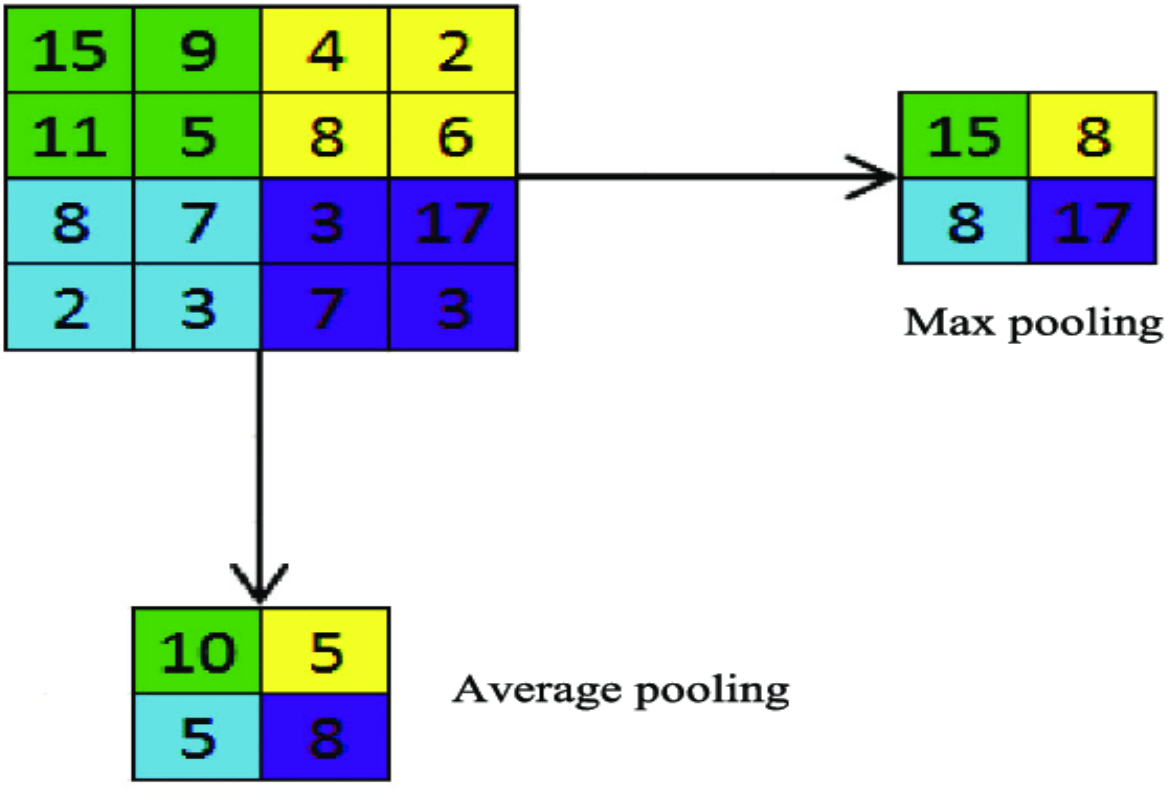


Figure 3.3: Pooling layer

3.4 Fully connected layer:

The final stage of the CNN is known as the Dense or fully connected layer. Every unit node in of that particular layer is in state M that has to be the part of the fully connected layer has a direct link to every unit node in both layers that are one level below and one level above it. In contrast to conventional ANNs, there is no connection between the nodes that make up the same layer.

As a direct consequence of this, this layer requires a significant amount of training and testing. The FC layer of the CNN is seen in figure 3.4

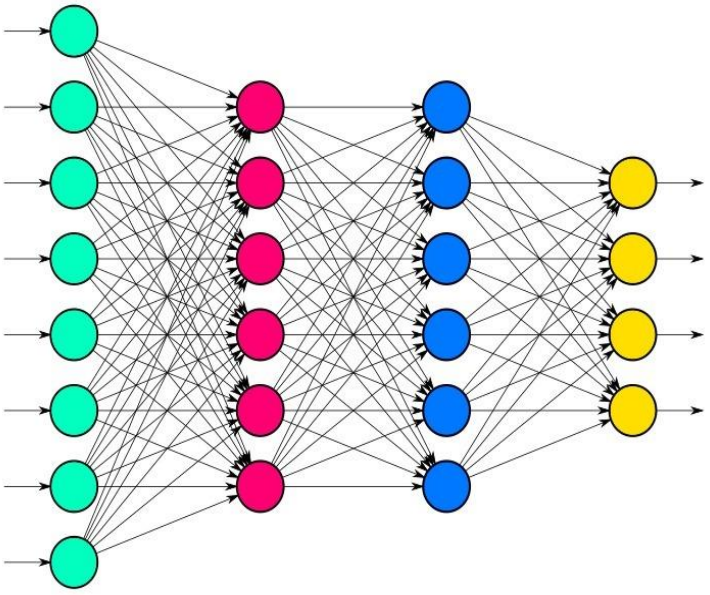


Figure 3.4: Dense or Fully connected layer

3.5 SoftMax Function:

The SoftMax function switches a vector of K values which are real into another vector of K values which are real in which all of the summarize up to 1. It does this by taking the original vector of K real values. The SoftMax algorithm transforms the input(source) values, that may contain validates or negated, could be zero or could be more or less around a single value, and those values are:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Equation 3.2

values ranging from 0 to 1, with the expectation that they will be understood in terms of probability. The formula for it may be found in the previous paragraph, in equation 3.2. The

values that are entered can also be greater than one. When given source input is very little or negative, it changes it into a low odd, and when given source input is very large, it transforms it into a high odd; nonetheless, never exceed 1 and will always be somewhere between 0 and 1.

The layer that produces real-valued outputs is often the final layer in multi-layer neural networks, as this layer is typically the last one in the network. As a result of the fact that these ratings are not just scaled, dealing with them might be difficult. That since it transforms the scores into a probability distribution that has been normalized., the SoftMax is a particularly useful tool in this context because of the nature of the issue. After then, this distribution seems to be exhibited by particular individual and valued by different setup. Both of these options are available. As a consequence of this, it is common practice to append this function at very top layer of the NN, as it serves as very final layer.

CHAPTER 4

PROPOSED WORK

4.1 PROBLEM STATEMENT

The application area of HCR includes many different industries and departments, such as hospitals, banks, the health care industry, electronic libraries, multimedia databases, and public work departments. These are all places that require handwritten forms or papers on a regular basis in order to finish their work. HCR is an active field because it assists the public in making documentation run smoothly. As we all know, in both the public and private sectors, in order to gather the required information, they all use documents so that all of their patients, clients, or customers can provide the checks, letters, and forms. This is why HCR is such an important field. After collecting all of these paper bands, they are all converted into digital pieces of information and stored as they are. As is well known, the standard procedure to feed a document into a computer hard drive is to manually input all of the entries, which can be a time-consuming process. Additionally, it is possible to make mistakes while entering all of the data in it, and if there are a large number of documents, it can be exhausting. Because of this, we want software that can detect handwritten characters. This programme will be able to read the text from the images that are entered of the various articles. The laborious task of entering the data into the digital form may be accomplished in a practical manner by employing software that is connected to HCR.

4.2 PROPOSED METHOD

Transfer learning, EfficientNetB2, and two thick layers are the essential components of our implementation architecture, which is essentially composed of all of these.

As a result of the following, which is followed by a little discussion on the subject, we will learn more about these:

4.2.1 Transfer Learning

It is a technique used in machine learning in which the model that was produced for one job is utilized for another task, but this time it is used as a starting point for the task that comes after it. In more straightforward language, we can explain that it is a method of machine learning in which we may as a new starting point of model by giving it a pretrained model working on a new job. This enables us to make quick progress when we try to operate for the new model. Because transfer learning is used so frequently, it is quite uncommon to train a model for a task related to natural language processing or image processing from the ground up.

In the well-known method of deep learning, models that have already been trained are used as the starting point for NLP and the vision of computer. This is because developing NN models requires a significant amount of time and resources, as mentioned in [12]. Because of this, there is an increased demand for learning that is transferable in this industry. The fundamentals of how transfer learning operates are defined in Figure 4.1.

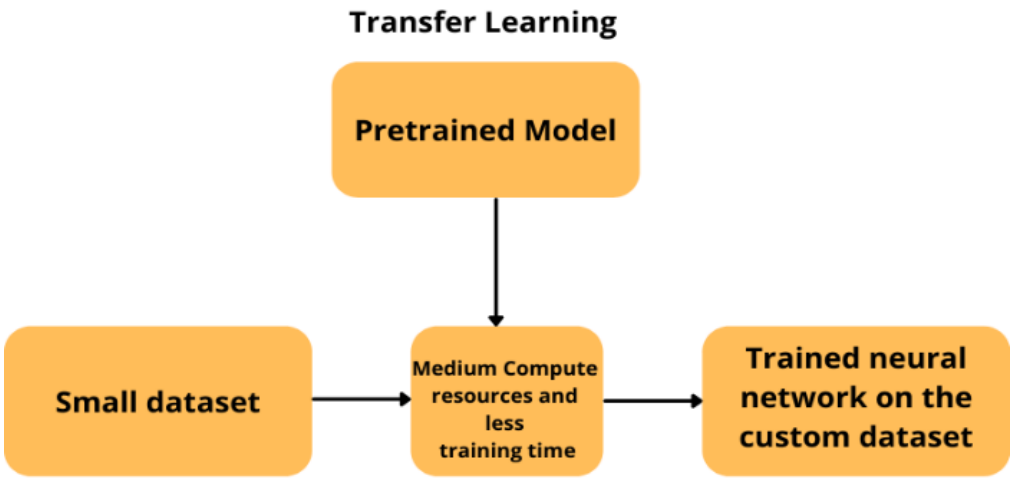


Figure 4.1: Transfer learning utilizing Efficient Net B2

4.2.2 Dense layer

In neural networks, a dense layer is defined as the layer is being deeply defined and connected from that state below. This indicates all neurons in the certain state are concatenated to each and every layer’s neuron and above them too. This method is often utilized in artificial neural networks.

In this layer, each neuron performs matrix-vector multiplication on the outputs of the preceding layer's neurons, using the results that were given to it by the previous layer's neurons. MVM is a procedure in which the number of rows in a vector representing the result from the layer below it corresponds to the column vector of the layer that is dense.

This layer includes a variety of parameters, some of which include units and activation, while others, such as kernel initializer and Bias constraint, are among the others. These parameters all have the potential to alter the processing flow. When we make use of dense layers, there is no need for us to redefine the input layer for the second dense layer. This is because the neurons of opening dense layer will function as source or input for the dense layer that comes after it, which allows for quick progression as defined in figure 4.2.

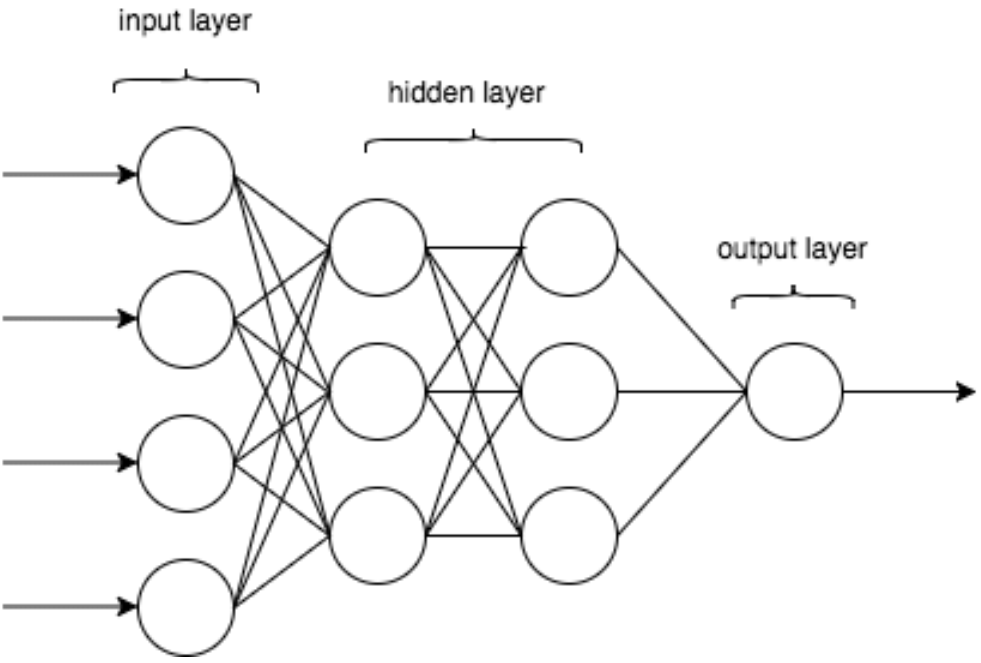


Figure 4.2: A basic architecture of dense layer.

4.2.3 Efficient Net B2

Efficient With the assistance of a compound coefficient, the CNN known as Net B2 — which has the formation of a CNN and a way of scaling — measures consistently all three dimensions of depth, breadth, and resolution. The standard CNN scales all of the components arbitrarily, but the Efficient Net scales all of the parameters and resolution in the same manner throughout. This approach makes use of the coefficient of compound in order to measure all parameters consistently and in a primary manner. These methods were revealed in 2019 and represent the new direction that CNN is heading towards.

The effective Net B2 algorithm provides a few arguments and parameters that may be used to pass any dataset's value. Include top, weights, input shape, pooling, input tensor, and classes are some of the arguments that are accepted, and this gives back an instance of the Keras model.

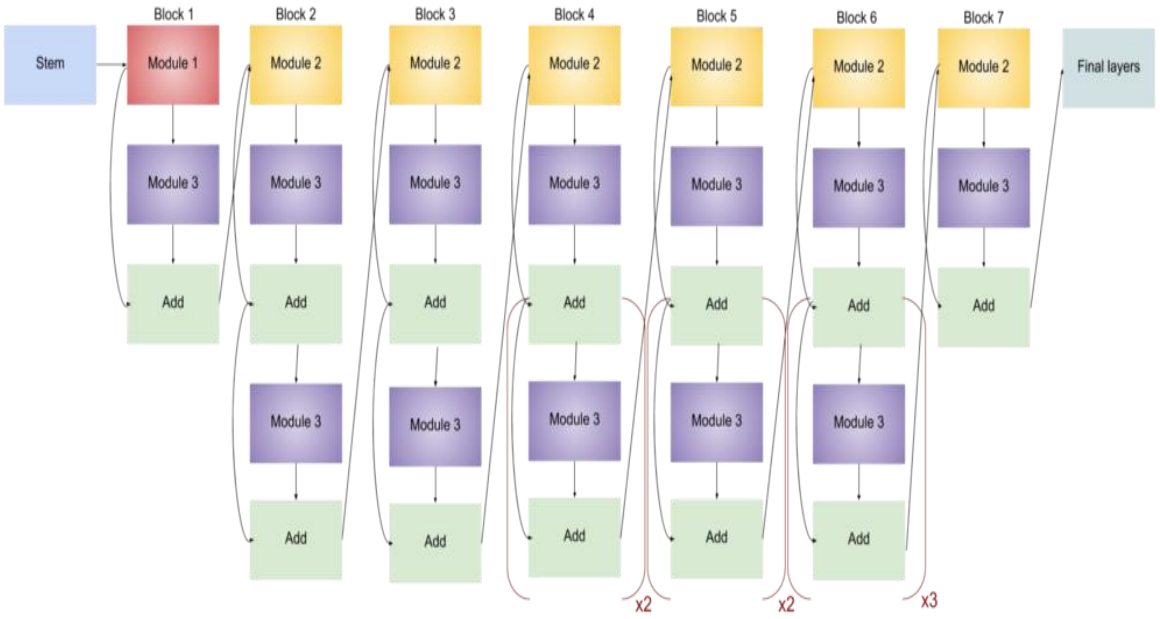


Figure 4.3: Layers of Efficient B2

There are seven different module blocks in an effective network design, in addition to an initial and a final layer as driven in above image or figure 4.3. The summed up amount of layers in efficientNetB2 with keras is 342, which allows for the data to be trained and tested in accordance with their proper context.

4.3 Flow Diagram of Proposed Model

The flow diagram of our model shows how we have used transfer learning with new form of CNN that is efficient Net B2 and applied it with max pooling and SoftMax. The input values may also be higher than one.

The base model transfers the knowledge to the new model we are training and for training method we split our data in to 3 parts that is training validation and testing which then provide the solution for the new model.

All the neurons in turn provide all the data to the dense layer which comes next in model as we know a dense layer is a layer where and hence provide the needed result a flow diagram of this technique is shown in figure 4.4.

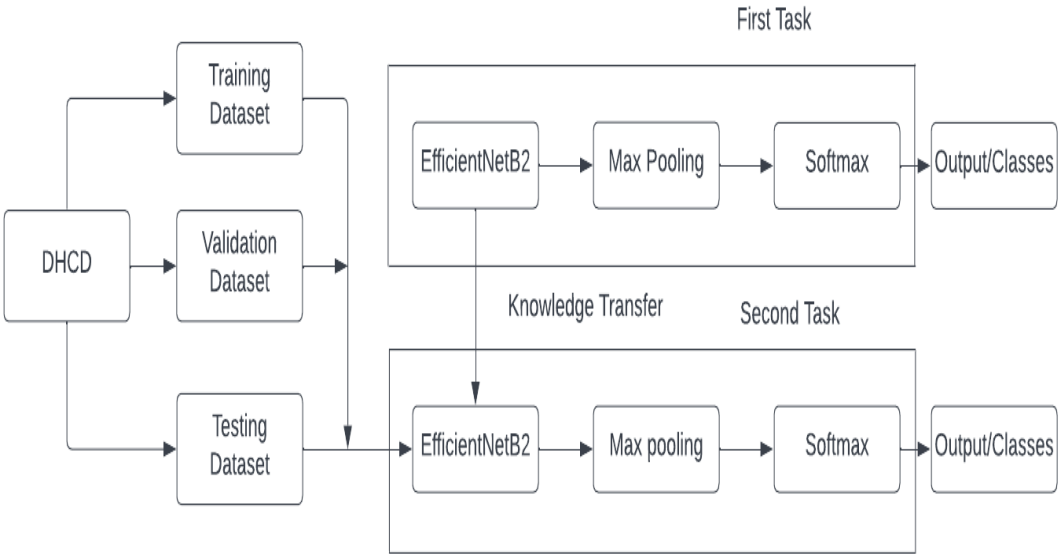


Figure 4.4: Flow Diagram of our Mode

CHAPTER 5

EXPERIMENTS AND RESULTS

We use a 40-epoch system to train this structure on the DHCD dataset. This system will pause the process after every 5 epochs and ask the user whether they want to continue with the training or halt the remaining process. If the user wants to continue with the training, they can enter the number of epochs that they want to run, but if they want to stop the progress after 5 epochs, they can hit the H key. For every character mentioned in the DHCD we provided with the error rate in the chart given below in figure 5.1.

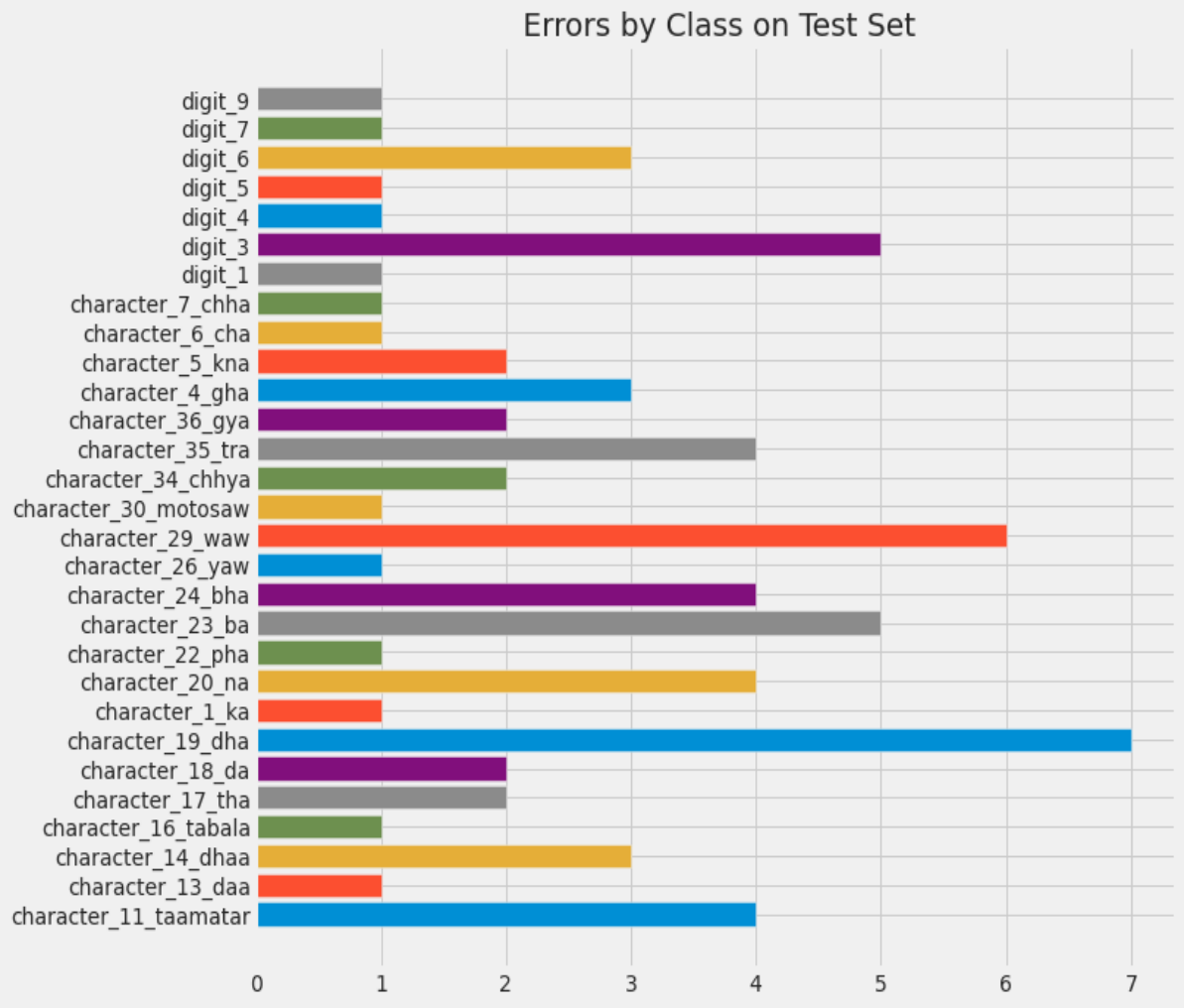


Figure 5.1: error by classes on character's test set

This result was accomplished by training this model for a total of forty times in a row. It was terminated since there was no increase in accuracy gained over a period of four epochs in a row. Therefore, despite the fact that there have been 40 epochs, we will end the loop at 33 epochs. In the table that is displayed above in table 5.1, the accuracy, loss, validation accuracy, and validation loss are all indicated.

Epochs	**Accuracy**	**Loss**	**Accuracy val**	**Loss val**
1	72.098	1.10086	96	0.001
2	94.543	0.52275	97.739	0.001
3	96.8	0.40313	98.565	0.001
4	97.752	0.33761	98.957	0.001
5	98.304	0.2968	98.957	0.001
6	98.641	0.27257	98.957	0.001
7	98.95	0.25753	99.087	0.001
8	99.009	0.23497	99.043	0.001
9	99.102	0.21617	99.348	0.001
10	99.261	0.2253	99	0.001
11	99.504	0.18839	99.522	0.0005
12	99.596	0.18285	99.304	0.0005
13	99.663	0.18465	99.174	0.0005
14	99.7	0.16773	99.522	0.00025
15	99.737	0.16197	99.609	0.00025
16	99.774	0.16006	99.391	0.00025
17	99.787	0.1597	99.348	0.00025
18	99.811	0.16371	99.304	0.00025
19	99.841	0.15394	99.435	0.00013
20	99.861	0.15091	99.435	0.00013
21	99.863	0.14652	99.348	0.00013
22	99.893	0.15001	99.478	0.00013
23	99.891	0.14528	99.522	0.00006
24	99.891	0.1452	99.522	0.00006
25	99.876	0.14182	99.478	0.00006
26	99.915	0.1419	99.478	0.00006
27	99.907	0.14236	99.478	0.00003
28	99.885	0.14162	99.478	0.00002
29	99.896	0.14183	99.478	0.00002
30	99.902	0.14193	99.478	0.00001
31	99.902	0.14124	99.478	0
32	99.902	0.1419	99.478	0
33	99.915	0.14166	99.478	0

Table 5.1: Accuracy table of EfficeintNetB2 with Transfer learning on DHC Dataset.

The model is implemented in Python 3.7 using the Anaconda platform. To comprehend the performance of provided model, we tested the DHCD dataset which is being used by various authors in their papers and it is validated by different researchers also.

on that database our model gives different results through its metrics as shown by the model for these classes mention below in the diagram in figure 5.2.

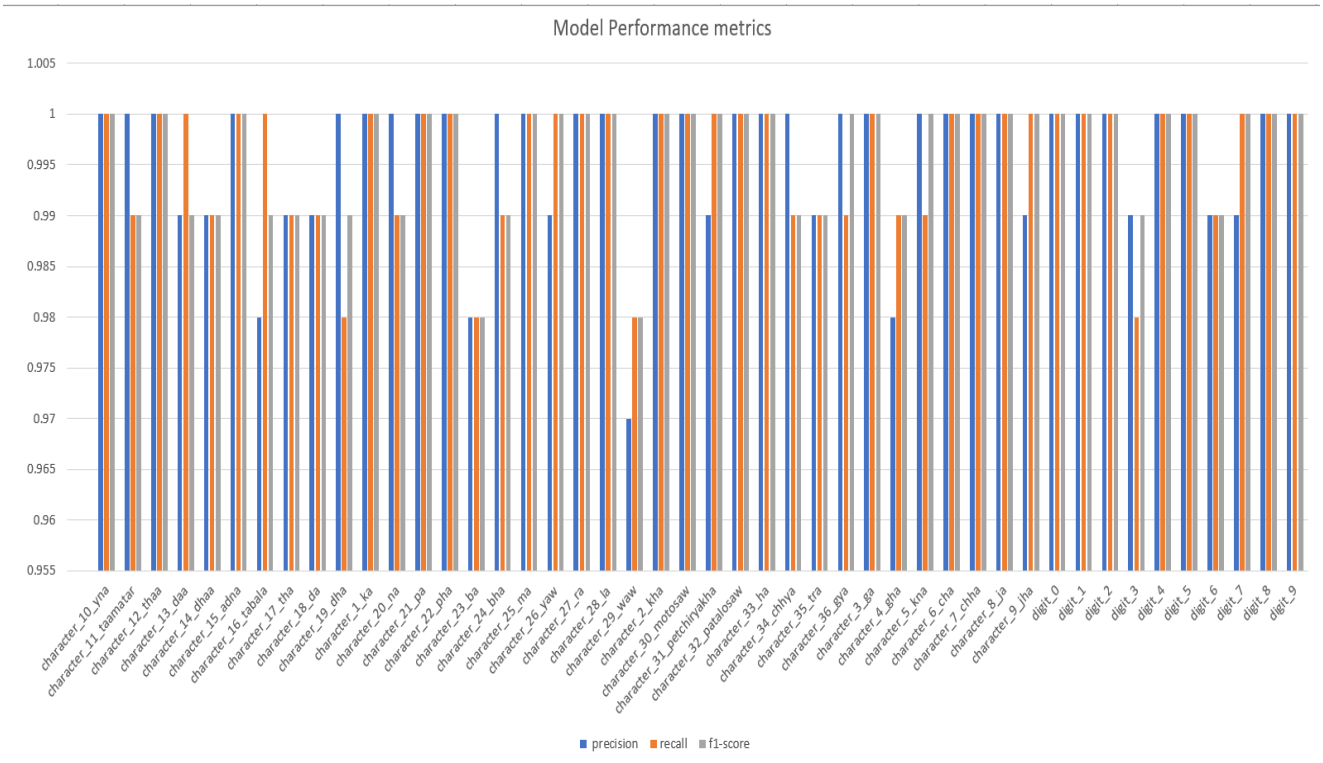
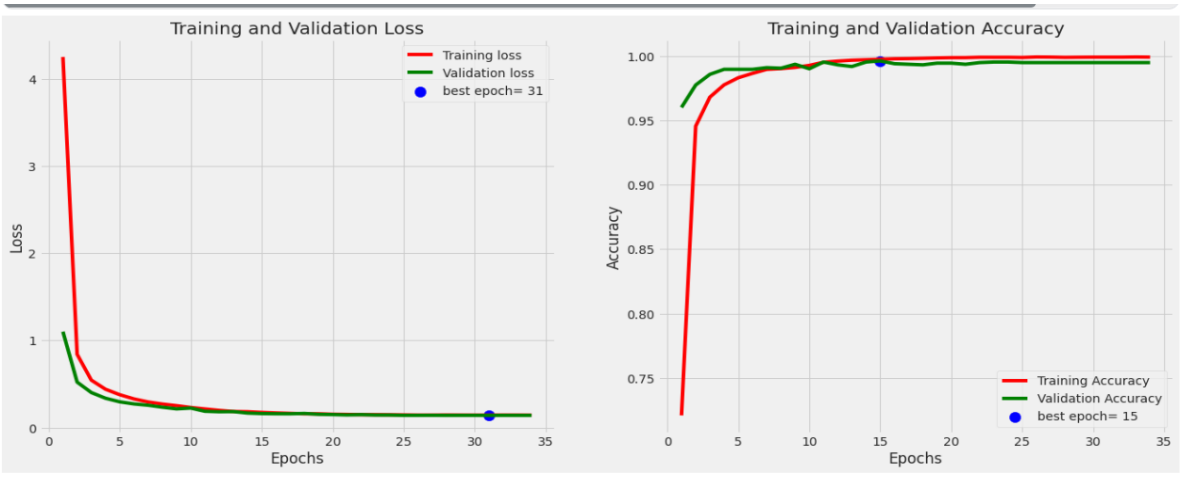


Figure 5.2: Model Performance Metrics (Vertical)

Training and validation loss and training and validation accuracy we achieve in this model are shown using the Loss and Accuracy graph in figure 5.3 given below.




```
184/184 [==============================] - 51s 275ms/step - loss: 0.1476 - accuracy: 0.9949
```

Figure 5.3: Training and validation loss with training and validation accuracy of our mode

Although the highest accuracy achieved was 99.609 on 15th epoch but the model was not completed as the accuracy was varying in each epoch. The accuracy achieved is 99.49 whereas the best epoch was 31 for loss where as the best epoch for the accuracy was 15th. Time taken per step was 51 second and 275 milli seconds for the 33 epochs we iterated.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

This article takes a comprehensive look at the most efficient, cutting-edge technologies and methods that have been used in HCR in the past. Although the amount of work that has been done in this field is substantial, the high demand for HCR necessitates the development of more effective and accurate algorithms that require less time and storage.

A complicated study topic has been presented as a result of the wide variety of human handwriting and the various character write-ups. This paper presents a concise analysis of all of the significant algorithms that were addressed. Researchers will get in-depth information of the ongoing work being done on this issue as a result of this study.

Handwritten Character Recognition Using Efficient Net B2 With Transfer Learning and Two Dense Layers Is the Focus of This Research This paper explores and concentrates on Handwritten Character Recognition using Dataset DHCD of Devanagari script, which has 92000 pictures of 46 distinct classes. Because these techniques scale all of the factors uniformly, unlike other CNN methodologies, which makes them superior, transfer learning model helps us in rapid progressive accuracy of our result, and The EfficientNet are the face of CNN after their publication in 2019. This will happen in 2019 because these techniques scale all of the factors uniformly. In addition, two thick layers facilitate the processing of data through the utilization of metrics vector multiplication from all of the neurons in the layer that comes before it. Even if a vast amount of work has already been done in this sector, there is still a lot of research that needs to be done in order to achieve a high level of accuracy while taking into account a large time factor.

This article assists scholars in this sector in examining new methods and comparing them to one another, which might be useful for further research and development in the years to come. Our model's calculations are dependent on the size of the picture contained in the DCH dataset. A more complicated dataset may be utilized in the testing of our model, which may lead to outcomes that are distinct from those predicted by the model.

The accuracy achieved by the other approaches is compared in table 1.1 above, along with the accuracy achieved by our method, which is superior to the accuracy achieved by the other ways. In addition, this precision may be enhanced by refining the tuning of a number of factors, although doing so may require a significant investment of time and may or may not be technically viable. This work is being presented as a starting effort, and the objective is to simplify the progress and process of identifying hand-written Indo characters. This yields an accuracy in validation that is 99.49 percent accurate.

REFERENCES

[1] Khan, H. U. (2017). “Mixed-sentiment classification of web forum posts using lexical and non-lexical features”. Journal of Web Engineering, 161-176.

[2] Al-behadili, H. N. K. (2016). “Classification algorithms for determining handwritten digit”. Iraq J. Electr. Electron. Eng, 12(1), 96-102..

[3] Ramzan, M., Khan, H. U., Awan, S. M., Akhtar, W., Ilyas, M., Mahmood, A., & Zamir, A. (2018). “A survey on using neural network based algorithms for hand written digit recognition”. International Journal of Advanced Computer Science and Applications, 9(9).

[4] Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009, June). “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations”. In Proceedings of the 26th annual international conference on machine learning (pp. 609-616).

[5] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). “Gradient-based learning applied to document recognition”. Proceedings of the IEEE, 86(11), 2278-2324.

[6] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). “Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems”, 25.

[7] Majumder, S., von der Malsburg, C., Richhariya, A., & Bhanot, S. (2018). “Handwritten digit recognition by elastic matching”. arXiv preprint arXiv:1807.09324.

[8] Shamim, S. M., Miah, M. B. A., Angona Sarker, M. R., & Al Jobair, A. (2018). “Handwritten digit recognition using machine learning algorithms”. Global Journal Of Computer Science And Technology.

[9] Ramzan, M., Khan, H. U., Awan, S. M., Akhtar, W., Ilyas, M., Mahmood, A., & Zamir, A. (2018). “A survey on using neural network based algorithms for hand written digit recognition”. International Journal of Advanced Computer Science and Applications, 9(9).

[10] Agrawal, M., Chauhan, B., & Agrawal, T. (2022). “Machine Learning Algorithms for Handwritten Devanagari Character Recognition: A Systematic Review”.

[11] Vashist, P. C., Pandey, A., & Tripathi, A. (2020, January). “A comparative study of handwriting recognition techniques”. In 2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM) (pp. 456-461). IEEE.

[12] Siddique, F., Sakib, S., & Siddique, M. A. B. (2019, September). “Recognition of handwritten digit using convolutional neural network in python with tensorflow and comparison of performance for various hidden layers”. In 2019 5th International Conference on Advances in Electrical Engineering (ICAEE) (pp. 541-546). IEEE.

[13] Garg, A., Gupta, D., Saxena, S., & Sahadev, P. P. (2019, March). "Validation of random dataset using an efficient CNN model trained on MNIST handwritten dataset". In 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 602-606). IEEE.

[14] Rani, N. S., Subramani, A. C., Kumar, A., & Pushpa, B. R. (2020, July). Deep learning network architecture based kannada handwritten character recognition. In 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 213-220). IEEE.

[15] Acharya, S., Pant, A. K., & Gyawali, P. K. (2015, December). Deep learning based large scale handwritten Devanagari character recognition. In 2015 9th International conference on software, knowledge, information management and applications (SKIMA) (pp. 1-6). IEEE.

[16] Devi, N., Rani, P. L., Gokul, A. G., & Sivagami, V. M. Offline Handwritten Character Recognition using Convolutional Neural Network.

[17] Dokare, I., Gadge, S., Kharde, K., Bhere, S., & Jadhav, R. (2021, May). Recognition of Handwritten Devanagari Character using Convolutional Neural Network. In 2021 3rd International Conference on Signal Processing and Communication (ICPSC) (pp. 353-359). IEEE.

[18] Aneja, N., & Aneja, S. (2019, July). Transfer learning using CNN for handwritten devanagari character recognition. In 2019 1st International Conference on Advances in Information Technology (ICAIT) (pp. 293-296). IEEE.

[19] Gupta, S., & Mohapatra, R. K. (2019, November). Performance improvement in handwritten devanagari character classification. In 2019 Women Institute of Technology Conference on Electrical and Computer Engineering (WITCON ECE) (pp. 60-64). IEEE.

[20] Jha, V., & Parvathi, K. Feature Selection for Character Recognition of Handwritten Devanagari and Odia Scripts.

[21] Acharya, S., Pant, A. K., & Gyawali, P. K. (2015, December). Deep learning based large scale handwritten Devanagari character recognition. In 2015 9th International conference on software, knowledge, information management and applications (SKIMA) (pp. 1-6). IEEE.

[22] Pant, A. K., Panday, S. P., & Joshi, S. R. (2012, November). Off-line Nepali handwritten character recognition using Multilayer Perceptron and Radial Basis Function neural networks. In 2012 Third Asian Himalayas International Conference on Internet (pp. 1-5). IEEE.

[23] http://yann.lecun.com/exdb/mnist/

[24] dataset, DHCD 92000 image, 46 class -> https://archive.ics.uci.edu/ml/datasets/Devanagari+Handwritten+Character+Dataset

[25] Agrawal, M., Chauhan, B., & Agrawal, T. (2022). Machine Learning Algorithms for Handwritten Devanagari Character Recognition: A Systematic Review.

[26] Chhabra, A. (2019). Deep Learning Based Real Time Devanagari Character Recognition.

[27] Maitra, D. S., Bhattacharya, U., & Parui, S. K. (2015, August). CNN based common approach to handwritten character recognition of multiple scripts. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR) (pp. 1021-1025). IEEE.

[28] Thakur, A., & Kaur, A. (2019). Devanagari handwritten character recognition using neural network. International Journal of Scientific & Technology Research, 8(10).

[29] Jayadevan, R., Kolhe, S. R., Patil, P. M., & Pal, U. (2012). "Automatic processing of handwritten bank cheque images: a survey". International Journal on Document Analysis and Recognition (IJDAR), 15(4), 267-296.

[30] Pratama, S. B. SVD Implementation on MNIST Image Classification Based on CNN.

[31] Gupta, N., & Goyal, N. (2021, January). Machine Learning Tensor Flow Based Platform for Recognition of Hand Written Text. In 2021 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-6). IEEE.

[32] Gope, B., Pande, S., Karale, N., Dharmale, S., & Umekar, P. (2021). Handwritten digits identification using MNIST database via machine learning models. In IOP Conference Series: Materials Science and Engineering (Vol. 1022, No. 1, p. 012108). IOP Publishing.

LIST OF PUBLICATIONS

Paper	Title. Author list. Conference/Journal	Status
Paper 1	**A Review on Techniques of Handwritten Character Recognition.** Prajjwal kumar, Shailender Kumar. International Conference on Advances in Computing, Communication Control and Networking (ICAC3N–22)	Accepted
Paper 2	**Handwritten Devanagari Character Recognition and Technique Comparison.** Prajjwal kumar, Shailender Kumar. International conference on contemporary Computing (IC3-2022)	submitted