

DIGITS RECOGNITION SYSTEM USING MACHINE LEARNING

Project report submitted in partial fulfilment of the requirement for
the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Samarth Sharma (191365)

Rohit Sharma (191403)

Under the supervision of

Mr. Prateek

to



Department of Computer Science & Engineering and Information
Technology

Jaypee University of Information Technology, Wagnaghat,

Solan - 173234, Himachal Pradesh

Candidate's Declaration

We hereby declare that the work presented in this report entitled “**Digits recognition system using Machine Learning**” in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Wagnaghat is an authentic record of my own work carried out over a period from January 2023 to May 2023 under the supervision of **Mr. Prateek**, Assistant Professor (Grade-II), Computer Science & Engineering and Information Technology.

We also authenticate that we have carried out the above-mentioned project work under the proficiency stream **Machine Learning**.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Samarth Sharma, 191365

Rohit Sharma, 191403

This is to certify that the above statement made by the candidate's is true to the best of my knowledge.

Mr. Prateek

Assistant Professor (Grade-II)

Computer Science & Engineering and Information Technology

Dated:

Plagiarism Certificate

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT PLAGIARISM VERIFICATION REPORT

Date:

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report B.Tech Project Report Paper

Name: _____ Department: _____ Enrolment No _____

Contact No. _____ E-mail. _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none"> • All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String 		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

Acknowledgement

Firstly, we express my heartiest thanks and gratefulness to almighty God for his divine blessing makes us possible to complete the project work successfully.

We really grateful and wish my profound indebtedness to Supervisor Mr. Prateek, Assistant Professor (Grade-II), Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat. The deep Knowledge & keen interest of my supervisor in the field of “Machine Learning” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to Mr. Prateek, Assistant Professor (Grade-II), Department of Computer Science & Engineering and Information Technology, for his kind help to finish our project.

We would also generously welcome each one of those individuals who have helped us straight forwardly or in a roundabout way in making this project a win. In this unique situation, we might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated us.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

Samarth Sharma
191365
CSE

Rohit Sharma
191403
CSE

Table of Content

Chapter	Title	Page No.
	Candidate's Declaration	i
	Plagiarism Certificate	ii
	Acknowledgement	iii
	Table of Content	iv
	List of Abbreviations	v
	List of Figures	vi – vii
	Abstract	viii
1.	Introduction	1 – 9
2.	Literature Survey	10 – 14
3.	System Design & Development	15 – 46
4.	Experiments & Result Analysis	47 – 53
5.	Conclusions	54 – 56
	References	57 – 58
	Appendices	59 – 62

List of Abbreviations

Abbreviations	Meaning
CNN	Convolutional Neural Network
ML	Machine Learning
AI	Artificial Intelligence
MNIST	Modified National Institute of Standards and Technology
HMM	Hidden Markov Models
SVM	Support Vector Machine
ANN	Artificial Neural Network
SGD	Stochastic Gradient Descent
RFC	Random Forest Classifier
SDLC	Software Development Life Cycle
PIL	Python Image Library
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
MB-SGD	Mini Batch Stochastic Gradient Descent
AdaGrad	Adaptive Gradient

List of Figures

Figure No.	Title	Page No.
1.1	Steps in Digit Recognition System	7
3.1	Software Development Life Cycle	15
3.2	MNIST Dataset	20
3.3	MNIST Neural Network Training	20
3.4	CNN Architecture	29
3.5	CNN for digit recognition system	31
3.6	Convolution Layer	33
3.7	Pooling Layer	34
3.8	Fully Connected Layer	35
3.9	Dropout Layer	36
4.1	Data Visualization	47
4.2	Data after Normalization	47
4.3	Summary of the model	48
4.4	Training of the model	49
4.5	Validation accuracy of the model	49
4.6	Accuracy Curve	50
4.7	Loss Curve	50
4.8	Confusion Matrix	51
4.9	Classification Report	51
4.10	Input Image for prediction of Zero	52
4.11	Output for the Input Image of Zero	52

4.12	Input Image for prediction of Five	52
4.13	Output for the Input Image of Five	52
4.14	Input Image for prediction of Six	53
4.15	Output for the Input Image of Six	53
4.16	Input Image for prediction of Eight	53
4.17	Output for the Input Image of Eight	53

Abstract

The digit recognition is one of the most important issues in the field of pattern recognition applications. The digit recognition system has various applications and can be used in various advanced projects as a prerequisite. The digit recognition system can be used to process the bank cheques, data entry forms and for postal mail sorting purposes. The purpose of this project is to develop a machine learning model in order to predict the digit and we also want to develop an application using Tkinter so that a user can use the application to get the output for his digit. We have used the CNN model for the purpose of prediction of digits. In this project, we have developed a model and an application that can work offline or predict the digits offline with a good efficiency. In this project, we have tried to increase the accuracy of our model at every stage and we have achieved a final accuracy greater than 98%.

CHAPTER 1: INTRODUCTION

1.1 Introduction

In the current time of digitization, Machine Learning and deep learning are the fields that play an important role. With the utilization of machine learning and deep learning, human exertion can be decreased in perceiving, learning, expectations and a lot more regions. There is still a lot of information present in the form of paper and this information requires a lot of human effort. Every day with machine learning we are trying to reduce the human effort.

Recognizing a thing or a person from the previous encounters or learning. Also, Digit recognition is only perceiving or distinguishing the digits in any archive. Digit recognition system is essentially the working of a machine to plan itself or decipher the digits. Handwritten Digit recognition is the limit of a computer system to decipher the physically composed digits from different sources like messages, bank checks, papers, pictures, etc and in different circumstances for electronic penmanship acknowledgment on computer tablets, distinguishing number plates of vehicles, dealing with bank checks, digits entered in any structures and so on. AI gives different techniques through which human endeavours can be decreased in perceiving the physically composed digits.

Machine Learning is an AI strategy that trains computer to do what effectively makes sense for individuals: learning through models. With the use of machine learning techniques, human endeavours can be lessened in seeing, getting the hang of, perceiving and in significantly more locales.

Utilizing machine learning, the computer figures out how to complete characterization functions from pictures or items from any report. Machine Learning models can achieve condition of-workmanship precision, past the human level execution. The digit recognition model purposes huge datasets to perceive digits from particular sources.

Digit recognition of characters has been around since the 1980s. The errand of manually written digit acknowledgment, utilizing a classifier, has exceptional importance and utilize, for example, - online digit acknowledgment on computer tablets, perceive postal districts on mail, handling bank check sums, numeric segments in structures topped off the hard way (for instance - tax

documents, etc. There are different difficulties looked while endeavouring to tackle this issue. The transcribed digits are not generally of a similar size, thickness, or direction and position comparative with the edges. The fundamental goal was to complete an example portrayal strategy to see the handwritten digits gave in the MNIST informational index of pictures of written by hand digits (0-9).

Digit recognition is as of now utilised in an assortment of client confirmation applications. Since the transcribed numerals contrast in size, thickness, style, and direction. Thus, these obstacles should be conquered to take care of the issue in my undertaking. We will use a novel type of deep neural network called a Convolutional neural Network, which is utilised to dissect visual symbolism overwhelmingly of pixel information into significant information that can be sent as info layer information to a convolutional neural Network for preparation. From that point onward, the framework will make a model for manually written digit recognition utilising stowed away layers of CNN. On the MNIST dataset, which contains 70,000 photos of manually written digits. Keras, a Python-based neural network library, is used. The organisation is prepared utilising the stochastic inclination and back engendering calculations, and afterward tried utilising the forward strategy. When the model is prepared, the client can transfer an image containing digits and get a legitimate estimate of their feedback.

To make machines wiser, the designers are jumping into deep learning and machine learning methods. A human figure out how to play out an errand by rehearsing and rehashing it and once more so it remembers how to play out the undertakings. Then the neurons in his mind consequently trigger also, they can rapidly play out the errand they have learned. Machine learning is likewise basically the same as this. It involves various kinds of neural network designs for various sorts of issues.

This project works on the recognizing of the handwritten digits (0 to 9) using the known MNIST dataset, using the convolutional neural network. We have tried to create a model which has a good accuracy, performance, time and performs well on their performance measures.

1.1.1 Digit recognition system

The digit recognition is the capacity of a computer to perceive human written digits. It is a hard undertaking for the machine in light of the fact that transcribed digits are noticeably flawed and can be made with a wide range of flavours. The digit recognition system is the solution to this problem which uses the image of a digit and recognizes the digit present in the image.

The need to create systems that can recognise and comprehend digital information has grown critical as we utilise digital technology more and more in our daily lives. The digit recognition system is one such technology that is used to recognise numbers in an image or a handwritten text. A crucial instrument in a variety of industries, including finance, healthcare, and education, is the digit recognition system.

To effectively recognise and categorise digits, the digit recognition system leverages cutting-edge machine learning techniques. These algorithms can learn the properties and traits of various digits since they have been trained on vast datasets of labelled photos. Once trained, the system can examine a handwritten text or image and recognise the digits that are present.

The digit recognition system can be used for a variety of tasks, including digitising handwritten notes, identifying the amounts on bank checks, and processing forms and surveys. In the area of biometrics, it has also been applied to the recognition and verification of people based on their handwriting.

An important turning point in artificial intelligence and machine learning was the creation of the digit recognition system. The digit recognition system has significantly impacted several fields and has evolved into an essential tool in our daily lives thanks to its capacity to recognise and comprehend digital information.

In addition, the digit recognition system has the potential to revolutionise human-computer interaction and change how we interact with technology in general. The digit recognition system may now be integrated into a variety of

devices, including smartphones, tablets, and laptops, making it possible for users to engage with these gadgets in more intuitive and natural ways.

The digit recognition system is an essential piece of technology that has a significant impact on our daily lives. It has the potential to revolutionise several fields and the way people interact with technology because of its capacity to recognise and comprehend digital information. The digit recognition system will undoubtedly play a more significant part in determining how our society develops as technology continues to advance.

We have designed such a model and we also designed the application for the same for the convenience of the user. We have used the various python and machine learning libraries for achieving our goal or objective.

We are using Tkinter library to create the application for the same. The application is as user friendly as possible and user can use the project or the model very easily and conveniently with no problem or any hindrance.

1.2 Problem Statement

The goal of this project is to make a model that will actually perceive and decide the written digits from its images by utilizing the ideas of a Convolution Neural Network. The objective is to make a model which can perceive the digits, it tends to be reached out to letters and a person's handwriting. The significant objective of the proposed framework is learning Convolutional Neural Network and applying it to the handwritten digit recognition system.

1.3 Objective

The main Objective of this project is to predict the digit from the digit written by the user from a direct image provided by the user. We are using convolutional neural network in order to achieve the task with the maximum accuracy. This approach significantly reduces the risk of false positives in the proposed system. The prediction that we are going to make can be used for further computations as the time is reduced due to the use of computerized software as compared to the traditional systems.

1.4 Methodology

1.4.1 Basic steps in constructing a machine learning model:

1.4.1.1 Data collection

- How accurate the model is going to be is completely depended on the data.
- This step determines that which data we are going to use for the training of our machine learning model.
- We can use different pre collected datasets by downloading them from Kaggle or another repository.

1.4.1.2 Data preparation

- Preparing the data by wrangling it.
- Cleaning the data which may require various data pre-processing operations.
- Randomizing the data which remove a order or sequence.
- Visualizing data to detect the useful relationships which prevent the bias and variance problem.
- Splitting the data into training and testing data.

1.4.1.3 Choose a model

- Choosing the best suitable algorithm for the model by analysing the problems in the other algorithms.

1.4.1.4 Train the model

- The focus of this step is to train our model using the training data so that it makes the correct predictions.
- With each iteration in this step the model is doing self-learning and rectifying itself to improve the accuracy.

1.4.1.5 Evaluating the model

- Makes use of a measurement or a combination of metrics to "measure" the model's objective execution.

- In contrast to test data, which doesn't help tune the model, this concealed information is meant to be fairly representative of how the model would operate in the real world.
- A good train/eval split would be 80/20, 70/30, or something similar, depending on the region, the availability of information, the characteristics of the dataset, etc.
- Test the model using hidden data from the beginning.

1.4.1.6 Parameter tuning

- This phase speaks to the art rather than science of hyperparameter setting.
- Fine-tune model bounds for expanded execution.
- Basic model hyperparameters could be the following: the quantity of preparation stages, the learning rate, the instruction values and conveyance, etc.

1.4.1.7 Making predictions

- Using additional (test set) information that has been withheld from the model up until this stage (and for which class marks are known) is used to test the model; a more accurate prediction of the model's behaviour in practise

1.4.2 Methodology for digit recognition system:

We involved MNIST as an essential dataset to prepare the model, and it comprises of 70,000 handwritten raster pictures from 250 distinct sources out of which 60,000 are utilised for preparing, and the rest are utilised for preparing approval. Our proposed strategy primarily isolated into stages, preprocessing, Model Development, Preparing and Approval, Model Assessment and Expectation. Since the stacking dataset is essential for any cycle, every one of the means comes after it.

First, the machine learning algorithms that will be utilised in the system need to be trained on a sizable dataset of labelled photos. The digit images in this dataset ought to be different in terms of their backgrounds, typefaces, and writing styles.

Next, training and testing sets are created from the dataset. The testing set is used to assess the accuracy of the system after training, while the training set is used to teach the machine learning algorithms to recognise the characteristics and patterns that distinguish different digits.

The machine learning algorithms are trained on the training set using a supervised learning strategy after the training and testing sets have been produced.

The algorithms pick up patterns and characteristics that are connected to various digits during this process, and they can utilise this knowledge to precisely categorise new digit images.

The algorithms are evaluated on the testing set once they have been trained to determine how accurate they are. If the accuracy is not up to par, the algorithms can be adjusted, or the dataset can be expanded with additional data to boost the system's performance.

Finally, the bigger system or application for which the digit recognition system was created is merged with it. To make sure the system functions as expected in real-world circumstances, this may need further testing and validation.

In general, the approach for creating a digit recognition system necessitates meticulous planning, data preparation, and training and testing of machine learning algorithms. This approach can be used to develop a precise and dependable system that can identify numbers in a variety of situations.

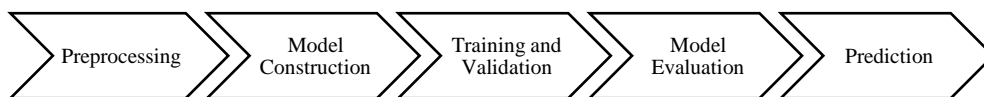


Fig.1.1 Steps in Digit Recognition System

1.5 Organization

This project report is divided into five chapters which are as follows: -

Chapter 1: - This chapter give the brief introduction of the project. The chapter provide the introduction about the project and gave a brief overview of the digit recognition system. The chapter also talks about the problem statement of the whole project and the objectives of the project. The chapter also provides a brief introduction to methodology used for the project and also provide the information about the steps in designing the digit recognition system using the machine learning and deep learning algorithms.

Chapter 2: - This chapter give the knowledge about the previous work related to the digit recognition system. This also provides the information related to the neural network, machine learning and deep learning. We have mentioned various Journals and related papers which give information about the work done earlier. The chapter gave us the information that how the various peoples have tried to use the various models in order create handwritten digit recognition and digit recognition systems. The techniques and the results for those techniques are mentioned in this chapter, and these help us to find the approach that we are going to use to create our model or project.

Chapter 3: - This chapter gave the information about the steps that we are going to follow to build the whole project. It talks about the system development and the model development. The information about the data set that we are going to use is provided in the chapter. Also, the complete information about the libraries that we are going to use is provided in the chapter. It also gives the information about the neural network that we are going to use. It tells the complete knowledge behind the Convolutional Neural Network by defining the all fragments of the neural network. It includes the information about the different layers in the neural network. It also includes the information about the optimizers. The chapter also include information about the data preprocessing which include data reduction, data cleaning and data transformation. It also includes information about data encoding, model creation, model training and the validation of the model. The various accuracy measures and validating tools

are also discussed. It also provides information about the system required to run the project and to create the project.

Chapter 4: - This chapter give the information about that how the whole project work is done and at every stage how we have kept check on the work. It provides information about the work done at different levels and also give the results obtained at the different levels. It provides the information about the model that we have created using the different modules and libraries. It also consists of the results from the various performance measure that we have used in the project. It provides the information about the accuracy of the model and the predictions made using the created model. The whole chapter is providing us the information about the performance our whole model or project.

Chapter 5: - This chapter consists of the whole conclusion of the work presented in this project report. It provides information about the whole phases of the project and it also mentions the future scope for the project. It also consists the information about the applications of the project and where the project can be utilized in order to make that sector more computerised. It gives the information that how we can improve the project and what we can do in future related to this project and how we can improve this project.

CHAPTER 2: LITERATURE SURVEY

In 1959, Grimsdale made a notable early attempt in the realm of pattern recognition research. Many of the examination projects that started in the middle of the 1960s relied on an approach called the analysis by-synthesis technique that Eden had suggested in 1968. The great value of Eden's work was in his formal demonstration—which was recalled for earlier works—that all manually produced characters are formed by a small number of schematic highlights.

The technique is then used for all the models presented till date in any paper or research work on character or pattern recognition.

- K. Gaurav, Bhatia P. K. [1], his work has the various pre-processing strategies for character recognition for various types of images ranges from a simple written by hand structure based on unending archives with shaded and sophisticated foundation and variable intensities. Various preprocessing techniques are discussed in this, including slant identification and amendment, picture improvement strategies of difference extending, binarization, commotion expulsion techniques, standardisation and division, and morphological handling techniques.
- Sandhya Arora et. al. [2], used the convergence, shadow included, code chain histogram, straight curve fitting highlights, and four component extraction techniques in total. For character pictures, shadow highlights are handled globally, whereas crossing point highlights, chain code histogram highlights, and line fitting components are calculated by dividing the character picture into different parts. Devanagari characters have a general acknowledgement rate of 92.80% after trial and error with a sample of 4900 cases.
- Brakensiek, J. Rottland, A. Kosmala, J. Rigoll [3] In this study, they present a framework for dispersed cursive calligraphy recognition based on (HMM) using discrete and mixed showing techniques. Examined are

penmanship acknowledgment tests that employ discrete and two crossover techniques, which include discrete and semi-constant design. Comparing a cross-breed demonstrating method for HMMs that relies on a neural vector quantizer (hybrid MMI) to discrete and half-breed Gee, in view of worn mixture structure (hybrid - TP), it is discovered that the execution of the acknowledgment rate can be improved. This improvement may be caused by an overall small informational collection.

- R. Bajaj, L. Dey, S. Chaudhari [4], utilized three various types of elements, to be specific, the thickness highlights, second elements and enlightening part includes for arrangement of Devanagari Numerals. They proposed multi classifier connectionist design for expanding the acknowledgment unwavering quality and they got 89.6% exactness for transcribed Devanagari numerals.
- G. Pirlo and D. Impedovo [5], in this work, the author proposed a new class of participation capabilities called Fuzzy membership capabilities used in drafting based grouping. These Fuzzy membership capabilities are easily customized to the any kind of traits of a classifying problem in hope to optimize classification performance. The author used a genuine genetic algorithm to identify the optimal Fuzzy membership capabilities and Voronoi tessellation-based clustering in a single optimization approach. The experiment was conducted in the field of user written digits and user written character identification, and the results showed that the optimal Fuzzy membership capabilities outperformed other enrolment capabilities on conceptual level, ranking level, and measure level weighting models present in the whole history.
- Sushree Sangita Patnaik and Anup Kumar Panda [6], the authors propose the use of particle swarm optimization and bacterial foraging optimization algorithms to optimize the performance of an active power filter by minimizing its undesirable losses. The efficiency and

effectiveness of both algorithms is compared on the basis of two unique conditions. The total harmonic distortion of the product current, that is a weight of the given thing's efficiency, is precisely come to approximately 1% using Bacterial foraging. The outputs depict that Bacterial algorithm has performed good as compared to the conventional or Particle-based methods by fully ensuring excellent active functionality and fast beat music in the product current, even in a zig-zag and unmeasured supply condition.

- M. Hanmandlu, O.V. Ramana Murthy [7], in their review, the authors have depicted the identification of user written Hindi and English numbers as a noteworthy contribution utilizing a fuzzy model. The identification is performed by changing the outstanding membership capabilities that are fitting for the fuzzy models. The membership capability is adjusted by two initial parameters that are estimated by optimizing the entropy subject to the satisfaction of membership capability to unity. The overall recognition rate is reported as 95% for Hindi numbers and 98.4% for English numbers.
- Renata F. P. Neves et. al. [8], a proposed approach for disconnected user written digit identification involves using a SVM classifier. The creators claim that SVM outperforms the MLP classifier. Experiments were conducted using the already defined dataset. One advantage of this algorithm is that it is able to handle the nonlinearly differentiable classes. Moreover, the algorithm will get stuck in a scope of local minima, where the process of training is stopping assuming that it has reached a good point in the error area. The other drawback of the algorithm is determining the optimal architecture, including the variable number of layers and perceptron in every unseen layer. Due to these limitations, a digit identifier using this kind of architecture is not able to achieve the expected low error rate.

- Narender Kumar, Himanshu Beniwal [9], have compared the previously presented models in the different papers. They try to find the best model by doing the comparison between the models and their accuracy. In their paper they have presented that the ConvNet or Convolutional Neural Network is much better than any other algorithm used for creating the model for predicting the digits.
- Pradeep et. al. [10], have proposed diagonal feature extraction method for the character written by hands by utilising feed-forward neural network calculation. This method utilises diagonal, flat and vertical elements for the arrangement reason.
- Vijayalaxmi R Rudraswamimath, Bhavanishankar K [11], have used multiple models like Support vector machine, K nearest neighbour, RFC and ConvNet with Theano and other libraries. They have created multiple models and find their accuracies. At the end they compared the accuracies of the model that they have created and come to the result that the ConvNet has performed best in terms of all performance measure and the ConvNet is predicting the digits more accurately and with an accuracy of 98.70%.
- Fathma Siddique, Shadman Sakib, Md. Abu Bakr Siddique [12], this paper presented the model that utilizing the SGD as the optimizer and they have also used backpropagation with the ConvNet in order to increase the accuracy of the model. The model has accuracy of 99.92% with 15 epochs but they have encountered with the problem that for less batch size the model accuracy decreases. But overall, the model has performed very good and they have achieved a good accuracy for the model.
- Yellina Navya Sree, Sreekar Kumar Jasti, Bharath Chandra Jampani, Dr. M. R. Narsing Rao [13], have used the ConvNet and Artificial Neural Network (ANN) for developing two different models for the comparison

purpose. They have found that the accuracy for the ConvNet is 96% and accuracy for the ANN is about 95%. But for some of the cases the ANN do not able to perform as good as the ConvNet performs. So, they concluded that ConvNet is better than the ANN model.

- Yellapragada SS Bharadwaj , Rajaram P , Sriram V.P , Sudhakar S , Kolla Bhanu Prakash [14], the user written digits recognition method using convolutional neural networks was proposed. Machine learning algorithms like KNN and SVM can't handle the unique way that digits are written, making it one of the intractable problems. In this study, CNNs are used using a MNIST dataset that has 70.000 digits and 250 different writing styles. This paper has achieved an accuracy of 98.51% for the real-world handwritten digits with 0.1% loss for the data.

CHAPTER 3: SYSTEM DESIGN & DEVELOPMENT

3.1 System Development

In order to develop a software or model we need to learn the concept of the Software Development Life Cycle (SDLC). In our project we are using the Waterfall model of the SDLC. The user can develop a good and efficient software by using this model of Software Development Life Cycle (SDLC). The main reason that it provides us an efficient and good software is that it uses the sequential way for the creation of the software. This method keeps check on the whole system design process as it doesn't allow a developer to move next step without completing the previous step. This structure gives us advantage that we do not have the overlapped processes. The result of the one step is used as an input for the next step and without completing the previous step we cannot move to the next stop. The Waterfall model design is given below. The all steps given below are important for a software or model creation.

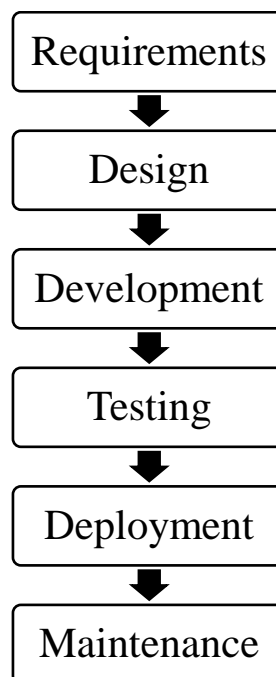


Fig. 3.1 Software Development Life Cycle

3.2 Model Development

The process of model development includes various steps. The various steps are: -

3.2.1 Importing the libraries:

The required libraries to create the model are:

3.2.1.1 Keras

A strong and very easy to utilise free open-source Python library for building and analysing ML models is called Keras.

It covers Theano and TensorFlow, two useful frameworks for mathematical computation, and enables you to characterise and prepare neural network models in just a few lines of code. It uses independent AI toolboxes, C#, Python, and C++ libraries. Theano and TensorFlow are tremendously powerful libraries for building brain organisations, but they are also challenging to understand.

Keras has a minimal architecture that offers a clear and straight forward technique for creating deep learning models in light of image processing libraries. The goal of Keras is to quickly characterise profound learning models. All things considered, Keras is the best option for situations requiring deep learning.

3.2.1.2 TensorFlow

Google created and offers TensorFlow, a library for rapid mathematical computation. It is a foundation library that can be used to create Profound Learning models directly or with the help of covering libraries that enhance the interaction built on top of TensorFlow.

The TensorFlow training exercise is designed for both beginners and experts. Our educational exercise covers all the fundamental and advanced concepts of AI and deep learning, such as deep brain organisation, photo handling, and emotion analysis.

One of the widely used deep learning architectures was developed by Google Group and is called TensorFlow.

This educational exercise is designed so that we may easily execute a programme using a free and open-source programming library that was created in the Python programming language.

3.2.1.3 NumPy

A Python library used for working with clusters is called NumPy. Additionally, it may operate in the areas of grids, Fourier transformation, and direct polynomial mathematics. A library called NumPy, which stands for Mathematical Python, contains a variety of schedules for managing those clusters in addition to multifaceted display objects.

NumPy may be used to execute numerical and intelligent operations on exhibitions. This educational activity clarifies NumPy's fundamentals, including its engineering and atmosphere. Additionally, it looks at various cluster capacities, ordering styles, etc. You are free to use it without restriction because it is an open-source task.

For Mathematical Python, use NumPy. NumPy aims to provide a display object that is ultimately 50x faster than typical Python records. The Nd array exhibit object in NumPy has a wealth of supporting features that make working with Nd array incredibly straightforward. In the field of information science, where efficiency and resources are crucial, exhibits are frequently used.

3.2.1.4 Pandas

Pandas is characterized as an open-source library that gives elite execution information control in Python. The name of Pandas is gotten from the word Board Information, and that implies an Econometrics from Multi-layered information. It is being used for information extraction and checking in Python and created by Wes McKinney in 2008.

Information examination requires heaps of handling, for example, rebuilding, cleaning or combining, and so on. There are various instruments are accessible for quick information handling, like Numpy, Scipy, Cython, and Panda. Yet, we lean toward Pandas since working with Pandas is quick, basic and more expressive than different devices.

The key functionality provided by Pandas:

- It has a quick and effective Data Frame object with the default and tweaked ordering.
- Utilized for reshaping and turning of the informational collections.
- Bunch by information for totals and changes.
- It is utilized for information arrangement and incorporation of the missing information.
- Process various informational collections in various configurations like network information, plain heterogeneous, time series.
- Handle numerous tasks of the informational collections, for example, sub setting, slicing, separating, group By, re-requesting, and the reshaping of data sets.
- It incorporates with different libraries like SciPy, and scikit-learn.
- Gives quick execution, and to speed it, much more, you can utilize the Cython.

3.2.1.5 Matplotlib

Python scripts can be used to create 2D charts and plots using the Matplotlib module. It provides a module called Pyplot that streamlines the plotting process by providing controls for line styles, text style properties, designing tomahawks, and other features. To be more precise, it supports a very broad range of diagrams and plots, including histograms, bar graphs, power spectra, error outlines, and others. It is used in conjunction with NumPy to provide a powerful open-source alternative to MATLAB. Additionally, it can be used with Python and PyQt's wxPython illustration toolkits.

3.2.1.6 Seaborn

Seaborn is a data visualisation library based on top of matplotlib and firmly coordinated with Pandas information structures in Python. Perception is the focal piece of Seaborn which helps in investigation and comprehension of information.

Seaborn offers the accompanying functionalities:

- Dataset situated Programming interface to decide the connection between factors.

- Programmed assessment and plotting of direct relapse plots.
- It upholds undeniable level deliberations for multi-plot lattices.
- Imagining univariate and bivariate circulation.

3.2.1.7 Pillow

With the help of the free and open-source Pillow package, you can easily create and control complex images using the Python computer language. Pad is based on PIL's top layer (Python Image Library). One of the key Python modules for handling images is PIL. In any event, Python 3 is not supported by the PIL module, which stopped being supported about 2011.

More features are offered by the Pillow module, which also works with practically every framework and Python 3.

It supports a variety of image formats, including "jpeg," "png," "bmp," etc. Using the pillow module, you can do almost anything with computerised images. In addition to basic picture handling functions, such as the point operations, picture separation using worked-in convolution sections, and variation space adjustments and manipulation

3.2.1.8 Tkinter

The inbuilt GUI library for Python is called Tkinter. When Python and this library is combined, it provides a fast and very simple way to create GUI applications. A reliable item-based connecting point to the Tk GUI toolbox is provided by Tkinter.

The modules that we will need to prepare our model are all ones that we actually wish to import. Currently, MNIST is one of the datasets in the Keras library. As a result, we can easily import the dataset with Keras. The preparation information and its names, along with the testing information and its names, are returned by the `mnist.load data()` method.

3.2.2 Loading the dataset:

3.2.2.1 MNIST dataset

Modified The National Institute of Standards and Technology (MNIST) hosts a sizable collection of computer vision datasets that are frequently used for

developing and evaluating different frameworks. The two distinct databases of the National Institute of Standards and Technology, which include parallel images of transcribed digits, were used to create it. 250 people's handwritten digits were included in the preparation set, of which half were employees of the Registration Department and the balance were secondary school students. Whatever the case, it is frequently cited as one of the first datasets to show the resilience of brain organisations amid other datasets.

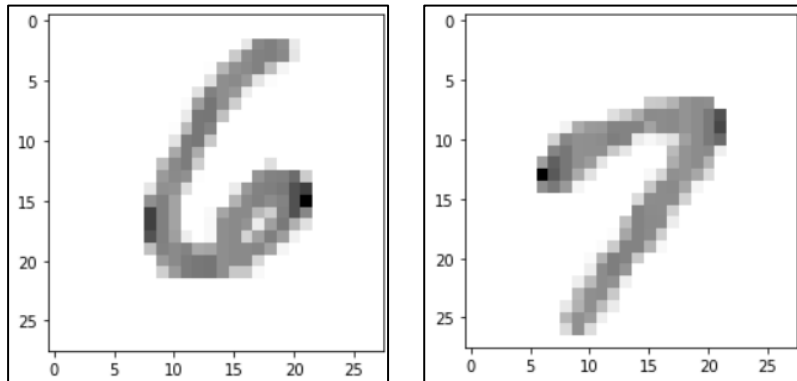


Fig. 3.2 MNIST Dataset

The dataset includes 10,000 photographs used for testing and 60,000 pictures used for preparation as few of them can be used for cross validation. Each of the numbers is grayscale and fixed in size, with the 28x28-pixel area at the picture's centre acting as the force. Each image is 784 pixels, forming a cluster that can be smoothed into a 784 layered matrix. Each component of the vector is a paired worth that represents the pixel's power.

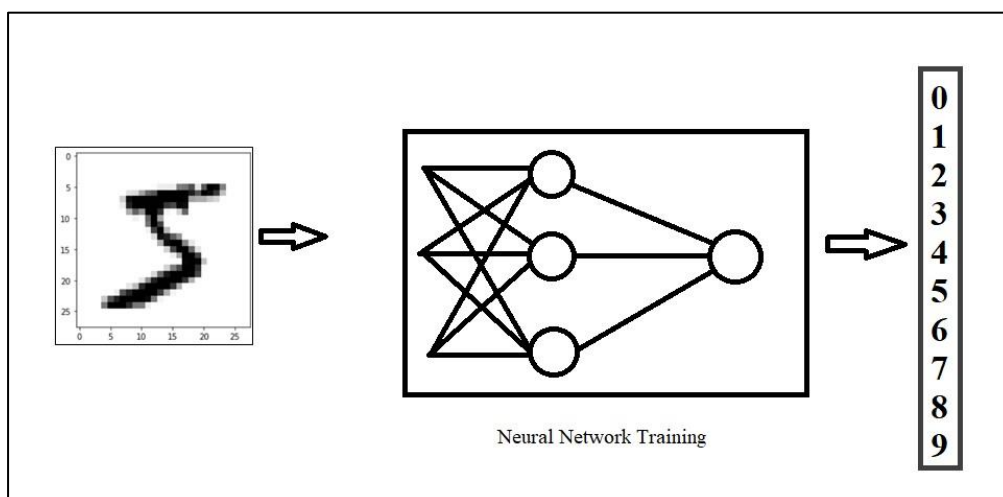


Fig. 3.3 MNIST Neural Network Training

3.2.3 Preprocessing:

Data pre-processing assumes a significant part in any acknowledgment cycle. Data pre-processing is a data mining strategy which is utilized to change the crude information in a valuable and proficient configuration. To shape the info pictures in a structure reasonable for division pre-handling is utilized. Data preprocessing is an essential step prior to building a model with these elements. It ordinarily occurs in stages:

- Data quality assessment
- Data cleaning
- Data transformation
- Data reduction

3.2.3.1 Data quality assessment

A data Quality Evaluation is an unmistakable stage inside the data quality life-cycle that is utilised to confirm the source, amount and effect of any information things that break pre-characterized information quality standards. The data Quality Evaluation can be executed as an oddball interaction or more than once as a component of a continuous data quality confirmation drive.

The nature of your data can rapidly rot after some time, even with rigid information catch techniques cleaning the information as it enters your data set. Individuals moving house, changing telephone numbers and dying all mean the information you hold can immediately become obsolete.

A data Quality Evaluation assists with recognizing those records that have become incorrect, the potential influence that error might have caused and the information's source. Through this evaluation, it very well may be corrected and other potential issues distinguished.

3.2.3.2 Data cleaning

The process of data cleaning is one of the important steps of AI. This step really plays a big role in constructing and creating the model. It doubtlessly is not the unique piece of AI and simultaneously, there is no any covered-up stunts or insider facts to present. It can be the original information cleaning, which represent the deciding factors of the whole undertaking. The most important information founders generally spend a very large or huge part of there

experience on the step. In view of the conviction that, "Better information beats fancier calculations". On the off chance that we have a very much cleaned dataset, we can come by wanted results even with an exceptionally straightforward calculation, which can demonstrate extremely helpful on occasion. Clearly, various sorts of information will require various sorts of cleaning. Notwithstanding, this deliberate methodology can constantly act as a great beginning stage.

3.2.3.3 Data transformation

In fact, by cleaning and smoothing the data, we have previously performed information adjustment. Notwithstanding, by information change, we comprehend the techniques for transforming the information into a proper configuration for the computer to gain from. Information change is the cycle where information is taken from its crude, siloed furthermore, standardised source state and change it into information that is combined, correspondingly displayed, de-standardized, and prepared for examination. Without the right innovation stack set up, information change can be tedious, costly, and monotonous. By and by, changing the information will guarantee greatest information quality which is basic to acquiring precise investigation, prompting important bits of knowledge that will ultimately engage information driven choices. Building and preparing models to handle information is a splendid idea, and more ventures have taken on, or then again plan to send, AI to deal with numerous reasonable applications. However, for models to gain from information to make significant expectations, the actual information should be coordinated to guarantee its examination yield important bits of knowledge.

3.2.3.4 Data reduction

Data reduction is a cycle that diminished the volume of unique information and addresses it in a lot more modest volume. Information decreases strategies guarantee the respectability of information while diminishing the information. The time expected for information decrease should not eclipse the time saved by the information mining on the diminished informational index.

Data reduction techniques:

- **Dimensionality Reduction:** - The dimensionality reduction refers to the strategy of reducing the component of an information data set. As a rule, AI datasets (highlight set) contain many segments (i.e., highlights) or a variety of focuses, making an enormous circle in a three-layered space. By applying dimensionality reduction, you can diminish or cut down the quantity of sections to quantifiable counts, in this way changing the three-layered circle into a two-layered object.

Types of dimensionality reduction are -

- a. **Principal Component Analysis:** - This approach of data reduction is one of the main direct procedures of dimensionality reduction. This technique plays out immediate planning of the information to a lesser layered space in a manner that boosts the difference of the information in the low-layered portrayal. Basically, it is a factual system that symmetrically changes over the 'n' directions of a dataset into another arrangement of n facilitates, known as the primary parts. This transformation brings about the formation of the primary head part having the greatest change. Each succeeding head part bears the most elevated conceivable change, under the condition that it is symmetrical (not corresponded) to the former parts. The PCA change is delicate to the general scaling of the first factors. Consequently, the information section ranges should initially be standardized prior to carrying out the PCA strategy. Something else to recall is that utilizing the PCA approach will make your dataset lose its interpretability.
- b. **Linear discriminant analysis:** - The linear discriminant analysis is a speculation of Fisher's linear discriminant technique that is generally applied in stats, pattern recognition, and AI. This strategy plans to find a direct mix of elements that can describe or separate between at least two classes of items. It addresses information in a manner that expands class distinctness. While objects having a place with a similar class are compared through

projection, objects from various classes are organized far separated.

- Numerosity Reduction: - In the numerosity reduction, the data volume is diminished by choosing another option, more modest type of data representation. These procedures can be parametric or non-parametric.
 - a. Parametric numerosity reduction: - Parametric technique expects a model into which the information fits. Information model boundaries are assessed, and just those boundaries are put away, and the other information is disposed of. Relapse and Log-Direct strategies are utilised for making such models. For instance, a relapse model can be utilised to accomplish parametric decrease in the event that the information fits the Straight Relapse model.
 - b. Non-Parametric numerosity reduction: - A non-parametric numerosity reduction strategy expects to have no model. The non-Parametric method brings about a more uniform decrease, regardless of information size, yet it may not accomplish a high volume of information decrease like the Parametric one. There are no less than four kinds of non-Parametric information decrease methods, Histogram, Bunching, Inspecting, Information Solid shape Collection, Information Pressure.

- Data Compression: - At the point when you work with a lot of information, it becomes more diligently to think of solid arrangements. Information decrease can be utilized to diminish how much information and decline the expenses of examination. In the wake of stacking the information, we isolated the information into X and y where X is the picture, and y is the name comparing to X. The principal layer/input layer for our model is convolution. Convolution accepts every pixel as a neuron, so we want to reshape the pictures to such an extent that every pixel esteem is in its own space, subsequently changing over a 28x28 grid of greyscale values into 28x28x1 tensor.

With the right aspects for every one of the pictures, we can divide the pictures into train and test for additional means. In the wake of stacking the information, we isolated the information into X and y where X is the picture, and y is the name comparing to X.

The primary layer/input layer for our model is convolution.

Convolution accepts every pixel as a neuron, so we really want to reshape the pictures with the end goal that every pixel esteem is in its own space, in this way changing over a 28x28 network of greyscale values into 28x28x1 tensor. With the right aspects for every one of the pictures, we can divide the pictures into train and test for additional means.

3.2.4 Data encoding:

This is an optional step since we are involving the cross-downright entropy as misfortune capability. We have to determine the organization that the given marks are absolute in nature. The crude information can contain different various kinds of information which can be both organized and unstructured and should be handled to bring to shape that is usable in the AI models. Since machine learning depends on numerical conditions, it would cause an issue when we keep downright factors with no guarantees.

Numerous calculations support downright qualities minus any additional control, yet at the same in those cases, it is as yet a subject of conversation on regardless of whether to encode the factors. After the ID of the information sorts of the highlights present in the informational index, the subsequent stage is to process the information in a manner that is reasonable to put to AI models.

The three well known strategies of changing values over completely to Numeric qualities are finished in two unique strategies:

- **Label Encoding:** - In Python Label Encoding, we want to supplant the all-out esteem utilizing a mathematical worth going among nothing and the complete number of classes short one. For example, on the off chance that the worth of the straight-out factor has six unique classes, we will utilize 0, 1, 2, 3, 4, and 5.

- One Hot Encoding: - In one-hot encoding, categorical information are addressed as vectors of zeros and ones. This is finished by involving a different sham variable for every class, and setting the worth of the spurious variable to 1 assuming the perception has a place with that classification and 0 in any case.
- Binary Encoding: - Binary encoding is a method used to change categorical data into mathematical information by encoding classifications as whole numbers and afterward changing them into two-fold code.
- Target Encoding: - Target encoding is a strategy for encoding the features of the elements by utilizing the objective worth. That's what the thought behind this procedure is on the off chance that the element is a decent indicator of the objective, its qualities ought to be nearer to the target.

Encoding changeability depicts the variety of encoding of separately inside a class. At the point when we talk about the changeability in one hot encoding, the fluctuation relies upon the hour of execution in which it chooses the quantity of classifications to take that truly do adequately affect the objective. Other encoding philosophies truly do show a huge inconstancy which is distinguished at the hour of approval.

3.2.5 Model construction:

Now, comes the real part where we at long last get to involve the fastidiously pre-arranged information for model building. Contingent upon the information type (subjective or quantitative) of the objective variable (usually alluded to as the Y variable) we are either going to fabricate an order (on the off chance that Y is subjective) or then again relapse (assuming Y is quantitative) model.

3.2.5.1 Learning algorithms

The machine learning algorithms can be classified into three categories:

1. Supervised Learning – In supervised learning, every model is a pair comprising of an information object (commonly a vector) and an ideal result esteem (likewise called the administrative sign). A supervised learning calculation analyze the preparation information and produces a derived capability, which can be utilized for planning new models.

An ideal situation will consider the calculation to accurately decide the class marks for concealed examples. It is an AI task that lays out the numerical connection between input X and yield Y factors. Such X, Y pair comprises the marked information that are utilized for model structure in a work to figure out how to foresee the result from the info. Supervised learning problems can be further classified into classification and regression problems.

- a. Classification: When the output variable is a category then it is a classification problem. Example: - “Red” or “Blue”.
 - b. Regression: When the output variable is a real value then it is a regression problem. Example: - “Weight” or “Rupees”.
-
2. Unsupervised Learning – In unsupervised learning, is an task that utilizes just the information X factors. Such X factors are unlabeled information that the learning calculation involves in demonstrating the inborn construction of the information. Unsupervised learning problems can be additionally classified into association and clustering problems.
 - a. Clustering: When we try to find the inherent grouping in the data, this kind of problem is known as clustering problem. Example: - Grouping customers by ordering sequence.
 - b. Association: When we try to find the rules that can describe the large portions of our data, this kind of problem is known as association problem. Example: - People selling A are also tend to sell B.

3. Reinforcement Learning – It is about making a reasonable move to expand compensation in a specific circumstance. It is utilized by different programming and machines to find the most ideal way of behaving or way it ought to take in a particular circumstance.

Reinforcement learning varies from the supervised learning in a manner that in supervised learning the preparation information has the response key with it so the model is prepared with the right respond to itself though in reinforcement learning, there is no response except for the support specialist chooses how to play out the given assignment. Without any a preparation dataset, it will undoubtedly gain from its insight. It is a machine learning task that settles on the following strategy and it does this by learning through experimentation with an end goal to augment the prize.

- Input: The input is the stage from where the model will start.
- Output: There can be different output due to the fact that there is different possible solution for a single problem.
- Training: The training is done on the basis of input. The model will give a result and the user will accept or reject the output and accordingly prize or penalise the model.
- The model will continue to learn.
- The highest reward point used to decide the best solution.

3.2.5.2 Model to be used for the project

Convolutional Neural Network (CNN)

CNN is an artificial neural network that takes a long time to select or recognise designs, to put it another way. Additionally, take control of them. CNN has therefore generally been useful for picture characterization. The channels in a CNN model come in a variety of shapes and counts. In essence, these pathways aid us in identifying the example. Although they can be used with one-layered and three-layered information, convolutional neural networks, or CNNs for

short, are a particular type of neural network model designed for working with two-layered picture information.

The convolutional layer, which gives the organisation its name, is the main component of the convolutional neural network. This layer performs a process known as "convolution."

Most commonly, convolutional and pooling layers make up a CNN model. The reason CNN works so well for picture order problems is because information that is treated as lattice structures performs better. While bearing in mind that preparation, the dropout layer is used to deactivate a section of the neurons, which lowers the model's ability to fit data. Our model is built using parallel order, convolution, and element extraction. Convolution and maximum pooling are used to separate the elements in the image. A 28x28 picture is applied with 32 3x3 convolution channels, followed by a 2x2 maximum pooling layer and a second convolution layer with 64 3x3 channels.

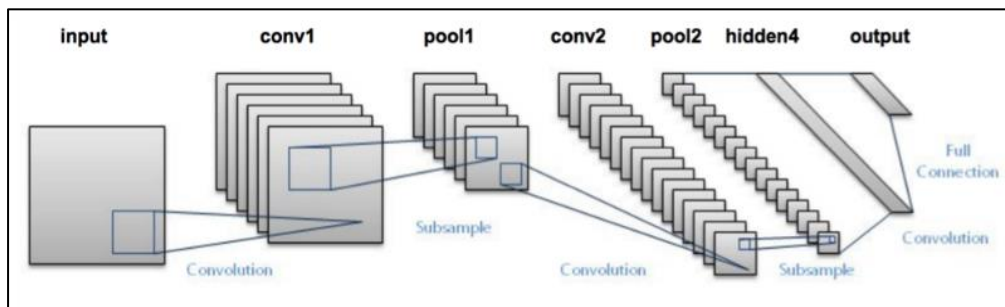


Fig. 3.4 CNN Architecture [15]

We eventually succeed in levelling 7x7 images. The level layer will blur the 7x7 images into a progression of 128 qualities that will be planned to a thick layer of 128 neurons that are connected to the obvious yield layer of 10 neurons.

The channel is smaller than the info information, and there is only a single item of duplication between the channel's estimated fix of the info and the info. A dab item is the component-wise rise between the information's channel-measured fix and channel, which is added to produce a single value continually. The activity is sometimes referred to as the "scalar item" because it produces a single value. It is intentional to utilise a channel that is smaller than the information because it enables the information show to repeatedly display the same channel (set of loads) at different focuses on the information. In

particular, the channel is methodically applied from top to bottom, left to right, to each covering portion or channel estimated fix of the info information.

Convolution in Computer vision

Applying the convolutional operation to picture information isn't new or remarkable to convolutional neural network, it is a typical procedure utilised in computer vision.

By and large, channels were planned by hand by computer vision specialists, which were then applied to a picture to bring about a component guide or result from applying the channel then, at that point, makes the investigation of the picture simpler here and there. The organisation will gain what sorts of elements to remove from the information.

In particular, preparing under stochastic slope drop, the organisation is compelled to figure out how to remove highlights from the picture that limit the misfortune for the particular errand the organisation is being prepared to address, for example remove highlights that are the most valuable for characterising pictures as canines or felines.

The one-layered convolutional layer and a two-layered convolutional layer guide to both

make the convolution activity concrete and give a worked instance of utilising the Keras layers.

- These models use a method to a build or to make an element matrix which add up to the presence of distinguished features in data.
- Channels can be hand tailored, like line identifiers, still the development of these networks is to know about them with the channels during creation with regards to specific type of issue.

Working of CNN

Layers of fictitious neurons are used to construct convolutional neural networks. Artificial neurons are numerical capabilities that calculate the weighted sum of many information sources and produce initiating esteem. These artificial neurons are a harsh imitation of their natural companions. Every neuron's

behaviour is characterised by its load. When taken care of by the pixel values, the artificial neurons of a CNN select various visual components. Each layer in a ConvNet generates a few initiation maps after receiving a picture as input. The pertinent components of the image are displayed on activation maps. Each neuron receives a set of pixels as information and duplicates their variety values by their loads before summarising them and processing them through the activation function.

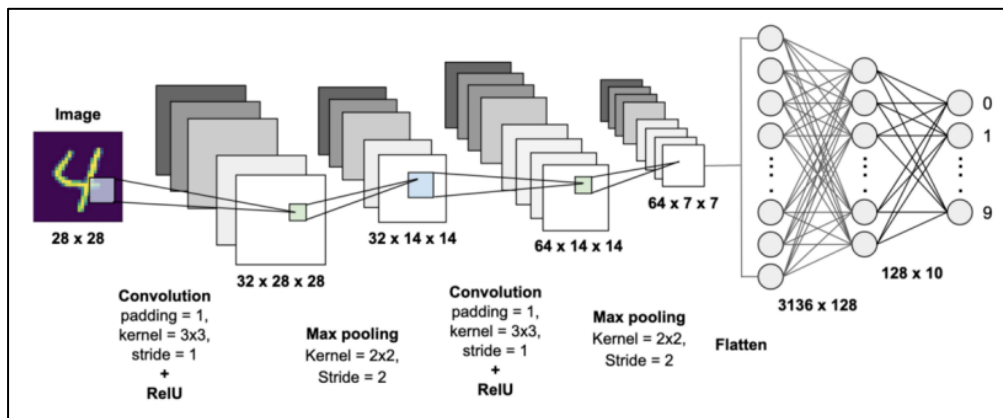


Fig. 3.5 CNN for digit recognition system [16]

In general, the first (or base) layer of the CNN recognises basic features like even, vertical, and askew edges. The subsequent layer, which removes more complex aspects like corners and edge blends, looks after the outcome of the primary layer as a contribution. Convolutional brain organisation layers start detecting more important level highlights, such as articles, faces, and more, as you go deeper into the organisation.

The process of greatly raising pixel values is known as "convolution" (thus to aggregate them the name convolutional neural network). A CNN typically consists of several convolutional layers, but it also includes other components. A CNN's last layer is a grouping layer, which uses the information from the previous convolution layer as input (recall, the higher convolution layers identify complex objects).

The order layer produces a number of certainty scores (numbers between 0 and 1) that represent how likely it is for the image to belong to a certain "class," given the guidance of the previous convolution layer. The outcome of the final layer is the likelihood that the info picture contains any of the identified animals,

for instance, on the off chance that you have a ConvNet that recognises cats, dogs, and ponies.

The following steps are taken after choosing the model:

Sequential is the model sort that will be used. The simplest way to build a model in Keras is sequential. It enables you to build models piece by piece.

The 'add ()' function is used to add layers to our model.

Conv2D layers make up our first two layers. These convolution layers, which may be thought of as two-layered networks, will control our feedback visuals.

The number of hubs in each layer is 64 in the top layer and 32 in the bottom layer. Depending on the amount of the dataset, this number can be modified to be sequential. 64 and 32 perform excellently in our situation, therefore we shall stick with this for the time being.

The channel lattice size for our convolution is called part size. We will therefore have a 3x3 channel network with a bit size of 3. For an update, refer back to the presentation and the main image.

The layer's ability to implement changes is known as activation. The ReLU will be the enactment capability we employ for our first two levels. It has been proved that neural networks perform brilliantly when using this enactment capacity.

An information shape is also taken in by our most memorable layer. As previously observed, each infographic is in this state: 28,28,1, where the number 1 denotes that the images are in greyscale.

There is a "Smooth" layer sandwiched in the midst of the Conv2D layers and the thick layer. The relationship between the convolution and thick layers is filled in by straighten.

We will utilise the layer type "Thick" for our result layer. In many instances, the layer type used for brain organisations is thick.

In our result layer, we shall have 10 hubs, one for each potential result (0-9).

"Softmax" activation is used. In order to translate the result into probabilities, Softmax summarises the outcome to 1. In light of the option with the highest likelihood, the model will next make its prediction.

We will have 10 hubs in our result layer, one for every conceivable result (0-9).

Layers in CNN model

The multiple occurring of these layers shows how profound our organization is, and this arrangement is known as the deep neural network.

- Input: crude pixel values are given as info.
- Convolutional layer: Information layers interpret the consequences of the neuron layer. There is a need to determine the channel to be utilised. Each channel must be a 5*5 window that slides over input information furthermore, gets pixels with the most extreme forces.
- Rectified Linear Unit (ReLU) layer: gave initiation capability on the information taken as a picture. On account of back engendering, ReLU capability is utilised which forestalls the upsides of pixels from evolving.
- Pooling layer: Plays out a down-examining activity in volume along the aspects (width, level).
- Completely associated layer: score class is engaged, and a greatest score of the info digits is found.

The different layers in CNN model: -

1. Convolution layer: -

The main layer used to segregate the many components from the information pictures is this one. Convolution is conducted numerically between the information picture and a channel of a certain size, $M \times M$, in this layer. The dab item is placed between the channel and the information picture components pertaining to the channel size by sliding the channel over the information image ($M \times M$).

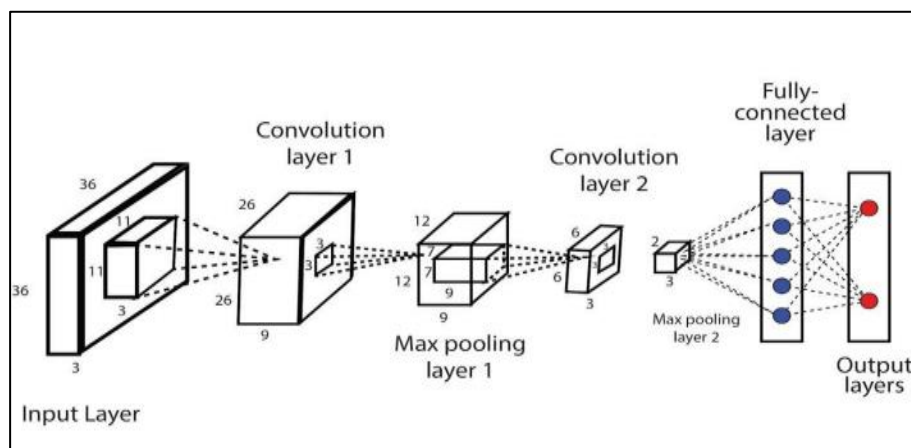


Fig. 3.6 Convolution Layer [17]

The outcome is known as the feature map, which provides information on the picture's corners and edges. In order to familiarise yourself with a few different highlights of the information picture, this element map is then handled by two different layers.

2. Pooling layer: -

A Pooling Layer typically follows a Convolutional Layer. The main goal of this layer is to reduce the convolved include guide's size in order to save on computational costs.

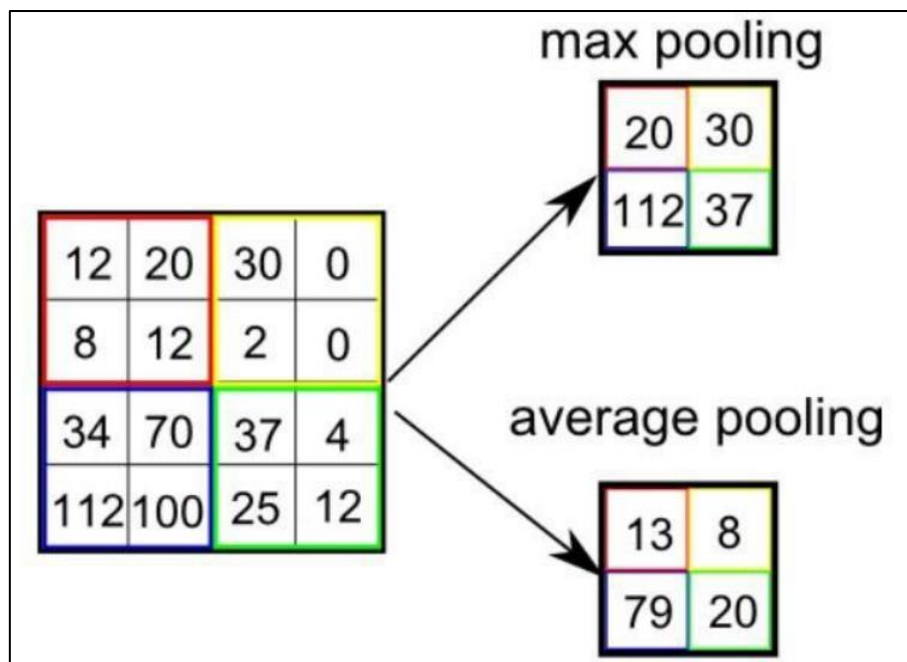


Fig. 3.7 Pooling Layer [17]

This is done by reducing the linkages between layers and working freely on each individual component map. There are many types of pooling tasks depending on the technique used. Max Pooling mostly uses the highlight map as one of its constituent parts. Normal pooling determines the normal of the elements in a measured picture segment that has been previously defined. Total Pooling calculates the total quantity of the components in the predefined segment. Between the Convolutional Layer and the Fully Connected Layer, the Pooling Layer typically serves as a scaffold.

3. Fully Connected layer: -

The Fully Connected layer comprises the loads and inclinations alongside the neurons and is utilised to associate the neurons between two distinct layers. These layers are normally positioned before the result layer and structure the last couple of layers of a CNN architecture.

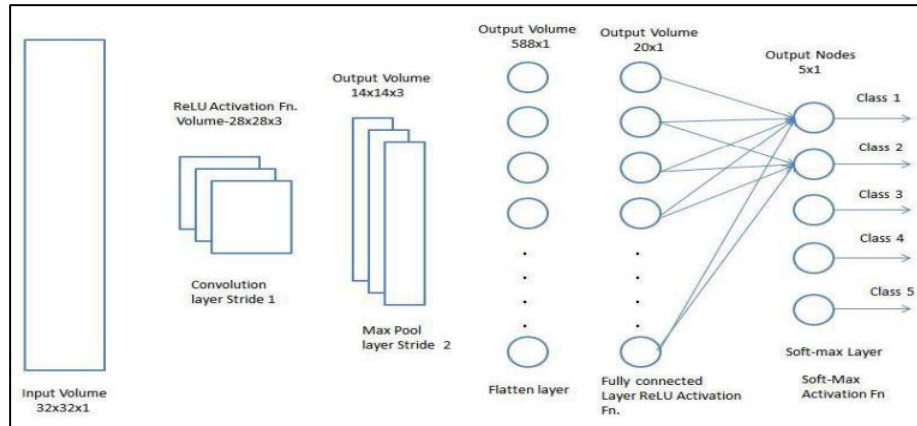


Fig. 3.8 Fully Connected Layer [17]

The straightened vector then, at that point, goes through not many more fully connected layers where the numerical capabilities tasks normally take place. In this stage, the grouping system starts to happen.

4. Dropout layer: -

For the most part, when every one of the elements are associated with the fully connected layer, it can cause overfitting in the preparation dataset. Overfitting happens when a specific model functions admirably on the preparation information causing a negative influence.

To overcome this issue, a dropout layer is used wherein a couple of neurons are dropped from the neural network during the preparation process bringing about diminished size of the model. On passing a dropout of 0.3, 30% of the hubs are exited arbitrarily from the brain organisation.

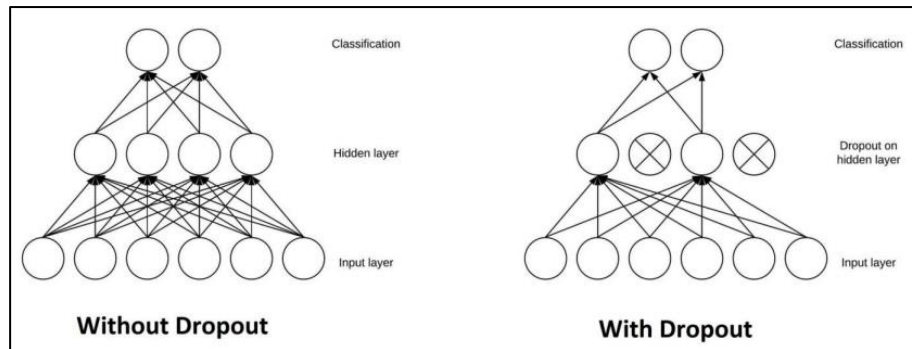


Fig. 3.9 Dropout Layer [17]

5. Activation Functions: -

An activation function in a neural network characterises how the weighted amount of the information is changed into a result from a hub or hubs in a layer of the organisation.

Some of the time the initiation capability is known as a "move capability." If the result scope of the enactment capability is restricted, then, at that point, it could be known as a "crushing capability." Numerous enactment capabilities are nonlinear; what's more, might be alluded to as the "nonlinearity" in the layer or the organisation plan.

The decision of initiation capability to a great extent affects the capacity and execution of the neural network, and different enactment capabilities might be utilised in various pieces of the model. In fact, the actuation capability is utilised inside or after the inner handling of every hub in the network, despite the fact that organisations are intended to involve a similar enactment capability for all hubs in a layer.

An organisation might have three kinds of layers: input layers that take crude contribution from the space, stowed away layers that take input from one more layer and pass result to another layer, and result layers that make a forecast.

All secret layers ordinarily utilise a similar enactment capability. The result layer will commonly utilise a different initiation capability from the secret layers and is subject to the kind of forecast expected by the model. Enactment capabilities are additionally normally differentiable, meaning the first-request subordinate can be determined for a given info esteem. This is required given that brain networks are ordinarily

prepared utilising the backpropagation of blunder calculation that requires the subordinate of expectation to refresh the loads of the model. There are a wide range of kinds of enactment capabilities utilised in neural network organisations, albeit maybe as it were few capabilities utilised by and by for buried and yield layers.

At long last, one of the main boundaries of the CNN model is the actuation capability. They are used to learn and inexact any sort of constant and complex connection between factors of the organisation. In basic words, it concludes which data of the model ought to fire in the forward heading and which ones shouldn't toward the finish of the organisation.

It adds non-linearity to the organisation. There are a few usually utilised initiations works, for example, the ReLU, SoftMax, tanH and the Sigmoid capabilities. Every one of these capabilities have a particular use. For a parallel order CNN model, sigmoid and SoftMax capabilities are favoured for a multi-class grouping, by and large SoftMax is utilised.

The type of activation function are: -

- Sigmoid function: -

The fundamental justification for our use of sigmoid capability is that it exists between (0 to 1). This makes it particularly useful for models where we need to predict the likelihood as a result of the input. The sigmoid is the best option because the range of likelihood for anything is only 0 to 1. The ability can be differentiated. That suggests that there are two locations where we can determine the sigmoid bend's slope. Although the capability is monotonous, the subsidiary is not. A brain network may become unresponsive during the preparation stage due to the strategic sigmoid capacity.

- ReLU: -

The ReLU is the most involved actuation capability on the planet right now. Since, it is utilised in real world all the brain

organisations or profound learning. The function is zero when z is under nothing and function will give the result as z when z is above or equivalent to nothing. It lies between 0 to infinity. The capability and its subsidiary both are monotonic.

In any case, the issue is that every one of the negative qualities become zero quickly which diminishes the capacity of the model to appropriately fit or train from the information. That implies any bad info given to the ReLU actuation capability transforms the worth into zero promptly in the chart, which in turns influences the subsequent diagram by not planning the negative qualities suitably.

- **Leaky ReLU:**
The hole assists with expanding the scope of the ReLU capability. Generally, the worth of an is 0.01 or somewhere in the vicinity. When an isn't 0.01 then it is called Randomised ReLU. In this way the scope of the Leaky ReLU is (vastness to boundlessness). Both Defective and Randomised ReLU capabilities are monotonic in nature. Likewise, their subsidiaries are additionally monotonic in nature.
- **SoftMax activation function: -**
The SoftMax activation function changes the raw outputs of the neural network into a vector of probabilities, basically a likelihood conveyance over the info classes. Consider a multiclass characterization issue with N classes. The softmax enactment returns a result vector that is N passages long, with the section at file I comparing to the likelihood of a specific information having a place with the class I .
The mathematical operation known as Softmax converts a vector of integers into a vector of probabilities, with the probabilities of each element equal to its overall magnitude.
The softmax capacity's use as an enactment capability in a brain network model is its most well-known application in applied AI.

The organisation is specifically made to produce N values, one for each class in the arrangement work, and the results are standardised by using the softmax capability to convert them from weighted aggregate qualities into probabilities that add up to 1. Each value in the softmax capability's output is translated into the likelihood of enrollment for each class.

3.2.6 Training & validation:

After the development of the model the model must be accumulated to prepare it with the accessible informational collection.

Optimizer are utilised to aggregate the model. Aggregating the output will have boundaries: optimizer, loss and metrics. Optimizer are calculations or strategies used to change the traits of the neural networks like loads and learning rate to diminish the misfortunes. Optimizers are utilised to tackle enhancement issues by limiting the capability.

The optimizer check for the learning rate of the given model. We will utilise 'Adam' as our optimizer. Adam is for the most part a great streamlining agent to use for some cases. The Adam optimizer changes the learning rate with the help of learning at each step.

The learning rate decides how quick the ideal loads for the model are determined. A more modest learning rate might prompt more exact loads (in a limited way), however the time it takes to figure the loads will be longer.

We will utilise 'sparse_categorical_crossentropy' for our misfortune capability. This is the most widely recognized decision for grouping. A lower score shows that the model is performing better. To make things much simpler to decipher, we will utilise the 'exactness' metric to see the precision score on the approval set when we train the model. The thought behind preparing and testing any information model is to accomplish greatest learning rate and greatest approval. Better Learning rate and better approval can be accomplished by expanding the train what's more, test information separately.

When the model is effectively gathered, then we can prepare the model with preparing information for 100 cycles, yet as the quantity of emphasis expands, there is an opportunity for overfitting. In this manner we limit the preparation

up to 98% exactness, as we are involving true information for expectation, test information was utilised to approve the model.

Different optimizers used in Neural Networks are:

1. Gradient Descent
2. Adam
3. Stochastic Gradient Descent (SGD)
4. Mini Batch Stochastic Gradient Descent (MB-SGD)
5. SGD with momentum
6. Adaptive Gradient (AdaGrad)
7. AdaDelta
8. RMSprop

3.2.6.1 Gradient descent

The most basic yet most common improvement computation is gradient descent. It is heavily used in estimates for straight relapse and characterisation.

Brain networks also use a slope-plunge computation for backpropagation.

A first-request streamlining computation called Gradient Descent depends on the main request subsidiary of a bad luck capability. It determines how the loads should be modified in order for the capability to be reduced to a minimum. Backpropagation involves moving the misfortune from one layer to the next and changing the model's limits, or loads, based on the misfortunes in order to restrict the misfortune.

3.2.6.2 Adam optimizer

A slope plunge advancement method computation is called adaptive movement estimation. The approach is incredibly effective when dealing with significant problems involving a lot of information or boundaries. It works well and takes minimal RAM. Naturally, the computation is a combination of the inclination drop with force and the RMSP calculations.

An extension of the stochastic angle plunge technique, the Adam optimization approach has recently gained increased popularity for machine learning applications in computer vision and natural language processing. Adam is a streamlined calculation that can be used in place of the conventional stochastic slope plunge approach to iteratively prepare information while refreshing

network loads. Adam was first mentioned in a 2015 ICLR work (banner) by Diederik Kingma from OpenAI and Jimmy Ba from the College of Toronto titled "Adam: A Strategy for Stochastic Enhancement." Adam is portrayed by the developers as combining the advantages of two distinct stochastic gradient descent augmentations.

Every cycle, the Adam's actual step size is roughly constrained by the step size hyper-boundary. This characteristic increases the past unintuitive learning rate hyper-boundary with instinctive comprehension.

The Adam update rule's step size is independent of the magnitude of the inclination, which is helpful for traversing areas with modest angles (for example, saddle focuses or gorges). SGD fights to quickly navigate through these locations.

Adam was designed to combine the advantages of RMSprop, which performs brilliantly in online environments, and AdaGrad, which performs admirably with modest slopes. Having both of these enables us to work with Adam on a wider range of projects. Adam can also be viewed as a combination of RMSprop and SGD with energy.

3.2.6.3 Stochastic gradient descent

It is a different version of gradient descent. It makes too many tries to update the model's limits. In this, after calculating misfortune on each prepared model, the model boundaries are adjusted. In contrast to Slope Plummet, if the dataset has 1000 lines, SGD will refresh the model boundaries more than once in a single pattern of dataset.

3.2.6.4 MB-SGD

The most effective gradient descent algorithm is the mini batch. It is an improvement for both regular slope drops and SGDs. After each bunch, the model bounds are refreshed. The borders are then renewed after each clump, dividing the dataset into various batches.

3.2.6.5 SGD with momentum

There is growing momentum to reduce the large SGD discrepancy and loosen the union. It minimises the transition to the unimportant course and speeds up

the blending toward the important bearing. In this method, the additional hyperparameter referred to as energy, denoted by the symbol ϵ , is used.

3.2.6.6 AdaGrad

The fact that the learning rate is constant for all parameters and for each cycle is one of the limitations of all optimizers. AdaGrad optimizer modifies the pace of learning. For each boundary and time step t , the learning rate η is altered. A second request has been sent to the optimizer algorithm. It functions using the error function's derivative.

3.2.6.7 AdaDelta

It is an expansion of AdaGrad which will in general eliminate the smalling rate of learning issue of it. Rather than aggregating recently squared slopes, AdaDelta will minimize the screen amassed old inclinations to few decent size w . For it dramatically moving normal is utilised as opposed to the amount of the multitude of angels.

3.2.6.8 RMSprop

The RMSprop optimizer is like the slope plummet calculation with force. The RMSprop optimizer confines the motions in the upward heading. Subsequently, we can build our learning rate and our calculation could steer bigger strides in the even bearing uniting quicker. The contrast between RMSprop and gradient descent is in how the slopes are determined. The accompanying conditions show how the slopes are determined for the RMSprop and gradient with momentum. The worth of force is signified by beta and is normally set to 0.9. In the event that you are not keen on the number related to the enhancer, you can simply skirt the accompanying conditions.

3.2.7 Model evaluation & prediction:

For real world image characterization expectation, we really want to do a little picture pre-processing on the certifiable pictures as model preparation was finished with greyscale raster pictures. The means of means pre-processing are:

1. Loading image
2. Convert the image to greyscale
3. Resize the image to 28x28

4. Converting the image into a matrix form
5. Reshape the matrix into 28x28x1

After pre-processing, we anticipate the mark of the picture by passing the pre-processed image through the neural network. The result we get is a rundown of 10 enactment values 0 to 9, individually. The position having the most elevated esteem is the anticipated name for the picture.

Neural networks are the name given to these patterns. It teaches the computer system to perform actions that naturally occur for individuals. Artificial neural networks (ANN), autoencoders, recurrent neural networks (RNN), and reinforcement learning are a few types of models used in deep learning. The Convolutional Neural Network (CNN) or the ConvNet is one particular model that has made a significant contribution to the field of computer vision and image analysis.

CNNs are a subclass of deep learning networks that are typically used for analysing visual images since they can recognise and rank certain highlights from photos. Their uses include image and video recognition, image characterisation, clinical image analysis, computer vision, and regular language processing.

There are two groups of methods for evaluating a model's exhibition: holdout and cross validation. The two approaches use a test set, or data that the model is blind to, to evaluate model execution. It's not advised to use the data we used to build the model to evaluate it. The reason for this is that because our model will only remember the entire preparation set, it will always predict the correct name for any point in the preparation set. Overfitting is the term for this.

Holdout: The reason for holdout assessment is to test a model on unexpected information in comparison to it was prepared on. This gives a fair-minded gauge of learning execution.

Cross validation: Cross-validation is a method that includes dividing the first perception dataset into a preparation set, used to prepare the model, and an autonomous set used to assess the examination.

3.3 System Configurations

3.3.1 Software requirements:

The software requirements for the project are: -

- Operating System: - Windows 10 or above.

- Language required: -
Python: - Python is an extremely famous broadly useful interpreted, interactive, object-oriented, and high-level programming language. Python is powerfully composed and trash gathered programming language. It was made by G Rossum in the 1980s. Like, Python source code is additionally accessible under the GNU GPL.
Python is extremely high popular and every one of the significant organizations are searching for incredible Python Software engineers to develop sites, programming parts, and implications or do the job with Information Science, simulated intelligence, and ML advancements. At the point for development of the instructional exercise in 2022, there is a high lack of Python Developers whereas market requests a greater number of Python Software engineers because of its application in Machine Learning and Artificial Intelligence.

- Software or IDE's required: -
 1. Anaconda: - Anaconda Python is the ideal stage for novices who need to learn Python. It's not difficult to introduce, and you can begin rapidly with the included Jupyter Notebook. Furthermore, Anaconda Python has many elements and libraries that you can use for your undertakings.

Anaconda Python is a free, open-source stage that permits you to compose and execute code in the programming language Python. It is by continuum.io, an organization that has practical experience in Python improvement. The Anaconda stage is the most well-known method for learning and use Python for logical processing, information science, and AI. It is utilized by over of thirty million individuals overall and is accessible for Windows, macOS, and Linux.

2. PyCharm: - PyCharm is a hybrid platform created by JetBrains as an IDE for Python. It is usually utilized for Python application improvement. A portion of the unicorn associations like Twitter, Facebook, Amazon, and Pinterest use PyCharm as their Python IDE. It upholds two renditions: v2.x and v3.x. We can run PyCharm on Windows, Linux, or Macintosh operating system. Also, it contains modules and bundles that assist software engineers with creating programming involving Python quicker than expected and with insignificant exertion. Further, it can likewise be modified by the prerequisites of designers.

3. Google Colab: - Google Colab was created by Google to give free admittance to GPU's and TPU's to any individual who needs them to fabricate an AI or profound learning model. Google Colab can be characterised as a superior rendition of Jupyter Journal.

Google Colab Features: -

Google Colab gives lots of invigorating elements that any cutting-edge IDE offers, and considerably more. Probably the most astonishing highlights are recorded underneath. Intuitive instructional exercises to learn AI and brain organisations. Compose and execute Python 3 code without having a nearby arrangement. Execute terminal orders from the Notebook. Import datasets from outer sources like

Kaggle. Save your Journals to research Drive. Import Notebooks from Google Drive. Free cloud administration, GPUs and TPUs. Incorporate with PyTorch, Tensor Stream, Open CV. Import or distribute straightforwardly from/to GitHub.

3.3.2 Hardware requirements:

The hardware requirements for the project are: -

- CPU: - Intel i5/AMD Ryzen 5 or above.
- GPU: - NVIDIA GeForce GTX1650 or equivalent.
- RAM: - 8GB or more.
- Hard Disk Drive (HDD): - 500GB or more.
- Solid State Drive (SSD): - 256GB or more.

CHAPTER 4: EXPERIMENTS & RESULT ANALYSIS

4.1 Performance Analysis at Various Stages

4.1.1 Visualization of the dataset:

This shows that how our dataset looks like: -

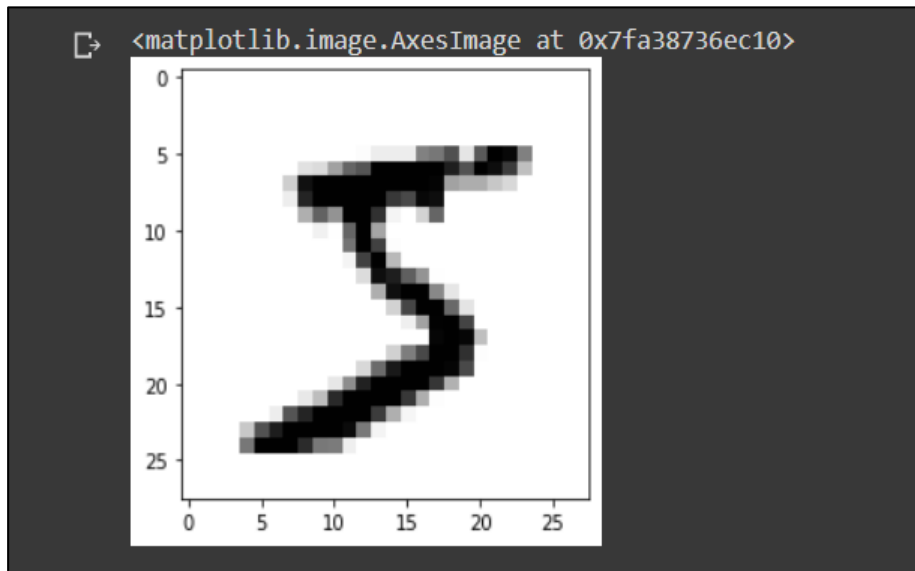


Fig. 4.1 Data Visualization

4.1.2 Dataset after normalization:

This shows the dataset after normalization: -

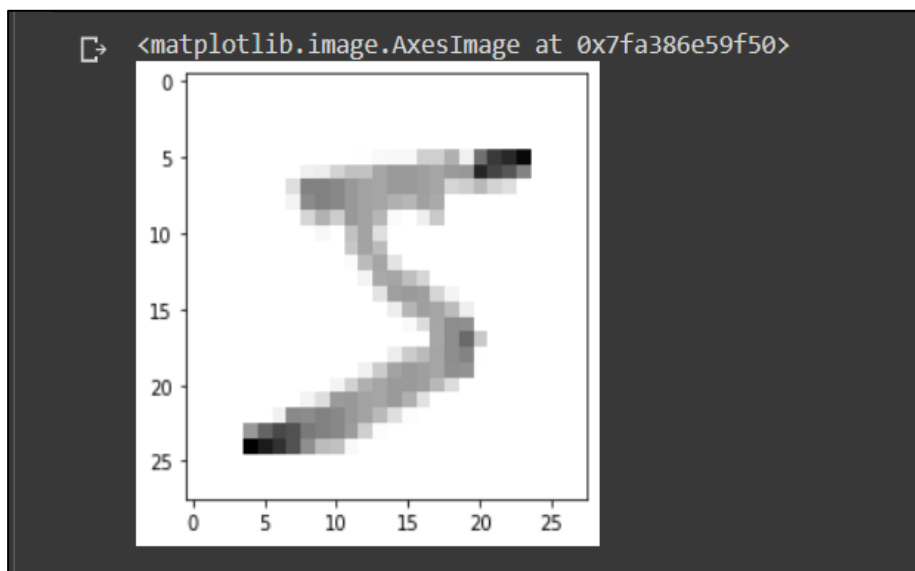


Fig. 4.2 Dataset after Normalization

4.1.3 Model summary:

Model summary talks about the model, it gave the all information about the model. It gave the information like how many layers are used and what kind of layers, activation functions and what kind of pooling layers are used to prepare the model.

```
Model: "sequential"
-----
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 64)	640
activation (Activation)	(None, 26, 26, 64)	0
max_pooling2d (MaxPooling2D)	(None, 13, 13, 64)	0
dropout (Dropout)	(None, 13, 13, 64)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	36928
activation_1 (Activation)	(None, 11, 11, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 64)	0
dropout_1 (Dropout)	(None, 5, 5, 64)	0
conv2d_2 (Conv2D)	(None, 3, 3, 64)	36928
activation_2 (Activation)	(None, 3, 3, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 1, 1, 64)	0
dropout_2 (Dropout)	(None, 1, 1, 64)	0
flatten (Flatten)	(None, 64)	0
dense (Dense)	(None, 64)	4160
activation_3 (Activation)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2080
activation_4 (Activation)	(None, 32)	0
dense_2 (Dense)	(None, 10)	330
activation_5 (Activation)	(None, 10)	0

```
=====  
Total params: 81,066  
Trainable params: 81,066  
Non-trainable params: 0
```

Fig. 4.3 Summary of the model

4.1.4 Training the model on training data:

This shows the training of the model on the training data: -

```
Epoch 1/10
2400/2400 [=====] - 16s 4ms/step - loss: 0.2658 - accuracy: 0.9165 - val_loss: 0.1047 - val_accuracy: 0.9676
Epoch 2/10
2400/2400 [=====] - 9s 4ms/step - loss: 0.0869 - accuracy: 0.9737 - val_loss: 0.0688 - val_accuracy: 0.9800
Epoch 3/10
2400/2400 [=====] - 9s 4ms/step - loss: 0.0641 - accuracy: 0.9807 - val_loss: 0.0679 - val_accuracy: 0.9808
Epoch 4/10
2400/2400 [=====] - 8s 4ms/step - loss: 0.0510 - accuracy: 0.9842 - val_loss: 0.0506 - val_accuracy: 0.9844
Epoch 5/10
2400/2400 [=====] - 9s 4ms/step - loss: 0.0416 - accuracy: 0.9873 - val_loss: 0.0542 - val_accuracy: 0.9845
Epoch 6/10
2400/2400 [=====] - 8s 4ms/step - loss: 0.0343 - accuracy: 0.9893 - val_loss: 0.0728 - val_accuracy: 0.9798
Epoch 7/10
2400/2400 [=====] - 9s 4ms/step - loss: 0.0295 - accuracy: 0.9906 - val_loss: 0.0501 - val_accuracy: 0.9881
Epoch 8/10
2400/2400 [=====] - 9s 4ms/step - loss: 0.0245 - accuracy: 0.9922 - val_loss: 0.0463 - val_accuracy: 0.9879
Epoch 9/10
2400/2400 [=====] - 9s 4ms/step - loss: 0.0212 - accuracy: 0.9932 - val_loss: 0.0599 - val_accuracy: 0.9858
Epoch 10/10
2400/2400 [=====] - 9s 4ms/step - loss: 0.0201 - accuracy: 0.9937 - val_loss: 0.0498 - val_accuracy: 0.9880
```

Fig. 4.4 Training of the model

4.1.5 Model evaluation:

This shows the evaluation of the model: -

```
313/313 [=====] - 1s 3ms/step - loss: 0.0498 - accuracy: 0.9880
Test Loss on 313 test samples = 0.04978605732321739
Validation Accuracy on 313 test samples = 0.9879999756813049
```

Fig. 4.5 Validation accuracy of the model

4.1.6 Visualizing the model accuracy:

Accuracy and loss curve are the two most important evaluating tools: -

- Accuracy Curve: - Accuracy is a technique for estimating an order model's exhibition. It is ordinarily communicated as a rate. Accuracy is the count of forecasts where the anticipated worth is equivalent to the genuine worth. It is double (valid/misleading) for a specific example. Exactness is in many cases charted and checked during the preparation stage however the worth is frequently connected with the general or last model precision. Exactness is simpler to decipher than misfortune.
- Loss Curve: - A loss function, otherwise called a cost function, considers the probabilities or vulnerability of an expectation in view of how much the forecast differs from the genuine worth. This gives us a more nuanced view into how well the model is performing. Dissimilar to Accuracy, loss isn't a rate — it is a summation of the mistakes made for each example in preparing or approval sets. Loss is many times

utilised in the preparation cycle to see as the "best" boundary values for the model (for example loads in neural network).

The accuracy curve for the model: -

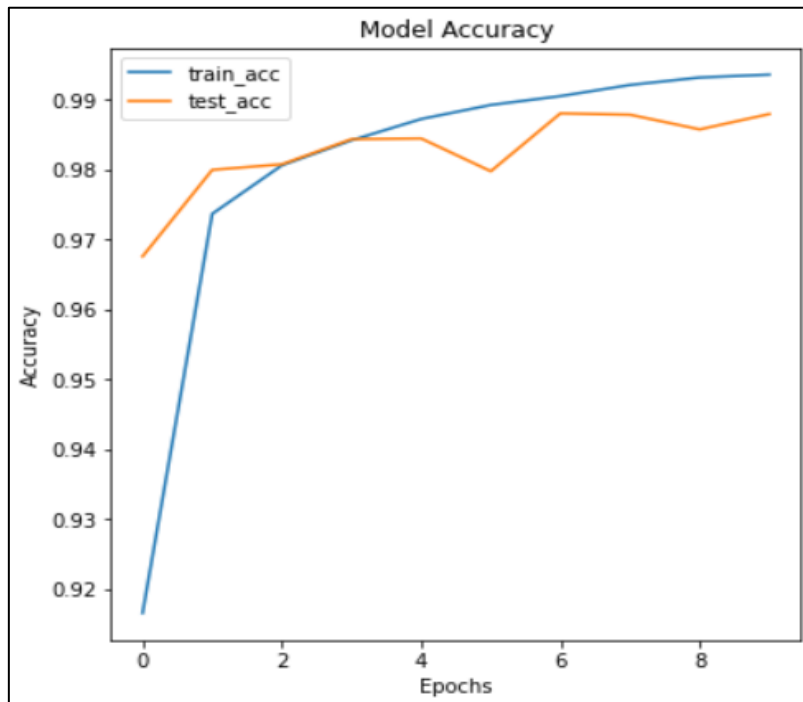


Fig. 4.6 Accuracy Curve

The loss curve for the model: -

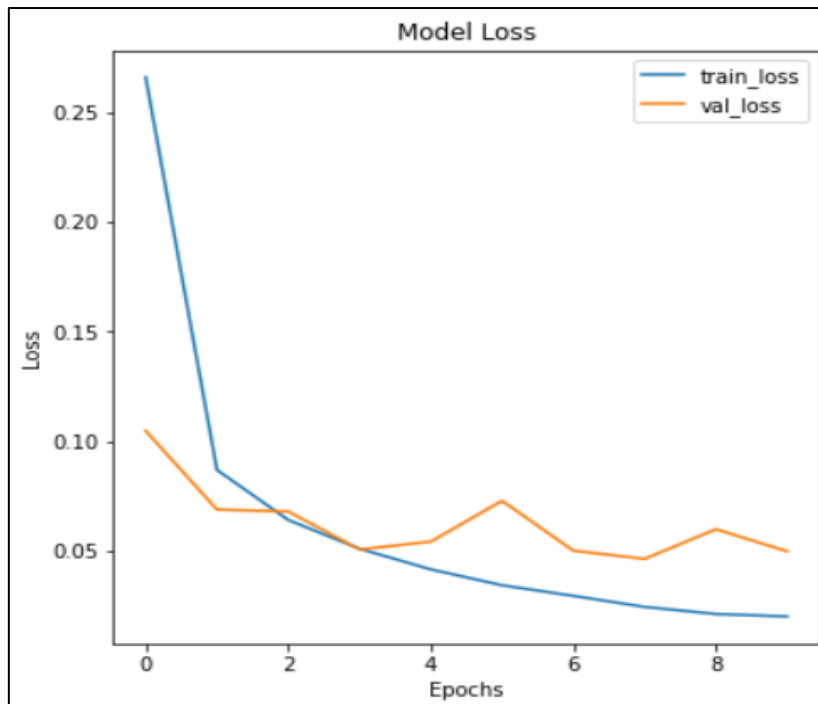


Fig. 4.7 Loss Curve

4.1.7 Confusion matrix:

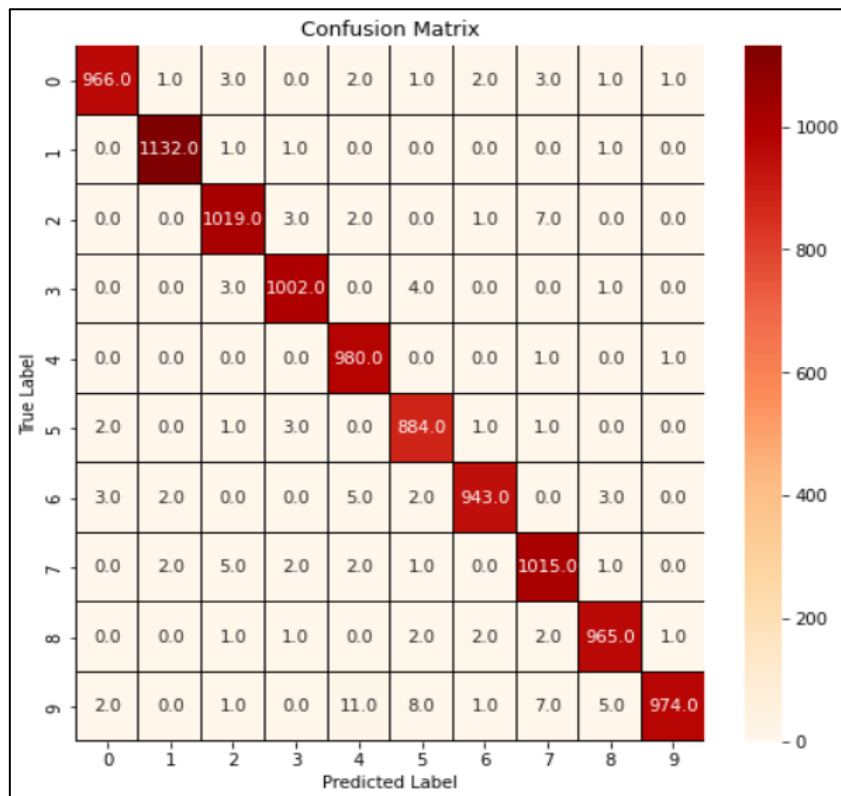


Fig. 4.8 Confusion Matrix

4.1.8 Classification report:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	980
1	1.00	1.00	1.00	1135
2	0.99	0.99	0.99	1032
3	0.99	0.99	0.99	1010
4	0.98	1.00	0.99	982
5	0.98	0.99	0.99	892
6	0.99	0.98	0.99	958
7	0.98	0.99	0.98	1028
8	0.99	0.99	0.99	974
9	1.00	0.97	0.98	1009
accuracy			0.99	10000
macro avg	0.99	0.99	0.99	10000
weighted avg	0.99	0.99	0.99	10000

Fig. 4.9 Classification Report

4.1.9 Making predictions:

- The input given to the model: -

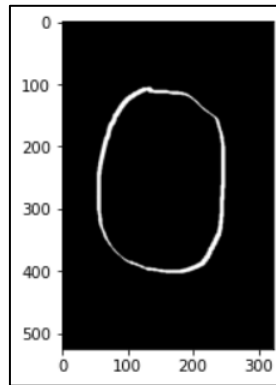


Fig. 4.10 Input Image for prediction of Zero

The output given by model: -

```
1/1 [=====] - 0s 21ms/step  
The model has predicted the given image as: - 0
```

Fig. 4.11 Output for the Input Image of Zero

- The input given to the model: -

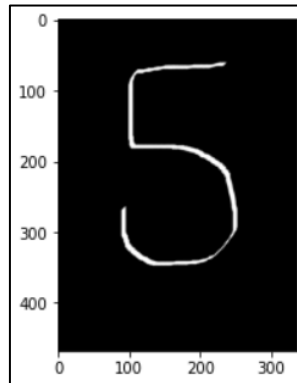


Fig. 4.12 Input Image for prediction of Five

The output given by model: -

```
1/1 [=====] - 0s 20ms/step  
The model has predicted the given image as : - 5
```

Fig. 4.13 Output for the Input Image of Five

- The input given to the model: -

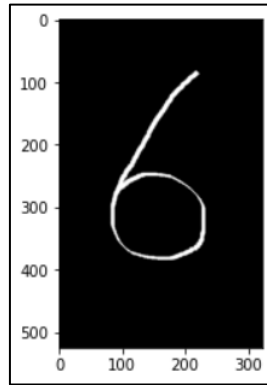


Fig. 4.14 Input Image for prediction of Six

The output given by model: -

```
1/1 [=====] - 0s 19ms/step  
The model has predicted the given image as : - 6
```

Fig. 4.15 Output for the Input Image of Six

- The input given to the model: -

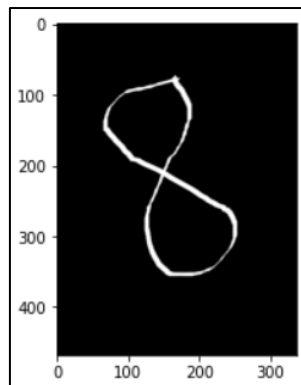


Fig. 4.16 Input Image for prediction of Eight

The output given by model: -

```
1/1 [=====] - 0s 20ms/step  
The model has predicted the given image as : - 8
```

Fig. 4.17 Output for the Input Image of Eight

CHAPTER 5: CONCLUSIONS

5.1 Conclusion

Our project Digit Recognition system deals with identifying and predicting the digits. The principal reason for this project is to design an efficient method for the recognition of digit written by a user.

There are various models present in Machine Learning which we can use the prediction of the digits. We analyse the models and studied their characteristics. We also studied different research papers in order to choose our model or algorithm. While doing the literature survey, we found that the model like K Nearest Neighbour, SVM (Support Vector Machine) and RFC (Random Forest Classifier) having been utilised in a lot research.

We also find that with the help of these models the accuracy achieved is nearly up to 99%. And no further better model using this model can be created. So, we are here trying to use the CNN (Convolutional Neural Network) which has a lot of future scope. The model has variations in accuracy if we do changes in the model or we can create our own model.

In this project, CNN (Convolutional Neural Network) model is utilised to achieve a high-performance model which is efficient for the digit recognition system. We are using principles of CNN in order to make our model. Before preparing the model, we have done the preprocessing of the data and we have normalized the data in order to have the best data to train our model.

In our model we have used three convolutional layers and with each convolutional layer we have used ReLU as the activation function and we have used three two-dimensional Pooling layers with the convolutional layers. We have also used the dropout layer with each convolutional layer to prevent the overfitting of the model.

We have used three fully connected layer in which two have ReLU as the activation function and the last fully connected layer has the Softmax as the activation function to map the output to multiclass classification.

We have used “sparse_categorical_crossentropy” as the loss measure and we have used the Adam as the optimizer to prepare our model.

We have trained the model on various fitting configurations that is on different number of epochs and batch size. While working on the model we found that the best model is trained on a batch size of 25 and number of epochs are 10.

For the other parameters the accuracy got decreased and the model gave us the wrong predictions. In our successfully created model, the accuracy is more than 98%. The accuracy for the train dataset is approximately 99%. While the accuracy for the test dataset is about 98%. We have used multiple accuracy measuring tools in order to reach to a result. We have created the curves, confusion matrix and run our model on the external input also to validate the model. So, we can conclude that the CNN model that we have created is giving good result and it performed very well.

5.2 Future Scope

In this Project, for implementing a Convolutional Neural Network (CNN) a thorough examination is needed to determine optimal choice for the different parameters.

Neural networks can be investigated with various decisions of number of hidden layers and number of nodes in each hidden layer. The Softmax activation function is utilised in the neural network executed in this project for the output layer. Nonetheless, other activation functions can likewise be investigated for those layers. The performance of our deep neural network most certainly relies upon that. And if we can find the more accurate and precise activation function and parameters than the accuracy of the model can be increased.

As a future work, we can address the confusing matches into digit classes by consolidating extra features in our further study. Also, we can change the parameters of the CNN model to improve the model.

Also, we can improve the deep CNN model by utilising the progressed information increase method and further developing a CNN model to show up at an official choice about the digit class. So that our CNN model will be able to classify the confusing pairs more accurately.

5.3 Applications

The digit recognition system is one of the most advanced topics in the field of pattern recognition. The digit recognition system can be used in postal systems, banks and form data entry. The postal system can use it to sort the postal mails and the banks can use it to process the cheques. The system can be used to detect the vehicle numbers from the traffic. This has a vast application as the vehicle number detection can be used by the police to generate automatic E-challan. So, these are some of the applications of the project or the fields where the project can be used or utilised.

REFERENCES

- [1] Kumar Gaurav, & Bhatia Pradeep Kumar. (2013). Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition. *International Conference on Emerging Trends in Engineering and Management*, 3(3), 14–22.
- [2] Arora, S., Bhattacharjee, D., Nasipuri, M., Basu, D. K., & Kundu, M. (2008). Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition. *2008 IEEE Region 10 and the Third International Conference on Industrial and Information Systems*, 1–6.
- [3] Brakensiek A, Rottland J, Kosmala A, Rigoll G (2004.). Off-Line Handwriting Recognition Using Various Hybrid Modeling Techniques And Character N-Grams. In Eprints-Book-Title.
- [4] Bajaj, R., Dey, L., & Chaudhury, S. (2002). Devnagari numeral recognition by combining decision of multiple connectionist classifiers. *Sadhana*, 27(1), 59–72.
- [5] Pirlo, G., & Impedovo, D. (2011). Fuzzy-Zoning-Based Classification for Handwritten Characters. *IEEE Transactions on Fuzzy Systems*, 19(4), 780–785.
- [6] Patnaik, S. S., & Panda, A. K. (2012). Particle Swarm Optimization and Bacterial Foraging Optimization Techniques for Optimal Current Harmonic Mitigation by Employing Active Power Filter. *Applied Computational Intelligence and Soft Computing*, 2012, 1–10.
- [7] Hanmandlu, M., Murthy, O. V. R., & Madasu, V. K. (2007). Fuzzy Model Based Recognition of Handwritten Hindi Characters. *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)*, 454–461.
- [8] Neves, R. F. P., Lopes Filho, A. N. G., Mello, C. A. B., & Zanchettin, C. (2011). A SVM based off-line handwritten digit recognizer. *2011 IEEE International Conference on Systems, Man, and Cybernetics*, 510–515.
- [9] Kumar, Dr. N., & Beniwal, H. (2018). Survey on Handwritten Digit Recognition using Machine Learning. *International Journal of Computer Sciences and Engineering*, 06(05), 96–100.

- [10] Handwritten Character Recognition using Neural Network and Tensor Flow. (2019). *International Journal of Innovative Technology and Exploring Engineering*, 8(6S4), 1445–1448.
- [11] Lakshmi S.V.S.S., Pragnya K. Sri, Krishna K. Rama, Sahithi B. Sri, & Lokesh M. (2021). Handwritten Digit Recognition using Machine Learning in Python. *International Journal of Management, Technology And Engineering*, 11(6), 104–107.
- [12] Siddique, Fathma & Sakib, Shadman & Siddique, Abu Bakr. (2019). Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers.
- [13] Sree Yellina Navya, Jasti Sreekar Kumar, Jampani Bharath Chandra, & Rao M. R. Narsing. (2020). Inspecting of CNN and ANN Algorithms using Digit Recognition Model. *International Research Journal of Engineering and Technology*, 7(6), 5867–5871.
- [14] Yellapragada SS Bharadwaj, Rajaram P, Sriram V.P., Sudhakar S, & Kolla Bhanu Prakash. (2020). Effective Handwritten Digit Recognition using Deep Convolution Neural Network. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(2), 1335–1339.
- [15] Lakshmi S.V.S.S., Pragnya K. Sri, Krishna K. Rama, Sahithi B. Sri, & Lokesh M. (2021). Handwritten Digit Recognition using Machine Learning in Python. *International Journal of Management, Technology And Engineering*, 11(6), 104–107.
- [16] Gupta, A., & Narwaria, Dr. R. P. (2021). Review on Deep Learning Handwritten Digit Recognition using Convolutional Neural Network. *International Journal of Recent Technology and Engineering (IJRTE)*, 9(5), 245–247.
- [17] Abu Ghosh, M. M., & Maghari, A. Y. (2017). A Comparative Study on Handwriting Digit Recognition Using Neural Networks. *2017 International Conference on Promising Electronic Technologies (ICPET)*, 77–81.

APPENDICES

Source Code: -

Code for importing libraries, data and data preprocessing: -

```
import tensorflow as tf

import matplotlib.pyplot as plt

import cv2

data= tf.keras.datasets.mnist

(x_train,y_train),(x_test,y_test)=data.load_data()

x_train.shape

x_test.shape

plt.imshow(x_train[151],cmap=plt.cm.binary)

print(x_train[0])

x_train = tf.keras.utils.normalize(x_train,axis=1)

x_test = tf.keras.utils.normalize(x_test,axis=1)

plt.imshow(x_train[0],cmap=plt.cm.binary)

import numpy as np

IMG_SIZE=28

x_trainr=np.array(x_train).reshape(-1,IMG_SIZE,IMG_SIZE,1)

x_testr=np.array(x_test).reshape(-1,IMG_SIZE,IMG_SIZE,1)

print("Training Samples dimension",x_trainr.shape)
```

```
print("Testing Samples dimension",x_testr.shape)
```

Code for model creation: -

```
from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense,Dropout,Activation,Flatten,Conv2D,MaxPooling2D

model=Sequential()

# 1st Convolution layer

model.add(Conv2D(64,(3,3),input_shape=x_trainr.shape[1:]))

model.add(Activation("relu"))

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Dropout(0.3))

# 2nd Convolution layer

model.add(Conv2D(64,(3,3)))

model.add(Activation("relu"))

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Dropout(0.3))

# 3rd Convolution layer

model.add(Conv2D(64,(3,3)))

model.add(Activation("relu"))

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Dropout(0.3))
```

```

# Fully Connected layer

model.add(Flatten())

model.add(Dense(64))

model.add(Activation("relu"))

# Fully Connected layer 2

model.add(Dense(32))

model.add(Activation("relu"))

# Last Fully Connected layer 3

model.add(Dense(10))

model.add(Activation("softmax"))

model.summary()

print("Total Training Samples = ",len(x_trainr))

```

Code for training and testing: -

```

model.compile(loss="sparse_categorical_crossentropy",optimizer
="adam",metrics=['accuracy'])

history=model.fit(x_trainr,y_train,epochs=10,validation_data=(x
_testr,y_test),batch_size=50,verbose=1)

test_loss , test_acc = model.evaluate(x_testr,y_test)

print("Test Loss on 313 test samples = ",test_loss)

print("Validation Accuracy on 313 test samples = ",test_acc)

predictions=model.predict([x_testr])

plt.figure(figsize = (12,6))

plt.subplot(2,1,1)

```

```
plt.plot(history.history['accuracy'], label="train_acc")
plt.plot(history.history['val_accuracy'], label = "test_acc")
plt.legend()
plt.subplot(2,1,2)
plt.plot(history.history['loss'], label = "train_loss")
plt.plot(history.history['val_loss'], label = "val_loss")
plt.legend()
from sklearn.metrics import classification_report
print(classification_report(y_test,Y_pred_classes))
```