HandWritten Digit Prediction Using Machine Learning *Authors*

1.DILSHAN KUMAR

ABSTRACT

Handwriting recognition has been an active and challenging area . Handwriting recognition system plays a very important role in today's world. At present time it is very difficult to find the correct meaning of handwritten documents as different people have different ways of writing digits or texts.
There are many areas where we need to recognize the words, alphabets and digits like postal addresses, bank cheques etc.
The main aim of this project is to make a GUI based application that allows users to predict the digits drawn on the interface using machine learning and shows the results in percentage how likely the prediction is correct.
The project will be built using python and the application will be trained with the help of dataset from MNIST using linear regression algorithm

Keyword:
HandWritten Digit Prediction, Machine Learning, convolutional neural network, MNIST dataset, linear regression algorithm

1.INTRODUCTION

Machine Learning Is now considered to be one of the biggest innovations in the field of Artificial Intelligence since the microchip. AI is on its way to becoming a crucial part of our daily routine in many people's lives. With the availability of so much data that is processed, it is now possible to build any predictive models that can analyze and study the complex data stored in the dataset to find useful insights and deliver more accurate results. Many companies build various Machine Learning models by using all the data received from people and use this in order to identify profitable opportunities and avoid risks.

Machine Learning is the study of computer algorithms that improves automatically through experience. It provides the ability to automatically learn and improve from experience without being explicitly programmed.

The algorithm finds natural patterns in data that generate insight and help you make better diagnosis.

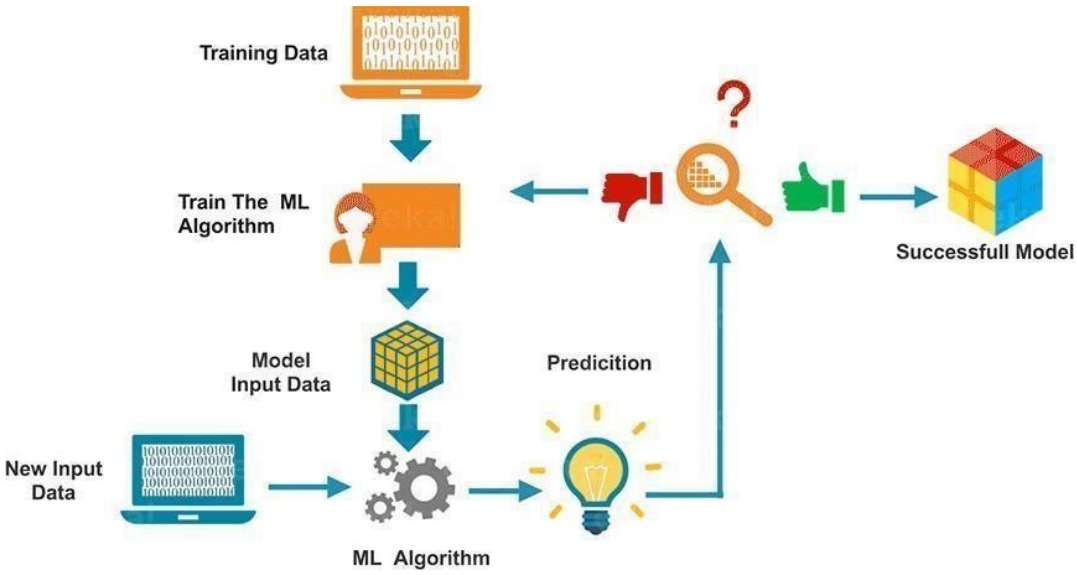


Fig 1.1 Block diagram of the working of Machine Learning

Under Machine Learning is a subset called Deep Learning, where artificial neural networks, algorithms are all based on the neurons of the human brain. These Deep learning models learn from large amounts of data. Similar to how people learn from previous experiences, the same goes for the deep learning algorithm which will perform the tasks repeatedly with each time (iteration), it tweaks the model little by little to improve the outcome. It is referred to as deep learning because the neural network has various layers that enable the model for learning the various patterns, data, etc.Because the amount of data the users generate every day is increasing at a staggering rate, it's the resource (data) that makes deep learning possible as these models need tremendous amounts of data to learn. This increase in data creation is one of the reasons that deep learning is used due to its capabilities of learning with strong computing power that is available today. This allows machines to solve complex problems even when the dataset is very diverse, unconstructed and interconnected. The

more the model learns, the better they perform and better the capabilities to solve with different datasets.

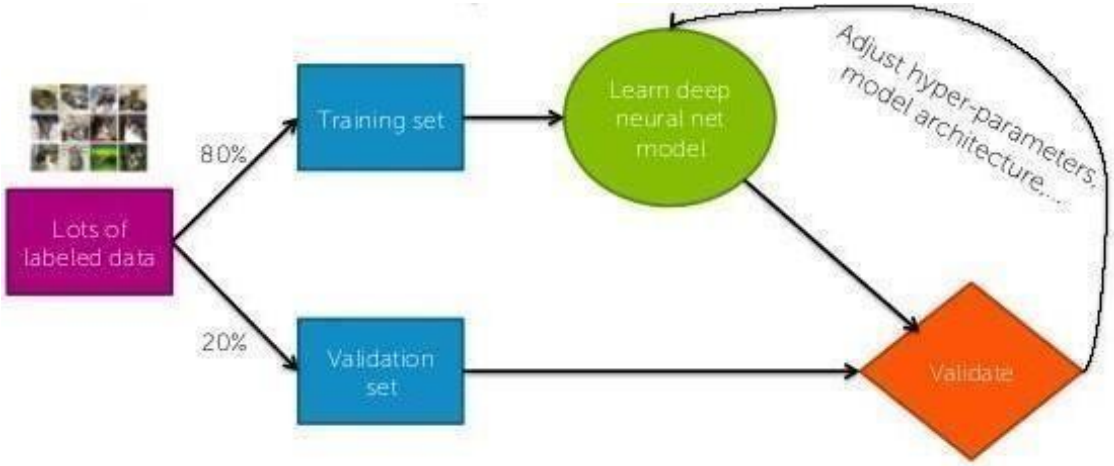


Fig 1.2 Block Diagram of Deep Learning

This project will demonstrate and explain a deep learning project called Handwritten Digit Recognizer. This illustrates how the digit is recognized and shows the prediction of how the number is recognized from the dataset that is fed into the model. The Handwritten Digit Recognition is the ability of the computers to recognize the human handwritten digits. With the help of the dataset, this allows the model to analyze and predict the outcome of the number fed into the model with various flavors of the handwriting available.

1.1 Purpose of Study:

The purpose for this project is to allow us to explain the implementation of the deep learning models and how the models use the dataset to analyze and predict the outcome. This project allows us to study the neural network of the Handwritten Digit Recognizer and the working of the model to see how the Deep Learning process works in the background. It shows the working of how the number fed into the model is recognized and displays the number that is recognized after analysis and comparing it with the dataset fed to the model. With the help of the MNIST dataset and the CNN

neural network, the project can explain the purpose, working and the benefits of deep learning in our daily lives

1.2Problem Statement

To develop and implement a Deep Learning model called Handwritten Digit Recognizer using the MNIST dataset, CNN in Python.

1.3 Motivation of Project

In the current world, Machine Learning is one of the most in demand technologies in today's market. The applications that Machine Learning can be used range from selfdriving vehicles to predicting deadly diseases and analyzing the pattern that can detect the type of disease and other various things. With the availability of so much data, it is possible now to build any predictive models that can have the ability to study and analyze the various patterns or complex data to find useful information and deliver the outcome needed. This high demand for Machine Learning skills is the motivation behind this project. The motivation of this project is to illustrate Deep

Learning with Handwritten Digit Recognizer and explain the process of the Neural Network used in this project.

2.1 MNIST Dataset(SYSTEM DESIGN)

To make this model, the data set needs to be imported which is very important in the analysis and recognizing the digits which are fed into the model. The dataset used in the project is called the **MNIST** dataset.

The **MNIST dataset (Modified National Institute of Standards and Technology database)** is a large database of handwritten digits that is used in different Deep Learning models for training various image pattern processing systems. This dataset contains 60,000 training images and 10,000 testing images. The MNIST dataset is normalized to fit into a 28x28 pixel bounding box, so that they are able to fit in all flavors of different types of handwriting of a specific digit.

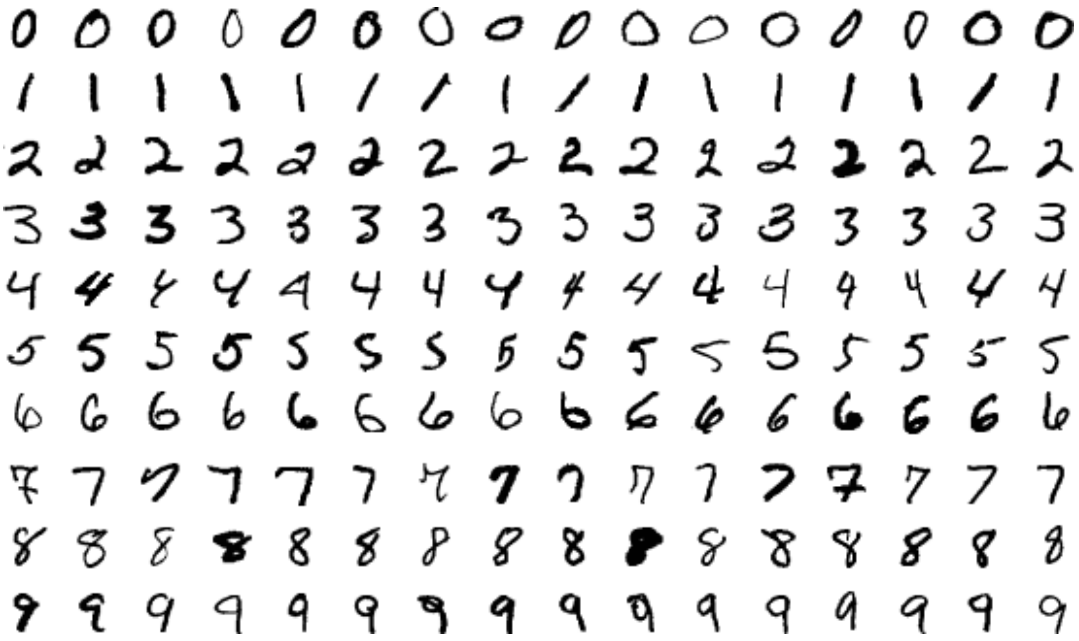


Fig 2.1 MNIST dataset of various handwritten digits

Each image is an array of floating point numbers which are represented in grayscale intensities ranging from 0(black) to 1(white).

2.2 Architecture

In the HandWritten Digit Prediction, the layers in the Neural Network perform the recognition operations that will alter the data along with the intent of learning the specific feature to the data. In the CNN architecture, the layers that work behind the process of the HandWritten Digit recognizer are as follows:

1. **Convolution:** In this layer, the model puts the input images through a set of convolutional filters, where each of the filters activates certain features from the images.
2. **Rectified Linear Unit:** The ReLU allows faster and more effective training of the deep learning model by mapping all the negative values to zero and maintaining only the positive values. This is referred to as activation because only the activated features are carried out and take place in the next layer.

3. **Pooling:** This layer simplifies the output by performing non-linear downsampling, that is, it reduces the number of parameters that the neural network learns.

All these operations performed by the above-mentioned layers are repeated over tens or hundreds of layers in the network, with each layer learning to identify different features of the data from the dataset.

Let's consider an example of how CNN works with a diagram of a picture which shows the step by step operation of the neural network.

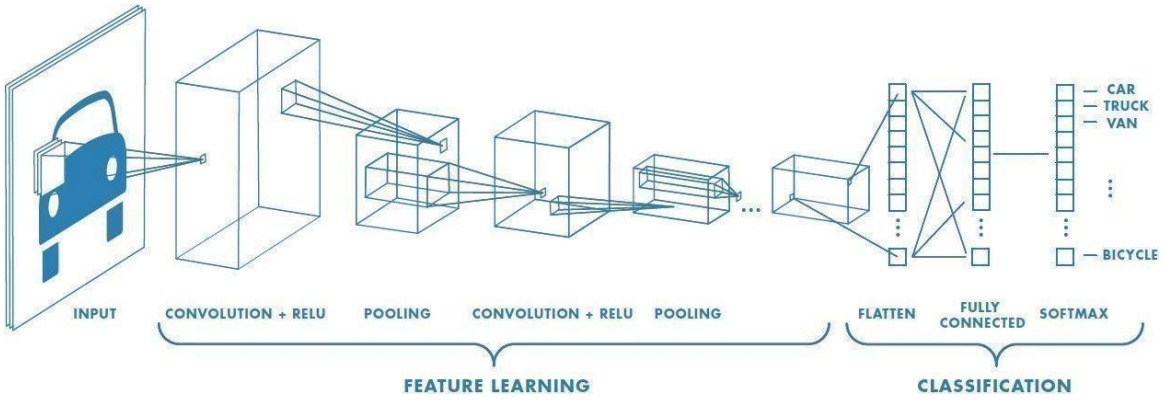


Fig 2.2 Example of the working of the Convolution Neural Network

As we can see in the figure 2.2, the network consists of many convolution layers. The filters are applied to each training image at different resolution, and the output of each convolved image is used as the input of the next layer in the neural network. After learning all the features in many layers in the model, the architecture of the CNN shifts to classification.

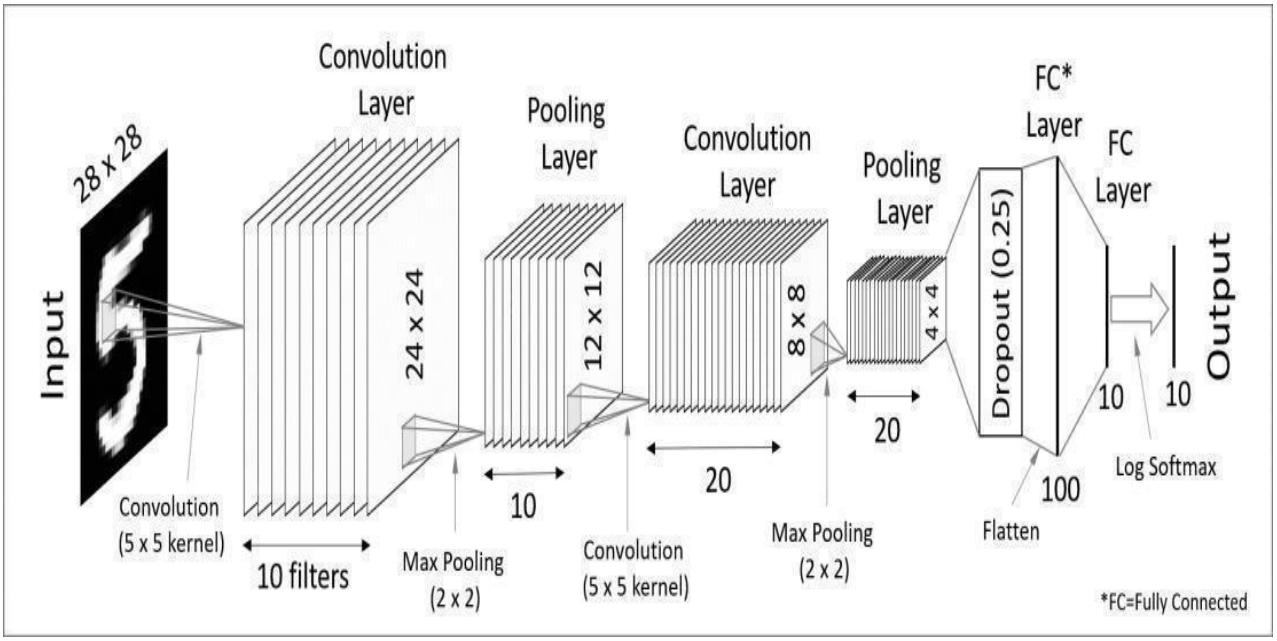


Fig 2.3 Architecture diagram

In the classification layer, the next to last layer consists of a fully connected layer that outputs to a vector K dimension, where K is the number of classes that the network will be able to predict the outcome. This contains the probabilities of the outcome for each class of the data being classified.

So in the same way, in the Handwritten Digit Recognizer the process is performed in the same way.

How does it work for recognizing a digit? Let us consider a filter size of 3x3 and an image size of 5x5. This will perform an element-wise multiplication process between the image pixel value that matches the size of the kernel (filter) and the kernel will sum up. This will provide a single value to the feature cell (the outcome fixed into the cell).

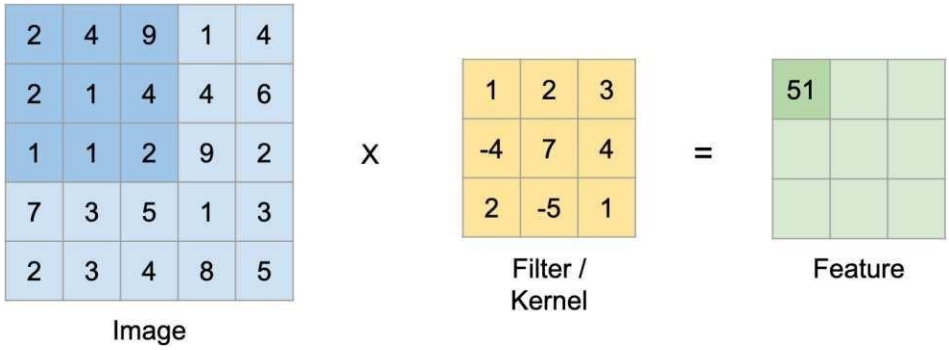


Fig 2.4(a) Working of the Kernel filter

The calculation is shown below:

2*1 + 4*2 + 9*3 + 2*(-4) + 1*7 + 4*4 + 1*2 + 1*(-5) + 2*1 = 51

The filter continues to run and analyze further on the image of the dataset and produce new values into the feature block as shown below.

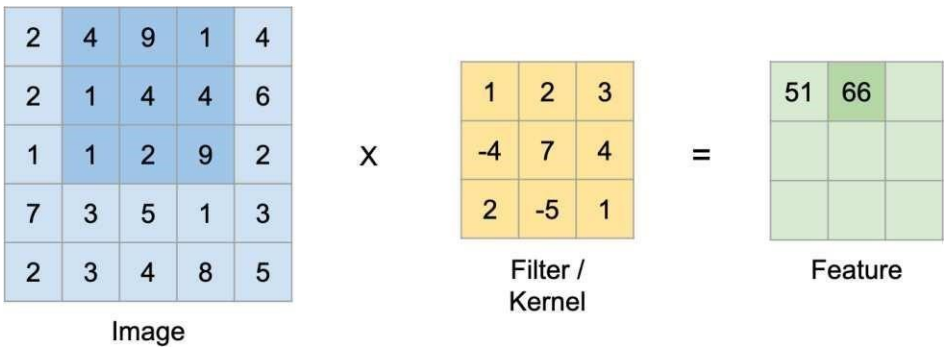


Fig 2.4(b) Continued process of Kernel filter

4*1 + 9*2 + 1*3 + 1*(-4) + 4*7 + 4*4 + 1*2 + 2*(-5) + 9*1 = 66

And the same process goes on to complete the feature block.

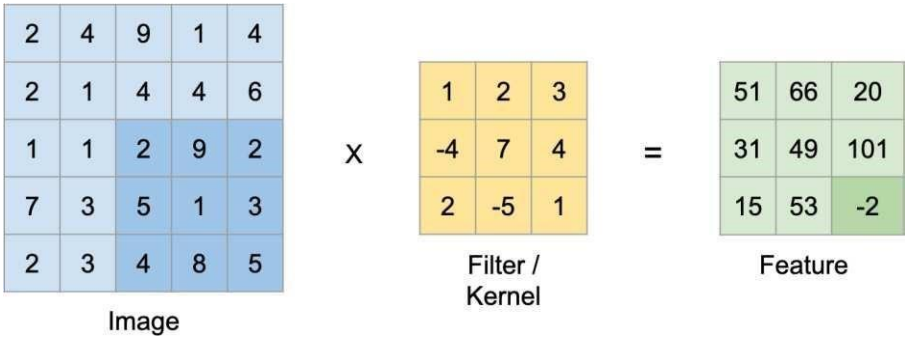


Fig 2.4(c) Completed process of Kernel filter

2*1 + 9*2 + 2*3 + 5*(-4) + 1*7 + 3*4 + 4*2 + 8*(-5) + 5*1 = -2

As you can see from the figure 2.4(c), this process is sliding the kernel by 1 pixel. This method is called striding. The kernel can be moved by different stride values to extract different kinds of feature data. Plus, the amount of stride that is chosen affects the size of the feature extracted.

The equation to calculate the size of the feature block for the given kernel size is shown below:

Feature size= ((Image size-Kernel size) /Stride) + 1

Now use these values from the above example to verify it.

Feature size = ((5-3) /2)) + 1 = 2

From the above equation, it shows that with a stride of 2 the kernel of size 3x3 on an image of size 5x5 would be able to extract the feature size of 2.

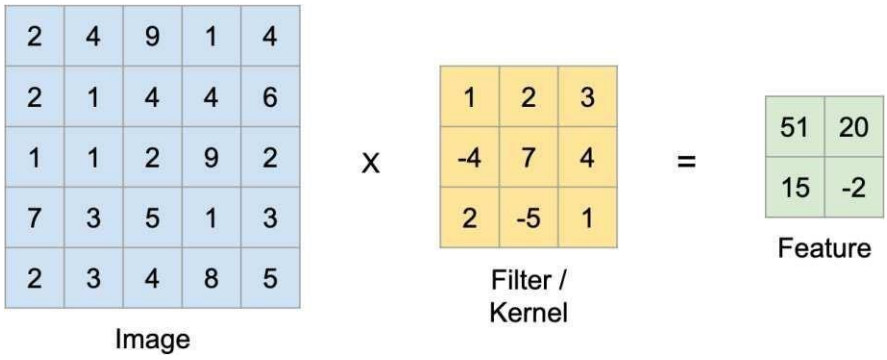


Fig 2.5 Convolution operation of kernel size 3 and stride 2

If you want the feature to be the same size as the input image that has to be analysed from the dataset, you can do this by padding the image.

Padding is a method to simply add zeros around the margin of the image to increase the dimension. Padding enables us to emphasize the border pixels of the data which has been sent to the model in order to lose less information.

Let's take an example with the input of an image of size 5x5 which has been padded to 7x7 , that is, the padding size of 1 and convoluted by a kernel size of 3x3 which has a stride of 1 resulting in a feature block size of 5x5.

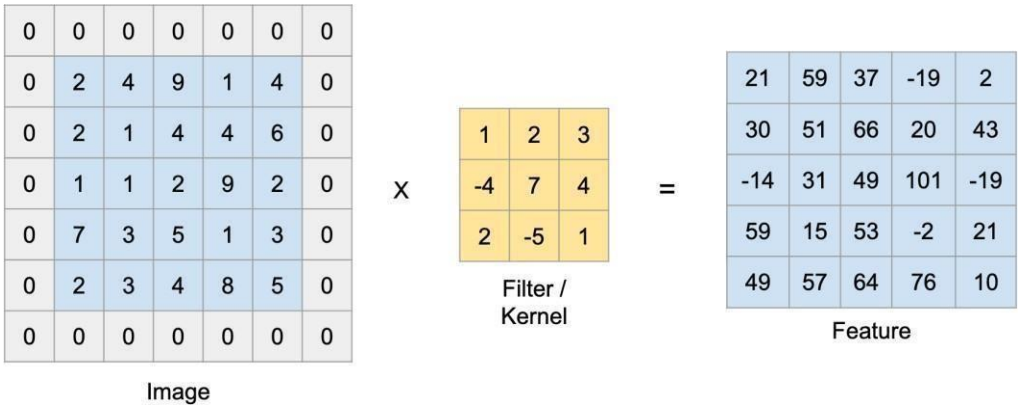


Fig 2.6 Padding process

The equation used for calculating the size of the feature in padding for this type of kernel size considering the padding image is shown below:

$$\textbf{Feature size} = ((\text{Image size} + 2 * \text{Padding size} - \text{Kernel size}) / \text{Stride}) + 1$$

From the above example, we shall take the values and calculate it.

$$\textbf{Feature size} = ((5 + 2 * 1 - 3) / 1) + 1 = 3$$

Now it needs to reduce the size of the features. And for that, take a look at the next part in the architecture called Max Pooling Layer.

Max Pooling Layer allows us to reduce the spatial size of the convolved features and enables us to reduce the over-fitting by providing the abstract representation. This is the sample based-discretization process. This is similar to the convolution layer but instead of taking the dot product between the input and the kernel, which will take the maximum part of the region from the input data which is overlapped by the kernel.

Shown below is an example of the Max pool operation with the kernel size of 2 and stride size of 1. This will show the steps that are performed during n this operation that takes place from the initial Max pooling layer to the final Max pooling layer.

Step 1:

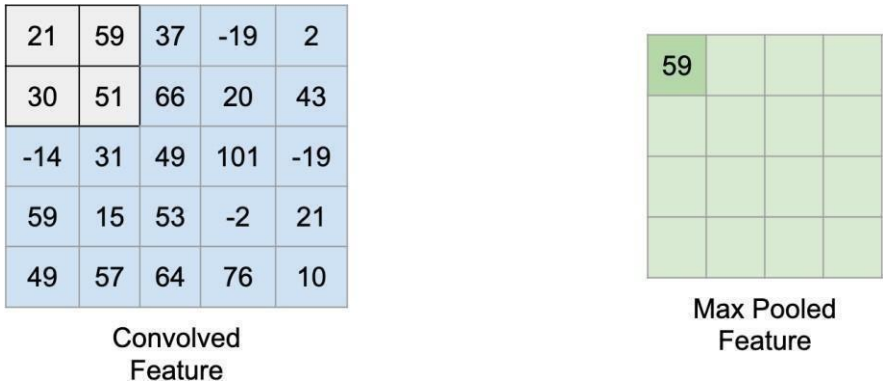


Fig 2.7(a) Step 1 of Max Pool operation

Step 2:

Fig 2.7(b) Step 2 of Max Pool operation

And this process steps continue till it fills every cells in the Max Pooled feature block.

Final Step:

21	59	37	-19	2
30	51	66	20	43
-14	31	49	101	-19
59	15	53	-2	21
49	57	64	76	10

Convolved Feature

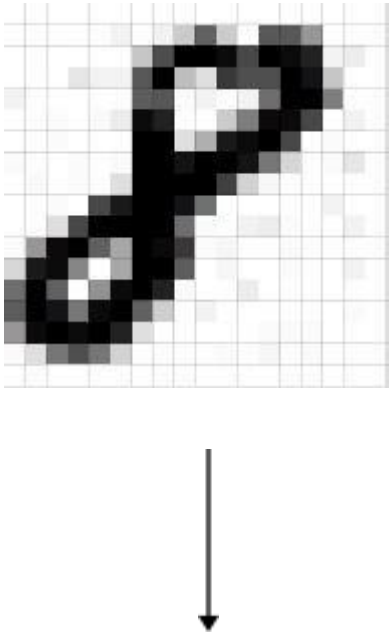
59	66	66	43
51	66	101	101
59	53	101	101
59	64	76	76

Max Pooled Feature

Fig 2.7(c) Final Step of Max Pool operation

There is another kind of pooling called **average pooling** which takes only the average values instead of the maximum value in the part of the dataset or in each stride. But Max pooling is preferred as it plays a very important role in reduction of noise by discarding the irrelevant data and makes it better than average pooling.

An image basically consists of a matrix of pixels. In CNN, the layer in the process flatten the image (as you can see in the image it converts a 3x3 matrix into a 9x1 vector). So essentially in a data set, every image can be represented as a matrix of pixel values.



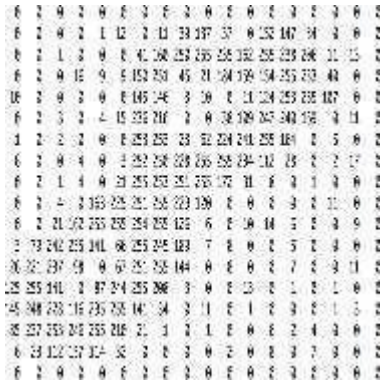


Fig 2.8 Pixelated values to represent a digit number 8

Therefore, by analyzing these values, the CNN does the operation through the network layers and gives the outcome.

3. Algorithm

Step 1: Start

Step 2: Import the libraries required to perform the operations for the model

Step 3: Import the MNIST dataset that can be used for the analysis

Step 4: Pre-process the data

Step 5: Develop the model with CNN

- Use filters to start the convolution process

-Feature size = ((Image size + 2 * Padding size − Kernel size) / Stride) +1

-Perform Padding operation

- Perform Max Pooling operation by taking the maximum value from each stride

- Set the values into the feature block

- Flatten the matrix into a one-dimension vector

Step 6: Set the epochs to 20. This is used for training the model 20 times to reduce the error rate

Step 7: Train the model

Step 8: Draw the number to be recognized

Step 9: Analyze and display the number that has been recognized and display the recognition (prediction) rate.

Step 10: Stop

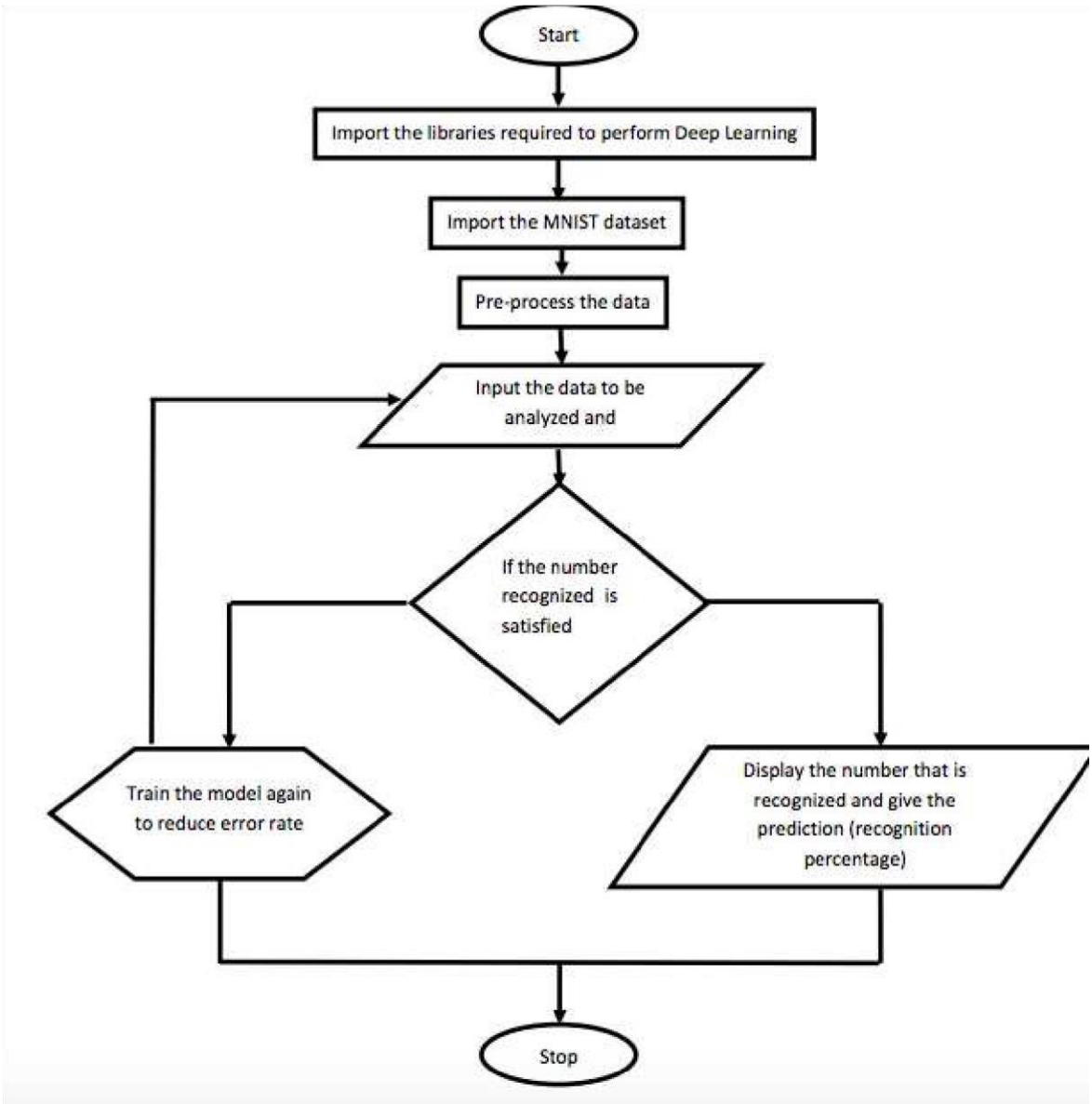


Fig 3: Flowchart

4.Result and Discussion

Before the input is fed into the model for processing, analyze and get the output, the model needs to be trained first. In the program, the epochs have to be set to 20. By setting this, the model will be trained for 20 times. It can be changed to how many times you want the model to be trained depending on the error rate of the prediction. After executing and training the model, the GUI program will be executed to allow users to input the number (draw).

Fig 4.1(a) Recognizing the number 1

In figure 4.1(a), the result shows it recognizes the number 1 with prediction rate of 99%

Fig 4.1(b) Recognizing the number 6

In figure 4.1(b), the result shows it recognizes the number 6 with a prediction rate of 77%.

Fig 4.1(c) Recognizing the number 5

In figure 4.1(c), the result shows it recognizes the number 5 with a prediction rate of 99%.

Fig 4.1(d) Recognizing the number 2

As you can see from figure 4.1(d), it shows the number recognized as 2 with the prediction rate of 99%.

Now the model is capable of recognizing the numbers after the analysis. However, the recognition rate (prediction rate) varies with the type of handwriting style. The neural layers work on the recognition of the Handwritten digits. In the background process, the Handwritten Digit Recognizer neural network is going through a series of various layers in the CNN process.

When the number has been fed to the model and on clicking the recognize button, the input is sent to the model. The input goes through a series of layers. These layers of the neural network perform the convolution and max pooling process. After this, the outcome is transferred to the GUI program and shows the digit that is recognized.

5.CONCLUSION

In this project, the HandWritten Digit Recognizer has been implemented and is able to recognize the number digits of different handwriting flavors. The CNN is one of the most widely used machine learning algorithms which has been trained and tested on the given dataset in order to compare and analyze. With this deep learning technique, a high amount of accuracy can be obtained. With Keras as the backend and Tensorflow as the software which is used for the machine learning analysis, this model is able to give a proper accuracy. With the increase of the data being generated every day, machine learning can play a vital role in our daily lives in the future. This project

shows the capability of deep learning and the application that can be used in various ways to analyze such patterns in the images, diseases etc. This concludes that the Handwritten Digit Recognizer is capable of recognizing any digit and also Deep Learning can make an impact in the coming future.

Declaration:

I declare that the work is an original report of our research project , has been written by us and has not been submitted for any others. The work is almost entirely our own work;the collaborative contributions have been indicated clearly and acknowledged.

*Funding - Not funded

*Conflicts of interest- It’s a project of one of our other where all of 3 work)

*Ethics approval - All have contributed almost equal from their sides

*Consent to participate - As per our decision we concluded to publish in journal *Consent for publication - In this research we have clearly stated each and every thing related to research with data and images

*Availability of data and material - All the output images and the code of our project are available with all of our members

*Code availability -

a)Software system configuration

1.Operating system: Windows 10

2.Programming language: Python 3.6

3.Compiler: idlec.exe

4.IDE: IDLE

b) Component requirements

1.Tensorflow

2.Keras

3.Pillow

4.Numpy

5.Win32gui

6.Tkinter

*Authors' contributions - I(Dilshan Kumar) have mainly focused on MNIST using linear regression algorithm , whereas mehnaz Ameer helps in the deburnging the code as she is expert in python and in last Alisha giri helped us ML in getting better result with less error.

6.REFERENCES

1. Arica, N.. One-dimensional representation of 2% two-dimensional information for HMMbased handwriting recognition, Pattern Recognition Letters, 200006
2. Butch Quinto. Next-Generation Machine 1% Learning with Spark, Springer Science andBusiness Media LLC, 2020
3. Practical Machine Learning and Image Processing - Springer, Himanshu Singh
4. Using Machine Learning", Springer Science and Business Media LLC, 2019 5. https://towardsdatascience.com/convolution-neural-networks-a-beginnersguideimplementing-a-mnist-hand-written-digit-8aa60330d022
6. https://towardsdatascience.com/a-comprehensive-guide-to-convolutionalneuralnetworks-the-eli5-way-3bd2b1164a53
7. Bishop, C. M. (2006), Pattern Recognition and Machine Learning, Springer
8. https://towardsdatascience.com/workflow-of-a-machine-learningprojectec1dba419b94
9. M. Wu and Z. Zhang, Handwritten Digit Classification using the MNIST Dataset,2010.
10. https://in.mathworks.com/solutions/deep-learning/convolutional-neuralnetwork.html
11. International Journal of Innovative Research in Computer and Communication Engineering – Handwritten Digit Recognition using CNN
12. https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/