PROJECT REPORT

on

HANDWRITTEN NUMERICAL RECOGNITION

Submitted in partial fulfillment of the requirements

for the award of degree

MASTER OF COMPUTER APPLICATIONS

of

KLE TECHNOLOGICAL UNIVERSITY's

Dr. M. S. Sheshgiri College of Engineering and Technology

by

Ms. Namrata Kamu

(SRN: 02FE21MCA028)

KLE TECH.

DEPARTMENT OF COMPUTER APPLICATIONS

KLE TECHNOLOGICAL UNIVERSITY's

Dr. M. S. Sheshgiri College of Engineering and Technology

Belagavi

2022-2023

A Project Report

On

Handwritten Numerical Recognition

Submitted in partial fulfillment of the requirements
for the award of degree

MASTER OF COMPUTER APPLICATIONS

of

KLE TECHNOLOGICAL UNIVERSITY
Dr. M.S. Sheshgiri College of Engineering and Technology

by

Ms. Namrata Kamu
SRN: 02FE21MCA028

Under the guidance of
Dr. Anusha R

DEPARTMENT OF COMPUTER APPLICATIONS

KLE TECHNOLOGICAL UNIVERSITY's

Dr. M.S. Sheshgiri College of Engineering and Technology

Belagavi

2022- 2023

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

KLE TECHNOLOGICAL UNIVERSITY's

Dr. M. S. Sheshgiri College of Engineering & Technology, Belagavi.

CERTIFICATE

This is to certify that the project work entitled "**Handwritten Numerical Recognition**" Submitted in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications of KLE Technological University Dr. M. S. Sheshgiri College of Engineering and Technology Belagavi Karnataka is a result of the bonafide work carried out by **Ms. Namrata Kamu, SRN:02FE21MCA028** during the academic year of 2022-2023.

Guide	HOD	Principal
Dr. Anusha R	*Dr. V. S. Malemath*	*Dr. S. F. Patil*

Viva-Voce Examination

Name of the Examiners	Signature with Date
1. ____________________	____________________
2. ____________________	____________________

ACKNOWLEDGEMENT

Every successful completion of any undertaking would be complete only after we remember and thank the almighty, the parents, the teachers, and the personalities, who directly or indirectly helped and guided during the execution of that work. The success of this work is equally attributed to all well-wishers who have encouraged and guided throughout the execution.

I express my deepest gratitude **Dr. Anusha R,** for their guidance and assistance throughout the project with great interest.

I avail this opportunity to express deepest gratitude to **Dr. V. S. Malemath, Professor & HOD** for encouraging us with his valuable guidelines.

I am grateful to **Dr. S.F. Patil,** Principal for his blessings. And we are grateful to **KLE Technological University, Dr. M.S. Sheshgiri College of Engineering and Technology, Belagavi** which has given us a bright future. I would like to thank all the people for their guidance and valuable suggestions throughout the project.

I also thank all teaching and non-teaching staff members of the MCA department for their invaluable cooperation.

Finally, I would like to express our sincere thanks to our **Parents and Friends** for their enormous encouragement and all other who have extended their helping hands towards completion of our project.

Ms. Namrata Kamu

ABSTRACT

One of the very crucial problems in applications of pattern recognition is handwritten character recognition. The main purpose of this research is to design a mechanism for automatically recognizing handwritten digits, with the end goal of correctly identifying sequences of digits that are handwritten. Initially, the numbers will be divided up into separate digits as part of the process to accomplish the recognition assignment. When the work of handwritten numeric string recognition is finished, a numeric recognition module is involves classifying each segmented number. Numerous tasks, such as form data entry, bank check processing, and postal mail sorting, include digit recognition. Establishing the proficiency to develop algorithm that can read handwritten digits given by users via a scanner, tablet, and other digital devices is what the challenge is really all about.

INDEX

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project Overview:

Artificial intelligence and computer technology both heavily rely on machine learning and deep learning. Human effort in identifying, learning, making predictions, and many other areas can be decreased with the application of deep learning and machine learning. This article compares KNN, PSVM, ANN, and Convolution Neural Network classifiers on the basis of performance, accuracy, time, sensitivity, positive productivity, and specificity with the use of various parameters with the classifiers. The handwritten digits (0 to 9) from the well-known MNIST dataset are presented for recognition.

Deep learning and there is an ongoing implementation of machine learning approaches explored by developers to make machines smarter. A human learns to do something by doing it repeatedly until it has learned how to do it by memory. His brain's neurons then spontaneously fire, enabling him to carry out the acquired task swiftly. Also extremely similar to this is deep learning. In order to solve various kinds of issues, it employs various neural network topologies. Computers can now understand human-written digits thanks to handwritten digit recognition technology. Handwritten digits can be generated in a multitude of shapes and patterns, making it challenging for the machine to accomplish this work accurately. This issue can be solved by using a visual representation of digit to recognize the number that is present in the image, which is done through handwritten digit recognition.

In a several real-world scenarios, such as online handwriting recognition on computers, tablets, or systems, or recognizing number plates of numeric entries in forms filled out by hand, numerical recognition systems work by teaching machines to recognize the digits from various sources, such as emails, bank checks, papers, images, etc.

1.2 Digit Recognition System:

The operation of a machine to train itself or recognize the numbers from various sources, such as emails, bank checks, papers, images, etc. and in various real-world scenarios, such as online handwriting recognition on computer tablets or systems, recognize number plates of, recognize numeric entries in forms filled out by hand, and the list goes on.

The primary aim of this project is to develop a model that will come into existence to capable of identifying and calculating the handwritten numbers from their image using the concepts underlying of CNNs. While the aim is to develop a model with the ability to identify digits, it also holds applicability for recognize letters and a person's handwriting. Understanding Convolutional Neural Network and utilizing within that proposed system's handwritten recognition method is its main objective.

1.3 Problem Statement:

The purpose of this research is to develop a model utilizing CNN principles that can identify and ascertain handwritten numbers from an image. The objective is to develop a model that possesses the capacity to detect digits, but it can also be extended to recognize characters and a person's handwriting. Understanding CNN and utilizing it in the proposed system's handwritten digit recognition system is its main objective.

1.4 Objective of the Project:

Our work focuses on creating pre-processing techniques that are optimal in addition to designing and implementing handwritten number recognition algorithms. Numbers are structurally analyzed to effectively pinpoint characters. Contrary to word recognition, character recognition can be enhanced by comparing the length of the word and the identified character with a dictionary. However, such association is not conceivable in number recognition. As a result, we have created very accurate hand written number recognition.

2. FEASIBILITY STUDY

2.1 Technical feasibility:

One can connect to the software utilized in this project whenever they want because it is entirely open source. The study notion of image processing and machine learning can be described as highly popular subject right now. As opposed to side of the python and open CV concepts. In addition, Google Colab's entire operating system is open source and simple to access when there is an internet connection. The user, who need not be a coder, may still set the digit in the camera screen and view the output by hitting the run button.

2.2 Economic Feasibility:

This project exclusively utilized open source software for its development monetarily free; hence, no fees or donations were incurred. The exclusively focus on study materials that are in the form of guides not available for free are those for designers or developers. Completely free of charge is the software or program.

2.3 Seasonal Feasibility:

This project can be started on a specific day and completed entirely within the allotted time, which is known as being time-feasible. The project was successfully completed on schedule thanks to a successful effort.

2.4 Profitability:

Two hot themes are covered in this project processing images with machine learning. We didn't apply the full machine learning idea here, but we did deal with the neural network, which is the machine learning's fundamental building component. These two subjects are excellent for research, and numerous academics and instructors are working with them daily to improve existing approaches or algorithms or develop new ones. The project's extension may be used extensively to quickly and accurately identify written characters in photos and extract them. Or in real-time picture chaining, filtering, object detection, license plate verification, banking signature recognition, etc. Due to the many aspects of this undertaking, it is completely profitable.

3. LITERATURE REVIEW:

Handwritten digit recognition has been a prominent and challenging problem in the fields of pattern recognition, machine learning, and computer vision. The advent of artificial intelligence and deep learning has revolutionized the landscape of character recognition, allowing for significant advancements in accuracy and performance.

Abu Ghosh and Maghari (2017) conducted a comparative investigation of several neural techniques, including K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Convolutional Neural Networks (CNN), focusing on performance metrics such as accuracy, time efficiency, sensitivity, positive productivity, and specificity. Their study demonstrated the superior performance of CNNs in handwritten digit recognition tasks, attributed to CNN's ability to capture intricate spatial patterns and hierarchical features within images. This finding resonates with the growing consensus in the literature that deep learning, especially CNNs, has revolutionized character recognition tasks due to their data-driven feature learning capabilities (LeCun et al., 1998).

Liu et al. (2017) extended this comparative analysis by applying multiple classifiers and feature vectors to recognize handwritten digits from well-known databases including CENPARMI, CEDAR, and MNIST. Their exploration reinforced the potential of deep learning techniques, revealing that CNNs achieved remarkable recognition accuracies across these diverse datasets. The study underlined the significance of robust feature extraction and classifier selection for optimal performance.

The proliferation of neural networks, particularly CNNs, in handwritten digit recognition is indicative of their capacity to automatically learn and extract relevant features from raw data, eliminating the need for manual feature engineering. This aligns with the findings of Simard et al. (2003), who introduced the concept of using deep architectures for character recognition tasks, demonstrating their superiority over traditional shallow networks. Real-world applications of handwritten digit recognition systems are widespread. They find use in online handwriting recognition on devices ranging from tablets to computers, contributing to efficient data input and text entry. These systems also play a pivotal role in sectors like banking, where they aid in check processing, and in postal services for mail sorting (Plamondon & Srihari, 2000).

4. ANALYSIS

4.1 Existing System:

An rising number of people are using images to transfer info these days. It is also common practice to isolate important information from images. For its many uses, image recognition is a vital study topic. The exact computer recognition of human handwriting is one out of the group challenging tasks in the realm of pattern recognition in general. The absence is evident doubt that the following is a highly challenging subject given the wide variety in handwriting that exists amongst individuals. Despite the absence of this differentiation from affect individuals in any way, teaching computers to understand standard handwriting has increased steadily challenging.

It is crucial to understand how information is represented on photographs to accomplish the task or solve problems with image recognition, such as handwritten classification.

4.2 Proposed System:

The proposed model contains four stages to classify and detect the digits:

A. Pre-Processing: Pre-processing is a portion of HDR. It's going to much simpler to find the limits if there certain limitations apply, like the following a box for each digit. Pre-processing's primary goals are to eliminate noise filtering, smoothing, and uniformity. A grayscale image is binaryized into a binary image.

B. Feature Extraction: The error rates of various feature extraction algorithms vary. Combining all of these methods results in a flawless identification rate, helps to reject the recognition of ambiguous digits, and increases the recognition rate of incorrectly classified digits that can be identified by humans. The errors caused by each individual algorithm do not overlap.

C. Classification and Recognition: The recovered feature vectors are provided as single input values to each classifier during the classification and recognition process. CNN Subsampling and convolution layers each have multiple layers that can be used. A different name for the down sampling layer is the pooling layer. To calculate a value for each area, the image is broken into tiny segments of tiny areas. A must be made in order to new image, the calculated values are then rearranged in order. The fuzzy filter procedure, which can strengthen the robustness of image feature withdrawal, is comparable to this method the combining of extracted features.

D. Training and Testing: A model can undergo training using the fit() technique. The usage of test data validation dataset to evaluate the trained model's performance. The test dataset is then utilized to assess a model. Because each module is created to address a certain sub issue, training is less complicated. Because each module is taught independently and it is simple to add or delete modules, it is believed that each module can handle the particular problem more effectively and precisely.

4.3 Functional Requirements:

A mathematical computation is made, handwritten digits are accurately detected and classified, a real-time prediction is made, and the results are displayed.

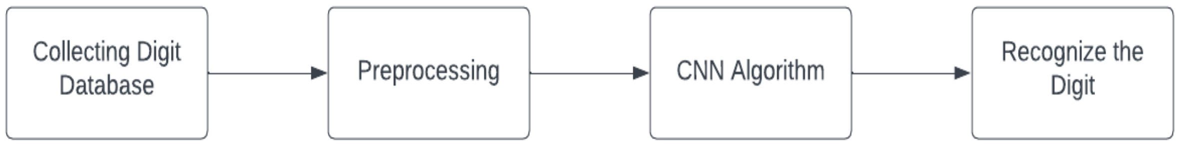
4.4 Non-Functional Requirements:

Performance, dependability, platform independence, accuracy, scalability, response speed, and maintainability are all important.

5. METHODOLOGY:

In this section, we'll cover a variety of innovations and acknowledgements, including general algorithms, techniques, datasets used, and the creation and testing of models.

The portrays the Block Diagram of the proposed system model.



5.1 Basic steps in constructing a Machine Learning model:

5.1.1. Data Collection: Your data's quantity and quality assess the precision of our model's performance is using pre-collected data, such as datasets from Kaggle, UCI, etc., still fits into this stage. • The outcome of this step is typically a representation of data (Guo simplifies to supplying a table).

5.1.2. Data Preparation: Handle data with care then prepare it for instruction. Check to make sure that everything that need cleaning is done (Eliminate duplicate entries, correct typographical errors, and handle missing numbers, normalize, convert data types, etc.). Use data visualization to explore data and search for important relationships between variables or class imbalances (bias alert!). Into sets for training and evaluating.

5.1.3. Choose a Model: Select the appropriate algorithm for the task at hand.

5.1.4. Train the Model: Making accurate predictions or responses to questions as frequently as feasible is the aim of training. Each iteration of the process is a training phase. For example, in a linear regression, the algorithm would need to learn values for m (or W) and b (x is the input, y is the result).

5.1.5. Evaluate the Model: Uses a metric or set of metric to "measure" the model's objective performance. Evaluate the model employing historical data never been seen before. This undetected information is intended to be somewhat indicative of model effectiveness in the real world, but it still aids in model tuning (unlike test data, which does not).

5.1.6. Parameter Tuning: This stage deals with the "art form" the term "hyper parameter tuning" not a science. Adjust model parameters for better efficiency. Examples of straightforward model hyper-parameters include initialization values and distribution, learning rate, and the number of training iterations.

5.1.7. Make Predictions: Additional (test set) previously published data is hidden from the model (and what classes are the labels used known) are utilized to test the model, providing a more precise representation determines the behavior of the model in practice.

5.2 LOADING THE DATA SET:

5.2.1 MNIST Data Set:

Changed a sizable collection of computer vision data derived from Modified National Institute of Standards and Technology (MNIST) is widely used for developing and evaluating various systems. It was produced using two unique datasets held in binary form by the National Institute of Standards and Technology (NIST) and representing handwritten digits. 250 persons contributed handwritten numbers to the training set, of which 50% were Census Bureau employees and the remaining 40% were high school students. However, it is sometimes it asserted that this dataset was the first among others to demonstrate the viability of neural networks.

Figure 1: MNIST Data Set

10,000 photos used throughout testing, but the database comprises 60,000 photographs utilized for training and a small number of cross validation reasons. The intensity is centred in the image with 28 X 28 pixels and is represented by all the digits being grayscale and fixed in size. Since each image is 28 by 28 pixels, the array of images can be flattened into a 28 by 28 dimensional vector with a 784-dimensional array. The binary values that make up each element of the vector each represent the pixel's intensity.

5.3 Pre-Processing:

5.3.1. Data quality assessment: The source, volume, and impact of any data items that violate established data quality guidelines are verified using a Data Quality Assessment, a separate part of the data quality life-cycle. It can be carried out once only or repeatedly as a part of an ongoing project in order to assure the data's quality. Even with strict data capture procedures that clean the data as it enters your database, the quality of your data can drastically deteriorate over time. The information you possess can soon become outdated as a result of people relocating, changing phone numbers, and dying. A data quality assessment aids in identifying records that have grown erroneous, the potential effects that inaccuracy may have had, and the source of the data. It can be fixed and additional possible problems found thanks to this assessment.

5.3.2. Data cleaning: One of the crucial components of machine learning is data cleaning. It is crucial to the process of creating a model. There are no hidden twists or secrets to discover, but it's also not the most fancy aspect of machine learning. But data cleaning done correctly can make or ruin your project. This stage typically takes up a significant percentage of the work of professional data scientists. Because the saying "Better data beats fancier algorithms" is widely held. Even a very simple method can get the desired results given a clean dataset, which can occasionally be quite useful. It's obvious that numerous things imply data kinds will require various methods of cleansing. But this methodical approach is always an excellent place to start.

5.3.3. Data transformation: In actuality, data manipulation has previously been carried out by cleaning and smoothing. However, when we talk about data transformation, we mean the processes that are used to transform the data into a format that the computer can use to learn from. Data transformation is the process by which data is changed from its unjoined, compartmentalized, and normalized source state into data that is dimensionally modelled, de-normalized, and prepared for analysis.

Data transformation can be laborious, costly, and time-consuming without the appropriate technological stack in place. However, data transformation will guarantee the highest data quality, which is essential for accurate analysis and the subsequent discovery of insightful information that will ultimately enable data-driven decisions. A wonderful idea, the adoption of machine learning by more businesses or is being planned for deployment to handle many different practical applications. But in order for models to learn from data and make useful predictions, the data itself must be arranged so that its analysis produces insightful results.

5.3.4. Data reduction: This represents a procedure that takes the original data's volume and it is shortened to a much lower volume. Its strategies protect data integrity while minimizing the size of the data. Data mining on the smaller data the duration of the set must not exceed the time needed for data reduction.

5.3.5. Data Encoding: As our loss function is the cross-categorical entropy, this step is optional. The given labels must be categorical in nature, and this must be specified to the network. It could be necessary to process the raw data to convert it into a format suitable for utilization by machine learning models. The unstructured and structured data types can both be present in the raw data. We would run into issues if we left category variables in their current state because machine learning is based on mathematical formulae. Many algorithms naturally support categorical values, although in those circumstances, whether or not to encrypt the variables is up for debate. The next stage is to process the information within a way that is appropriate to provide input for Machine Learning models after identifying the data kinds and features that are present in the data set.

Two different options exist to execute the three widely used conversion procedures from categorical to numeric values.

1. Label Encoding.

2. One Hot Encoding.

3. Binary Encoding.

The difference in how certain items within a category are encoded is known as encoding variability. When discussing the variability of a single hot encoding, it should be emphasized that the variability is dependent upon the timing of execution, which determines the appropriate number of categories to use to sufficiently affect the target. Other encoding techniques do exhibit sizable variability, which is seen during validation.

5.4 MODEL CONSTRUCTION:

5.4.1 MODELS CAPABLE OF BEING USED IN THE PROJECT:

1. CONVOLUTION NEURAL NETWORK: CNN is a synthetic neural network that specializes in identifying and deciphering patterns, to put it another way. CNN has so proven to be the most effective at classifying images. The types, sizes, and quantities of filters in a CNN model vary. In essence, it is these filters that enable us to recognize the pattern. Convolutional neural networks, sometimes known as CNNs, are a particular kind of NN model made for use with two-dimensional picture data, although it has the potential to be applied to one- and three-dimensional data.

The convolutional layer, which is what gives the convolutional neural network its name, is at its core. A process known as "convolution" is carried out by this layer. Convolutional and pooling layers typically make up a CNN model. CNN performs effectively for challenges involving picture classification because it performs better information that is depicted as grid structures. Some neurons are deactivated using the dropout layer, which also decreases the model's offer fitting during training. Convolutional feature extraction and binary classification make up our model. To extract the features from the image, convolution and max pooling are used. A 28x28 image is first processed using 32 3x3 convolution filters, then a 2x2 max-pooling layer, and finally a final 64 3x3 convolution layer.

CNN

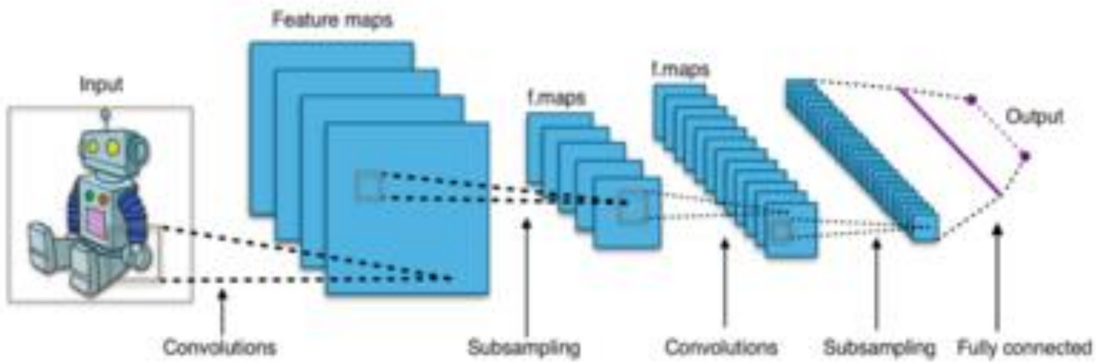


Figure 2: CNN Architecture

By flattening the final photos, we get 7x7 images. The flatten layer will convert the 7x7 images into a set of 128 values that will be mapped to a dense layer of 128 neurons that is connected to the category output layer of 10 neurons. A patch of input data the size of the filter is multiplied by the filter, and the result is a dot produc1t because the filter is smaller than the input data. A dot product is created by multiplying the input and filter's filter-sized patch element-by-element. This result is then added together, always producing a single value.

The operation is frequently called the "scalar product" because it yields a single value. It is deliberate to use a filter that is smaller than the input because it enables the same filter (set of weights) to be multiplied by the input array numerous times at various locations on the input. To be more precise, the filter is consistently applied from top to bottom, left to right, to each overlapping portion or filter-sized patch of the incoming data. A single value is produced when the filter is multiplied by the input array just once. A two-dimensional array of output values representing a filtering of the input is produced by applying the filter to the input array numerous times. This operation's two-dimensional output array is referred to as a "feature map" as a result.

WORKING OF CNN:

The clusters of synthetic neurons that make up CNN are numerous. A set of synthetic neurons crude emulation of its mathematical functions that are biological counterparts and provide compute the rounded total of numerous a value for activation is input and output. Each neuron's weights determine its behavior. A CNN's artificial neurons can recognize a variety of visual cues when given with pixel values. Each of a ConvNet's layers produces a number of activation maps when you feed it a picture. In-depth areas of the image are highlighted by activation maps. Each neuron receives a patch of pixels as input and multiplies the colour values by its weights before adding them all up and running the input through the activation function.

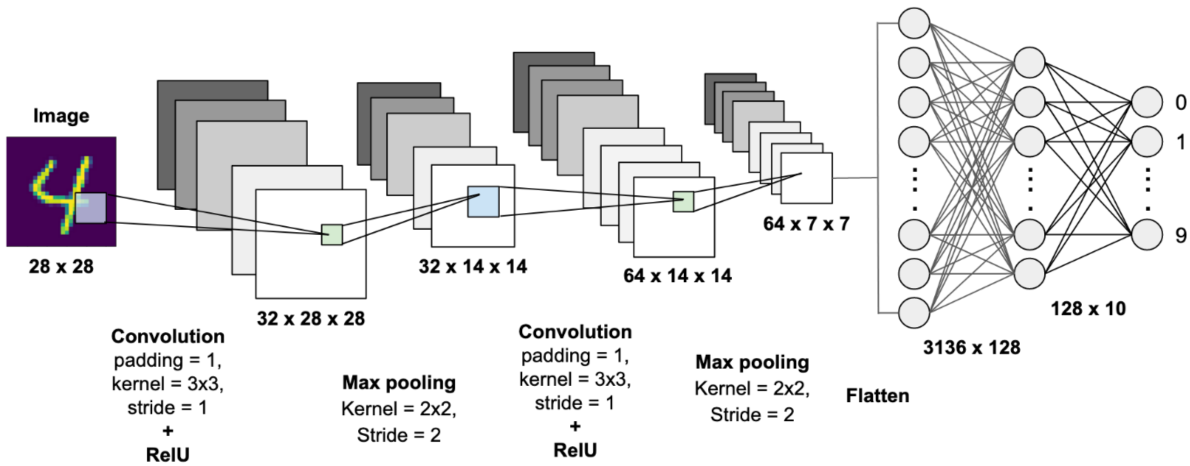


Figure 3: CNN for handwritten numerical recognition

Basic features like edges that run vertically, horizontally, and diagonally are often detected by the first (or bottom) layer of the CNN. The second layer extracts more complicated features, such as corners and combinations of edges, using the produced by the first layer as its input. CNN have layers that recognize higher-level elements like objects, faces, and other features as you go deeper into the network. Convolution is a method for multiplying pixel values by weights and adding them; hence, convolutional neural network. Several convolutional layers typically make up a CNN, but it also has additional parts.

Our model's layers are added using the 'add()' function. We start with two Conv2D layers. These layers of convolution will work with our input photos, which are represented as 2-dimensional matrices. Each layer has a different number of nodes, with the top layer having 64 and the second layer having 32. If the dataset is large enough, one can alter this amount to be higher or lower. We will continue using this for the time being because 64 and 32 work nicely in our situation. The filter matrix for our convolution is how big the kernel is. Consequently, a 3x3 filter matrix will be used if the kernel size is 3. For a review of this, think back to the introduction and the first image. The layer's activation function is called activation.

ReLU, or Rectified Linear Activation, is the activation function we'll be employing for our first two layers. There is evidence that neural networks successfully use this activation function. Also accepting an input shape is our initial layer. As was previously mentioned, each input image has the following shape: 28,28,1, where the number 1 denotes that the photos are in grayscale. There is a "Flatten" layer sandwiched between the dense layer and the Conv2D layers. A link between dense layers and convolution is provided by flatten. For our output layer, we will utilize the layer type "Dense".

A common layer type for neural networks is dense. In our output layer, there will be 10 nodes total, one for each potential result (0–9). "Softmax" activation is used. Softmax increases the output's sum to 1 so that it can be read as probabilities. The choice has the greatest likelihood will then be used given by the algorithm to produce its forecast.

5.5 Training & Validation:

To facilitate the training of the model with the available data set, it must be assembled after creation. The model is put together using optimizers. The optimizer, loss, and metrics are the three inputs required for the model's compilation. Optimizers are programs or techniques that alter the neural network's parameters, such as its learning rate and weights, in order to minimize losses. Utilizing minimization techniques, optimizers can address optimization issues. The learner's rate is regulated by the optimizer.

Adam will serve as our optmizer in this case. In many situations, Adam is a reliable optimizer. The Adam Optimizer modifies tempo of learning as training progresses. How quickly the model's ideal weights are calculated depends based on learning rate. Up to a specific extent point, a reduced learning rate would lead to more accurate weights, but the computation of the weights would take longer. We'll use 'categorical_crossentropy' as our loss function. This option for categorisation is the most popular. Better model performance is indicated by a lower score.

To make things even simpler to understand, when we train the model, we will use the 'accuracy' statistic to see the accuracy score on the validation set. Getting the highest learning rate and the greatest amount of validation is the goal of testing and training any data model. By expanding the train and test data, respectively, one can improve both validation and learning rate. Once the model has been effectively put together, it can undergo training through the utilization of training data for 100 iterations, but as the quantity of iterations rises, the risk of over fitting grows.

As a result, we set a training accuracy criterion of 98%. Test data was utilized to evaluate the model since we are using real-world data for prediction.

6. CNN ARCHITECTURE:

It is common practice to analyze visual images using CNNs, a kind of DNN that can identify and categorize specific properties from images. Their uses include picture and video recognition, categorization of images, study of images used in medicine, computer vision, and natural language processing. CNN uses the term "convolution" to refer to a mathematical procedure known as convolution, which multiplies two functions to create a third capability that conveys the form of an individual entity function is altered by the alternative.

In plain English, two matrices that correspond to two images are multiplied to give an output that serves as extract information from the image. Technically, each input image is sent through a sequence of convolution layers with filters (Kernals), Pooling, fully connected layers (FC), and the application of the Softmax function to identify an object with probability values ranging from 0 to 1. The flow of CNN to process an input image and classify the objects based on values as depicted in the figure below.

6.1 Basic Architecture: A CNN architecture mostly consists of two components:

• Feature extraction is a procedure applied by a convolution tool to separate and identify the distinct characteristics of a picture for analysis.

• A layer that is fully connected and utilizes the convolutional process's output and determines the class of the image using the features that were previously extracted.

CNN Layers: The fact that these layers occur repeatedly indicates the depth of our network, and this structure is referred to as a deep neural network.

● **Input:** The input consists of raw pixel data.

● **Convolutional layer:** Transforming the neuron layer's findings are input layers. The chosen filter has to be specified. Each filter is only capable of 5 by 5 window that slides across the information provided and extracts pixels utilizing the finest intensities.

● **Rectified linear unit [ReLU] layer:** offered activation feature for image data that was captured. ReLU function is employed in the back propagation scenario to prevent changes in pixel values.

● **Pooling layer:** carries out a volume down-sampling procedure along the dimensions (width, height).

● **Fully connected layer:** A maximum score of the input digits is identified after focusing on the scoring class.

There is a significant rise in complexity as we dig further and deeper into the layers. However, it would be worthwhile to go since while accuracy might improve, time consumption does, unfortunately, too.

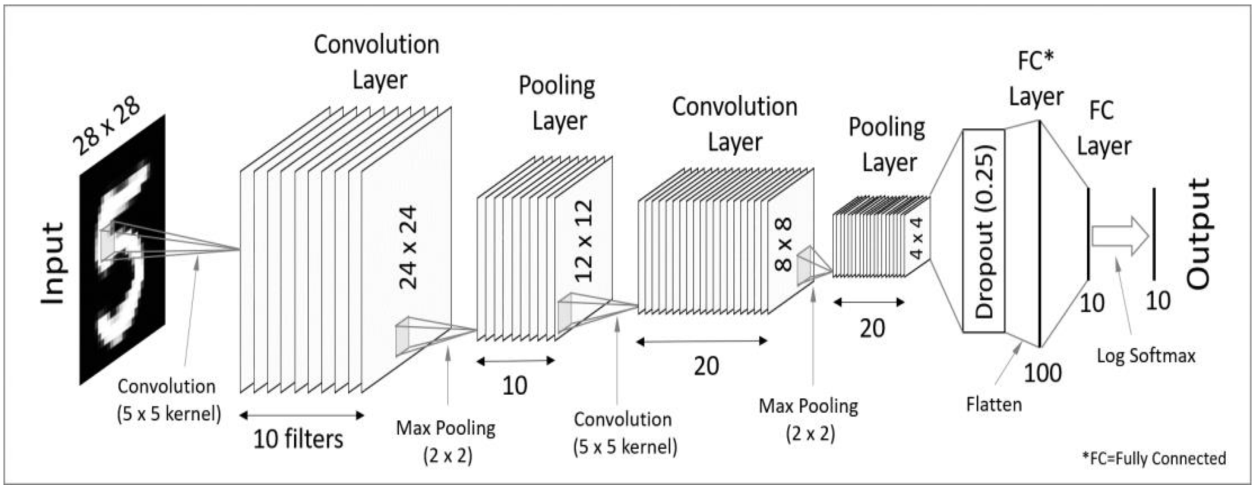


Figure 4: CNN Architecture for Handwritten Numerical Recognition

6.2 CNN Layers:-

1. Convolutional Layer: The various attributes found in the input photographs are extracted from the first layer, which is utilised. The image input alongside a filter with the specific size M x M are convolution mathematically in this layer. The scalar product of the values among filter and the portions of the image provided as input are captured by swiping the filter over the image, taking the account of the filter's size (M x M).

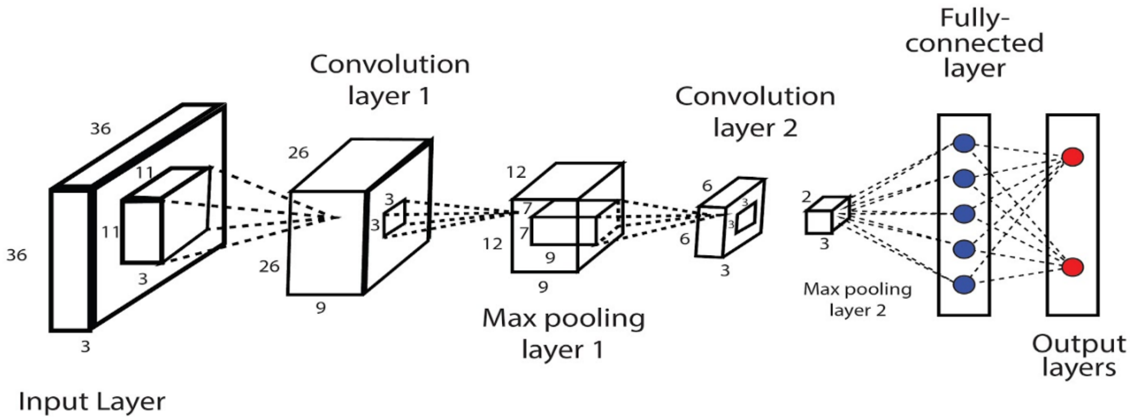


Figure 5: Convolutional Layer

The outcome is referred to as the Feature map, which imparts information about the image, encompassing its corners and edges. This particular feature map is subsequently fed into subsequent layers to impart supplementary features from the initial input image.

2. Pooling Layer: A Pooling Layer typically comes after a Convolutional Layer in most situations. This layer's main the aim is to scale back the convolved feature map to be able to save on computational expenses. As a result, each attribute map is individually operated upon while the links between layers are reduced. There are many Pooling operations, according to the approach utilized. The greatest component in Max Pooling is derived from a feature map. The elements inside a predetermined sized Image slice are averaged using the average pooling technique. Sum Pooling calculates the aggregate sum of the components in the predefined section. Between the Convolutional Layer and the FC Layer, the Pooling Layer generally serves as a connection..

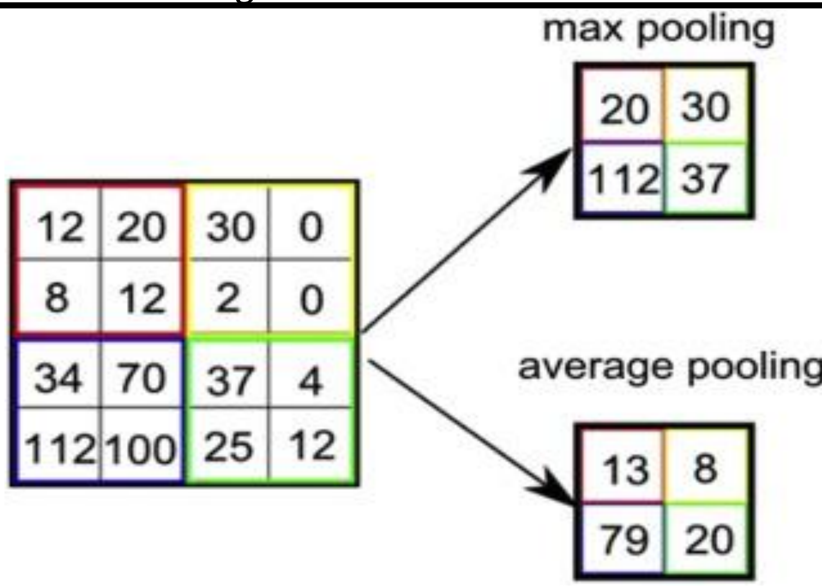


Figure 6: Pooling Layer

3. Fully Connected Layer: To connect the neurons between two separate layers, the Fully Connected (FC) layer, which includes the weights and biases as well as the neurons, is employed. These layers generate an alternative version of final few of CNN Architecture and are typically positioned before the output layer.

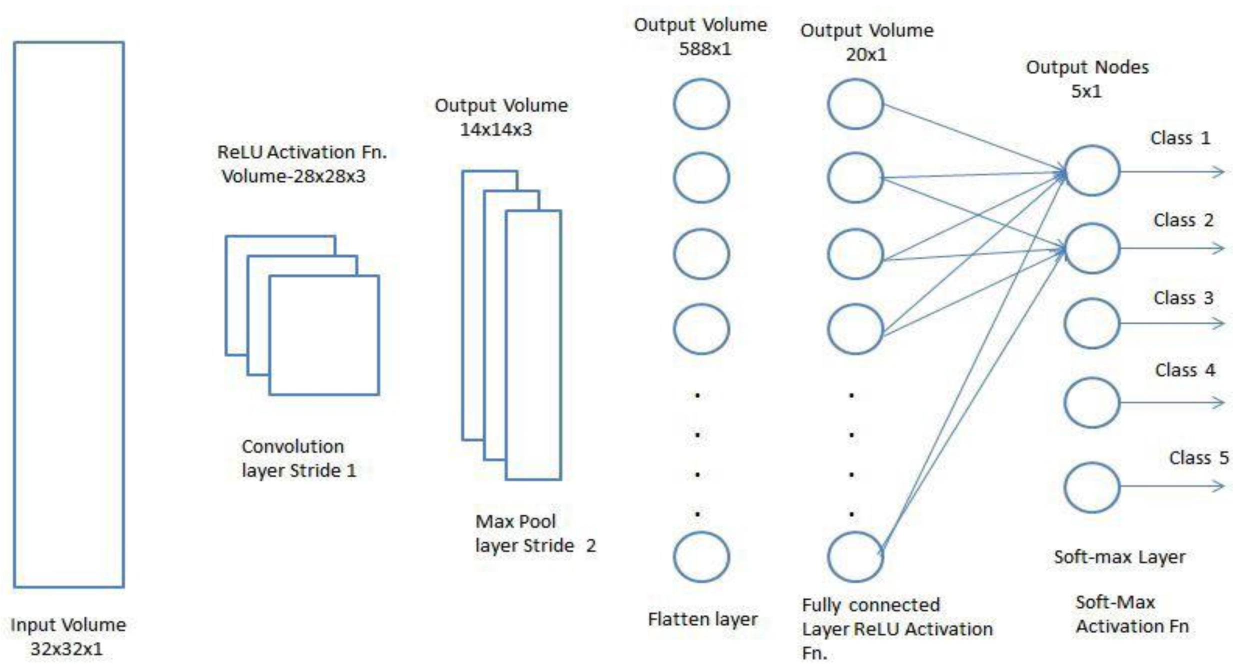


Figure 7: Fully Connected Layer

4. Dropout: Over fitting might occur in the training dataset when all the features are coupled to the FC layer. When a given model performs so well on training data that it performs poorly on new data, this is referred to as over fitting.

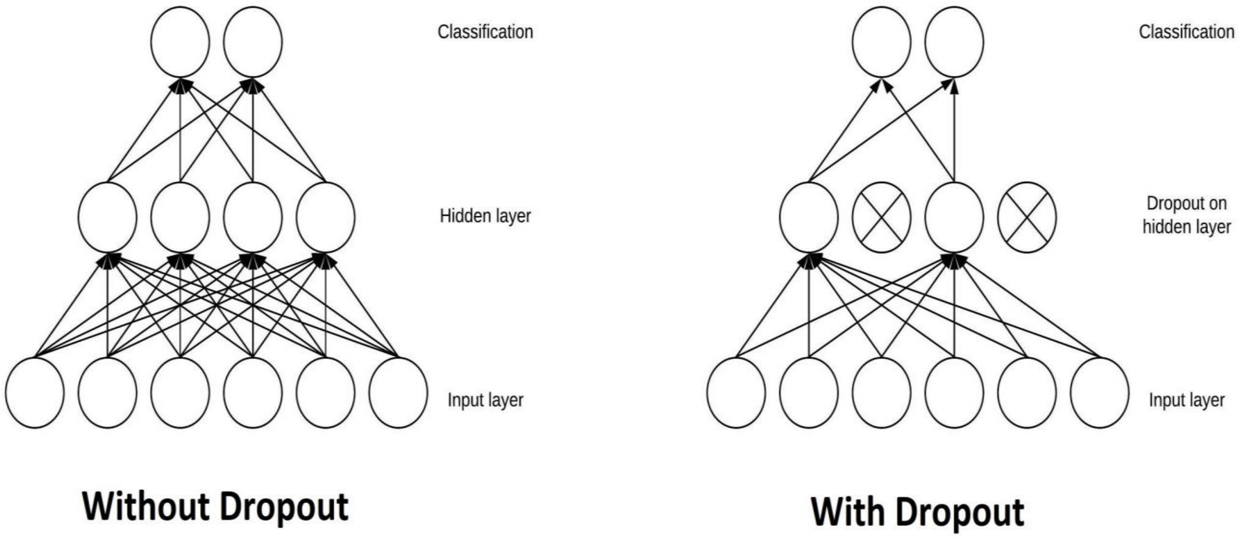


Figure 8: Dropout layer

A dropout layer is serves as solve this issue, which causes the dimensions of the model to be lowered by removing a small number of neurons from the neural network during training. 30% of the nodes in the neural network are randomly removed when a dropout of 0.3 is reached.

5. Activation Functions: In a neural network, the activation function specifies how the result of a node or nodes in a layer of the network is created from the averaged total of the input. In some cases, the activation function is commonly known as a "transfer function". The mechanism of activation may be commonly known as a "squashing function" if its output range is constrained. The term "nonlinearity" in the layer or network architecture may get used to describe the nonlinearity of several activation functions. The neural network's capabilities and performance are significantly influenced by the activation process that is selected, and multiple activation functions may be applied to different regions of the model.

Networks are built to utilizing a uniform activation function for all nodes in a layer, however technically, a function that activates is applied before or after each node's internal processing. Three different layer types are possible in a network: input layers that take unprocessed data from the domain, concealed layers that take input from a single layer and pass it along to another, the output layers that provide predictions. The activation function used by all buried levels is often the same. Depending on the kind of prediction needed by the model, the output layer will often employ a different activation function than the hidden layers.

Additionally, most activation functions are differentiable, allowing one to compute the first-order derivative for any given input value. Considering that NN are often trained using the back propagation of error algorithm, which needs the derivative regarding the forecast error to update the model's weights, this is necessary. Despite the fact that there are numerous variations kinds of activation functions utilized in neural networks, possibly only a few functions are really used in reality for the hidden and output layers.

Without a doubt, the activation function one among the most popular crucial variables in the CNN model. They are employed to discover and approximation any form of continuous and complex link between variables of the network. Simply said, it determines which model information should shoot forward and which information should not at the network's end. It gives the network more nonlinearity. Numerous activation functions, including the ReLU, Softmax, tanh, and Sigmoid functions, are frequently utilized.

These functions each have a particular use. Softmax is typically utilized for multi-class classification while sigmoid functions are forward for binary classification CNN models.

• Feature extraction is a procedure used by a convolution tool to separate and identify the distinct characteristics of a picture for analysis.

• A densely(fully) connected layer that utilizes the complete set of connections convolutional process's output and determines the class that picture using the features that were previously extracted.

7. System Specification:

7.1 Software Requirements:

- Operating system : Windows 10.
- Programming Languages : Python
- IDE : Jupyter Notebook.
- Browser : Google Chrome, Firefox etc.
- Code Editor : VSCode

7.2 Hardware Requirements:

- Processor : Intel CORE i3 and above
- Processor speed : 2.30 GHz
- RAM : Minimum 4 GB or more
- HDD : 100 GB or higher
- Keyboard : Standard 101 keys keyboard

7.3 TOOLS AND TECHNOLOGIES

- **OpenCV:** The Open Source Computer Vision Library is a python DIL library which is very updated in the work of picture taking. One image file and pixel values may be simply incorporated surface by this library. This library provides a common infrastructure and module related to computer vision technologies. The most crucial aspect of this is tool it is totally free and can be easily modified and changed respective to input by the programmer.

- **Numpy:** Numpy serves as a Python toolkit designed for array manipulation and handling. Its goal is to offer an array entity that exhibits a performance boost of up to 50 times compared to conventional Python lists.

- **Keras:** Keras is a freely available open-source Python library that is both potent and user-friendly, designed for the creation and assessment of deep learning models. Keras is designed to quickly define deep learning models. You can create and train neural network models using only a few lines of code.

- **TensorFlow:** TensorFlow, a swift numerical computing Python library, was developed and launched by Google. It serves as a fundamental framework for building Deep Learning models either directly or through additional libraries that streamline the process and are built upon the TensorFlow foundation.

- **Tkinter:** Tkinter serves as the primary GUI library for Python, enabling the swift and straightforward development of GUI applications when paired with the language. It furnishes a robust, object-oriented interface to the Tk GUI toolkit, offering substantial capabilities.

8. MODEL DETAILS:

8.1 Image Processing: At the early beginning of the project, an elaborate image processing technique was used**.**

Then what exactly is image processing? Image processing is a technique used to apply certain operations to an image in order to produce an improved image or to draw out some relevant information from it. This constitutes a type of signal processing where the input consists of an image, and the output may involve an illustration or one containing features or attributes associated with that picture. Today, among the technologies that is quickly developing is image processing. Within the engineers and computer scientist work in a variety of fields, it also serves as a core research area.

In this project, the idea of image processing utilised in the detection phase to take input straight from the webcam and to modify that image moreover, in the very first phase of training with real-world datasets and the MNIST dataset. The various image processing ideas will be discussed one by one in this project. Since this project is still in two stages, the influence of image processing will be covered in two sections.

1) Image processing in training data.

2) Image processing in testing data.

1. Image processing in Training Data

The neural network model in the training data was utilized for training purposes two separate datasets:-

i) MNIST (Modified National Institute of Standards and Technology database) dataset

ii) Dataset self-created.

This two dataset nearly fills the training model with 60500 entries and successfully iteratively trains the model almost five times.

2. Image processing in Testing Data

When it comes to testing data, at first I assumed it would be a photo that was already in the database and had been created by a human. Real-time photos are taken by the computer's webcam during the testing process to add interest. This component has made the project and curriculum more difficult while also making it more engaging. The training dataset's handwritten digit removal was then used for the purpose of three sections. There exist two primary elements to this process:-

1) Utilize the webcam to capture an image.

2) Resize, scale, and noise cancellation are all used.

1) Utilize the webcam to capture an image: Google Colab is used as an interactive webcam program in this section used colab's IPYTHON and javascript to set the fitting image into the webcam. Javascript will launch the video and display whatever is captured by the mirror camera. The stream will remain steady till the capture button turns "True." If the capture succeeds, the chosen pixel will be counted and saved as photo.jpg in the Google Colab's internal storage. In the IPYTHON display, a 2D array is manually set to obtain the image's pixelated input. The capture is configured as a promise function.

2) Resize, scale, and noise cancellation are all used: Py means filtering has been eliminated because the photograph that has been captured in the webcam has been transmitted through the numpy kernel at first because the illustration captures single character only. The image design is done by copying the pixel intensity values from the python display to a numpy array, normalizing that image is done as previous training dataset entry, scaling and resizing done as the same, and here's the visual designed using a 20, 20 kernel. Image processing has the potential for use to transform the complete unsupervised dataset into supervised data. In the project, the preparation work includes webcam access, photo inputs, scaling, resizing, filtering, noise cancellation, and finally transmitting the machine learning method as a kernel for testing.

9. EVALUATION

TEST CASES:

Test Case ID: TC01

Test Case: Accuracy Evaluation

Pre-Condition: Utilizing a different test dataset, assess the overall correctness of your model.

Expected Result: As percentage of numbers that were correctly predicted to all of the test samples, determine the accuracy.

Actual Result: As percentage of numbers that were correctly predicted to all of the test samples, determine the accuracy.

Status: Pass

Test Case ID: TC02

Test Case: Confusion Matrix

Pre-Condition: For every distinct integer, make a matrix of perplexity.

Expected Result: This can show you whether some numerals are routinely misclassified more often than others.

Actual Result: This can show you whether some numerals are routinely misclassified more often than others.

Status: Pass

Test Case ID: TC03

Test Case: Misclassified samples

Pre-Condition: Give a few instances of incorrectly labeled numerals, together with their expected and actual labels.

Expected Result: This can highlight any potential weak points in your model and offer recommendations for potential use development.

Actual Result: This can highlight any potential weak points in your model and offer recommendations for potential use development.

Status: Pass

10. SCREENSHOTS:

Figure 1: After running the code, the Tkinter blank screen is shown below.

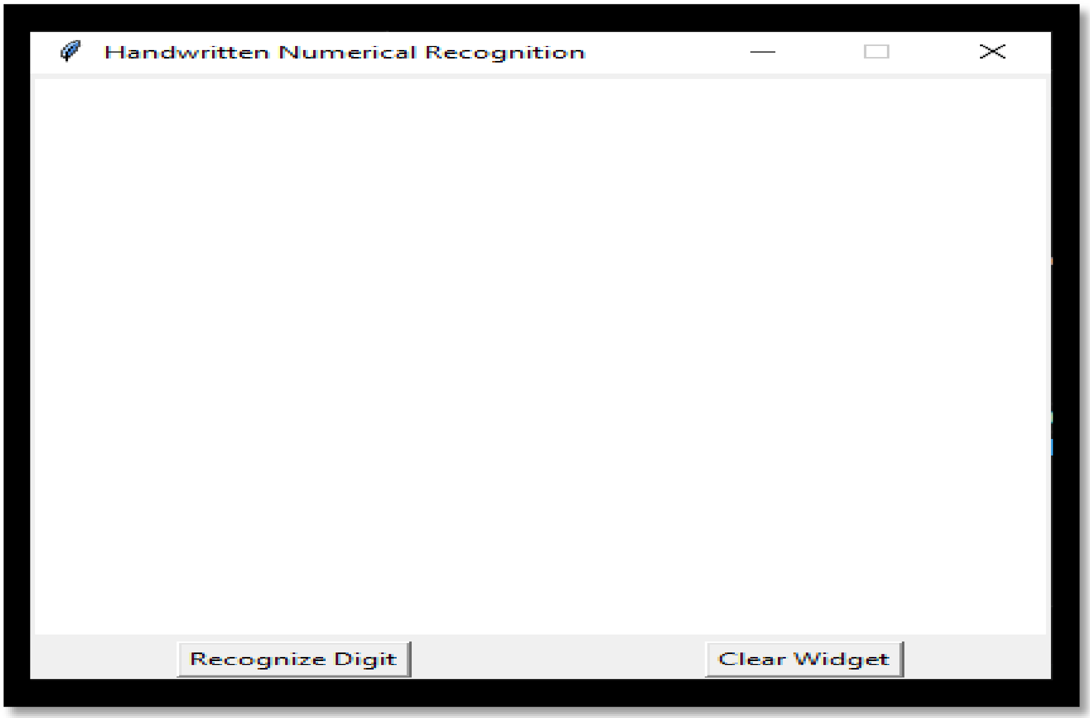


Figure 2: Then, draw the number on the screen. then press the button that says "recognize digit."

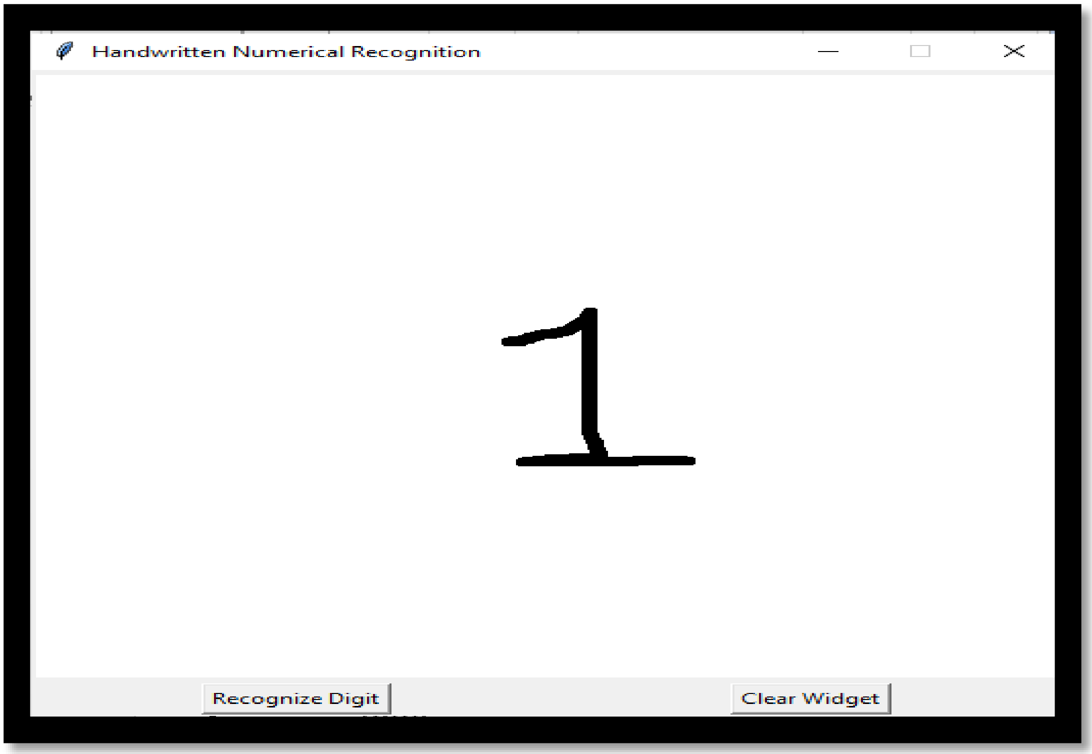


Figure 3 : When a number is recognized, the results are displayed.

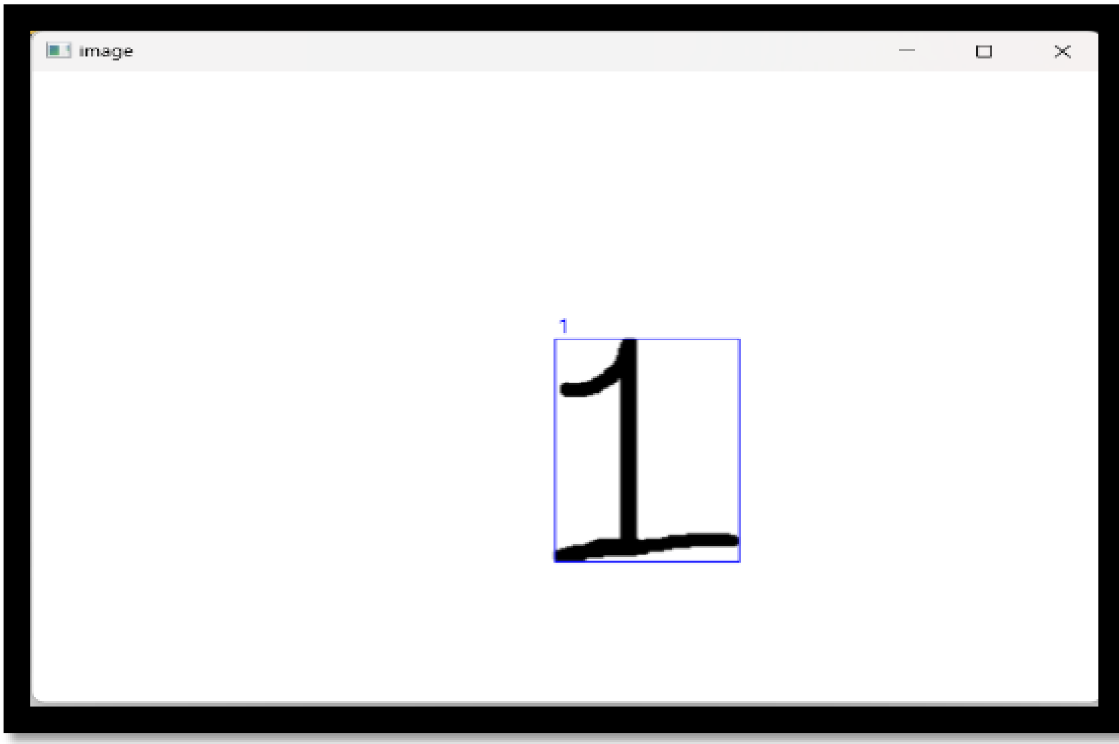


Figure 4: You can draw several digits on the screen and then select the button that says "recognize digit."

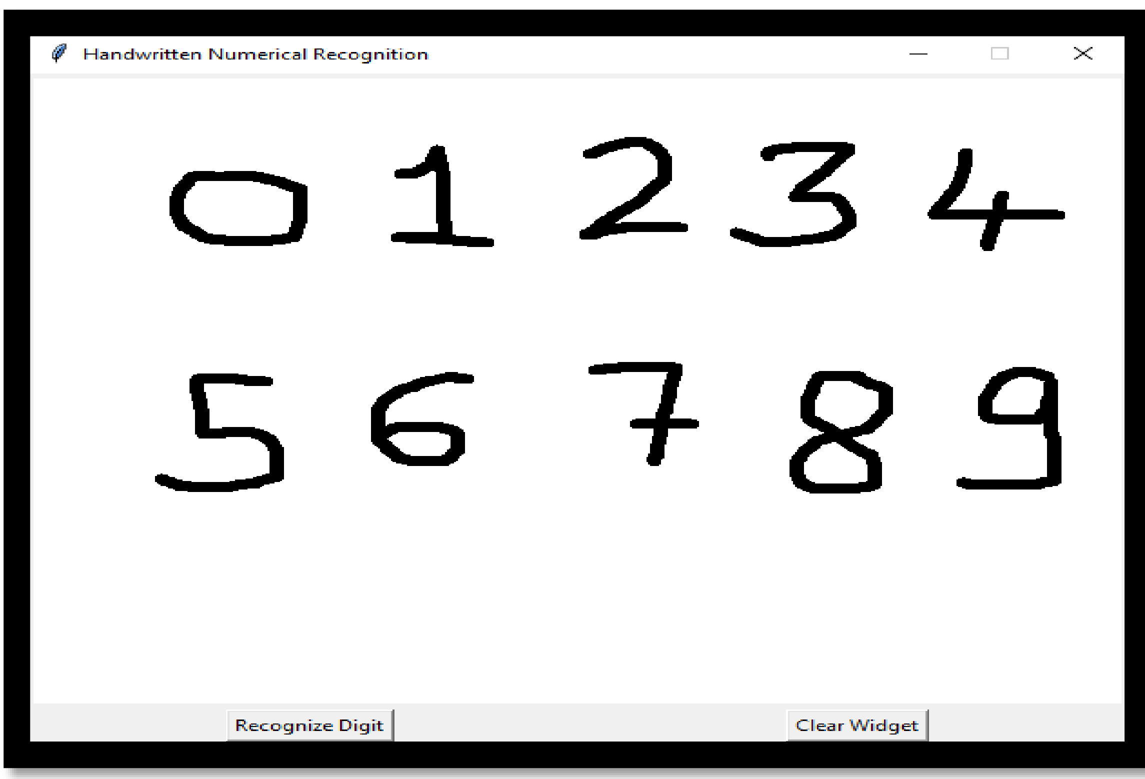
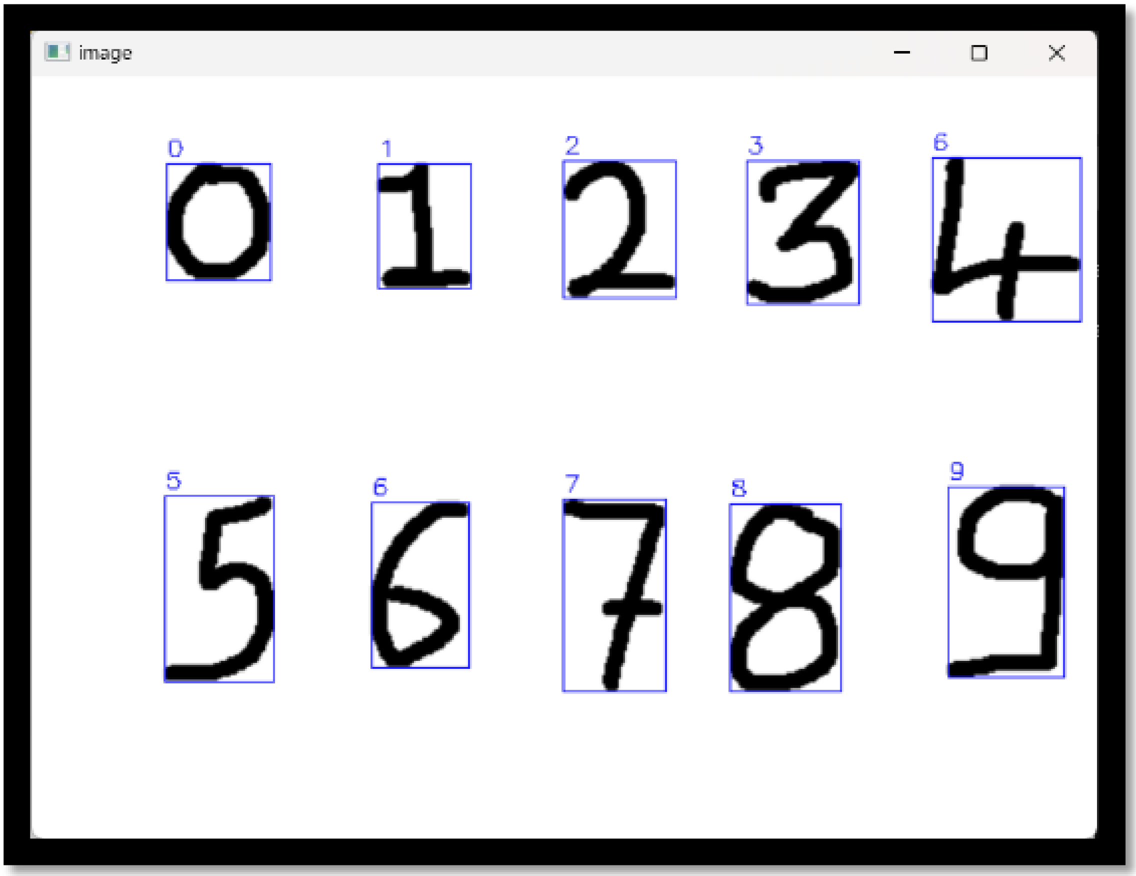


Figure 5: When numerous numbers are identified, the results are displayed on the screen.



CONCLUSION:

Using deep learning techniques, handwritten numerical recognition has been implemented in this paper. It has in relation to recognizing the numbers. The primary purpose of this research is to develop a robotic process for reading handwritten digit strings. The handwritten numerical recognition task was successfully completed by CNN. Real-world photographs can be used to identify utilizing suggested method, which also achieved accuracy of over 98%; both training and evaluation loss rates are virtually non-existent. Only the noise during input canvas image poses a challenge, and it must be addressed. In how many dense neurons and the cross-validation measure both significantly affect the rate at which learning occurs for the given model.

FUTURE WORK:

The suggested system accepts picture that have a size of 28x28 pixels as input. The Handwritten Character Recognition technology, which identifies human handwritten characters and forecasts the output, may be built using the same technology with further adjustments and enhancements in the dataset and the model.

11. REFERENCES:

[1] “Handwriting recognition”: https: //en.wikipedia.org/wiki/Handwrittingrecognition

[2] “What can a digit recognizer be used for?”: https: //www.quora.com/What−can−a−digit−recognizer− be − used − f or

[3] L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]," in IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 141-142, Nov. 2012.

[4] M. M. A. Ghosh and A. Y. Maghari, "A Comparative Study on Handwriting Digit Recognition Using Neural Networks," 2017 International Conference on Promising Electronic Technologies (ICPET), Deir El-Balah, 2017, pp. 77-81.

[5] Attacks, W., 2019. International Journal of Advanced Trends in Computer Science and Engineering. International Journal, 8(1.2).

[6] K. Fukushima, ‘‘Neocognitron for handwritten digit recognition,’’ Neurocomputing, vol. 51, pp. 161–180, Apr. 2003.

[7] Shawon, A., Rahman, M.J.U., Mahmud, F. and Zaman, M.A., 2018, September. Bangla handwritten digit recognition using deep cnn for large and unbiased dataset. In 2018 international conference on Bangla speech and language processing (ICBSLP) (pp. 1-6). IEEE.

[8] “Handwritten Digit Recognition using Machine Learning Algorithms”, S M Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana & Abdullah Al Jobair.

[9] Jain, M. Kaur, G. Quamar, M.P. and Gupta, H., 2021, February. Handwritten digit recognition using CNN. In 2021 International Conference on Innovative Practices in Technology and Management (ICIPTM) (pp. 211-215). IEEE.

[10] K. B. Lee, S. Cheon, and C. O. Kim, "A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes," IEEE Transactions on Semiconductor Manufacturing, vol. 30, no. 2, pp. 135-142, 2017.

[11] Zhan, H., Lyu, S. and Lu, Y., 2018, August. Handwritten digit string recognition using convolutional neural network. In 2018 24th International Conference on Pattern Recognition (ICPR) (pp. 3729-3734). IEEE.

[12] “Handwritten recognition using SVM, KNN, and Convolutional Neural Networks”, Norhidayu Binti Abdul Hamid, Nilam Nur Binti, Amir Sharif.

DrillBit

The Report is Generated by DrillBit Plagiarism Detection Software

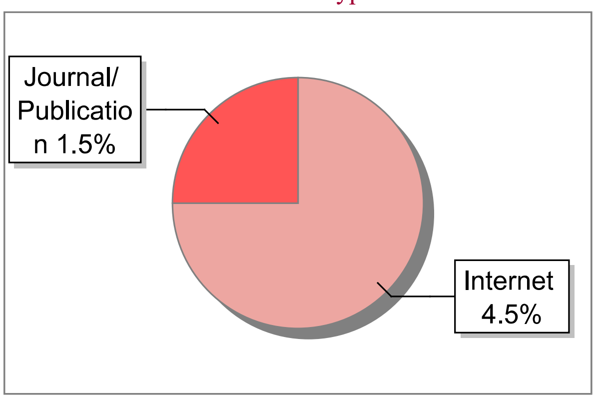
Submission Information

Author Name	02FE21MCA028
Title	02FE21MCA028
Paper/Submission ID	933971
Submission Date	2023-08-28 15:32:39
Total Pages	28
Document type	Project Work

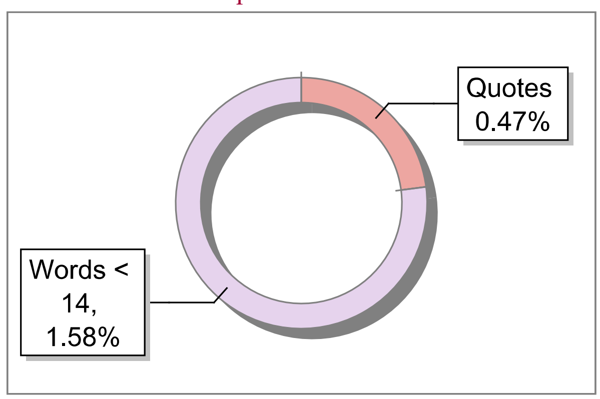
Result Information

Similarity **6 %**

Sources Type



Report Content



Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Sources: Less than 14 Words Similarity	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

6	16	A	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)
SIMILARITY %	MATCHED SOURCES	GRADE	

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	1library.co	1	Internet Data
2	journals.sagepub.com	1	Internet Data
3	moam.info	1	Internet Data
4	drupalsun.com	<1	Internet Data
5	deeplearn.org	<1	Internet Data
6	www.dx.doi.org	<1	Publication
7	www.diva-portal.org	<1	Internet Data
8	arxiv.org	<1	Publication
9	amedpost.com	<1	Internet Data
10	genomebiology.biomedcentral.com	<1	Publication
11	dspace.nwu.ac.za	<1	Publication
12	acrwebsite.org	<1	Internet Data
13	uir.unisa.ac.za	<1	Publication
14	1library.co	<1	Internet Data

15	Some Metallurgical Applications of Principal Components by Spurrell-1963	<1	Publication
16	Using neural networks to aid CVSS risk aggregation An empirically validated ap by Beck-2016	<1	Publication