Project Report
On

"Handwritten Equation Solver Using CNN"

Submitted in partial fulfillment for the award of degree
Of

BACHELER OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

Submitted by

1. Shruti More (17141105)
2. Anjali Kevale (17141134)
3. Simran Shinde (17141135)
4. Sneha Koli (17141140)

Under the Guidance of
Prof. R. S. Mawale
Prof. C. P. Garware

Government College of Engineering, Karad
(An Autonomous Institute of Government of Maharashtra)
Academic Year 2020-2021

Department of Information Technology

CERTIFICATE

This is to certify that the project entitled "Handwritten Equation Solver Using CNN" has been carried out by team:

1. **Shruti More (17141105)**
2. **Anjali Kevale (17141134)**
3. **Simran Shinde (17141135)**
4. **Sneha Koli (17141140)**

of B. Tech IT class under the guidance of Prof. R. S. Mawale and Prof. C. P. Garware during the academic year 2020-21 (Se m-VIII).

Prof. R. S. Mawale,
Prof. C. P. Garware

Project Guide

Dr. S. J. Wagh

Head
Information Technology Department

External Examiner

ACKNOWLEDGEMENT

Apart from individual efforts, the success of any project depends largely on the encouragement & guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental throughout the project work

It is our privilege to express our gratitude towards our project guide**, Prof. R. S. Mawale** and **Prof. C. P. Garware**, for their valuable guidance, encouragement, inspiration and wholehearted cooperation throughout the project work. We thank her for being a motivation through all our highs and importantly, our lows.

We deeply express our sincere thanks to our Head of Department **Dr. S. J. Wagh** for encouraging and allowing us to present the project on the topic "**Handwritten Equation Solver Using CNN**" and providing us with the necessary facilities to enable us to fulfill our project requirements as best a s possible. We take this opportunity to thank all faculty members and staff of Department of Information Technology, who have directly or indirectly helped our project.

We pay our respects to honorable Principal **Dr. A. T. Pise** for their encouragement. Our thanks and appreciations also go to our family and friends, who have been a source of encouragement and inspiration throughout the duration of the project.

ABSTRACT

Context - Solving mathematical equations is an integral part of most, if not all forms of scientific studies. In this project, we present a application that can process an image of a handwritten mathematical equation captured using the camera, recognize the equation, form the corresponding string that can be parsed by a computer algebraic system and display all possible solutions. We aim to make the whole experience of experimenting with equations very user friendly and to remove the hassle of learning a mathematical tool just for mathematical experimentation. We propose a novel machine learning approach to recognize handwritten mathematical symbols achieving a 97% cross validation percentage accuracy on the kaggle math symbol dataset with reduced symbols. The application covers useful features like simultaneous equation solving, and simple arithmetic computations from images. Overall it is a very user friendly equation solver that can leverage the power of existing powerful math packages.

Keywords - Machine learning, Python, Handwritten Equation, Simplification, Preprocessing, Segmentation, Implementation, Recognition, CNN, Polynomial expressions, Image processing.

LIST OF FIGURES

ABBREVATIONS

Acronym	Definition
ML	Machine Learning
OCR	Optical Character Recognition
CNN	Convolutional neural network
HCR	Handwritten Character Recognition
CUDA	Compute Unified Device Architecture
GPU	Graphical Processing Unit
Open CV	Open Source Computer Vision Library

TABLE OF CONTENTS

Topics

- Abstract
- List of Figures
- Abbreviations

1.INTRODUCTION

Recently Convolutional Neural Networks (CNN) became one of the most appealing approaches and has been an ultimate factor in a variety of recent success and challenging machine learning applications such as challenge image object detection, image segmentation and face recognition. Therefore, we choose CNN for our challenging tasks of image classification. Handwritten digit recognition has gained so much popularity and developing such a system includes a machine to understand and classify the images of handwritten digits as 10 digits (0–9) and solves the equations.

1.1. Background

Generally Handwriting Character Recognition (HCR) is categorized into six phases which are acquisition of image, pre-processing of enter image, segmentation, feature extraction, classification and put up processing.

A. Image Acquisition

The input photo is supplied to the consciousness gadget at the Image Acquisition stage. The input can be either in an photograph layout such as JPEG, BMT, etc., or a scanned image, digital camera, or any other gorgeous digital input machine or can be taken from the canvas on the person interface.

B. Pre-Processing

The 2nd method, known as pre-processing, is the entry approach for personality cognizance and is very essential in finding out the focus quality. Preprocessing operates to normalize strokes and also to take away deviations that can minimize the accuracy rate.

C. Segmentation

Segmentation is used to transform the enter representation of many characters to the individual characters. The methods used are the segmentation of words, strains and characters. Typically, it is carried out by way of isolating a single persona from a word picture.

D. Feature Extraction

The aim of the extraction characteristic is to permit the extraction of the sample that is most important for classification. By flattening the array into a vector of 28 * 28 = 784 numbers, the photograph now converges to a minimal bunch of arrays in a quite high-quality structure 784-cell dimension. The photograph now turns into a n dimensional array tensor.

E. Classification

Decision-making takes vicinity at some point of the classification process. The extracted attributes are used to become aware of the characters. Different classifiers algorithms are used, such as SVM and Neural Networks. The classifiers sort the precise input function with reserved sample and find the best matching input classification for which Soft Max Regression is being used.

F. Post-Processing

The Post-processing is the last and ultimate phase of personal recognition. It is the procedure whereby herbal language is used to right the misclassified output. It procedures output by means of getting it after a recognition of the shape. If the shape is diagnosed basically then the accuracy can be increased in accordance to language knowledge. For exclusive handwriting inputs, shape recognizers behave differently.

1.2. Importance of project

After this project work completes, our system can be used to convert mathematical equations into computer readable form which in turn used to solve the equations and it will provide the answer of that equation. It also presents the characters in an easy to understand pattern that can be further improved to carry out calculation and further train to be able to recognize equations after feeding the data in the model.

This project can also be implemented in other complex projects such as online calculators which will be able to just take the image of the equation and compute the solution for this equation without additional human input and intelligence. This has number of advantages which mainly lead to saving of time during such equation solving. Our system is certainly be helpful in many fields which currently have a schedule to go with the time loss in solving and parsing such equations.

Using the high computation power of current computers and using our system one can solve even complex equations in matter of seconds. Hence after completion of this project, it can be released publically which in turn can be used by many people for purposes like parsing, extraction, recognition and calculation of characters comprising of the equation and also for further use in personal or advanced projects.

1.3. Motivation

Machine Learning is a method which trains the machine to do the job by itself without any human interaction. At a high level, machine learning is the process of teaching a computer system on how to make accurate predictions when fed the data. Those predictions will be the output. There are many

sub-branches in machine learning like Neural Networking, Deep Learning. Among these, Deep Learning is considered to be the most popular sub-branch of Machine Learning. A human can easily solve and recognize any problem, but this is not the same in the case of a machine. Many techniques or methods should be implemented to work as a human. In online handwriting recognition of letters, an on-time compilation of letters is performed while writing because stroke information is captured dynamically. Whereas, in offline recognition, the letters aren't captured dynamically. Online handwriting recognition is more accurate when compared to offline handwriting recognition because of the lack of information. [3]

1.4. Expected Outcome

The application will be able to predict and solve handwritten mathematical equations from the given image. The system should be capable of solving expression involving arithmetic operations (addition, subtraction, multiplication, division).

It involves just the use of manual photo capture of the mathematical equation and feeding the photo to the software to obtain the characters then the mathematical equation is comprised of for further processing and calculation without additional human input and intelligence.

2. LITERATURE SURVEY

So here is a comprehensive summary of previous research on a topic. It focuses on Current developments in this area of research, existing problem solving techniques used by researchers in this field, the efficacy of their methods and a comparison of various methods/solutions existing and why we choose to use some of them for our project.

Sr.no.	Author	Title	Advantages
I.	1)Y. A. Joarder World University of Bangladesh 2)Md. Bipul Hossain 3)Feroza Naznin 4)Md Zahidul Islam	Recognition and Solution for Handwritten Equation Using Convolutional Neural Network	A single quadratics and also series of quadratics is used for the recognition. Projection analysis especially compact horizontal projection is used for the segmentation of each line of quadratics. Connected component which has very high success rate is used for character segmentation. [5]
II.	1)Shifat Nayme 2)Fuad Hasan 3)SyedAkhter Hossain 4)Sheikh Abujar	Handwritten Polynomial Equation Recognition and Simplification Using Convolutional Neural Network	In this experiment, thousands of equations are tested and some of them Recognized falsely and generated the wrong string equation. Almost 97% of images are recognized correctly out of a thousand images and generate the correct string form. [8]

III.	1)Mohini Lokhande 2)Kirti Murudkar 3)Arbaaz Nadaf 4) Niyati Shah	Handwritten Math Problem Solver Using Convolutional Neural Network	Handwritten Equation Solver trained by handwritten digits and mathematical symbol using Convolutional Neural Network with some image processing techniques to achieve a decent accuracy of 98.46%.[1]
IV.	1)Anmol Adhikari	Handwritten Digit Recognition using CNN	The system shows 97% of accuracy.[4]
V.	1)Huimin Wu Research School of Computer Science, The Australia National University, Canberra	CNN-Based Recognition of Handwritten Digits	According to the model, the final accuracy is stable at more than 90%.[6]

VI.	1)A. Choudhary 2) Savita Ahlawat 3)Harsh Gupta 4)Anirudh Bhandari 5) Ankur Dhall 6)M. Kumar	Offline Handwritten Mathematical Equation Evaluator Using Convolutional Neural Network	In this Experiment, the expressions/equations are evaluated by tokenizing, converting into postfix expressions and then solving using a custom-built parser. [9]
VII.	1)Giang Son Tran 2)C. Huynh 3)Thanh-Sach Le 4) T. Phan	Handwritten Mathematical Expression Recognition Using Convolutional Neural Network	For structural analysis, we use the DRACULAE parser since it has high accuracy given that the symbols were

	5) Khanh-Ngoc Bui		correctly detected. [7]
VIII.	1)Sanjay S. Gharde 2) P. Baviskar 3)K. P. Adhiya	Identification of Handwritten Simple Mathematical Equation Based on SVM and projection Histogram	This evaluation study suggests projection histogram for extraction technique and support vector machine for classification technique for implementation. Using projection profile and support vector machine two different dataset are recognized then 97.58% and 98.40% (as an average it resulted into 98.26%) recognition rate is achieved for simple handwritten mathematical equation.[2]
IX.	1) Ronald Clark 2) Quik Kung 3) A. V. Wyk	System for the Recognition of Online Handwritten Mathematical Expressions.	The classification is performed using a support vector machine (SVM). Two methods are implemented and evaluated to perform the structural analysis: one based on the open-source DRACULAE parser and another on a custom adjacency matrix technique. The results show that the SVM has a 94.83% symbol classification accuracy while the DRACULAE parser and

			custom adjacency matrix technique have expression recovery rates of 80% and 43.3%, respectively.[3]

3. PROBLEM DEFINITION AND RELATED THEORY

3.1. Problem Definition

Handwritten Equation Recognizer is a software program written to ease the process of recognizing the characters that comprises in any given mathematical equations. It involves just the use of manual photo capture of the mathematical equation and feeding the photo to the software to obtain the characters the mathematical equation is comprised of for further processing and calculation. Hand-written digit cognizance is the capacity of a PC system to apprehend hand-written inputs such as digits.

Several classification methods using Machine Learning have been developed and used for this purpose, such as K-Nearest Neighbors, SVM Classifier, Random Forest Classifier, etc., but these methods, whilst having the accuracy of 97%, are not adequate for real-world purposes. In current years, the research community has been gaining significant interest in deep learning-based strategies to remedy a range of supervised, unsupervised and reinforced getting to know problems.[4] One of the most regularly occurring and broadly used strategies is Convolution neural networks (CNN's), A kind of neural networks which can extract relevant features robotically from enter information.

3.2 Related Theory

Machine Learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. It can also be defined as the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit human intervention, relying on patterns and inference instead. Similarly, a mathematical equation is a statement that defines the equality of two expression which can be used to define almost all the remaining mathematical theorems and science theories.[8]

Handwritten Equation Recognizer is a software program written to ease the process of recognizing the characters that comprises in any given mathematical equations. It involves just the use of manual photo capture of the mathematical equation and feeding the photo to the software to obtain the characters the mathematical equation is comprised of for further processing and calculation. For a normal human being, pattern recognition is the most basic of the human learning concepts which comes as a second nature to most of us. The same is not true for computers and machines. To identify patterns and be able to compute on the information that is obtained through the identification of the pattern is a great accomplishment of a computer. This poses a great challenge in making a computer to be able to understand and parse mathematical expressions which in turn are typical example of pattern

recognition. Being able to design a neural network that can successfully ‘teach’ a computer to recognize mathematical patterns so that it can finally carry out further computation on the expression automatically or with least manual input from humans will most certainly bring about a great ease in mathematical equations calculations. The software created on the end of this project work aims to do just the things mentioned above. It trains a computer model to recognize mathematical patterns and is able to return to the user a list of characters that are involved in making the equation.

4. DESIGN METHODOLOGY

In our proposed method at first noise from the original input image is removed by applying binarization to it. After that we use compact horizontal projection for segmenting each line of equation from the input image. Then we consider each part of the segmented image as a full image for further process. For each line of equation image we then find specific characters in the form of connected component. Each segmented character is then providing as input to the convolutional neural network model for classification of the character. The resulting character that is the output of CNN is then used for making a character string which is similar to the original equation.

4.1. Proposed System Architecture:

This represents the design possibilities of the proposed system, in order to define the steps such as pre-processing, feature extraction, segmentation, training, classification and recognition of digits. The above Figure illustrates the block diagram of the proposed system. The proposed model contains the five stages in order to classify and detect the digits

Block Diagram:

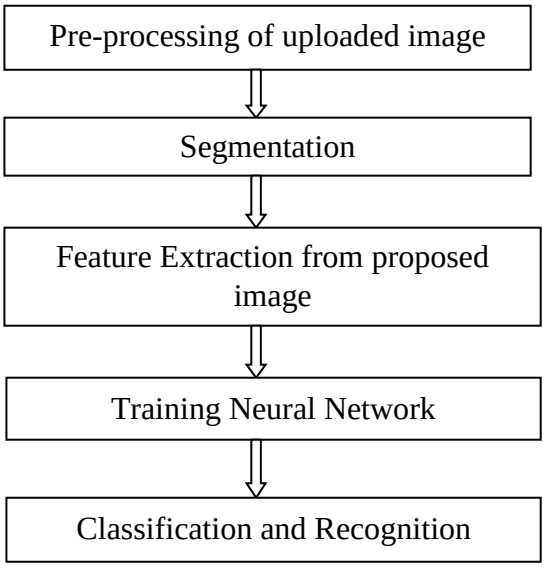


Fig.4.1.Block diagram of proposed method

A. Dataset Preparation

Preparation of the dataset is the fundamental concern for this work. Characters such as English digit, alphabet and mathematical symbol all can be well defined by their edges. For this reason we first prepare dataset as the most precedence given to its edges that is enlighten the edges. We prepare some

dataset by ourselves and we also used some modified version of NIST dataset which is similar as the popular MINIST dataset for digit. For each category we use 2000 data item for the training of the network. And most of the case our network training we gained more than 97% training accuracy. Image size we used in dataset is 32×32 gray level image.

B. Pre-Processing

Pre-processing of the input image is the procedure which encompasses changes and modifications to the image to make it fit for recognition. The following techniques may be used for image enhancement.

a)Conversion of RGB to Gray-Scale

First of all this coloured image is transformed into a typical gray-scale image and is represented through a single matrix because the detection of characters on a coloured image is more challenging than on a gray-scale image. If the gray bitmap Y and color bitmap is R, G and B then the formula we used is: $Y = 0.299R + 0.587G + 0.114B$

C. Binarization

Binarization is the procedure of choosing a threshold value is for adaptation of pixel values into 0's and 1's. In this research for horizontal projection calculation 1's represent the black pixels while 0's characterize the white pixels. The threshold choice of binarization can approve two ways: overall threshold and partial threshold. Otsu's method is an overall threshold method, basing on image statistical characteristics.

D. Segmentation

Identifying the objects or other significant information in digital images segmentation is mostly used in image processing and computer vision application which is the process of dividing an image into multiple parts[5]. The segmentation involves two main steps in our proposed method.

a)Equation line Segmentation

Equation segmentation is the split-up of the different lines of characters existing in the image. Each line is well-defined by a minimum vertical gap between the characters existing on a line and on the line overhead and below it. This gap can be used for the detection and separation of not the same lines of characters.

b) Character Segmentation

Character segmentation is a procedure that look for to decompose an image of a series of characters into sub images of individual symbols. In our proposed method we use connected component analysis method for the segmentation of specific character from the image.

E. Classification model

In this research convolutional neural network is used as a classification model , 32×32 gray scale image is used as input of the CNN input layer for the training section and also for the testing section. 7×7 filter is used at the convolutional layer after the convolution of the input image with the filter for each input image a 28×28 feature vector is produced. It is concord to apply a nonlinear layer (or activation layer) instantly after convolutional layer. The persistence of this layer is to introduce nonlinearity to a system[9]. To performs down-sampling by separating the input into rectangular pooling regions, and figuring the maximum of each region a max pooling layer is used. Pooling layers perform down-sampling operations.[5]

F. Training Data using Convolutional Neural Network

Since convolutional neural network works on two-dimensional data and our dataset is in the form of 785 by 1. Therefore, we need to reshape it. Firstly, assign the labels column in our dataset to variable y_train. Then drop the labels column from the dataset and then reshape the dataset to 28 by 28. Now, our dataset is ready for CNN.

G. Fitting Model to Data

It will take around three hours to train our model with an accuracy of 98.46%. After training, we can save our model as json file for future use, So that we don’t have to train our model and wait for three hours every time.

H. Testing our Model or Solving Equation using it

Now, input an image containing a handwritten equation. Convert the image to a binary image and then invert the image (if digits/symbols are in black).Now obtain contours of the image, by default, it will obtain contours from left to right. Obtain bounding rectangle for each contour. Sometimes, it will result in two or more contours for the same digit/symbol. To avoid that, check if the bounding rectangle of those two contours overlaps or not. If they overlap, then discard the smaller rectangle. Now, resize all

the remaining bounding rectangle to 28 by 28.Using the model, predict the corresponding digit/symbol for each bounding rectangle and store it in a string. After that use ‘eval’ function on the string to solve the equation.

4.2. Internal Logic of system:

4.2.1. Algorithm:

Convolutional Neural Network:

CNN or the Convolutional Neural Network (CNN) is a class of deep learning neural networks. In short think of CNN as a machine learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. CNN works by extracting features from the images. Any CNN consists of the following:

1. The input layer which is a grayscale image
2. The Output layer which is a binary or multi-class labels
3. Hidden layers consisting of convolution layers, ReLU (rectified linear unit) layers, the pooling layers, and a fully connected Neural Network

It is very important to understand that ANN or Artificial Neural Networks, made up of multiple neurons is not capable of extracting features from the image. This is where a combination of convolution and pooling layers comes into the picture. Similarly, the convolution and pooling layers can’t perform classification hence we need a fully connected Neural Network.

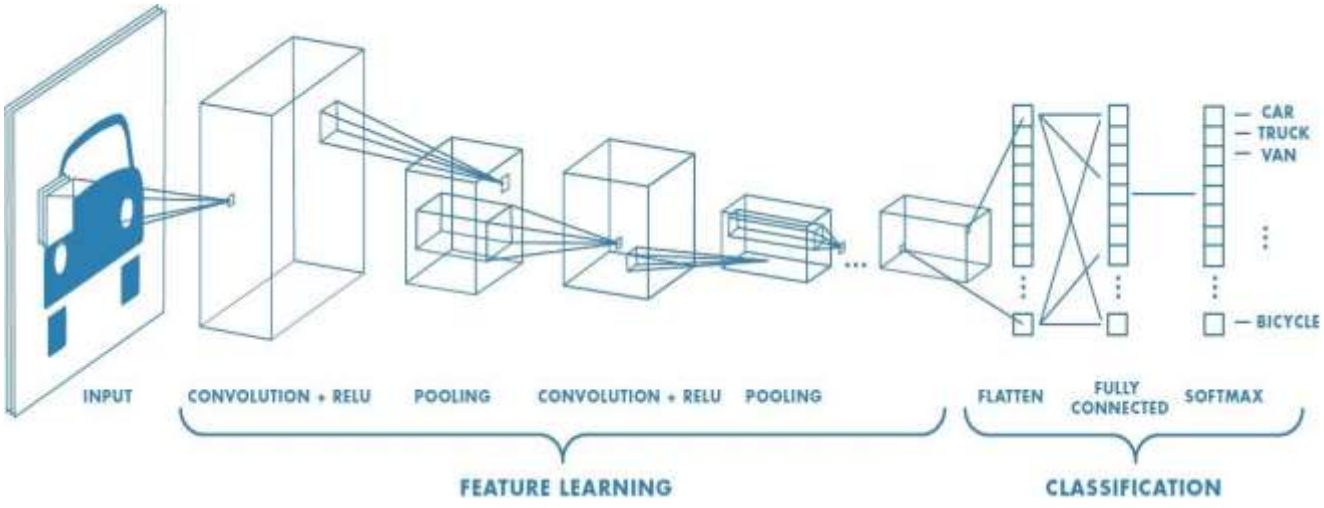


Fig.4.2.1 CNN process from input to Output Data

Let's consider that we have access to multiple images of different vehicles, each labeled into a truck, car, van, bicycle, etc. Now the idea is to take these pre-label/classified images and develop a machine learning algorithm that is capable of accepting a new vehicle image and classify it into its correct category or label. Now before we start building a neural network we need to understand that most of the images are converted into a grayscale form before they are processed. The challenge with images having multiple color channels is that we have huge volumes of data to work with which makes the process computationally intensive. In other worlds think of it like a complicated process where the Neural Network or any machine learning algorithm has to work with three different data (R-G-B values in this case) to extract features of the images and classify them into their appropriate categories.

The role of CNN is to reduce the images into a form that is easier to process, without losing features critical towards a good prediction. This is important when we need to make the algorithm scalable to massive datasets. So in our proposed model, CNN object classification mannequin takes, analyzed and classifies an enter photo which in our case is digits under a positive category.

4.2.2 CNN Layers:

ACNN consists of a lot of layers. These layers when used repeatedly, lead to a formation of a Deep Neural Network.

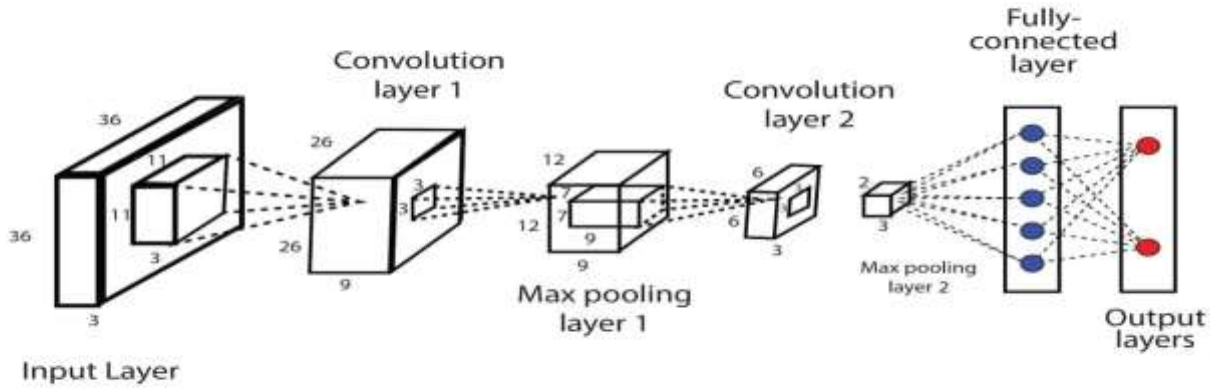


Fig.4.2.2. Convolutional Neural Network Layers

The fundamental types of layers used to build a CNN are:

1. Input

Input layers are connected with convolutional layers that perform many tasks such as padding, striding, the functioning of kernels, and so many performances of this layer, this layer is considered as a building block of convolutional neural networks. This layer holds the uncooked pixel values of photograph and convert it to grayscale pictures using 28x28 matrix of pixels.

2. Convolutional Layer

The convolutional layer's main objective is to extract features from images and learn all the features of the image which would help in object detection techniques. The input layer will contain some pixel values with some weight and height, our kernels or filters will convolve around the input layer and give results which will retrieve all the features with fewer dimensions. This layer gets the effects of the neuron layer that is linked to the enter regions. The wide variety of filters to be used in this layer is described here. Each filter may additionally be a 5x5 window that slider over the input records and receives the pixel with the most intensity as the output.

3. Rectified Linear Unit (ReLU) Layer

In this layer we remove every negative value from the filtered image and replace it with zero. This function only activates when the node input is above a certain quantity. So, when the input is below zero the output is zero. This layer applies an thing smart activation function on the picture records and makes use of again propagation techniques. ReLU function is utilized in order to preserve the equal values of the pixels and not being changed by means of the returned propagation.

4. Pooling Layer

The addition of a pooling layer after the convolutional layer is a common pattern used for ordering layers within a convolutional neural network that may be repeated one or more times in a given model. The pooling layer operates upon each feature map separately to create a new set of the same number of pooled feature maps. Down-sampling operation along the spatial dimensions (width, height), resulting in volume is utilized in this layer.

5. Fully Connected Layer

Fully Connected layers in a neural networks are those layers where all the inputs from one layer are connected to every activation unit of the next layer. In most popular machine learning models, the last few layers are full connected layers which compiles the data extracted by previous layers to form the final output. This layer is used to compute the score instructions that potential which class has the maximum score corresponding to the enter digits. The category label with the largest likelihood is chosen as the ultimate classification from the network and proven in the output.

4.3. Dataset

The MNIST database, an extension of the NIST database, is a low-complexity data collection of handwritten digits used to train and test various supervised machine learning algorithms. The database contains 70,000 28x28 black and white images representing the digits zero through nine. The data is split into two subsets, with 60,000 images belonging to the training set and 10,000images belonging to the testing set. The separation of images ensures that given what an adequately trained model has learned previously, it can accurately classify relevant images not previously examined.

Fig.4.3.Sample Dataset

So to Loads the MNIST dataset, we use load_data function as:

tf.keras.datasets.mnist.load_data(path="mnist.npz")

Arguments

path: path where to cache the dataset locally (relative to ~/.keras/datasets).

Returns

Tuple of NumPy arrays: (x_train, y_train), (x_test, y_test).

x_train**:** uint8 NumPy array of grayscale image data with shapes (60000, 28, 28), containing the training data. Pixel values range from 0 to 255.

y_train**:** uint8 NumPy array of digit labels (integers in range 0-9) with shape (60000,) for the training data.

x_test**:** uint8 NumPy array of grayscale image data with shapes (10000, 28, 28), containing the test data. Pixel values range from 0 to 255.

y_test**:** uint8 NumPy array of digit labels (integers in range 0-9) with shape (10000,) for the test data.

Python Libraries

A. Tensorflow

Tensorflow is used as backend in the application of this project. TensorFlow is an brilliant records circulation in the Machine Learning Library made by means of the Google Brain Team and made open supply in 2015. It is designed to ease the use and greatly relevant to each numeric and neural gadget troubles simply like different spaces. TensorFlow is essentially a low-level math-entangled tool that pursuits experts who apprehend what they're doing to construct exploratory studying structures, play around with them, and turn them into running programs. For the most part, it can be considered as a programming context in which equations can be entitled as graphs. Math things to do are spoken by using nodes in the graph, and the edges include the multidimensional facts clusters (tensors) linked to them.

B. Keras

Keras is used to build model to arrange the layers in the course of the implementation of this project. Keras is a high-level neural community API written in Python that can run on top of TensorFlow, CNTK, or Theano. It used to be developed with a focal point on allowing for quickly experimentation. The key to doing true lookup is being capable to go from notion to result with the least delay viable. Keras approves for handy and speedy prototyping (through person friendliness, modularity, and extensibility). Similarly, it supports each convolutional networks and recurrent networks, as properly as combinations of the two and runs seamlessly on CPU and GPU.

C. NumPy

NumPy is used for mathematical calculations to print out the predict records in this project. NumPy is the core bundle with Python for scientific computing. It is a versatile sophisticated (broadcasting) with N-dimensional array object characteristic software program for combining C / C++ and Fortran code, advantageous linear algebra, Fourier transform, and random number capabilities.

D. Matplotlib

Matplotlib is used to plot model accuracy and loss in a graphical view in this project. Matplotlib is a Python 2D plotting library that produces pleasant figures for the publication throughout platforms in a variety of hardcopy formats and interactive environments. Matplotlib can be used in Python scripts, Python and IPython shells, Jupyter notebook, Web software servers, and four interface toolkits for graphical users.

4.4. Data Flow and UML Diagram

The purpose of this design document is to explore the logical view of architectural design, sequence diagram, user interface of software for performing operations.

4.4.1. Flow Chart:

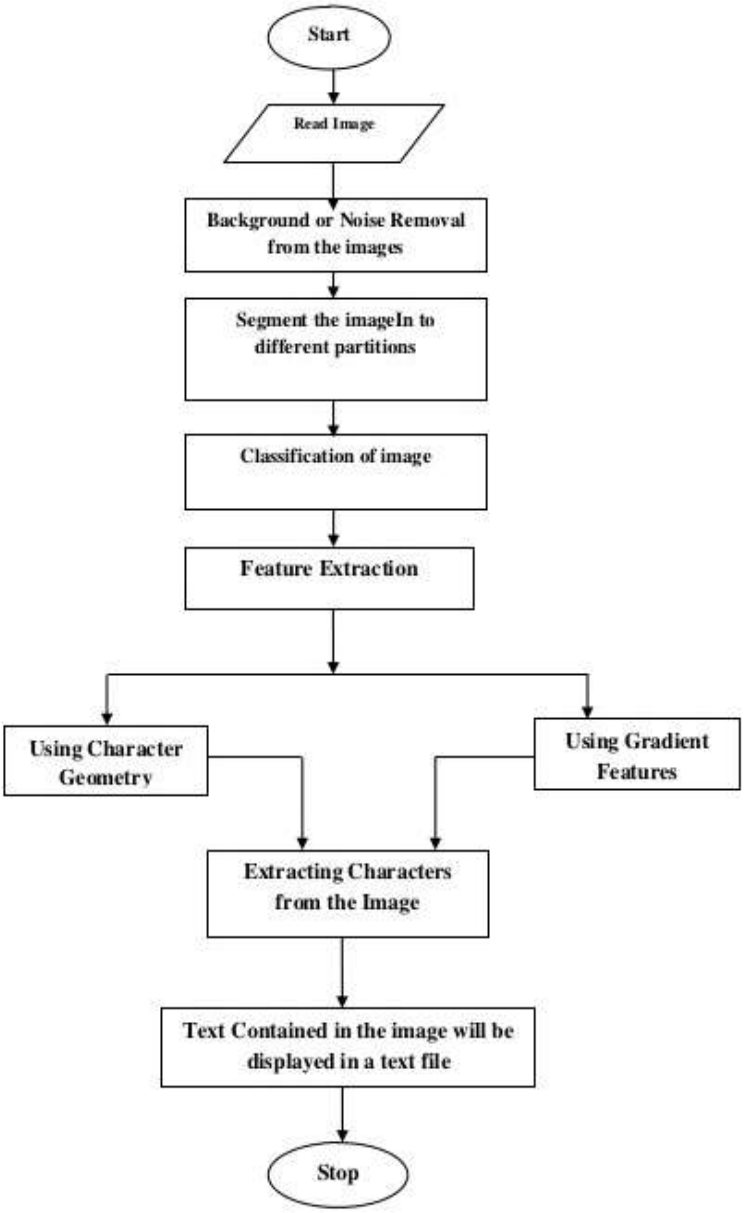


Fig.4.4.1.Flow chart of handwritten equation solver model

4.4.2. UML Diagram:

4.4.2.1.User Case Diagram:

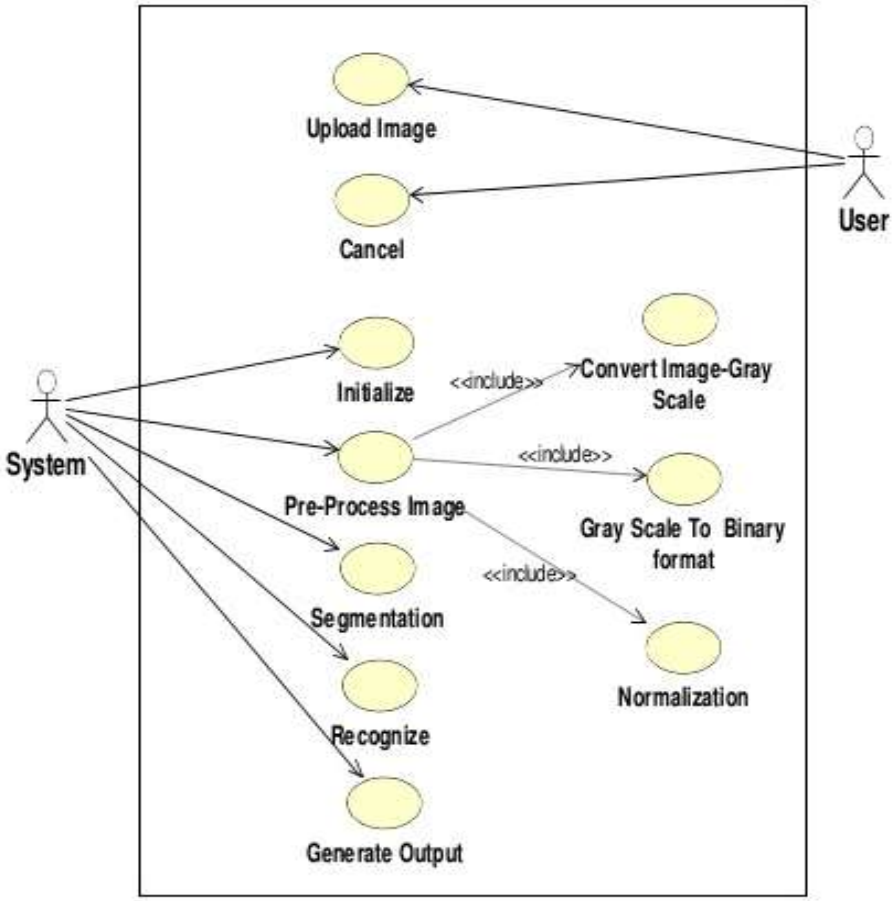


Fig.4.4.2.1 User case diagram

User Case	Description
Actor	User
Precondition	Input image should be available.
Main Scenario	User uploads image.
Extension Scenario	If the image is not compatible. Not possible to upload file.
Post Condition	Image successfully uploaded.

Fig.4.4.2.2 User module

Pre-processing Module

User Case	Description
Actor	**System**
Precondition	Uploaded input image
Main Scenario	Pre-processing is carried out by converting the image from RGB format to binary format.
Post Condition	Extract characters Before segmentation

Fig.4.4.2.3 Pre-processing module

Segmentation Module

User Case	Description
Actor	**System**
Precondition	Pre-processed image should be available.
Main Scenario	The pre-processed input image is segmented into isolated characters by assigning a number to each character using a labeling process. This labeling provides information about number of characters in the image. Each individual character is uniformly resized into pixels.
Extension Scenario	If the image is not compatible. Not possible to upload file.
Post Condition	Image successfully uploaded.

Fig.4.4.2.4 Segmentation module

4.3. Sequential Diagram:

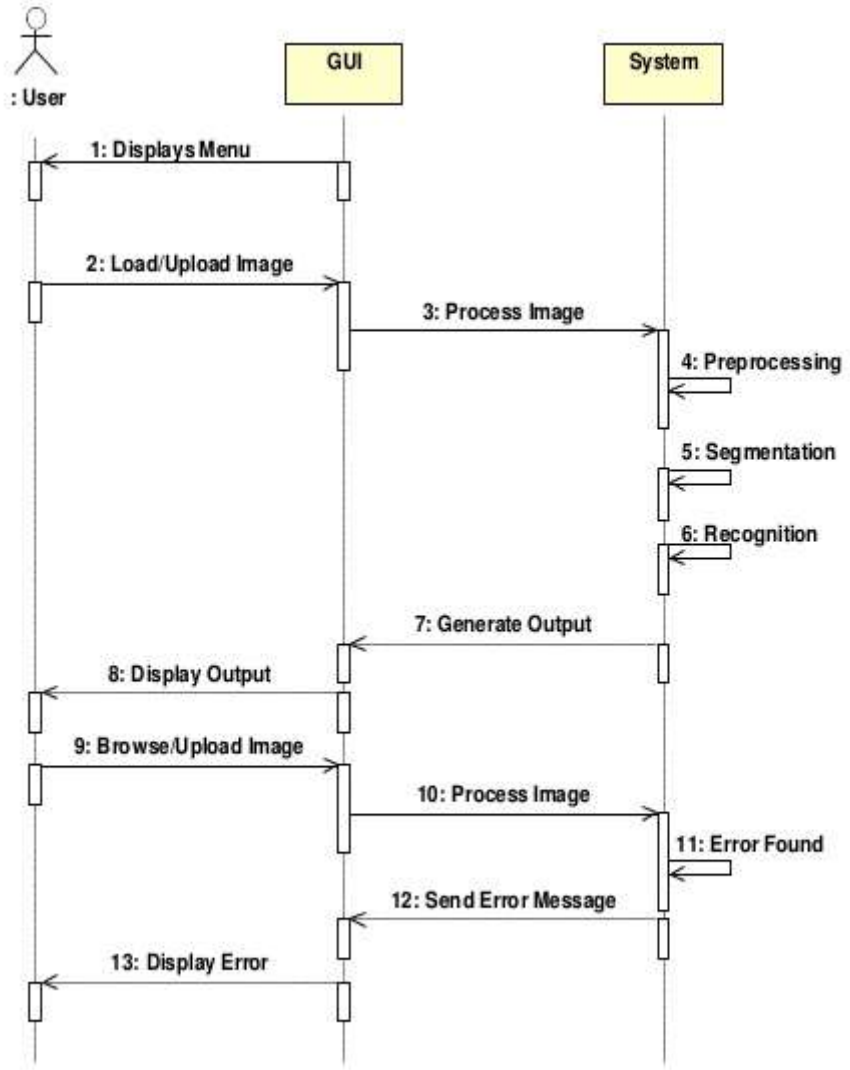


Fig.4.3. Sequential diagram

5. IMPLEMENTATION

Convolutional Neural Network (CNN) was implemented to recognize and predict handwritten digits from 0 to 9 and mathematical symbols. The proposed neural system was trained and tested on a dataset achieved from MNIST. The handwritten equation would be predicted and after that we would use 'eval' function on the string to solve the equation.

5.1. Implementation of Proposed System

At first, required libraries and packages are imported during the implementation of project.

```
In [1]:
import tensorflow as tf # Import tensorflow library
import matplotlib.pyplot as plt # Import matplotlib library
import numpy as np # Import numpy library
```

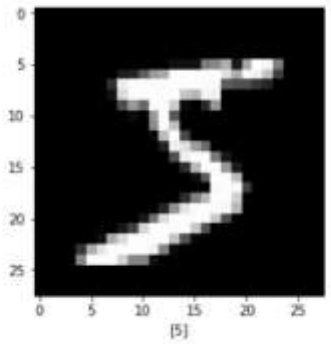
Then, object of the MNIST dataset is created and later loaded through the help of Tensorflow backend using Keras. After that, total number of train along with their dimension and test data set is printed out in order to view total number of elements present in the dataset.

```
In [2]:
mnist = tf.keras.datasets.mnist # Object of the MNIST dataset
(x_train, y_train),(x_test, y_test) = mnist.load_data() # Load data
print("x_train shape:", x_train.shape, "y_train shape:", y_train.shape,
      "x_test shape:", x_test.shape, "y_test shape:", y_test.shape)
```

```
x_train shape: (60000, 28, 28) y_train shape: (60000,) x_test shape: (10000, 28, 28) y_test shape: (10000,)
```

After the dataset is loaded, one of the images from the training dataset is loaded and displayed in gray scale format by using matplotlib library.

```
In [3]:
# Show one of the images from the training dataset
plt.imshow(x_train[0], cmap="gray") # Import the image
plt.xlabel([y_train[0]]) #Add label of the image
plt.show() # Plot the image
```



Then the training and testing data sets are normalized where image data values are converted in terms of 0 and 1.

```
In [4]:
x_train = tf.keras.utils.normalize(x_train, axis=1) # Normalize the training dataset
x_test = tf.keras.utils.normalize(x_test, axis=1) # Normalize the testing dataset
```

After normalizing the data, a CNN model is created using keras library. Then the Flatten layer is added into the model. Then after, input and hidden layer followed by output layers are built using CNN algorithm.

```
In [5]:
#Build the model object
model = tf.keras.models.Sequential()
# Add the Flatten Layer
model.add(tf.keras.layers.Flatten())
# Build the input and the hidden layers
model.add(tf.keras.layers.Dense(128, activation=tf.nn.relu))
model.add(tf.keras.layers.Dense(128, activation=tf.nn.relu))
# Build the output layer
model.add(tf.keras.layers.Dense(10, activation=tf.nn.softmax))
```

After building the model successfully, model is compiled using Adam optimization algorithm where this algorithm is used for training Deep Neural Networks (DNN).

```
In [6]:
#Compling the model using adam optimization algorithm which is used for training Deep NN.
model.compile(optimizer="adam", loss="sparse_categorical_crossentropy", metrics=["accuracy"])
```

```
WARNING:tensorflow:From C:\Users\BIPIN\Anaconda3\lib\site-packages\tensorflow\python\ops\resource_var
iable_ops.py:435: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be remo
ved in a future version.
Instructions for updating:
Colocations handled automatically by placer.
```

After compiling the model, model is fit to train data train and training is started using train data set, due to which cross entropy loss and accuracy of the model can be achieved while training the data from dataset.

```
In [7]: # Start training process using train data sets
        model_log=model.fit(x=x_train, y=y_train, batch_size=60, verbose=1, epochs=5, validation_split=.3)

Train on 42000 samples, validate on 18000 samples
Epoch 1/5
42000/42000 [==============================] - 3s 62us/sample - loss: 0.3592 - acc: 0.8984 - val_los
s: 0.2077 - val_acc: 0.9376
Epoch 2/5
42000/42000 [==============================] - 2s 55us/sample - loss: 0.1483 - acc: 0.9544 - val_los
s: 0.1460 - val_acc: 0.9551
Epoch 3/5
42000/42000 [==============================] - 2s 55us/sample - loss: 0.0994 - acc: 0.9699 - val_los
s: 0.1355 - val_acc: 0.9582
Epoch 4/5
42000/42000 [==============================] - 3s 62us/sample - loss: 0.0749 - acc: 0.9771 - val_los
s: 0.1244 - val_acc: 0.9625
Epoch 5/5
42000/42000 [==============================] - 3s 70us/sample - loss: 0.0552 - acc: 0.9832 - val_los
s: 0.1189 - val_acc: 0.9648
```

After the completion of training of the data set, performance of the model is evaluated using test data set, as well as accuracy and loss of data set is achieved.

```
In [8]: # Evaluate the model performance
        test_loss, test_acc = model.evaluate(x=x_test, y=y_test)
        # Print out the model accuracy
        print('\nTest Accuracy:', test_acc)
        print('\nTest Loss:', test_loss)

10000/10000 [==============================] - 0s 35us/sample - loss: 0.1084 - acc: 0.9687

Test Accuracy: 0.9687

Test Loss: 0.10838432838106528
```

After evaluation of model, prediction of the model is made using test data set.

```
import cv2
import numpy
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.utils import np_utils
from keras import backend as K
#K.keras.backend.image_dim_ordering()
#K.common.set_image_dim_ordering('th')
#K.common.image_dim_ordering()
K.set_image_data_format('channels_last')
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Conv2D, Flatten, Dense,Dropout,MaxPooling2D
from keras.models import model_from_json
```

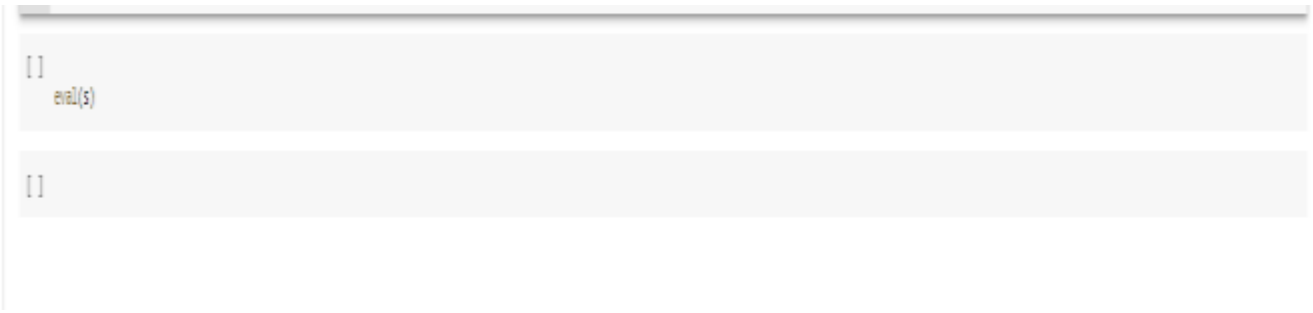
Firstly, import our saved model using the following line of codes.

```
+ Code  + Text                                                    Connect  Editing

[ ]
    json_file = open('model_final.json', 'r')
    loaded_model_json = json_file.read()
    json_file.close()
    loaded_model = model_from_json(loaded_model_json)
    # load weights into new model
    loaded_model.load_weights("model_final.h5")
```

Now, input an image containing a handwritten equation. Convert the image to a binary image and then invert the image(if digits/symbols are in black).Now obtain contours of the image, by default, it will

obtain contours from left to right. Obtain bounding rectangle for each contour. Sometimes, it will result in two or more contours for the same digit/symbol. To avoid that, check if the bounding rectangle of those two contours overlaps or not. If they overlap, then discard the smaller rectangle. Now, resize all the remaining bounding rectangle to 28 by 28.Using the model, predict the corresponding digit/symbol for each bounding rectangle and store it in a string. After that use ‘eval’ function on the string to solve the equation.



6. RESULT

We have successfully built a project “Handwritten Equation Solver Using CNN”. We have built and trained the convolutional neural network which is very effective for image classification purposes. Accuracy and speed of recognition are considered the better measure.

6.1. Result

After the completion of the project we obtained the following results:

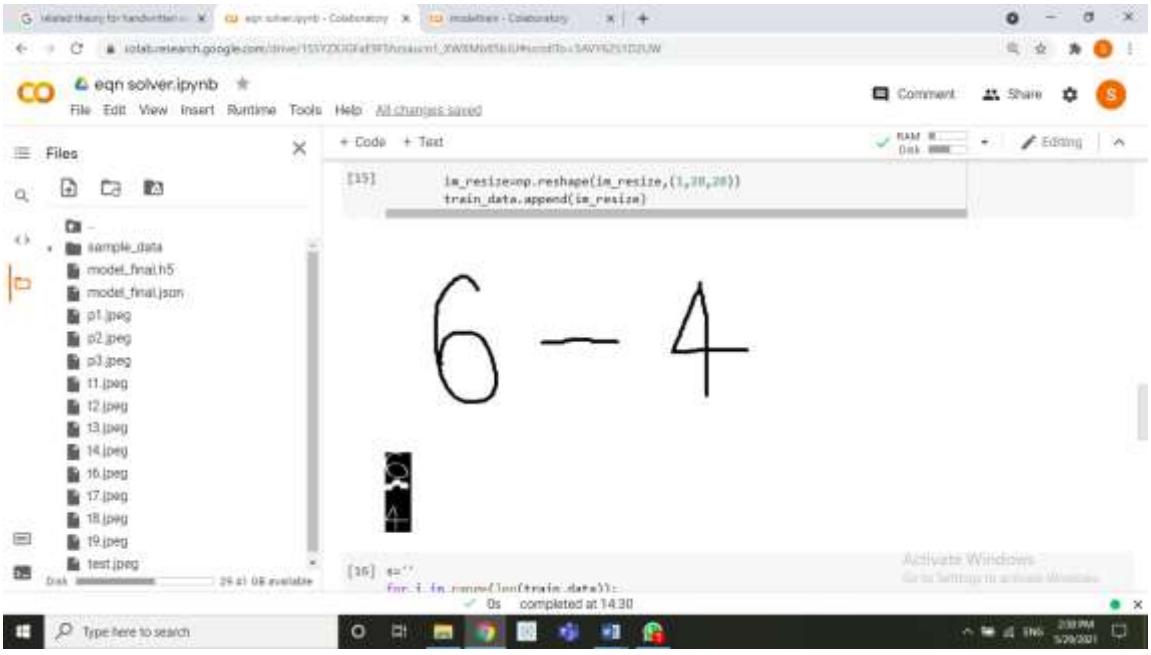
1. The accuracy of Training was found to be 97%.
2. The accuracy of Testing was found to be 96%.

These accuracies denote very good performance of our created model.

6.2. On-field Result Analysis

Some Screenshots of Handwritten Equation Recognizer are given below:

1] 6 - 4 = ?



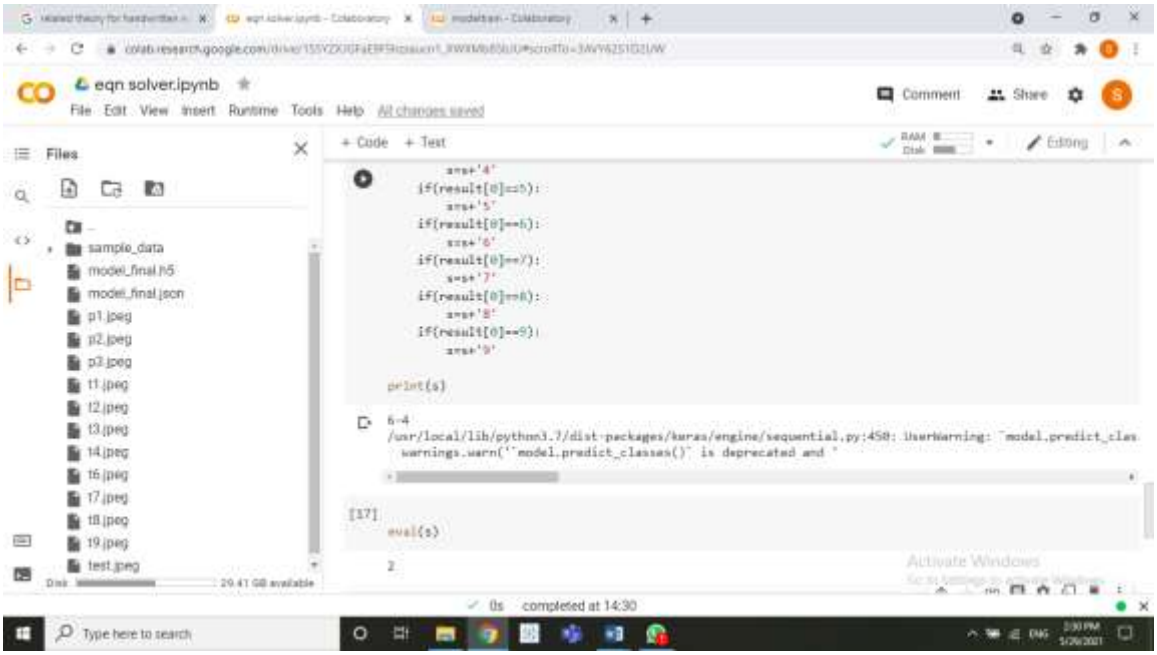
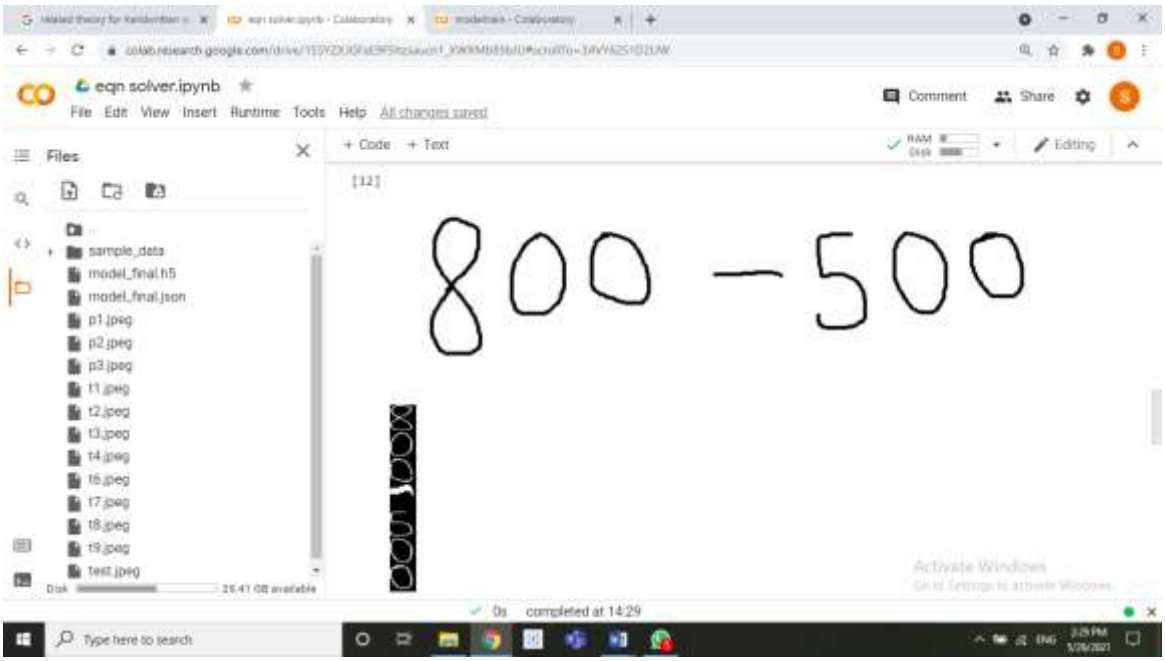


Fig 6.2.1 Sample Output (a)

When we perform a handwritten equation **6-4** ,firstly the model predicts handwritten equation and after that we had used 'eval' function on the string to solve the equation and gives output as **2**.

2] 800 -500 = ?



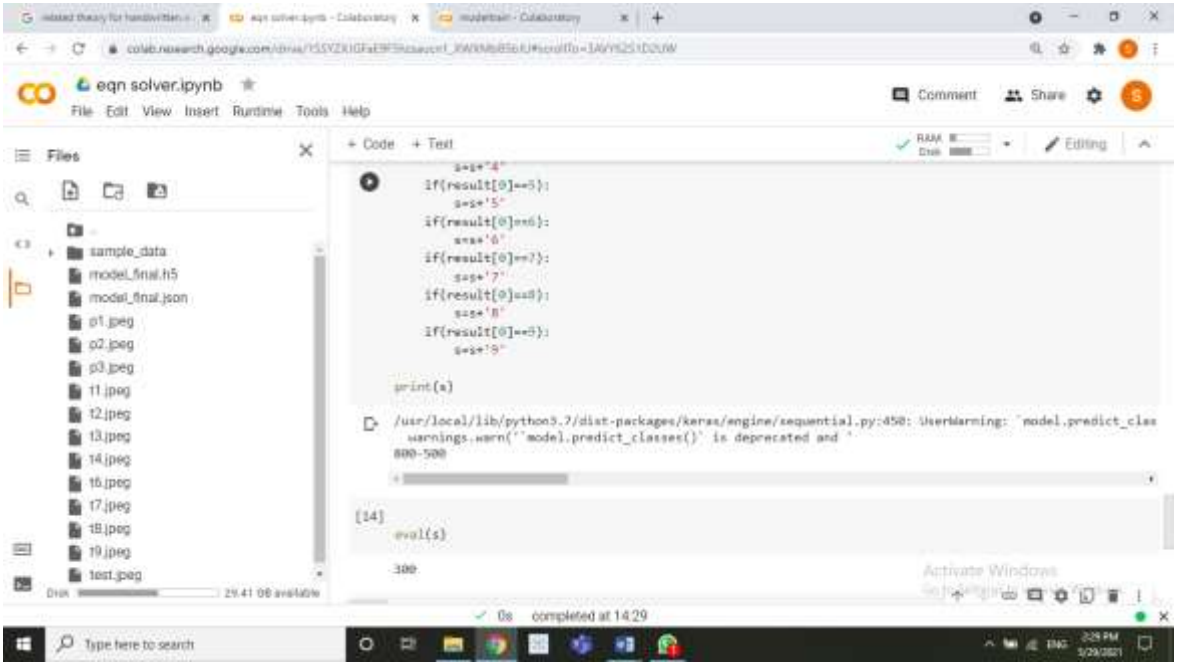
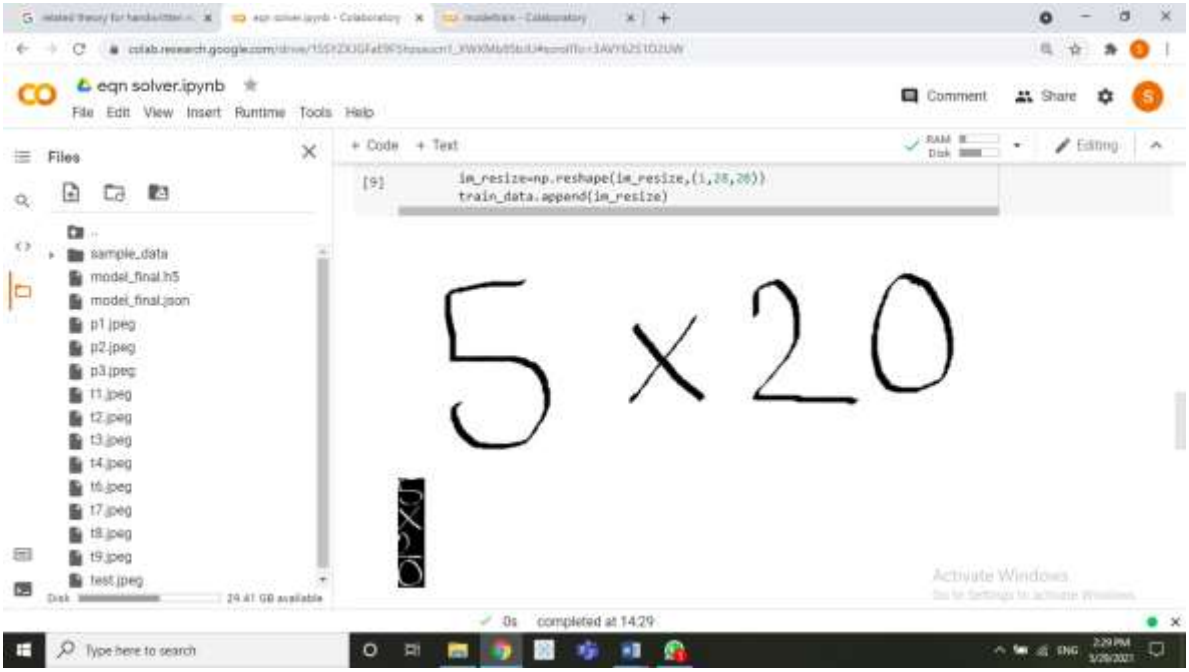


Fig 6.2.2 Sample Output (b)

When we perform a handwritten equation **800-500,** firstly the model predicts handwritten equation and after that we had used 'eval' function on the string to solve the equation and gives output as **300**.

3] 5* 20 =?



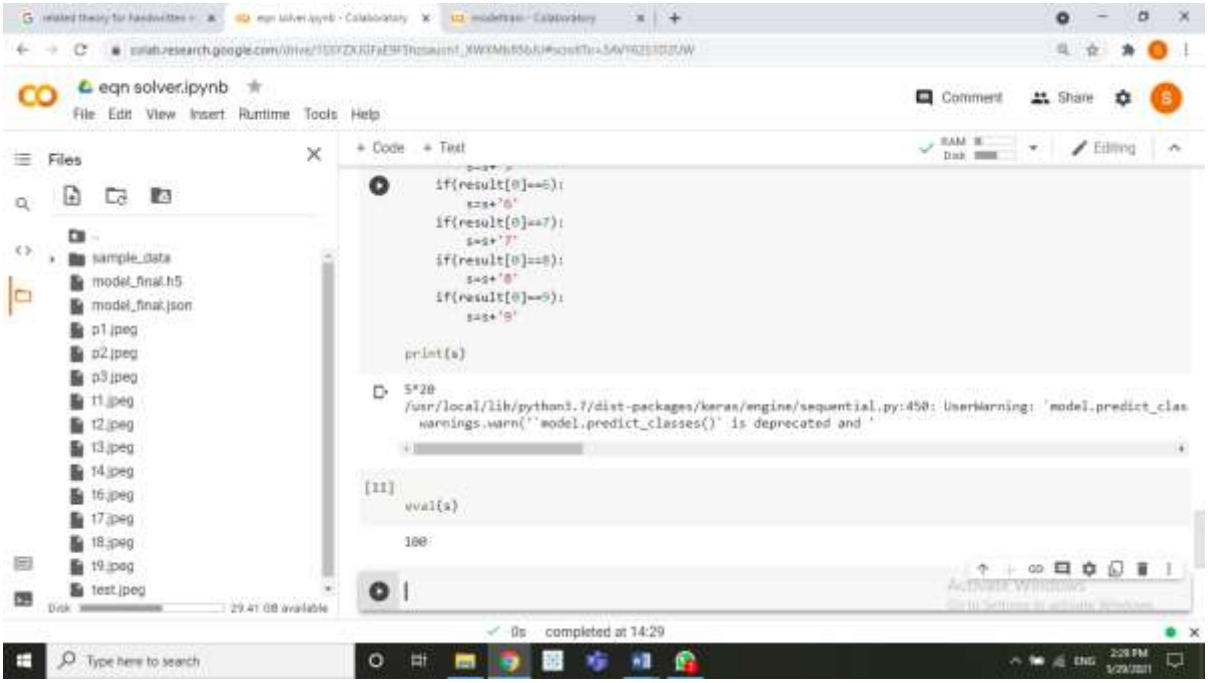
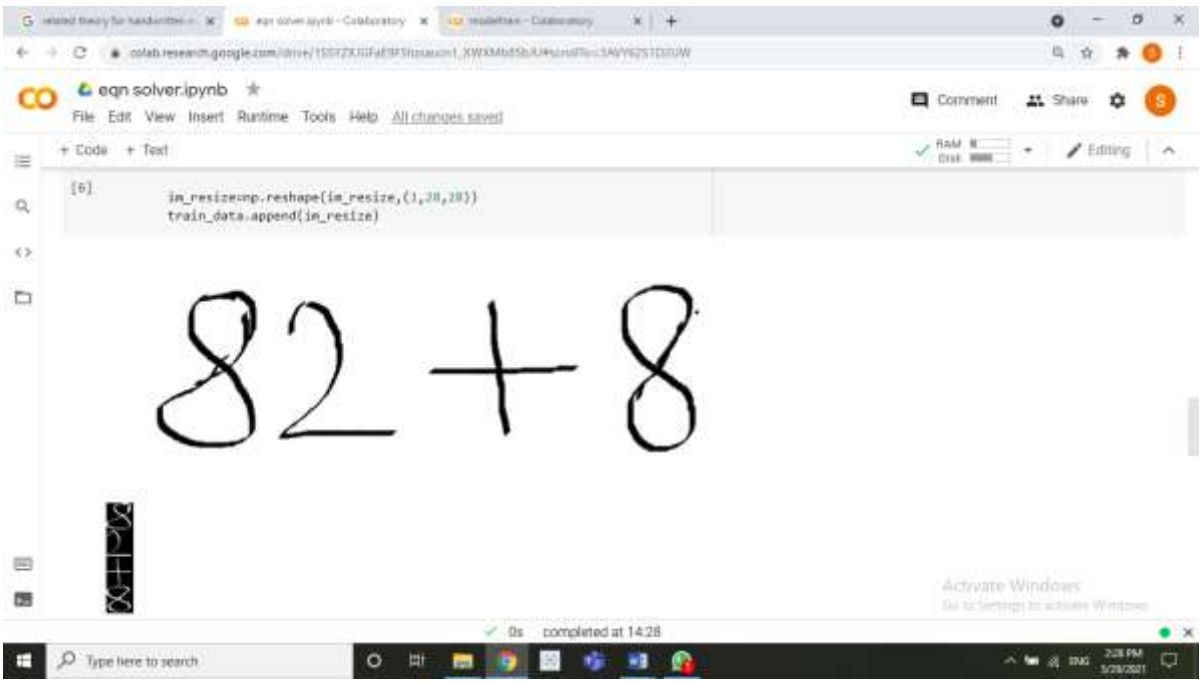


Fig 6.2.3 Sample Output (c)

When we perform a handwritten equation **5*20** ,firstly the model predicts handwritten equation and after that we had used 'eval' function on the string to solve the equation and gives output as **100**.

4] 82 + 8 = ?



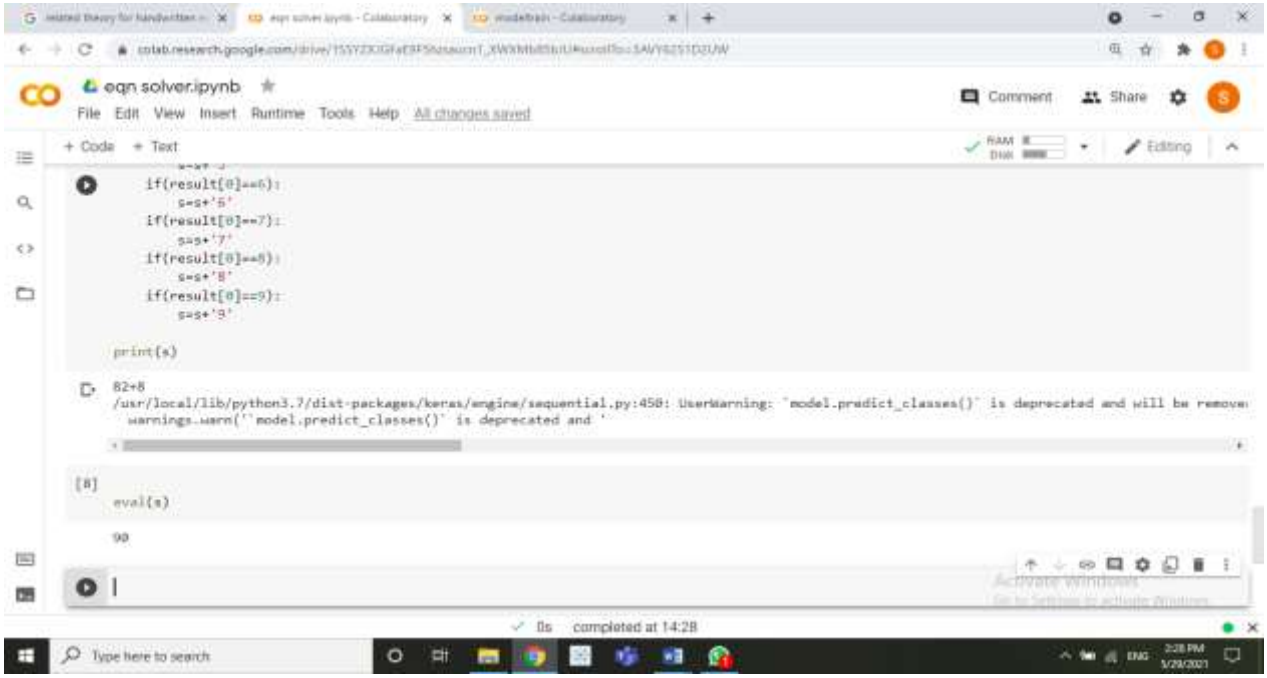
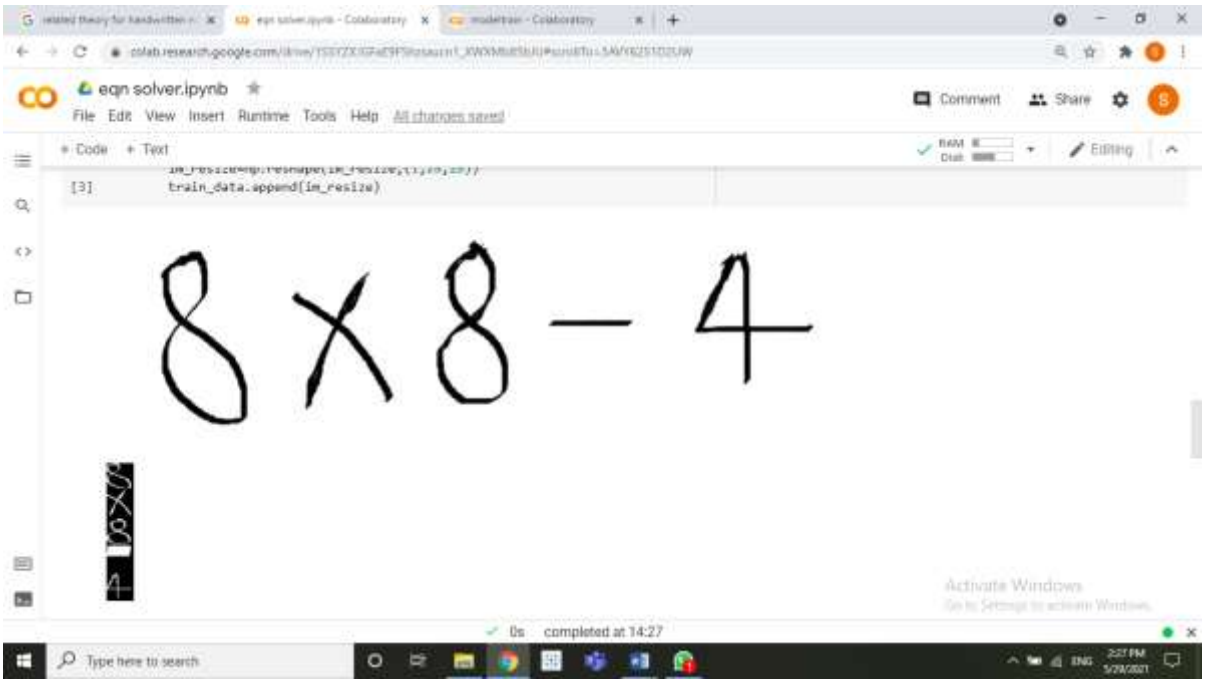


Fig 6.2.4 Sample Output (d)

When we perform a handwritten equation **82+8**,firstly the model predicts handwritten equation and after that we had used 'eval' function on the string to solve the equation and gives output as **90**.

5] 8*8-4 = ?



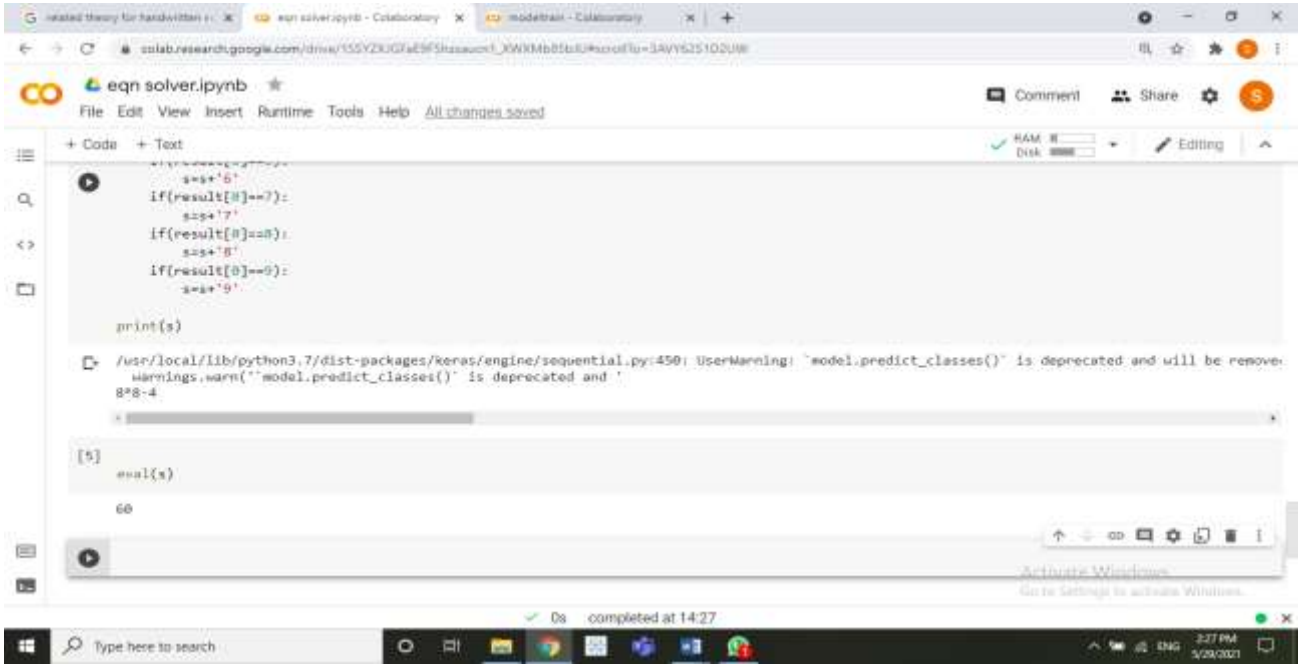
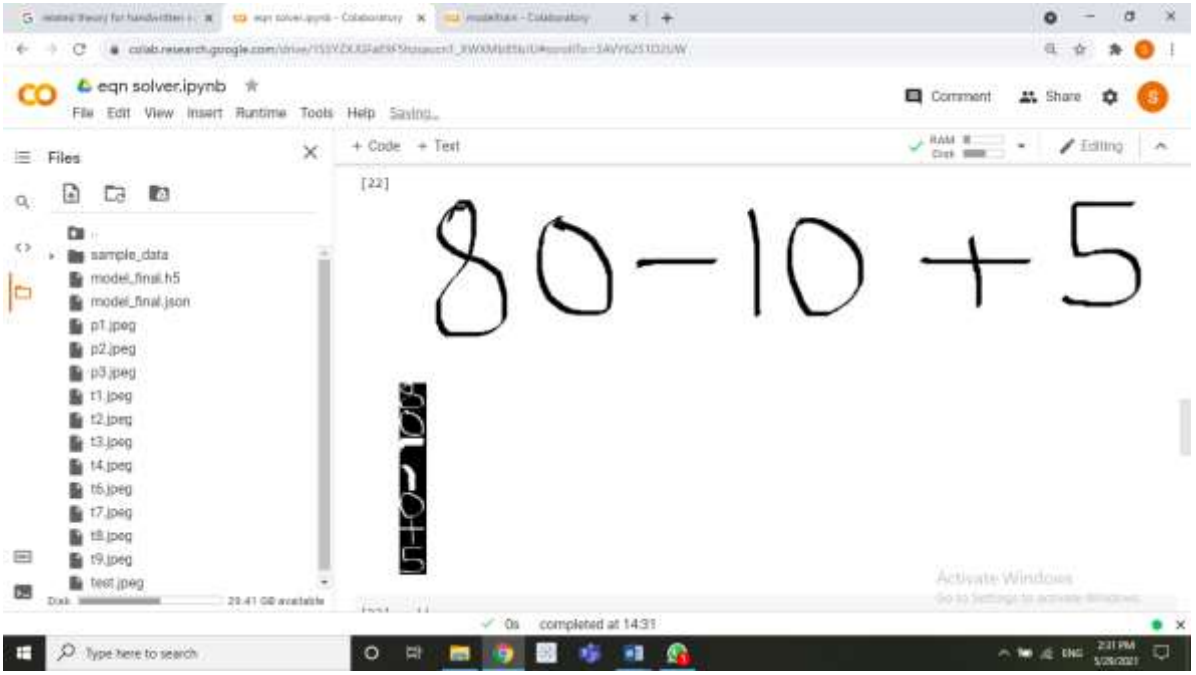


Fig 6.2.5 Sample Output (e)

When we perform a handwritten equation **8*8-4** ,firstly the model predicts handwritten equation and after that we had used 'eval' function on the string to solve the equation and gives output as **60**.

6] 80 -10 + 5 =?



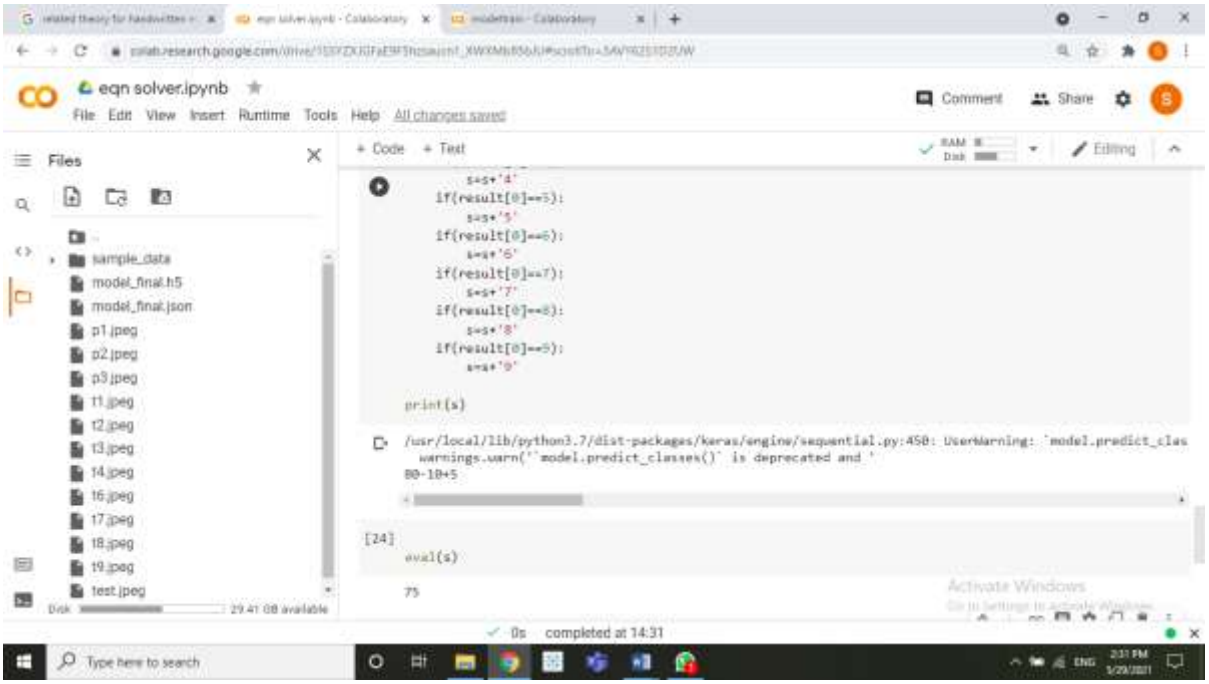
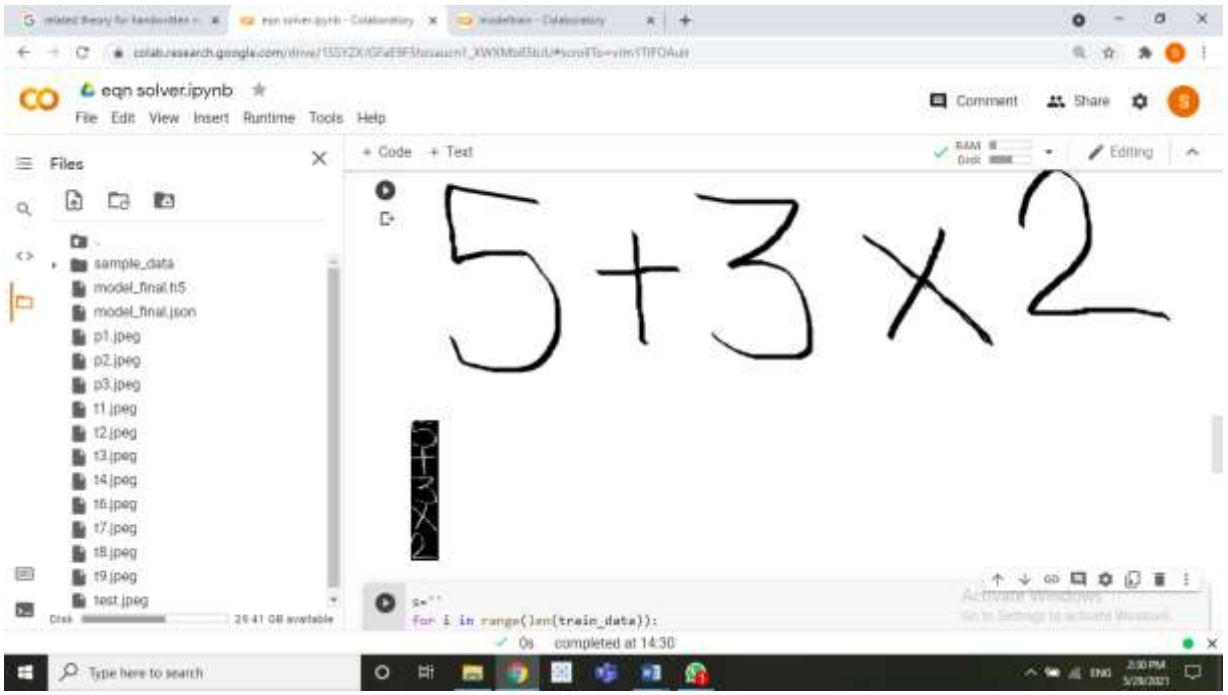


Fig 6.2.6 Sample Output (f)

When we perform a handwritten equation **80-10+5,** firstly the model predicts handwritten equation and after that we had used 'eval' function on the string to solve the equation and gives output as **75**.

7] 5+3*2=?



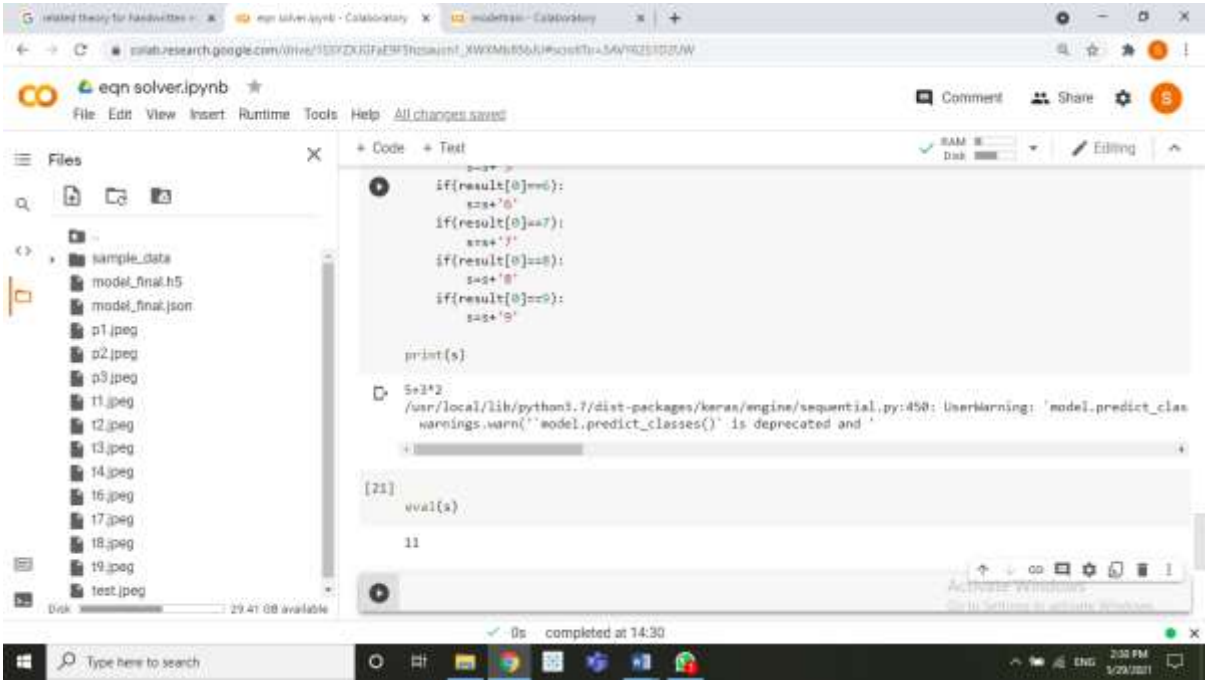


Fig 6.2.7 Sample Output (g)

When we perform a handwritten equation **5+3*2** ,firstly the model predicts handwritten equation and after that we had used ‘eval’ function on the string to solve the equation and gives output as **11**.

7. CONCLUSION AND FUTURE SCOPE

7.1. Conclusion

Handwritten mathematical equation solver can be used in every field of our day to day life. With the widespread usage of touch screen based devices and internet one can do their studies and works from remote areas. By using an equation solver such persons can save a lot of time and energy. In this paper we mainly focused on recognizing handwritten mathematical equations. We considered a simple 2 variable equations and also complex 2 or more variable equations for the recognition. Connected component which has very high success rate is used for character segmentation [5]. Improved version of connected component is used for the symbol like '*', '+', '-' detection which is a single symbol combined with two distinct connected component. Feature extraction is the most complicated part of classification. Convolutional Neural Network the most powerful classification model is used in the classification part[5]. Once successful recognition of the equation in any combination, we further process the detected equation for finding the solution of the equation. After that we apply an string operation algorithm for finding the value of a, b, c of each equations in the form of like a+b*c=? Finally we obtained a state of art performance in the detection phases and also in the solution phases.

7.2. Future Scope

This project can also be implemented in other complex projects such as online calculators which will be able to just take the image of the equation and compute the solution without additional human input and intelligence. This has number of advantages which mainly lead to saving of time which will certainly be helpful for many fields. Using the high computation power of current computers and the result of this project even complex solutions of equations can be solved in matter of seconds. In case of long and complex mathematical equations, the recognized mathematical equation may be displayed to the user for cross verification. If the input equation is same as the recognized equation then the system may proceed for solving.

REFERENCES

[1] Mohini Lokhande, Kirti Murudkar, Arbaaz Nadaf, Niyati Shah's "Handwritten Math Problem Solver Using Convolutional Neural Network", International Journal of Advanced Research In Computer and Communication Engineering (IJARCCE)(2012).

[2] Sanjay S. Gharde, P. Baviskar, K. P. Adhiya's "Identification of Handwritten Simple Mathematical Equation Based on SVM and Projection Histogram"International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-2, May 2013

[3] Ronald Clark, Quik Kung, A. V. Wyk's "System for the Recognition of Online Handwritten Mathematical Expressions", Eurocon 2013.

[4] Anmol Adhikari, Kunal Jain, Sunil Ray, Aishwarya Singh's "Handwritten Digit Recognition using CNN" , Networks' cs231n project report stanford 2015.

[5] Hossain, Md & Naznin, Feroza & Joarder, Y. A. & Islam, Md Zahidul & Uddin, Md. (2018). Recognition and Solution for Handwritten Equation Using Convolutional Neural Network. 250-255. 10.1109/ICIEV.2018.8640991. Huimin Wu, Research School of Computer Science, The Australia National University, Canberra @u6342493@anu.edu.auCatherine "CNN Based Recognition of Handwritten Digits"(2018).

[6] Giang Son Tran, C. Huynh, Thanh-Sach Le, T. Phan, Khanh-Ngoc Bui's "Handwritten Mathematical Expression Recognition Using Convolutional Neural Network", 2018 3rdInternational Conference on Control, Robotics and Cybernetics (CRC).

[7] Shifat Nayme Shuvo, Fuad Hasan, S. A. Hossain, Sheikh Abujar's "Handwritten Polynomial Equation Recognition and Simplification Using Convolutional Neural Network" 2020 11thInternational Conference on Computing, Communication and Networking Technologies (ICCCNT)

[8] A. Choudhary, Savita Ahlawat, Harsh Gupta, Anirudh Bhandari, Ankur Dhall's "Offline Handwritten Mathematical Equation Evaluator Using Convolutional Neural Network",Published in 2021.