A Project-1 Report
on

HANDWRITTEN EQUATION SOLVER USING CNN

Submitted to
The Department of Computer Science and Engineering

In partial fulfilment of the academic requirements of
Jawaharlal Nehru Technological University
For

The award of the degree of

Bachelor of Technology
in
Computer Science and Engineering
(2018 – 2022)

By

SHREYA MALRAJU
(18311A0599)

Under the Guidance of
Mr. R.Arun Kumar
Assistant Professor

Sreenidhi Institute of Science and Technology

(An Autonomous Institution)
Yamnampet, Ghatkesar, R.R. District, Hyderabad - 501301

Department of Computer Science and Engineering

Sreenidhi Institute of Science and Technology

CERTIFICATE

This is to certify that this Project-1 report on "Handwritten Equation Solver Using CNN", submitted by SHREYA MALRAJU (18311A0599) in the year 2021 in-partial fulfillment of the academic requirements of Jawaharlal Nehru Technological University for the award of the degree of Bachelor of Technology in Computer Science and Engineering, is a bona-fide work that has been carried out by them as part of their **Project-1 during Fourth Year First Semester**, under our guidance. This report has not been submitted to any other institute or university for the award of any degree.

Internal guide	**Project Co-ordinator**	**Head of Department**
Mr.R.Arun Kumar	Mrs. P. Vasavi	Dr. Aruna Varanasi
Assistant Professor	Assistant Professor	Professor & HOD
Department of CSE	Department of CSE	Department of CSE

External Examiner

Date:-

DECLARATION

I **SHREYA MALRAJU (18311A0599),** student of **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY, YAMNAMPET, GHATKESAR,** studying IVth Year Ist Semester, **COMPUTER SCIENCE AND ENGINEERING** solemnly declare that the Project-1 work, titled **"Handwritten Equation Solver Using CNN"** is submitted to **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY** for partial fulfillment for the award of degree of Bachelor of technology in **COMPUTER SCIENCE AND ENGINEERING**.

It is declared to the best of our knowledge that the work reported does not form part of any dissertation submitted to any other University or Institute for award of any degree.

ACKNOWLEDGEMENT

I would like to express my gratitude to all the people behind the screen who helped me to transform an idea into a real application.

I would like to express my heart-felt gratitude to my parents without whom I would not have been privileged to achieve and fulfill my dreams. I am grateful to our principal, **Dr. T. CH. SIVA REDDY,** who most ably run the institution and has had the major hand in enabling me to do my project.

I profoundly thank **Dr. ARUNA VARANASI**, Head of the Department of Computer Science & Engineering who has been an excellent guide and a great source of inspiration to my work.

I would like to thank our Coordinator & my internal guide for the Project-1 **Mrs. PALLEPATI VASAVI** for her constant guidelines, encouragement and support in carrying out my project on time at college.

The satisfaction and euphoria that accompany the successful completion of the task would be great but incomplete without the mention of the people who made it possible with their constant guidance and encouragement crowns all the efforts with success. In this context, I would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

SHREYA MALRAJU 18311A0599

Abstract

During the second semester of 3rd year, I was a part of the project Handwritten Digit Recognition System which classifies the handwritten numbers into its respective class ranging from 0-9.This semester, as an extension to the previous project, I am in a process of developing a system which solves Handwritten Arithmetic Expressions. Methods of CNN is used to train the data , which contains numbers from 0-9 and arithmetic operators (such as + , - , *, /) hence classification of elements is done into 13 classes. Robust handwritten character recognition is a tricky job in the area of image processing. Among all the problem handwritten mathematical expression recognition is one of the complicated issue in the area of computer vision research. Segmentation and classification of specific character makes the task more difficult. In this paper a group of handwritten quadratic equation as well as a single quadratic equation are considered to recognize and make a solution for those equation. Horizontal compact projection analysis and combined connected component analysis methods are used for segmentation. For classification of specific character we apply Convolutional Neural Network. Each of the correct detection, character string operation is used for the solution of the equation. Finally the experimental results shows the great effectiveness of our proposed system.

	LIST OF FIGURES		
S.No.	**Fig No**	**Title of Figure**	**Page No.**
1	1	Figure 1: Dataset	04
2	2	Figure 2: List of layers	05
3	3	Figure 3: Testing image	06
4	4	Figure 4: Segmentation of testing image	07
5	5	Figure 5: Changing dimensions	07
6	6	Figure 6: Equal dimensions of all elements	08
7	7	Figure 7: Architecture Diagram	11
8	8	Figure 8: Use Case Diagram	12
9	9	Figure 9: Class Diagram	13
10	10	Figure 10: Sequence Diagram	14
11	11	Figure 11: Activity Diagram	15
12	12	Figure 12: Processed Testing Image	26
13	13	Figure 13: Segmentation of Testing Image	26
14	14	Figure 14: Pre-processed Testing Image	27
15	15	Figure 15:Training Data	27
16	16	Figure 16:Predicting the value	28
17	17	Figure 17:Execution	45
18	18	Figure 18:Execution	45
19	19	Figure 19:Execution	46
20	20	Figure 20:Execution	46
21	21	Figure 21:Execution	47
22	22	Figure 22:Execution	47
23	23	Figure 23:Execution	48
24	24	Figure 24:Execution	48
25	25	Figure 25:Execution	49
26	26	Figure 26:Execution	49
27	27	Figure 27:Execution	50

	LIST OF TABLES		
S.No	**Table No**	**Title of Table**	**Page No**
1	1	Table 1. Existing System	03
2	2	Table 2: Test cases	34

1. INTRODUCTION:

1.1 PROBLEM DEFINITION:

Humans will see and visually sense the planet around them by exploitation their eyes and brain. pc vision works on sanctionative computers to envision and method pictures within the same manner that human vision will. many algorithms are developed in pc vision to acknowledge pictures.

The model which will be ready to establish and verify the written digits and arithmetic operators from its image with higher accuracy and cipher the worth of that expression once determination. exploitation the ideas of Convolutional Neural Network and extended MNIST dataset, this drawback may be resolved.

We can extend this model to acknowledge combination of numbers and digits from 2 or additional polynomial equations and look at determination them. The features are then extracted from processed images. Now, System splits the data into train and test data. Then System sends the data to the ML model, now the secondary actor User, trains the model or fits the model and also tests the model and the accuracy results are sent back to the System.

The goal of my work is going to be to form a model which will be ready to establish and verify the written digits and arithmetic operators from its image with higher accuracy and cipher the worth of that expression once determination. exploitation the ideas of Convolutional Neural Network and extended MNIST dataset, this drawback may be resolved. We can extend this

model to acknowledge combination of numbers and digits from 2 or additional polynomial equations and look at determination them.Firstly, System collects the images and makes the dataset. Then the System pre-processes the image which includes image resizing, so that all the images are in the same size.

1.2 OBJECTIVE OF PROJECT:

The aim of the project is to :

1. Create a CNN model that can identify numbers and arithmetic operators from an image.
2. Deduce the mathematical expression and calculate the value of expression
3. If predicted wrong, train the model with the correct value

The project aims to help us understand how CNN models can be used to make simple tools like the calculator. With the advancement in technology, machine learning and deep learning are playing a crucial role in present times. Now, machine learning and deep learning techniques are being employed in handwriting recognition, robotics, artificial intelligence, and many more fields. Developing such system requires training our machines with data, making it capable to learn and make required prediction. The model gives the perfect answers with an accuracy of 99 percent.

What makes Machine learning algorithms so powerful is its ability to learn from its mistakes. In order to do that the model needs to be retrained with correct information. If there is a wrong prediction, the code asks for the correct expression.It then compares the predicted with the correct expression and identifies the elements that are wrongly predicted. The code then trains the model from its second hidden layer till its output layer using the correct data (first hidden layer is not trained).

2. LITERATURE SURVEY:

2.1 RELATED WORK:

Convolutional Neural Networks are a subclass of Deep Learning algorithms mainly used for analyzing visual imagery.Lots of training data, increase in the computational powers and latest deep learning algorithms have given the way for CNNs to perform complex operations.

It all started in 1958 when David H. Hubel and Torsten Wiesel performed a series of experiements to understand the structure of the visual cortex (for which the won the Nobel Prize in 1981). The authors found that some neurons had larger receptive fields and would react to complex patterns that were a combination of lower-level patterns detected by other neurons.

These observations led to the idea that the higher level neurons are based on the outputs of neighbouring lower-level neurons.The goal of my work is going to be to form a model which will be ready to establish and verify the written digits and arithmetic operators from its image with higher accuracy and cipher the worth of that expression once determination. exploitation the ideas of Convolutional Neural Network and extended MNIST dataset, this drawback may be resolved. We can extend this model to acknowledge combination of numbers and digits from 2 or additional polynomial equations and look at determination them.Firstly, System collects the images and makes the dataset. Then the System pre-processes the image which includes image resizing, so that all the images are in the same size.

The features are then extracted from processed images. Now, System splits the data into train and test data. Then System sends the data to the ML model, now the secondary actor User, trains the model or fits the model and also tests the model and the accuracy results are sent back to the System.

Now on comparing the results sent by the ML model, the System comes to a conclusion which is the accuracy and the value of the given expression and displays the results as output.

2.2 EXISTING SYSTEM:

SNO	YEAR	NAME	AUTHOR	METHODS USED	ACCURACY
1	2018	Maximum Margin Clustering Made Practical	Kai Zhang Ivor W. Tsang James T. Kwok	Maximum Margin Clustering	74%
2	2019	A paradigm for handwriting-based intelligent tutors.	Lisa Anthony Jie Yang Kenneth R. Koedinger	Optical Character Recognition	78%
3	2020	ANN-Based Handwritten Digit Recognition and Equation Solver	Amarjit Malhotra Megha Gupta Rishabh Yadav	Artificial Neural Networks	89%

Table 1 : Existing systems

2.3 PROPOSED SYSTEM:

- DATASET

It is an extension of the MNIST dataset and contains 85,709 images of Arabic numerals (0 to 9) and mathematical operators (+, - , * and /). Each image is of size 28 X 28 pixels. In order to be easily encoded, the mathematical operators are numerically labelled as follows:

⇒ 10 represents "/" (division)

⇒ 11 represents "+" (addition)

⇒ 12 represents "-" (subtraction)

⇒ 13 represents "/" (multiplication)

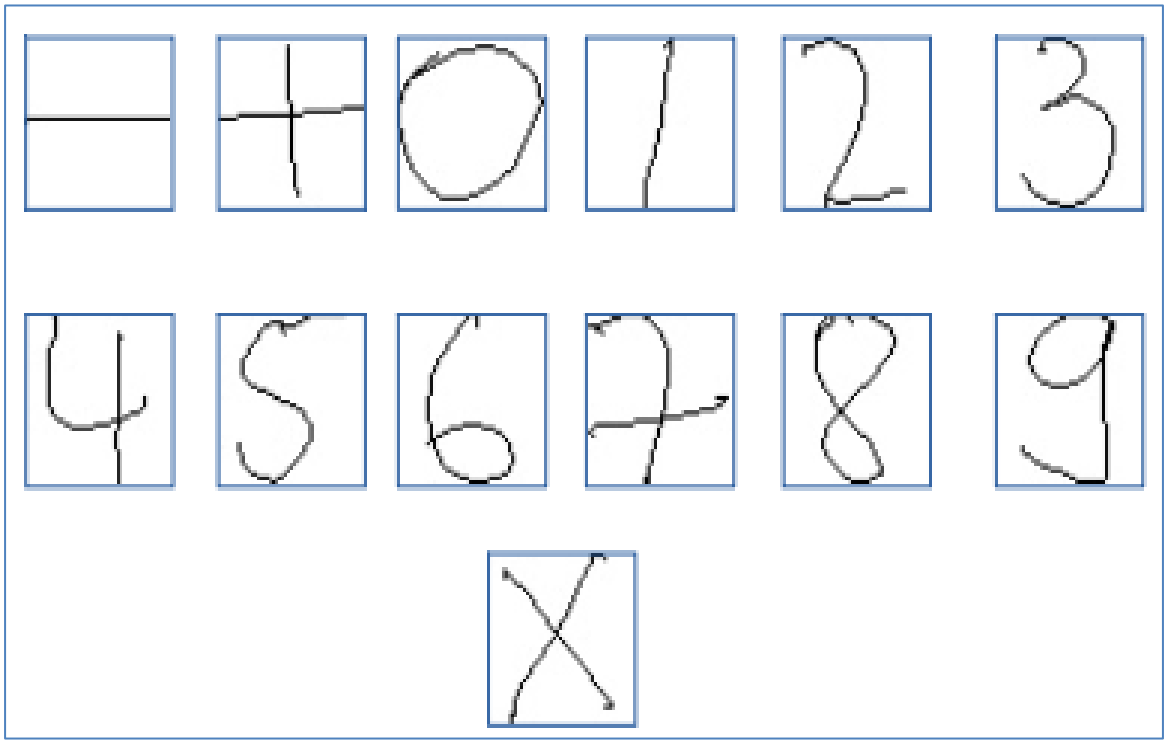


Fig 1: Dataset

- PRE-PROCESSING

The dataset is loaded as a Pandas dataframe and therefore the labels (y) area unit area unit from the dataset (X).

The dataset is grayscale normalized by dividing the dataset by 255 to cut back the effect of illumination. Then, the dataset is reshaped to work the 4D Tensor.

Next, the labels area unit area unit from class vectors (14 class vectors) to binary class matrices. Then, the data is divided into training and validation ie., 90 percent and 10 percent respectively. Now, training can be done on the dataset.

- BUILDING AND TRAINING CNN MODEL

The CNN model is built by using sequential model class from Keras. The table below describes the list of layers that are passed to make the model.

Layer Index	Type	Feature Maps	Size	Kernel Size	Stride	Padding	Parameters	Activation
-	Input	-	28 X 28	-	-	-	-	-
0	Convolution	32	28 X 28	5 X 5	1 X 1	same	832	ReLU
1	Convolution	32	28 X 28	5 X 5	1 X 1	same	25632	ReLU
2	Max pooling	32	14 X 14	2 X 2	1 X 1	-	0	-
3	Dropout (25%)	-	14 X 14	-	-	-	0	-
4	Convolution	64	14 X 14	3 X 3	1 X 1	same	18496	ReLU
5	Convolution	64	14 X 14	3 X 3	1 X 1	same	36928	ReLU
6	Max pooling	64	7 X 7	2 X 2	1 X 1	-	0	-
7	Dropout (25%)	-	7 X 7	-	-	-	0	-
8	Flatten	-	3136	-	-	-	0	-
9	Dense	-	256	-	-	-	803072	-
10	Dropout (25%)	-	256	-	-	-	0	-
11	Dense	-	14	-	-	-	3598	Softmax

Fig 2: List of Layers

There are 3 layers used in the architecture

1. Input and First Hidden layer
2. Second Hidden Layer
3. Fully Connected Layer and output layer

First Hidden layer and Input Layer

1. A set of two Convolutional layers to identify the low level image patterns.
2. The Max Pooling layer to down sample the filters to reduce memory usage , computational load and number of parameters.
3. A Dropout is used as regularization method. It randomly ignore quarter of the nodes during each training iteration.

Second Hidden Layer

1. A set of two Convolutional layers to identify complex patterns that are a mixture of patterns, detected by the first layer.
2. Max Pooling layer for down sampling.
3. Dropout as regularization method.

Output and Fully Connected layer

1. Flatten layer is used to chane the final feature maps into a 1D vector.
2. A fully-connected layer that acts as an ANN.
3. Dropout as regularization method.
4. A fully-connected layer with Softmax activation function as the output layer. The class with highest probability is taken for the model prediction. Softmax is used because the elements of the output are transformed into net Output distribution of each class.

Once the layers are added, the training parameters are defined.

To prevent overfitting, the dataset is enlarged by artificially modifying the original dataset. This process is called Data Augmentation and is executed using Keras Image Data Generator. The model is then trained with 5 epochs and each batch size of 86. The model reached an accuracy of 99.04 % after the 5th epoch.

- MODEL PREDICTIONS

First, take the handwritten expression in the form of an image and give it as the input to the model.

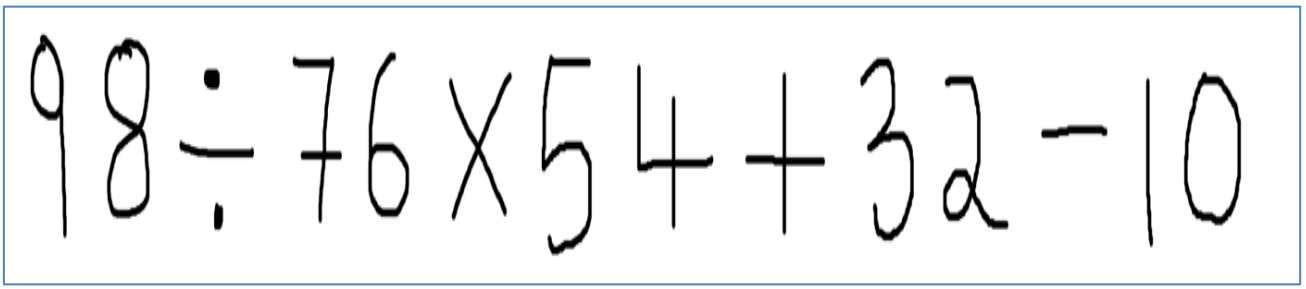


Fig 3: Testing Image

To analyze the mathematical expression, break down the image into individual elements that the model can identify.

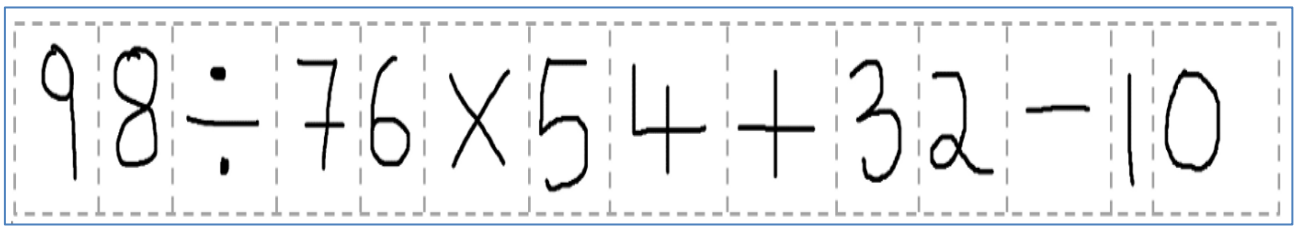


Fig 4: Segmentation of testing image

To do that, first the image file is loaded in grayscale. Later, the image is reshaped to a height of 28 px. Maintain the aspect ratio.

The image is then modified into to a Numpy array. The background is changed to black by subtracting 255 and the image is normalized by dividing with 255.

After that, the image array is split into individual element arrays.

To identify the element , we must have the image size of 28px*28px. The height is 28px. But, in the process of splitting, the width is not 28px.

Hence, after splitting, width of element is made to 28 px. by adding zero value columns to the element arrays.

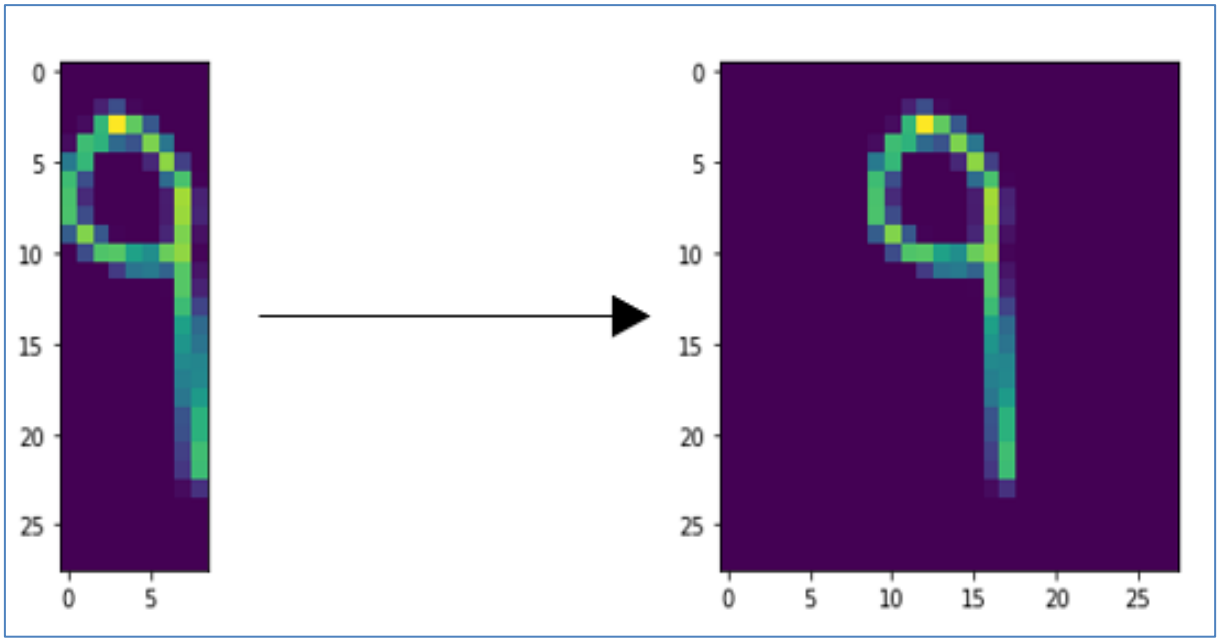


Fig 5: Changing the dimensions

This process is repeated for all parts of the expression until all are 28px *28px.

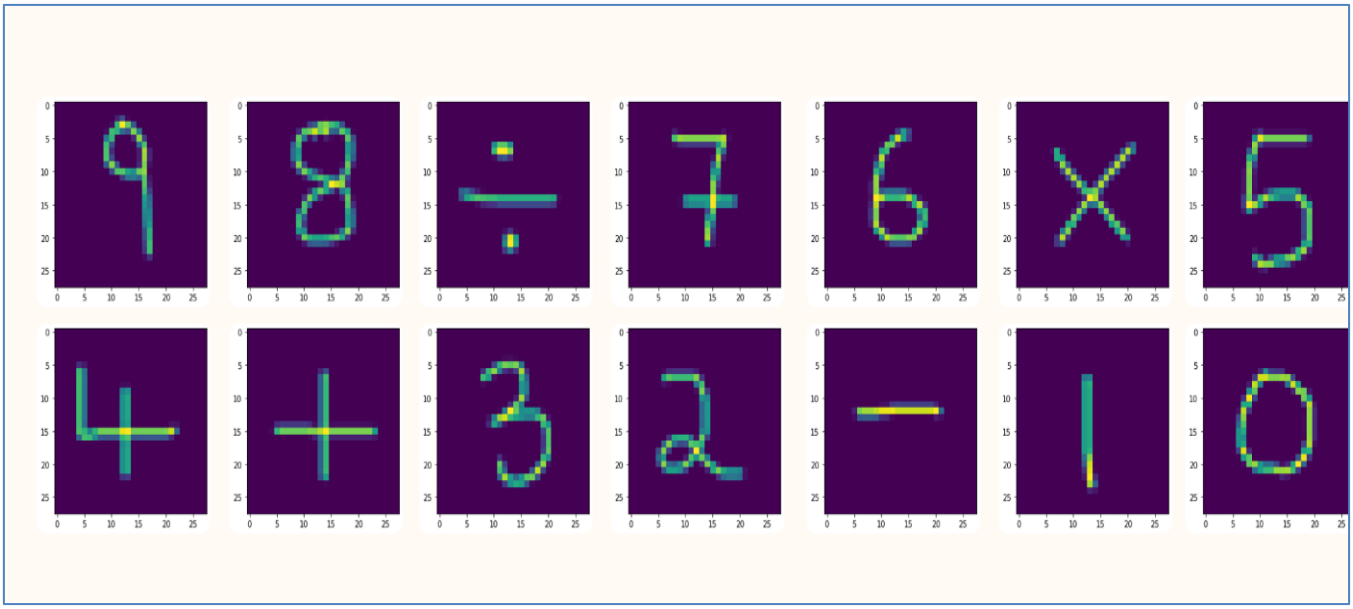


Fig 6: Equal dimensions of all elements

After the elements list is converted back into an array of shape , now these elements array is ready for the model to predict values.

The array of elements is passed to the model for prediction. The model returns the probabilities of all the number classes i.e., 14 different classes (0-9 digits, +, -, * , /) .The class with the highest probability is selected.

- CREATING A CALCULATOR

To generate the mathematical equation, the number classes (0 to 9) are regrouped to form the multi digit numbers and 10, 11, 12 and 13 classes are converted to “/”, “+”, “-” and “*” operators.

Lastly, the numbers and operators are joined together to form a string representing the mathematical expression.

The eval() method of python is used on the string to calculate the value of the given input image.

3.SYSTEM ANALYSIS:

This System Analysis is strongly related to the analysis of requirements. It's also a formal inquiry conducted to assist someone in identifying a better course of action and making a better conclusion than he would have chosen otherwise.

This process entails breaking down the system into individual components to examine the situation, breaking down what needs to be built, analysing project goals and seeking to engage users in order to identify specific requirements.

3.1 FUNCTIONAL REQUIREMENT SPECIFICATIONS:

1.Admin Module:

The admin is responsible for creating a dataset. Preprocessing the dataset images. Admin converts the images into machine readable format. It splits the data into test and train data, where this data is sent to the Model module. Then the Model module sends the results by which the admin module compares the results and chooses the best algorithm. It also predicts the input image sent by user.

2.Model Module:

This module maintains all the information of Machine Learning Algorithms. It trains and tests the data sent by the admin and sends the results to the admin. These results are based on the performance of each Machine Learning Algorithm.

3.Dataset Module:

The Dataset Module handles the data of images. This module adds or deletes the images from the dataset.

3.2 SOFTWARE REQUIREMENTS:

Operating System	**:**	Microsoft Windows 7 or above versions
Tools	**:**	Jupyter Notebook, PyCharm, IDLE, Spyder, Google Colab
Libraries	**:**	Skimage, sklearn, seaborn, cv2, numpy, scipy, pandas, matplotlib

3.3 HARDWARE REQUIREMENTS:

Processor	**:**	Intel Dual core Processor
RAM	**:**	>=4 GB
Hard Disk	**:**	>= 512 MB

4. SYSTEM DESIGN:

4.1 ARCHITECTURE:

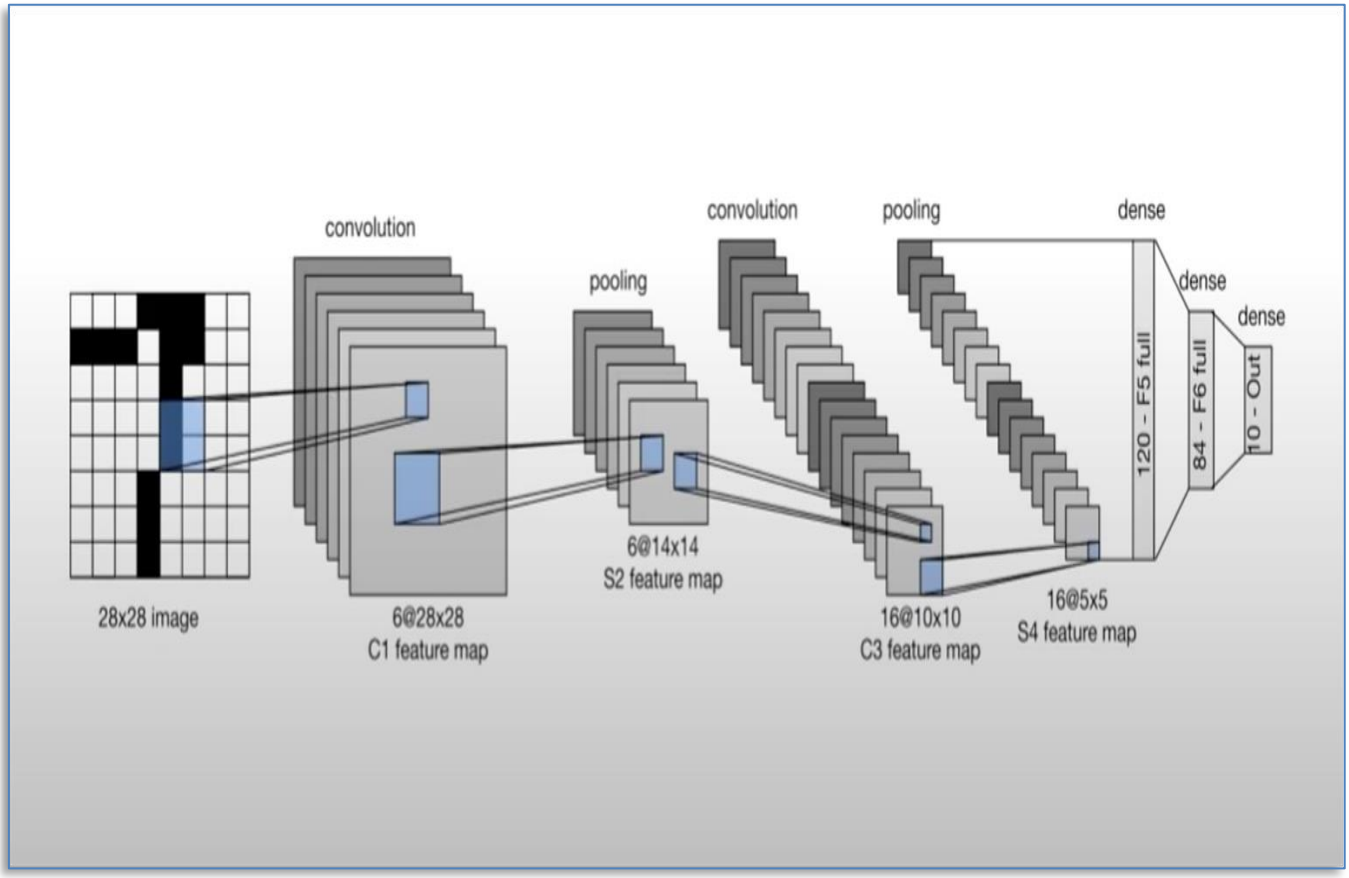


Fig 7: Architecture Diagram

First, the dataset is loaded which consists of images of numbers and operators, each of size 28*28px.For each image, training is done i.e., first the image enters the convolution layer, and features are extracted. After that, the selected features are sent to the first pooling layer, where size reduction happens.

There can any number of combination of Convolution and pooling layers. More the number, better the accuracy.

After the final pooling layer, there is the first dense layer, where the summation of all the pooling layers happen. The first dense layer has 120 classes, the second has 84.

The last dense layer must consist the total number of classes equal to the number of different images, which in this case is 13 i.e., 0-9 digits and 4 arithmetic operators such as +, -, *, /.

In the final dense layer, the image is classified into that class where the features are with high probability. In this way, all the images in the dataset are classified into their respective classes and hence now the model is ready for predicting the output of the given arithmetic equation.

4.2 UML DIAGRAMS

4.2.1 USE CASE DIAGRAM:

Fig 8: Use case diagram

A use case determines the behavior of system and it is a description of a set of sequence of actions that a system undergoes. Use case consists of actors and use cases and the connection between use case and actor is association.

An association between an actor and a use case shows that the actor and the use case exchange information with each other. Here for this project, we have taken 2 actors, System and User.

System is the main actor so placed at left side and the User is a secondary actor hence, placed on the right side. Firstly, System collects the images and makes the dataset. Then the System preprocesses the image which includes image resizing, so that all the images are in the same size.

The features are then extracted from the images. Now, System splits the data into train and test data. Then System sends the data to the ML model, now the secondary actor User, trains the model or fits the model and also tests the model and the accuracy results are sent back to the System.

Now on comparing the results sent by the ML model, the System comes to a conclusion which is the accuracy and the value of the given expression and displays the results as output.

4.2.2 CLASS DIAGRAM

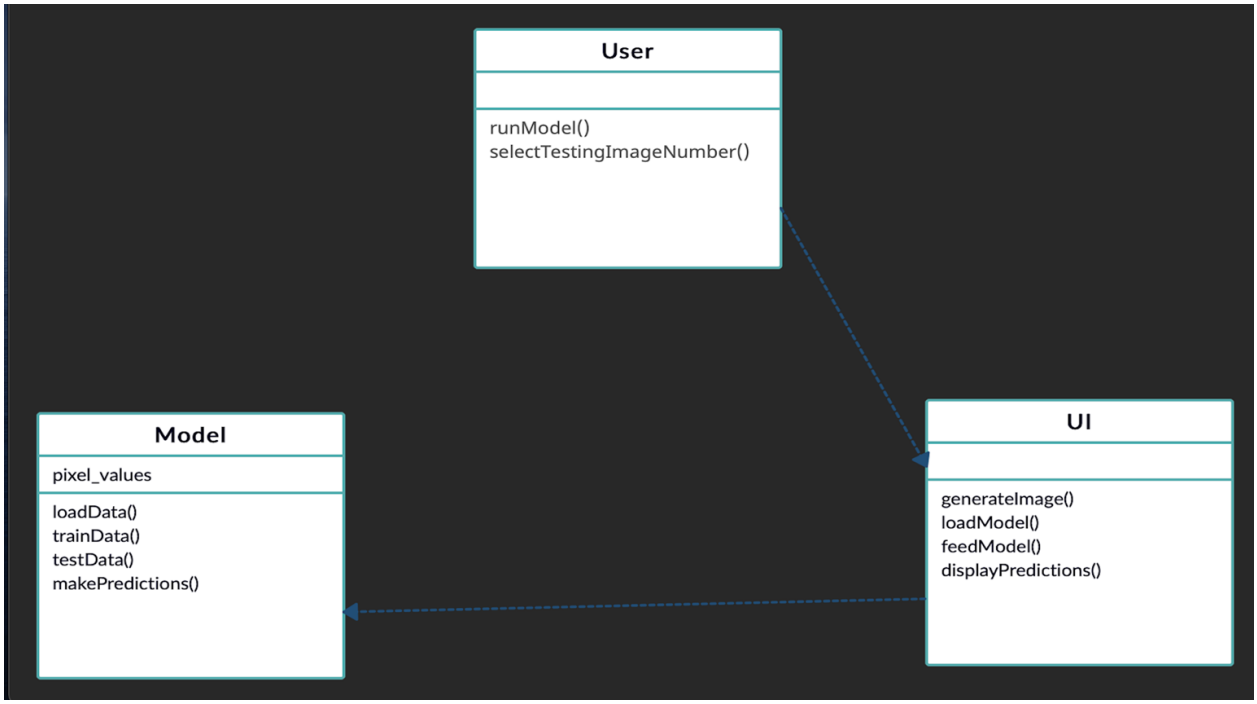


Fig 9: Class Diagram

We have used a total of 3 classes here. As we already know, each class contains attributes and operations. Coming to User class, It is the main class of the total system. Operations done by User are firstly making a dataset, after this function call we eventually go to UI class and generate images.

Then the images in the dataset are pre-processed and features are extracted from the obtained pre-processed images. Now the dimensions of the image are reduced by the User class and the data is split into train and test and the split data is then sent to ML model. UI class performs the operations like training the model and testing the model.
Then the results obtained by algorithm are stored and sent to the User class. Then the User class displays the results as the value of the output arithmetic expression.

4.2.3 SEQUENCE DIAGRAM:

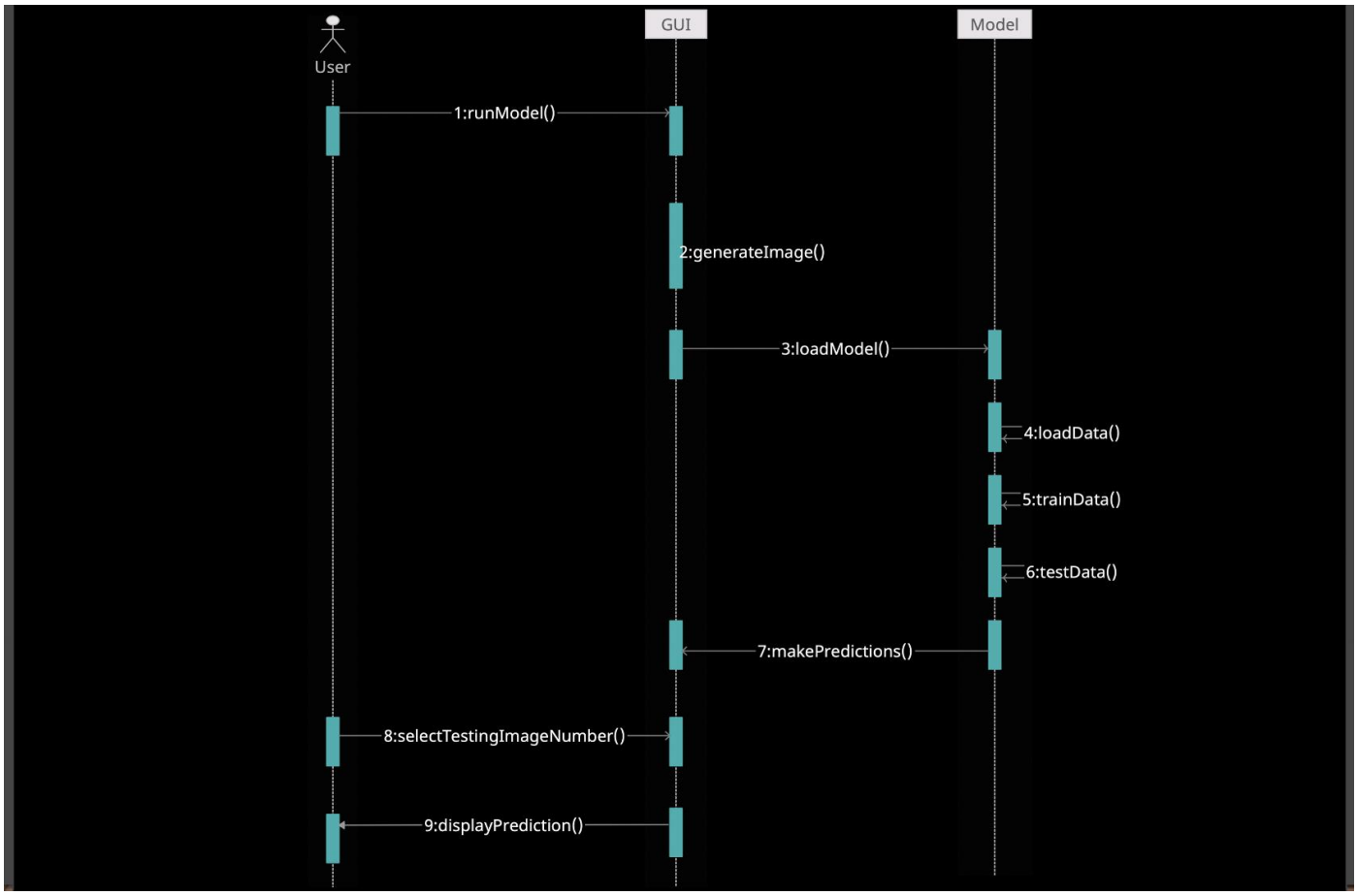


Fig 10: Sequence Diagram

A sequence diagram is an interaction diagram that emphasizes the time ordering of messages. sequence diagrams commonly contain

- Objects
- Links
- Messages

As we see here, we form a sequence diagram by first placing the objects that participate in the interaction at the top of the diagram, across the X axis. Typically, we place the object that initiates the interaction at the left, and increasingly more subordinate objects to the right.

We have used a total of 3 classes here. As we already know, each class contains attributes and operations. Coming to User class, It is the main class of the total system. Operations done by User are firstly making a dataset, after this function call we eventually go to UI class and generate images.

Then the images in the dataset are pre-processed and features are extracted from the obtained pre-processed images. Now the dimensions of the image are reduced by the User class and the data is split into train and test and the split data is then sent to ML model. UI class performs the operations like training the model and testing the model.

Then the results obtained by algorithm are stored and sent to the User class. Then the User class displays the results as the value of the output arithmetic expression.

4.2.4 ACTIVITY DIAGRAM:

Activity diagram is a flowchart to represent the flow from one activity toother. The activity diagram represents the dynamic aspects of the project.

The flow is drawn from one operation to other. This flow can be sequential, branched, or concurrent. They deal with all type of flow control by using fork, join, etc.

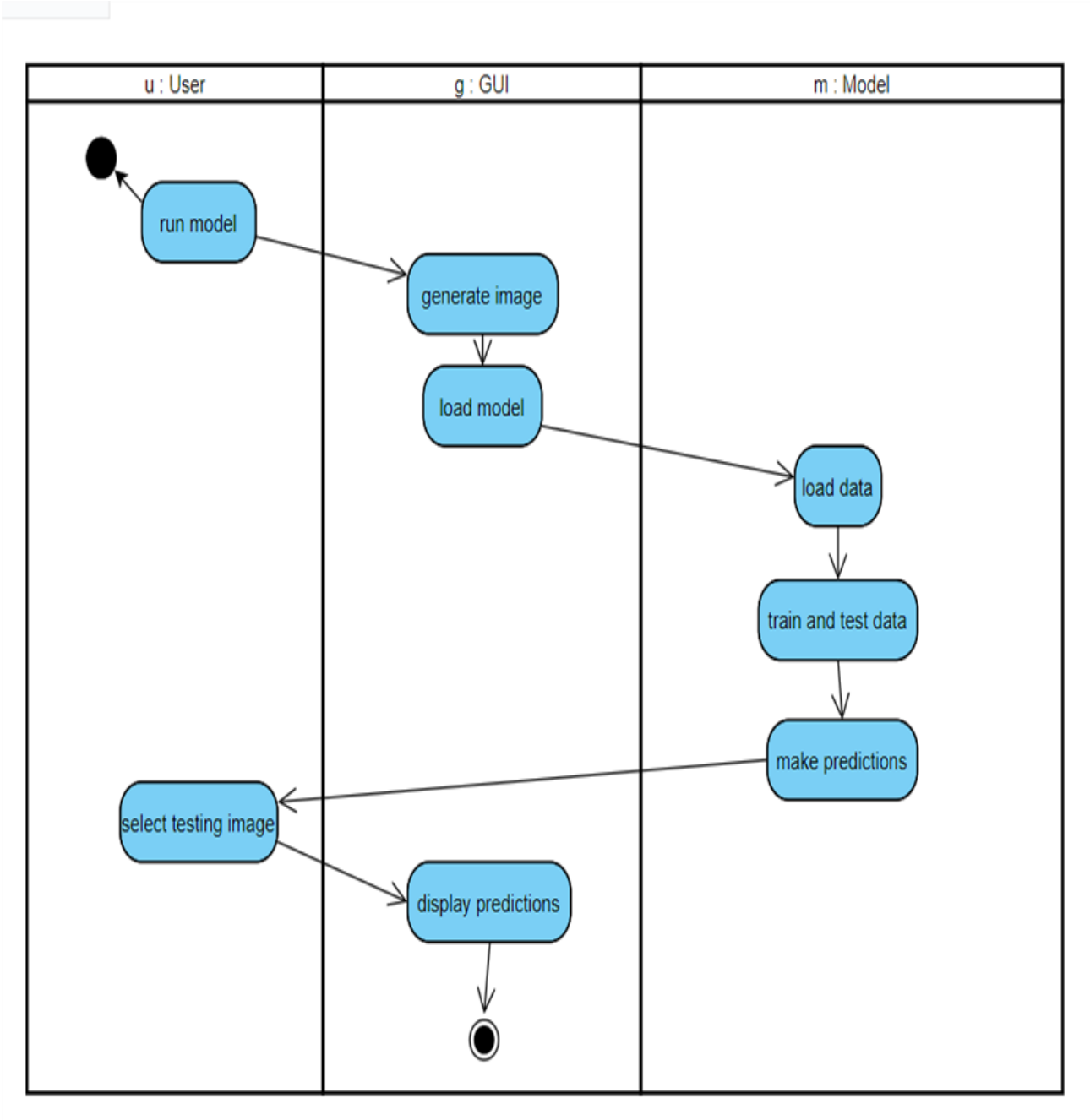


Fig 11: Activity Diagram

We have drawn a swimlane consisting of 3 objects namely user,GUI, model. Firstly user collects the data of images. Then the dataset is created by adding and deleting the images. Then the user pre-process the images. Next the user extracts the features from the pre-processed images and reduces the dimensions of the feature pool. Then the feature pool is split into train and test data.

Then the split data is sent to the ML model. The model object trains and tests the data over Machine learning algorithm. Then the results obtained by algorithm are stored and sent to the User class. Then the User class displays the results as the value of the output arithmetic expression.

5. IMPLEMENTATION AND RESULTS

5.1 LANGUAGE/ TECHNOLOGY USED:

The language and technology used is Python. Tool used is Google colaboratory

Google Colab is a product of Google. It is a notebook environment that runs fully in the cloud. Colab supports many high-level machine learning libraries which can be easily loaded into the notebook.
Google is involved in the AI research and development part. One of the AI frameworks was made by Google called TensorFlow and also a development tool called Collaboratory. TensorFlow is one of the most used open-source tools. Google Collaboratory was made free by Google to encourage AI community. An attractive feature is the use of GPU. Google Colab is popular because of the support of GPU.

Irrespective of all the hidden reasons, the introduction of Colab has really made AI researchers and developers' life easy for the development of machine learning applications.

The model which will be ready to establish and verify the written digits and arithmetic operators from its image with higher accuracy and cipher the worth of that expression once determination. exploitation the ideas of Convolutional Neural Network and extended MNIST dataset, this drawback may be resolved.

We can extend this model to acknowledge combination of numbers and digits from 2 or additional polynomial equations and look at determination them. The features are then extracted from processed images. Now, System splits the data into train and test data. Then System sends the data to the ML model, now the secondary actor User, trains the model or fits the model and also tests the model and the accuracy results are sent back to the System.

5.2 ALGORITHMS USED:

Convolutional Neural Networks are a subclass of Deep Learning algorithms mainly used for analyzing visual imagery. Lots of training data, increase in the computational powers and latest deep learning algorithms have given the way for CNNs to perform complex operations.

It all started in 1958 when David H. Hubel and Torsten Wiesel performed a series of experiements to understand the structure of the visual cortex (for which the won the Nobel Prize in 1981). The authors found that some neurons had larger receptive fields and would react to complex patterns that were a combination of lower-level patterns detected by other neurons. These observations led to the idea that the higher level neurons are based on the outputs of neighboring lower-level neurons.

This powerful visual architecture evolved over the years until, a paper by Yann LeCun et al. in 1998, formulated the famous *LeNet-5* architecture. In this paper, the concept of convolutional and pooling layers was introduced.

Today, many companies employ CNN for their applications. Few of them are:

- Tesla
- Google
- Facebook

5.3 CODE

```
#---1. Data set ---

import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
import seaborn as sns

np.random.seed(2)

'load the dataset'
dataset = pd.read_csv("...\\dataset.csv")

'creating label'
y = dataset["label"]

'dropping label'
X = dataset.drop(labels = ["label"], axis = 1)

'deleting dataset to reduce memory usage'
del dataset

'overview of dataset'
g = sns.countplot(y)
y.value_counts()

'Grayscale normalization to reduce the effect of illumination differences.'
X = X / 255.0

'reshaping the dataset to fit standard of a 4D tensor of shape [mini-batch size,
height = 28px, width = 28px, channels = 1 due to grayscale].'
```

```
X = X.values.reshape(-1,28,28,1)

'categorical conversion of label'
y = to_categorical(y, num_classes = 14)
```

```
'90% Training and 10% Validation split'
random_seed = 2
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size = 0.1 ,
random_state = random_seed, stratify = y)
```

#---2. Model---

```
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D
from keras.optimizers import RMSprop
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ReduceLROnPlateau
from tensorflow import keras

'creating the instance of the model'
model = Sequential()

'adding layers to the model'
#Layer: 1
model.add(Conv2D(filters = 32, kernel_size = (5,5), padding = "Same", activation
= "relu", input_shape = (28, 28, 1)))
model.add(Conv2D(filters = 32, kernel_size = (5,5), padding = "Same", activation
= "relu"))
model.add(MaxPool2D(pool_size = (2,2)))
model.add(Dropout(0.25))


#Layer: 2
model.add(Conv2D(filters = 64, kernel_size = (3,3), padding = "Same", activation
= "relu"))
model.add(Conv2D(filters = 64, kernel_size = (3,3), padding = "Same", activation
= "relu"))
model.add(MaxPool2D(pool_size = (2,2)))
model.add(Dropout(0.25))
```

```
#fully connected layer and output
model.add(Flatten())
model.add(Dense(256, activation = "relu"))
model.add(Dropout(0.25))
model.add(Dense(14, activation = "softmax"))



'Set the optimizer and annealer'
optimizer = RMSprop(lr = 0.001, rho = 0.9, epsilon = 1e-08, decay=0.0 )

model.compile(optimizer = optimizer, loss = "categorical_crossentropy", metrics =
["accuracy"])

learning_rate_reduction = ReduceLROnPlateau(monitor = "val_accuracy",
                                            patience = 3,
                                            verbose = 1,
                                            factor = 0.5,
                                            min_lr = 0.0001)
```

```
'data augmentation'
datagen = ImageDataGenerator(
        featurewise_center=False,  # set input mean to 0 over the dataset
        samplewise_center=False,  # set each sample mean to 0
        featurewise_std_normalization=False,  # divide inputs by std of the dataset
        samplewise_std_normalization=False,  # divide each input by its std
        zca_whitening=False,  # apply ZCA whitening
        rotation_range=10,  # randomly rotate images in the range (degrees, 0 to
180)
        zoom_range = 0.1, # Randomly zoom image
        width_shift_range=0.1,  # randomly shift images horizontally (fraction of
total width)
        height_shift_range=0.1,  # randomly shift images vertically (fraction of
total height)
        horizontal_flip=False,  # randomly flip images
        vertical_flip=False)  # randomly flip images

datagen.fit(X_train)
```

```
'fitting the model'
epochs = 5
batch_size = 86


history = model.fit_generator(
                                datagen.flow(X_train,y_train,
batch_size=batch_size),
                                epochs = epochs, #An epoch is an iteration over
the entire x and y data provided
                                validation_data = (X_val,y_val), #Data on which to
evaluate the loss and any model metrics at the end of each epoch.
                                verbose = 1, #output
                                steps_per_epoch=X_train.shape[0]  //  batch_size,
# Total number of steps (batches of samples) before declaring one epoch finished
and starting the next epoch.
                                callbacks=[learning_rate_reduction]
                               )


'saving the model in HDF5 binary data format'
model.save(".....//model.h5")
```

#---3. Prediction---

```
from PIL import Image
from itertools import groupby

'loading image in grayscale'
image = Image.open(".....//testing.png").convert("L")

'resizing to 28 height pixels'
w = image.size[0]
h = image.size[1]
r = w / h # aspect ratio
new_w = int(r * 28)
new_h = 28
new_image = image.resize((new_w, new_h))

'converting to a numpy array'
```

```
new_image_arr = np.array(new_image)

'inverting the image to make background = 0'
new_inv_image_arr = 255 - new_image_arr

'rescaling the image'
final_image_arr = new_inv_image_arr / 255.0

'splitting image array into individual element arrays using non zero columns'
m = final_image_arr.any(0)
out = [final_image_arr[:,[*g]] for k, g in groupby(np.arange(len(m)), lambda x:
m[x] != 0) if k]


'''
iterating through the element arrays to resize them to match input
criteria of the model = [mini_batch_size, height, width, channels]
'''
num_of_elements = len(out)
elements_list = []

for x in range(0, num_of_elements):

    img = out[x]

    #adding 0 value columns as fillers
    width = img.shape[1]
    filler = (final_image_arr.shape[0] - width) / 2



   if filler.is_integer() == False:    #odd number of filler columns
       filler_l = int(filler)
       filler_r = int(filler) + 1
   else:                               #even number of filler columns
       filler_l = int(filler)
       filler_r = int(filler)

   arr_l = np.zeros((final_image_arr.shape[0], filler_l)) #left fillers
   arr_r = np.zeros((final_image_arr.shape[0], filler_r)) #right fillers

   #concatinating the left and right fillers
   help_ = np.concatenate((arr_l, img), axis= 1)
```

```
    element_arr = np.concatenate((help_, arr_r), axis= 1)

    element_arr.resize(28, 28, 1) #resize array 2d to 3d

    #storing all elements in a list
    elements_list.append(element_arr)


elements_array = np.array(elements_list)

'reshaping to fit model input criteria'
elements_array = elements_array.reshape(-1, 28, 28, 1)

'predicting using the created model'
model = keras.models.load_model("...//model.h5")
elements_pred =  model.predict(elements_array)
elements_pred = np.argmax(elements_pred, axis = 1)
```

#---4. Mathematical Operation---

```
def math_expression_generator(arr):

    op = {
              10,    # = "/"
              11,    # = "+"
              12,    # = "-"
              13     # = "*"
                  }



m_exp = []
temp = []

'creating a list separating all elements'
for item in arr:
    if item not in op:
        temp.append(item)
    else:
        m_exp.append(temp)
        m_exp.append(item)
        temp = []
```

```
    if temp:
        m_exp.append(temp)

    'converting the elements to numbers and operators'
    i = 0
    num = 0
    for item in m_exp:
        if type(item) == list:
            if not item:
                m_exp[i] = ""
                i = i + 1
            else:
                num_len = len(item)
                for digit in item:
                    num_len = num_len - 1
                    num = num + ((10 ** num_len) * digit)
                m_exp[i] = str(num)
                num = 0
                i = i + 1
        else:
            m_exp[i] = str(item)
            m_exp[i] = m_exp[i].replace("10","/")
            m_exp[i] = m_exp[i].replace("11","+")
            m_exp[i] = m_exp[i].replace("12","-")
            m_exp[i] = m_exp[i].replace("13","*")

            i = i + 1


        'joining the list of strings to create the mathematical expression'
        separator = ' '
        m_exp_str = separator.join(m_exp)

        return (m_exp_str)

'creating the mathematical expression'
m_exp_str = math_expression_generator(elements_pred)

'calculating the mathematical expression using eval()'
while True:
```

```
try:
    answer = eval(m_exp_str)     #evaluating the answer
    answer = round(answer, 2)
    equation  = m_exp_str + " = " + str(answer)
    print(equation)    #printing the equation
    break

except SyntaxError:
    print("Invalid predicted expression!!")
    print("Following is the predicted expression:")
    print(m_exp_str)
    break
```

5.4 RESULTS

The following is the original image which is used for testing

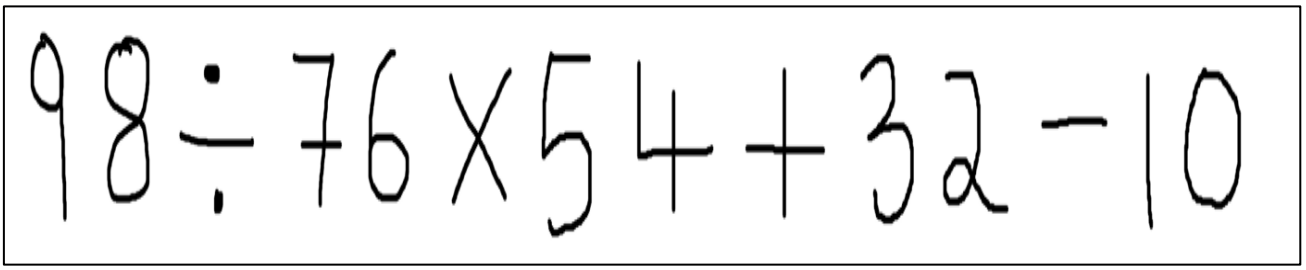


Fig 12: Processed Testing Image

The following is the tokenized image of above image:

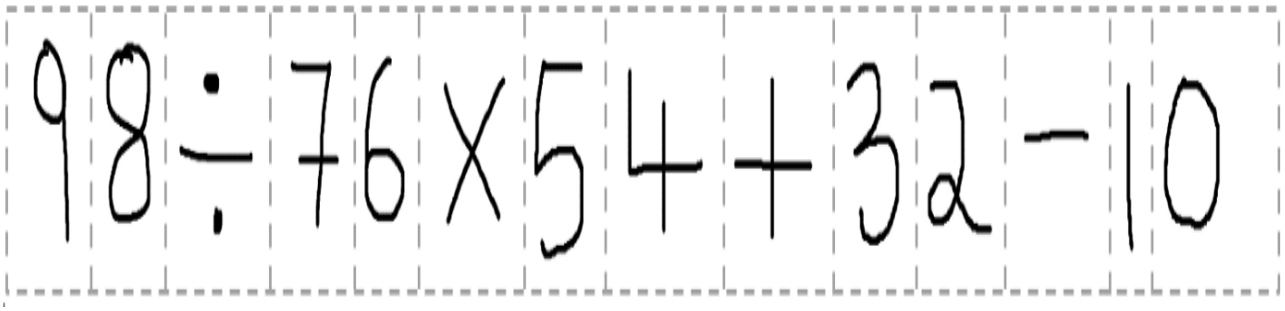


Fig 13: Segmentation of Testing Image

The following is the image after pre-processing:

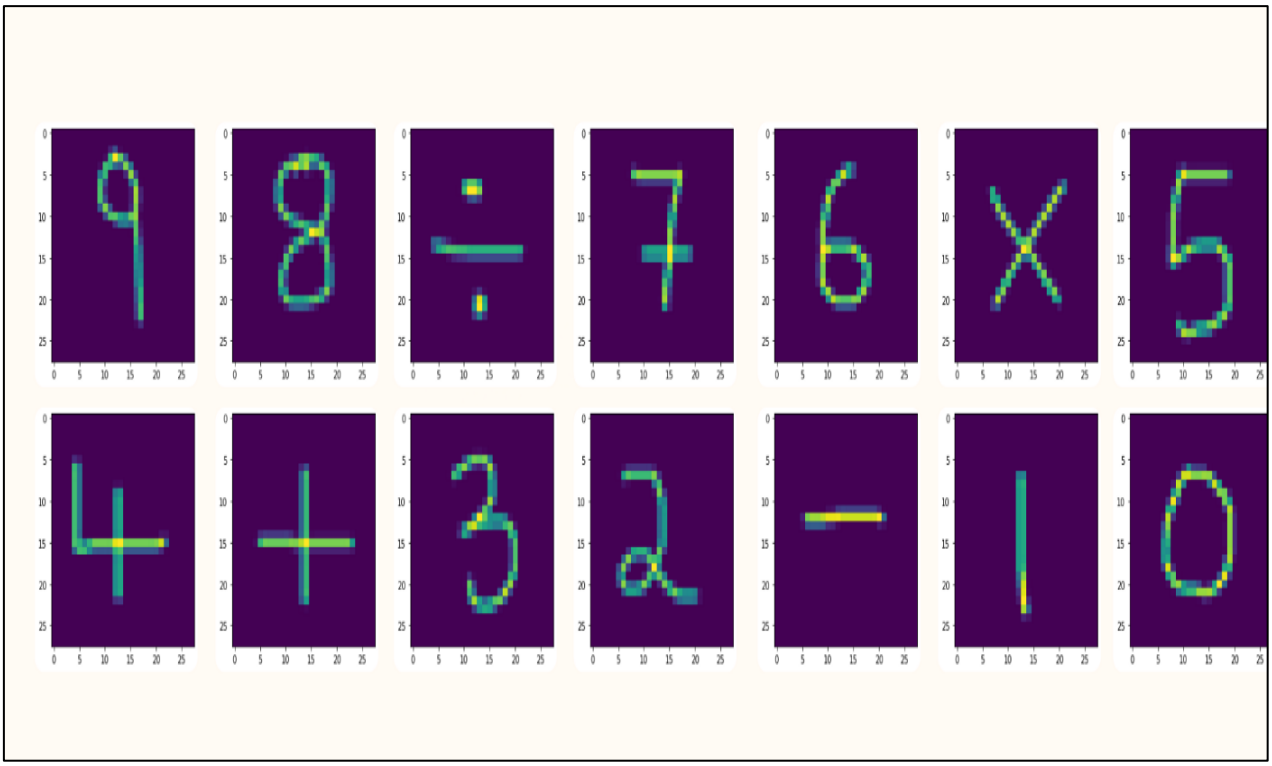


Fig 14: Pre-Processed Testing Image

TRAINING DATA

```
OUTPUT[]:
Epoch 1/5
 - 412s - loss: 0.2837 - accuracy: 0.9143 - val_loss: 0.0606 -
val_accuracy: 0.9831
Epoch 2/5
 - 439s - loss: 0.0776 - accuracy: 0.9771 - val_loss: 0.0348 -
val_accuracy: 0.9901
Epoch 3/5
 - 410s - loss: 0.0622 - accuracy: 0.9818 - val_loss: 0.0292 -
val_accuracy: 0.9919
Epoch 4/5
 - 413s - loss: 0.0578 - accuracy: 0.9837 - val_loss: 0.0285 -
val_accuracy: 0.9916
Epoch 5/5
 - 410s - loss: 0.0540 - accuracy: 0.9845 - val_loss: 0.0322 -
val_accuracy: 0.9917
```

Fig 15: Training Data

PREDICTION OF VALUE

```
In []: m_exp_str = math_expression_generator(elements_pred)

In []: print(m_exp_str)

Out[]: 98 / 76 * 54 + 32 - 10
```

```
In []: print(equation)

Out[]: 98 / 76 * 54 + 32 - 10 = 91.63
```

Fig 16: Predicting the value

EXECUTION SCREENS

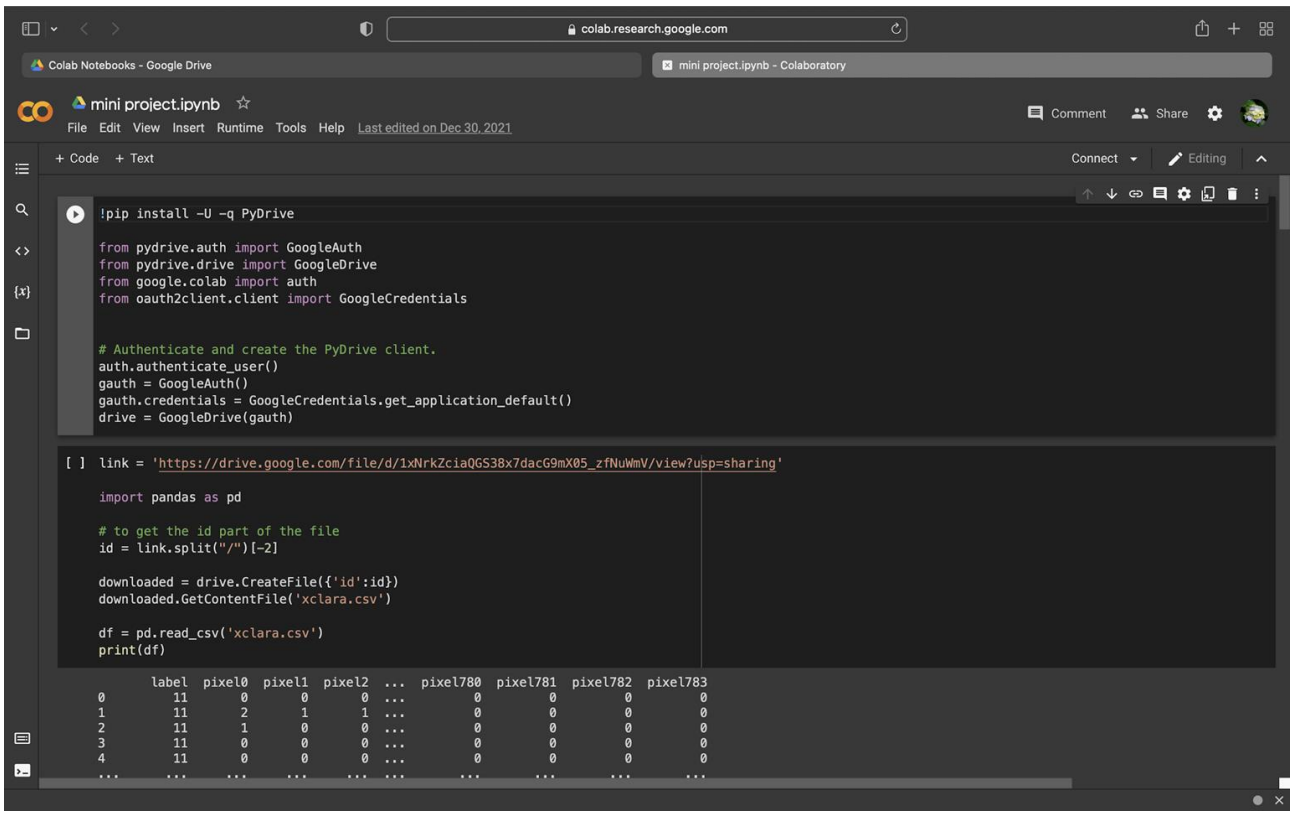


Fig 17: Execution

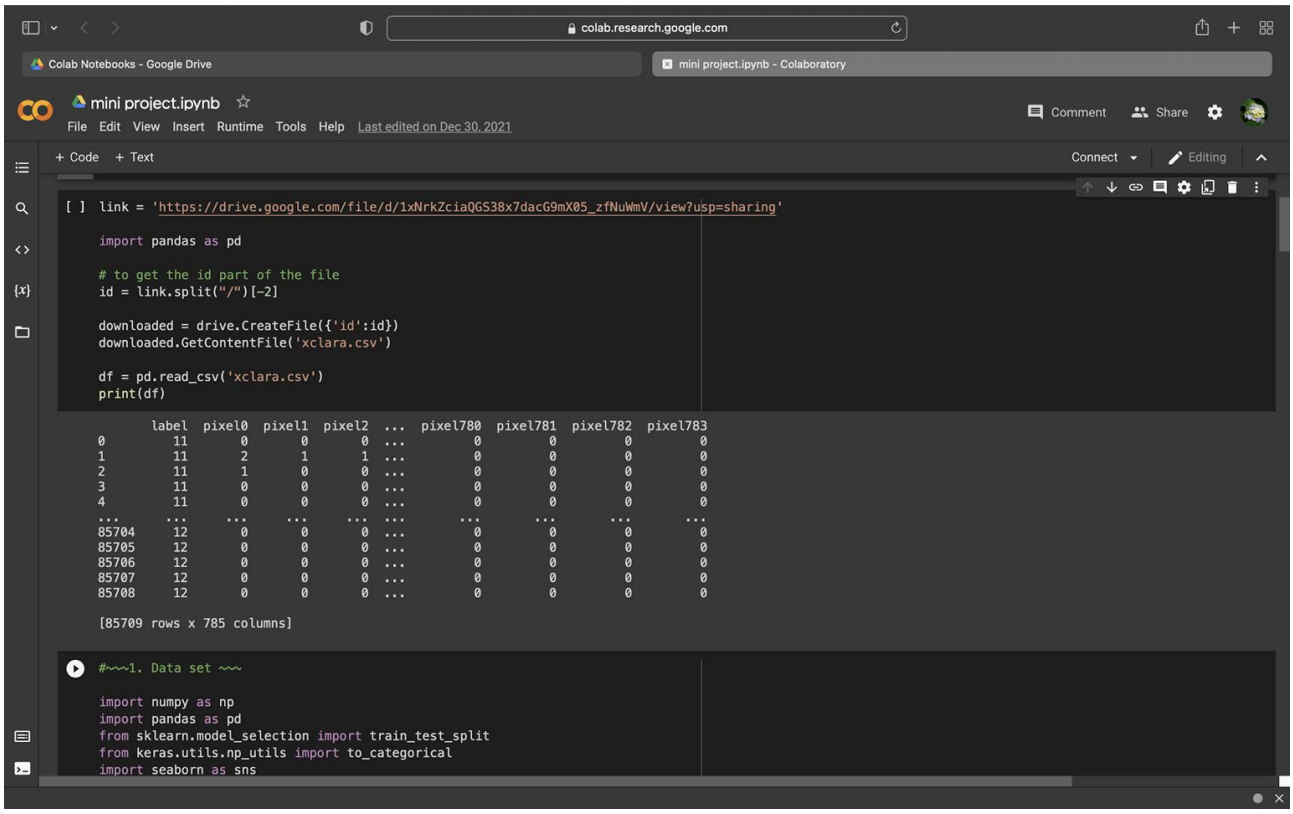


Fig 18: Execution

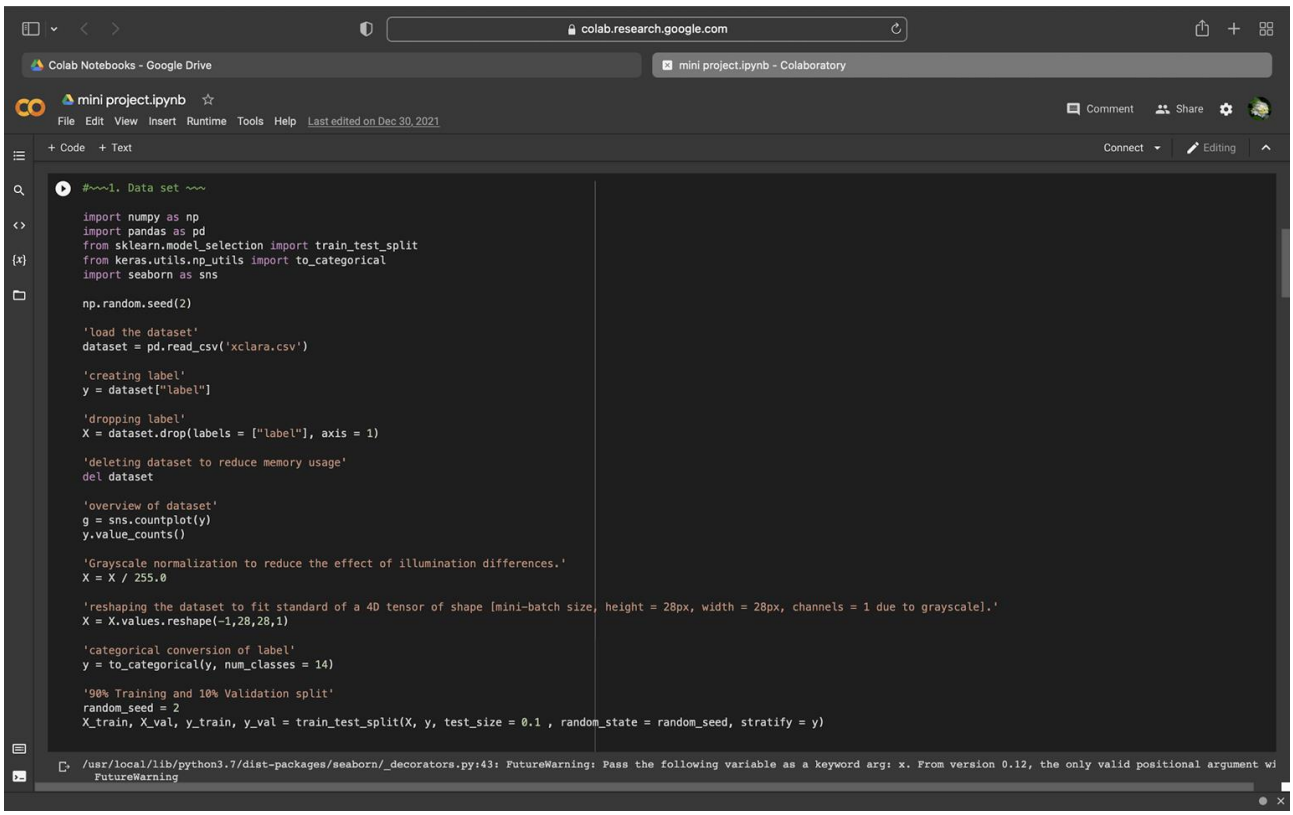


Fig 19: Execution

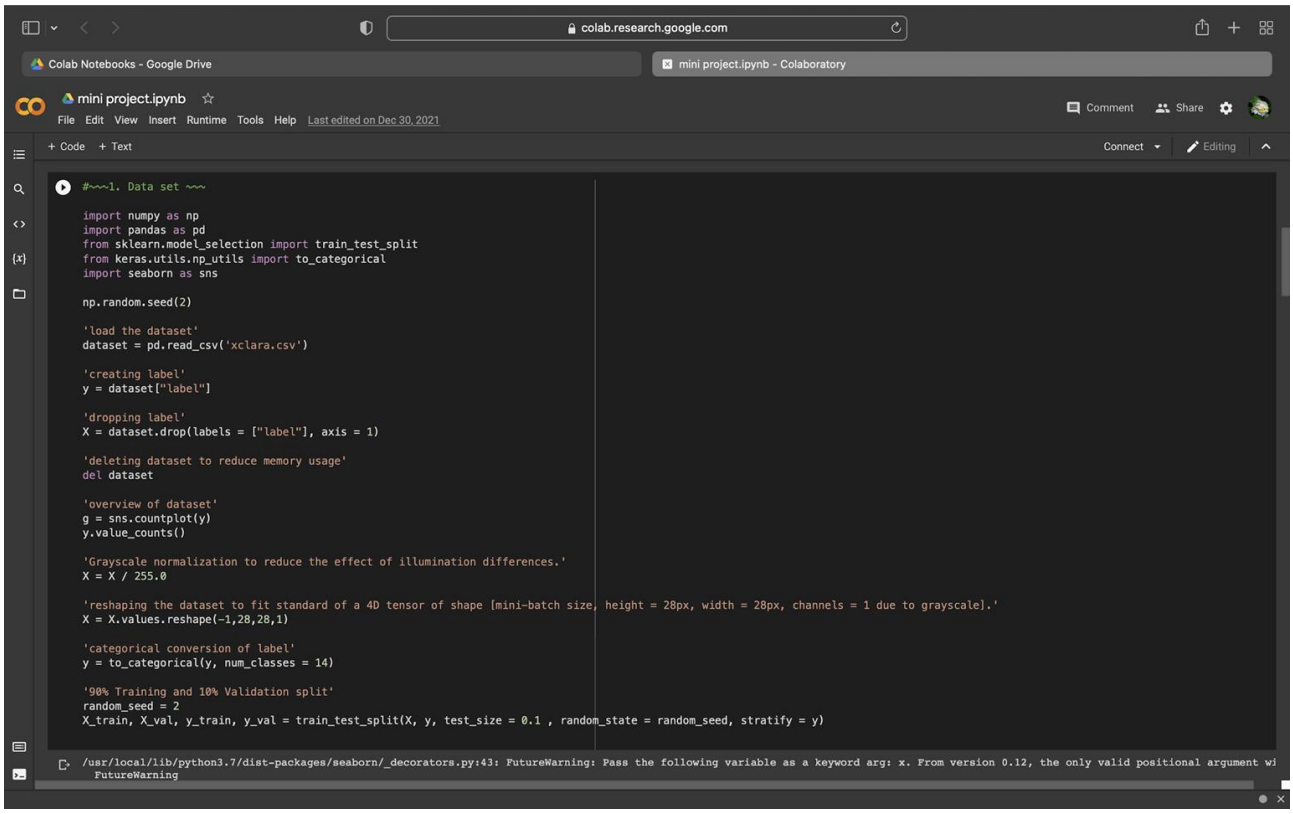


Fig 20: Execution

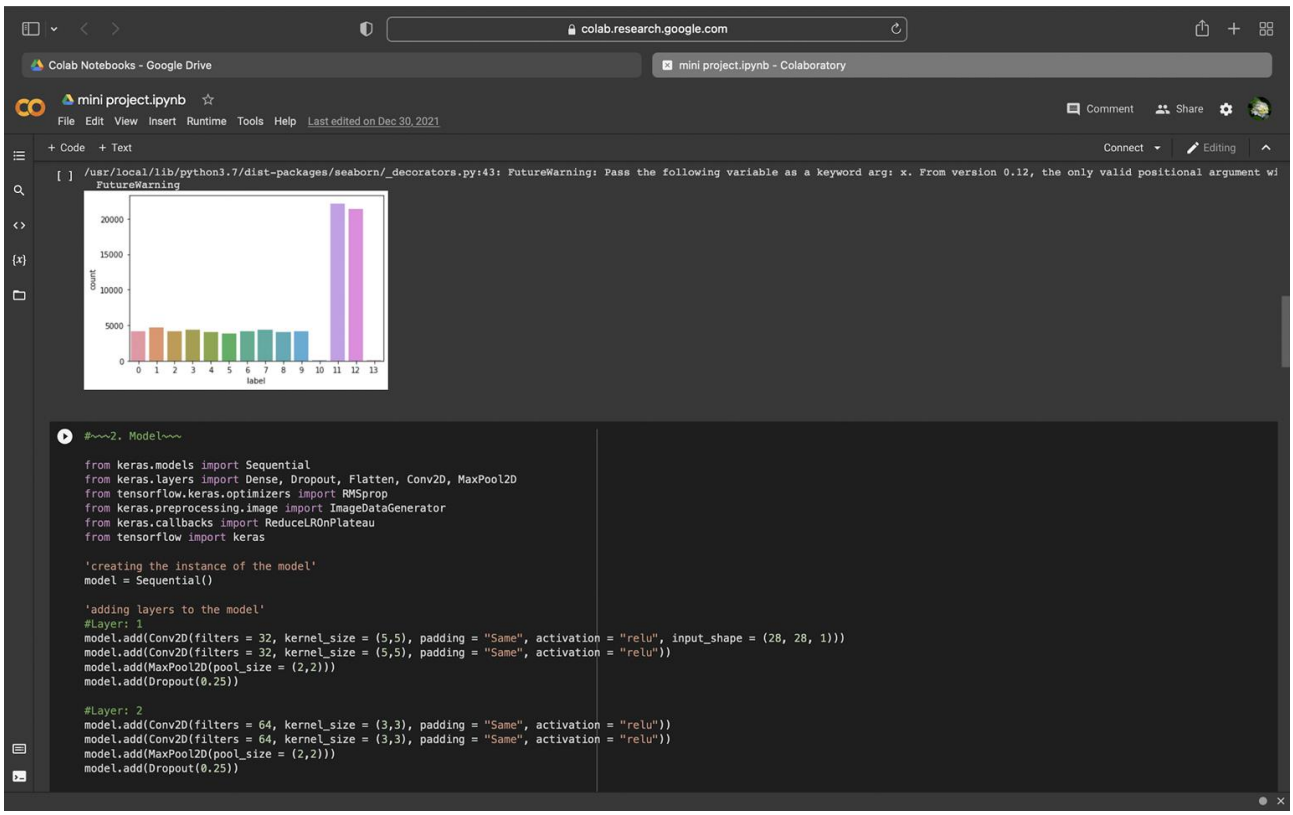


Fig 21: Execution

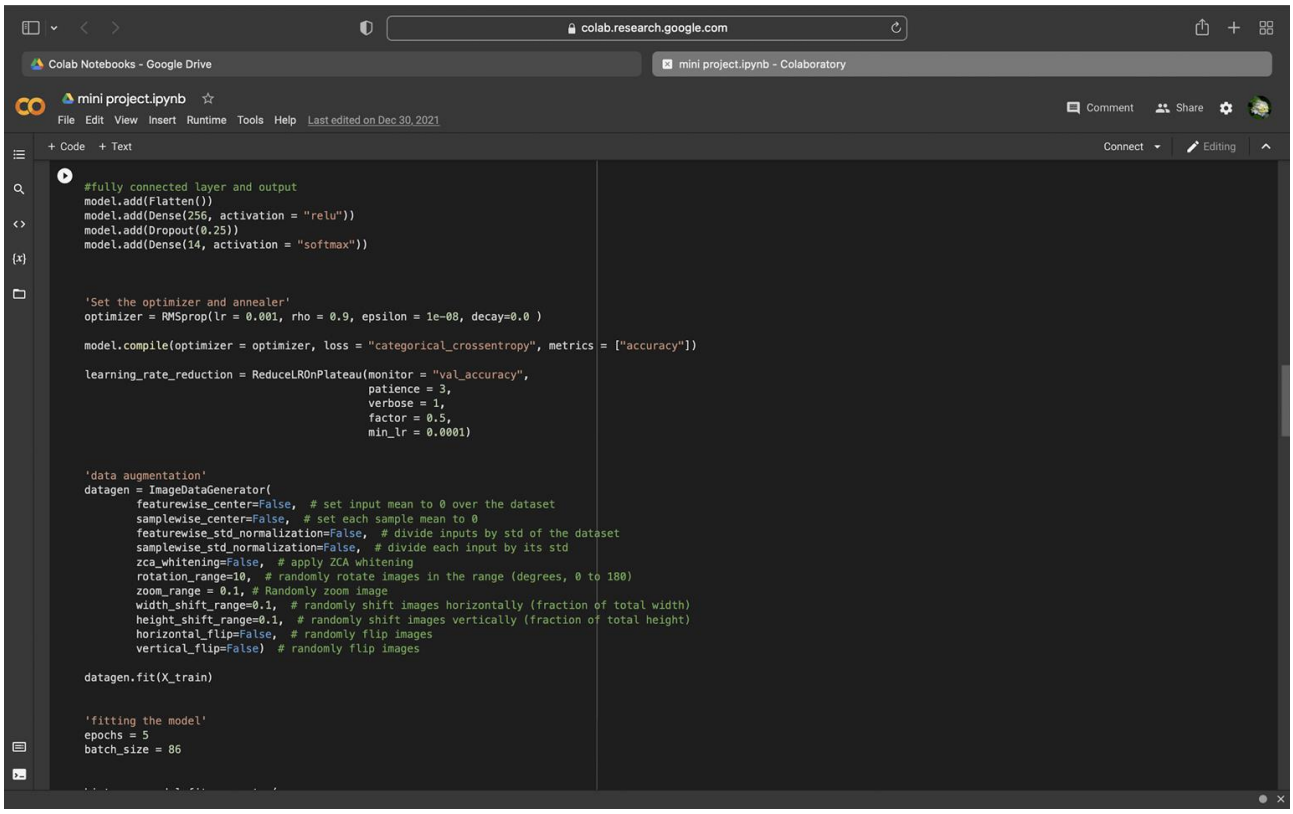


Fig 22: Execution

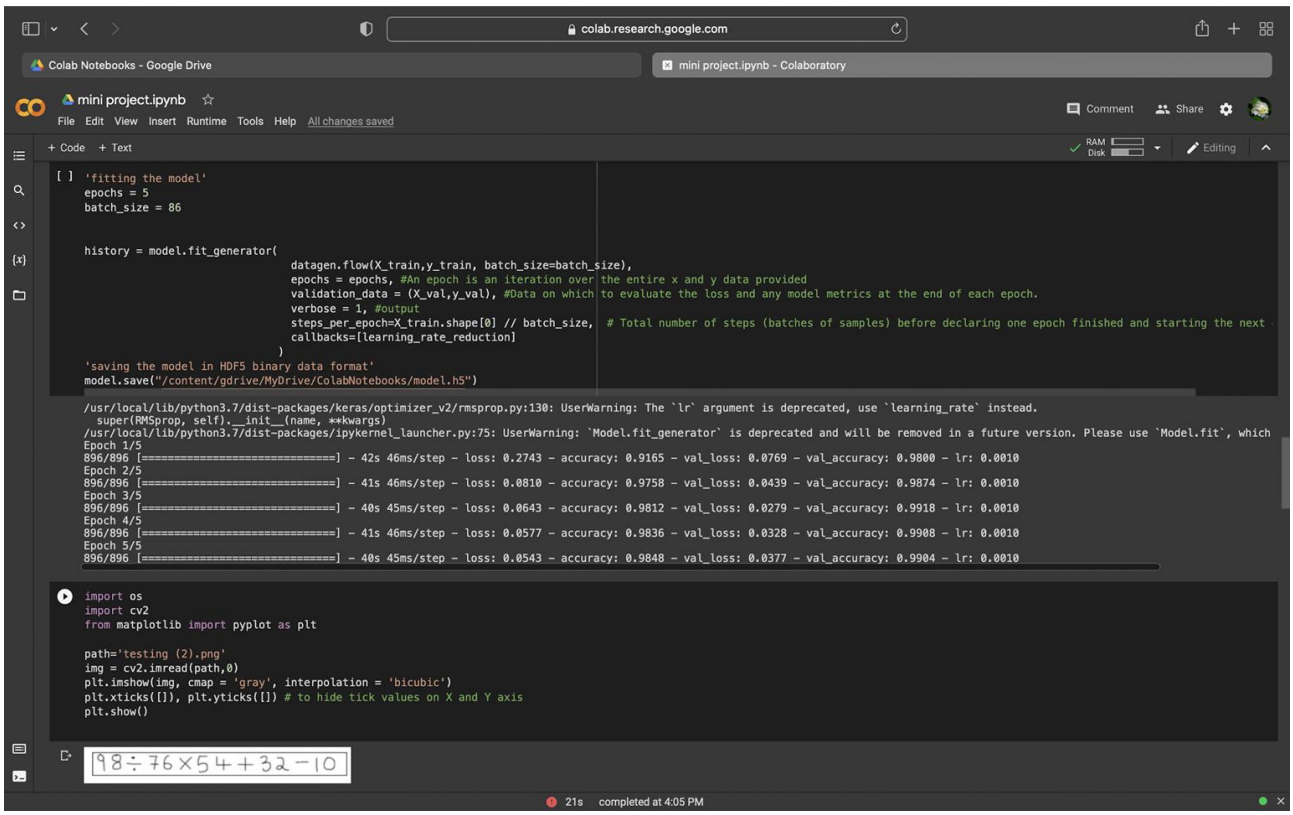


Fig 23: Execution

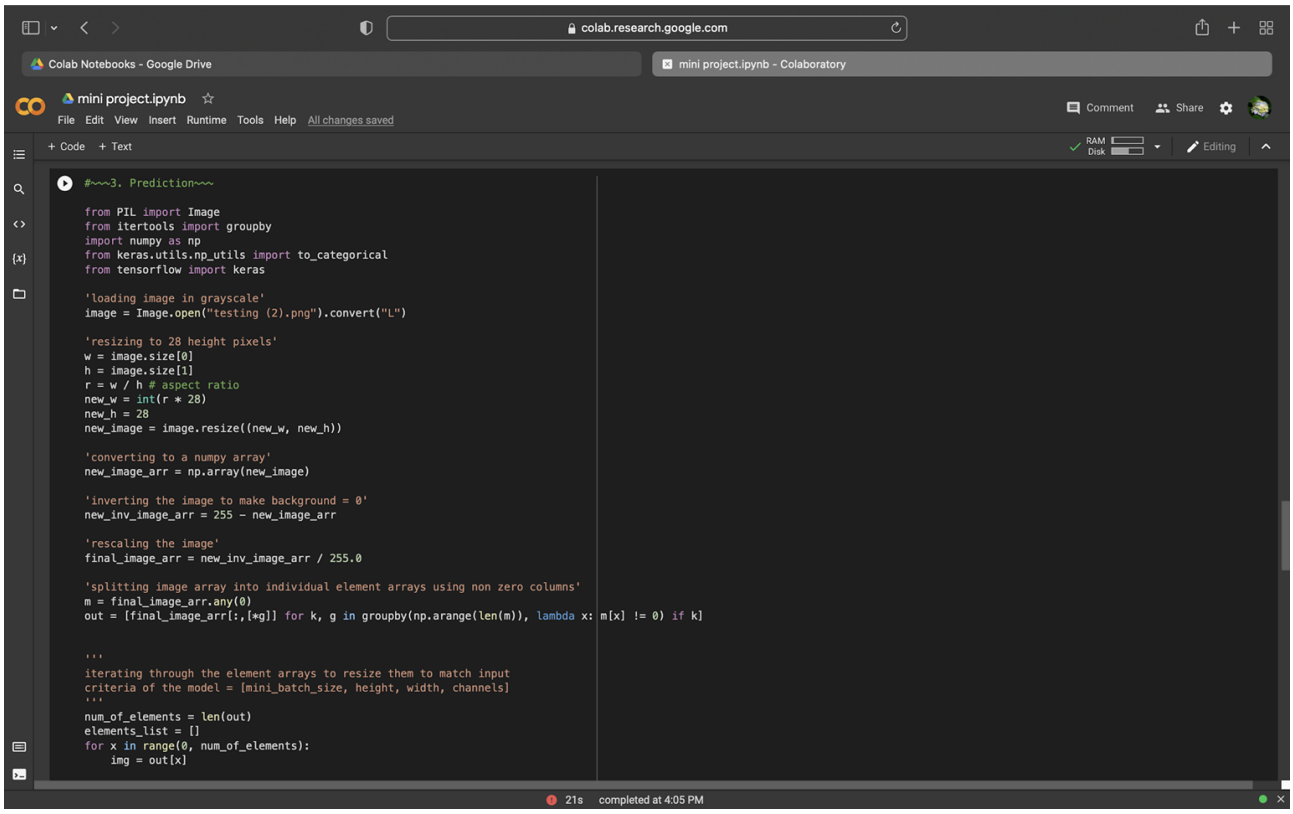


Fig 24: Execution

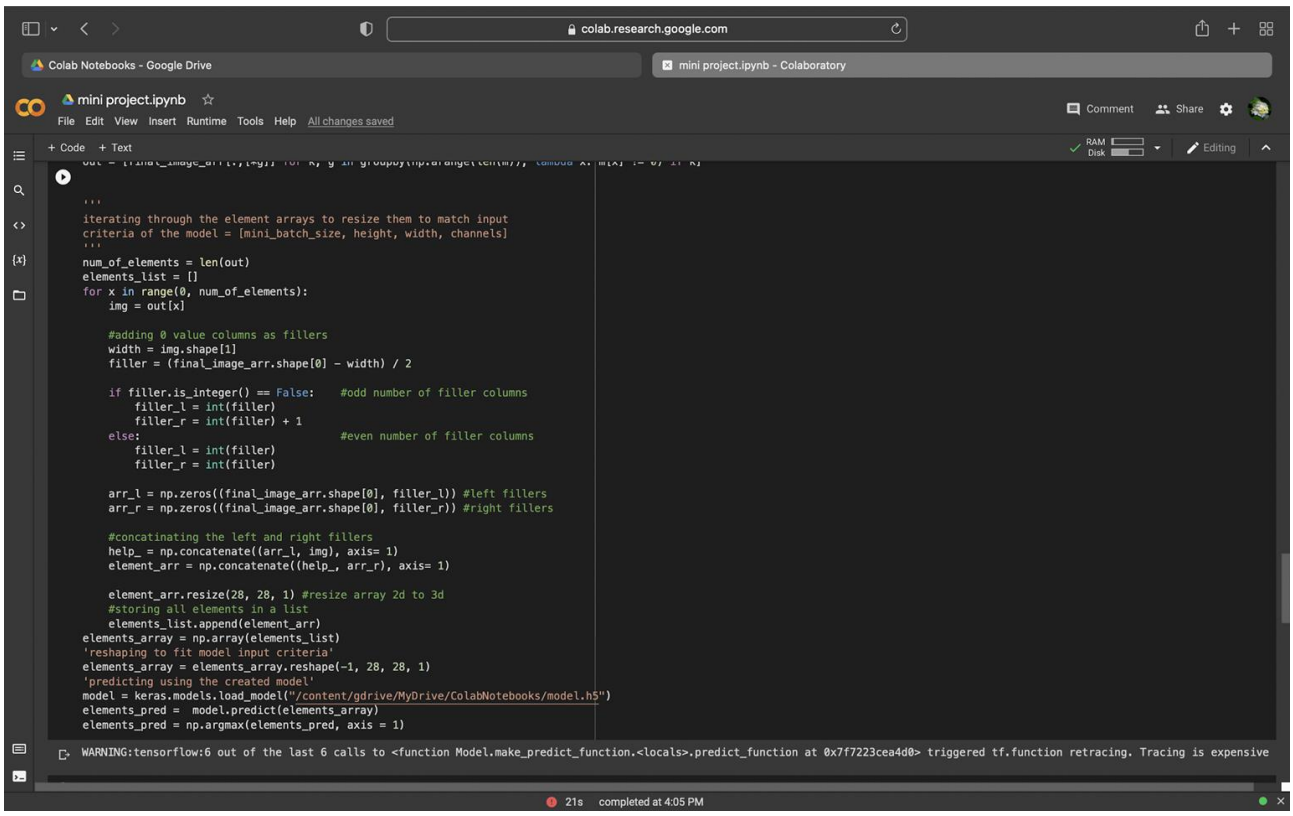


Fig 25: Execution

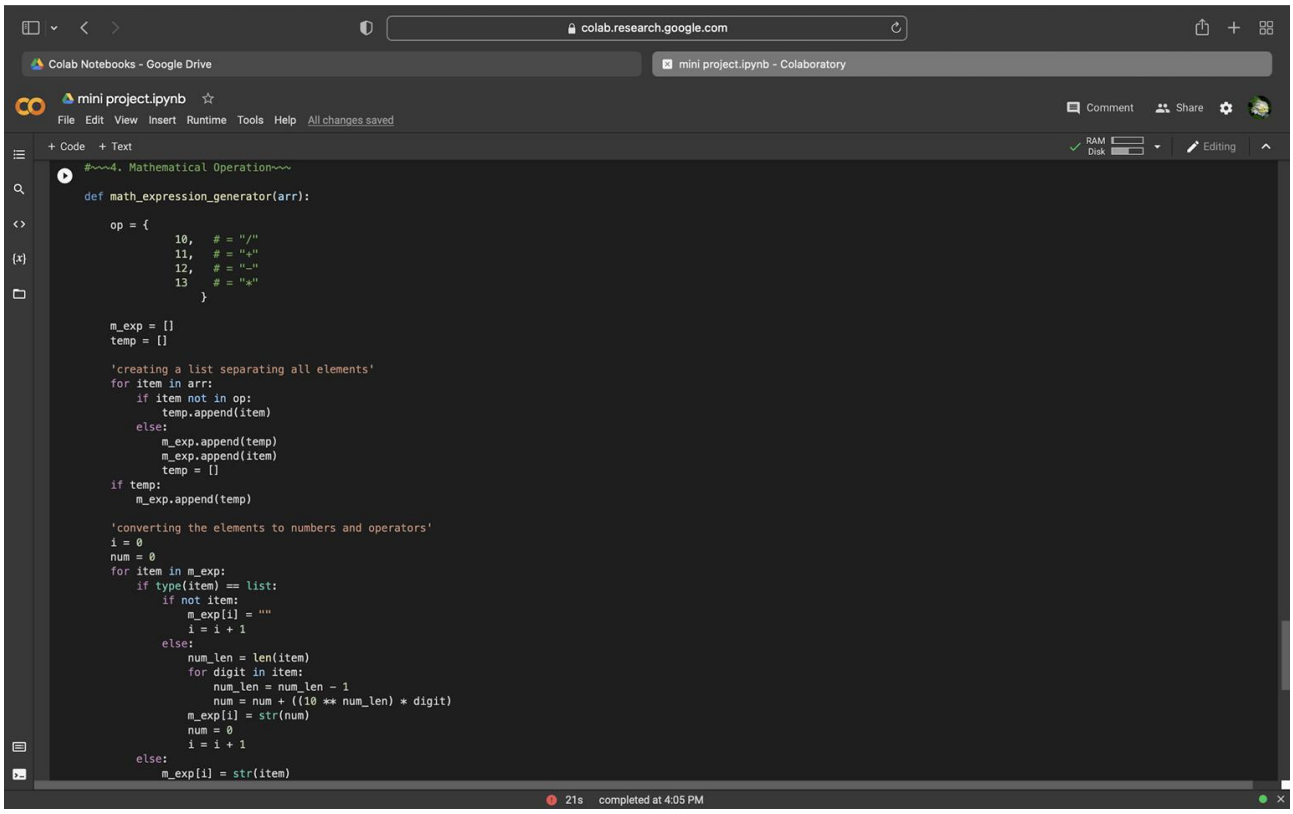


Fig 26: Execution

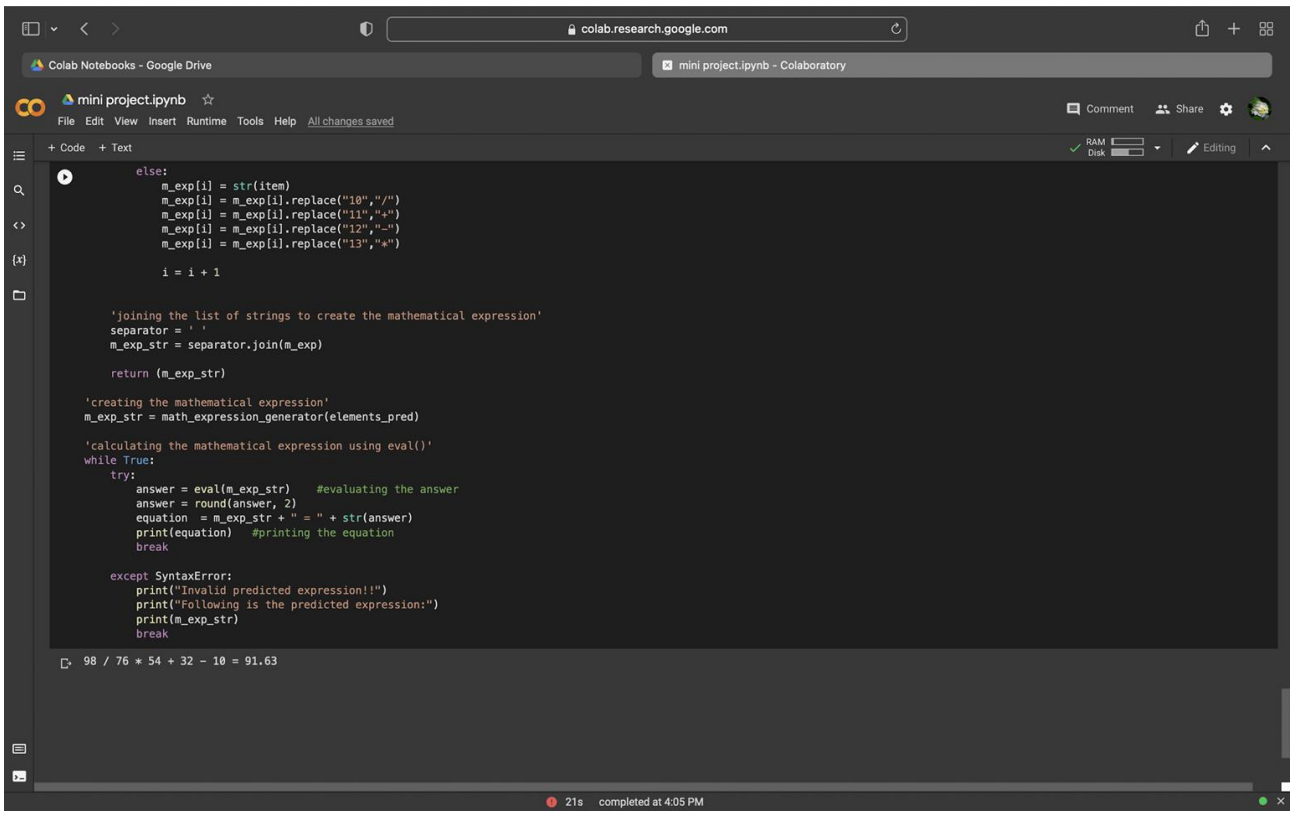


Fig 27: Execution

6. TESTING

The main purpose of testing is to prove that there are no errors in the system and to discover the errors present in the system. The main intention of testing is to find errors. Through testing we can check the functionality of all the components, assemblies, sun assemblies and also finished products. It is done to ensure that every part of the program is/ system is working properly. It helps in verifying the software requirements and to ensure that user expectations are met. There are various types of tests. Each test type addresses a specific testing requirement.

6.1 Types of Testing:

Unit testing:

Unit testing is done to check if the main program logic is working properly or not. It is done to check if the program is accepting the correct set of inputs and if it is giving the right outputs. Unit testing helps to validate the source code and find any errors within the code. It involves statement testing, branch testing and path testing to ensure proper flow of code. It helps to ensure that all the branches are accessible by at least one test case. Before integrating all the units to form a module unit testing is done. Unit testing is done to perform tests at the component level and to ensure that a certain path of the program does exactly as required by the user and gives the results as desired by the user.

Integration testing:

Integration testing is done to ensure that the integration of all the units was successful and the module is working as per the required specifications. This is an event driven type of testing. Integration testing is done to show that the single units that were working just fine individually are also working correctly

and consistently after being combined together. With the help of integration testing one aims to find any error at the integration stage and to ensure that all the components complement each other. The combination of the different units should give a module as desired by the user.

System Testing:

System testing is done to ensure that the integrated system is meeting the requirements of the user. This involves testing the configuration. This is a configuration oriented system integration test. It focuses on the flow of the processes. It also focuses on the various integration points and any pre driven process links.

Acceptance Testing:

Significant participation by the end user is mainly required in acceptance testing since it is a critical phase in any project. It ensures that all the requirements like functional, non- functional and user requirements are met by the system.

White Box Testing:

White Box testing is structural type testing. To conduct white box testing the tester should have knowledge about the code structure. He should know the language and platform of the program. He should have full idea about the code and the flow of the entire process and only then can he do this testing. White box testing is done by the developer.

Black Box Testing:

This type of testing does not need to have knowledge about the internal structure, programming language used, modules used. It should be taken from the requirements specification documentation. In this testing we provide inputs and check the outputs irrespective of knowledge we have on the software used or how it works.

Performance Testing:

Performance testing is the process of evaluating software's run-time performance as a part of a larger system. It's frequency is used in conjunction with stress testing and typically necessitates both hardware and software. It can detect circumstances that contribute to system degradation and failure.

6.2 Test cases

TEST CASE SCENARIO	TEST ID	CATEGORY	FEATURE DESCRIPTION	PREREQUISITES	STEPS	INPUT DATA	EXPECTED RESULTS	ACTUAL RESULTS	PASS/FAIL
VERIFYING HANDWRITTEN EQUATION	TC-1	FU	Verify whether the image is of correct extension	images should be loaded from dataset	load the images from dataset	Digits and operators images	uploaded successfully	uploaded successfully	P
	TC-2		Verify that the image viewed is uploaded	images in the repository	upload the images to dataset	Digits and operators images	uploaded successfully	uploaded successfully	P
	TC-3		Verify whether pre-processing	images in the dataset	load the image and	Digits and operator	Pre-processed image	Pre-processed image	P

SOLVER USING CNN		NCTIONAL	can be applied on images		apply the method	s images			
	TC-4		Verify that normalization is done on the images.	images should be pre-processed	load the pre-processed images from dataset	Pre-processed image	Normalized image	Normalized image	P
	TC-5		Check if all images are of size 28*28 px	images should be pre-processed	load the pre-processed images from folder	pre-processed image	Verified successfully	Verified successfully	P

	TC-6		Check that the background of the image is white	images should be pre-processed	load the pre-processed images from folder	pre-processed image	Verified successfully	Verified successfully	P
	TC-7		Verify that the digit is written in black on each image.	images should be pre-processed	load the pre-processed images from folder	pre-processed image	Verified successfully	Verified successfully	P
	TC-8		Verify that image features are obtained	images should be pre-processed	load the pre-processed images	pre-processed image	Verified successfully	Verified successfully	P

					from folder				
	TC-9		Verify that features obtained are classified into 14 classes.	images should be pre-processed	load the pre-processed images from folder	pre-processed image	Verified successfully	Verified successfully	P
	TC-10		Apply Softmax activation function.	Features to be extracted	Feature stored in a feature pool	Feature pool	Grouped images into the class with high probability	Grouped images into the class with high probability	P
	TC-11		Verify if the testing image	Testing image	Upload the testing	Image	Verified properly	Verified properly	P

			has correct extension		image into colab.				
	TC-12		Pre-process the testing image	Testing image	load the image and apply the method	Image	Pre-processed image	Pre-processed image	P
	TC-13		Apply testing process on the image	Testing image	load the image and apply the method	Image	Accuracy	Accuracy	P
	TC-14		Get the expression from the image	Testing image	load the image and apply	Image	Expression extracted	Expression extracted	P

					the method				
	TC-15		Evaluate the expression	Expression extracted	Testing the image	Image	Value of expression	Value of expression	P
	TC-16		Verify if the value of the expression is same as the original value	Value of expression	Testing the model	Data obtained	Verified successfully	Verified successfully	P

Table 2: Test Cases

7. CONCLUSION AND FUTURE SCOPE

The project aims to help us understand how CNN models can be used to make simple tools like the calculator. With the advancement in technology, machine learning and deep learning are playing a crucial role in present times. Now, machine learning and deep learning techniques are being employed in handwriting recognition, robotics, artificial intelligence, and many more fields. Developing such system requires training our machines with data, making it capable to learn and make required prediction. The model gives the perfect answers with an accuracy of 99 percent.

What makes Machine learning algorithms so powerful is its ability to learn from its mistakes. In order to do that the model needs to be retrained with correct information. If there is a wrong prediction, the code asks for the correct expression.

It then compares the predicted with the correct expression and identifies the elements that are wrongly predicted. The code then trains the model from its second hidden layer till its output layer using the correct data (first hidden layer is not trained).

The updated model is saved and can be used later to make improved predictions. In future, these concepts help to do complex visualization analysis and build models.

8. BIBLIOGRAPHY

1. Chuangxia, H. New studies on dynamic analysis of inertial neural networks involving non-reduced order method. Neurocomputing 2019.
2. Alvear-Sandoval, R. On building ensembles of stacked denoising auto- encoding classifiers and their further improvement. Inf. Fusion 2018.
3. Cai, Z.W. Finite-time synchronization by switching state-feedback control for discontinuous Cohen– Grossberg neural networks with mixed delays. Int. J. Mach. Learn. Cybern. 2018.
4. Zeng, D. Adversarial learning for distant supervised relation extraction. Comput. Mater. Contin. 2018.

B15.pdf

ORIGINALITY REPORT

24%	21%	6%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	towardsdatascience.com Internet Source	7%
2	Submitted to Sreenidhi International School Student Paper	6%
3	ieeexplore.ieee.org Internet Source	2%
4	balu051989.files.wordpress.com Internet Source	1%
5	Submitted to Engineers Australia Student Paper	1%
6	Submitted to South Bank University Student Paper	1%
7	Submitted to University of Surrey Roehampton Student Paper	1%
8	www.slideshare.net Internet Source	1%
9	www.semanticscholar.org Internet Source	<1%

10	seminarprojects.com Internet Source	<1%
11	Submitted to American University of Beirut Student Paper	<1%
12	Submitted to Charotar University of Science And Technology Student Paper	<1%
13	themanagersguide.blogspot.com Internet Source	<1%
14	Submitted to University of Hertfordshire Student Paper	<1%
15	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
16	ntnuopen.ntnu.no Internet Source	<1%
17	Submitted to Middlesex University Student Paper	<1%
18	baadalsg.inflibnet.ac.in Internet Source	<1%
19	www.researchgate.net Internet Source	<1%
20	index-of.es Internet Source	<1%

21	www.ijiert.org Internet Source	<1%
22	iapsop.com Internet Source	<1%
23	www.escholar.manchester.ac.uk Internet Source	<1%
24	downloads.optimumg.com Internet Source	<1%
25	Petraq J. Papajorgji, Panos M. Pardalos. "Software Engineering Techniques Applied to Agricultural Systems", Springer Science and Business Media LLC, 2014 Publication	<1%
26	Submitted to University of Northampton Student Paper	<1%

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		

9. APPENDIX

A. About Python:

Python is a programming language, which means it's a language both people and computers can understand. Python was developed by a Dutch software engineer named Guido van Rossum, who created the language to solve some problems he saw in computer languages of the time. Python is an interpreted high-level programming language for general purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems. C Python, the reference implementation of Python, is open source software and has a community based development model, as do nearly all of its variant implementations. C Python is managed by the non-profit Python Software Foundation.

YOU CAN USE PYTHON FOR PRETTY MUCH ANYTHING:

One significant advantage of learning Python is that it's a general purpose language that can be applied in a large variety of projects. Below are just some of the most common fields where Python has found its use:

- ✓ Data science
- ✓ Scientific and mathematical computing

✓ Web development

✓ Computer graphics

✓ Basic game development

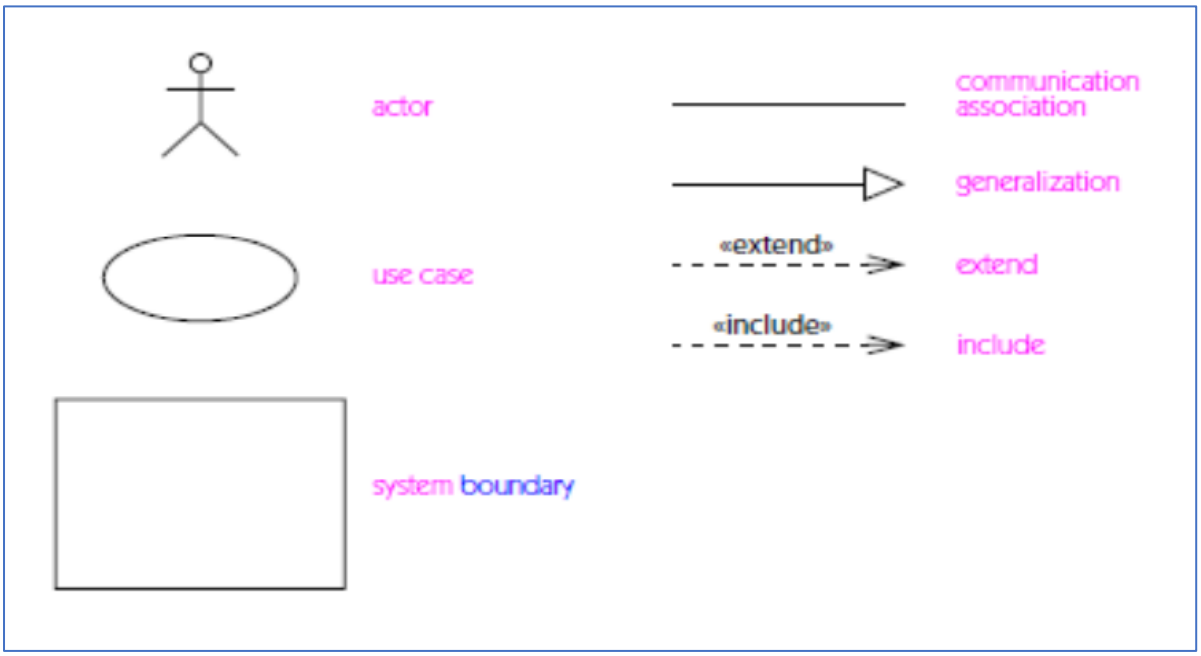
✓ Mapping and geography (GIS software)

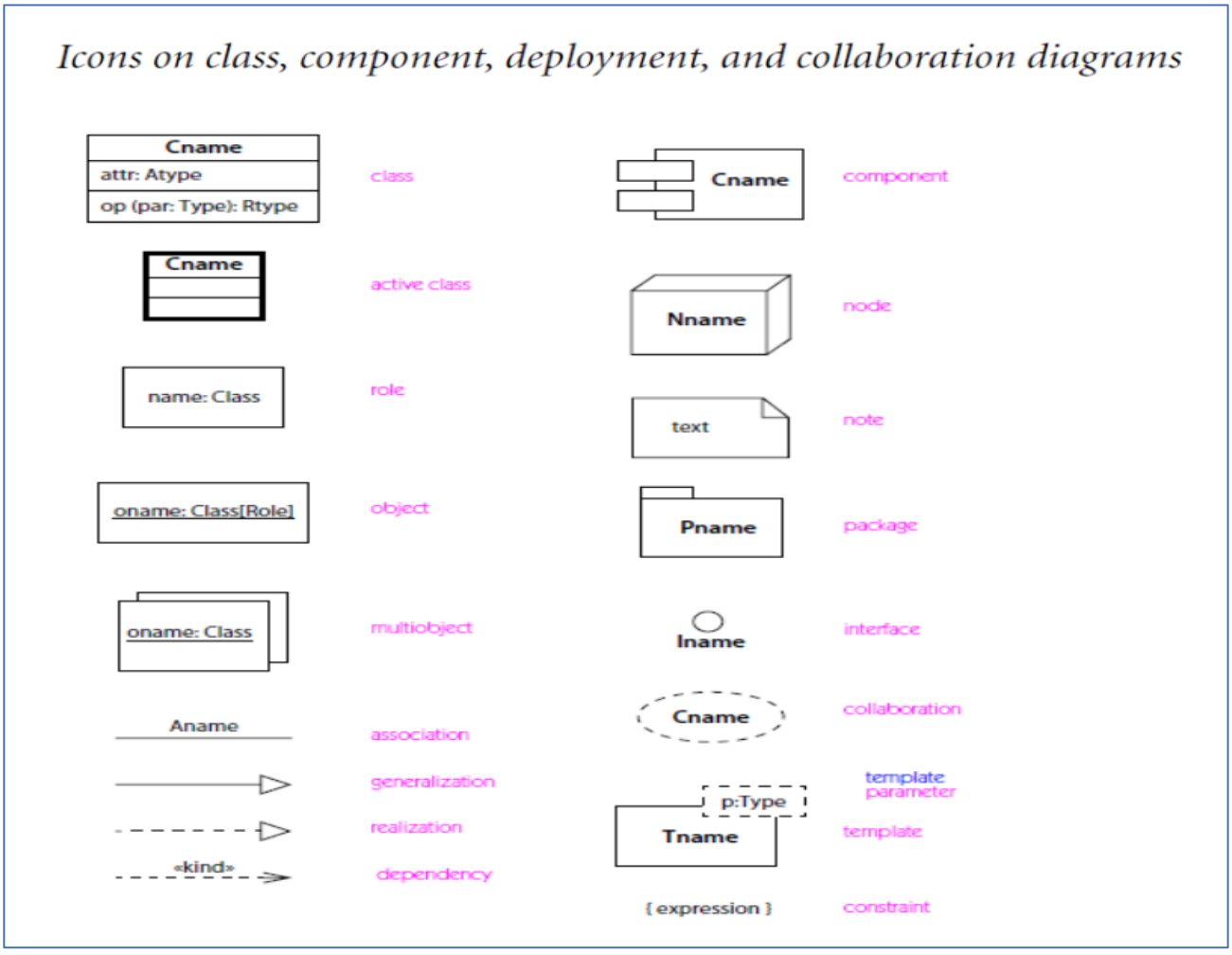
B. About UML:

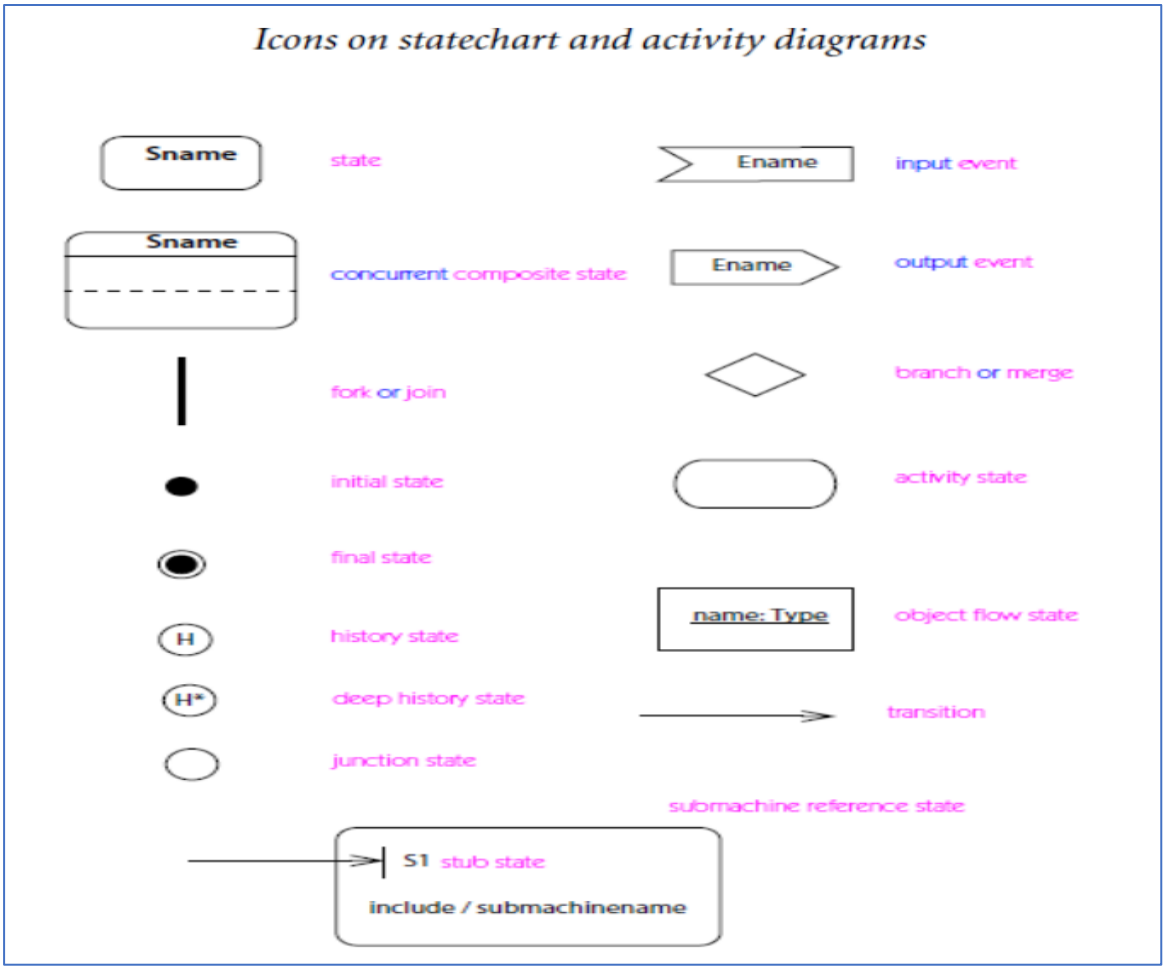
The Unified Modeling Language (UML) is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of a software system. It captures decisions and understanding about systems that must be constructed. It is used to understand, design, browse, configure, maintain, and control information about such systems. It is intended for use with all development methods, lifecycle stages, application domains, and media. The modeling language is intended to unify past experience about modeling techniques and to incorporate current software best practices into a standard approach.

UML includes semantic concepts, notation, and guidelines. It has static, dynamic, environmental, and organizational parts. It is intended to be supported by interactive visual modeling tools that have code generators and report writers. The UML specification does not define a standard process but is intended to be useful with an iterative development process. It is intended to support most existing object oriented development processes.

The UML captures information about the static structure and dynamic behavior of a system. A system is modeled as a collection of discrete objects that interact to perform work that ultimately benefits an outside user. The static structure defines the kinds of objects important to a system and to its implementation, as well as the relationships among the objects. The dynamic behavior defines the history of objects over time and the communications among objects to accomplish goals.


actor
use case
system boundary
communication association
generalization
«extend»
extend
«include»
include


Icons on class, component, deployment, and collaboration diagrams
Cname
attr: Atype
op (par: Type): Rtype
class
Cname
active class
name: Class
role
oname: Class[Role]
object
oname: Class
multiobject
Aname
association
generalization
realization
«kind»
dependency
Cname
component
Nname
node
text
note
Pname
package
Iname
interface
Cname
collaboration
p:Type
Tname
template parameter
template
{ expression }
constraint


Icons on statechart and activity diagrams
Sname
state
Ename
input event
Sname
concurrent composite state
Ename
output event
fork or join
branch or merge
initial state
activity state
final state
name: Type
object flow state
H
history state
H*
deep history state
transition
junction state
submachine reference state
S1 stub state
include / submachinename

Sreenidhi Institute of Science and Technology
Department of Computer Science and Engineering
PROJECT – I

BATCH : B15		**TITLE OF THE PROJECT**
ROLL NO	**NAME**	
18311A0599	SHREYA MALRAJU	Handwritten Equation Solver Using CNN

ABSTRACT

During the second semester of 3rd year, I was a part of the project Handwritten Digit Recognition System which classifies the handwritten numbers into its respective class ranging from 0-9.This semester, as an extension to the previous project, I am in a process of developing a system which solves Handwritten Arithmetic Expressions. Methods of CNN is used to train the data , which contains numbers from 0-9 and arithmetic operators (such as + , - , *, /) hence classification of elements is done into 13 classes. Robust handwritten character recognition is a tricky job in the area of image processing. Among all the problem handwritten mathematical expression recognition is one of the complicated issue in the area of computer vision research. Segmentation and classification of specific character makes the task more difficult. In this paper a group of handwritten quadratic equation as well as a single quadratic equation are considered to recognize and make a solution for those equation. Horizontal compact projection analysis and combined connected component analysis methods are used for segmentation. For classification of specific character we apply Convolutional Neural Network. Each of the correct detection, character string operation is used for the solution of the equation.

Project Coordinator	**Project Guide**	**Head of the Department**
Mrs. Pallepati Vasavi	**Mr. R.Arun Kumar**	**Dr. Aruna Varanasi**
Assistant Professor	**Assistant Professor**	**Professor & HOD**

Shreya Malraju (18311A0599)

BATCH : B15		TITLE OF THE PROJECT
ROLL NO	NAME	
18311A0599	SHREYA MALRAJU	Handwritten Equation Solver Using CNN

Table 1: Project correlation with appropriate POs/PSOs (Please specify level of Correlation, H/M/L against POs/PSOs)

H	High	M	Moderate	L	Low

SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Projects Correlation with POs/PSOs

PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
M	M	M	L	H	M	M	M	H	L	L	M	M	M	M

Project Coordinator
Mrs. Pallepati Vasavi
Assistant Professor

Project Guide
Mr. R.Arun Kumar
Assistant Professor

Head of the Department
Dr. Aruna Varanasi
Professor & HOD

Shreya Malraju (18311A0599)

BATCH : B15		TITLE OF THE PROJECT
ROLL NO	NAME	
18311A0599	SHREYA MALRAJU	Handwritten Equation Solver Using CNN

Table 2: Nature of the Project (Please tick √ Appropriate for your project)

Batch No.	Title	Nature of Project		
		Product	Application	Research
B15	Handwritten Equation Solver using CNN			✓

Project Coordinator	Project Guide	Head of the Department
Mrs. Pallepati Vasavi	Mr. R.Arun Kumar	Dr. Aruna Varanasi
Assistant Professor	Assistant Professor	Professor & HOD

Shreya Malraju (18311A0599)

BATCH : B15		TITLE OF THE PROJECT
ROLL NO	NAME	
18311A0599	SHREYA MALRAJU	Handwritten Equation Solver Using CNN

Table 3: Domain of the Project (Please tick √ Appropriate for your project)

Batch No.	Title	Domain of the Project				
		ARTIFICIAL INTELLIGENCE, MACHINE LEARNING AND DEEP LEARNING	COMPUTER NETWORKS, INFORMATION SECURITY, CYBER SECURITY	DATA WAREHOUSING, DATA MINING, BIG DATA ANALYTICS	CLOUD COMPUTING, INTERNET OF THINGS	SOFTWARE ENGINEERING, IMAGE PROCESSING
B15	Handwritten Equation Solver using CNN					✓

Project Coordinator	Project Guide	Head of the Department
Mrs. Pallepati Vasavi	Mr. R.Arun Kumar	Dr. Aruna Varanasi
Assistant Professor	Assistant Professor	Professor & HOD

Shreya Malraju (18311A0599)