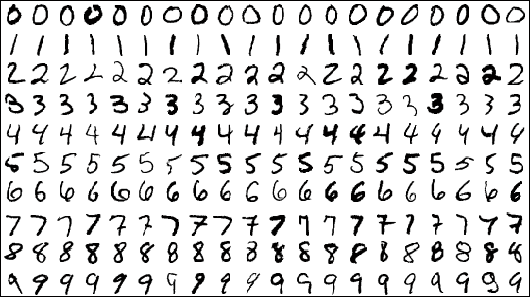


# Project Name: Calculator for Hand-written equations

**Subject Name: Deep Learning and Neural Networks Subject Code: 15CSE380**



**Project By:**

Kandula Kamal (BL.EN.U4CSE17510) Y V Sravan Kumar Reddy (BL.EN.U4CSE17524) Kandimalla Aashrith (BL.EN.U4CSE17525) Eskala Srekar (BL.EN.U4CSE17553)

**Project guide: Dr. Suja.P**

**Abstract**

## Hand-written character recognition is one of the practically essential issues in pattern recognition applications. One of them is digit classification. There are numerous applications for digit recognition. Some of them are postal mail sorting, bank check processing, form data entry, etc. The heart of the problem lies within the ability to develop an efficient algorithm that can recognize hand-written digits, which is submitted by users by way of a scanner, tablet, and other digital devices. This project presents an offline approach to recognize an essential operation like addition or subtraction (that accepts a digit and then operator either plus or minus and then a number) using Convolutional Neural Networks and performs the calculation based on it. The main objective of this project is to provide an effective and reliable approach for recognition of hand-written digits. The data set we are using contains forty five thousand images of 28\*28 pixels to train the neural net.

All the signs we detect are: ‘1’,’2’,’3’,’4’,’5’,’6’,’7’,’8’,’9’,’0’,’+’,’-’,’\*’

Where ‘+’ := Indicates addition

‘-’ := Indicates subtraction ‘\*’ := Indicates Multiplication

***Implementation of CNN in our Project:***

**model = Sequential()**

-A Sequential neural network object is created and assigned to model variable.

**model.add(Conv2D(32, (5,5), activation='relu', input\_shape=(1,28,28), data\_format='channels\_first'))**

-Added a Input Layer with required parameters.

**model.add(MaxPooling2D(pool\_size=(2, 2)))**

-Added a Max Pooling layer of size 2by2.

**model.add(Conv2D(15, (3, 3), activation='relu'))**

-Added a convolution layer with activation function as **Rectified Linear unit**(‘relu’).

**model.add(MaxPooling2D(pool\_size=(2, 2)))**

-Added a Max Pooling layer of size 2by2.

**model.add(Dropout(0.2))**

-Dropped out 20 percent features .

**model.add(Flatten())**

-Flattened all the features and given as input to the neural network.

**model.add(Dense(128, activation='relu'))**

-Created first neural network layer with activation function as relu and number of neurons is 128.

**model.add(Dense(50, activation='relu'))**

-Created first neural network layer with activation function as relu and number of neurons is 50.

**model.add(Dense(13, activation='softmax'))**

#Output layer we used softmax as activation function.

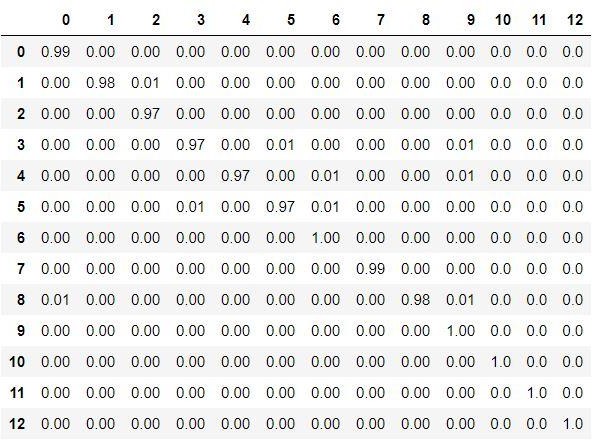
**model.compile(loss=’categorical\_crossentropy’, optimizer=’adam’, metrics=['accuracy'])**

#Finally compiled the network .

**model.fit(np.array(l), cat, batch\_size=200, epochs=10, shuffle=True, verbose=1)**

#We trained the model.

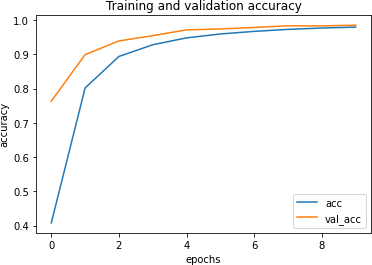
**Confusion matrix**



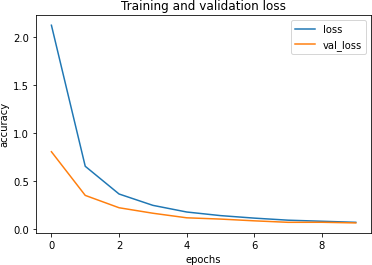
**Tools Used:**

1. Keras
2. OpenCV

**Training and validation accuracy:**

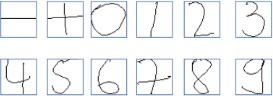


**Training and Validation loss:**



**Data set:**

1. We are using data set provided by Kaggle.
2. It contains more than 47504 images.
3. Each of size 28 by 28 pixels.
4. Sample data set image:

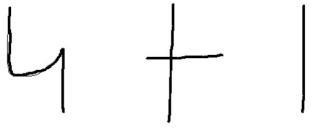


**Python Libraries used:**

1. numpy
2. pandas
3. matplotlib
4. keras
5. cv2
6. pickle
7. os

**Test images:**

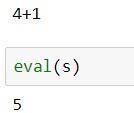
**-Sample input:**



**-gray scale images(sample processed image of input):**



**-Sample Output:**



**Problems encountered and how it was solved:**

*PROBLEM:* Even the accuracy was high the results are not satisfactory

*SOLUTION :* Decreased the number of layers in the convolution net and dropped out the irrelevant features that are not contributing the classification.

**Result:**

Built a model to evaluate the hand written expressions which include operations like ‘+’, ‘-’ , ‘ \*’. The images are classified into 13 different classes with an accuracy of 98%

**Contribution of each team member:**

Kamal: Image processing and built model Sravan: Image processing and built model Aashrith : Image processing and built model Srekar: Image processing and built model

**Scope of future work:**

There are numerous applications for digit recognition. Some of them are postal mail sorting, bank check processing, form data entry, etc.

**Whether you will extend the project:**

No, but we are planning to do some other projects in CNN and also want to work on time series data using RNN.