A Project report on

HAND WRITTEN MATH EQUATION SOLVER

submitted in partial fulfilment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

by

M. Girish kumar (17JJ1A0535)
K.Anish (17JJ1A0528)
Sk. Vazeer pasha (17JJ1A0548)
T. Nikhil Teja (17JJ1A0551)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Accredited by NBA)

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY
HYDERABAD COLLEGE OF ENGINEERING

NACHUPALLY(KONDAGATTU), JAGTIAL DIST-505501,
TELANGANA 2020-2021

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

COLLEGE OF ENGINEERING

NACHUPALLY(KONDAGATTU), JAGTIALDIST-505 501, TELANGANA

DEPARTMENT OF COMPUTER SCIENCE ANDENGINEERING

(Accredited by NBA)

CERTIFICATE

This is to certify that the major project named "**HAND WRITTEN MATH EQUATION SOLVER**"is a bonafide work carried out by M. GIRISH KUMAR (17JJ1A0535), K.ANISH (17JJ1A0528), T. NIKHIL TEJA (17JJ1A0551), SK. VAZEER PASHA (17JJ1A0548) in partial fulfilment of the requirements for the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2020-2021.

The results embodied in this report have not been submitted to any other University or Institution for the award of any other degree or diploma.

----------	----------
P. SreenivasaRao	Dr T.Venugopal
AssociateProfessor	Professor & head of Department
Department of CSE	CSE
Project Guide	

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to P. Sreenivasa Rao Garu, Associate Professor, CSE department, our Project Guide for his able, vigorous guidance and useful suggestions, which helped us in completing the project work in time.

We show gratitude to our beloved Principal Dr. N.V. RAMANA,Vice PrincipalDr S. VISWANADHA RAJU, for providing necessary infrastructure and resources for the accomplishment of our project report at JNTUHCEJ.

We also thank all the staff members of Computer Science and Engineering department, JNTUHCEJ for their valuable support and generous advice. Finally, thanks to all my friends and family member for their continuous support and enthusiastic help.

DECLARATION

We do declare that the project work entitled "**HAND WRITTEN MATH EQUATION SOLVER**" submitted by us in the Department of Computer Science and Engineering. JNTUH College of Engineering Jagtial, Telangana State in partial fulfilment of degree for the award of BACHELOR OF TECHNOLOGY is a bonafide work, which was carried out under the supervision of P. Sreenivasa Rao Garu, Associate Professor, CSE department, JNTUH College of Engineering, Jagtial.

Also, we declare that the matter embedded in this thesis has not been submitted by us in full or partial thereof for the award of any degree/diploma of any other institution or University previously.

Place :

Date :

M. GIRISH KUMAR (17JJ1A0535)

K. ANISH (17JJ1A0528)

T. NIKHIL TEJA (17JJ1A0551)

SK. VAZEER PASHA (17JJ1A0548)

ABSTRACT

Mathematics plays an important role in everybody's life. We mainly depend on calculators available in our electronic gadgets to solve mathematical equations. Calculators available in these gadgets are so advanced that it can solve most complex mathematical equations. Remembering the fact that almost all electronic gadgets are touch screen based now-a-days, developing a system that recognizes and solves mathematical equations from handwriting is a potential area of research.

This work develops an automatic equation solver, which solves the mathematical equations written by different persons. The data set is downloaded from the Kaggle website. Our system is trained with that dataset consisting of Numbers,Symbols and Alphabets with the Accuracy of around 98%. Convolution Neural Networks(CNN) is used for recognition of the symbols and alphabets.

The system is successful in classifying and solving Arithmetic Equations and polynomial equations of power up to degree 9. Finally the experimental results show the great effectiveness of our proposed system.

Contents

1.INTRODUCTION

1.1 PROJECT OVERVIEW

Due to the swift improvement in computer technology and internet technology, most of the documents, books and literatures in the area of computer science as well as others are increasingly becoming digitalized. Mathematics is broadly used in almost all areas of science, such as physics, engineering, medicine, economics, etc. Digital document analysis and understanding is the major research concern today. For the recognition of English characters and numbers in electronic books OCR (optical character recognition) can attain higher recognition exactness. Handwritten mathematical expression recognition is still a most challenging job to do in the area of computer vision. Due to the 2-dimensional nesting assembly and different sizes, the correction rate of symbol segmentation and recognition still cannot achieve its actual requirements.

The primary task for the recognition of mathematical expression is to segment the character and then classify those character. An operation that seeks to decompose an image of a sequence of characters into sub images of individual symbols is Character segmentation. Histogram grounded projection analysis is a well-known method and is commonly used in the segmentation of line and character from the image and also, we can practice it at some detection stage.

Convolutional neural network (CNN) is one of the mostly used classification model in computer vision area. In the last few years, deep Convolutional Neural Network (CNN) leaning has provedthe outstanding performance in the field of image classification, machine learning and pattern recognition. Above all existing model, CNN is one of the most popular models and has been providing the state-of-the-art recognition accuracy on object recognition, segmentation, human activity analysis, image super resolution, object detection, scene understanding, tracking, and image captioning. For the task of image classification CNN outperforms above all the previous classification method. CNN extract feature from the image by a series of operations.

In this paper a CNN model based handwritten mathematical equation solver is proposed. In this system Handwritten mathematical equation contained in an image evaluated by correctly recognising the alphanumeric characters and operators in the handwritten equation and solved using Math libraries of Python. The system consists of two parts:

(I) Recognition of mathematical equation in an image

(II) Solving the recognised mathematical equation.

In recognition stage, back propagating neural network is used for recognition of characters and symbols in the mathematical equation. The system recognises alphabets, numbers and mathematical symbols. The system solves polynomial equation and simple math equations consisting of arithmetic operations.

1.2 PROBLEM STATEMENT

The situational problems that exist with current similar implementation of Handwritten Equation Recognizer have numerous problems like not being based on pure machine learning which won't be able to take advantage of the huge processing power of the machines that exist in today's world, and not having enough data to accurately predict the characters that are extracted from the image being fed into the software. Additionally, using conventional way of computing equations are also not productive due to requiring huge amount of manual input from the user.Further problems encountered in manual input is that due to complex nature of certain problems there can occur user error during manual input of data inside a calculator. This causes discrepancies and errors in the obtained data and can cause devastating effect on some fields. Handwriting Equation Recognizer solves this problem by requiring very less.

1.3 OBJECTIVES

To develop an application that is able to predict and solve handwritten Arithmetic and Polynomial equations from the given image.

1.3.1 PRIMARY OBJECTIVES

- To recognize characters and symbols from the Handwritten image.

- To solve the polynomial equations in the image.
- To solve the arithmetic equations in the image.
- To completely utilize the modern-day computational power.

1.4 SCOPE

After this project work will have completed, the software can be used to convert mathematical equations into computer readable form which in turn can be written directly down for further calculations or simply data entry in any software solutions. 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 in multiple language 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 solution which will certainly be helpful for 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 the result of this project even complex solutions of equations can be solved in matter of seconds.

Hence after completion of this project, it can be released 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 project

2. LITERATURE REVIEW

Several works have been done so far for handwritten character recognition. A scheme for mathematical expression recognition (MER) based on SVM and projection histogram for simple ME's have been proposed. which is a part ofoffline handwritten expression recognition. This paper basically focusses on numerous techniques used for feature extraction and recognition. An effective and robust system for recognition of printed and handwritten mathematical characters have been proposed by Zanibbi et. al. (2002). Which evaluates an expression by using three successive passes i.e., the layout pass, lexical pass and operator tree. This tree manipulation which used in each pass can be represented by tree transformation. Bage et.al. (2013) has proposed recognition system for offline handwritten mathematical symbol. For feature extraction, shape of the character is considered. The proposed system is based on relative study of feature extraction methods. Recognition has been carried out via SVM. Ramteke et.al. (2012) have transformed scientific and engineering documents into electronic form by bearing in mind of printed mathematical symbols and expressions. Features are computed using centroid of the image here and for segmentation bounding box technique is used. Classification has been carried out using neural network approach which recognition rate of 90%. Ahmad-Montaser Awal et al (2010) discussed some issues related to the problem of online mathematical expression recognition. Ha. et. al. (1995, August) has proposed a system that understand mathematical expression from the printed document images. In the development of this system, they have adapted object-oriented methodology to describe the data abstraction for the hierarchical structure of the mathematical expression is given in the form of the expression tree. Pradeep et. al. (2010) has proposed diagonal feature extraction technique for the handwritten character by using feed-forward neural network algorithm. This technique uses diagonal, horizontal and vertical features for the classification purpose.In recent years, Convolutional Neural Network (CNN) has made a series of revolution research results in the fields of image classification and detection. The past few years a branch of artificial neural networks called deep learning has shown great potential in solving classification problems. The field of deep learning started gaining popularity in 2012 when Alex et al. demonstrated their deep network architecture named Alex Net for image classification which outperformed.

3. SYSTEM ANALYSIS

3.1 PROJECT FEATURES

The features can be considered as the independency, Ease-to-use, Interactive & Responsive.

a. INDEPENDENCY

b. EASE-TO-USE

c. INTERACTIVE &RESPONSIVE

Independency can be viewed as the extent to which the project runs on different devices i.e., on various configurations of devices without any support from third party helping tools or libraries and if support is must from third party, then minimizing this support depends on the extent of independency of a project. However, this project is dependent to some extent on other libraries like pandas, OpenCV, etc.but these are third party libraries instead they provide very important functionalities to the project.

Ease-to-Use is the feature where the end user can feel comfortable in making use of this project. As this is a project not a product, implementation, functionality and efficiency are considered instead of user look and feel.

As above said, Handwritten math Equation Solver is a project not a product and hence the instructiveness and responsiveness are considered to a little extent. Of course, that do matter and these features like ease-to-use, interactive and responsiveness are provided by Flask which the implementation platform of this project.

3.2SYSTEM REQUIREMENTS

3.2.1 SOFTWARE REQUIREMENTS

The main system requirements for the Handwritten Equation Recognizer using Convolution Neural Network are mentioned below:

(I) A CUDA (Compute Unified Device Architecture) Application Programming Interface which will need to provide GPU (Graphical Processing Unit) accelerated neural network for high performance and high-speed model(optional).
(II) An Anaconda3 installation with Python interpreter for actual code interpreting and important machine learning and deep learning libraries.
(III) A labelled dataset consisting of the numbers and characters used in mathematical equations to train the model for accuracy.
(IV) Various Additional Libraries that are used for programming. For this project we used NumPy, Scikit-Learn, Open-CV2, and Matplotlib.

Additionally, the above requirements are for training the model, for the implementation phase We need JavaScript, HTML, CSS and Flask.

3.2.2 HARDWARE REQUIREMENTS

•Laptop,Desktop	: I3 and above configuarion,2.7GHZ
•RAMCapacity	:8GB
•HardDisk	: 10GB
•GPU	: AMD or Nvidia 2gb(optional)

3.2.3 NON-FUNCTIONALREQUIREMENTS

Non-Functional Requirements are those which describes the threshold capability of the project for functioning without any breaks or errors. They are usually performance and safety requirements.

Performance Requirements are those which increases the accessing, processing speed of application. It is the time within which the output is shown and also the size of the input it accepts.

Safety Requirements the safety requirements to be satisfied in such a way that no other file formats can be selected other than dataset. It only implicitly allows the datasets like images of symbols and characters in csv, excel etc. files avoiding the rise of file format exceptions.

3.2.4 FUNCTIONALREQUIREMENTS

Functional requirements are those which clearly describes the functions that the application should perform to ensure the specifications.

Input Handling: Project is enough strong to handle all the types of pixel values in files like csv, excel etc.

Output: Output of the project is solving the Handwritten Math Equation in any given image.

Data Preprocessing: Data preprocessing is done by using the OpenCV2 module by converting the input image into greyscale format and finding the contours from the image.

Model: Here we used Convolution Neural networks (CNN) for classifying the characters and the symbols.

3.3 FEASIBILTY STUDY

A Feasibility Study is a research, testing and experimentation designed to determine if a strategy, design, product or process is possible and practical. The attributes that need to be considered are:

Cost:

Cost would Comprise of the development cost, performance cost, deployment and maintenance cost. In this case, development and performance cost are considered. The time needed to implement this project is 1 month and during execution of the project, the time needed to build the model and process the input and gives output within 2-3 minutes. Performance Cost would depend on the processing speed of GPU where the minimum is 2.7 GHZ and also on the RAM whose min capacity is 8GB.

Technology:

The technology used to build this project is available for open source and is very feasible to construct. It is the python programming language with Deep Learning technology, especially using CNN model. Other libraries used are OpenCV, pandas, NumPy and Math which are also in python language.

4. IMPLEMENTATION TOOLS

4.1 PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning Python:

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Applications of Python

As mentioned before, Python is one of the most widely used language over the web. I'm going to list few of them here:

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.

- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

4.2 ANACONDA PACKAGE

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage Conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages

4.2.1 JUPYTER NOTEBOOK

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

4.2.2 SPYDER

Spyder Is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython,as well as other open-source software. It is released under the MIT license.

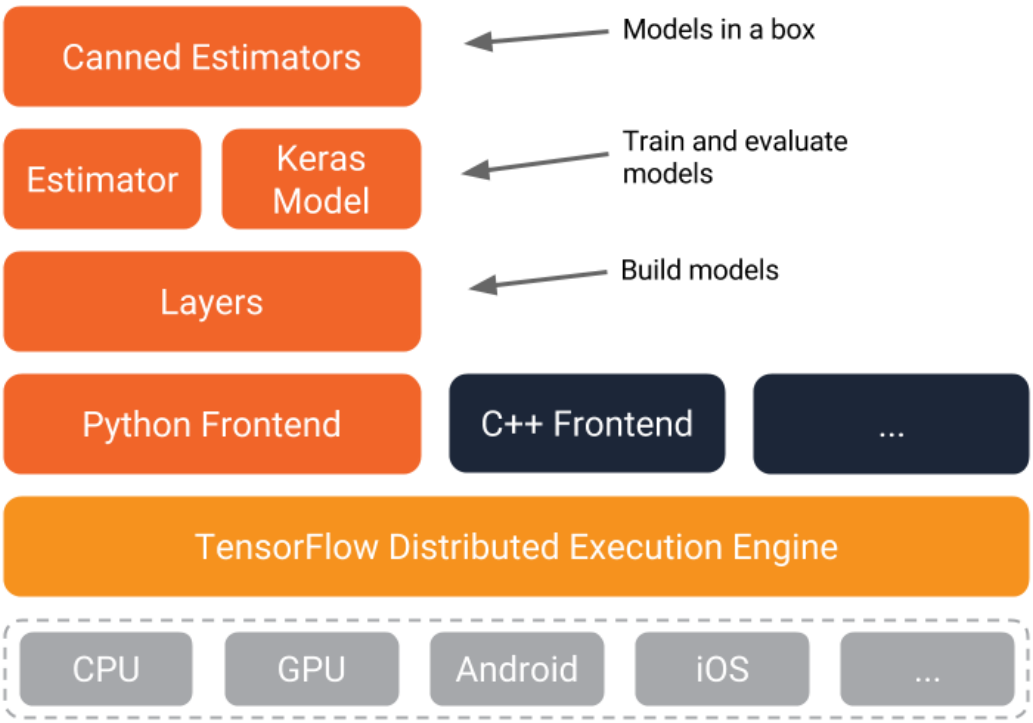
4.3 TENSORFLOW 2.0

TensorFlow is an open-source software library for numerical computation using data-flow graphs. It was originally developed by the Google Brain Team within Google's Machine Intelligence research organization for machine

learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

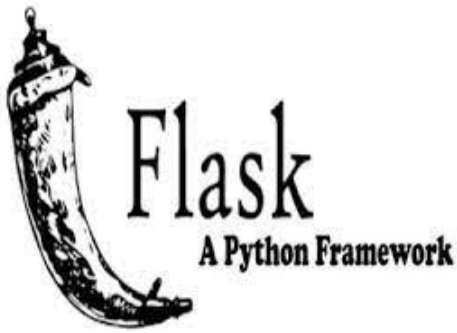
TensorFlow is cross-platform. It runs on nearly everything: GPUs and CPUs—including mobile and embedded platforms—and even tensor processing units (TPUs), which are specialized hardware to do tensor math on. They aren't widely available yet, but we have recently launched an alpha program.

The TensorFlow distributed execution engine abstracts away the many supported devices and provides a high performance-core implemented in C++ for the TensorFlow platform.



On top of that sit the Python and C++ frontends (with more to come). The Layers API provides a simpler interface for commonly used layers in deep learning models. On top of that sit higher-level APIs, including Keras (more on the Keras.io site) and the Estimator API, which makes training and evaluating distributed models easier.And finally, a number of commonly used models are ready to use out of the box, with more to come.

4.4 FLASK



Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

4.5 WEB BROWSER

A web browser (commonly referred to as a browser or internet browser) is an application software for accessing the World Wide Web. When a user requests a web page from a particular website, the web browser retrieves the necessary content from a web server and then displays the page on the user's device.

A web browser is not the same thing as a search engine, though the two are often confused. A search engine is a website that provides links to other websites. However, to connect to a website's server and display its web pages, a user must have a web browser installed.

The most commonly used browsers are Google Chrome ,Firefox, Internet Explorer ..etc.

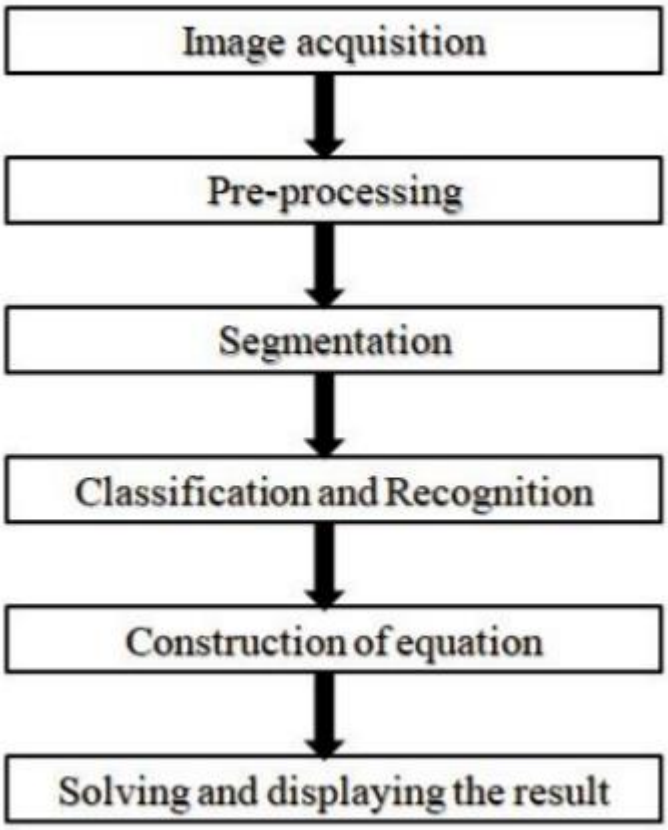
5. METHODOLOGY

5.1 ABOUT DATASET

The Dataset consists images of Handwritten Math Symbols and numbers from 0-9 and also the arithmetic operators. In Each folder an average of 10000 images are collected from different sources. This data set is downloaded from the Kaggle.

5.2 HANDWRITTEN MATH EQUATION SOLVER

The system architecture of handwritten mathematical equation solver using convolution neural network is:



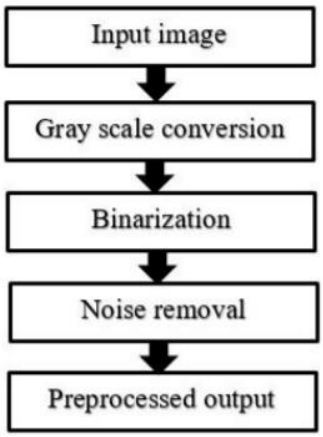
5.2.1 IMAGE ACQUISITION

In this project we are solving a mathematical equation written on any surface by capturing an image of it and then processingthe same. The image is captured using electronic devices like digital cameras, mobile cameras or

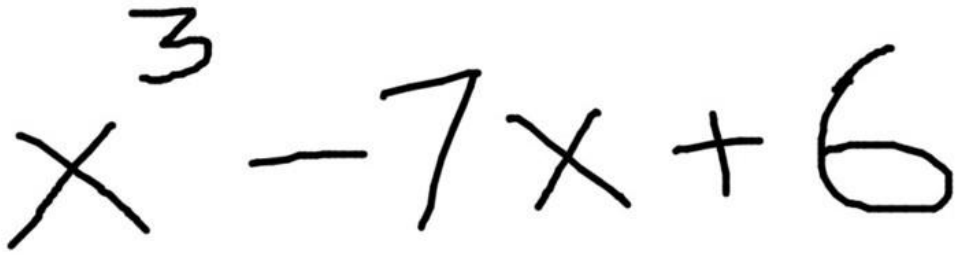
webcam. It has to be ensured that the captured image is not blurred, because a blurred image can adversely affect the performance of the entire system.

5.2.2 PRE-PROCESSING

In this stage the acquired image is processed further to make character recognition process easier. The image is enhanced to make it more suitable for segmentation. The block diagram of pre-processing stage is:



Input image:



Gray Scale Conversion:

The first and the most important step in image pre-processing is Gray scale conversion [14]. If the Gray bitmap is Y and colour bitmap is R, G and B then the formula we used is:

Y = 0.114R + 0.587G + 0.299B

Binarization:

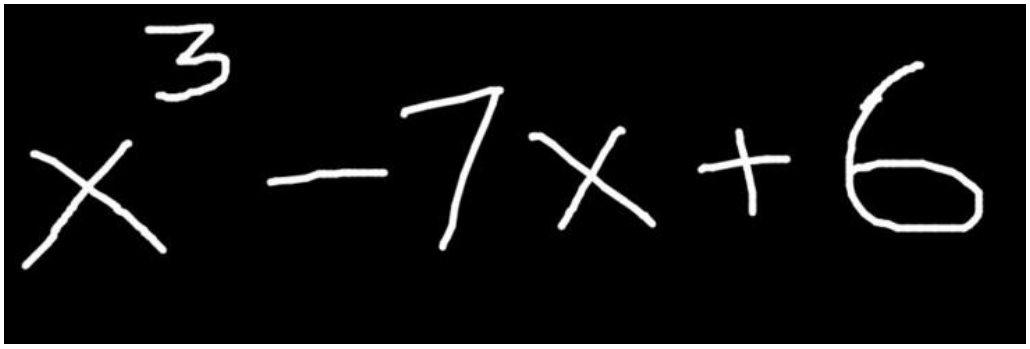
In the binarization stage the Gray scale image is converted into black and white image in which the pixel is having only two values 0 or 1. For acquiring good binarization a suitable threshold is to be selected

depending on the image characteristics, here we are using Matlab Gray threshold library. Otsu's binarization [19] which is one of the most popular binarization methods is used in this paper. It automatically selects a global threshold value based on the shape of the histogram. The goal is to find the threshold value with the minimum entropy for sum of foreground and background.

Noise removal:

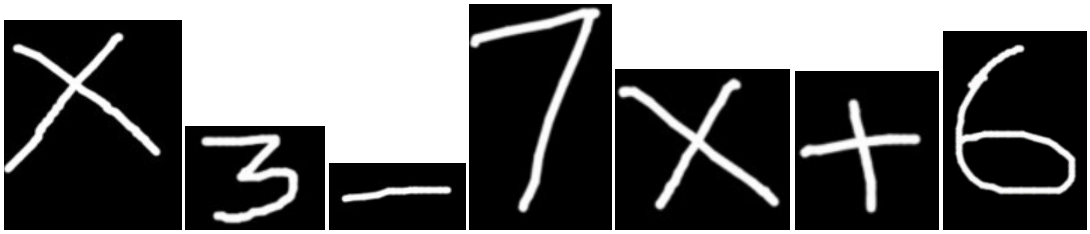
In this stage cleaning of the black and white image is taking place. The components which have more than 100 pixels are removed to remove the unwanted spots from the image.

Pre-processed output:



5.2.3 SEGMENTATION

In the segmentation stage, an image of mathematical equation is decomposed into sub-blocks of individual character. In this stage the pre-processed image is scanned for connected objects and the identified connected components are assigned with special labels. Each labelled portion containing characters is segmented using bounding box method. These bounding boxes are called contours. In bounding box method, the labelled regions are enclosed with rectangular boxes. The corner co-ordinates, height and width of the labelled connected region is determined by the bounding box. The bounding boxes always specifies the boundaries of connected objects. Filled bounding boxes which completely relay on covering the corresponding connected objects and also measures the image region properties.



5.2.4 CLASSIFICATION AND RECOGNITION

Classification is one of the most active research and application area of neural networks. Classification and recognition is done using convolution neural network.

In this project convolutional neural network is used as a classification model, 32×32 grey 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 conv layer, The persistence of this layer is to introduce nonlinearity to a system. 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. Calculation of the output size, O of a pooling layer with input size ,I Pooling filter size Padding ,P and stied ,S has done by

$$O = (I – F + 2×P) / S + 1$$

In our proposed method we use 2×2 pool size and the output of the pooling layer is 14×14. Random selection of input elements to zero with a given probability is done in a dropout layer. It is a simple way to prevent the overfitting in the neural network. Overfitting is a stern difficulty in such networks. Large networks are also sluggish to use, making it more challenging to contract with overfitting by combining the predictions of many different large neural nets at the test time. The technique for addressing this problem is Dropout. The core idea is to randomly drop units (along with their connections) from the neural network during training [23]. In our proposed method we use the probability 0.2 at the dropout layer during the training. After the convolutional and down sampling layers one or more fully connected layers is used. The layer in which the neurons associate to all the neurons in the

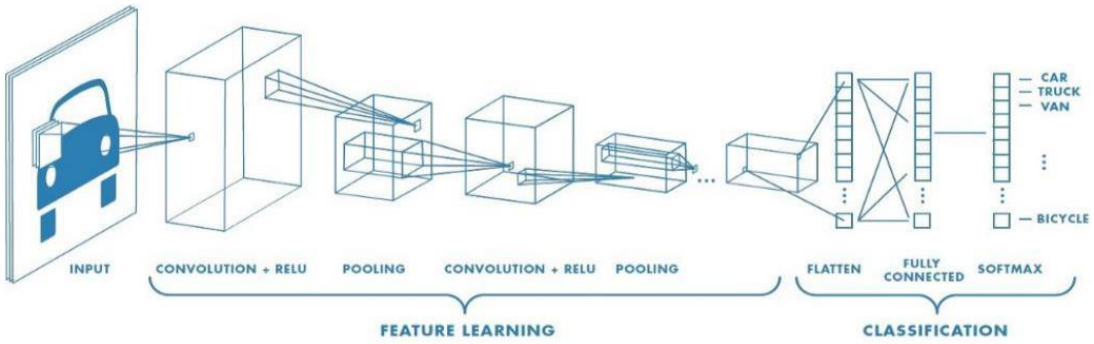
previous layer is fully connected layer. Fully connected layer combines all the features learned by the previous layers through the image to recognize the larger patterns. To classify the images the last fully connected layer combines the features. Hence, the Output parameter in the last fullyconnected layer is equal to the number of classes in the target data. In our work, the output size is 14, corresponding to the 14 classes. An activation function SoftMax which normalizes the output of the fully connected layer is also used in the convolutional neural network model. The SoftMax layer output comprises of positive numbers that sum to one, in the nest that can be used as the classification probabilities at the classification layer. At the final layer, classification layer uses the probabilities given by the SoftMax activation function to find the input image classes and find the loss by comparing it with the pre assigned ground truth classes.

WORKING OF CNN

In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Object's detections, recognition faces etc., are some of the areas where CNNs are widely used.

CNN image classifications take an input image, process it and classify it under certain categories (E.g., Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see h x w x d (h = Height, w = Width, d = Dimension). E.g., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of grayscale image.

Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply SoftMax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.



Example of CNN

Convolution Layer

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

- An image matrix (volume) of dimension **(h x w x d)**
- A filter **(f_h x f_w x d)**
- Outputs a volume dimension **(h - f_h + 1) x (w - f_w + 1) x 1**

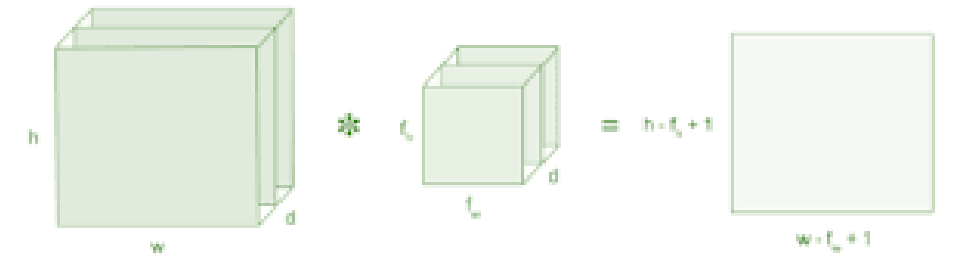
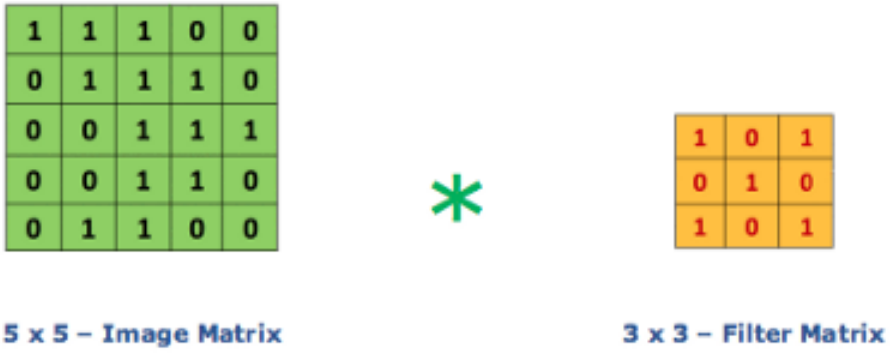
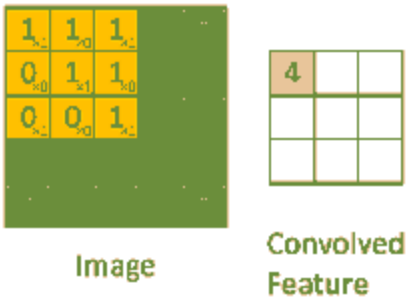


Image Matrix

Consider a 5 x 5 whose image pixel values are 0, 1 and filter matrix 3 x 3 as shown in below



Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 filter matrix which is called **"Feature Map"** as output shown in below



Strides

Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on.

Padding

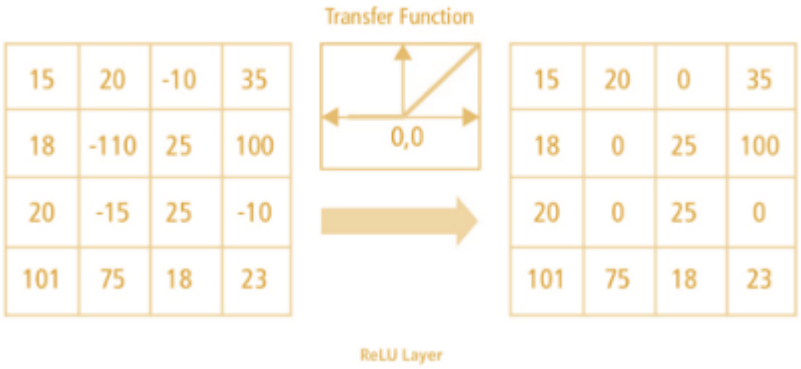
Sometimes filter does not fit perfectly fit the input image. We have two options:

1. Pad the picture with zeros (zero-padding) so that it fits

2. 2 Drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image.

Non-Linearity (ReLU)

ReLU stands for Rectified Linear Unit for a non-linear operation. The output is ƒ(x) = max (0, x). Why ReLU is important: ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real-world data would want our ConvNet to learn would be non-negative linear values.



There are other non-linear functions such as tanh or sigmoid that can also be used instead of ReLU. Most of the data scientists use ReLU since performance wise ReLU is better than the other two.

Pooling Layer

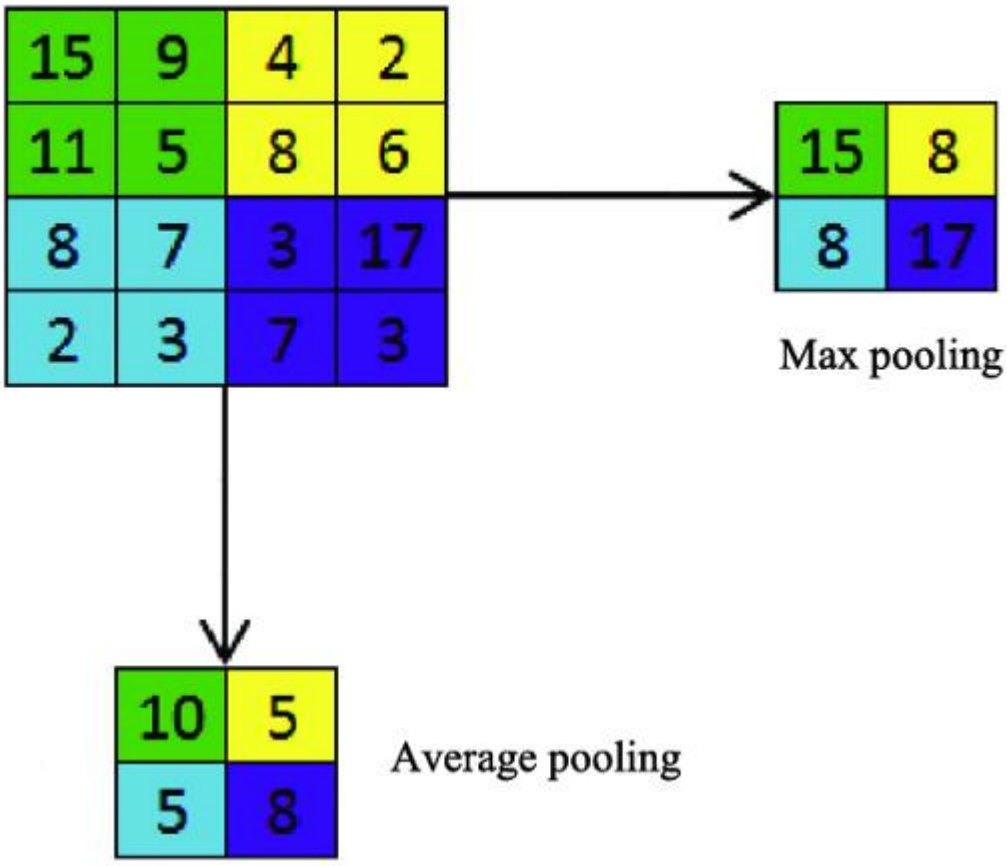
Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or down sampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:

1. Max Pooling

2. Average Pooling

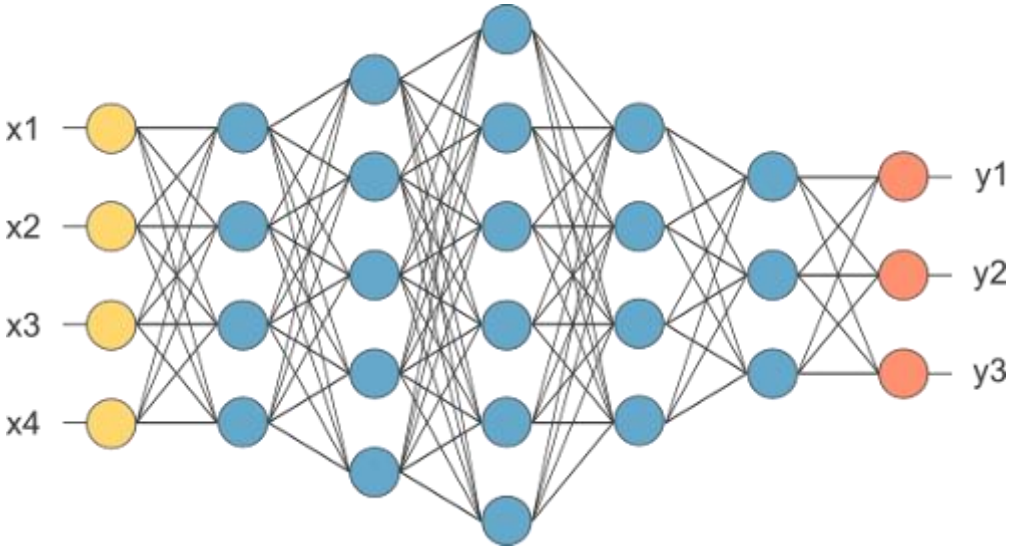
3. Sum Pooling

Max pooling takes the largest element from the rectified feature map. Taking the average of elementsis known as average pooling. Sum of all elements in the feature map call as sum pooling.

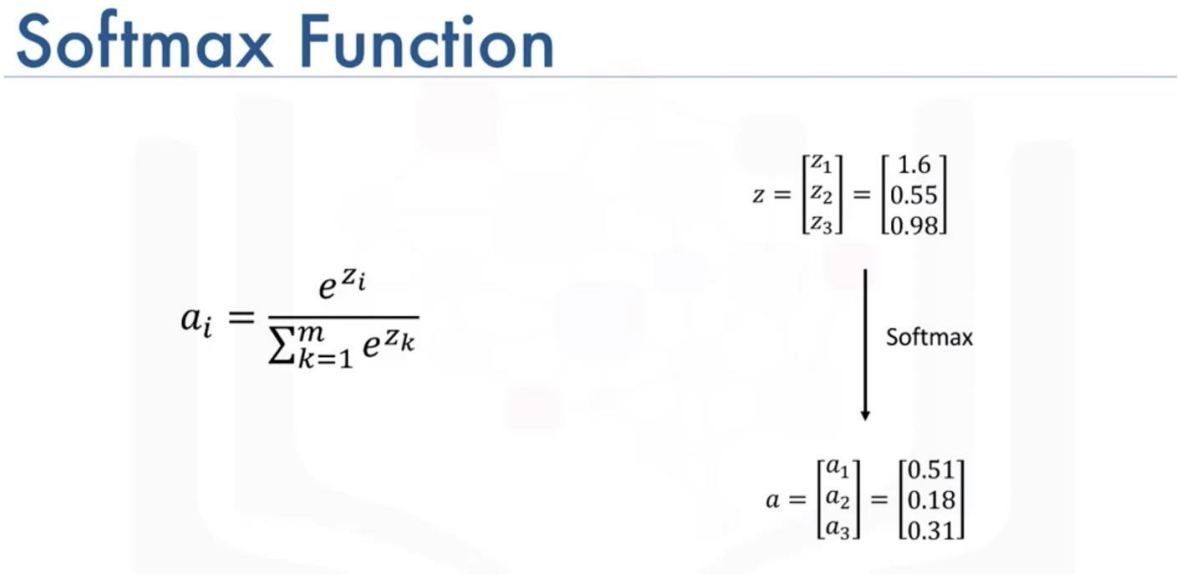


Fully Connected Layer

The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like a neural network.



SoftMax Function



The SoftMax function is ideally used in the output layer of the classifier where we are actually trying to get the probabilities to define the class of each input. So, if a network with 3 neurons in the output layer outputs [1.6, 0.55, 0.98] then with a SoftMaxactivation function, the outputs get converted to [0.51, 0.18, 0.31]. This way, it is easier for us to classify a given data point and determine to which category it belongs.

Finally, the object is classified which gets the highest probability that object is classified.

5.2.5CONSTRUCTION OF EQUATION

In this stage the actual handwritten mathematical equation is reconstructed using the characters and operators identified during the recognition stage. Each character and operators are placed at their actual positions according to their labelling. If the contour or bounding box is after the variable without any operator in between then it is considered as the power of that variable and if it is before it is considered as coefficient of that equation.

5.2.6 SOLVING AND DISPLAYING THE RESULT

After the recognition of operators and operands in the equation contained in the image, the equation is to be solved. Solving the mathematical equation is

done using Matlab mathematical toolbox. The system has successfully solved the system of linear equations with two and three variables and also polynomial equation of degree up to 9.

Type 1: polynomial equation:

$a_n x\ n + a_{n-1} x\ n-1 + \ldots\ldots + a_2 x\ 2 + a_1 x\ 1 + a_0 x\ 0 = 0$

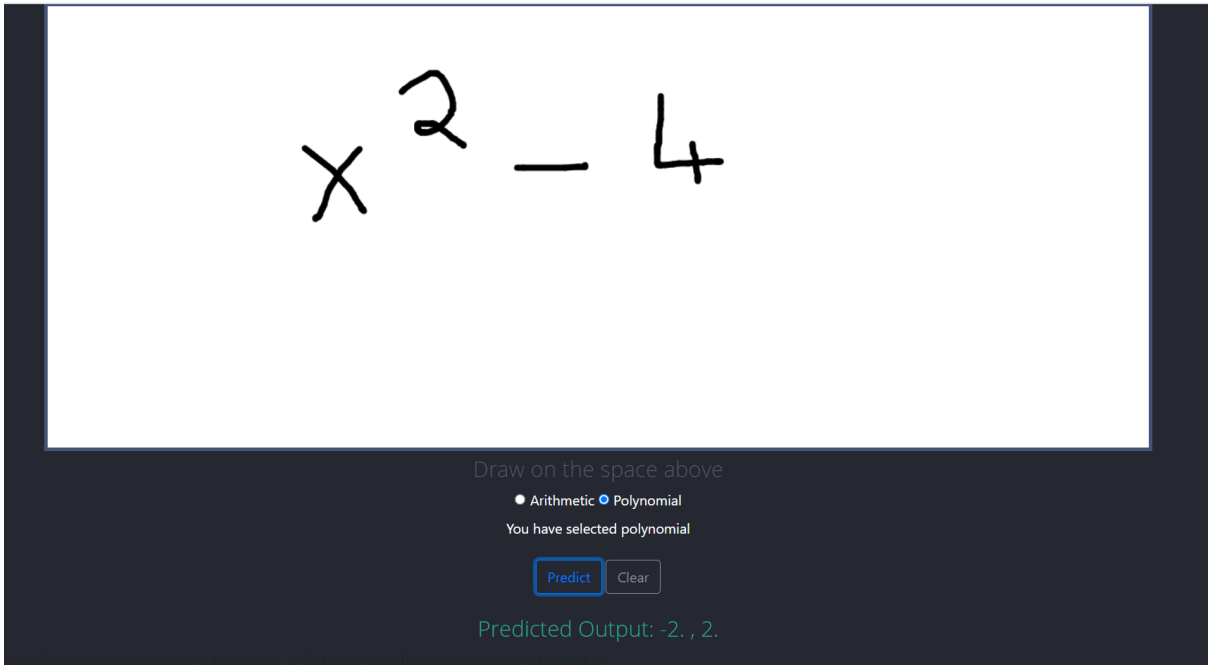
where an, an−1, ... a0 are the coefficients of the polynomial equation. The coefficients of the polynomial equation are extracted and solved using roots (coefficients) function in the NumPy.

Type 2: arithmetic expression:

7+8*3

Arithmetic expressions are solved using the eval() inbuilt function in the python language.

Sample result of an polynomial equation



6 CODE IMPLEMENTATION

6.1 DATA PREPAIRATION

In [1]:
```
import numpy as np
import cv2
from PIL import Image
from matplotlib import pyplot as plt
%matplotlib inline
import os
from os import listdir
from os.path import isfile, join
import pandas as pd
```

In [5]:
```
def load_images_from_folder(folder):
    train_data=[]
    for filename in os.listdir(folder):
        img = cv2.imread(os.path.join(folder,filename),cv2.IMREAD_GRAYSCALE)
        img=~img
        if img is not None:
            ret,thresh=cv2.threshold(img,127,255,cv2.THRESH_BINARY)
            ret,ctrs,ret=cv2.findContours(thresh,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
            cnt=sorted(ctrs, key=lambda ctr: cv2.boundingRect(ctr)[0])
            w=int(28)
            h=int(28)
            maxi=0
            for c in cnt:
                x,y,w,h=cv2.boundingRect(c)
                maxi=max(w*h,maxi)
                if maxi==w*h:
                    x_max=x
                    y_max=y
                    w_max=w
                    h_max=h
            im_crop= thresh[y_max:y_max+h_max+10, x_max:x_max+w_max+10]
            im_resize = cv2.resize(im_crop,(28,28))
            im_resize=np.reshape(im_resize,(784,1))
            train_data.append(im_resize)
    return train_data
```

We get the data in the form of the images for every character 0-9 and the symbols +,-,* and variables X are to be pre-processed so we extract the pixel values from the that images and save them into the csv file so that it is easy to process the csv and load values in the 2d format.

We append the pixel values after pre-processing the image finding the contours and changing into greyscale image and append the values into the csv file.

We also append the label values of the images if it id 1 or 0 and for +,- we give 11,12 ..etc.

The after creating the pixel values we save in the csv file so that we can use it reding with the pandas library.

6.2 TRAINING THE MODEL

```
In [2]: import pandas as pd
        import numpy as np
        import pickle
```

```
In [3]: df_train=pd.read_csv('train_final.csv',index_col=False)
        labels=df_train[['784']]
        df_train.shape
```

```
Out[3]: (183211, 785)
```

```
In [4]: df_train.drop(df_train.columns[[784]],axis=1,inplace=True)
        df_train.shape
```

```
Out[4]: (183211, 784)
```

```
In [5]: np.random.seed(1212)
        import keras
        from keras.models import Model
        from keras.layers import *
        from keras import optimizers
        from keras.layers import Input, Dense
        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
```

```
In [6]: labels=np.array(labels)
        from keras.utils.np_utils import to_categorical
        cat=to_categorical(labels,num_classes=14)
```

First and foremost, load the csv file containing the pixel values of the characters and the symbols. By importing Pandas we can read the csv then we separate the label values .

Import all the libraries and modules needed for the training of the model such as Kera's models, Kera's layers for building the CNN model.

After that we convert the label values to categorical this function converts the labels in the form of binary numbers 0,1 so as to ease the computation of the model we use this function.

```
In [9]: l=[]
        for i in range(len(df_train)):
            l.append(np.array(df_train[i:i+1]).reshape(1,28,28))
```

```
In [10]: l = np.array(l)
         l = np.reshape(l, (-1, 28, 28, 1))
```

Then we resize the input pixel values into 2d matrix of size 28x28 so that we apply the CNN model the convolution operation takes place.

```
model = Sequential()
model.add(Conv2D(first_conv_num_filters, first_conv_filter_size, input_shape=(28, 28, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=pool_size))
model.add(Conv2D(second_conv_num_filters, second_conv_filter_size, activation='relu'))
model.add(MaxPooling2D(pool_size=pool_size))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(50, activation='relu'))
model.add(Dense(14, activation='softmax'))


model.compile( optimizer='adam', loss='categorical_crossentropy',metrics=['accuracy'],)
```

In [12]:
```
model.fit(l, cat, epochs=10, batch_size=200,shuffle=True,verbose=1)
```

```
Epoch 1/10
917/917 [==============================] - 58s 45ms/step - loss: 1.2562 - accuracy: 0.7444
Epoch 2/10
917/917 [==============================] - 41s 45ms/step - loss: 0.1627 - accuracy: 0.9471
Epoch 3/10
917/917 [==============================] - 41s 45ms/step - loss: 0.1146 - accuracy: 0.9602
Epoch 4/10
917/917 [==============================] - 43s 47ms/step - loss: 0.0943 - accuracy: 0.9659
Epoch 5/10
917/917 [==============================] - 43s 47ms/step - loss: 0.0824 - accuracy: 0.9699
Epoch 6/10
917/917 [==============================] - 43s 47ms/step - loss: 0.0702 - accuracy: 0.9738
Epoch 7/10
917/917 [==============================] - 44s 48ms/step - loss: 0.0625 - accuracy: 0.9760
Epoch 8/10
917/917 [==============================] - 44s 48ms/step - loss: 0.0561 - accuracy: 0.9788
Epoch 9/10
917/917 [==============================] - 44s 48ms/step - loss: 0.0517 - accuracy: 0.9804
Epoch 10/10
917/917 [==============================] - 43s 47ms/step - loss: 0.0478 - accuracy: 0.9821
```

Out[12]:
```
<keras.callbacks.History at 0x233eaa6d4f0>
```

Then we build the model by adding 2 convolutional layers ,2 Max pooling layers ,1 dropout layers , 2 dense networks with activation function Relu containing 128,50 nodes and lastly the SoftMax function.

The above mentioned layers help us building the CNN model easily.

These layers as mentioned in the Methodology create the model and we compile the model by giving the Hyperparameters.

As we train the model we got the Training accuracy of 98.21%

```
In [13]: model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 24, 24, 30)        780
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 12, 12, 30)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 10, 10, 15)        4065
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 5, 5, 15)          0
_________________________________________________________________
dropout (Dropout)            (None, 5, 5, 15)          0
_________________________________________________________________
flatten (Flatten)            (None, 375)               0
_________________________________________________________________
dense (Dense)                (None, 128)               48128
_________________________________________________________________
dense_1 (Dense)              (None, 50)                6450
_________________________________________________________________
dense_2 (Dense)              (None, 14)                714
=================================================================
Total params: 60,137
Trainable params: 60,137
Non-trainable params: 0
_________________________________________________________________
```

By using the summary method, we can know about the model what are the layers we created in the model.

- shape of the Cnv2d is 24x24 size
- dropout of 5,5
- Dense networks 128,50

This summary function helps us know the model in detail.

```
In [13]: model_json = model.to_json()
         with open("model.json","w") as file:
             file.write(model_json)
```

```
In [14]: model.save_weights("weights.h5")
```

And lastly, we save our model in json file and save the weights of the CNN model so that we don't need to train every time we can just load the weights of the model and predict the expression.

6.3 PREDICTING THE OUTPUT

6.3.1 INPUT AQUATION

In Input aquation we need to take image from the user so we created a sketch pad where he can write the equation on the sketch pad so we can save the image and predict the output.

For this we used HTML and JavaScript for creating sketch pad

HTML code:

```
<html>

<head>
    <title>index</title>

    <!--Bootstrap styling-->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css" integr
        crossorigin="anonymous">

    <!-- Our Own Custom Style Sheet -->
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='pagecss.css') }}">//"C:/Users/Anish/D
    <!-- Libraries needed -->
    <!-- jQuery library -->
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js" integrity="sha384-alpBpk
        crossorigin="anonymous"></script>
    <!--Bootstrap library-->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
</head>

<body>
    <div classs=" ll">
        <div class="container-fluid">
            <div classs="row">
                <!--Canvas and results floated to left side of page-->
                <div class="draw-area float-left col-5">
                    <div class="centered ">
                        <canvas id="canvas" width="1000" height="500"></canvas>
                    </div>
                    <h1 class="instructions">Draw on the space above</h1>
                    <div class="centered">
            <input type="radio" id="arithmetic" name="operation" onclick="ShowHideDiv()"/>
            <label style="color:white">Arithmetic</label>
            <input type="radio" id="polynomial" name="operation" onclick="ShowHideDiv()"/>
```

JavaScript code:

JavaScript code is used for creating the sketch pad

```javascript
// Gerard Naughton G00209309 index.js EmergingTechProject
// Code adapted from https://github.com/sleepokay/mnist-flask-app
(function() {
    var canvas = document.querySelector("#canvas");
    var context = canvas.getContext("2d");
    canvas.width = 1000;
    canvas.height = 500;

    var Mouse = {x:0, y:0};
    var lastMouse = {x:0, y:0};
    context.fillStyle = "white";
    context.fillRect(0, 0, canvas.width, canvas.height);
    context.color = "black";
    context.lineWidth = 7;
    context.lineJoin = context.lineCap = 'round';

    debug();

    canvas.addEventListener("mousemove", function(e) {
        lastMouse.x = Mouse.x;
        lastMouse.y = Mouse.y;

        Mouse.x = e.pageX - this.offsetLeft-33;
        Mouse.y = e.pageY - this.offsetTop-25;
    }, false);

    canvas.addEventListener("mousedown", function(e) {
        canvas.addEventListener("mousemove", onPaint, false);
    }, false);

    canvas.addEventListener("mouseup", function() {
        canvas.removeEventListener("mousemove", onPaint, false);
    }, false);

    var onPaint = function() {
```

Executable.py:

```python
    if num == "arithmetic":
        #print("Entered Aritha")
        operation = ''
        for i in range(len(img_data)):
            img_data[i] = np.array(img_data[i])
            img_data[i] = img_data[i].reshape(-1, 28, 28, 1)
            #print("Once")

            result = model.predict_classes(img_data[i])

            if result[0] == 10:
                operation += '-'
            elif result[0] == 11:
                operation += '+'
            elif result[0] == 12:
                operation += '*'
            elif result[0] == 13:
                operation += '*'
            else:
                operation += str(result[0])
       # print("String Found:", operation)
        res=eval(operation)
        return res
    else:
        operation=''
        #print("Entered poly")
        for i in range(len(img_data)):
            img_data[i] = np.array(img_data[i])
            img_data[i] = img_data[i].reshape(-1, 28, 28, 1)

            result = np.argmax(model.predict(img_data[i]),axis=-1)

            if result[0] == 10:
                operation += '-'
            elif result[0] == 11:
                operation += '+'
            elif result[0] == 12:
                operation += 'x'
            elif result[0] == 13:
                operation += 'x'
```

This file evaluates equation by predicting every character and if that input is arithmetic or Polynomial then solves and returns the output to the flask and in turn flask displays in HTML

Flask code:

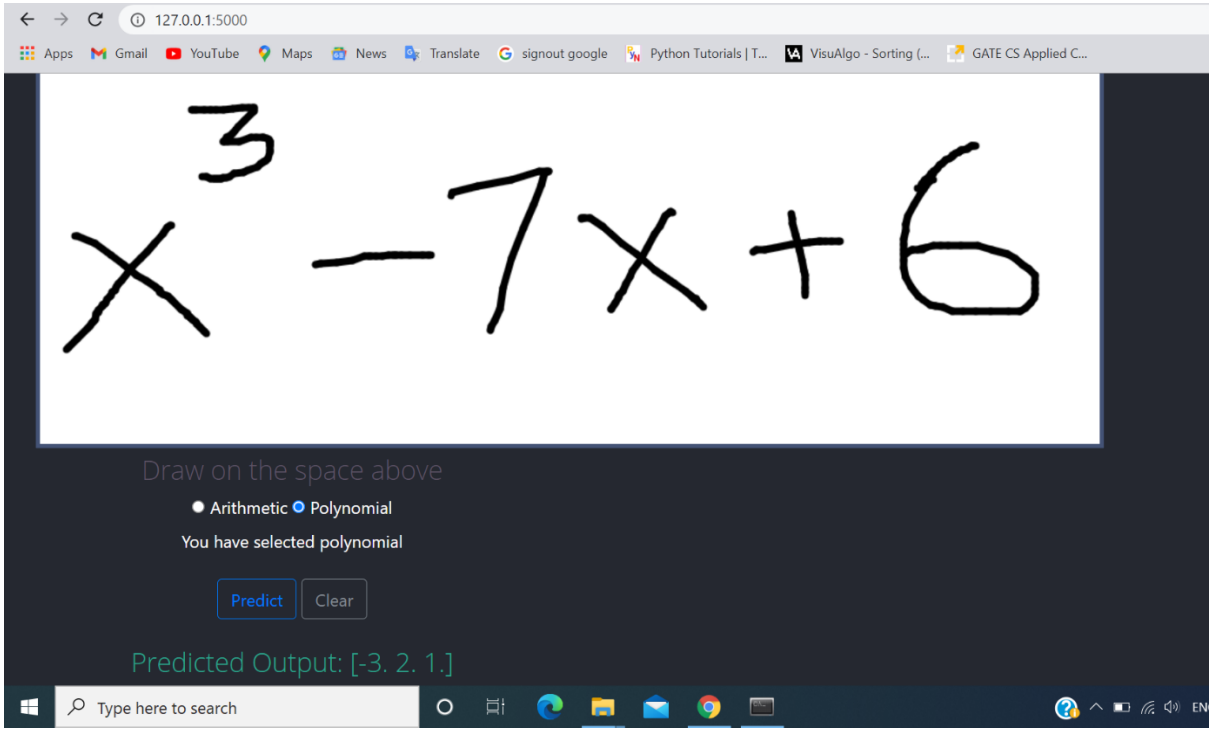
```
from flask import Flask, render_template, request
import base64
import io
from io import BytesIO
from PIL import Image
from Executablefile import predic
from keras.models import model_from_json
app = Flask(__name__)

@app.route('/')
def index():
    return render_template("interface.html")


@app.route('/predict/', methods=['GET','POST'])
def predict():
    oper = BytesIO(base64.urlsafe_b64decode(request.form['operation']))
    value = request.form['value']
    func(oper)
    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")
    res=predic(value,loaded_model)
    return str(res)
# Parsing Image function
def func(imgData):
    rawBytes = imgData.getvalue()
    rawIO = io.BytesIO(rawBytes)
    rawIO.seek(0)
    byteImg = Image.open(rawIO)
    byteImg.save('output.png', 'PNG')

if __name__ == '__main__':
    app.run(threaded=True)
```

With flask we integrate the model with input image from the user the home route renders interface.html page where user draws the equation and when clicks predict then we load our saved model and predicts the output on the local host of our device here external util function is used to pre-process the image and predicting the model.

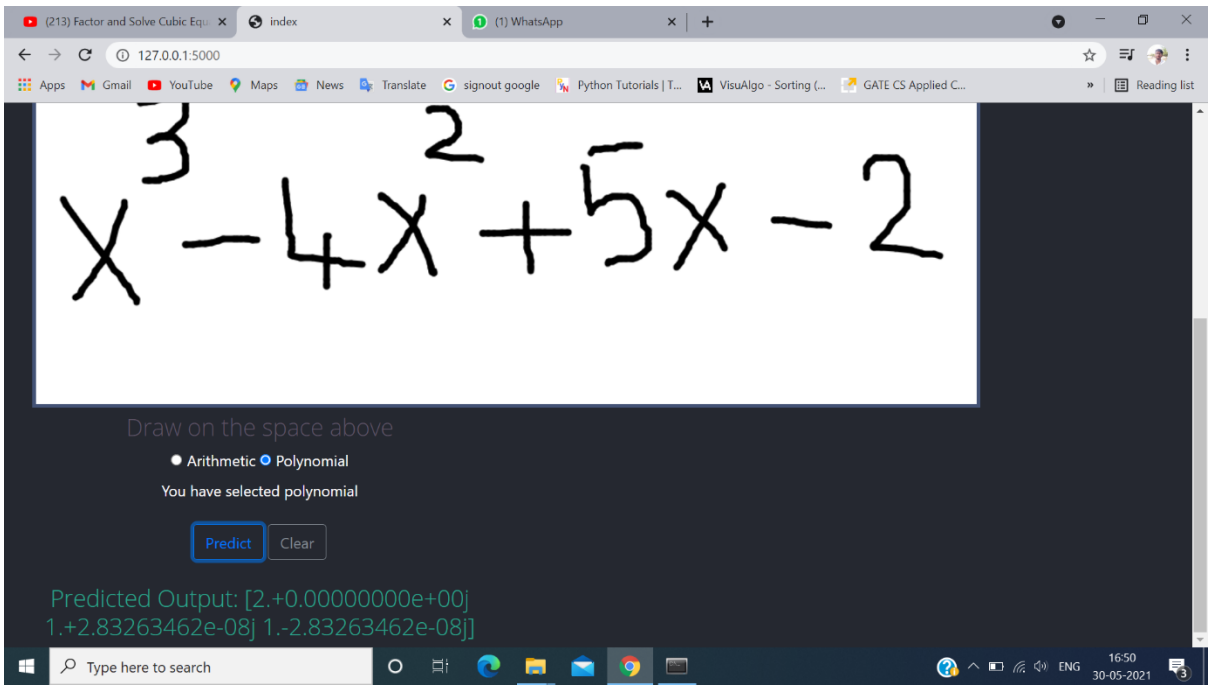
The output is shown on localhost.

7.RESULT

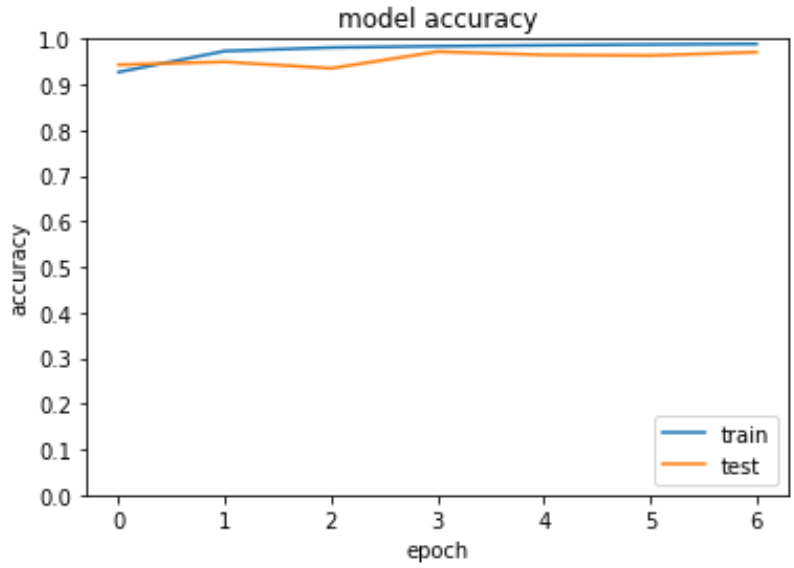
The result of sample equation x3-7x+6 is [-3,2,1]



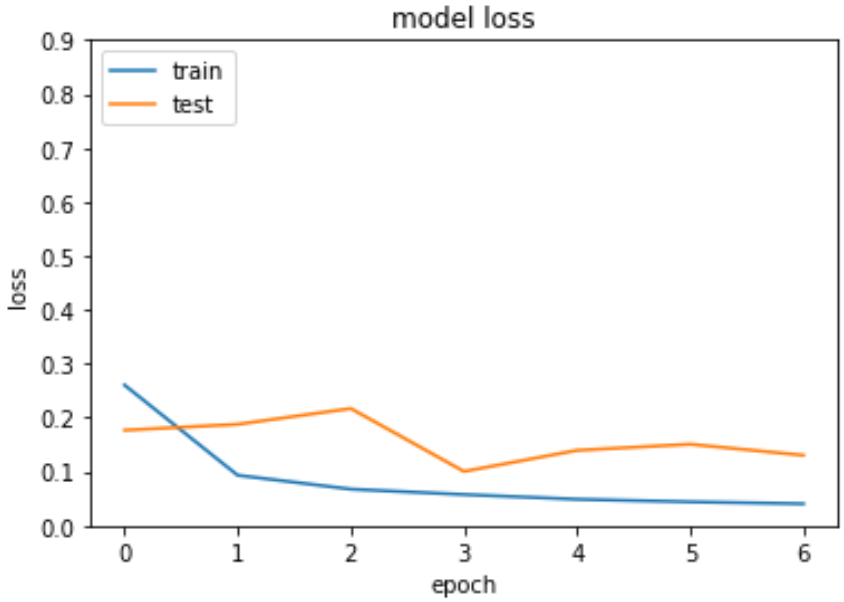
The result of sample equation x3-4x2+5x-2 is [2. +0j,1+2.83j,1-2.83j]



The accuracy curves of training and validation sets



The Loss curves are



8 FUTURE SCOPE

- Mathematics plays an important role in everybody's life. We mainly depend on calculators available in our electronic gadgets to solve mathematical equations Now a days all most all devices are Touch screen based Solving Math Helps in many ways
- In Digital Exams where aptitude and polynomial Equations plays key role solving quickly helps students
- In Online Learning where it helps teachers work easier and students too
- We can make Handwritten calculators for touch screen-based devices

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. The system is implemented and simulated using CNN model and performance is tested on real images.

9 REFERENCES

Books

- Russel, Stuart and Peter Norvig. Artificial Intelligence: A Modern Approach Second Edition. Dorling Kindersley. India. 2011.

Research Papers

- Gupta, Prachi, Neelam Pal, Lavanya Agrawal. Recognition of Handwritten Mathematical Equations. IMS Engineering College. India. 2017.

  Accessed on: 16th May, 2021 4:40 pm
- Mohapatra, Hitesh. Handwritten Character Recognition (HCR) using Neural Network. Veer Surendra Sai University of Technology. India. 2009.

  Accessed on: 24th May, 2021 8:00 am
- Lu, Catherine, Karanveer Mohan, Recognition of Online Handwritten Mathematical Expression Using Convolution Neural Networks. Stanford University.

  Accessed on: 24th May, 2021 8:00 am

Websites

- http://www.willforfang.com/computer-vision/2016/4/9/artificial-intelligence-for-handwritten-mathematical-expression-evaluation

  Accessed on: 24th May, 2021 7:55 am
- https://towardsdatascience.com/ Accessed on: 20th May, 2021 9:10 am
- https://en.wikipedia.org/wiki/Google_Translate Accessed on: 28th May, 2021 8:50 pm