

# Simple Equation Analyzer from Images

**Group Members:**

M.Haris Iftikhar (BSCS19069 )

Moazzam Ali (BSCS 19046)

Huzaifa Naseer (BSCS19074)

Zain-ul-Abidin (BSCS19082)

## PROBLEM STATEMENT:

The idea was to design a model which basically reads an equation from the image (a handwritten equation) and shows that equation in a textual form. The idea was to make the Artificial intelligence field more interesting in research and in experiments as well. The question was raised: How will it be beneficial for people around us? So Our world is mostly about researches based on mathematical equations formulas and many more etc on other hand the robotics field are future of our world so that's why scientist are working to make an idea robot which can more likely perform human tasks and increasing there intelligence rate upto 100% this model will helps Artificial intelligence field more advance that and equation is readed from the image and then in future it will be solved by a robot such idea was more liked by the people around in this field and is very much interesting as well. This model is somehow difficult as well because there are many mathematical formulas which are so difficult to be solved by human beings as well so to train a huge amount of data and then getting an accuracy of 100% is very difficult to attain. The question here raised that How will you solve a huge equation in a while? So that needs a huge amount of train data as well and the main task of this project was to get the data to perform various tasks.

## INTRODUCTION:

Our study works on the research of converting an image equation into a textual equation and which results in qualitative and quantitative research work. The study helps in the artificial intelligence field in many fields and has a very clear view about the computer vision fields and its progress in the modern world. Our idea contains a large amount of framework data in which the train data is very good and adaptive in many linear equations reading about 10 thousands of written models are trained so that the accuracy rate would reach maximum. Not any model is ideal but our research made it mostly the ideal one. In reading and equation form image and converting into a textual image and furthermore in future this textual equation can get solved in a very swift way. Our research contains the training of data and then using that trained data and converting the written into text in such a way that the

train data is used by the computer vision application library and hence perform a very efficient task. The idea comes through various researches in this field and many observations in artificial intelligence that there is a need for such a system to enhance the mathematical side of this field in a modern way.

## CONTEXT:

While working on how to train the model for analyzing the equation from image the following are the problems we faced.

- The libraries we used were tensorflow, keras and OpenCV for which we have to study the documentation for proper understanding.
- The models trained on such a dataset performed poorly in real world scenarios.
- So that's why we took a good dataset from research works.
- The computer vision based work was interesting and well acknowledged.
- The output equations are drawn from train data sets.

## LITERATURE REVIEW:

Mathematics is full of patterns by which the expressions are understood and conclusions are drawn. Every pattern consists of following datasets(variables ,operators and digits). When these three combined they form a unique pattern and this pattern is basically called an equation.

Here the operators are the certain functions which are to be performed and the final result is the solution of the mathematical equation which makes the equality true .

So here are some mathematical equations.

### Linear equation

Linear equations are those which contain either the constants and product of constants and sometimes a single variable as well; it is also called an algebra equation.

Example:  $(a)x + (b)y + (c) = 0$

### Quadratic Equation

The second degree equation in one variable with an exponent of 2 is known as a quadratic equation with an exponent of 2.

The general form:  $(a)x^2 + (b)x + (c) = 0, a \neq 0$

### Radical Equation

The equation which contains at least one variable in under the square root

Examples:  $\sqrt{(a)x + (b)} = (c)$

### Exponential Equation

An equation that has variables in the place of exponents.

Examples:  $a^b = 0$  Here "a" is base and "b" is exponent.  $(a)^x = b$

**KERAS:** keras is basically a high level neural network library and it contains two running algos TensorFlow and Theano. Keras was designed to perform very difficult tasks with a very high efficiency and a fast experimentation on neural networks.

*"Being able to go from idea to result with the least possible delay is key to doing good research"*

The keras allows you to do easy and fast prototyping. And it allows you to take multi inputs and multi output training and it works seamlessly on CPU AND GPU . Keras is basically a well designed model which contains the linear stack of layers.

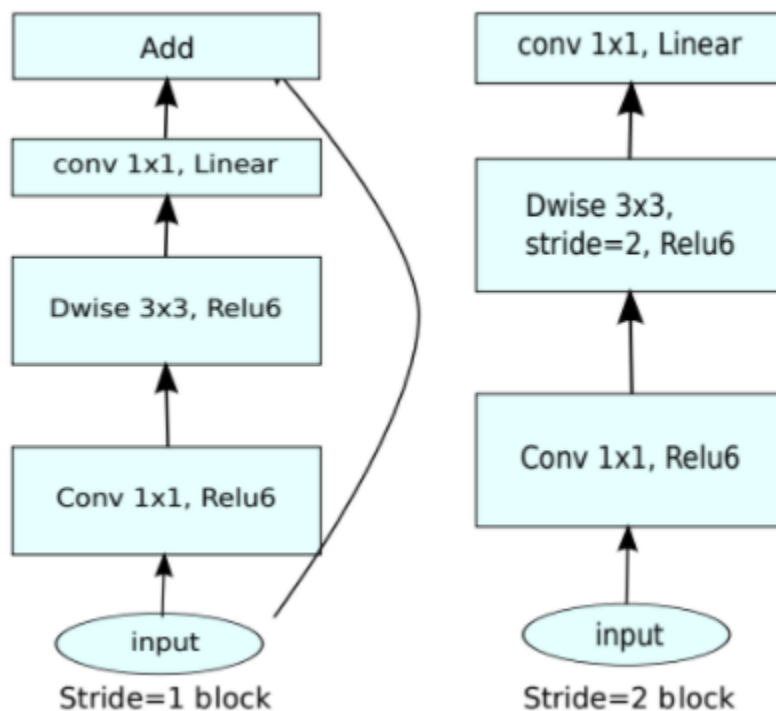
By default , keras will use TensorFlow as its tensor manipulation library.

**TENSOR FLOW:** tensor flow is an end to end open source platform for machine learning. It contains a comprehensive , flexible and efficient ecosystem which leads the ML to perform powered applications in various artificial intelligence fields.

Tensorflow builds and trains ML models easily using intuitive high level APIs like keras with eager execution which makes for immediate model iteration and easy debugging.

A simple and flexible architecture to take new ideas from concepts to code , to state of the art , and the publication faster.

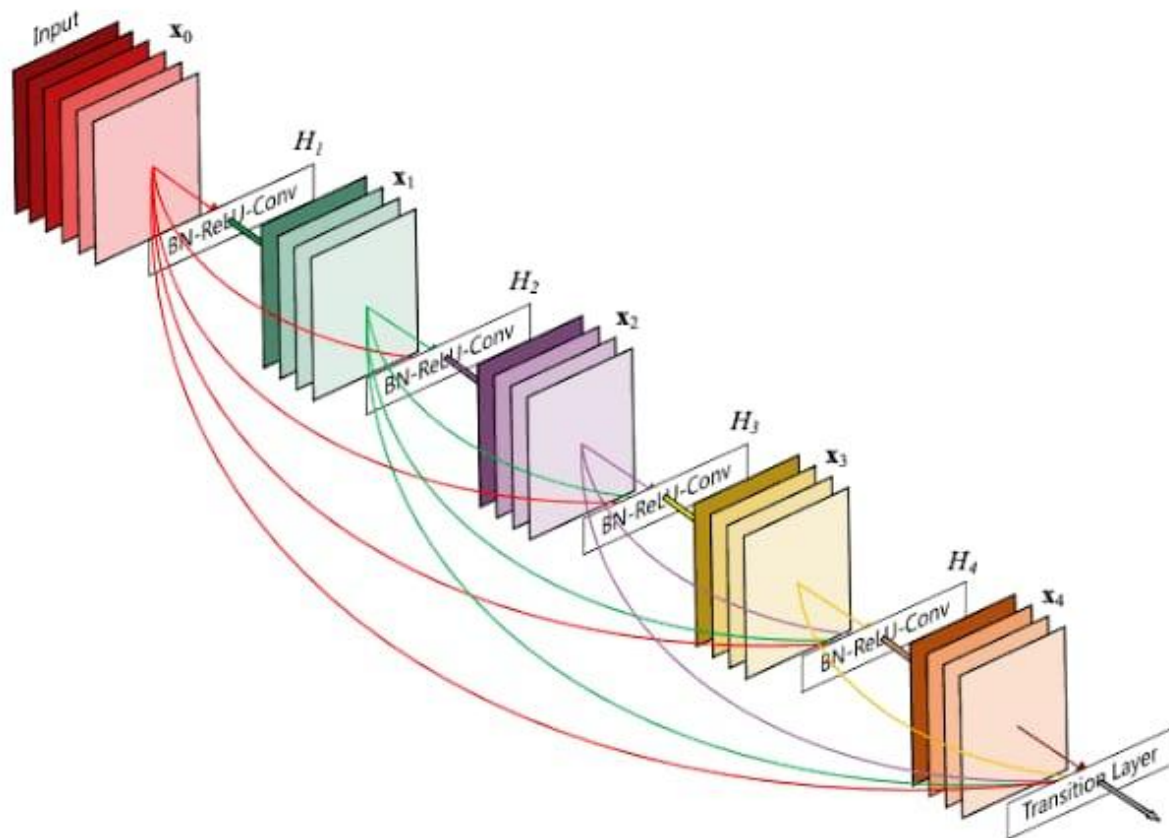
**MOBILENETV2:** It is a convolutional neural network system which works well on mobile devices . It is based on residual connections that are between the bottleneck layers. Hence it is a sub module of the KERAS which optionally loaded the weights pre trained on image net which returns the classifications model of image.It is very similar to the original MobileNet as mentioned earlier that it works well on mobile devices so it uses inverted residual blocks with the bottleneck features. Its main features perform actions on large images more than 32 x 32 and also perform better performances well.



While studying the mobilenetv2 in detail we get to know that this architecture contains 53 convolutional layers and 1 average pool with nearly 350 GFLOP but it's two main components are **Inverted Residual blocks and Bottleneck Residual blocks** . While on the other hand this architecture contains two types of convolution layers

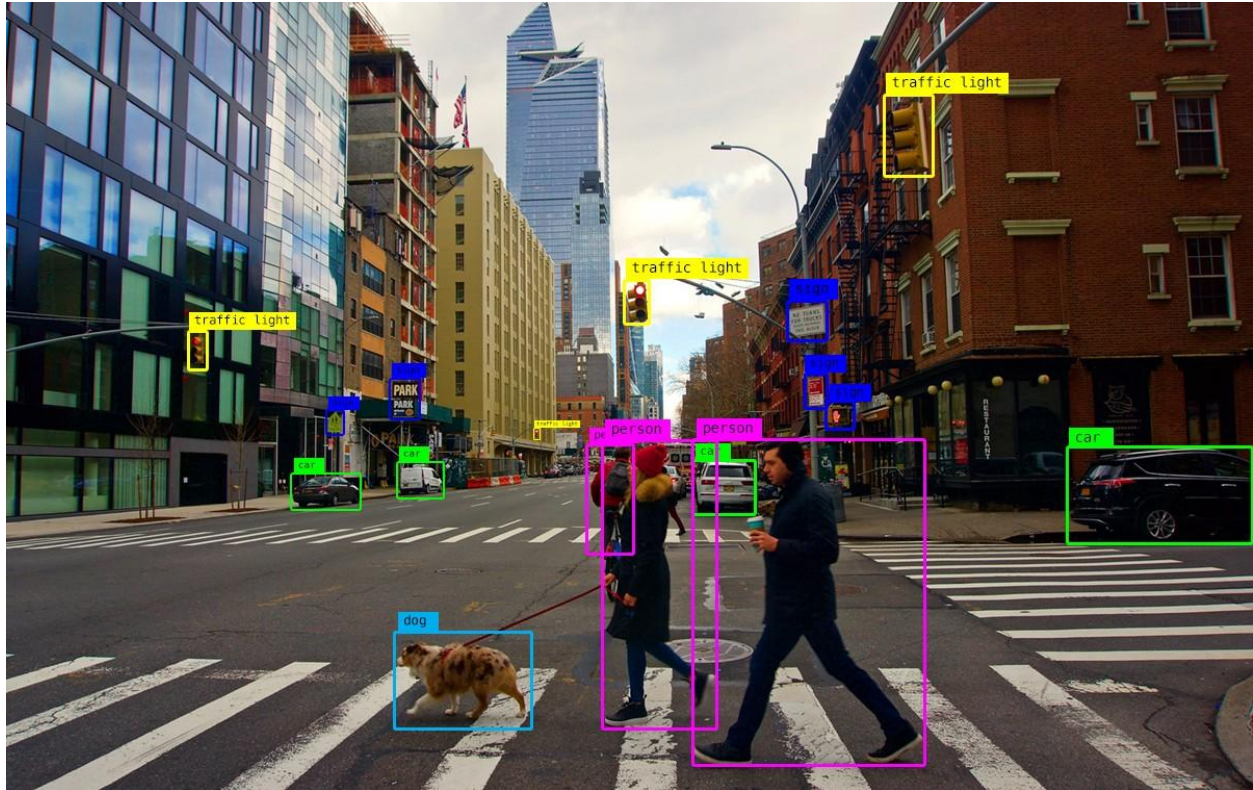
**1x1 convolution and 3x3 depthwise Convolution.** The mobilenetv2 is a depth wise separable convolution in which there is dramatical reduction in the cost and model size of the network which is suitable on any devices with low computational power. Mobilenetv2 works in a way that it contains convolutional layer so by applying the relu activation so first we have bottlenecks and we are converting them from small dimensions to larger dimension then we do channel wise and depth wise 3x3 convolution and after that we do 1x1 convolution to reduce its dimension and a last resultant has same dimension as of input, so we started from  $h * w * k$  and resultant is with computational cost of  $k^2 + 9k$ .

**DENSE NET 201:** It is a densely convolutional neural network; it's also loaded with a pre-trained imagenet which returns the classification model of image. DenseNet's main features are that it works between two convolution layers. It takes your input and does the convolution in order to batch normalization which results in feature maps. From here further batch normalization occurs but now the new layer takes the  $x_1$  and  $x_0$  both as input and generates the  $x_3$  from here the  $x_4$  and then what we are doing is just concatenation with  $x_0$  and push it through convolution and it continues till  $x_4$  and the dimensions are increasing. which results in 1x1 convolution. In this convolution we have densely blocks which change your input into a low output channel while remaining its properties in equilibrium.



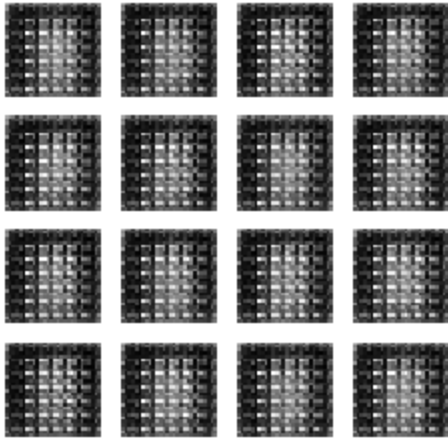
**OPEN CV2 :** open cv2 is an open source library and a very useful library for computer vision based applications in which the video analysis , image processing and many more computer vision applications can be performed . Cv2 is a basic library in python which performs various tasks in which image reading processing and splitting the image into many parts are performed in a very good way so that's why in the present world many actions are performed on the basis of this library . This library is used by many companies such as google and microsoft and amazon and many more digital marketing companies are using this in very useful projects like safety cameras and security purposes while in weapon fields the computer vision is very common and performs very useful tasks. Major tasks performed by this library are image reading , rotating , getting information from image , image writing , getting components from image and many more .





The computer vision field is very advancing and a future of this modern world , In computer vision there are many more task every generation performing because the scientist wants to change the world into more intelligence by inventing many artificial ways somehow we can say that the people are trying to finish out the labour from the market and invent many artificial ways for his or her convenience and peace in every kind of field related to lifes because there are some assumptions that in future the natural resources will finished and we need manya artificial ways to save those natural resources. So making these artificial technologies requires computer vision a lot.





The world around us is advancing day by day . The reason behind is that the world want to be more virtual so that's why the idea of math equation reader is a very beneficial for the future purpose because the more the equation will be solved in the virtual world the more will be the robotics fields become efficient the idea of manufacturing robots which mostly works like the human being need this type of applications as well because this will help in many mathematical fields and now a days almost every field of sciences need a mathematician which lead this more efficient idea to be implemented in future. Japan is making a robot named EVA. It is a kind of artificial robot which mostly works similar to human beings in performing the tasks while on the other hand the application can make this robot work in the teaching field as well and in a very efficient way in the future world.

## METHODOLOGY:

The methodology used in training the dataset is following.

- First of all we create a directory of our data set path where our dataset was placed.
- From There we read out the file's name in our data set as present in figure.

```
directory = r'C:\Users\Muhammad Haris\dataset'
```

```
Name=[]  
for file in os.listdir(directory):  
    if file != '.directory':  
        Name += [file]  
print(Name)  
print(len(Name))
```

```
['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'add', 'dec', 'div', 'eq', 'mul', 'sub', 'x', 'y', 'z']  
19
```

- After that we map the folders with numbers and vice versa folders with numbers from 0 to 18.

```
N=[]  
for i in range(len(Name)):  
    N += [i]  
  
normal_mapping=dict(zip(Name,N))  
reverse_mapping=dict(zip(N,Name))  
print(normal_mapping)  
print(reverse_mapping)  
def mapper(value):  
    return reverse_mapping[value]
```

```
{'0': 0, '1': 1, '2': 2, '3': 3, '4': 4, '5': 5, '6': 6, '7': 7, '8': 8, '9': 9, 'add': 10, 'dec': 11, 'div': 12, 'eq': 13, 'mul': 14, 'sub': 15, 'x': 16, 'y': 17, 'z': 18}  
{0: '0', 1: '1', 2: '2', 3: '3', 4: '4', 5: '5', 6: '6', 7: '7', 8: '8', 9: '9', 10: 'add', 11: 'dec', 12: 'div', 13: 'eq', 14: 'mul', 15: 'sub', 16: 'x', 17: 'y', 18: 'z'}
```

- Now our major work of dataset and test set was done we created two arrays of data set and a test set . by running a loop we check whether we need jpg or png but here in our testing we get both the jpg and png images.
- Each folder contains around 550 images both jpg and png so we run out a loop if less than 400 the dataset other remaining images will store in the testset and the count increases.
- Before this loop we divided the image with 255 to approximate our images to 0 1 form.
- Now we are placing each image of dataset into data and testset to test

```

dataset=[]
testset=[]
count=0
for name in Name:
    path=os.path.join(directory,name)
    t=0
    for im in os.listdir(path):
        if im[-4:]!='.jpg' or im[-4:]!='.png':
            # print(im)
            image=load_img(os.path.join(path,im), grayscale=False, color_mode='rgb', target_size=(40,40))
            image=img_to_array(image)
            image=image/255.0
            if t<400:
                dataset.append([image,count])
            else:
                testset.append([image,count])
            t+=1
    count=count+1

```

- To\_categorical function is used to convert a class vector (integers) to a binary class matrix.
- So the labels0 which are of data are now converted into matrix form and stored into labels1 and the same for the test.
- Both the data and the test are converted into a single dimension array.
- And here our data is split into testing and training.
- ImageDataGenerator Generates batches of tensor image data with real-time data augmentation. (here the image data is generated and the properties of image is set)
- Now here we will get a densely pretrained model from keras which is a proper and dense neural network .
- So then we will check the accuracy and loss from our trained model.
- Fit the model where our training data x and y are given and a validation data is given.
- A second last step of our value prediction where we predicted our model and we calculated the precision , recall , f1-score from our each class.
- After that we want some sample images with equations written on it.
- Now we implement an open cv library from where we first read the image and after that convert BGR to RGB.
- After that gaussian blur is used to decrease the noise distortion. So that the edges could be recognized more clearly.
- After that canni to detect the edges .
- After that we find contours by joining all of the points where the edges are detected.
- After that, sort the contours from left to right .

- Then these are sorted one by one which gives us the x y points of starting of that contours along with the width and height of the bounding rectangle that covers the current contoured element.
- These bounds are then used to extract the region of interest from the original image in this region of image is sent to the prediction function.
- The prediction function resizes the received image to 128 by 128 . and again apply the gaussian blur to reduce the noise . The image is then normalized and reshaped to the requirement of our model.
- It is then sent to the predict function of the model to get predictions.and hence sent to the mapper function for the predicted value comparison with labels.
- CONFUSION MATRIX:

## DATASET:

The dataset we will be using in this project is “Handwritten Math Symbols” from Kaggle.com.

This dataset contains over 9000 handwritten digits and arithmetic operators.

Total no of classes:

10 Digits: 0 1 2 3 4 5 6 7 8 9

Operators: Plus Minus Multiplication Division Decimal Equals

Most images are of resolution 400x400 pixels.

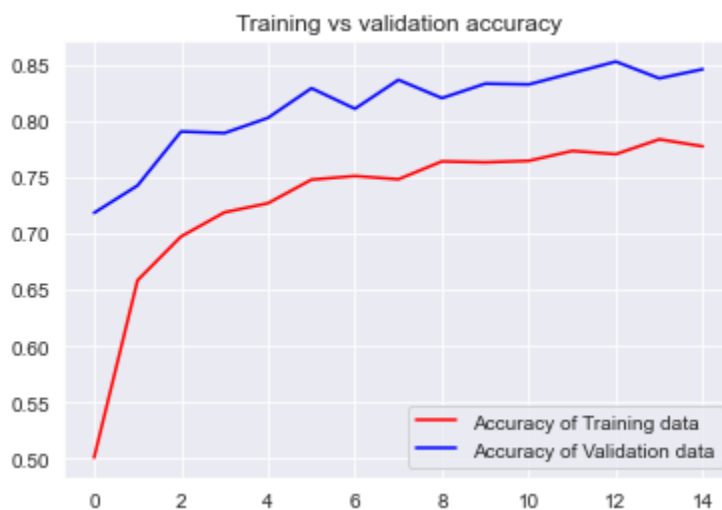
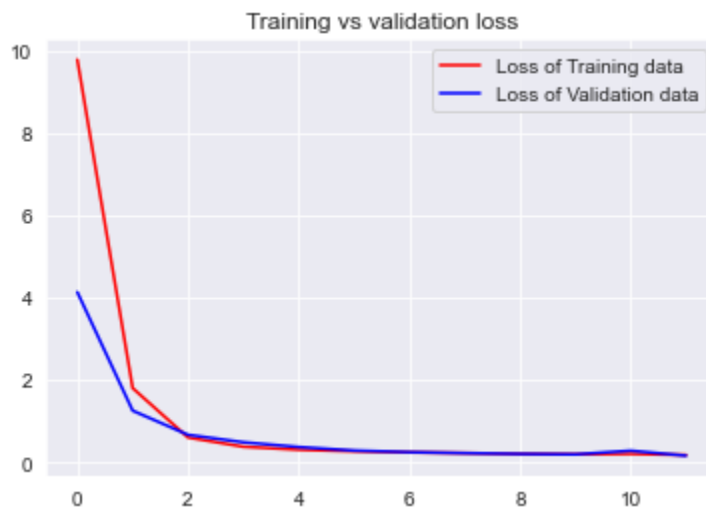
## EXPERIMENTS:

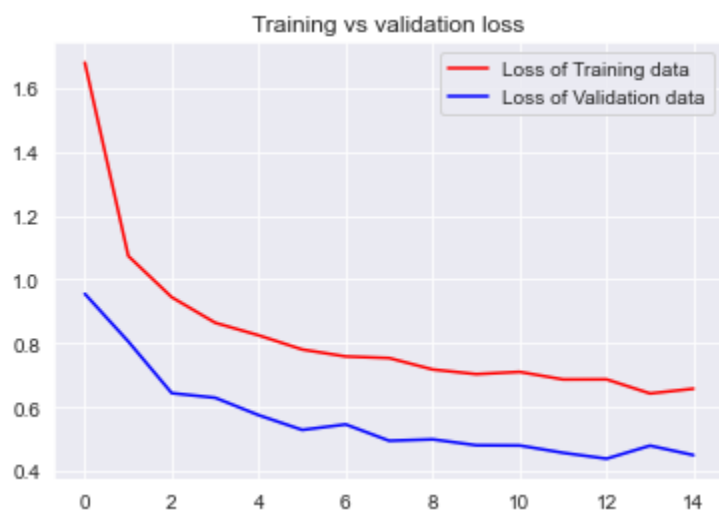
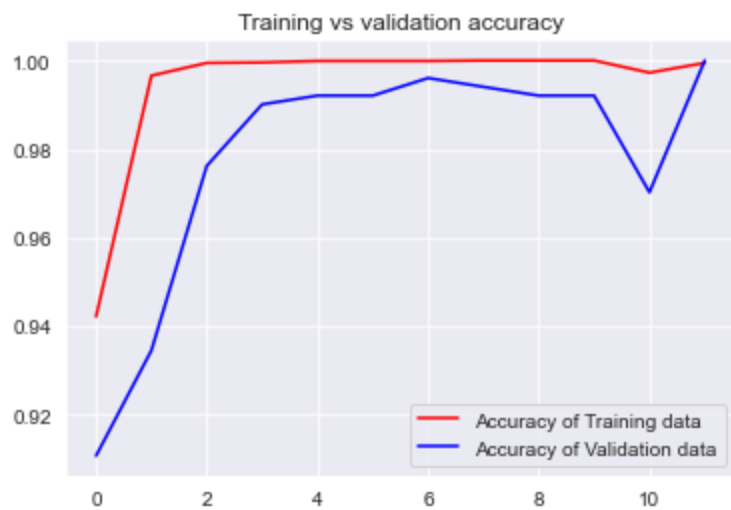
The experiments made during this research was based on manual as well as internet research we take different ideas for many models and the result was some models were using tesseract OCR for there model it is basically a Optical character recognition software which contains trained languages which easily converts an image into text but it was like a built in project so we decided to use our train model and it was a true research work for us.

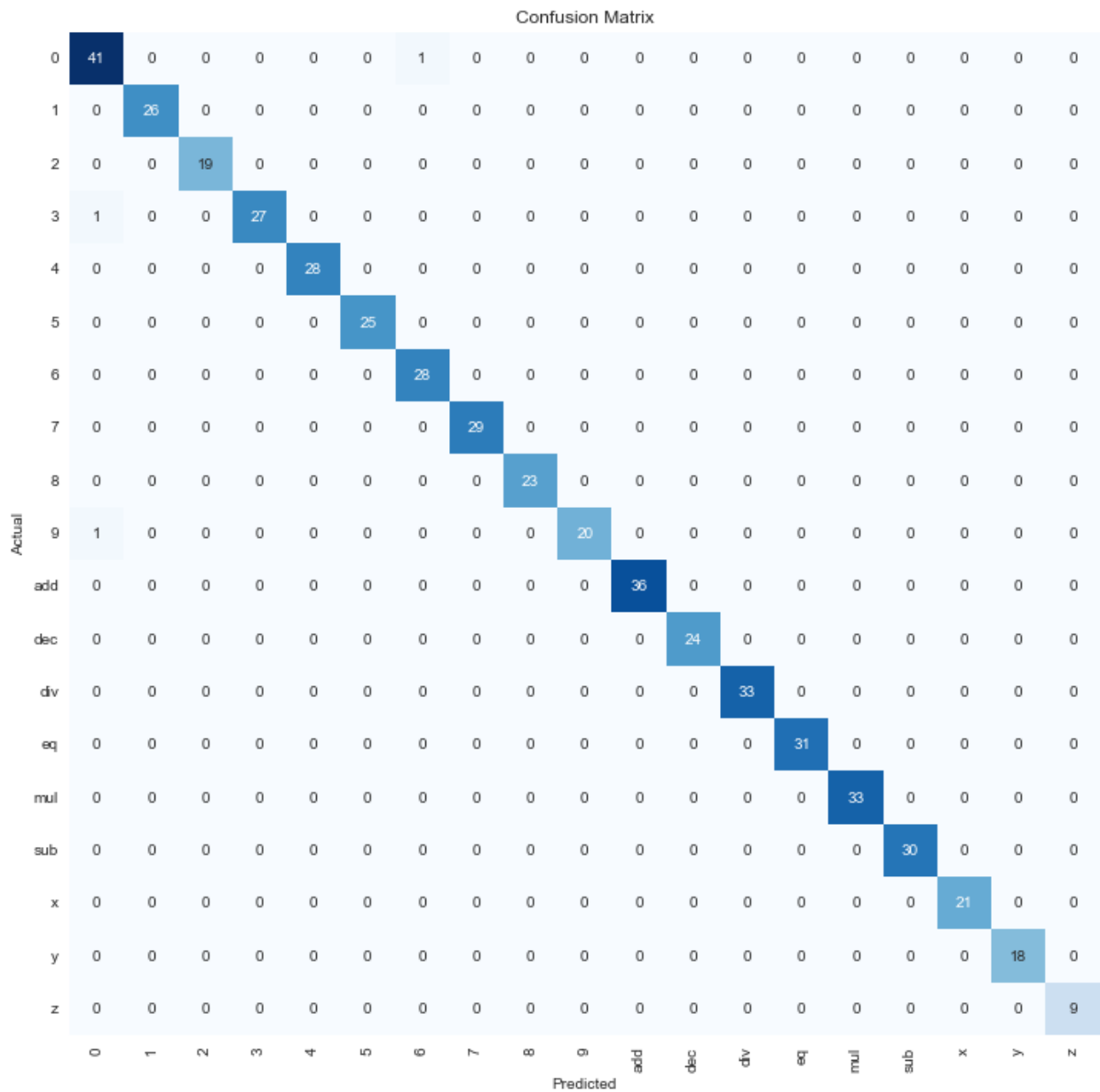
The first experiment was how to collect the data of a very good nature which can make our research more valuable so we studied the major mathematical equations which are built up from 9 digits and 3 variables so we collect that data in a huge amount . About 10 thousand models were trained.

The second is how to train models so we use keras and tensor flow as our model training idea and our loss curves and accuracy curves are shown below.

After that our main work was to use that data and convert it into a representative model so we use computer vision application library named as OpenCv which is and open source library on other side there was OCR present but the open cv was a research based works where the image reading and writing from our train data was much comprehensive and experimentally good for our model.









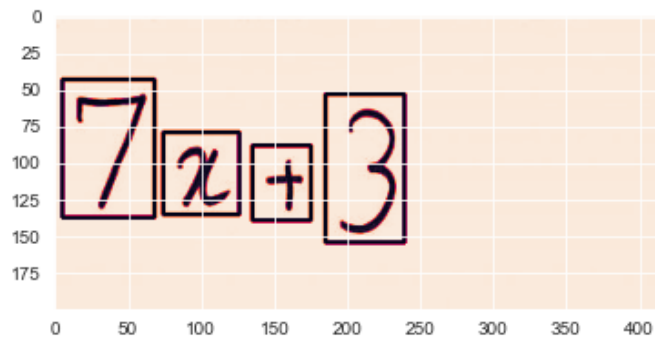
0	78	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	
1	0	64	0	0	0	0	0	3	0	0	0	0	0	1	0	1	0	0	
2	0	0	53	0	0	4	3	11	0	0	0	0	0	0	1	0	1	0	
3	0	0	6	43	0	19	0	2	5	0	0	0	0	0	0	1	5	0	
4	0	0	0	1	49	0	2	1	1	1	0	0	0	1	0	1	12	0	
5	0	0	9	5	0	61	0	3	1	0	0	0	0	0	0	1	1	0	
6	9	0	1	0	3	1	54	1	2	11	0	0	0	0	0	0	2	0	
7	0	5	1	0	0	3	0	71	0	0	2	0	0	0	0	1	3	0	
8	0	0	0	0	0	1	1	0	66	1	0	0	0	0	0	0	3	0	
9	3	1	9	0	1	1	8	1	6	54	0	0	0	0	0	0	5	0	
add	0	0	0	0	1	0	0	1	0	0	77	0	0	0	1	0	3	0	
dec	0	0	0	0	0	0	0	0	0	0	0	82	0	0	1	1	0	0	
div	0	0	0	0	0	0	0	0	0	0	0	0	76	2	0	3	1	0	
eq	0	0	0	0	0	0	0	0	0	0	0	0	0	68	0	8	0	0	
mul	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81	0	1	2	
sub	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	91	0	0	
x	0	0	0	1	0	0	0	0	0	0	1	0	0	0	3	0	76	1	
y	0	0	0	0	1	0	0	1	2	0	0	0	0	0	0	0	1	68	
z	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	42	
	0	1	2	3	4	5	6	7	8	9	add	dec	div	eq	mul	sub	x	y	z
	Predicted																		

## RESULTS:

The results were very interesting. From this research an equation was successfully given in a textual form from a handwritten equation , we had tried many different equation images and their outputs were very accurate . The methodology applied was successful and the output with minimum loss was given by the model .

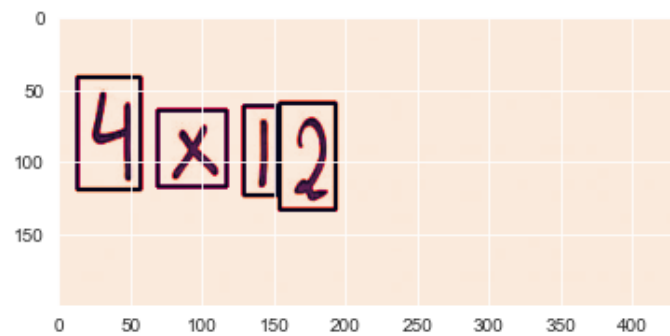
```
element_extractor_from_image(r"F:\itu\sem5\AI\peeraject\test5.jpg",10,10,15,15,1200)
```

= 7 x + 3



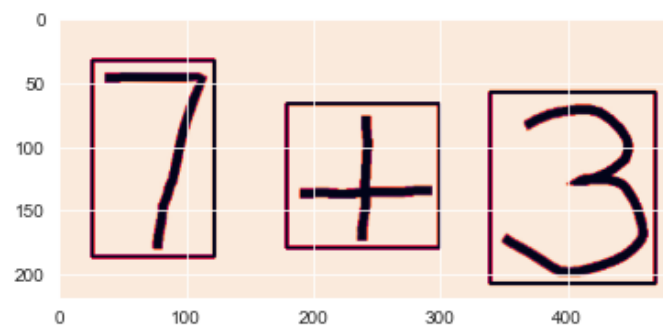
```
element_extractor_from_image(r"F:\itu\sem5\AI\peeraject\test7.jpg",10,10,15,15,1200)
```

= 4 \* 1 2



```
element_extractor_from_image(r"F:\itu\sem5\AI\peeraject\test1.jpg",10,10,15,15,1200)
```

= 7 + 3



CONCLUSION:

The research was based on how to get an equation from a hand written image . So there were many other models which were presenting textual data like some sentences etc and then converting them into textual written data but our project aims toward the mathematical field in which the data in the form of an equation is readed and we get the equation . the comparison with the other models were that they were getting a whole equation as a text in on text form while on other hand our model gets each attribute from the handwritten image and storing it on separate indexes of the arrays which says that if in future we want to perform the mathematical functions on that attributes we can easily perform that and this is our model quality which differs it from other researches as well . The model can be used in a wide field of artificial intelligence applications as impoverished by the government and we can take advantage of such a model and can make progress in this field for our country as well.

## REFERENCES:

1. [2018 CVPR] [MobileNetV2] [MobileNetV2: Inverted Residuals and Linear Bottlenecks](#)
2. Haponik, A., 2022. *Text Extraction From Images Using Machine Learning (update: June 2021)*. [online] Addepto. Available at: <<https://addepto.com/text-extraction-from-images-using-machine-learning/>> [Accessed 31 January 2022].
3. Paperswithcode.com. 2022. *Papers with Code - KOHTD Benchmark (Handwriting Recognition)*. [online] Available at: <<https://paperswithcode.com/sota/handwriting-recognition-on-kohtd>> [Accessed 31 January 2022].
4. "What Is the Opencv Library and Why Do We Need to Know About It?" *Topcoder Top Technology Talent On-Demand*, <https://www.topcoder.com/thrive/articles/what-is-the-opencv-library-and-why-do-we-need-to-know-about>
5. <https://keras.io>
6. [http Goldfarb, D.: A family of variable-metric methods derived by variational means. Mathematics of computation 24\(109\), 23–26 \(1970\)ps://www.tensorflow.org/guide/keras/sequential\\_model](#)
7. [He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 \(2016\)](#)

8. [https://www.tensorflow.org/guide/keras/sequential\\_model](https://www.tensorflow.org/guide/keras/sequential_model)
9. **MobileNetV2: Inverted Residuals and Linear Bottlenecks** (CVPR 2018)