Hand Written Mathematical Equation Solver

1[st] Alavala Maneela,2[nd]Kolepaka Pragnya
Department Of Information Technology.
Vardhaman College Of Engineering
Hyderabad,India
maneelaalavala001@gmail.com,kolepakapragnya118@gmail.com

3[rd] Nazeer Ahamed Abdul,4[th]Muni Sekhar Velpuru
Department Of Information Technology.
Vardhaman College Of Engineering
Hyderabad,India
nazzerabdul28@gmail.com,munisek@gmail.com

***Abstract*—Mathematics plays an important role in everybody's life. We use the calculators available in our electronic gadgets to solve mathematical equa-tions. Calculators available in these gadgets are so advanced that it can solve most complex equations. Developing a model that can recognize and solve mathematical equations from handwriting is potential area of research. This handwritten equation solver is model that can solve mathematical equations written by various persons. Hand written equations are given as input. The model recognizes the numbers and symbols, forms a string of equation, solve it and give it as output. Model training uses a handwritten mathematical symbol dataset that contains numerous images of numbers and symbols. The output is generated using a convolutional neural network. Use a convolutional neural network to classify specific functions. Each of the correct recognition string operations is used to solve the equation.**

I. Introduction

Due to the swift improvement in technology and internet technology, most of the documents, books and literature's within the area of computing also as others are increasingly becoming digitalized. Mathematics is broadly utilized in most areas of science, like physics, engineering, medicine, economics, etc. Digital document analysis and understanding is one of the major research concern today. For the recognition of English characters and numbers in electronic books OCR (optical character recognition) is mostly used to attain higher recognition accuracy. With the advancement in technology, machine learning and deep learning are playing an important role in present times. Currently, machine learning and deep learning techniques are used Handwriting recognition, robotics, AI and many other areas. Convolutional neural networks (CNNs) are the most commonly used classification model. In the field of computer vision. There is one in the Deep Convolutional Neural Network (CNN) Excellent performance in image classification and machine learning And pattern recognition. CNN extraction function from a series of images operation. This project focuses primarily on arithmetic recognition After successfully recognizing the equation and the equation, apply the string An operation to solve these equations. Feature extraction from each Custom characters are the most difficult part of handwriting To its different shape and structure. Apply CNN to solve this problem Does not require predefined features to classify specific features.After character classification, recognition of character is done and then it is solved.

II. OBJECTIVE

The main objective of our project is to recognise, solve and display the generated output of the given arithmetic equation as input.Hand written mathematical equation solver uses handwritten math symbols dataset, from Kaggle, to train the model.

III. LITERATURE SURVEY

So far, some work has been done on handwriting recognition. .. Formula recognition MER system based on SVM Proposed a projection histogram of a simple formula [1] is part of the offline recognition of handwritten expressions. This paper Basically, it focuses on a number of techniques used for feature extraction and recognition. An effective and robust system for recognizing printed matter and handwriting Mathematical symbol to evaluate an expression Three consecutive paths: layout path, lexical path, and Operator tree. This system takes mathematical characters as input. The input goes through three paths, after which the expression is evaluated [2]. Another system recognition system for offline handwritten math symbols bol has been proposed [3]. In this feature extraction system The sign is taken into account. It is based on a relative study of feature extraction Method. Recognition was performed by the SVM support vector Ma chine, supervised machine learning algorithms available Both classification and regression challenges. So another system is proposed Diagonal feature extraction method for handwritten characters using A feedforward neural network algorithm is used. This technique uses diagonal lines. Horizontal and vertical features for classification. Some papers Available online to recognize formulas with convolution Neural network [4]. In recent years, a branch of artificial neural networks A work called deep learning shows great potential in resolving classifications. Problem [5].

IV. PROPOSED METHODOLOGY

In our proposed method at first data set is processed.Each and every image is segmented using contour extraction. Then we consider each part of the segmented image as an full image for the further process.Image is given as input to CNN model. Image is then pre-processed that is image is converted to gray scale and is resized.Next, for each symbol in the equation diagram, Search for a specific character in the format of the connected component. Each segmented [7] character is then provided as input to the convolutional neural network.

Character classification model. The resulting character, this is The output from the CNN is used to create a similar string To the original equation. Then the obtained equation string is evaluated and is projected as output. By using Convolutional Neural Network, accuracy is highly increased.Handwritten arithmetic equation solver solves the given equation with high accuracy i.e; 99 percent and less computational time.

A. ALGORITHM AND FLOWCHART

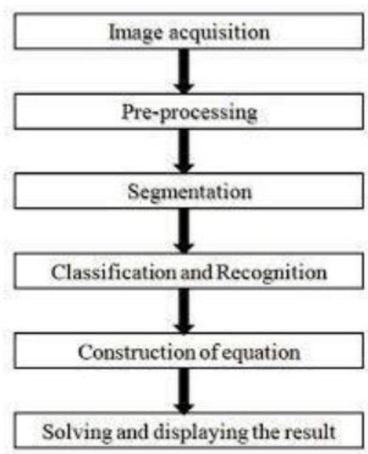

Fig. 1. FLOW CHART

Algorithm:

Step 1: Download the dataset, handwritten math symbols.
Step 2: Process each and every digit and also arithmetic symbols.
Step 3: Extract contours from every image using contour extraction.In this way features are extracted from data.
Step 4: Build convolutional neural network model and train the data using convolutional neural network model.
Step 5: Test the model with images of equations.
Step 6: After taking image as input, segmentation of characters in equation isdone .Segmentation is done using contour extraction.
Step 7: Each character in equation is recognised by the model and the equation is generated by appending the recognised haracters as a string.
Step 8: The string equation is evaluated, considering the precedence.Generated output of equation is displayed to user.

B. DESIGN

At first the hand written image is given as input.Input image is given to the model by resizing the image after converting it into greyscale.

Initially, the image is given as input to convolution layer which consists of 30 filters with kernel size of size 5X5 by using Relu activation function,Relu(Rectified Linear Unit) is a linear function that will output the input itself, if input is positive,otherwise, it will output zero.

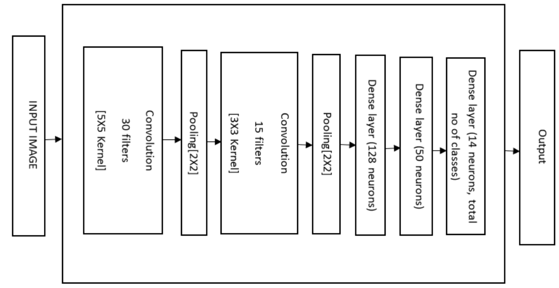

Fig. 2. DESIGN FLOW

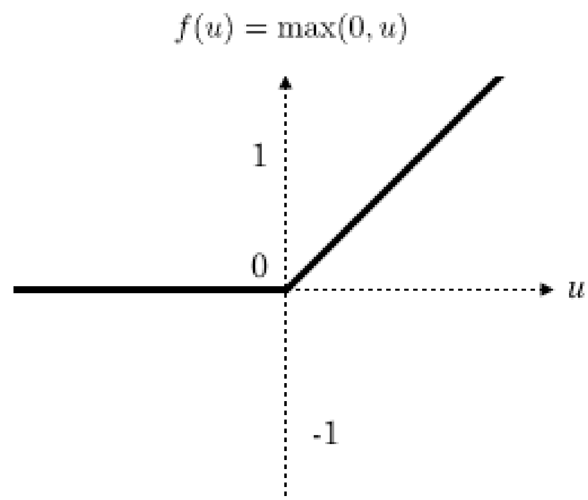

Fig. 3. ReLu activation function

Output from convolutional layer is given as input to MaxPooling Layer, with kernel of size 2X2. It is a pooling operation that selects the maximum element from the region of the feature map covered by the filter.

Then the output from maxpooling layer is given as input to Convolution layer which consists of 15 filters with of size 3X3 kernel.

Then output from previous layer will be given to MaxPooling Layer of size 2X2.

Finally, output from pooling layer is flattened and it passes 3 dense layers . in which the first one consists 128 neurons , second consists of 50 neurons and third consists 14 neurons(total number of classes).

After passing through all these layers output is generated.

V. IMPLEMENTATION

Implementation of hand written equation solver model is done using Convolutional neural network.Initially, Hand written math symbols data set is taken from kaggle which contains over 100000 images.In the data set some symbols have more number of images than others. To remove this bias consider same number of images from all the digits and symbols. In this model we only consider 14 different classes that is 0 to 9 digits,+,-,multiplication symbol and x variable.

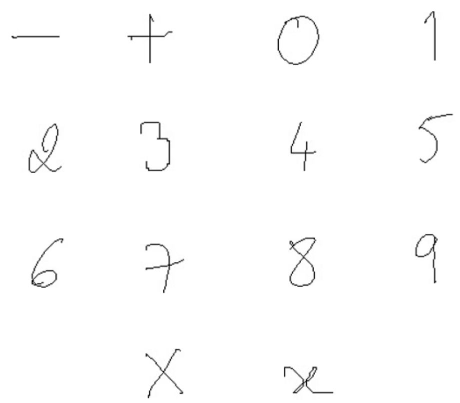

Fig. 4. Sample images from dataset

A. *Construction of Convolution Neural Network*

Main layers used in building the Convolution Neural Network are

1. Convolution Layer
2. Max Pooling Layer
3. Fully Connected or Dense Layer

1. Convolution Layer: The convolutional layer is the main building Block used in convolutional neural networks. It is a simple application of the filter that results in input activation. The innovation of the cumulative neural network is the ability to automatically learn many parallel filters specific to a training data set in limitations of a particular predictive modeling problem, such as as an image classifier. The result is very specific features can be recognized anywhere in the input image.

2.Max Pooling Layer: Pooling is done to shrink the image matrix into a smaller size. Pooling is done after passing through a activation layer, here we used max pooling layer after convolution layer with ReLu activation function. In max pooling maximum value is considered in every patch of feature map. Hence input representation is down sampled.

3.Fully connected layer: Fully Connected Layer is simply, feed forward neural networks. Fully Connected Layers form the last few layers in the network. The input to the fully connected layer is the output from the final Pooling or Convolutional Layer, which is flattened and then fed into the fully connected layer. In this model, we used 2 convolutional and max pooling layers one after the other. Each convolutional layer has 30 filters in which one is of size 5x5 and other is 3x3, and max pooling were performed on every 2x2 pixels. The output of this was flattened and fed into a multi-layer perceptron network for classification.

Batch Normalization was done before every convolutional layer to improve the training speed of the model. The key idea is to normalize the inputs to layers within the network along with the input to the whole network.

After flattening the output of the last max-pooling layer in each of the models we use a multi-layer perceptron network to classify the images. We use two dense layers having 128 neurons and 50 neurons respectively. The final output layer has 14 neurons for the 14 different classes that is all the digits,symbols and x variable. We used 'ReLU' activation function in the hidden layer and a SoftMax activation function in the final layer. The SoftMax function, or normalized exponential function, transforms Kdimensional vector of arbitrary real values to a Kdimensional vector of real values in the range [0, 1] that add up to 1.

We trained the model using the data from data set where we considered same number of images for each digit or symbol to remove bias. Initially 500 images from each symbol is considered. Time taken to process all the training images is nearly 10 minutes but the accuracy achieved is less. We first run the model for 30 epochs. After 30 epochs the loss is not considerably decreasing. The difference in accuracy and loss after 30 epochs is differing in only decimals. Hence we stopped at 30 epochs.

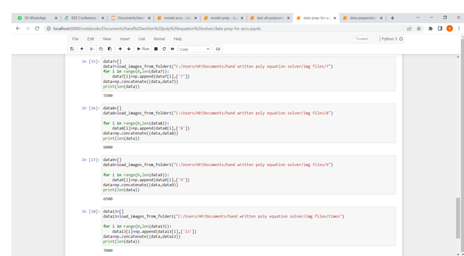

Fig. 5. Data preparation of 500 images from each class

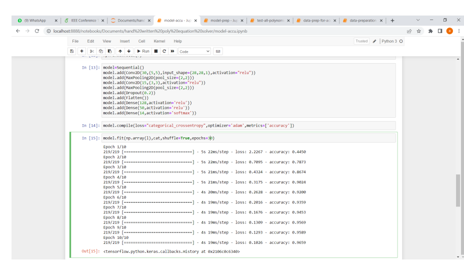

Fig. 6. Accuracy achieved by considering 500 images from each class

By considering more number of images training of model is done again to verify the results. We increased number of images from each symbol.Now we considered 1000 images from each symbol. Time taken to process all the training images is more than that of 500 images from each class but the accuracy achieved is also relatively more.

Same process is repeated by increasing the number of images. Considering 1500, 2000, 2500 and 5000 images from

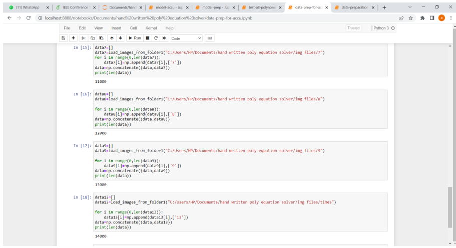

Fig. 7. Data preparation of 1000 images from each class

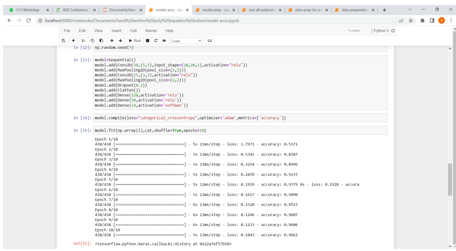

Fig. 8. Accuracy achieved by considering 500 images from each class

each class, model is trained. The results that is accuracy is increasing as the number of images is increasing and also loss is decreased. But the time taken in processing data increased significantly as the number of images increased. Finally, model is trained by considering 5000 images from each symbol. Accuracy achieved by considering 5000 images from each class is nearly 98 percent. We trained the model for 30 epochs. Some symbols have even more images but some have less images. So as to remove the bias we only considered 5000 images from each symbol.

VI. RESULTS

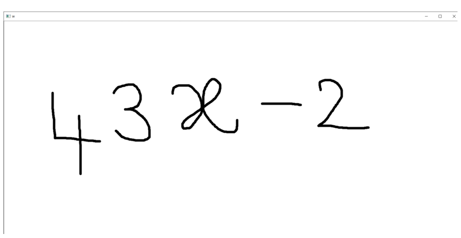

Fig. 9. INPUT IMAGE

Hand written equation is given as input to model.After taking the image of equation as input, contours are generated. Contours are simply a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition. For each digit and symbol contour is generated.

Bounding rectangles are the dimensions of the contours generated for each character or symbol in the equation. Figure 10 shows the bounding rectangles that are obtained for the image given as input. Here we have considered the input that

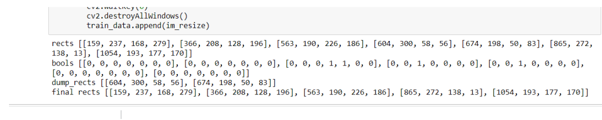

Fig. 10. BOUNDING RECTANGLES

have seven characters in the equation. Hence there are seven final rectangles that are obtained.

Fig. 11. EQUATION AS A STRING

After recognition, all the symbols or digits in the equation are appended in the form of a string. The output will initially be generated in the form of a string as shown in figure 11.

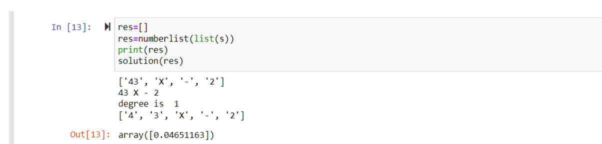

Fig. 12. FINAL OUTPUT

The string that is generated is evaluated, roots are generated and is given as final output as shown in the figure 12.

We ran the proposed model for 30 epochs and validation and training accuracy is presented in figure 13. Arithmetic equations and polynomial are recognized accurately and solution for arithmetic equations and roots for polynomial equations are generated.Above are results for linear equation. Results obtained for considering a quadratic equation are

Above image 14 is the hand written quadratic equation given as input to model.

Image 15 are the bounding rectangles obtained as output to the quadratic equation given as input.

After segmentation each and every symbol is classified using the constructed CNN model. Every symbol present in

```
2108/2108 [==============================] - 27s 13ms/step - loss: 0.0536 - accuracy: 0.9840
Epoch 2/10
2108/2108 [==============================] - 27s 13ms/step - loss: 0.0498 - accuracy: 0.9853
Epoch 3/10
2108/2108 [==============================] - 27s 13ms/step - loss: 0.0476 - accuracy: 0.9855
Epoch 4/10
2108/2108 [==============================] - 28s 13ms/step - loss: 0.0470 - accuracy: 0.9862
Epoch 5/10
2108/2108 [==============================] - 30s 14ms/step - loss: 0.0430 - accuracy: 0.9872
Epoch 6/10
2108/2108 [==============================] - 33s 16ms/step - loss: 0.0437 - accuracy: 0.9879
Epoch 7/10
2108/2108 [==============================] - 33s 16ms/step - loss: 0.0424 - accuracy: 0.9877
Epoch 8/10
2108/2108 [==============================] - 31s 15ms/step - loss: 0.0391 - accuracy: 0.9887
Epoch 9/10
2108/2108 [==============================] - 29s 14ms/step - loss: 0.0424 - accuracy: 0.9878
Epoch 10/10
2108/2108 [==============================] - 30s 14ms/step - loss: 0.0389 - accuracy: 0.9886

Out[16]: <tensorflow.python.keras.callbacks.History at 0x20502146130>

In [17]: model.fit(np.array(l),cat,shuffle=True,epochs=5)

Epoch 1/5
2108/2108 [==============================] - 31s 15ms/step - loss: 0.0404 - accuracy: 0.9886
Epoch 2/5
2108/2108 [==============================] - 32s 15ms/step - loss: 0.0373 - accuracy: 0.9892
Epoch 3/5
2108/2108 [==============================] - 30s 14ms/step - loss: 0.0371 - accuracy: 0.9896
Epoch 4/5
2108/2108 [==============================] - 31s 14ms/step - loss: 0.0359 - accuracy: 0.9901
Epoch 5/5
2108/2108 [==============================] - 31s 15ms/step - loss: 0.0340 - accuracy: 0.9905

Out[17]: <tensorflow.python.keras.callbacks.History at 0x2050216f7c0>
```

Fig. 13. ACCURACY

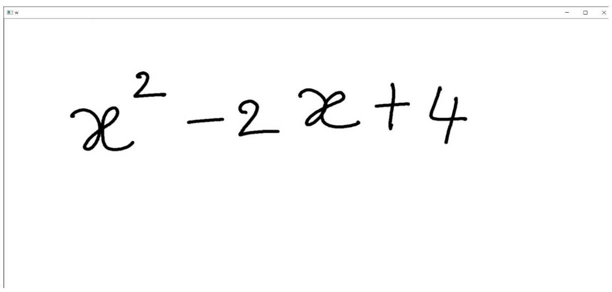

Fig. 14. Input Image

```
rects [[166, 232, 163, 119], [206, 305, 30, 37], [252, 240, 31, 43], [326, 139, 83, 71], [462, 262, 94, 17], [591, 214, 105, 9
8], [742, 184, 158, 107], [767, 245, 34, 32], [815, 199, 39, 29], [936, 170, 89, 126], [1073, 179, 97, 153]]
bools [[0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0], [0,
0, 0, 0, 0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0]]
dump_rects [[206, 305, 30, 37], [252, 240, 31, 43], [767, 245, 34, 32], [815, 199, 39, 29]]
final rects [[166, 232, 163, 119], [326, 139, 83, 71], [462, 262, 94, 17], [591, 214, 105, 98], [742, 184, 158, 107], [936, 17
0, 89, 126], [1073, 179, 97, 153]]
```

Fig. 15. Bounding Rectangles

```
In [179]: s=""
for i in range(len(train_data)):
    train_data[i]=np.array(train_data[i])
    train_data[i]=train_data[i].reshape(1,28,28,1)
    result=np.argmax(loaded_model.predict(train_data[i]), axis=-1)
    if(result[0]==10):
        s=s+'-'
    if(result[0]==11):
        s=s+'+'
    if(result[0]==12):
        s=s+'X'
    if(result[0]==13):
        s=s+'*'
    if(result[0]==0):
        s=s+'0'
    if(result[0]==1):
        s=s+'1'
    if(result[0]==2):
        s=s+'2'
    if(result[0]==3):
        s=s+'3'
    if(result[0]==4):
        s=s+'4'
    if(result[0]==5):
        s=s+'5'
    if(result[0]==6):
        s=s+'6'
    if(result[0]==7):
        s=s+'7'
    if(result[0]==8):
        s=s+'8'
    if(result[0]==9):
        s=s+'9'
print(s)

X2-2X+4
```

Fig. 16. Equation as a string

input equation image is recognised as one of the 14 classes, which closely matches to input symbol. After classification each symbol is concatenated to from equation string. Image 16 shows the string equation formed by concatenating each symbol from quadratic equation given as input.

```
In [183]: solution(list(s))

X^2 - 2 X + 4
degree is  2
['X', '2', '-', '2', 'X', '+', '4']
[ 1. -2.  4.]
[1.+1.73205081j 1.-1.73205081j]

Out[183]: array([1.+1.73205081j, 1.-1.73205081j])
```

Fig. 17. Final Output

After the concatenated string is formed, degree of the equation is identified that is the highest power in the equation. A list of size degree of equation is formed to store the roots of equation. We use python module to find the roots of the equations. Roots of the identified equation is given as output.Image 17 shows the roots generated by solving the recognised polynomial equation.

VII. CONCLUSION AND FUTURE SCOPE

To simplify the math, the main task is to extract the features Image recognition using CNN model. For CNN The model correctly classifies and generates all segmented images Correct arithmetic equation. In the CNN-based architecture proposed there Two convolutional layers and a maximum pooling layer. This is a successful presentation State-of-the-art. Compared to other deep learning In the approach, the implemented deep learning model has better predictive power. From the perspective of accuracy and loss. Time required to train the model It was much lower than other machine learning approaches. With this In the model, the equation is recognized successfully with good accuracy. Anyone can easily use this as the process is simple and unnecessary Prior knowledge of technology.
Here is a model that can recognize handwritten polynomials Solve the equation and this equation. Results can be made more More precisely, it uses more convolutional layers and more hidden neurons. You can completely eliminate the need for typing. We will strive to improve in the future Increase accuracy and allow the system to be used for multiple mathematical purposes Expression at the same time. All the above points are for Future research.

VIII. REFERENCES

[1]Azzeddine Lazrek Widad Jakjoud. "Segmentation method of offline math- ematical symbols". In: International

Conference on Multimedia Computing and Systems (ICMCS) (2011).

[2]Jyothy R.L et al. Shilpa C Vijayan. "Histogram Based Connected Com- ponent Analysis for Character Segmentation". In: International Journal of Scientific and Research Publications 6 (2016).

[3]Baviskar P.V. Adhiya K.P. Gharde S.S. "Identification of Handwritten Simple Mathematical Equation Based on SVM and Projection Histogram". In: International Journal of Soft Computing and Engineering (IJSCE) (2000), pp. 425–429.

[4]Course note of CNN Stanford. url: http://cs231n.github.io/convolutional- networks/.

[5]Garg V.K. Cipolla R. Shankar S. "Deep-carving: Discovering visual attributes by carving deep neural nets". In: IEEE Conference on Computer Vision and Pattern Recognition (2015), pp. 3403–3412.

[6]Sonu Agrawal Pooja Kamavisdar Sonam Saluja. "A Survey on Image Classification Approaches and Techniques". In: international Journal of Advanced Research in Computer and Communication Engineering 2 (2013).

[7]Eric Lecolinet Richard G.Casey. "A survey of methods and strategies in character segmentation". In: (1996).

[8]Fuad Hasan, Shifat Nayme Shuvo et al, Bangladesh Continuous Handwriting Character and Digit Recognition Using CNN, 7th international Conference on Innovations in Computer Science and Engineering (ICICSE 2019), vol 103, spring, Singapore, pp. 555-564

[9]Jalal Uddin Mahmuad, Mohammed Feroz Raihan and Chowdhury Mofizur Rahman, "A Complete OCR System for Continuous Bangla Characters", IEEE TENCON-2004: Proceedings of the Conference on Convergent Technologies for the Asia Pacific, 2004.

[10]Gharde, S. S., Baviskar, P. V., Adhiya, K. P. (2013) Identification of Handwritten Simple Mathematical Equation Based on SVM and Projection Histogram. International Journal of Soft Computing and Engineering (IJSCE),3(2),425-429

[11]Francisco Álvaro and Joan Andreu Sánchez, "Comparing Several Techniques for Offline Recognition of Printed Mathematical Symbols", IEEE,ICPR 2010.

[12]Christopher Malon, Masakazu Suzuki, and Seiichi Uchida, "Support Vector Machines for Mathematical Symbol Recognition", Technical Report of IEICE.

[13]Fotini Simistira, Vassilis Katsouros and George Carayannis, "A Template Matching Distance for Recognition of On-Line Mathematical Symbols", Institute for Language and Speech Processing of Athena - Research and Innovation Center in ICKT, Athens, Greece.

[14]Dipak D. Bage et. al,"A new approach for recognizing offline handwritten mathematical symbols using character geometry", International Journal of Innovative Research in Science, Engineering and Technology Vol. 2, Issue 7, July 2013, ISSN: 2319-8753

[15]Catherine Lu Karanveer Mohan.' Recognition of Online Handwritten Mathematical Expressions Using Convolutional Neural Networks' cs231n project report stanford 2015.

[16]Sanjay S. Gharde, Vidya A. Nemade and K. P. Adhiya,"Evaluation of Feature Extraction and Classification Techniques on Special Symbols", IJSER, Volume 3, Issue 4, 2012

[17]Ahmad-Montaser Awal ,Harold Mouchère, Christian ViardGaudin.'Towards Handwritten Mathematical Expression Recognition' 2009 10th International Conference on Document Analysis and Recognition.

[18]G. Louloudisa *, B.Gatosb,1, I.Pratikakisb,1, C.Halatsisa (2009). Text line and word segmentation of handwritten documents

[19]Richard G. Casey, Eric Lecolinet " A survey of methods and strategies in character segmentation'1996

[20]Srivastava, Nitish Hinton, Geoffrey Krizhevsky, Alex Sutskever, Ilya Salakhutdinov, Ruslan. (2014).Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research.15.1929-1958.