Handwritten Polynomial Equation Solver Using Convolutional Neural Network

Dr.Prashant Kumbharkar$^{\#1}$, Dr. Deepak Mane^{*2}, Sakshi Takawale$^{\#3}$, Shweta Bankar$^{\#4}$, Divya Shingavi$^{\#5}$, Divya Patil$^{\#6}$

JSPM'S RajarshiShahu College of Engineering, Tathawade, Pune, Maharashtra, India - 411033

e-mail: [1]pbk.rscoe@gmail.com,[2]dtmane@gmail.com,[3]sakshitakawale@gmail.com, [4]shwetabankr54@gmail.com, [5]shingavidivya@gmail.com, [6]divyapatil478@gmail.com

Article Info

Page Number: 4412 – 4425

Publication Issue:

Vol 71 No. 4 (2022)

Abstract

More the number of people more is the diversity in handwritten charactersand hence task of recognizing characters becomes intricate. In this paper, we proposed a Customized Convolutional neural network (CCNN) model which has made recognition more precise than the existing recognitionalgorithms. Customized ConvolutionalNeuralNetwork(CCNN)extractsfeaturesautomaticallyandhandleslarge amount of data thus resulting in better accuracy. Classification and Segmentation of characters is a challenge on based of machine vision system. The study looks at polynomial equations written by hand as well as dataset. Dataset of 10 digits, 4 operators and 6 variables is used for performance evaluation of proposedmodel.This input image, we apply projection horizontally for the equation to be segmented. After this, each segmented character is sent into our neural network model i.e. CNN, for character indexing. A character string that looks similar the earlier inputequation. Every approach is required for solution. Ourmodeldoesn'trestrictanyn-dimensionaldataandproducesbetter results. Customized Convolutional Neural Network (CCNN) extracts features automatically and handles a large amount of data thus resulting into a better accuracy. Finally, the strategy used is effective as described in results of experiment.This paper explains the methodology and architecture of our CNN model which can be further referred to in future and used for solving other pattern and recognition problems.

Article History

***Article Received**: 25 March 2022*

***Revised**: 30 April 2022*

***Accepted**: 15 June 2022*

***Publication**: 19 August 2022*

Keywords: -Handwritten Equation, Pre-processing, Segmentation, CNN, Recognition, Polynomial Equation

1. Introduction

Mathematics and natural science play very important role in human life. Polynomial equations can be used for the graphs, 3D curves, motion of particle, for profits [9]. Currently solving mathematical equations using AI and ML has much scope in research field. Mathematical equations nowadays have much different powers, Sizes and 2 – dimensional values. For creating and effective software algorithm which can take input in image from and recognize, solve the equation given to CNN (Convolutional Neural Network) can be used. CNN is trendy model for recognizing objects, segmentation and pattern recognition with good accuracy. CNN extracts pattern and features from the images in series of steps. Character recognition mostly shortened OCR (optical character recognition) which translates all kinds of text in machine readable text [2].

Objective:

- In this project, we have tried to deal with problems related to recognizing the Handwritten Polynomial Equation and further steps for solving.
- Our primary goal was to create a good user interface and trained CNN model which can work together and get the end results.
- We have achieved the goals of recognition, Simplification and getting the result of equations.

Following is the overview of our paper: In Section 2 we have discussed few past relevant work. Section 3 is all about strategies that can be utilized to develop a good and reliable algorithm / model for our project. About algorithm used to train and test our dataset is explained in Section 4. The experimental result of project is explained in Section 5. Finally, in section 6, the authors defined future scope and concluded.

2. Related Work

Rajwardhan [1] developed a system which can recognize handwritten mathematical equations and recognized equation is solved using fundamental operations such as addition, multiplication, division and subtraction. In this project, image of handwritten mathematical equation is taken as input then system recognizes equation and finally result is displayed. They used Dense layer for feature extraction, softmax activation function for multiclass classification as there 17 classes. Along with this, RNN model is utilized for result prediction. Presently, system is able to recognize and solve simple equations faster than complex equations. Further accuracy of model can be increased, to make it solve complex polynomial equations.

Mohini [2] proposed a model for recognition and simplification of handwritten math equations. In this work, image of handwritten math equation is taken as input and recognized equation is displayed to the user. CNN is used for recognition of input handwritten equation. Currently, system is able to solve simple mathematical equations. In future, polynomial equations of any degrees can be recognized and solved using CNN model.

Ankita [4] used CNN for recognition of handwritten mathematical expression by performing various operations on dataset. Their primary aim was to build CNN model for feature extraction. Softmax activation function is used for classification. MINIST and NIST datasets are used for digits. System can solve quadratic equations only. In upcoming days, system should be able to solve various kinds of polynomial equations.

Prathamesh [5] discovered handwritten mathematical expression solver to recognize handwritten equations quickly and accurately. Image of equation is given as input to the system. After performing various steps like noise removal, segmentation and feature extraction, output equation is converted into LaTeX format. At the end, solution is displayed. In this system CNN is used for character classification. Classification is achieved by using softmax layer of CNN model. In the future, they will try to increase accuracy of given method. In addition to this, SVM or ELM algorithms can also be used for classification of symbols used in equation.

Md Bipul Hossain [11] developed a system which is able to recognize handwritten quadratic equations and further recognized equation is solved using string operations. In this project, image of handwritten quadratic equation is taken as input then it is recognized, solved and finally output is given back to the user. To design this system, they also used CNN model and they prepared dataset for accurate results. System is able torecognize to multiple equations from an image but unable to solve them simultaneously. By using Convolutional neural network, mathematical formula dictionary can be made.

3. Proposed Methodology

In the proposed model, we have used a customised neural network to train the model. It doesn't restrict any n-dimensional data and produces better results. Customised Convolutional Neural Network (CCNN) extracts features automatically and handles large amount of data thus resulting in better accuracy. Initially, noise from the earlier input image is decreased with the help of our proposed method by virtue of binarization of it. Next, from this input image, we apply projection horizontally for the equation to be segmented. Now, for later processing, each newly segmented image from the original one will be considered as a single new image. Now from the input image of the equation we will search the certain attribute in the form of associated components from every line of the image provided. After this, each segmented character is sent into our neural network model i.e. CNN, for character indexing. The output character given by the CNN model, is now used to create a character string that looks similar the earlier input equation.

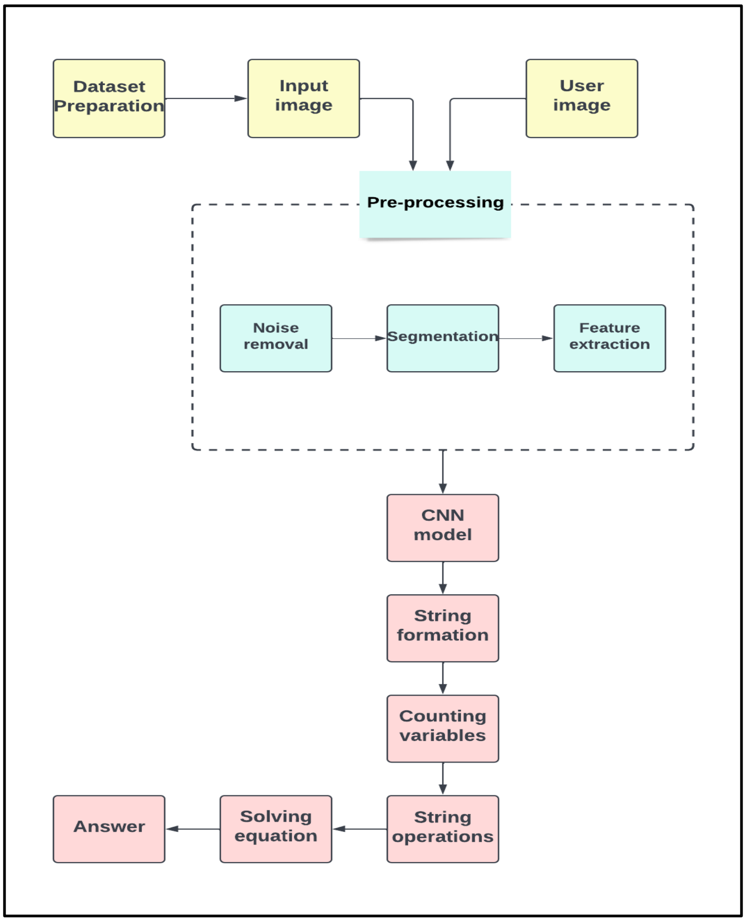
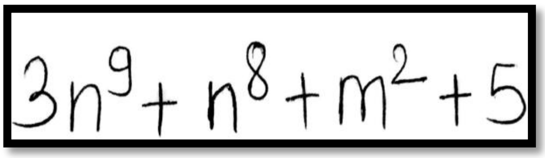


Fig. 1: System Architecture

The proposed system architecture of our handwritten polynomial equation solver using a customised convolutional neural network is shown above in fig 1.

3.1. Processing

In this essential step of beforehand processing the image which is provided as an input to the model is transformed and modified to make it of satisfactory standard or value for the recognition as an aim. Firstly, transform earlier input image to the grey-scale image that's because it gets extremely effortful to identify the character in the coloured image. Now getting rid of the unessential things from the grey-scale image will turn it into more satisfactory standards. Finally binary inverse of an image must be performed. Hence all the values of the pixels will be turned from 0 to 255. This helps us to abate our processing time that is required to segment the image. Further elimination of every unessential value of the pixel is done to a great extent. Input image and pre-processed images have been shown below respectively.



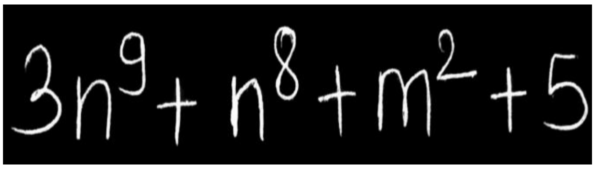


Fig 2(a). Input Image Fig 2(b). Post pre-processing

Input image fig 2(a) when given to model is processed to grey scale image and post processed image fig 2(b) respectively has been shown above.

3.2. Segmentation

Segmentation is quite seldom used in the Image processing to detect objects or important features and information in an image. It is the partition of one image into various parts. In the proposed methodology of ours, technique utilized to obtain peculiar images of the character from the original image of the equation known as the segmentation of character. As the pre-processed image has the pixel values which are binary in nature only, this makes it convenient to find the indexes belonging to the character. There is a bare minimum gap between every character. For locating or identifying a character, calculation of the totality is performed of all the pixels. Here all the columns of the image are considered in the summation. It will be known as gap; in case the sum of all columns is equal to

5 or less than it and will be suspected as a character. After identifying the gaps present linking every character. This will become very convenient to detach the images. At last, the size of 28x28 should be maintained in the newly segregated images.

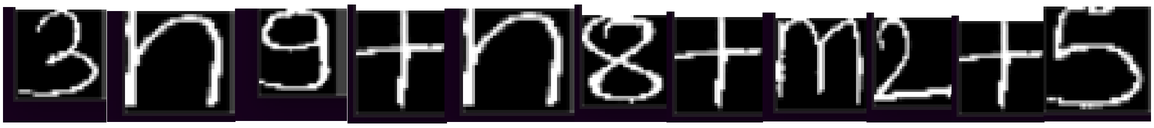


Fig 3. Segmented 28×28 pixels character images

Above images of fig 3 are maintained at a size of 28x28 shows the segmentation of a previously grey scaled image.

3.3. Dataset preparation

This is the quite important facet of this project. The borders of the character like numeral, alphabet in English, and mathematical symbols are all precisely defined. Result of this is, we started preparing the dataset with the great importance given to the edges of images that is to lighten up the edges. We have created our own dataset for the alphabets consisting of ‘m’, ‘n’, ‘v’, ‘w’, ‘d’ and ‘c’, also the popular MINIST dataset for the numerals. For the sake of training our network, we used total 54000 images from which 80% of data is used for training purposes and remaining 20% data for testing. Most of the cases of our training model were accurate up to 99.19%. For worthier computation, all the images are resized into 28x28 pixels and retained in the CSV format. Sample of the images is given below.

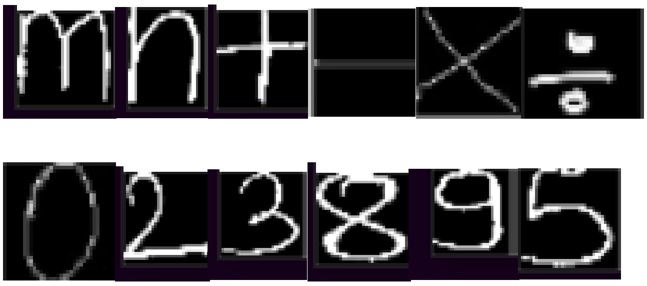


Fig 4. Dataset Sample Images

Sample of the images is given from our dataset in shown above in fig 4, in which the dataset of 6 alphabets is created by us.

4. Convolutional Neural Network Architecture

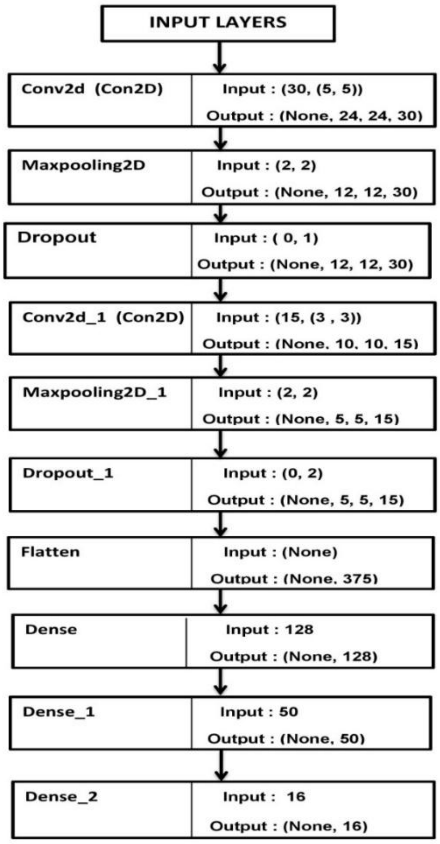


Fig 5. Architecture of model

The above fig 5 is the representation of our customized neural network which has ensure us increased accuracy.

Learning algorithm for Convolutional Neural Network:

Step I: In the first step of the training, 28x28 images are provided as an input layer. Further on the construction of a conv2d layer is done with 30 as the filter size and kernel size as 5x5. Also, the activation function named ReLU is applied in this layer.

The ReLU function is given below:

$$f(z) = \max(0,z)$$

(1)

where z is any integer value.

ReLU is non-linear and simple activation function with minimal computation. ReLU assists in forbidding the rampant advancement which is observed in neural network computation.

Step II: The output acquired from the preceding layer is utilized as an input for this layer. Here summarization of features takes place using the maxpooling2d layer; it is then linked with dropout. Adding this dropout layers has helped to reduce the overfitting and this has helped us to gain more accuracy than the previous architectures.

Dimension (h*w*c) of feature map will produce the following output:

$$\frac{(h-f+1)}{s*(w-f+1)*c} \qquad (2)$$

In calculation of 2d max-pooling height, width, total channels of feature map, filter size and length of stride are taken into consideration.

Step III: Now the output of the preceding layer will be the input for this layer. The current layer is the second convo2d layer. This layer has the padding of 10, filter size of 15 and 3x3 as the kernel size.

The convolution is mathematically expressed as below:

$$Z = X * f(3)$$

Here, X and f represents input image and filter respectively.

Convolution layer is used for feature extraction by taking image matrix and filter/kernel as input.

Step IV: Once more the maxpooling2d is performed where again the output will consists of feature map of important features of earlier feature map. Further on it is connected to the second dropout layer.

Dropout is expressed as follow,

$$U\, diag(r)V^T x = \begin{pmatrix} u_{1,1} & \cdots & u_{9,1} \\ \vdots & \ddots & \vdots \\ u_{19,} & \cdots & u_{9,9} \end{pmatrix} \begin{pmatrix} r_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_9 \end{pmatrix} \begin{pmatrix} h_1 \\ \vdots \\ h_9 \end{pmatrix} = \begin{pmatrix} r_1 u_{1,1} & \cdots & r_9 u_{9,1} \\ \vdots & \ddots & \vdots \\ r_1 u_{1,9} & \cdots & r_9 u_{9,9} \end{pmatrix} \begin{pmatrix} h_1 \\ \vdots \\ h_9 \end{pmatrix}$$

(4)

Step V: Completing the above layer processes, the output goes to the only flatten layer of the model; where the data is converted to a one-dimensional array.

Step VI: Finally, the dense layers come into play. Here three dense layers are implemented on the output of previous layer. All the three dense layers such as dense, dense_1 and dense_2 have the output nodes as 128, 50 and 16 respectively. For the first two dense layers the ReLU activation function is used while for the last dense layer the softmax activation function has been used.

The softmax function is given below:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}(5)$$

5. Experimental Results

We required 4 GB of RAM and a total of 500 GB of hard disk space. This is the minimum hardware requirement of the project. We have used different sets of characters and digits for our project.

After training our model we have used 10800 images for testing our model. We get 99% accuracy in the testing. After recognition of the image given by the user, we send the result to solve that equation. For every segmented peculiar numeral, arithmetic symbols and alphabets from the earlier image, prediction is performed consecutively one after the other and output is stored in a list. From the given input image, we count the characters from every single image which helps to perform the string operations. By using the eval function we solved the recognized equation.

We pre-process the string into equation form as below.

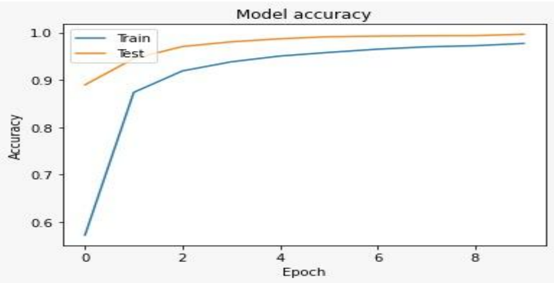
```
eq = ‘3n9+n8+m2+5’
Variable = m,n
Any int = 1,2,3,4, … , 9

if s[i].isalpha():
        if s[i+1].isnumeric():
                s[i]+='**'
if s[i].isnumeric():
        if s[i+1].isalpha():
                s[i]+='*'
```

Final generated equation is,

Equation = 3*n**9+n**8+m**2+5

After this we are taking values of variables from the user and solving equations using python eval function.

After executing 10 epochs this model has acquired 97.69% accuracy on training and 99.19% accuracy on validation. From our model we conclude that our model performs well and produce output correctly. Following graphs show model accuracy and model loss.



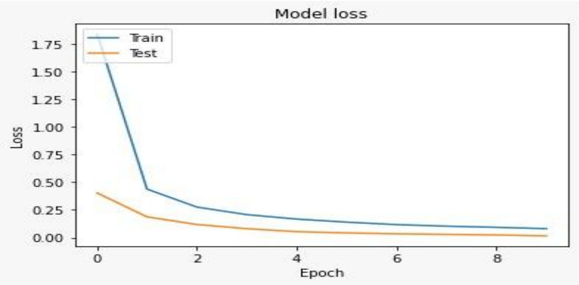
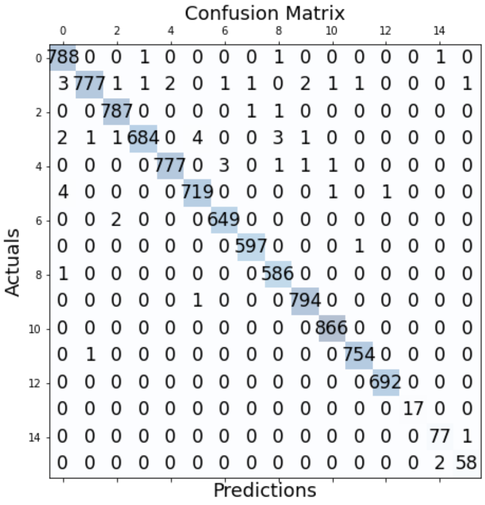


Fig 6(a). Accuracy graph

Fig 6(b). Loss graph

Above figures 6(a) and 6(b) are the graphs of our model which shows model accuracy and model loss with train and test data.



	0		2		4		6		8		10		12		14	
0	788	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
	3	777	1	1	2	0	1	1	0	2	1	1	0	0	0	1
2	0	0	787	0	0	0	0	1	1	0	0	0	0	0	0	0
	2	1	1	684	0	4	0	0	3	1	0	0	0	0	0	0
4	0	0	0	0	777	0	3	0	1	1	1	0	0	0	0	0
	4	0	0	0	0	719	0	0	0	0	1	0	1	0	0	0
6	0	0	2	0	0	0	649	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	597	0	0	0	1	0	0	0	0
8	1	0	0	0	0	0	0	0	586	0	0	0	0	0	0	0
	0	0	0	0	0	1	0	0	0	794	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	866	0	0	0	0	0
	0	1	0	0	0	0	0	0	0	0	0	754	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	692	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	58

Fig 7. Confusion matrix

Above fig 7 shows the 16x16 confusion matrix of our model which is used for evaluating the performance of proposed model where 16 is the number of target classes.

Table 1. Comparative analysis

Title	Method used	Dataset	Accuracy
2013[8]	PHOG features	CROHME, 22000 samples	92%
2015[5]	CNN	CROHME, 25165 samples	60%
2016[6]	CNN	SymCROHME, 10584 samples	87.53%
2018[2]	CNN	MINIST, 20000 samples	39.11%
2019[7]	LeNet and SqueezNet	CROHME, MINIST, 26000 samples	90%
2022[1]	Dense model	MINIST, 49000 samples	98%
Proposed work	Customised CNN	MINIST(for digits), Self prepared(for variables), 54000 samples	99.19%

Table 1. is comparing results achieved by previously proposed architecture along with their accuracy.

6. Conclusion and Future Work

This paper aims to extract features and recognise the polynomial equations provided to the model by the user, up to the power of 9 and providing their solution with enhanced accuracy. We have implemented the sequential model of CNN. Other methods generate flawed results due to unwanted noise. In comparison to LeNet model, CNN yields better accuracy and avoids the overfitting. Adding dropout layers after pooling layer helps to reduce noise and in turn gives higher accuracy of 99.19% in comparison to previous CNN architectures.

The only limitation faced by the model is, if the exponent is written too high above the root variable then that exponent might have a chance to go undetected.

In upcoming days, we look ahead to implement this technique on major calculus concepts such as integration and differentiation.

7. References

[1] Rajwardhan Shinde, Onkar Dherange, Rahul Gavhane, Hemant Koul, Nilam Patil, “Handwritten Mathematical Equation Solver”, International Journal of Engineering Applied Sciences and Technology (IJEAST), Vol. 6, Issue 10, pp 146-149, February 2022.

[2] Mohini Lokhande, Kirti Murudkar, Arbaaz Nadaf and Niyati Shah ,“Handwritten Math Problem Solver Using Convolutional Neural Network”, International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Vol. 10, Issue 5, pp 675-678, May 2021.

[3] Deepak T Mane, Rushikesh Tapdiya, Swati V Shinde, “Handwritten Marathi numeral recognition using stacked ensemble neural network”, International Journal of Information Technology (IJIT), July 2021.

[4] Ankita Kawade, Prof. Vrushali Dhanokar, “Handwritten Equation Solver Using CNN”, International Research Journal of Engineering and Technology (IRJET), Vol. 8, Issue 12, pp 1284-1287, December 2021.

[5] Prathamesh Tope, Sarvesh Ransubhe, Mohammad Abdul Mughni, Chinmay Shiralkar and Mrs. Bhakti Ratnaparkhi, “Recognition of Handwritten Mathematical Expression and Using Machine Learning Approach”, International Research Journal of Engineering and Technology (IRJET) Vol. 08, Issue 12, pp 1437-1443, December 2021.

[6] Shifat Nayme Shuvo, Fuad Hasan, Syed Akhter Hossain and Sheikh Abujar “Handwritten Polynomial Equation Recognition and Simplification Using Convolutional Neural Network”, 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp 1-6, 1st July 2020.

[7] Ritik Dixit, Rishika Kushwah and Samay Pashine, “Handwritten Digit Recognition using Machine and Deep Learning Algorithms”, International Journal Of Computer Applications (IJCA), Vol. 176 no. 42, July 2020.

[8] Azadeh Nazemi, Niloofar Tavakolian, Donal Fitzpatrick, Chandrika Fernando and Ching Y. Suen, “Offline handwritten mathematical symbol recognition utilising deep learning”, Computer Vision and Pattern Recognition, 16 Oct 2019.

[9] A.S.V.Praneel,T.Srinivasa Rao, M.Ramakrishna Murty “ A survey on Accelerating the classifier Training Using various boosting schemes within casecades of bossted ensembles” International Conference with Springer SIST series, Vol. 169, pp 809-825, 2019.

[10] DT Mane, UV Kulkarni, “Visualizing and understanding customized convolutional neural network for recognition of handwritten Marathi numerals”, Procedia computer science, pp 1124-1137, 2018.

[11] Md Bipul Hossain, Feroza Naznin, Y.A. Joarder, Md Zahidul Islam and Md Jashim Uddin, “Recognition and Solution for Handwritten Equation Using Convolutional Neural Network”, Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Vol. 7, pp 250-255,2018.

[12] Lyzandra D’souza and Maruska Mascarenhas, “Offline Handwritten Mathematical Expression Recognition using Convolutional Neural Network”, International Conference on Information, Communication, Engineering and Technology (ICICET), August 2018.

[13] S. Anandh Kishan, J. Clinton David and P.Sharon Femi, “Handwritten character recognition using CNN”, International Journal of Research and Analytical Reviews (IJRAR), Volume 5, Issue 3, pp 241-245, September 2018.

[14] Pillai, Arvind, Rajkumar Soundrapandiyan, Swapnil Satapathy, Suresh Chandra Satapathy, Ki-Hyun Jung, and Rajakumar Krishnan, "Local diagonal extrema number pattern: A new feature descriptor for face recognition." Future Generation Computer Systems 81, pp 297-306, 2018.

[15] N. A. Hamid and N. N. A. Sjarif, “Handwritten recognition using SVM, KNN and neural network”, Computer Science, February 2017.

[16] Hai Dai Nguyen, Anh Duc Le and Masaki Nakagawa, “Recognition of Online Handwritten Math Symbols using Deep Neural Networks”, IEICE Transaction and Systems, Vol. 99-D no. 12, pp 3110-3118, October 16.

[17] Y. Shkarupa, R. Mencis, and M. Sabatelli, “Offline handwriting recognition using LSTM recurrent neural networks,” 28th Benelux conference on artificial intelligence (BNAIC), pp. 88–95, November 2016.

[18] Sagar Shinde and Rajendra Waghulade, “Handwritten Mathematical Expressions Recognition using Back Propagation Artificial Neural Network”, Communications on Applied Electronics (CAE) – ISSN : 2394-4714 Foundation of Computer Science FCS, New York, USA Volume 4– No.7, March 2016.[5] Catherine Lu and Karanveer Mohan, “Recognition of Online Handwritten Mathematical Expressions Using Convolutional Neural Networks”, cs231n project report Stanford, 2015.

[19] Nicolas D. Jimenez and Lan Nguyen, “Recognition of Handwritten Mathematical Symbols with PHOG Features”, Computer Science, 2013.