Evaluation of Handwritten Arithmetic Equations using Convolution Neural Networks

Chandra Sekhar Sanaboina[1*], Rahul Sidda[2], Ahmed Akram[3], Sruthi Samuel Sarella[4], Jennifer Rogers[5]

[1]Assistant Professor, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Kakinada, India

[2]UG Student, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Kakinada, India

[3]UG Student, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Kakinada, India

[4]UG Student, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Kakinada, India

[5]UG Student, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Kakinada, India

[1]chandrasekhar.s@jntucek.ac.in, [2]rahulsidda133@gmail.com, [3]devahmedakram@gmail.com, [4]sruthi.sarella467@gmail.com,[5]janetrogers266@gmail.com

Article Info
Volume 83
Page Number:7613 - 7622
Publication Issue:
May-June 2020

Article History
Article Received: *19 November 2019*
Revised: *27 January 2020*
Accepted: *24 February 2020*
Publication: *18 May 2020*

Abstract:
This paper aims to develop a user interactive assistance system that evaluates the handwritten arithmetic equations based on handwriting recognition algorithms. Although recognizing handwritten characters and symbols is generally easy for anyone but recognizing them is difficult for a machine. By following a deep learning approach, this challenge can be solved by designing a system that recognizes the operands and operators. Being able to solve handwritten arithmetic equations through the model will bring faster and accurate results. The model will identify pictures of handwritten arithmetic equations and will be able to emit the corresponding characters into a list and evaluates the results.This includes digit classification which involves feature extraction and classification. For this purpose, computer vision is used to input the image and obtain contours, Convolutional NeuralNetwork(CNN) is the algorithm used to build the model, which does feature extraction and classify the operators and operands and develop a model with 92.6% accuracy.

Keywords - Computer Vision (CV), Feature extraction, Contour extraction, Convolutional Neural Network (CNN), TensorFlow, Training model, Keras, ReLu, Softmax.

I. INTRODUCTION

Handwritten character recognition is being very challenging in the field of image processing and pattern recognition. Handwritten equations can be extracted from the images and solved. Automation of handwritten equation solving brings many advantages. This system can be used for paper evaluation and could replace the calculator. The image that contains the equation is given as an input. To identify the equation in an image, Computer Vision plays an important role. Computer Vision defines how an image is to be read by the system and to do so, the system needs to extract each character and recognize them. Open CV is a library, that extracts boundaries for each character. This image is processed into a grayscale image using OpenCV. This image is sent into a Convolutional Neural Network that identifies the patterns in the image. CNN is an artificial neural network that identifies features and extracts patterns from them. The major components of the project are data extraction and training. A comprehensive set of images that contain handwritten operands and operators is used as a data set. The data from the data set is extracted and the model is trained by the extracted dataset. Once the model is trained it could identify the operands and operators in the image and can solve them.

CNN is an efficient way of identifying patterns from an image. In the implemented system, CNN does three major tasks - Convolution, Activation, and Pooling.

- Convolution refers to both the result function and to the process of computing it.
- Activation refers to the use of activation functions to decide the network output, and
- Pooling is performed to slowly decrease the representation's spatial scale to minimize the number of parameters and calculations within the network.

CNN uses Tensorflow as a backend. Tensorflow is an open-source machine learning platform and is widely used to implement neural networks. Tensorflow helps to train the model to identify patterns in the image. The image processing is done by using Open CV. The Open CV is a library that is widely used for image processing. It is used to determine the contours in an image. A contour is a line drawn along the boundary of a feature holding the same color intensity. The image is read by the model as a matrix of integers. The input image is in the form of an RGB image, which has three channels whereas a grayscale image has only one channel. In this system, the RGB image taken is transformed into a grayscale image which is the second step, then the features are extracted from the image. The identification of contours is done, this is the third step. In the next step, the CNN model extract features and then classify them. This process includes a filter with relatively less size than the image matrix is taken. This filter strides over the image matrix from left to right and again hops to the left side of the matrix in the next row moving all over the matrix. This filter contains certain stride values such that each filter gives different outputs, thus recognizing different patterns. Once the full matrix is traversed, an activation function called ReLu is applied to allow non-linearity, and the image pattern is determined. This undergoes a pooling layer where the dimension of the matrix is reduced. The next step is the fully connected layer, where it consists of a multilevel perceptron that holds the values. The softmax activation function is used to classify the operator/operand. The identified operator/operand is given to a string. The result is obtained when the evaluation function is applied to the string. Fig. 1 given below, depicts important steps involved during evaluating the system.

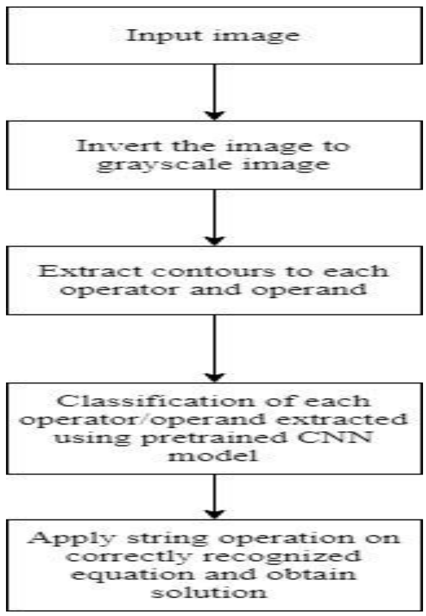

Figure 1: Workflow of the System

II. Related work

Zanibbiet. al [1] suggested an efficient and reliable method for the identification of mathematical symbols. Using three passes, the structure pass, lexical pass, and operator tree, this proposed method solves an expression. The manipulation of the tree used in each move which is expressed by the transformation of the tree. They used architecture DRACULAE which is a format to represent input style, makes use of the search function to decipher mathematical notation. This model is useful for dialect recognition, structure isolation of words, lexical analysis for HME syntax, and semantic analysis.

D’souzaet. al [2], in their paper, proposed a system to recognize HME (Handwritten Mathematical Equations) using neural networks for classification of characters. The input consists of datasets consisting of symbols and digits. Preprocessing steps and done to the datasets.

May – June 2020
ISSN: 0193-4120 Page No. 7613 - 7622

Classification is done using a neural network model and characters are identified.

Pradeepet. al [3] implemented a neural network recognition system, which consists of 54 characteristics of datasets and 69 feature functions. Horizontal and vertical feature extraction techniques are used to train this system. The efficiency of this system is increased by the diagonal feature extraction process. They used 6 separate networks to identify the characters. The results have shown that the diagonal approach has higher accuracy of 97.8% for 54 characteristics and 98.5% for 69 feature functions.

Quan Nguyen et. al [4], in their paper, stated that some of the machine learning algorithms like Support Vector Machine(SVM), Random Forests, and K - nearest neighbors algorithms for the detection handwritten characters. They have implemented some edge detection on the image. And obtained features from them. This result also showed a high rate of detection using the SVM algorithm rather than the other two in feature detection.

Jaderberget. al [5], used a new module for neural networks called, the spatial transformer. The input for this system is regressed transform parameters and these are used for subtasks to do in the system. This experiment has shown that it is effective when it comes to the case of concurrent models as it requires object reference frames and on the other hand it is easy to convert into a 3 dimension transform.

Attigeri [6] in his paper, attempted to recognize handwritten English letters without a function but using a feed-forward multilayer neural network. The dataset contains each set of all 26 alphabets. For neural network training, they have used 50 such sets of datasets. The system comprises of recognition and classification of a letter. Each letter is reduced to a size of 30 by 20, which is done at data pre-processing. These pixels serve as input to the neural network for classification. The accuracy of the system obtained is 85.5%.

III. METHODOLOGIES

3.1. DATA PRE-PROCESSING:

A dataset containing handwritten images of both operands and operators are loaded.Features are extracted using contour extraction. Contour tracing is a method applied to digital images to retrieve boundaries from them. Contours can be defined as a curve joining all the parallel points along the boundary with the same color or intensity.The input image is inverted into a grayscale image, which contains shades of gray and white for better feature recognition. We find contours using findContour() and boundaries around the image using boundingRect(). The bounding rectangle is the smallest rectangle surrounding the contour extracted. Since there are only operands and operators, there is a need to find a bounding rectangle with maximum area. Resize the maximum bounding rectangle to size 28 X 28 and reshape it to 784 X 1. It represents that there are 784-pixel features. Add labels to each operand and operator. Once features are extracted, save the extracted data to a .CSV file that contains the pixel data.

3.2. TRAINING THE MODEL:

The trained data is extracted to ConvolutionNeuralNetwork (CNN) model. CNN is a type of artificial neural network that is explicitly programmed to process pixel data for image recognition and processingpurposes. They are also used for image recognition, grouping, segmentation, and other auto-related details as well.Since CNN works on two-dimensional data, there is a need to reshape the one-dimensional data to two-dimensional data which is in the form of 784 X 1. A variable is assigned to the labels column extracted from the dataset and drop the labels column from the dataset and again reshape it to 28 X 28. This matrix serves as an input matrix to the model. Now the model is built. Fig. 2 shows the architecture of the CNN model which includes feature extraction as the first part followed by the classification of features extracted and obtaining the output from the output nodes.

May – June 2020
ISSN: 0193-4120 Page No. 7613 - 7622

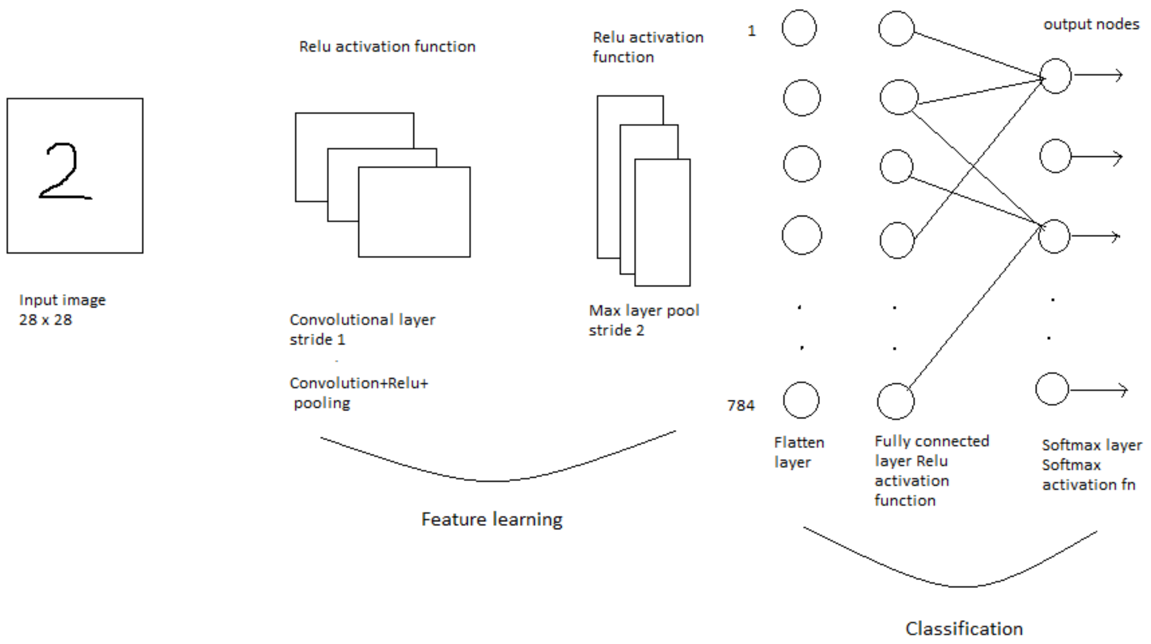

Figure 2: Architecture of CNN model

3.3. BUILDING THE MODEL:

Import required libraries to build up the model. Import Keras which is a high-level neural network API, which is capable of runningboth TensorFlow and Theano.TensorFlowwhich manipulates data by constructing a DataFlow graph. It consists of nodes and edges that work and control such as inserting, subtracting, multiplying, andso on. Convert the label data stored in the variable to categorical data using the function toCategorical().

The model built is sequential (prebuilt in Keras). This model consists of the Dense layer, predicts the labels. The dropout layer reduces overfitting and converts a three-dimension array to one dimension. The convolutional layer consists of a convolved feature in which a filter (relatively less matrix size) strides over the matrix to obtain the output matrix. The activation function used is ReLu. The pooling layer is responsible for reducing the dimension of the convolved feature matrix. And a fully connected layer which results in the classification of a pattern using the softmax activation function. This model consists of two steps: feature extraction and classification.

3.3.1 FEATURE EXTRACTION:

It involves obtaining features and patterns from the image. The steps involved in feature extraction are Convolution, ReLu Activation, Pooling. Convolutional layer, ReLu Activation function, and Pooling layer are used during feature extraction.

3.3.1.1. CONVOLUTIONAL LAYER:

The Convolutional layer is composed of a collection of filters. A filter is a relatively small size matrix than the input matrix. The main purpose of using the filter is to reduce the noise in the image and obtain only necessary parts of the image. The filter moves with particular stride value from left to right over the matrix and moves with the same stride value all over the matrix. The multiplication result of the input image and the filter is given by the feature map. Fig. 3 shows the convolution operation, the input image matrix of size 5 x 5 is taken, and it is convolved (matrix multiplication) with filter matrix of size 3 x 3. The result of this matrix multiplication gives the feature map of size 3 x 3 which holds the convolved matrix result, for example, 4 in the diagram below respectively. Many such filters stride on the input matrix resulting in different values, these values are summed up with a bias value to give the output. Once the whole matrix is traversed, the matrix is allowed to non-linearity, this is done using ReLu function.

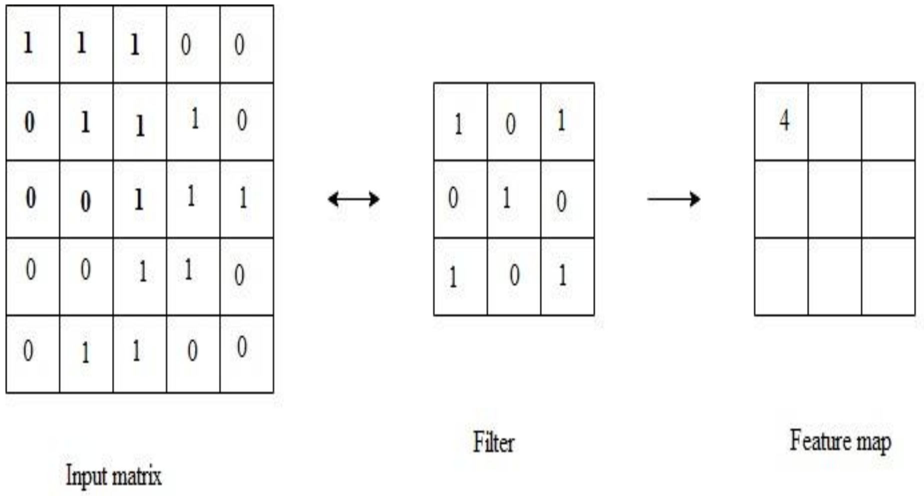

Figure.3: Convolution Process

3.3.1.2 ReLu ACTIVATION:

The output matrix obtained after the convolution process is given to the ReLu Activation function. This function allows non-linearity by converting negative values to zero. The ReLU is half rectified function. If f(z) is a function such that, f(z) is zero if z is less than zero and f(z) is equal to z if z is above or equal to zero. This activation function holds all positive values including zero (i.e., Range: [0 to infinity)).Then the matrix dimension is reduced using the Pooling layer.Fig. 4 represents how the values are changed after applying ReLu activation function.

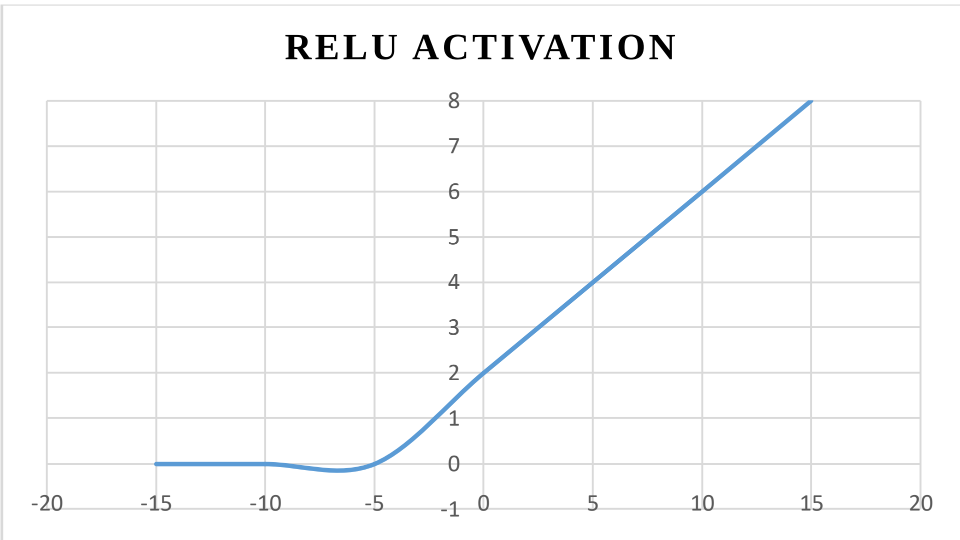

Figure 4: ReLuActivation Function

3.3.1.3 POOLING LAYER:

The pooling layer decreases the spatial size of the matrix. This process is done to increase computational power by reducing the size of the matrix. This can be done in two ways: average pooling and Max pooling. Here max pooling is used, maximum value from the relative 2 X 2 matrix is taken from the image matrix. Fig.5 shows how the feature map is reduced by dimension using max pooling. Here 2 x 2 matrix strides starting from the left and the maximum value is extracted to a 2 x 2 matrix. Thus, reducing the size of the matrix by half of its dimensionality.

May – June 2020
ISSN: 0193-4120 Page No. 7613 - 7622

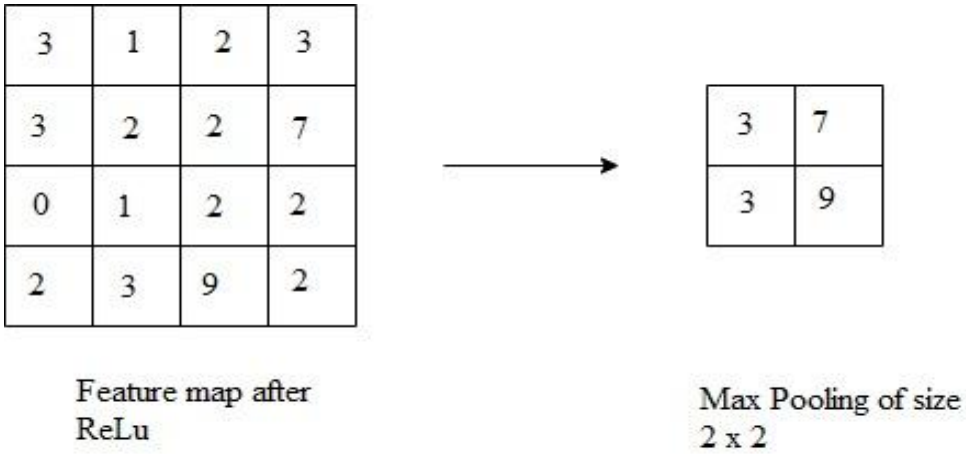

Figure 5: Max Pooling

This is again given to the second convolutional and pooling layer where the size of the matrix is almost reduced. This data need to be given to a fully connected layer for classification, the data need to be reshaped to do so. The matrix is reshaped and the data is flattened. The dropout layer with value 0.2 is added, stating that 20% of weights are dropped to zero. This is only done during the training of the model. The dense layer creates hidden layers containing neurons that hold the classification values and gives the output. Once features are extracted, classification is done on each pattern, and the image is recognized.

3.4. CLASSIFICATION:

The last step is to build a multilevel perceptron which holds all the flattened values of the input image. The output of one layer acts as input to the next layer. Backpropagation is applied after each training process, after several epochs, the model will be able to classify the features and identified using the Softmax Activation function.

3.4.1. SOFTMAX FUNCTION:

Softmax function results in the probability of an event over 'n' different events. It gives the maximum possibility of occurring the same feature. This function gives the probability distribution of the values in the dense layer so that all the values summed up to give a whole value 1. The value ranges between 0 and 1. The graph below shows the value of inputs on the x-axis to their corresponding softmax scores on the y-axis and graph between the isa plot. The range also varies from 0 to 1.

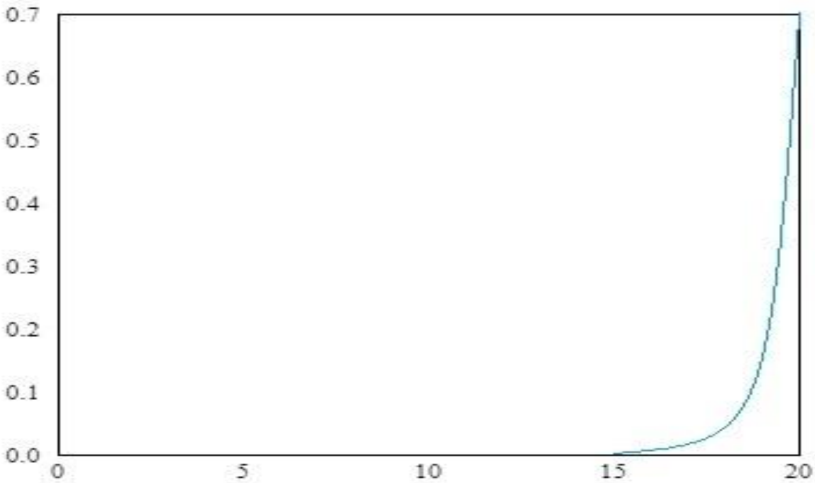

Figure 6: Softmax Activation Function

Now fit the model to data using model.fit() function. The model is ready to classify new instances. When an input image is given to the system. It converts it into a grayscale image. Contours are extracted. Now using the pre-trained model we predict each character using 'model.predict_classes()' which predicts the pattern with the final trained model obtained. The pattern recognized is compared to the labels of operators and operands, if matched, label values are added to a string. This string holds the equation. The result is obtained by applying eval() on the string.

IV. RESULTS AND DISCUSSIONS

This model can now recognize handwritten arithmetic equations. An image consisting of an arithmetic equation is given as input to the system. For example,the image given in Fig. 7 is taken as an input image.

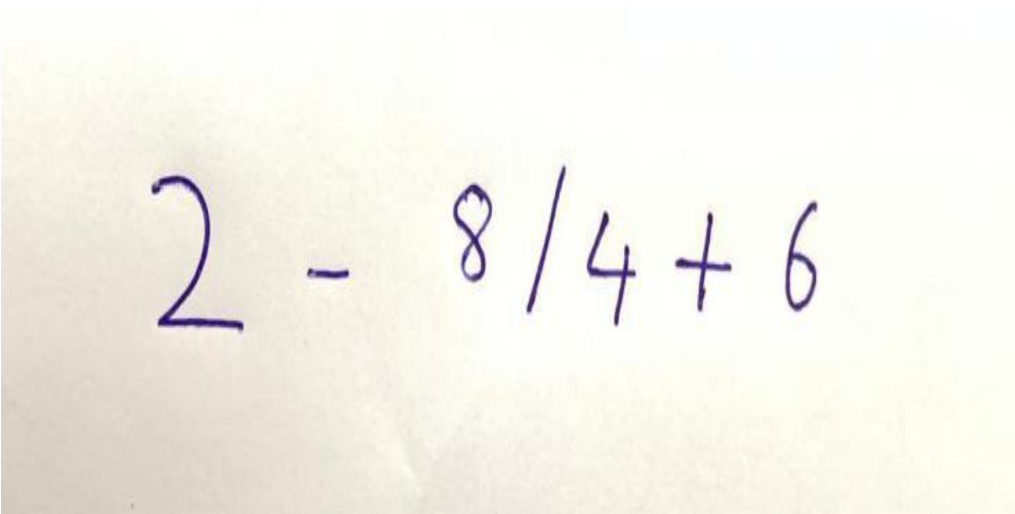

Figure 7: Input Image

The user interface of the proposed system is shown in Fig 8. The system contains a display window that has two buttons, one to upload the image path and the other to solve the equation. First, the image path is uploaded. Then solve button is clicked.

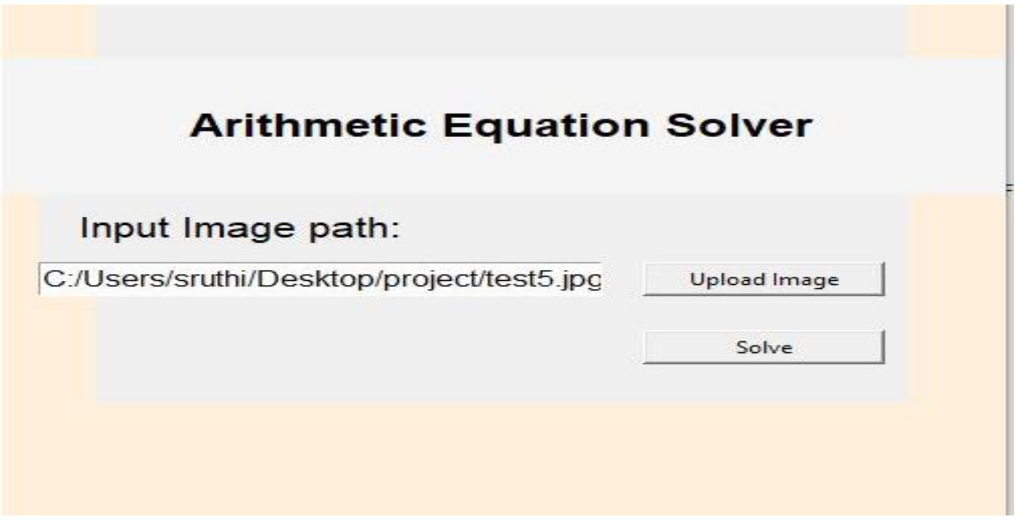

Figure 8: Input Window

Now, the system converts the input image into a grayscale image. Then, contours are extracted for each operand and operator as shown in Fig 9. (from left to right by default) below using the bounding rectangle.

May – June 2020
ISSN: 0193-4120 Page No. 7613 - 7622

Figure 9: Contours Extraction

Using the model, classify the corresponding operators and operands for each contour extracted using the bounding rectangle. Compare each correctly recognized operands and operators to the labels. Append each recognized label name to the string. This string contains the arithmetic equation, which is solved using a predefined function in the python eval() function. The result of this function is displayed on the output window along with the arithmetic equation. The output window is shown in Fig. 10.

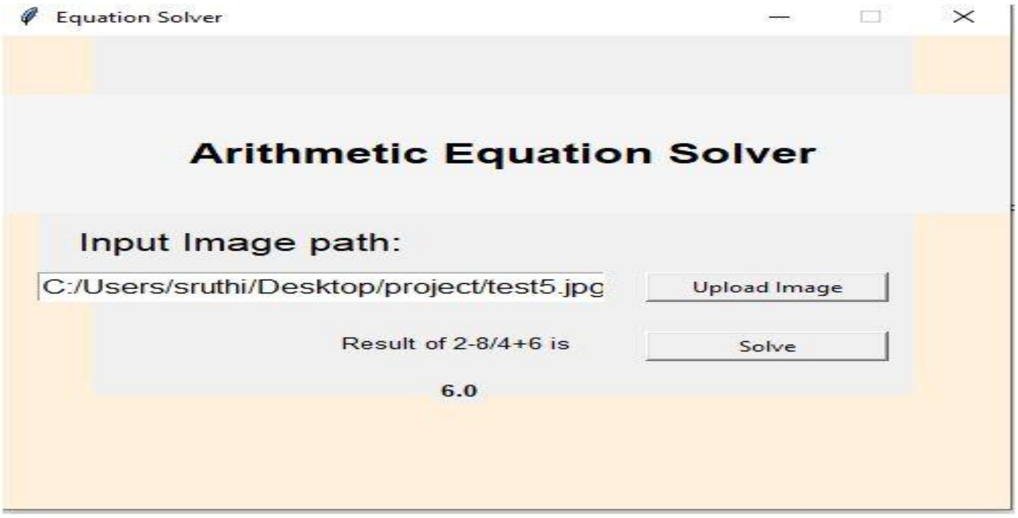

Figure 10: Output Window

V. CONCLUSION

This paper mainly focused on identifying handwritten arithmetic equations. Computer vision(cv2) has been used for recognizing the features in the image. Contours are extracted for each operator/operand recognized. The most efficient classification model used in the classification of operators/operands is the Convolutional Neural Network (CNN). And it is the most complicated part since different users writeoperators in different ways. Once successful recognition of operands and operators is done, the further process is to find the solution of the equation, which includes, passing the recognized string to a predefined function in python called eval() to evaluate the results. The accuracy of the system achieved is 92.6%. This application can be used to evaluate mathematical calculations on the answer sheet and can also replace the calculator. It acts as a virtual tutor for the students. Students can verify their answers by just clicking the picture of the equation.

VI. FUTURE WORK

At present, this technique is applied to handwritten arithmetic equations written on white paper alone but in the future, it can be extended by solving equations that are written on any type of paper. It can be achieved by implementing some noise reduction techniques to remove noise on paper. Training with better datasets and implementing new deep learning algorithms further can increase the accuracy of the system.

REFERENCES

[1] D. Blostein, E. Lank, A. Rose, and R. Zanibbi, “User Interfaces for On-Line Diagram Recognition,” in *Graphics Recognition Algorithms and Applications*, 2002, pp. 92–103.

[2] D’souza, Lyzandra, and MaruskaMascarenhas. “Offline Handwritten Mathematical Expression Recognition using ConvolutionalNeural Network.” *2018 International Conference on Information, Communication, Engineering, and Technology (ICICET)* (2018): 1-3.

[3] I. Based, X. For, and Y. Using, “Diagonal Based Feature Extraction for Handwritten Alphabets Recognition,” *Int. J. Sci. Inf.*

Technol., vol. 3, no. 1, pp. 27–38, 2011, DOI: 10.5121/ijcsit.2011.3103.

[4] Q. Nguyen, M. Wang, and L. C. Fan, "On the Recognition of Handwritten Math Equations," pp. 1–5, 2014.

[5] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," *Adv. Neural Inf. Process. Syst.*, vol. 2015-Janua, pp. 2017–2025, 2015.

[6] S. Attigeri, "Neural Network-based Handwritten Character Recognition system," *Int. J. Eng. Comput. Sci.*, vol. 7, no. 03, pp. 23761–23768, 2018, DOI: 10.18535/ijecs/v7i3.18.

[7] S. M. Shamim, M. B. A. Miah, A. Sarker, M. Rana, and A. Al Jobair, "Handwritten digit recognition using machine learning algorithms," *Indones. J. Sci. Technol.*, vol. 3, no. 1, pp. 29–39, 2018, DOI: 10.17509/ijost.v3i1.10795

[8] A. K. Seewald, "Digits–A dataset for Handwritten Digit Recognition," *Austrian Res. Inst. Artif. Intell. Tech. Rep.*, no. January 2005, 2005,

[9] Z. Rao*et al.*, "Research on a handwritten character recognition algorithm based on an extended nonlinear kernel residual network," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 1, pp. 413–435, 2018, DOI: 10.3837/tiis.2018.01.020.

[10] M. Zohra and D. RajeswaraRao, "A comprehensive data analysis on handwritten digit recognition using machine learning approach," *Int. J. Innov. Technol. Explore. Eng.*, vol. 8, no. 6, pp. 1449–1453, 2019.

AUTHORS PROFILE

First Author: Chandra SekharSanaboinais currently working as an Assistant Professor in the Department of Computer Science and Engineering from JNTUK University Kakinada. He obtained his B.Tech. In Electronics and Computer Science Engineering in 2005 and M.Tech in Computer Science and Engineering from Vellore Institute of Technology in 2008. He has over 10years of teaching experience and currently working as an Assistant Professor in the Department of Computer science and Engineering, JNTUK Kakinada. His areas of interest include Wireless Sensor Networks, Internet of Things, Machine Learning, and Artificial Intelligence.

Second Author: Rahul Sidda is currently studyingB.Tech in the Department of Computer Science and Engineering of Jntuk, Kakinada. He has completed B.Tech under the supervision of Sri.S.ChandraSekhar. Assistant Professor of CSE, JNTUK – Kakinada. His areas of interest include Artificial Intelligence and the Internet of Things.

Third Author: Ahmed Akram is currently studyingB.Tech in the Department of Computer Science and Engineering of Jntuk, Kakinada. He has completed B.Tech under the supervision of Sri.S.ChandraSekhar. Assistant Professor of CSE, JNTUK – Kakinada. His areas of interest include Graphic design, UI design, and Web Development.

Fourth Author: Sruthi Samuel Sarella is currently studyingB.Tech in the Department of Computer Science and Engineering of Jntuk, Kakinada. She has completed B.Tech under the supervision of Sri.S.ChandraSekhar. Assistant Professor of CSE, JNTUK – Kakinada. Her areas of interest include Data analyticsand Artificial Intelligence.

Fifth Author: Jennifer Rogers is currently studyingB.Tech in the Department of Computer Science and Engineering of Jntuk, Kakinada. She has completed B.Tech under the supervision of Sri.S.ChandraSekhar. Assistant Professor of CSE, JNTUK – Kakinada. Her areas of interest include Artificial Intelligence and Ethical hacking.

May – June 2020
ISSN: 0193-4120 Page No. 7613 - 7622