© 2019 IJSRSET | Volume 6 | Issue 2 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology
DOI : https://doi.org/10.32628/IJSRSET1962169

Handwritten Gurmukhi Digit Prediction using Convolutional Neural Network with Keras and Tensorflow

Sonia Flora[1], Divya Ebenezer Nathaniel[2]

[1]Department of Computer Science and Engineering, PIET, Parul University, Vadodara, India
[2]Assistant Professor, Department of Computer Science and Engineering, Babaria Institute of Technology Vadodara, India

ABSTRACT

Intelligent Character Recognition is a term which is specifically used for the recognition of handwritten character or digit. It is a prominent research area of computer vision field of machine learning or deep learning which trained the machine to analyze the pattern of handwritten character image and identify it. Recognition of handwritten character is a hard process because single person can handwrite the same text in number of ways by making a little variation in holding the pen. Handwriting has no specific font style or size. It differs person to person or more specifically it differs how one is holding the pen. Deep Leaning has brought the breakthrough performance in this research area with its dedicated models like Convolutional Neural Network, Recurrent Neural Network etc. In this paper, we have trained model with Convolutional Neural Network with different number of layers and filters over 10,559 handwritten gurmukhi digit images and validate over 1320 images. Consequently we could achieve the maximum accuracy of 99.24%.

Keywords : Gurmukhi Digit Recognition, Handwritten Digit, Prediction, Convolutional Neural Network, Keras

I. INTRODUCTION

India is moving slowly towards complete automation world. Automation means almost 99% of the tasks are performed by the machines. This will improve the maximum efficiency in operation and administration of a task. There are other numerous benefits of automation like office automation. Despite of transforming into paperless world, there are numerous situation where handwrite document are mandatory part of the task. To convert the handwritten, printed or typed document into computer editable form is a process of interest for researchers since 4 decades. Optical Character Recognition (OCR) technique was used by the researchers to convert any physical document into computer editable form. The area of OCR is becoming an integral part of many applications such as postal processing, script recognition, banking, security (i.e. passport authentication) and language identification. To convert the typed or printed document in to computer recognizable form is an easy task because of certain fixed parameters such as font size and font type. But to convert the handwritten document into computer recognizable form is a tough process because no two handwritten same characters are identical no matter if written by single person. In modern world, Intelligent Character Recognition (ICR) is a term specifically used for handwritten character recognition. OCR technique is a process of 5 stages; every stage is equally significant and necessary for the generalization and better accuracy of the trained model [2]. First stage is Digitalization or computer readable form; it is to convert the physical document into digital form i.e. information is represented in the form of bits then to bytes and so on. And the resultant representation will be saved either in the form of image, document or sound extension. In OCR process, the physical document will be saved in the form of image after scanning. Second stage is pre-processing or preliminary processing of the scanned document image for the enhancement of the image quality and suppress the unwanted distortions using various techniques like contrast stretching, smoothing, sharpening, histogram processing, image threshold to remove background, de-noise to remove water marks , skew detection & correction etc. It does refine the input image for further processing. Third stage is Segmentation; it means to partition the image into multiple segments. It can be applied in a number of ways like text segmentation from the graphic background, character segmentation, word segmentation, line segmentation or page segmentation depending upon the requirement. Fourth stage is Feature extraction; it is one of the most important stages of the OCR process. It can be defined as the dimensionality reduction stage which converts the raw values represents an image in to reduced set of values with various manual feature extraction techniques and stored in feature vector so that while processing the task can be performed with this feature vector and this feature vector still accurately describing the original data. In general it is a process to reduce the amount of data or keep only the data of interest else more memory and processing power will be required and the issue of over fitting can also occur. But with the development of the deep learning models, manual feature extraction stage has been vanished because deep learning models have done feature extraction automatically. Deep learning has categorized its models according to its applications like Convolutional neural network model for computer vision application, Recurrent Neural network model for time series application like speech recognition etc. Fifth stage of OCR process is Classification model or classifier; it is defined as the process of predicting the class label of unseen data when trained with similar dataset. Convolutional neural network classifier, a model of deep learning is used for the pattern recognition or handwritten character recognition. It uses the convolutional function with the filter or kernel and learns the input image data with a sequence of layers and gives the better result than primitive classifier.

II. RELATED WORK

This section briefly reviews the literature survey on offline handwritten character recognition models of various Indic and non-Indic scripts. Sidharth et al. [4] have used Zonal Density and Background Distribution as feature extraction technique and trained three different models Support Vector Machine(SVM) classifier with RBF kernel, K-Nearest Neighbour Classifier(KNN), Probablistic Neural Network(PNN) classifier over 200 samples of handwritten 35 different gurmukhi characters. They got a highest accuracy of 95.04% with SVM classifier evaluated using 5-fold cross validation evaluation. Singh G. et al.[5] have implemented an experiment on first 5 ‘vyanjans’ or consonants of Hindi Character set and trained multilayer neural network model over 1000 handwitten sample images and yields recognition accuracy of 93 %. Kumar et al. [6], have used the four types of topological features extraction technique for offline handwritten Gurmukhi characters and trained the SVM classifier over 3500 sample images. They also compared the accuracy results obtained from SVM classifier by using different feature selection methods like PCA, CON and CFS. Verma et al. [7], have experimented Zone identification algorithm and rule based approach for strokes identification on Online handwritten Gurmukhi character recognition and trained SVM classifier over 428 dataset and obtained recognition accuracy of 95.3% for zone identification and 74.8 % for character identification. Ashutosh et al. [8], have performed Gradient based two types of feature extraction methods on offline handwritten Gurmukhi character and Numerals and trained SVM classifier with RBF kernel over 7000 characters and 2000 numerals sample binary images and obtained an accuracy for character as 97.38 % and for numerals as 99.65 %. Akm et al. [9] have compared the result of two classifiers i.e Multilayer perceptron classifier and Convolutional neural network classifier when trained over 3000 images of offline handwritten Arabic numeral and got recognition accuracy 97.4%. S.Acharya et al. [10] have trained the convolutional neural network model to classify the 92,000 images of devanagari character in 46 different class labels and got recognition accuracy of 98.47%. T.K.das et al.[11] have trained the neural network model over capital multi handwritten english alphabets and predicted and displayed the unseen image results as recognized alphabets. Saha et al. [12], trained the convolutional neural network using divide and merge strategy over 166,105 isolated images of offline handwritten Bangla characters and numerals and got an accuracy of 97.21 %.

Gurmukhi Script and Dataset

India has 29 states and each state has its own local language which is used by the people of that state. Each language has some barriers to read and write that language, how the particular language will be written and read is mentioned in a Script of that language. Gurmukhi is a script used to read and write the Punjabi language. Punjabi language is the local language of Punjab state situated in northern part of India. Local language has a major impact on the state work. All major tasks like voter id form, driving licence form etc are available in local language of particular state for the ease of rural areas of a state. The numeric set of Punjabi language is as shown below

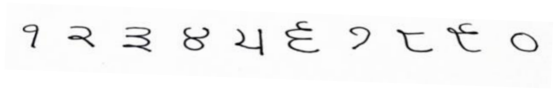


Fig1 Numeric Set of Punjabi Language

Dataset for the experiment of proposed work has been collected manually. Offline handwritten Gurmukhi Digit dataset consist of 13,199 images of 0-9 gurmukhi digits written by 197 different people of Punjab who belong to different educational background, professions and different age group. Each person wrote 70, 0-9 gurmukhi digits on a plain A4 size sheet with black permanent marker, tip size 0.8.

III. PROPOSED RECOGNITION SYSTEM

In our proposed approach of isolated handwritten gurmukhi digit recognition, two main stages have played the main role. Preprocessing and Classifier. As we are utilizing deep neural network model as classifier, it has one primary commitment in pattern recognition that properly designed convolutional neural network model can learn complex mappings over large amount of data without requiring hand-crafted feature extractors. As we have collected the dataset manually, after scanning of all the sheets, these were saved in the jpg format in one directory. Preprocessing is crucial for the effective result.

Following is the stage wise descriptions of the experiment:

4.1 Preprocessing

Data preprocessing is a preliminary step which refines the input data. It consists of series of operations performed on scanned images and transforms the data into form which can be easily and effectively processed by the computer. As deep learning does not require feature extraction, so preprocessing becomes essential stage to contribute in efficient result. The main objective is to remove the distortion, and focus on where our exact information is located and normalize the input to uniform size for fast processing. Following are the steps of algorithm to crop the 70 digits from a single sheet and normalize them using python and its libraries opencv and numpy:

- Read the first image from the directory where all scanned sheet were saved as jpg format.

- Apply threshold and invert the image into binary image. Threshold is a method to convert the gray scale image in to binary image.
- Apply Gaussian blur or smoothing, to reduce the noise.
- To detect the numbers, define bounding rectangle using contour coordinates.
- To find contour, apply findcontour, an inbuilt function and this will find all the coordinates.
- Crop the part which has rectangle and resize the image to 32x32 sizes and save it in another directory.
- Apply other morphological function to enhance the image quality like erosion, dilation and label the dataset manually in separate directories.

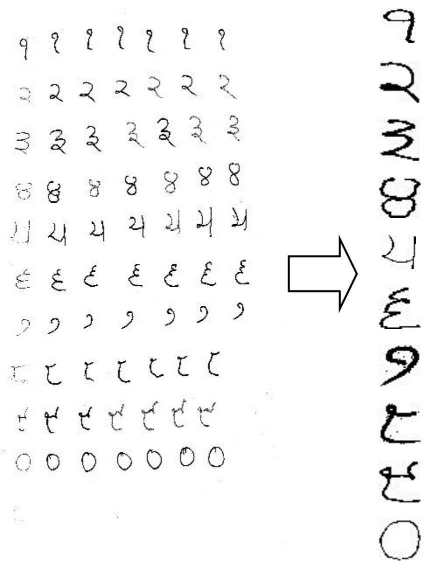


Fig.2. Pre-processing transformation samples

4.2 Convolutional Neural Network Classifier

Classification is one of the method of supervised learning to classify the categorical output. Classification algorithm consist of output labels which is to be predicted from a given set of inputs. Earlier artificial neural network model were shallow in nature due to less processing power and memory. But deep neural network models are giving supreme accuracy results when trained with huge amount of data. It has specified model which is used for the classification for visionary applications i.e. convolutional neural network. Deep neural network are the traditional neural network with more number of layers. Convolutional neural network, a model of deep learning for computer vision application is consist of two main layers which can be used as many times followed by the traditional neural network layers [1]. The description of layers is as follows:

- Convolutional Layer: This layer applies convolutional function on the image with the help of filter or kernel. Convolutional function helps in edge detection. Filter or kernel can be understood as part or grid of image to be processed one by one by sliding each time to fetch the important information. After sliding the filter over all the locations of an image, we obtained a feature map, which will the input to the next layer. The filter or kernel is consisting of numbers called weights and it has same number of channels as in input. Channel in image represents the number of colors like there are 3 channels in RGB image and 1 channel in gray scale image. The size of filter and number of filter to be used to process the input is a hyper parameter i.e. trial and error to find the best value. A relu (rectified linear unit) activation function will be used to get the output of this layer. The model will make the parameters learn or train in this layer. Learnable parameters can be calculated with the help of number of filters and size of filters used. One can use this layer as many times, but it will keep on shrinking the image size. The size of the image can be preserved by either using the padding parameter or it can be preserved by using the more number of filter but simultaneously processing will become very slow if you are using more than three convolutional layers and training model on CPU.
- Pooling Layer: In this layer, model will not learn any parameters. The output obtained from previous layer stored as feature map will

now be the input to this layer. This layer can be applied as Max Pooling or Average Pooling. This layer is used to decrease the dimension of data to reduce the computational cost and provides basic translation invariance to the internal representation. One has to define the pool size or grid size to implement either of two Max Pooling or Average Pooling. Max Pooling fetches the maximum value out of the grid. Average Pooling fetches the average value of all the values present in the grid.

- Fully connected Layer or Dense Layer (traditional neural network hidden layer): Before using this layer, one has to convert the data into one dimensional array using a Flatten function, which is the inbuilt function available in keras. The output as feature map obtained from previous layer is input to this layer. Generally fully connected layers are called dense layer or traditional neural network hidden layer because each neuron in one layer is connected to each neuron in the other layer. The model will learn the final round of parameters in this layer. The number of neurons in this layer will be a hyper-parameter. The output of this layer will be drawn with the help of relu activation function. This layer is generally followed by the output layer. Output layer is also a fully connected layer but it has same number of neurons as class labels and softmax activation function is used to get the probability of each class label.

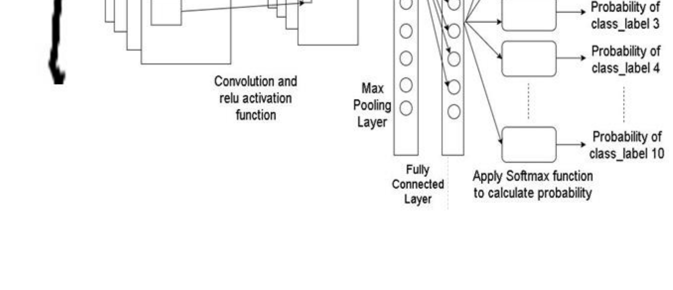

Convolution and relu activation function
Max Pooling Layer
Fully Connected Layer
Apply Softmax function to calculate probability
Probability of class_label 1
Probability of class_label 2
Probability of class_label 3
Probability of class_label 4
Probability of class_label 10

Fig. 3. Convolutional Neural Network Architecture

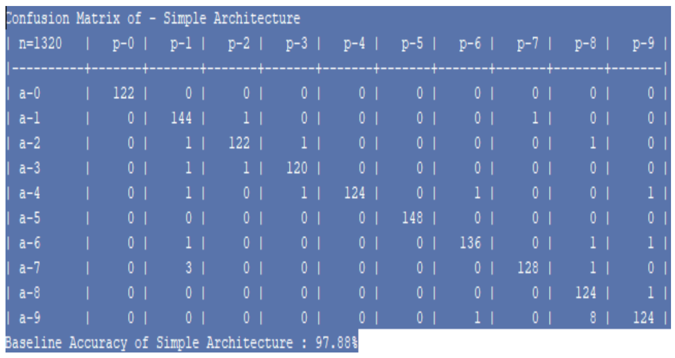
IV. RESULT AND ANALYSIS

This experiment is implemented on a 64-bit intel core i5 (7th Generation) ,2.70 Ghz having 8 GB RAM. The IDE used for Python (3.6 v) programming is Pycharm. Pycharm is very easy to use tool. One can create different interpreters for different projects. Packages can be easily installed with a single click. After the installation of required packages like numpy, opencv, keras, tensorflow. Odd size filter 3x3 , 5x5 were used in different convolutional layers. And the standard pool size 2x2 was used in MaxPooling layer. Rectified Linear Unit (reLu) and Hyperbolic Tangent(tanh) are two activation functions used to get the output of convolutional layer. Softmax function is used in the output dense layer to calculate the probabilities of each class label i.e 0 to 9 in our case. The model was trained over 10559 images and validate on 1320 sample with different values of epochs and batch size. The project was implemented with three different architectures of Sequential Convolutional neural network:

CNN1- Single Convolutional layer followed by single max pooling is used in this. Rectified Linear Unit (reLu) was used as activation function. The loss was calculated using categorical cross entropy function and optimize using adam algorithm. The sequence of layers used are as follows:

Conv2D ➔ MaxPooling2D ➔ Dropout ➔ Flatten ➔ Dense ➔ Output

Total Learnable Parameters was 347,138 in this case. Machine took almost 97 seconds to train and validate and got accuracy of 97.88 %



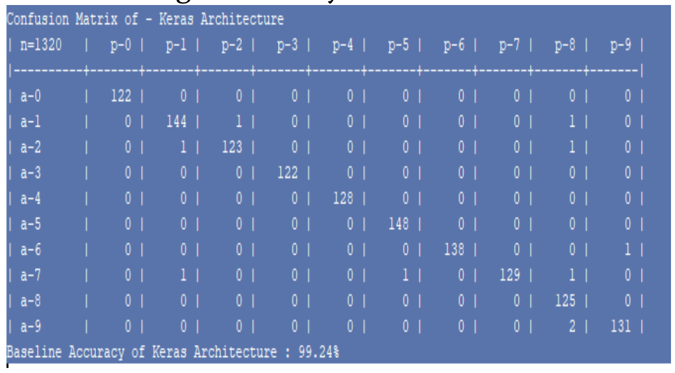
```
Confusion Matrix of - Simple Architecture
| n=1320   |  p-0 |  p-1 |  p-2 |  p-3 |  p-4 |  p-5 |  p-6 |  p-7 |  p-8 |  p-9 |
|----------+------+------+------+------+------+------+------+------+------+------|
| a-0      |  122 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
| a-1      |    0 |  144 |    1 |    0 |    0 |    0 |    0 |    1 |    0 |    0 |
| a-2      |    0 |    1 |  122 |    1 |    0 |    0 |    0 |    0 |    1 |    0 |
| a-3      |    0 |    1 |    1 |  120 |    0 |    0 |    0 |    0 |    0 |    0 |
| a-4      |    0 |    1 |    0 |    1 |  124 |    0 |    1 |    0 |    0 |    1 |
| a-5      |    0 |    0 |    0 |    0 |    0 |  148 |    0 |    0 |    0 |    0 |
| a-6      |    0 |    1 |    0 |    0 |    0 |    0 |  136 |    0 |    1 |    1 |
| a-7      |    0 |    3 |    0 |    0 |    0 |    0 |    0 |  128 |    1 |    0 |
| a-8      |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |  124 |    1 |
| a-9      |    0 |    0 |    0 |    0 |    0 |    0 |    1 |    0 |    8 |  124 |
Baseline Accuracy of Simple Architecture : 97.88%
```

Fig.4. Confusion Matrix for Single layer Architecture

CNN2- Two Convolutional layers followed by single max pooling layer are used in this. Rectified Linear Unit (reLu) was used as activation function. The loss was calculated using categorical cross entropy function and optimize using adadelta algorithm. The sequence of layers used as follows :

Conv2D→ Conv2D→ MaxPooling2D → Dropout→ Flatten → Dense → Dropout→ Output

Total Learnable Parameters was 1,625,866 in this case. Machine took almost 749 sec to train and validate and got accuracy of 99.24%

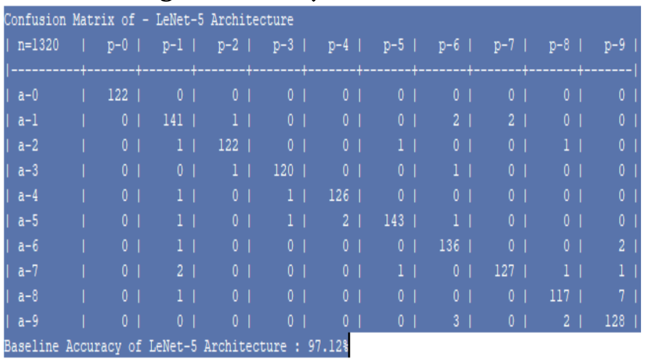


```
Confusion Matrix of - Keras Architecture
| n=1320   |  p-0 |  p-1 |  p-2 |  p-3 |  p-4 |  p-5 |  p-6 |  p-7 |  p-8 |  p-9 |
|----------+------+------+------+------+------+------+------+------+------+------|
| a-0      |  122 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
| a-1      |    0 |  144 |    1 |    0 |    0 |    0 |    0 |    0 |    1 |    0 |
| a-2      |    0 |    1 |  123 |    0 |    0 |    0 |    0 |    0 |    1 |    0 |
| a-3      |    0 |    0 |    0 |  122 |    0 |    0 |    0 |    0 |    0 |    0 |
| a-4      |    0 |    0 |    0 |    0 |  128 |    0 |    0 |    0 |    0 |    0 |
| a-5      |    0 |    0 |    0 |    0 |    0 |  148 |    0 |    0 |    0 |    0 |
| a-6      |    0 |    0 |    0 |    0 |    0 |    0 |  138 |    0 |    0 |    1 |
| a-7      |    0 |    1 |    0 |    0 |    0 |    1 |    0 |  129 |    1 |    0 |
| a-8      |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |  125 |    0 |
| a-9      |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    2 |  131 |
Baseline Accuracy of Keras Architecture : 99.24%
```

Fig.5. Confusion Matrix for Two layer Architecture

CNN3- Three Convolutional layers are used in this. Hyperbolic Tangent (tanh) was used as activation function. The loss was calculated using categorical cross entropy function and optimize using Stochastic Gradient Descent (SGD) algorithm. The sequence of layers used as follows :

Total Learnable Parameters was 61706 in this case. Machine took almost 114 sec to train and validate and got accuracy of 97.12 %



```
Confusion Matrix of - LeNet-5 Architecture
| n=1320   |  p-0 |  p-1 |  p-2 |  p-3 |  p-4 |  p-5 |  p-6 |  p-7 |  p-8 |  p-9 |
|----------+------+------+------+------+------+------+------+------+------+------|
| a-0      |  122 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
| a-1      |    0 |  141 |    1 |    0 |    0 |    0 |    2 |    2 |    0 |    0 |
| a-2      |    0 |    1 |  122 |    0 |    0 |    1 |    0 |    0 |    1 |    0 |
| a-3      |    0 |    0 |    1 |  120 |    0 |    0 |    1 |    0 |    0 |    0 |
| a-4      |    0 |    1 |    0 |    1 |  126 |    0 |    0 |    0 |    0 |    0 |
| a-5      |    0 |    1 |    0 |    1 |    2 |  143 |    1 |    0 |    0 |    0 |
| a-6      |    0 |    1 |    0 |    0 |    0 |    0 |  136 |    0 |    0 |    2 |
| a-7      |    0 |    2 |    0 |    0 |    0 |    1 |    0 |  127 |    1 |    1 |
| a-8      |    0 |    1 |    0 |    0 |    0 |    0 |    0 |    0 |  117 |    7 |
| a-9      |    0 |    0 |    0 |    0 |    0 |    0 |    3 |    0 |    2 |  128 |
Baseline Accuracy of LeNet-5 Architecture : 97.12%
```

Fig.6. Confusion Matrix for three layer Architecture

If we look closely at the architecture of all three models, we can conclude that accuracy of the model depends on number of parameters like batch size, number of epochs used to train the model. When model is getting trained, it is actually learning the parameters in each layer, more the number of parameters learnt by model, better will be the accuracy. Learnable parameters depend upon the number of filters used and size of filter used. Sometimes this leads to overfitting issue. So it is better to use the various regularization techniques to truncate some data after the convolutional layer and before the dense layer to avoid overfitting. In two convolutional layer architecture, the learnable parameters are more so accuracy is better. In case of three convolutional layer architecture, the learnable parameters are very less that is why average accuracy is obtained. So we can also say that number of layer used has less impact on accuracy. Accuracy is mostly effected by the setting of hyper-parameters. The table shows the different values of epochs and batch size and corresponding accuracy achieved.

epochs	batch size	CNN1	CNN2	CNN3
20	100	97.73%	99.24%	95.53%
20	200	97.65%	99.02%	93.03%

30	100	97.58%	98.79%	96.67%
30	200	97.70%	98.86%	95.08%
10	50	97.35%	98.94%	94.92%
20	50	97.80	98.79%	97.12%

The comparison of all the three different architecture of convolutional neural network model is shown below using graph in Fig.7 in which epoch and batch size are taken along x-axis and accuracy obtained are taken along y axis.

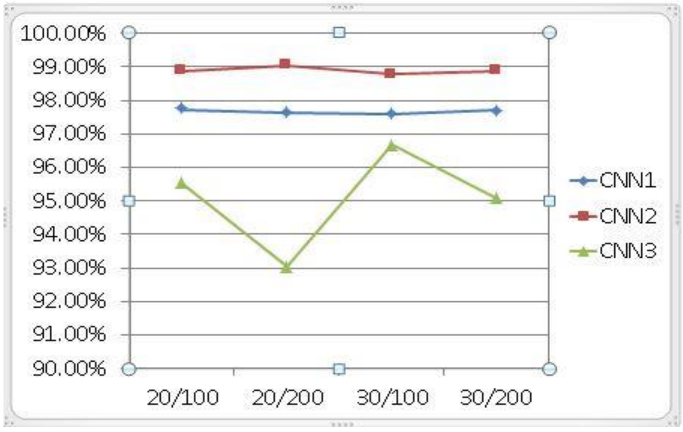


Fig.7. Comparison of three different Architecture of CNN

V. CONCLUSION

Our experiment concludes that the two layer convolutional neural network when trained over handwritten gurmukhi digits, gives excellent result. The accuracy can be further improved by increasing the number of epochs and batch size. When the model was saved and it has predicted almost 99% correct images labels. In future, the work can be extended to train the model for handwritten multi digits like pin code prediction or to predict the combination of handwritten digits and characters like for UID number for example adhar card.

VI. REFERENCES

[1]. Alex KrizhevskyIlya SutskeverGeoffrey E. Hinton "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012

[2]. Kumar, Munish & K. Jindal, M & K. Sharma, R. (2017). A Novel Technique for Line Segmentation in Offline Handwritten Gurmukhi Script Documents. National Academy Science Letters. 40. 273-277. 10.1007/s40009-017-0558-1.

[3]. Pal, Umapada & Chaudhuri, Bidyut. (2004). Chaudhuri, B.B.: Indian script character recognition-A survey. Pattern Recognition 37, 1887-1899. Pattern Recognition. 37. 1887-1899. 10.1016/j.patcog.2004.02.003.

[4]. Siddharth, Kartar & Jangid, Mahesh & Dhir, Renu & Rani, Rajneesh. "Handwritten Gurmukhi Character Recognition Using Statistical and Background Directional Distribution Features", Internat. J. Computer Sci. Eng. (IJCSE). 3. 2332-2345, 2011.

[5]. Singh G. and Lehri S., " Recognition of Handwritten Hindi Characters Using Back-propagation Neural Network", International Journal of Computer Science and Information Technologies, pg. 4892-4895, Vol. 3, Issue 4, 2012.

[6]. Kumar, M., Jindal, M.K. & Sharma, R.K. ," A Novel Hierachical Technique for Offline Handwritten Gurmukhi Character Recognition" Natl. Acad. Sci. Lett. 37: 567. https://doi.org/10.1007/s40009-014-0280-1,2014

[7]. Verma, K. & Sharma, R.K. Sādhanā.," Recognition of Online handwritten Gurmukhi characters based on zone and stroke identification" 42: 701. https://doi.org/10.1007/s12046-017-0632-x , 2017

[8]. Ashutosh Aggarwal , Karamjeet Singh, "Use of Gradient Technique for extracting features from Handwritten Gurmukhi Characters and Numerals",Procedia Computer Science, Volume 46, PP 1716-1723, 2015

[9]. Akm Ashiquzzaman and Abdul Kawsar Tushar, "Handwritten Arabic Numeral Recognition using Deep Learning Neural Network",IEEE International Conference on Imaging, Vision & Pattern Recognition, At University of Dhaka, Dhaka, Bangladesh, 2017

[10]. S. Acharya, A. K. Pant and P. K. Gyawali, "Deep learning based large scale handwritten Devanagari character recognition," 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Kathmandu, pp. 1-6,2015.

[11]. T. K. Das, A. K. Tripathy and A. K. Mishra, "Optical character recognition using artificial neural network," 2017 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 2017, pp. 1-4.doi: 10.1109/ICCCI.2017.8117703

[12]. Saha Sourajit, Nisha Saha, "A Lightning fast approach to classify Bangla Handwritten Characters and Numerals using newly structured Deep Neural Network", Procedia Computer Science, vol. 132, pp. 1760-1770, 2018.

Cite this article as :

Sonia Flora, Divya Ebenezer Nathaniel, "Handwritten Gurmukhi Digit Prediction using Convolutional Neural Network with Keras and Tensorflow", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 6 Issue 2, pp. 777-784, March-April 2019. Available at doi : https://doi.org/10.32628/IJSRSET1962169
Journal URL : https://ijsrset.com/IJSRSET1962169