Handwritten Bangla Digit Recognition

Md Mania Ahmed Joy
Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
2018-1-60-042@std.ewubd.edu

Abdur Rahman Tumzied
Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
2018-1-60-076@std.ewubd.edu

Md Habibur Rahman
Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
2018-1-60-063@std.ewubd.edu

***Abstract*— Nowadays, machine learning is playing an important role in the field of image classification. In our project we have used three different machine learning classifier algorithms, Convolutional Neural Networks, K-Nearest Neighbor and Decision Tree, in order to recognize Bangla handwritten digits. We have used two Python libraries known as Keras and Scikit-learn in our models. The training of the models is optimized using Adam Optimizer. This paper gives an overview of multi-class classification of the dataset images using the three aforementioned models and their performance evaluation. In the categorization of handwritten digits, Convolutional Neural Networks are found to be more accurate than K-Nearest Neighbor and Decision Tree. The core motivation of our work is to make a significant impact in the field of Computer Vision in the context of Bangla digits recognition.**

Keywords—CNN, KNN, Decision Tree, Adam Optimizer, Machine Learning, Image Classification.

I. INTRODUCTION

Humans live has changed drastically with every industrial revolution. We are on the verge of the 4th industrial revolution and our lives are going to ascend to an era of Artificially Intelligent systems. Major strides are being made in the field of Machine Learning. Convolutional Neural Networks (CNNs) have recently been one of the most appealing techniques and have been widely used as a key component in a number of recent successful and difficult machine learning applications, including challenge ImageNet [1]–[6], object identification [1], [7], [8] picture segmentation [9], [10] and face recognition [11]–[13]. Although English handwriting datasets appear to be readily available, and major advances have been achieved for English digit datasets such as CENPARMI, CEDAR, and MNIST, there are few studies done on Bangla digit datasets. The reason for which there has not been much work on Bangla handwritten digit classification is the unavailability of datasets. We managed to obtain a handwritten dataset of Bangla digits that consists of more than 19,000 images. Furthermore, we investigate and demonstrate the application of CNN on this dataset. Apart from applying CNN we also applied Decision tree and KNN to our dataset and found that the Convolutional Neural Network achieves superior results in classifying Bangla handwritten digits..

II. DATASET AND PREPROCESSING

A. Dataset

The dataset that we have used is a sub set of a published dataset titled “BanglaLekha-Isolated” in Mendeley Data [14]. Our dataset consist of samples of 10 Bangla Handwritten digits. For each sample we have almost 2000 images. For our training purpose we have used 12638 images. Our dataset’s sample images and a graphical representation for the number of images in different classes has been depicted below:

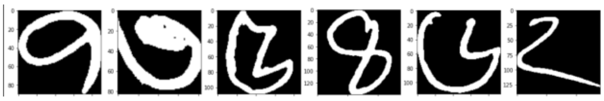

Fig 1: Sample train images

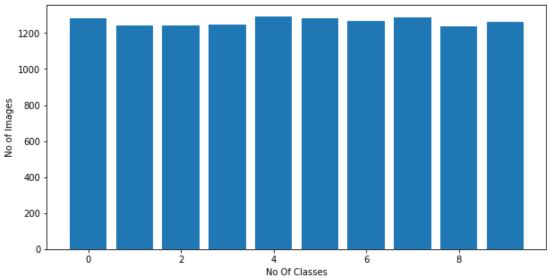

Fig 2: Bar Plot of Classes & Images

B. Data Pre-processing

Before running our algorithm on the dataset, we needed to pre-process the data. The aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing. We converted the images to binary values that is 0 and 1 by applying a threshold. If the pixel values were lower than 170, we set the value to 0 and if it was over 170, we set the value to 1. We have also used histogram equalizer in our pre-processing for setting all the images to a standard intensity.

III. METHODOLOGY

Our core research relies on the use of Convolutional neural network which we implemented using python code. In the dense layer of our model, we have used Relu and softmax as activation function. We used Adam optimizer for compiling the model and categorical cross entropy for Loss Calculation. For comparison of our model’s result, we have trained two more models which are Decision Tree and KNN.

A. *Convolutional Neural Network*

A Convolutional Neural Network (CNN) is a special kind of neural network where there are multiple convolution layers followed by fully linked layers.

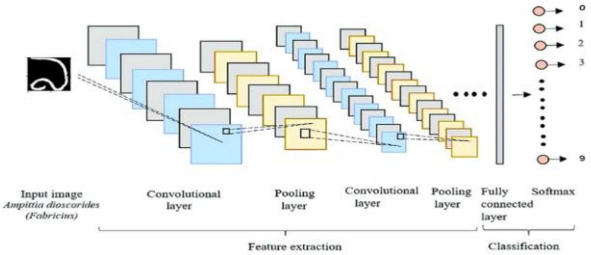

Fig 3: Basic CNN Model architecture

In this illustration, CNN has been used to do class identification, which the network does quite well. Each takes inputs from tiny feature sets in its feature neighborhood. This makes extracting edges and points easy. Finally, the upper levels in the system integrate these network. The feature extraction procedure is carried out using with the aid of convolution filters.

Since CNNs combine the weights of convolution layers during feature extraction with fully connected layers used for classification, they provide greater performance in terms of complexity and memory needs. As a result, image recognition, speech recognition, and video analysis are all typical applications for these networks. Our paper focuses on the classification of Bangla handwritten digit.

1. *Convolution Layer:*

   Convolution is used in these layers to extract various characteristics from the input. Filters are tiny matrices that are used to identify characteristics. Sliding a filter across an image and computing the dot product yields a convolved feature or activation map. A feature output of (N-n+1) x (N-n+1) elements is obtained by convolving a filter of size n x n with an input of size N x N. Each filter is pushed lower, then left to right, until the entire input is covered, starting at the top right corner. We have used a total of 6 convolution layers in our model.

   1. First two convolution layer uses (5x5) kernel size and output 80 feature map, with stride of 2 and padding.
   2. Second two convolution layer uses (3x3) kernel size and output 40 feature map, with stride of 2 and padding.
   3. Last two convolution layer same as second layer but output feature map is 40.

2. *Pooling Layer:*

   The resolution of features is reduced by the pooling layers. It is a technique for moving a window across a 2D window space, with the output being the maximum/minimum value in the window. This is determined by the size of the pooling layer selected by the user. In our model we have used max pooling.

3. *Non-linear layer:*

   In our model we have used two non-linear layers. They are:

   a. *RELU:*

   ReLu is an activation function which stands for rectified linear unit. It is a simple and effective function that aids in the resolution of the vanishing gradients problem in neural networks. It eliminates any negative numbers from the output and ensures that the layer sizes of the input and output are the same. The equation for ReLu is given below:

   $$y = \max(0, x)$$

   b. *Softmax:*

   Softmax is an activation function which is employed in a neural network to normalize the output of the network to a probability distribution over predicted output classes, supported Luce's choice axiom. The formula for softmax is given below:

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

   σ=Softmax
   $\vec{z}$= input vector
   e^{z_i}= standard exponential function for input vector
   K= number of classes in the multi-class classifier
   e^{z_j} = standard exponential function for output vector

4. *Fully Connected layer:*

   These are CNN's final layers, which add together the weights from the preceding layers to arrive at a certain goal result. To give each piece of output a feature, all elements in the preceding levels are involved.

5. *Optimization of weights:*

   In our model we have used Adaptive Moment Estimation optimizer. Adaptive Moment Estimation (Adam) is a deep learning optimization algorithm that replaces stochastic gradient descent. Adam combines the finest features of the AdaGrad and RMSProp methods to provide an optimization solution for noisy situations with sparse gradients. The equation for Adam optimizer is given below:

   $$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}}+\varepsilon}\, \widehat{m_t}$$

   Here,

   $$\widehat{m_t} = \frac{m_t}{1-\beta_1^t}$$

$$\widehat{v}_t = \frac{v_t}{1-\beta_1^t}$$

B. K-Nearest Neighbor (KNN):

We have also used KNN in our system in order to compare our CNN output. The KNN classifier is the simplest straightforward classifier method. The KNN classification system does not learn anything. The distance between feature vectors is used in this technique. This algorithm's training procedure consists only of gathering feature vectors as well as labels from training pictures. During the testing or predicting the unlabeled point, it simply allocated to the label of its *k* closest neighbors.

Typically, objects are classified by major vote depending on the labels of their *k* closest neighbors. If *k*=1, the object is typically labeled according to the class of the object that is closest to it. When just two classes are present, k must be an odd number. We applied the most popular distance function in KNN, which is Euclidean distance, after converting each picture to a fixed-length vector of real values. Euclidean distance equation given below:

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

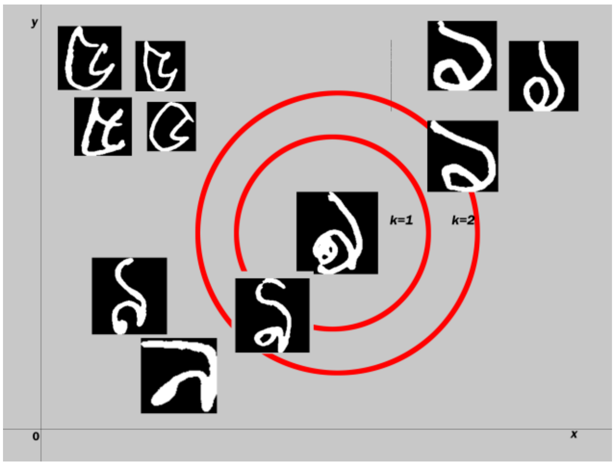

Fig.4: KNN classification, where *k is* 1 and 2.

C. Decision Tree

We have also used Decision tree to compare our CNN output result. Decision tree is one of the most basic form of classifier method. In Decision Tree, internal decision-making logic is shared, which is not available in machine learning algorithms like Neural Network. Compared to the neural network approach, it takes less time to coach. When compared to the neural network approach, it takes less time to train. The number of records and characteristics in the provided data determine the temporal complexity of decision trees. The decision tree is a non-parametric or distribution-free approach that predicts the value of a target variable by learning simple decision rules inferred from the data features. With good accuracy, decision trees can handle high-dimensional data.

In our project we have used "entropy" for the information gain as supported criteria. As we have used entropy we want to debate "Information Gain". Information gain refers to the impurity within the dataset. supported supplied attribute values, information gain computes the difference between entropy before split and average entropy after split of the dataset. The equation of entropy of a random variable X is given below:

$$\mathrm{H(X)} =- \sum_{i=1}^{n} \mathrm{p(x = i)} \log_2 P(X = i)$$

The equation for Information gain is:

$$\mathrm{I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)}$$

IV. Result Analysis

We have applied KNN, Decision tree and CNN to our dataset. The results are analyzed below:

A. KNN

For KNN, the result shows for value *k* equal 1 to 15 have been shown below:

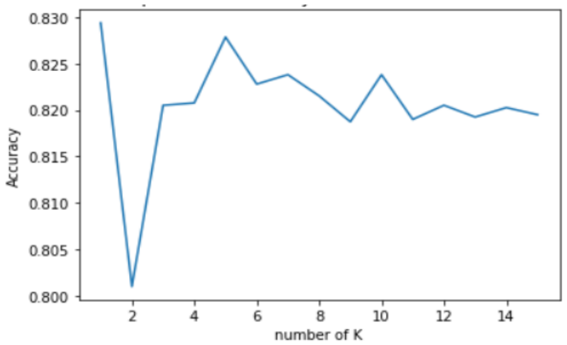

Fig. 5: KNN prediction accuracy base of different k values

The highest accuracy 83% is achieved at *k* = 1.
The confusion matrix is shown below using heat map.

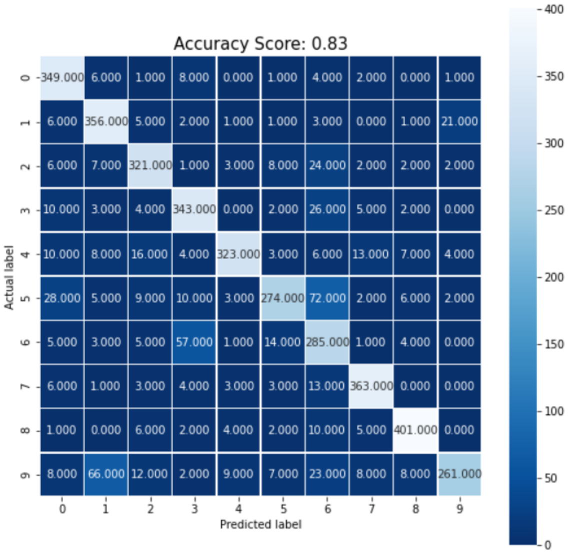

Fig. 6: KNN Confusion Matrix

Here, the digits 0,1,2,3,4,7,8 has been correctly classified over 300 times. The digits 9, 6 and 5 has been correctly classified over 200 times. There have also been a few miss classifications. For example: The digit 9 has been miss classified as 1 in 66 occasions.

B. *Decision Tree*

Applying Decision Tree gives an accuracy of 82.9% . The confusion matrix is shown below using heat map:

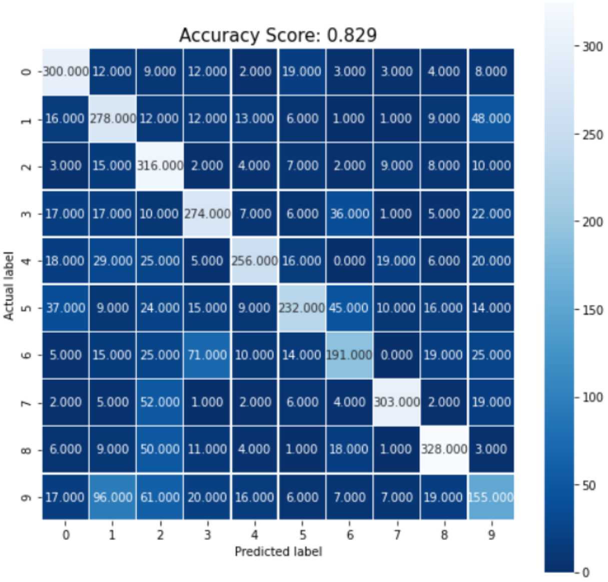

Fig. 7: Decision Tree Confusion matrix.

Here the digit 0, 2, 7 and 8 have been correctly classified more than 300 times. The digit 6 and 9 has been correctly classified 191 and 155 times. All the other digits have been correctly classified more than 200 times each.

The highest instance of misclassification is the digit 9 and it has been highly misclassify as 1.

C. *Convolutional Neural Network:*

Convolutional Neural Network has yielded the highest accuracy. We have found the accuracy to be 96% when we applied it to our dataset. The heat map for this is shown below:

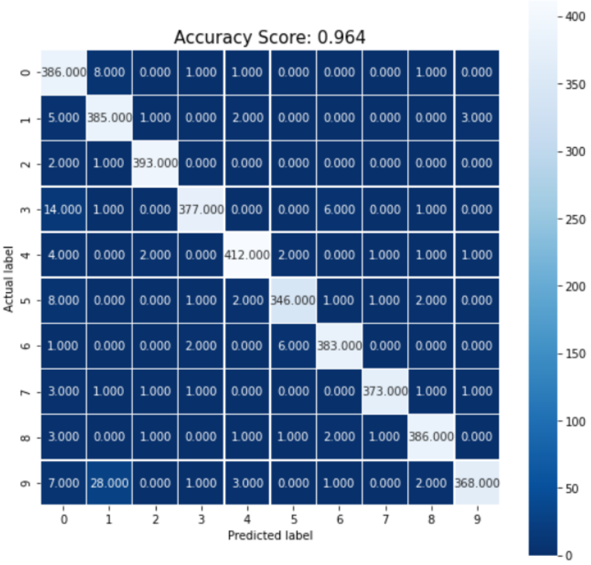

Fig.8: CNN Confusion Matrix

Here 4 has been classified correctly 412 times. The digits 0,1,2,3,6,7,8 and 9 have been correctly classified more than 350 times. The digit 5 has also been correctly classified more than 345 times.

The incorrect classifications are also very low here. The highest instance of misclassification is the digit 9. It has been misclassified as 1 in 28 occasions.

Now let us observe the overall results:

TABLE I. OVERALL RESULT

Classifier Model Type	Accuracy
Decision Tree	**82.9%**
KNN	**83%**
CNN	**96.4%**

V. CONCLUSION

We have successfully implemented CNN, KNN and Decision tree to our Bangla Handwritten digit dataset. Convolutional neural networks outperform other types of classifier algorithms when it comes to classifying or recognizing pictures. `We have demonstrated the efficacy of Convolutional Neural Network in classifying handwritten Bangla digits in this project. This can be put to various uses in our country as the lion’s share of government documentations are still done in Bangla.

REFERENCES

[1] H. M., H. A., and N. E., “Robust Convolutional Neural Networks for Image Recognition,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 11, 2015, doi: 10.14569/ijacsa.2015.061115.

[2] F. Lauer, C. Y. Suen, and G. Bloch, “A trainable feature extractor for handwritten digit recognition,” *Pattern Recognit.*, vol. 40, no. 6, 2007, doi: 10.1016/j.patcog.2006.10.011.

[3] C. Y. Lee, S. Xie, P. W. Gallagher, Z. Zhang, and Z. Tu, “Deeply-supervised nets,” in *Journal of Machine Learning Research*, 2015, vol. 38.

[4] M. A. Fischler and R. A. Elschlager, “The Representation and Matching of Pictorial Structures Representation,” *IEEE Trans. Comput.*, vol. C–22, no. 1, 1973, doi: 10.1109/T-C.1973.223602.

[5] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?,” 2009, doi: 10.1109/ICCV.2009.5459469.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, 2017, doi: 10.1145/3065386.

[7] and J. S. Kaiming He, Xiangyu Zhang, Shaoqing Ren, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, 2015.

[8] X. Wang, M. Yang, S. Zhu, and Y. Lin, “Regionlets for Generic Object Detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, 2015, doi: 10.1109/TPAMI.2015.2389830.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” 2014, doi: 10.1109/CVPR.2014.81.

[10] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, “Indoor semantic segmentation using depth information,” 2013.

[11] M. Wang and W. Deng, “Deep face recognition: A survey,” *Neurocomputing*, vol. 429, 2021, doi: 10.1016/j.neucom.2020.10.081.

[12] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, vol. 2, doi: 10.1109/CVPR.2006.100.

[13] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, 2015, doi: 10.1007/s11263-015-0816-y.

[14] Mohammed, Nabeel; Momen, Sifat; Abedin, Anowarul; Biswas, Mithun; Islam, Rafiqul; Shom, Gautam; Shopon, Md (2017), “BanglaLekha-Isolated”, Mendeley Data, V2, doi: 10.17632/hf6sf8zrkc.2.