



# SVM, KNN, random forest, and neural network based handwritten nepali barnamala recognition

Bal Krishna Nyaupane<sup>a</sup>, Rupesh Kumar Sah<sup>a</sup> and Kiran Chandra Dahal<sup>b,\*</sup>

<sup>a</sup>Department of Electronics and Computer Engineering, Paschimanchal Campus, Institute of Engineering, Tribhuvan University, Pokhara, Nepal

<sup>b</sup>Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering, Tribhuvan University, Thapathali, Kathmandu, Nepal

## ARTICLE INFO

### Article history:

Received 05 Jul 2021  
Received in revised form  
04 Oct 2021  
Accepted 27 Oct 2021

### Keywords:

Accuracy  
Recognition  
Handwritten  
Neural network  
Nepali barnamala

## Abstract

Nepali Barnamala consists of 36 consonants, 12 vowels, and 10 Nepali digits. Among them, this paper uses the 36 consonants and 10 Nepali digits for the recognition using machine learning-based algorithm mainly: Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), and several architectures of neural networks. In this paper, different kernel tricks of SVM with different regularization parameters have been used to train the model and have compared their accuracy and F1-score. In KNN, accuracy and F1-score are compared with different values of K and distance metrics. In Neural Networks, training accuracy, training loss, validation accuracy, and validation loss are compared with the different numbers of hidden layers' regularization parameters and learning rate. Different hyperparameters of random forest are changed and compared to their corresponding result. This paper uses the Kaggle dataset of school students' Nepali handwritten characters. The dataset is CSV format with 78,200 rows for forty-six different classes with 1024 (32\*32 image size) columns plus one column for a label of characters for training and 13,800 rows for testing. For handwritten Nepali Barnamala recognition, the best average accuracy is 93.51% of neural networks with four hidden layers.


©JIEE Thapathali Campus, IOE, TU. All rights reserved

## 1. Introduction

Handwriting recognition has been studied for many years by researchers in the field of machine learning and pattern recognition and their many approaches are proposed. Handwriting recognition is a fairly complex problem, and even now there is no single approach that solves it efficiently and completely in every context [1]. The writing pattern of the Nepali language is diverse in terms of syntax and semantic. Devanagari script is the main domain of the Nepali language which reading and writing fashion is left to right. Handwritten characters or word recognition is the process to convert the image of the handwritten characters to the respective character text. This conversion process is one of the major

tasks in the field of machine learning and deep learning. Handwritten Nepali Barnamala recognition has many applications in the government office records keeping and analysis system, ancient document digitization system, bank check identification and verification system, etc. Nepali Handwritten Barnamala recognition is the major task of digital technology to recognize handwritten characters as input from different sources such as handwritten papers or images; printed documents or images; or input from other digital means. In the handwriting recognition process, a handwritten image containing Nepali Barnamala must be provided and must perform the preprocessing steps such as grayscale conversion, image filtering, binarization and thresholding of images [2]. And then the image either undergoes segmentation or feature extraction process. The extracted features feed into the proposed model to train the model for handwritten barnamala recognition and then finally

\*Corresponding author:

 [dahalkc@ioe.edu.np](mailto:dahalkc@ioe.edu.np) (K.C. Dahal)

testing is conducted to verify the result.

Nepali, Sanskrit, Hindi, and other several languages are derived from the Devanagari script which is one of the oldest ancient scripts. Nepali language is very common and widely spoken in Nepal and it is the official language of Nepal. Even though a lot of effort has been applied for handwritten character recognition for English and other languages, there are very few papers have researched Nepali handwritten barnamala recognition [3]. Thus, to fulfill the need for research for an efficient algorithm to recognize Nepali handwritten barnamala is considered under this paper.

However, many academicians and researchers are researching the field of Handwritten character recognition, no researchers have been achieved a hundred percent accuracy. I Khandokar et al. in [4] researched handwritten digits recognition for the NIST (National Institute of Standards and Technology) dataset using a Convolutional Neural Network. In this approach, the researcher had employed two convolutional layers with max-pooling and cascading with two fully connected layers. The maximum accuracy achieved is 92.91% when the training images are 1000. Narayan and R. Muthalagu et al. [5] implemented a Convolutional Neural Network (CNN) for Image character recognition. The Chars74K dataset was used for training and test in the ratio of 80:20. In this research, the researcher had employed two convolutional layers with max-pooling and cascading with one fully connected layer. The accuracy of their test is reported as 97.59%. In [6] Samay Pashine et al. used Machine learning algorithm-SVM, Multi-Layer Perceptron (MLP), and CNN for digit recognition on the MNIST dataset (Modified National Institute of Standards and Technology database). The groups of researchers obtained an accuracy of three models as SVM accuracy: 93.77%, MLP accuracy 98.85%, and CNN accuracy 99.31%.

## 2. Methodology

### 2.1. Dataset

Nepali Barnamala has 36 Baynjanbarnas, 12 Swarbar-nas, and 10 digits. In this research, only 36 Baynjanbarnas and 10 digits have been used for training and test purpose. Kaggle dataset "Nepali Barnamala Handwritten CSV format Dataset" is used for training and test purposes. This dataset contains 36 consonants characters "Ka" to "Gya" and ten-digit characters from zero to nine. It has two CSV files - training and test dataset. The CSV has total 1025 columns:  $32 \times 32$  [image size] = 1024 columns for image information and one for label of the image. The label for "Ka" is 0 and "Kha" is 1 and so on. The label of digit zero is 36 and digit one is 37

and so on. The training CSV dataset, screenshot shown in Figure 1, contains  $1700 \times 46 = 78200$  rows and 1025 columns. Similarly, the test CSV file contains  $300 \times 46 = 13800$  rows and 1025 columns. All characters of the dataset are as follows:

‘क’, ‘ख’, ‘ग’, ‘घ’, ‘ङ’, ‘च’, ‘छ’, ‘ज’, ‘झ’, ‘ञ’, ‘ट’, ‘ठ’, ‘ड’, ‘ढ’, ‘ण’, ‘त’, ‘थ’, ‘द’, ‘ध’, ‘न’, ‘प’, ‘फ’, ‘ब’, ‘भ’, ‘म’, ‘य’, ‘र’, ‘ल’, ‘व’, ‘श’, ‘ष’, ‘स’, ‘ह’, ‘क्ष’, ‘त्र’, ‘ज्ञ’, ‘०’, ‘१’, ‘२’, ‘३’, ‘४’, ‘५’, ‘६’, ‘७’, ‘८’, ‘९’

	1'1	1'2	1'3	1'4	1'5	1'6	1'7	1'8	1'9	1'10	...	32'24	32'25	32'26	32'27	32'28	32'29	32'30	32'31	32'32	Label	
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0	0
1702	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0	1
17000	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0	10
34000	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0	20
60000	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0	35
66000	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0	40

Figure 1: CSV data of Nepali Handwritten Characters

The image view of some of the characters from the CSV training dataset is shown in the Figure 2.

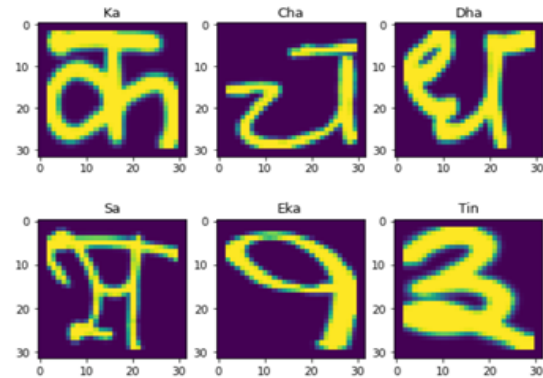


Figure 2: Image view of the CSV data of Nepali handwritten characters

### 2.2. Training algorithms

#### 2.2.1. Support Vector Machine (SVM)

SVM is a tutor-based algorithm called a supervised algorithm. SVM equally focuses on classification-based problems and regression-based problems; however, it has more application for classification-based problems. Both linear and nonlinear data can feed into the SVM model for classification and regression [7]. The SVM always finds a hyperplane to separate one class from another. In SVM, the number of features or attributes of the data plays a key role. If data has a  $d$  number of features, then it is called  $d$ -dimensional space. If  $d$  is two it is called two-dimensional space and the straight

line works as a hyperplane for classification. If  $d$  is three, the hyperplane is a plane to separate tuples of one class from another.

The hyperplane should be as far as possible from any data points. If we have made a small error in the location of the hyperplane, this gives us the least chance of causing a misclassification. Linear SVM is the simplest kind of SVM with a maximum margin [7]. The maximum margin is determined by the support vectors, which are the data points closest to the separating hyperplane. Therefore, SVM is much less prone to overfitting, robust to outliers, and has a strong generalization ability than other methods. SVM has many applications in machine learning and computer vision fields such as image classification, handwriting character recognition, and classification, medicine and bio-sequence analysis, and stock market prediction, etc. The SVM gained popularity as a result of its success in handwritten digit recognition, with a test error rate of 1.1 percent. This is the same as the error rates of LeNet 4, a well-built neural network.

A kernel is nothing more than a comparison of data points. Kernel trick is used for non-linearly separable data. A kernel trick is a process to map the non-linearly separable data into higher dimensional space. For example, there are two-dimensional non-linearly separable data that by using kernel trick this data can be transformed into three-dimensional space where the data becomes linearly separable. As a result, it can be mapped any non-linearly separable data of any dimension to a higher dimension and then make linearly separable. In  $d$ -dimensional feature space, a kernel function is defined as a function that corresponds to the dot product of two feature vectors. The kernel function merely needs to compute and does not need to do it explicitly in high-dimensional space [7]. Finally, the kernel function determines how similar the points in the newly transformed feature space are.

There are several kernel functions, this paper only used three of them:

- **Linear Kernel**

It is a very simple and easiest type of kernel, and it's usually one-dimensional. When there are several features, the linear kernel works as the best kernel function. The text classification problems have several features and they are linearly separable, so the linear kernel is usually preferred for text recognition or classification. In comparison to other kernel functions, the linear kernel is very fast. Linear kernel formula is given in Equation 1

$$K(x_i, x_j) = \text{SUM}(x_i, x_j) \quad (1)$$

Here  $x_i, x_j$  represents the data objects for classification and  $K$  is the Kernel function.

- **Radial Basis Function Kernel (RBF)**

The default kernel trick is RBF for the SVM. The RBF calculates the similarity in the transformed feature space between an exponentially decaying function of the distance between the support vectors and the original input space.

$$K(x_i, x_j) = e^{-\gamma(x_i - x_j)^2} \quad (2)$$

$$0 \leq \gamma \leq 1$$

Here  $x_i, x_j$  represents the data objects for classification. The most common value of gamma is 0.1.

- **Polynomial Kernel**

The 'degree' parameter of the polynomial kernel influences the model's complexity and computing cost of the transformation. The default degree is 3 for SVM.

$$K(x_i, x_j) = (1 + x_i \cdot x_j)^d \quad (3)$$

Here  $x_i, x_j$  represents the data tuples trying to classify and  $d$  denotes the degree of a polynomial.

### 2.2.2. Random Forest (RF)

The Random Forest is a supervised Machine Learning technique that uses decision trees to do classification, regression, and other tasks. Random forest is an ensemble method that employs decision trees as its foundation classifiers and manipulates its input features. Manipulation of input features can be used to create an ensemble of classifiers. Each training set is made up of a subset of input features in this method. The subset might be picked at random or based on domain experts' recommendations. Random forest is an ensemble method intended primarily for decision tree classifiers [8]. It combines the predictions of numerous decision trees, each of which is based on the values of a separate collection of random vectors.

Suppose there are  $N$  numbers of training samples and  $d$  numbers of features in the data. The steps taken to implement a Random Forest are as follows [8]:

- Select the  $N1$  number of samples from the training data set randomly with replacement.
- Select  $d1$  features randomly and whichever feature gives the best split is used to split the node.
- At the next node, choose other  $d1$  features randomly from all features and do the same.

- iv The tree is grown to the largest.
- v Repeat steps P times and prediction is based on the majority voting. Whichever class label gets the majority vote, random forest predicts that class label.

### 2.2.3. K-Nearest Neighbor (KNN)

In Machine Learning, KNN is one of the most basic supervised learning algorithms. Pattern recognition, data mining, and intrusion detection are all areas where they can be used. KNN classifiers work by comparing a given test tuple to training tuples that are similar to it. There are several attributes to characterize the training tuples. Each tuple represents an n-dimensional feature space [9]. A KNN classifier explores the pattern for the K training tuples that are closest to the unknown tuple

when given an unknown tuple. These K training tuples are the unknown tuple's K "nearest neighbors."

The value of K and the distance matrices are two crucial elements to consider while using the KNN. The value of K is determined by the model's error rate. To get the right number for K, start with K= 1 and utilize a test set to estimate the classifier's error rate. This technique can be done as many times as necessary by increasing K to accommodate one more neighbor [9]. The K value that yields the lowest error rate can be chosen. The larger the number of training tuples, the higher the value of K will be in general. Minkowski distance is the most extensively used distance matrix for the KNN. It's a combination of the Euclidean and Manhattan distances together. Minkowski distance is defined as in Equation 4

$$d(i,j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \dots + |x_{ip} - x_{jp}|^h} \quad (4)$$

Where h is a real number. If h = 1 it becomes the Manhattan distance and if h = 2 it becomes the Euclidean distance.

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

### 2.2.4. Neural Network (NN)

A neural network is a computing system that consists of several simple, highly interconnected processing components that process data based on their dynamic state responses to external inputs. Neural networks are a great tool for solving a variety of real-world issues. They can learn from their mistakes to enhance their performance and adapt to changing circumstances. They can also deal with incomplete or noisy data, making them particularly useful in circumstances where it is difficult to identify the rules or procedures that lead to a problem's solution.

Where,

$\sigma$  = Softmax

$e^{z_i}$  = Exponential function for input vector

$\vec{Z}$  = Input vector

$e^{z_j}$  = Exponential function for output vector

K = Number of classes

This paper has used the dense neural network. It has focused on five important parameters of dense neural networks such as number of hidden layers, number of neurons in the hidden layer, regularization parameter, activation function, and learning rate.

### 2.3.2. ReLU activation function

ReLU stands for Rectified Linear Unit and it is a non-linear activation function. It is the most popular activation function for the hidden layer. ReLU has simple and very few mathematical operations with comparison to tanh and sigmoid activation functions. The ReLU function is defined as in Equation 6 and represented as in Figure 3.

## 2.3. Activation function

### 2.3.1. Softmax activation function

SoftMax expands the concept into a multi-class environment. In a multi-class problem, SoftMax assigns decimal probabilities to each class. It's the activation function that isn't linear. The SoftMax function is best employed at the classifier's output layer, where we're attempting to define the class of each input using probabilities. The SoftMax function is defined as in Equation 5

$$y = \max(0,x) \quad (6)$$

## 3. Result and discussion

To conduct this research, 8th Generation Intel i7 Processors with four cores and 8GB of RAM have been used to run code through a Jupiter notebook of the Anaconda environment. All four models- KNN, SVM, RF, and NN are trained one after another to complete this research.



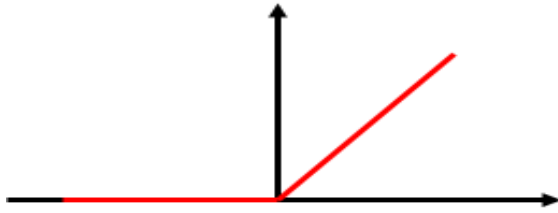


Figure 3: ReLU activation function

The main objective of this research is to tune the hyperparameters to improve the accuracy of handwritten Nepali Barnamala recognition using CSV data of a pre-processed image as an input. In addition to this, train the CSV data by using SVM, RF, KNN, and NNs models. In each model, different combinations of hyperparameters have been tuned to obtain the best results. All the combinations of hyperparameters of respective algorithms are shown in Table 1, 2, 3 and graph in Figure 4.

All the models had been tested on the test dataset after the completion of the training phase. The performance evaluation matrix for all models - SVM, Random Forest, and KNN was accuracy and F1-score. In this research, there is a total of forty-five different handwritten characters for recognition. So, this paper had only included the average accuracy and average F1-score rather than the individual scores of each handwritten character.

Table 1: The performance of SVM for different kernels and regularization values.

S.N.	Kernel	Regularization Parameter (C)	Avg. Acc.	Avg. F1-Score
1	Linear	0.1	77.19	0.77
2	Linear	1	76.78	0.76
3	Linear	50	76.78	0.76
4	Linear	500	76.78	0.76
1	rbf	0.1	80.35	0.80
2	rbf	1	84.01	0.84
3	rbf	50	87.48	0.87
4	rbf	500	87.48	0.87
1	poly	0.1	65.31	0.65
2	poly	1	79.36	0.79
3	poly	50	82.09	0.82
4	poly	500	82.09	0.82

The C parameter uses to manage the SVM optimization. For large values of C, the optimization will select a hyperplane with a smaller margin. A small value of C, on the other hand, will cause the optimizer to select for a larger-margin separating hyperplane, whether or not it misclassifies more points. The SVM with "RBF" kernel and C=500 has the best accuracy compared to

other hyperparameters.

Table 2: The performance of Random Forest classifier with hyperparameters: number of trees in the forest, best split criteria, and the number of features to consider when looking for the best split.

S.N.	K	No. of Trees	Best Split Criteria	Avg. Acc.	Avg. F1-Score
1	51	"gini"	"auto"	87.82	0.88
2	51	"gini"	"log2"	87.06	0.87
3	51	"entropy"	"auto"	88.60	0.89
4	51	"entropy"	"log2"	87.59	0.88
1	101	"gini"	"auto"	89.60	0.90
2	101	"gini"	"log2"	89.02	0.89
3	101	"entropy"	"auto"	89.92	0.90
4	101	"entropy"	"log2"	89.36	0.89
1	151	"gini"	"auto"	90.34	0.90
2	151	"gini"	"log2"	89.82	0.90
3	151	"entropy"	"auto"	90.61	0.91
4	151	"entropy"	"log2"	90.40	0.90
1	251	"entropy"	"auto"	90.81	0.91

Where K represent Kernel.

The number of trees to create a random forest is used 51,101, 151, and 251. The best attribute selection method is used both the Gini index and entropy method. The number of maximum features taken to select the best attribute is the auto or log2 method.

- If 'auto', then maximum\_features=sqrt(number of features).
- If 'log2', then maximum\_features=log2(number of features).

The Random forest with hyperparameters such as 251 number of trees, entropy-based best attribute split criteria, and 'auto' method to select the maximum number of features to build a tree has given the best average accuracy of 90.81%.

Table 3: The performance of SVM for different kernels and regularization values.

S.N.	No. of Nearest Neighbor	Distance Metric	Avg. Acc.	Avg. F1-Score
1	3	Manhattan	88.69	0.89
2	3	Euclidean	89.77	0.90
3	5	Manhattan	88.46	0.88
4	5	Euclidean	89.77	0.90
5	7	Manhattan	88.15	0.88
6	7	Euclidean	89.16	0.89

This research has used two different metrics to compute

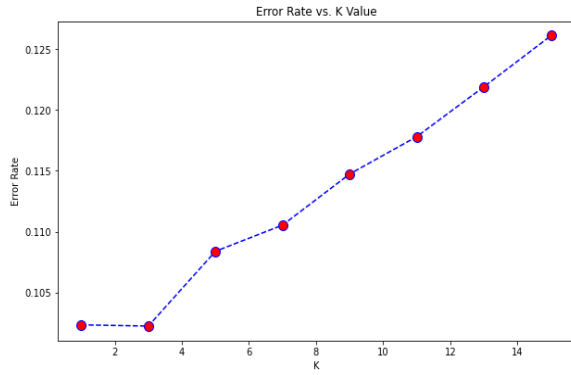


Figure 4: Error rate vs. number of K using distance matrix = “Euclidean”

the similarity between the handwritten characters for KNN. From fig 3, it is concluded that KNN has the minimum error rate for K=3 when the distance matrix is Euclidean distance. From the six different combinations of hyperparameters, it is concluded that KNN has the highest average accuracy 89.77% when distance matrix is Euclidean and K=5.

This research has built six different models of Neural networks. Among them, the best model has the highest test average accuracy is 93.51%, and minimum training loss 0.6476 represented in Figure 5 and 6. The first three models have suffered from the overfitting problem. The overfitting problem was solved by using the l1 and l2 regularization in kernel and bias. The learning rate is also decreased from 0.001 to 0.00019 to reduce the overfitting problem. In addition, this research has also reduced the number of neurons in the hidden layer from 1024 to 300 gradually and finally resolve the overfitting problem.

### 3.1. Testimony images

The following characters are randomly selected from the test dataset and correctly predicted by our models as shown in Figure 7.

## 4. Conclusion

This research has implemented four different models: Support Vector Machine, KNN, Random Forest, and Neural Network for handwritten Nepali Barnamala recognition. This paper mainly focused to compare them based on their average accuracy and F1-score value. It is found the neural network has the highest average accuracy compare to rest three models. The highest accuracy of the neural network is 93.51%, while SVM average accuracy is 87.84% with “RBF” kernel and C=500; Random Forest has average accuracy is 90.81%, and KNN has average accuracy is 89.77%. Even though

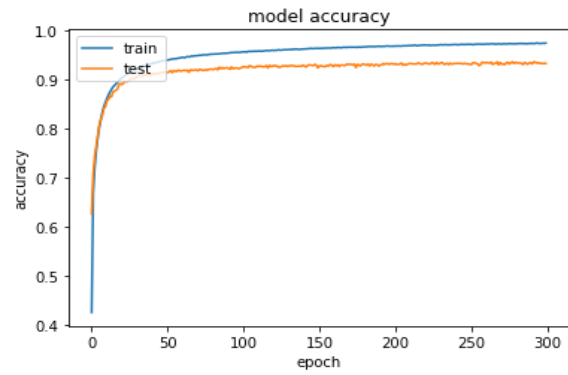


Figure 5: The training and test accuracy vs. Number of epochs when the hyperparameter is set as kernel\_ regularizer = regularizers.l1\_l2 (l1=1e-4,l2=1e-3), bias\_ regularizer = regularizers.l2 (1e-3), batch size = 256 and learning\_ rate = 0.00019 with four hidden layers of the Neural network model.

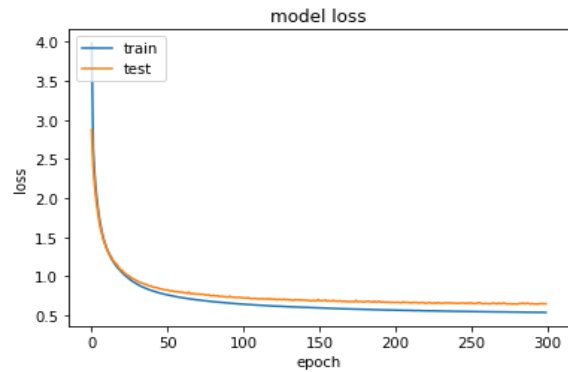


Figure 6: The training and test loss vs. Number of epochs when the hyperparameter is set as kernel\_ regularizer = regularizers.l1\_l2 (l1=1e-4, l2=1e-3), bias\_ regularizer = regularizers.l2 (1e-3), batch size = 256 and learning\_ rate = 0.00019 with four hidden layers of the Neural network model.

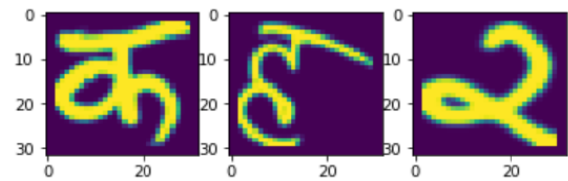


Figure 7: Randomly Selected testimony images

the average accuracy of the model is between 87.84% to 93.51%, the average accuracy of handwritten digits recognition is in the range of 91% to 98% for different models with different hyperparameters.

During the model training process, it is found that SVM

took almost five to six hours to train the complete model, whereas Random Forest took just four to six minutes to train the complete model. Depending on the number of hidden layers and number of neurons in the hidden layer, the neural networks just take four to ten seconds per epoch. The obtained average accuracy for all handwritten characters is considerably better than the previous state-of-art methods.

## References

- [1] Dixit R, Kushwah R, Pashine S. Handwritten Digit Recognition using Machine and Deep Learning Algorithms[J/OL]. International Journal of Computer Applications, 2020, 176 (42): 27-33. <http://www.ijcaonline.org/archives/volume176/number42/dixit-2020-ijca-920550.pdf>. DOI: <https://doi.org/10.5120/ijca2020920550>.
- [2] Bora M B, Daimary D, Amitab K, et al. Handwritten Character Recognition from Images using CNN-ECOC[J/OL]. Procedia Computer Science, 2020, 167: 2403-2409. <https://linkinghub.elsevier.com/retrieve/pii/S1877050920307596>. DOI: <https://doi.org/10.1016/j.procs.2020.03.293>.
- [3] Dawadi B R, Sharma S, Basnet A. Dictionary Based Nepali Word Recognition using Neural Network[J/OL]. International Journal of Scientific and Engineering Research, 2017, 8(June). <https://www.ijser.org/researchpaper/Dictionary-Based-Nepali-Word-Recognition-using-Neural-Network.pdf>.
- [4] Khandokar I, Hasan M, Ernawan F, et al. Handwritten character recognition using convolutional neural network[J/OL]. Journal of Physics: Conference Series, 2021, 1918(4). DOI: <https://doi.org/10.1088/1742-6596/1918/4/042152>.
- [5] Narayan A, Muthalagu R. Image Character Recognition using Convolutional Neural Networks[C/OL]// 2021 Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII). IEEE, 2021: 1-5. <https://ieeexplore.ieee.org/document/9445136/>. DOI: <https://doi.org/10.1109/ICBSII51839.2021.9445136>.
- [6] Dixit R, Kushwah R, Pashine S. Handwritten Digit Recognition using Machine and Deep Learning Algorithms[J/OL]. International Journal of Computer Applications, 2020, 176 (42): 27-33. <http://www.ijcaonline.org/archives/volume176/number42/dixit-2020-ijca-920550.pdf>. DOI: <https://doi.org/10.5120/ijca2020920550>.
- [7] Hearst M, Dumais S, Osuna E, et al. Support vector machines[J/OL]. IEEE Intelligent Systems and their Applications, 1998, 13(4): 18-28. <http://ieeexplore.ieee.org/document/708428/>. DOI: <https://doi.org/10.1109/5254.708428>.
- [8] Breiman L. Random forests[J/OL]. Machine Learning, 2001, 45 (1): 5-32. DOI: <https://doi.org/10.1023/A:1010933404324>.
- [9] Hand D J. Principles of data mining: volume 30[M/OL]. The MIT Press, 2007: 621-622. DOI: <https://doi.org/10.2165/00002018-200730070-00010>.