RESEARCH PAPER

Analyzing Deep Learning Methods for Handwritten Hindi Special Character & Numerals Recognition

Milee Brahmakshatriya[1]*, Dr. Devarshi Mehta[2]

[1*]Research Scholar Email: mileemadhu45@gmail.com University Name: GLS University
[2]Associate Professor Email: devarshi.mehta@glsuniversity.ac.in

Abstract –
In this research, we provide a system for recognizing handwritten Hindi Special characters and Numerals by a number of deep learning methods. Researchers have paid a lot of attention to character recognition because of its potential uses in helping the blind and visually handicapped, human-robot interaction, automated data input for commercial documents, and other areas.
In this research, we offer a solution to the recognition of handwritten Hindi characters by making use of deep learning techniques such as Convolutional Neural Network (CNN) with Optimizer RMSprop, Adaptive Moment Estimation (Adam), and Deep Feed Forward Neural Networks (DFFNN).
A big collection of photographs of handwritten Hindi characters is utilized to train the system. Samples from a user-defined dataset are used to test the trained algorithm. The experimental data shows that the recognition accuracy is rather good. To further evaluate the efficacy of the suggested system, it is compared to existing neural network-based methods.
In sum, the research unveils an innovative approach to the problem of detecting handwritten Hindi characters by making use of deep learning. The success of the recognition attempts shows the practical use of the method in other fields.
Keywords: DFFNN, Convolutional Neural Network (CNN), Optimizer RMSprop, Adaptive Moment Estimation (Adam), Deep Learning
How to cite this article: Brahmakshatriya M, Mehta D. Analyzing Deep Learning Methods for Handwritten Hindi Special Character & Numerals Recognition. Int J Drug Deliv Technol. 2026;16(72s): 1133-1144. DOI: 10.25258/ijddt.16.72s.126
.

1. INTRODUCTION

Deep learning strategies have been successfully used in a variety of domains, including image classification, voice recognition, the identification of faces in medical pictures, the recognition of traffic signs, and the detection of pedestrians, amongst other domains. The effects of using deep learning methods are also quite noticeable, and in some instances, the outcomes are better to those achieved by human specialists over the course of the previous years. Since the last few years, the majority of issues have also been re-experimented using deep learning approaches in the hope of generating improvements in the results that have already been obtained. In recent years, several designs of deep learning, such as convolutional neural networks, deep networks, and recurrent neural networks, have been brought to light. The whole of the architecture has shown proficiency in a variety of domains. One of the domains in which machine learning approaches have been explored with to a significant extent is character recognition. In 1998, using the MNIST database, the first deep learning approach was developed for character identification. Today, deep learning is considered to be one of the most advanced machine learning techniques.

The deep learning systems are primarily composed of a number of hidden layers, and each hidden layer is composed of a number of neurons that are tasked with calculating the proper weights for the deep network. To compute these weights needs a substantial amount of computing power, and it was required to utilise an advanced computer, which at the time was not readily available to a large number of people. Since then, "the researchers have focused their efforts on developing a method that reduces the amount of power that is required to process photos by first transforming those images into feature vectors. Over the course of the most recent few decades, a great variety of different strategies for the extraction of characteristics have been developed. These approaches, which include HOG (histogram of oriented gradients) and many more, are utilised as significant feature extraction methods and have been experimented with for a number of issues, including image identification, character recognition, face detection, and so on. Feature extraction is a sort of dimensionality reduction that reflects the relevant parts of a large image as a feature vector". This may be accomplished by reducing the number of dimensions involved. After that, this feature vector may be used to the analysis of the picture. The design of these elements has unquestionably been given a significant amount of attention and consideration by the scholarly community. The amount of experience and prior information that each researcher have is directly related to the resiliency and performance of these qualities. When the researchers are attempting to extract the characteristics from the image, there are some circumstances in which certain key parts may slip unobserved by the researchers; this may lead to a major classification mistake. Deep learning replaces

*Author for Corresponden mileemadhu45@gmail.com

the conventional method of constructing and developing features for a particular issue via human labour with a method that makes use of automation to compute the features that are most effective for finding a solution to that problem. The information may be automatically extracted using a convolutional neural network, which is formed of several layers of the convolutional processing. In the majority of shallow learning models, the features are only extracted once. On the other hand, in the case of deep learning models, several convolutional layers have been inserted in order to extract discriminating characteristics many times. This is done in order to make the results more accurate. Because of this element, deep learning models are generally successful, which is one of the reasons why these models are effective. In addition, the characteristics of deep feed forward neural networks are generated automatically via the utilisation of a variable number of hidden layers included within the network itself.

Character recognition is one use of a deep convolutional neural network, and the LeNet is a good example of one. In recent times, several other instances of deep learning models have been available, including AlexNet, ZFNet, VGGNet, and spatial transformer networks, to name a few. Image classification and character recognition are two areas that have benefited from the use of these models. As a result of their enormous amount of success, several industry leaders have also begun producing deep models. “The GoogLeNet that was developed by Google Corporation is comprised of 22 layers that alternate between convolutional and pooling layers. In addition to this model, Google has built an open-source software library known as Tensorflow” for the purpose of carrying out research in the field of deep learning. In the same year (2015), Microsoft also presented their very own deep convolutional neural network architecture, which was given the moniker ResNet. A new record has been set in terms of detection, localisation, and classification because to the 152-layer network topologies that ResNet has. A novel concept known as residual learning, which allows for the optimisation of various data types, was presented by this model.

Within the field of image processing known as “character recognition,” the process of identifying a picture and turning it into a format that a computer can read is carried out. This may be thought of as a combination of optical character recognition and optical character recognition software. For “the goal of photo identification and recognition, deep learning techniques, in particular convolutional neural networks, have been used, as was just said. In addition to this, it has shown to be beneficial when used to the Roman (MNIST) language, in addition to Chinese, Bangla, and Arabic. This study makes use of both a convolutional neural network and a Deep Feed Forward neural network in order to recognise handwritten Hindi characters. The goal of this research is to recognise handwritten Hindi characters”.

The following is the organisational scheme for the remaining portions of the paper: The related work may be found in Section 2, which is the next section. You will find a presentation of the recommended method in Section 3 of this document. In Section 4, we present both the results that were obtained as well as the particulars of the tests that were carried out. In the fifth and last part, we will provide our conclusion.

2. RELATED WORK

The authors Amit Choudhary et al. proposed “An Off-Line Handwritten Character Recognition Using Features Extracted from Binarization Technique. In order to identify handwritten characters of the English language, the goal of this work is to extract features achieved via the use of the binarization method. The use of a multi-layered feed forward artificial neural network as a classifier allowed for the successful recognition of handwritten character graphics”. This allowed for a higher degree of precision. This approach has remarkable success in terms of classification accuracy, as seen by its score of 85.62%.

For the sole purpose of recognising handwritten Gujarati digits, Baheti M. J. and colleagues provided a comparison of offline handwritten character recognition systems. An affine invariant moments-based model was employed throughout the process of feature extraction. After that, the numerals were categorised by using “the KNN classifier and PCA to minimise the dimensionality of feature space.” Finally, the numerals were classified by using a Euclidean similarity measure. In contrast, the PCA classifier was only able to reach an identification rate of 84%, while the KNN classifier was successful in achieving a recognition rate of 90%. After the comparison, it was found out that the KNN classifier had exhibited better performance than the PCA classifier had showed.

“Sonu Varghese K et al. came up with the idea for a novel tri-stage recognition method for handwritten Malayalam character recognition. The first thing that we do at the beginning of the process is separate the characters into their own independent groups based on the number of corners, bifurcations, loops, and endings that are included inside each of them. In the second phase of the process, we are employing a wide number of feature extraction techniques, each of which has been particularly built for a certain class, in order to recognise specific characters that are included inside the class. In the third phase, we are going to check the criteria that have been created for the building of words to see how likely it is that the current character will appear in the given location. We are now working on putting into practise a three-stage feature extraction approach that takes use of the structural, statistical, and moment variant aspects of the character”. This method will allow us to take full advantage of all of the character's unique qualities. Recognition that is carried out in a number of steps helps to an increase in both the recognition rate and the accuracy of the system that is being provided.

The R-HOG Feature served as the foundation for a method for distinguishing handwritten Marathi characters that was proposed by Parshuram M. Kamble. This method was developed by Kamble. The viability of the method was investigated by applying it to a large sample of handwritten Marathi characters and analysing the results. "It is possible to draw the following conclusion from the findings that have been presented: the utilisation of the R-HOG based feature extraction technique and the FFANN based classification will be more successful, with enhanced processing speed and accuracy."

Verma was able to successfully recognise the handwritten characters of "the Devanagari script by utilising radial basis function in addition to multilayer perceptron neural networks." Increasing the utilisation of the back propagation error approach is another method that might be used to boost the recognition rate. "They analyse the performance of radial basis function (RBF) networks and multi-layer perceptron models side by side and compare the results using the strategy that was recommended. The dataset includes 245 unique user instances, each of which was produced by a distinct user. Each of these examples was included in the dataset by accident. According to the data that were obtained in this way, the functionality of multilayer perceptron (MLP) networks is better to that of radial basis functions in terms of performance. On the other hand, the amount of time necessary to train MLP networks is much more time-consuming than the amount of time necessary to train radial basis function networks. Using radial basis function and multilayer perceptron (MLP) networks, the greatest recognition rates that have been achieved to this point are, respectively, 85.01% and 70.8%".

3. PROPOSED APPROACH

The recommended method may be divided into four separate steps or procedures to better understand it. During the first phase, "we will be collecting the character data from the Kaggle dataset as well as photographs from a variety of users," the author writes. After the character data of grayscale photographs have been gathered, the images will go through pre-processing, which consists of checking for null and missing values. This will happen after the character data have been acquired. In order to create the vector form of the data, the following stages will be taken: first, normalisation techniques will be used to change the grey level values to a range of values between 0 and 1; next, labelling Hindi characters from 0 to 35 with one hot code will generate a vector form of the data; finally, the vector form of the data will be constructed. During the third phase, we were able to distinguish handwritten letter systems by extracting features atomically from a number of different deep learning algorithms. These techniques included convolutional neural networks (CNN) and deep feed forward neural networks (DFFNN). In the last stage of the process, we used an optimisation approach such as Adaptive Moment (Adam) Estimation and Optimizer RMSprop (Root Mean Square Propagation) to get very promising results. Figure 1 provides a block diagram representation of the proposed method, which can be viewed further down on this page.

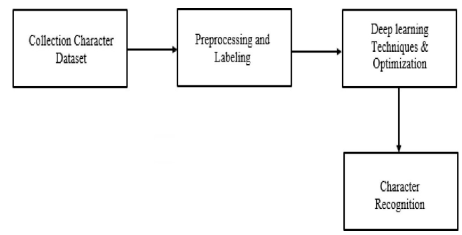

Figure 1: Proposed block design for a handwritten number recognition system

A. Convolutional neural network

The discipline of computer vision and pattern recognition is seeing "the most rapid expansion" of all the subfields that go under the umbrella of "image processing." The science of computer vision is primarily reliant on convolutional neural networks, often known as CNNs, which play an important part in the industry. CNNs is now working on a wide variety of applications in the field of image classification, and it serves as the central processing unit in the majority of computer vision and pattern recognition systems in use today. These systems, which may be used in applications as diverse as the automatic tagging of images on Facebook and autonomous cars, are able to distinguish numbers, alpha-numerals, traffic signal boards, and other types of object categories. CNN is presently focusing its efforts on developing these apps. The model that we worked with was known as a Convolutional Neural Network (CNN), and it had five different layers. "On them, there is one layer for convolutional processing, one layer for maximum pooling or subsampling, one layer for flattening, which transforms a two-dimensional array into a one-dimensional array, and finally, there are two completely connected layers for classification. The very first layer is a convolutional (Conv2D) layer, and it has 32 output mapping". The layer that follows it is a max pooling layer, and it has 14 output mapping.

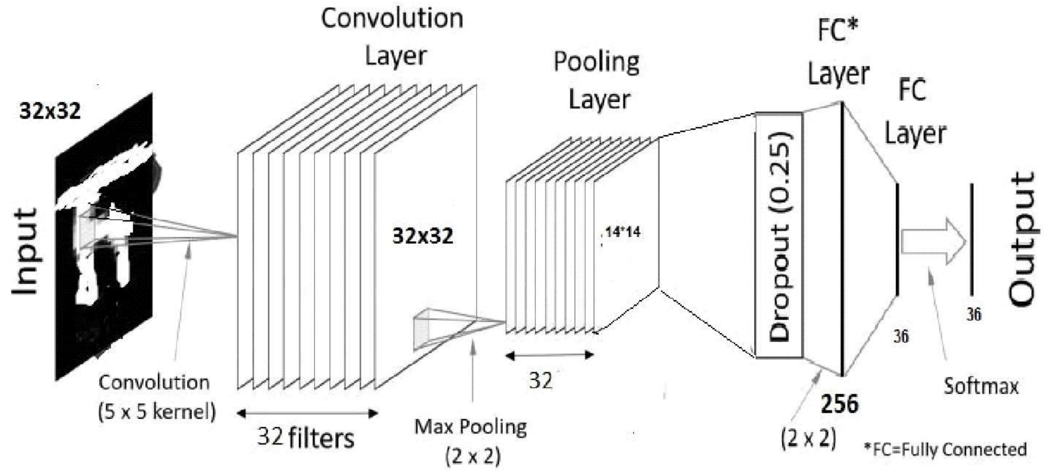

Figure 2: The multi-layer architecture of the CNN model that underpins our suggested solution.

Adam and RMSprop are two of the most famous adaptive stochastic algorithms that are used for the training of deep neural networks. The decade of the 1980s saw the conception and development of both of these algorithms.

There have been numerous attempts made, some of which include decreasing “an adaptive learning rate, employing a big batch size, adopting a temporal decorrelation technique, seeking for a comparable surrogate, and a great deal of other things. There has not been a single one of them that has been successful. In addition, we show that Adam is basically a specially weighted AdaGrad with exponential moving average momentum. This provides a new perspective from which to understand Adam and RMSProp, which is quite helpful. In the end, we were able to demonstrate that the essential condition was met by using Adam and RMSProp to handle the counterexamples and to train deep neural networks”.

B. RMSprop Optimizer

The RMSprop optimizer, which is also known as “the Root Mean Square Propagation optimizer, performs its duties in a way that is analogous to the gradient descent approach with momentum. The RMSprop optimizer imposes limits on the amount of oscillation in both the positive and negative directions. As a result, we are able to accelerate the pace at which we are learning, and our algorithm will be able to take more expansive leaps in the direction of horizontal convergence”. It has been shown to be effective for training Recurrent Models.

$$S_{dw} = \beta s_{dW} + (1-\beta)\left(\frac{\partial J}{\partial W}\right)^2 \quad (1)$$

$$W = W - \alpha \frac{\frac{\partial J}{\partial W}}{\sqrt{\left(s_{dW}^{corrected}\right) + \varepsilon}} \quad (2)$$

Where,

ss - the average square of previous gradients, weighted exponentially

- re $\frac{\partial J}{\partial W}$ the present weight tensor of layers, there is a cost gradient W - weight tensor

β - hyperparameter to be tuned α - the learning rate

ϵ - small enough that division by zero is avoided

Similar to RMSprop, the average squared gradients are divided by an exponentially decaying rate to get the learning rate.

C. Adam Estimation Optimizer

Another approach that computes adaptive learning rates for each parameter is called the Adaptive Moment (Adam) Estimation. Adam is a method for first-order gradient-based optimisation of stochastic objective functions. It is based on adaptive estimates of lower-order moments and is designed to maximise the value of stochastic objective functions. The approach is simple to develop, has high computing efficiency, requires a little amount of memory, is unaffected by diagonal rescaling of the gradients, and works very well for situations that include a lot of data and/or parameters.

$$m_t = \beta_1 * m_{t-1} + (1-\beta_1) * g_t \quad (3)$$

$$v_t = \beta_2 * v_{t-1} + (1-\beta_2) * g_t^2 \quad (4)$$

The name of the approach comes from “the fact that m_t and v_t are estimates of the first moment (also known as the mean) and the second moment (also known as the uncentered variance) of the gradients, respectively. The authors of Adam note that since m_t and v_t are initialised as vectors of zeros, they have a bias towards zero, particularly during the initial time steps and particularly when the decay rates are low (i.e. β_1 and β_2 are near to 1)”. This is because m_t and v_t are initialised as vectors of zeros. They do this by producing bias-corrected first and second moment estimates, which helps them fight against these biases:

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t} \quad (5)$$

$$\widehat{v_t} = \frac{v_t}{1-\beta_2^t} \quad (6)$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{v_t+\varepsilon}} * \widehat{m_t}$$

Similar to what we've seen in Adadelta and RMSprop, they utilise these to modify the parameters, resulting in the Adam update rule:

D. Deep Feed-Forward Neural Network (DFFNN)

This particular "Neural Network is known as a feed-forward network, and it is considered to be "the neural network that is the least complicated to analyse." A feed-forward neural network's input layer includes an n-dimensional vector as one component of the network's input, and it includes L-1 hidden layers as one component of the network's intermediate layers". The typical number of concealed layers that are employed is two; however, the number of these layers might potentially increase depending on the circumstances. In conclusion, there is one output layer that has a total of k output classes inside it. This output layer is the only one. There are neurons present in both the hidden layer and the output layer, and these neurons may be broken down into two distinct categories: the pre-activation component and the activation portion. The hidden layers in a deep feed-forward network are able to generate a wide network of features because "the hidden layers in a deep feed-forward network are considered to be of size m-dimensional vector size," which is much larger than the hidden layers in a normal feed-forward network.

Because there is such a vast number of features, we are able to find the right match for each of the classes that are predicted. A deep feed-forward network is used in order to distinguish the various kinds of testing pictures that are included in the data set and to automatically extract features from photos. This research makes use of Figure 3, which provides a visual representation of the DFFNN. This figure is being used in the current investigation.

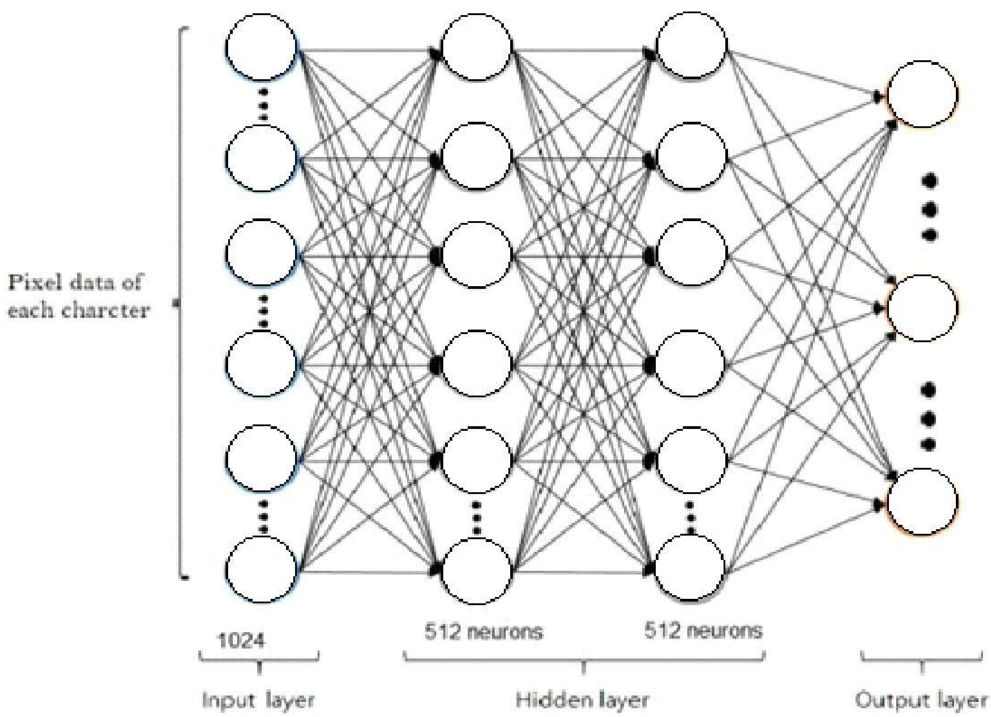

Figure 3: The multi-layered architecture of the DFFNN model that underpins our proposed solution.

4. ANALYSIS AND REFLECTION

A "Feed-forward Neural Network" will, in most cases, consist of a number of different hidden layers. "The majority of FFNN will contain two hidden layers with 16 or 32 neurons, and maybe even more. This is because hidden layers are multiplied with different random weights of picture pixel data, which can range from 0 to 1. However, the Deep Feedforward Neural Network was created with the same two hidden layers, and each hidden layer consists of a significant number of neurons; more precisely, we used 512 neurons and multiplied them with random weights for each hidden layer. By using this extensive network, we were able to get some really positive results". On this page, labels have been assigned to some of the characters that may be found in Table 1.

Layer Type	Layer Operation	No of Feature Maps	feature Maps Size	Window Size	Total Parameters
C1	Conv2D	32	32×32	5×5	832
MP1	Max-Pooling2	32	32×32	2×2	0
FL1	Flatten	6272	1×1	N/A	0
F1	Fully Connected	256	1×1	N/A	1605888
F2	Fully Connected	36	1×1	N/A	9252

Table 1: Labelling data sample representation.

In this experiment, we used a five-layer convolutional neural network (CNN) optimised with Adam. The model comprised one convolutional layer, one max-pooling layer, one flattening layer that converted the two-dimensional feature maps into a one-dimensional vector, and two fully connected layers for classification. Table 2 lists the parameters for each layer.

Table 2. Adam Optimisation is used to build up CNN's parameters.

Layer Type	Layer Operation	No of Feature Maps	feature Maps Size	window Size	Total Parameters
C1	Conv2D	32	32×32	5×5	832
MP1	Max-Pooling2	32	32×32	2×2	0
FL1	Flatten	8192	1×1	N/A	0
F1	Fully Connected	256	1×1	N/A	2097408
F2	Fully Connected	36	1×1	N/A	9252

During this round of testing, we made use of a "five-layered Convolutional Neural Networks (CNN) model that was optimised using RMSprop. On them, there is one layer for convolutional processing, one layer for maximum pooling or subsampling, one layer for flattening, which transforms a two-dimensional array into a one-dimensional array, and finally, two completely connected layers for classification". Table 3 has a comprehensive summary of all the factors in relation to the layers that are significant to them.

Table 3. Adam Optimisation is used to build up CNN's parameters.

ta	la	ma	sa	gya
ट	ल	म	स	ज्ञ

Table 4: A centralised repository for individual test scores DFFNN

Labeled Number	Hindi Character	No.of Images	Correctly Classified	Not Correctly Classified	% Accuracy
1	क	375	358	17	95.47
2	ख	380	375	5	98.68
3	ग	390	388	2	99.49
4	घ	398	390	8	97.99
5	ड़	375	370	5	99.05
6	च	395	380	15	96.20
7	छ	370	360	10	97.3
8	ज	395	390	5	98.73
9	झ	385	370	15	96.10
10	ञ	395	385	10	97.47
11	ट	398	387	13	97.24
12	ठ	405	389	16	97.22
13	ड	413	400	13	96.85
14	ढ	402	390	12	97.01
15	ण	395	384	16	97.22
16	त	395	384	11	97.22
17	थ	380	358	22	94.21
18	द	409	394	15	96.33
19	ध	378	353	25	93.39
20	न	405	383	22	94.57
21	प	393	370	23	94.15
22	फ	407	393	14	96.56
23	ब	395	379	16	95.95
24	भ	401	372	29	92.77
25	म	398	387	11	97.24
26	य	378	360	23	95.24
27	र	391	379	12	96.93
28	ल	395	390	5	98.73
29	व	378	352	26	93.12
30	श	378	364	14	96.3
31	स	382	360	16	96.21
32	ष	391	368	23	94.12
33	ह	383	354	29	92.43
34		378	360	18	95.24
35		388	370	18	95.36
36		382	360	22	94.24
Average Recognition Percentage					96.17

Table 5: Individual Test Database Results Using CNN-Adam's Approximate

Labeled Number	Hindi Character	No.of Images	Correctly Classified	Not Correctly Classified	% Accuracy
1	क	383	370	13	96.61
2	ख	401	389	12	97.01
3	ग	403	398	5	98.76
4	घ	387	373	15	96.38
5	ड़	352	336	16	95.45

6	च	370	352	18	95.14
7	छ	374	346	28	92.51
8	ज	413	409	4	99.03
9	झ	370	351	19	94.86
10	ञ	415	410	5	98.8
11	ट	421	414	7	98.34
12	ठ	389	373	16	95.89
13	ड	400	397	3	99.25
14	ढ	390	378	12	96.92
15	ण	414	408	6	98.55
16	त	384	373	11	97.14
17	थ	358	336	22	93.85
18	द	394	379	15	96.19
19	ध	376	351	25	93.35
20	न	383	375	8	97.91
21	प	370	363	7	98.11
22	फ	393	381	12	96.95
23	ब	379	369	10	97.36
24	भ	372	351	21	94.35
25	म	387	378	9	97.67
26	य	387	365	22	94.32
27	र	409	397	12	97.07
28	ल	390	385	5	98.72
29	व	397	392	5	98.74
30	श	404	398	4	98.51
31	स	406	392	14	96.55
32	ष	382	365	17	95.55
33	ह	413	385	28	93.22
34		418	393	25	94.02
35		390	375	15	96.15
36		395	372	23	94.18
Average Recognition Percentage					96.89

Table 6: Dataset-specific CNN Evaluation Outcomes-RMSprop

Labeled Number	Hindi Character	No.of Images	Correctly Classified	Not Correctly Classified	% Accuracy
1	क	427	424	3	99.3
2	ख	392	388	4	98.99
3	ग	426	419	7	98.36
4	घ	386	371	15	96.11
5	ङ	375	359	16	95.73
6	च	397	389	8	97.98

7	छ	349	329	20	94.27
8	ज	355	345	10	97.18
9	झ	410	401	9	97.80
10	ञ	387	382	5	98.71
11	ट	384	381	3	99.22
12	ठ	395	389	6	98.5
13	ड	402	399	3	99.26
14	ढ	378	366	12	96.83
15	ण	400	394	6	98.52
16	त	393	392	1	99.75
17	थ	373	351	22	94.10
18	द	415	410	5	98.81
19	ध	386	371	15	96.11
20	न	408	396	12	97.06
21	प	386	373	13	96.63
22	फ	396	382	14	96.46
23	ब	354	348	6	98.31
24	भ	391	376	15	96.16
25	म	376	365	11	97.07
26	य	385	362	23	94.03
27	र	395	383	12	97.07
28	ल	416	411	5	98.81
29	व	407	403	4	99.03
30	श	395	379	16	95.95
31	स	370	363	7	98.11
32	ष	393	380	13	96.69
33	ह	365	346	19	94.69
34		384	366	18	95.31
35		407	398	9	97.79
36		360	345	15	95.83
Average recognition percentage					97.24

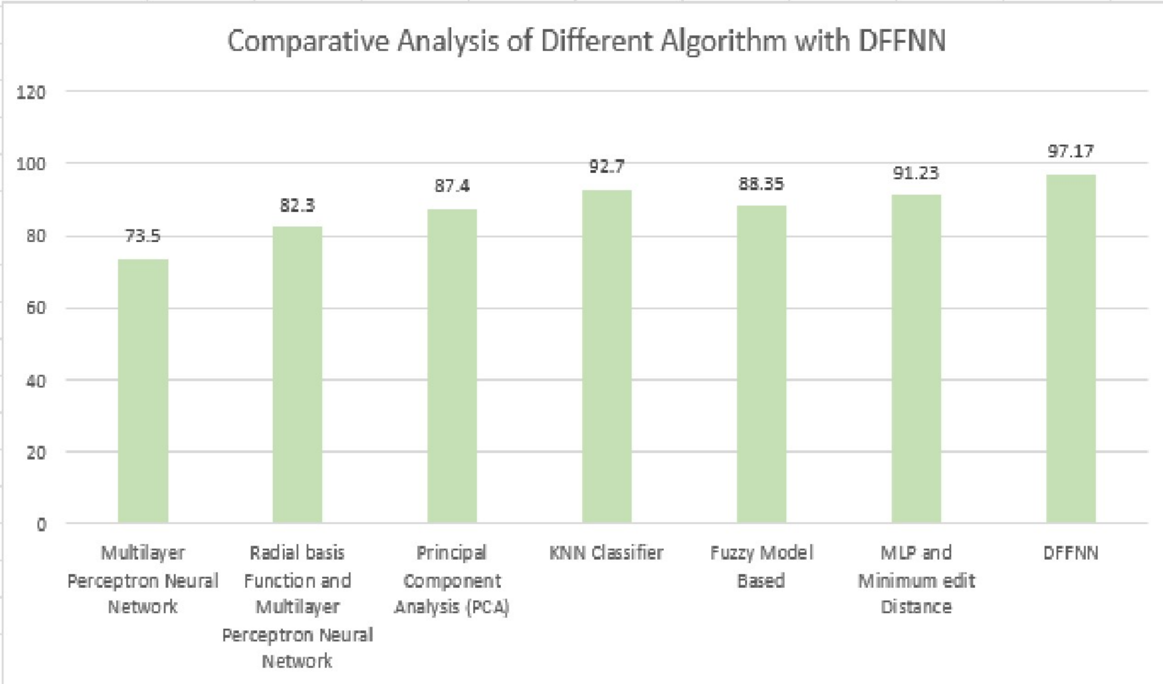

Figure 4: DFFNN and other known technique percent recognition shown graphically

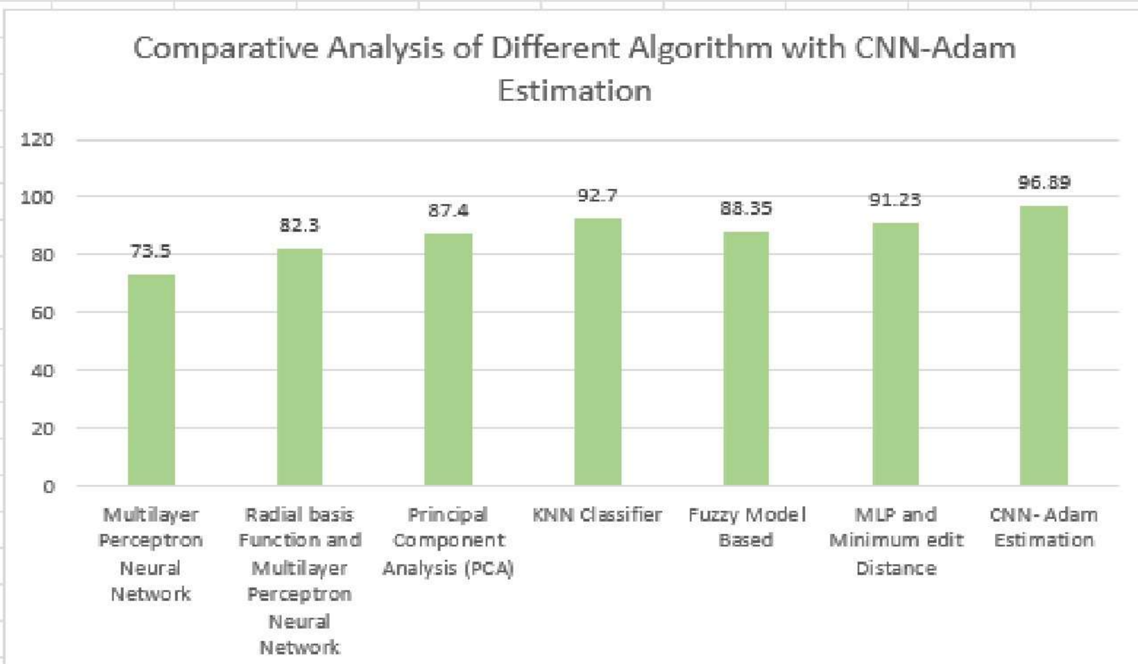

Figure 5: Comparison of the CNN-Adam approach and other current methods in terms of percent recognition

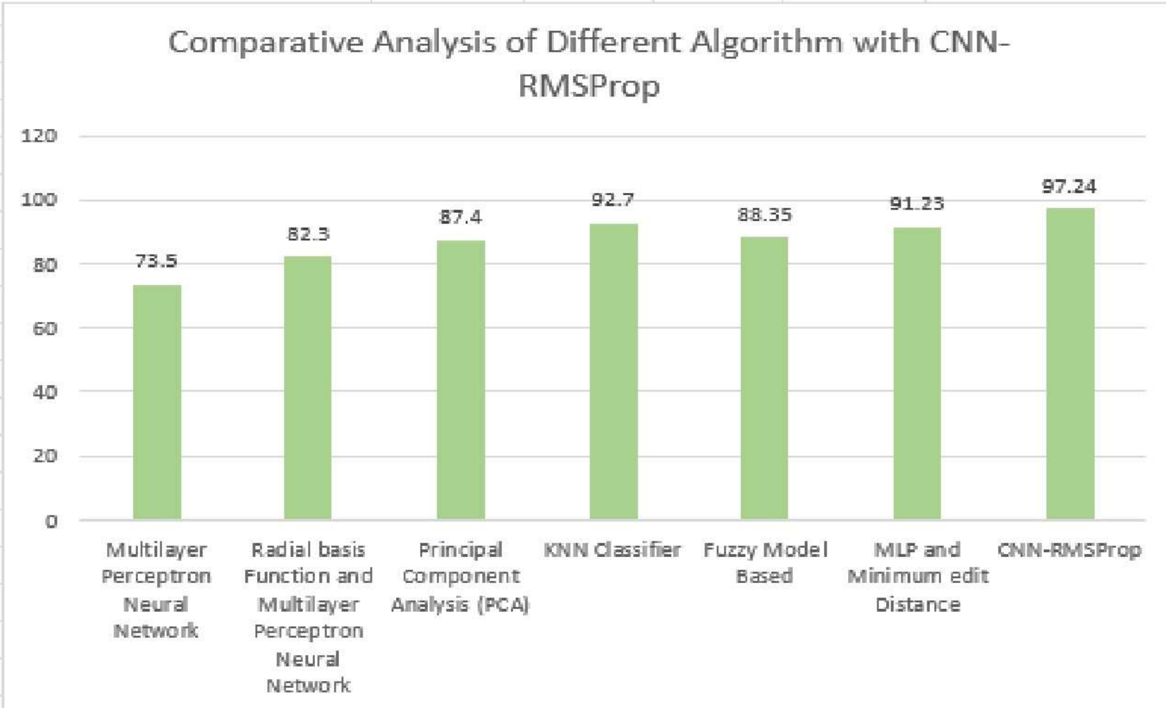

Figure 6: Comparison of the CNN-RMSprop approach to various current methods in terms of percent recognition

5. CONCLUSION

Within the scope of this research, we investigated many different neural network strategies for the purpose of character recognition in handwritten Hindi. We studied the performance by applying optimisation techniques to Deep Feed Forward Neural Networks and Convolutional Neural Networks (CNNs). These algorithms are trained and evaluated on a standardised user-defined dataset that is compiled from the contributions of a number of different people. According to "the findings of the experiments, it has been shown that DFFNN, CCN-Adam, and CNN-RMSprop provide the highest level of accuracy for handwritten Hindi characters when compared to the other approaches," DFFNN, CCN-Adam, and CNN-RMSprop provide the highest level of accuracy for handwritten Hindi characters. Using the method that was proposed, which had a high degree of accuracy, we were able to generate some positive data.

REFERENCES

- Ciregan, D., Meier, U.& Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence*, 16–21.
- . Krizhevsky, A., Sutskever, I. Hinton, G.E (2012). Imagenet classification with deep convolutional neural networks. *In Proceedings of the Advances in Neural Information Processing Systems*, 3–8.
- Lecun, Y., Bottou, L., Bengio, Y. & Haffner P. (1998). Gradient-based learning applied to document recognition. *Proc.* IEEE *86*, 2278–2324.
- Navneet, D.; Triggs, B. "Histograms of oriented gradients for human detection", In Proceedings of the CVPR2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA,20–25 June 2005; Volume 1
- Wang, X.; Paliwal, K.K. "Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition", Pattern Recognit. **2003**, 36, 2429–2439.
- Zeiler, M.D.; Rob, F. "Visualizing and understanding convolutional networks", In Proceedings of the EuropeanConference on Computer Vision, Zurich, Switzerland, 6–12 September 2014.
- Simonyan, K.; Andrew, Z. "Very deep convolutional networks for large-scale image recognition". arXiv, 2004.
- Jaderberg, M.; Simonyan, K.; Zisserman, A, "Spatial transformer networks", In Proceedings of the Advances inNeural Information Processing Systems, Montreal, QC, Canada, 11–12 December 2015.
- Cire¸san, D.; Ueli, M. "Multi-column deep neural networks for offline handwritten Chinese characterclassification", In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN),Killarney, Ireland, 12–17 July 2015.
- . Sarkhel, R.; Das, N.; Das, A.; Kundu, M.; Nasipuri, M. "A Multi-scale Deep Quad Tree Based

Feature Extraction Method for the Recognition of Isolated Handwritten Characters of popular Indic Scripts", Pattern Recognition.**2017**, 71, 78–93.

- . Ahranjany, S.S.; Razzazi, F.; Ghassemian, M.H. "A very high accuracy handwritten character recognition system for Farsi/Arabic digits using Convolutional Neural Networks", In Proceedings of the 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), Changsha,China, 23–26 September 2010.
- Choudhary, A., Rishi, R., and Ahlawat, S., "Off-Line Handwritten Character Recognition using Features Extracted from Binarization Technique", AASRI Conference on Intelligent Systems and Control, 2013, pp. 306-312.
- Baheti M. J., Kale K.V., Jadhav M.E., "Comparison of Classifiers for Gujarati Numeral Recognition", International Journal of Machine Intelligence, Vol. 3, Issue 3, pp. 160-163, 2011.
- Sonu Varghese K, Ajay James, Dr.Saravanan Chandran , "A Novel Tri-Stage Recognition Scheme for Handwritten Malayalam Character Recognition", International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST - 2015),2016 pp-1333-1340.
- Parshuram M. Kamble, Ravinda S. Hegadi, "Handwritten Marathi character recognition using R-HOG Feature", International Conference on Advanced Computing Technologies and Applications (ICACTA- 2015),2015 pp- 266 – 274.
- B. K. Verma, "Handwritten Hindi character recognition using multilayer perceptron and radial basis function neural networks in Neural Networks", 1995. Proceedings, IEEE International Conference on. vol. 4, pp. 2111-2115, 1995.
- Hailiang Ye, Feilong Cao, Dianhui Wang, Hong Li , "Building feed forward neural networks with random weights for large scale datasets", Expert Systems with Applications Volume 106, 2018, pp. 233-243
- Feilong Cao, Dianhui Wang, Houying Zhu, Yuguang Wang, "An iterative learning algorithm for feedforward neural networks with random weights", Information Sciences, Volume 328, 2016, pp.546-557.