Applying AI to Biometric Identification for Recognizing Text using One-Hot Encoding and CNN

Abhishek Jha[1], Dr. Hitesh Singh[2], Dr. Vivek KumarDr[3], Dr. Kumud Saxena[4]
[2]Associate Professor, [3]Professor, [4]HOD,
[1,2,3,4]Computer Science and Engineering, NIET, AKTU, India

Abstract:- Text on an image often contains important information and directly carries high-level semantics in academic institutions and financial institutions. This makes it an important source of information and a popular research topic. Many studies have shown that CNN-based neural networks are very good at classifying images, which is the foundation of text recognition. By combining AI with the process of biometric identification, a technique for text recognition in academic institutions and financial institutions is performed using Convolutional Neural Network (CNN). Initially, preprocessing is done for making the document image suitable for feature extraction. One hot encoding-based feature extraction is performed. Two-dimensional CNN is used to classify the final features. Finally, RMSprop is used to optimize the results and improve the accuracy. Results of the proposed method show that the accuracy is 99%, which is more when compared to the existing methods.

Keywords:- *Adam optimizer, AI, CNN, RMSprop, One hot encoding .*

I. INTRODUCTION

In today's world, optical character recognition (OCR) technologies are utilised for a wide variety of tasks, some of which include scanner-based data entry, bank checks, business cards, automatic mail sorting, handheld price label scanners, and a variety of document recognition applications [1]. These days, the market also contains the commercial character identification scheme that are accessible to purchase. As a direct result of this innovation, researchers are now able to work on the issue of online handwriting recognition. This was made possible as a result of the development of electronic tablets that collected the x, y coordinate data as well as the movement of the pen tip [2].

The quality of the source document that is supplied into any OCR system has a noteworthy influence on the performance of that system. It is possible that the original document is rather old and that it's undergone some physical wear and tear. It is possible that the original document was of poor quality because to the changes in toner density that were present in it [3]. There is a possibility that the scanning process will miss some of the fainter sections that were present in the original document, which might result in the corruption of many characters in the original text. Gradations in the text picture can be brought about by scanning on low-quality paper or by printing of low-quality copies. The performance of optical character recognition is also significantly impacted by other factors, including accurate line and word segmentation [4].

The method of selecting and extracting the most unique characteristics from an image is known as feature extraction and selection. This peculiar 5 feature has the significant ability to accentuate or expand the difference between different class patterns while maintaining its invariance to the same class patterns of other kinds. This is one of its important characteristics [5].

The class of methods that are used for character recognition in practice is, in principle, not dissimilar to the class of methods that are used for any broad pattern recognition issue. On the other hand, based on the characteristics that are employed, the various methodologies for character recognition may be roughly categorized as follows: Approaches for matching templates and performing correlations, as well as techniques for analyzing and matching features [6].

Template matching and correlation techniques: These approaches are high-level machine vision algorithms that identify the entirety or a portion of an image that match a standard prototype template. These techniques may be used to either a single picture or several images simultaneously. In this step, the pixels of an input character are compared, one by one, with character prototypes that have been saved before [7].

Feature analysis and matching techniques: At the present time, the methods of character recognition that are utilized most frequently include feature analysis and matching algorithms. This strategy is also sometimes referred to as the structural analysis method. These approaches replicate human thinking better than template matching did, which was the previous standard. Using this methodology, relevant characteristics are first retrieved from the input character, and then the extracted features are compared with the feature descriptions of the trained characters. Recognition can be achieved by using the description that is the closest match [8].

Many of today's OCR systems are modelled based on mathematical models in order to reduce the amount of categorization errors that occur. A method for recognizing characters like these might make use of either structural or pixel-based information. The following list includes some of them. Using hyper surfaces in multi-dimensional feature spaces, discriminant function classifiers attempt to reduce the mean-squared classification error by separating the feature descriptions of characters that belong to various semantic classes [9]. Bayesian classifiers make use of probability theory in order to minimize a loss function that is connected with character misclassification. The theories of human and animal vision are the basis for Artificial

Neural Networks (ANN), which use error BP methods to decrease the possibility of incorrect categorization. SVM is an important innovation in the field of machine learning algorithms. It is a technique that is built on a kernel. There is a collection of techniques for supervised learning that may be used for classification or regression, and these are the techniques [10][11].

Most recent articles have incorporated a machine learning method in certain way. Particularly it is utilized widely for the identification of visual characters largely owing to the massive dataset availability. The researchers frequently utilize a machine learning strategy for a language with big dataset for machine learning in order to study the meaningful method. This improvement has come with more computational complexity, as was indicated above, even if the frameworks depending on machine learning approaches achieve greater accuracy during classification [12]. There have only been a handful of studies conducted in recent years that have used a traditional method of feature extraction in conjunction with algorithmic approaches to feature selection and have succeeded in producing state-of-the-art results. Additional research difficulty that requires the research community attention is the construction of systems that are able to recognize on-screen text and characters in various circumstances in everyday life situations [13].

To overcome these issues, we applied AI to Biometric Identification for Recognizing Text Using One-hot encoding and CNN in case of documents related to academic institutions and financial institutions to significantly improve performance and reduce data storage needs. The novelty of the suggested work is that it combines one-hot encoding feature extraction, CNN classifier, and RMSprop optimizer to produce the highest recognition accuracy for text recognition among peer researchers. When compared to other machine learning models used by peer researchers, the recognition accuracy obtained in this work utilising the CNN + RMSprop model is superior.

The rest of the paper is structured as follows: The related work in the field of text recognition is described in Section 2, the proposed approach with the pseudocode is described in Section 3, the results are discussed and a comparative analysis is presented in Section 4, and the conclusion and recommendations for the future are presented in Section 5.

II. LITERATURE REVIEW

A deep learning-based model for ACR has been proposed by Guptha et al. [14]. The data is gathered and then pre-processed with Gaussian filtering and skew detection. As an added step, projection profile and thresholding methods are used to separate the lines and characters from the denoised images. Elephant Herding Optimization (EHO) is used to pick discriminative features from the extracted features of the segmented images, while Enhanced Local Binary Pattern and Inverse Difference Moment Normalized descriptors will be used to extract features from the images.

When it comes to saving space and maximising performance, the idea of applying partial least square (PLS) based feature reduction for optical character recognition was introduced by Akhtar et al. [15]. Here, we offer a novel approach to automated OCR based on the fusion and selection of several attributes from a given set of features. Parallel LR (PLS) based selection is used to combine the features serially. An entropy fitness function is used for the selection process. An ensemble classifier is then used to label the completed features. This paper presents a method that can be used to make printed text machine-readable, which has applications in areas such as licence plate recognition.

Nadeem and Rizvi [16] proposed using template matching to identify both typed and handwritten characters. Rather than manually recognising each input pattern, one of the goals is to create a system that can automatically categorise them into the class to which they belong based on the information they contain. The system is responsible for character recognition. When compared to the recognition rate of typewritten Standard English alphabets fonts (94.30%), the rate at which handwritten English alphabets are understood is significantly lower at 75.42%.

Adhvaryu [17] presented an algorithm for matching templates for alphabets. The letters of the alphabet, from A to Z, were utilised in tests, and photos were grayscale with Times New Roman font type so that the prototype system could recognise the letters by comparing them. The prototype's limitations are that it can only be used to test with the alphabet, that it can only use grayscale photographs with the font type, and that it can only use the Template Matching method to recognise the letters.

Ziaratban et al. [18] suggested a method for character recognition they called "template matching." In order to do feature extraction, this technique looks for distinct templates within the input photos. Each template's success rate is recorded as a feature, and the optimal matching area in an image is pinpointed and archived.

Rajib et al. [19] offer an HMM-based method for recognising English handwritten characters. In their research, they have used both system-wide and neighborhood-specific techniques for extracting features. The HMM is trained with these features, and trials are run. They have compiled a database of 13,000 examples, with five examples produced for each character by each of 100 writers. After 2600 samples were used to teach the HMM, we are now using the remaining samples to validate the recognition architecture. A recognition rate of 98.26% is achievable with the proposed system.

Jamtsho et al. [20] presented a model for a system where the system can read aloud the characters/text in the photos. A model for OCR and a model for speech synthesis make up the system's two major subsystems. The characters and text in the photos are recognised by the OCR model, which then turns them into editable text by a variety of techniques including preprocessing, segmentation, and classification. Deep learning (DL) neural networks and machine learning (ML) are used to train both models.

A segmentation-free OCR system that incorporates DL techniques, the creation of synthetic training data, and data augmentation strategies was presented by Namysl and Konya [21]. With the use of enormous text corpora and more than 2000 fonts, we render synthetic training data. We add geometric distortions to collected samples as well as a suggested data augmentation method called alpha-compositing with background textures to emulate text appearing in complex natural situations. The CNN encoder in our models is used to extract features from text images.

The main focus of Surana et al. [22] was on various ML techniques that may be used to extract text from handwritten documents and photos, recognise them in digital format, and then translate it in accordance with the user's needs.

An innovative concept and implementation of an OCR-based application for Automated NGO Connect using ML were offered by Sharma et al. [23]. Image de-noising, binarization, data extraction, and data conversion are among the phases put into practise. Tesseract OCR based on DL is included into the framework along with image processing and data visualisation modules.

III. PROPOSED METHODOLOGY

A. Overview

In this paper, optical character recognition is performed for the images of the document. Initially, median filtering is applied to eliminate the noise in the image. After that, RGB to HSV transformation is done. The representation of the HSV image should be chained to binary using binarization method. One hot encoding-based feature extraction is performed. CNN is used to classify the final features. Finally, RMSprop is used to optimize the results and improve the accuracy. The entire block diagram of the proposed algorithm is illustrated here in Figure 1.

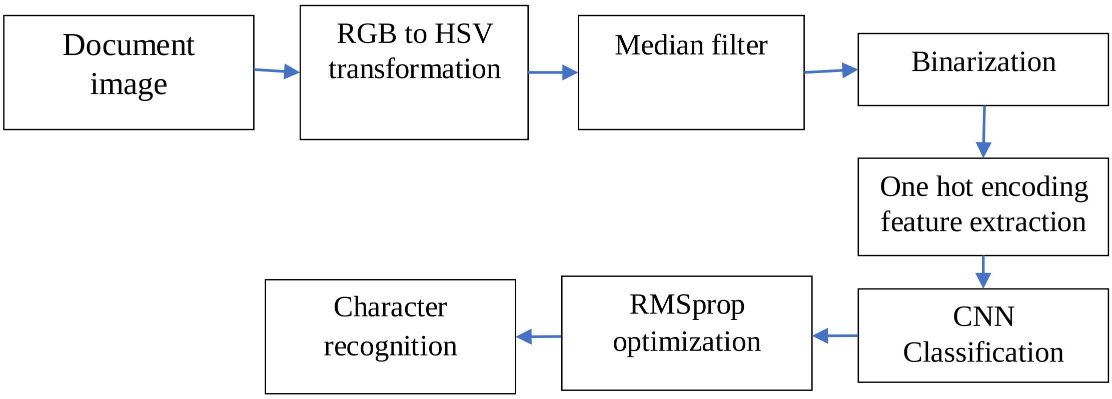

Fig. 1: Block Diagram

In this chapter, we discuss the proposed OCR algorithm by using geometric and texture features. The algorithm has four main phases: (a) preprocessing, (b) features extraction, (c) classification, (d) Optimization.

B. Preprocessing

The OCR system can acquire images of the document either by scanning the text or by taking a photograph of the text. Before carrying out any processes, photos are scaled down to 256 by 256 pixels for normalization purposes. After that, the proposed method iterated four times to construct an existing three-dimensional median filter, which ensures that there is no noise, maintains the image's quality, and prevents edges. It did a good job of getting rid of the salt and pepper sounds by preventing a significant level of blurriness. After being filtered, the images in RGB are converted to the HSV color system. Utilize the HSV color space, which differentiates chromatic information, so that color visibility can be differentiated. The median filter will serve as the foundation for the purpose of noise elimination. The adjacent pixels will be ranked based on their intensity, and the median value is used to determine the novel value for the central pixel after the median filter has been applied. The final product of the filtering method can still contain a background that is faintly textured or coloured, which could disrupt the operation of the future phases. In order to remedy this situation, a method known as binarization is applied to the filtered image to produce a binary representation of it. In this method, a threshold value is determined, and the pixels whose intensity values are more than the threshold are made to be white (0), while less than the threshold are made to be black. The value of the threshold is determined by finding the document's overall average pixel intensity and then subtracting that amount from 1.

C. Feature Extraction Using One-hot encoding

Many significant properties in real-world datasets are categorical rather than numerical. These categorical features must be changed or converted into a numeric format in order to be used in training and fitting machine learning algorithms because they are crucial for increasing the accuracy of ML models. While there are many ways to convert these qualities into a numerical format, One Hot Encoding is the most popular and widely used method.

In one hot encoding method, the representation of categorical data is changed into numeric data by splitting the column into multiple columns. The numeric data can be fed into algorithms for deep learning and machine learning. It is a binary vector representation of a categorical variable, with all values in the vector being 0 except for the ith value, which will be 1 and represent the ith category of the variable. The length of the vector is equal to the number of unique categories in the variable.

The encoding of categorical variables requires a collection of finite and gathered categories with elements that are mutually exclusive. A vector of numerical values is used by CNN to represent the input images for every subclass. The one-hot encoding method used in this work denotes the input length as l and the total number of input values as 8×l. The benefit of utilising this method is that it just needs a small amount of time for a large dataset. With fewer computations, it uses fewer memory, though. It is possible to transform a one-dimensional vector (1× l) into a two-dimensional matrix (m × n).

D. CNN

CNN uses neural networks as part of its architecture, which is widely utilised for image-based classification. CNNs include a variety of trainable parameter-based filters that are integrated with the image to recognise characteristics like edges and shapes. The spatial qualities of the image are captured by these high-level filters, which are constructed using learnt weights from the spatial attributes at each subsequent level. Hyper Parameters are a type of parameter that interpret this architecture. These factors have an impact on how the network is trained and how it is structured, both of which are controlled prior to training.

The dataset is used to train and test a CNN with three hidden layers. Three convolutional layers, Maxpooling layers, and a fully connected layer make up the network. A Convolution2D layer serves as the first hidden layer. Following the Maxpooling layer with a pool size of 22, 256 filters have been utilised, each with a kernel size of 33 and zero padding. Padding is used in the architecture, preserving the initial input size. The second convolution layer, which has 256 filters with 33 kernel sizes apiece, is the following phase. This is followed by the Maxpooling layer, which has a 22 pool size. In addition to a Maxpooling layer with a pool size of 22, a third convolution layer with 256 filters, each with a kernel size of 33 and padding and stride equal to 1, has been developed. Before creating the completely connected layers, a flatten layer is utilised to transform the 2D matrix data into a 1D vector. A completely connected layer with the ReLu activation function was then used. Then, to reduce overfitting, a regularisation layer called Dropout is set up to arbitrarily remove 20% of the layer's neurons. Finally, a 16-neuron output layer with a sigmoid activation function is put into practise. A vital component of CNN, the activation function finds a neuron's output based on its inputs. The activation function's function is to add nonlinearity to the model. An effective activation function can enhance a CNN model's performance.

Rectified Linear Unit (ReLu) activation function is used in the suggested model. An activation function called ReLu has substantial biological and mathematical backing. For every negative input, ReLu returns 0, and for positive input, it returns the value. ReLu's max operation allows it to compute more fast than other activation functions. It is frequently the default activation function for a wide variety of neural networks.

$$f(x)=\max(0,x) \quad (1)$$

This is typically applied element-by-element to the result of another function, like the matrix-vector product. Since each layer of the network applies a nonlinearity, CNN needs this function.

E. RMS prop Optimization

Optimizers must be used to increase accuracy and reduce losses. Optimizers modify the weight settings in order to minimise the loss function. The objective of optimisation is to produce the optimal design in regard to a set of prioritised constraints or criteria. Root Mean Square Propagation (RMSprop) is similar to a gradient descent method with momentum in that it restricts oscillations to the vertical direction. As a result, as learning rate is increased, the algorithm progresses significantly in the horizontal direction and quickly converges. It uses the magnitude of the current gradient descents to normalise the gradient. The learning rate is automatically modified by choosing a different learning rate for each parameter. The parameters are updated using the learning rate, that has a value of 0.001, the exponential average of the squares of the gradients, and the gradient at time t.

F. Pseudocode for CNN

```
for from 1 to m do                                          -inter-output
for j from 1 to n do                                        -intra-output
for r from 1 to Ro, do
for c from 1 to Ro, do
tmp = 0
for ii from 1 to k do
for jj from 1 to k do
tmp = tmp + K[ii][jj] × X[j][s × (r − 1) + ii][s × (c − 1) + j
end for
end for
Y [i][r][c] = Y [i][r][c] + tmp
if j == n
Y [i][r][c] = f(Y[i][r][c] + bias)
end if
end for
end for
end for
end for
```

IV. RESULTS AND DISCUSSION

A. Experimental Setup

The dataset used in the proposed experiment is given below.

https://www.kaggle.com/datasets/sachinpatel21/az-handwritten-alphabets-in-csv-format

The dataset comprises 26 files (A-Z), each of which has a handwritten image with a size of 2828 pixels and a box size of 2020 pixels.

Figure 2 illustrates that the sample dataset image.

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	...	0.639	0.640	0.641	0.642	0.643	0.644	0.645	0.646	0.647	0.648
372445	25	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
372446	25	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
372447	25	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
372448	25	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
372449	25	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 785 columns

Fig. 2: Sample dataset

The use of a convolutional neural network model to identify text in academic and financial institution papers is discussed in this research. The model has proved successful in recognising characters in real-time, expanding its use. Preprocessing plays a key role in ensuring that the model performs well. Preprocessing approaches for images improve the features of the image, improving recognition precision. Figure 3 illustrates the pre-processing process.

```
#change the data into "numpy array".
dataset = np.loadtxt('J:/RD/dataset/A_Z Handwritten Data.csv', delimiter=',')
#Divide dataset into  "Explanatory variable", and "Taget".
X = dataset[:,0:784]
Y = dataset[:,0]
#Split the "X,Y" data into the ratio of 7:3, 3 is the test size.
(X_train, X_test, Y_train, Y_test) = train_test_split(X, Y, test_size=0.3, random_state=2)
#Reshape the data and change it into float 32 as usual.
X_train = X_train.reshape(X_train.shape[0], 28, 28, 1).astype('float32')
X_test = X_test.reshape(X_test.shape[0], 28, 28, 1).astype('float32')
```

Fig. 3: Pre-processing

Figure 4 illustrates the feature extraction process done by using one hot encoding technique. The preprocessed features are converted to the format suitale for CNN classifier.

```
# One-Hot-Encoding of the target.
Y_train = np_utils.to_categorical(Y_train)
Y_test = np_utils.to_categorical(Y_test)
# Define the classification of 26 alphabets.
num_classes = Y_test.shape[1]
```

Fig. 4: Feature extraction using one hot encoding technique

The data is normalized. The data is split into training and testing in the ratio of 70:30.

```
#Now normalize the data
X_train = X_train / 255
X_test = X_test / 255
```

Both CNN model with RMSprop and CNN model with Adam optimizer is run to compare the results.

- *CNN model with Adam Optimizer*

```
#Build an ordinary "Deep Learning" model with CNN and maxpooling by using Keras.
model = Sequential()
model.add(Conv2D(32, (5, 5), input_shape=(28, 28, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
#Choose an optimizer and compile the model.
model.compile(optimizer = Adam(learning_rate = 0.01), loss = 'categorical_crossentropy', metrics = ['accuracy'])
#And print the summary of the model.
print(model.summary())
```

➢ *CNN model with rmsprop Optimizer*

```
#Build the second model to Look for best or better models.
# "Adam" is way better than RMSprop (comparing)
model = Sequential()
model.add(Conv2D(32, (5, 5), input_shape=(28, 28, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
print(model.summary())
#Now the second model's optimizer is "RMSprop".
model.compile(optimizer = "rmsprop", loss = 'categorical_crossentropy', metrics = ['accuracy'])
```

B. Performance Analysis

The parameters used in the implementation are given below:

- **Training Loss:** It is a term utilized for assessing how well a deep learning method matches the training set of information.
- **Test Loss:** Test loss is a statistic that is used to assess how well a deep learning algorithm performed on the validation set.
- **Test Accuracy:** The precision of the validation dataset is referred to as test accuracy.
- **Training Accuracy:** The precision of the training dataset is referred to as training accuracy.

Accuracy is estimated as follows.

$$\text{Accuracy} = \frac{T1+T2}{T1+F1+T2+F2} \quad (7)$$

Loss is estimated as follows.

$$\text{Loss} = (T1 + T2) - (F1 + F2) \quad (8)$$

Where T1 - true positive (no. of correctly classified as covid-19), F1 - false positive (no. of images misclassified as covid-19), T2 - True negative (no. of correctly classified normal images), and F2 - False negative (no. of images misclassified as normal). In the majority of deep learning applications, the training loss and validation loss are often mixed on a graph. This is done to assess the model's performance and identify the components that need to be adjusted.

his paper discusses the implementation of Convolutional Neural Network Model to recognize English handwritten characters along with text images. The model has been successfully able to recognize characters in real-time which further increases its scope. Preprocessing is an important factor in ensuring high performance of the model. Image preprocessing techniques enhance the features of the image, thereby increasing the accuracy of recognition. The recognition is shown in Fig.7.

The proposed model was tested with the database given above. The accuracy and loss results are reported in Figures 5 and 6. It has been seen that a nice enhancement in the recognition accuracy was recorded by CNN and RMSprop model. Both testing and training accuracy is 99% in the proposed method. The results were obtained when 70% of the samples were used for training and the rest used for testing. It is also shown that the loss of the proposed model is considerably low.

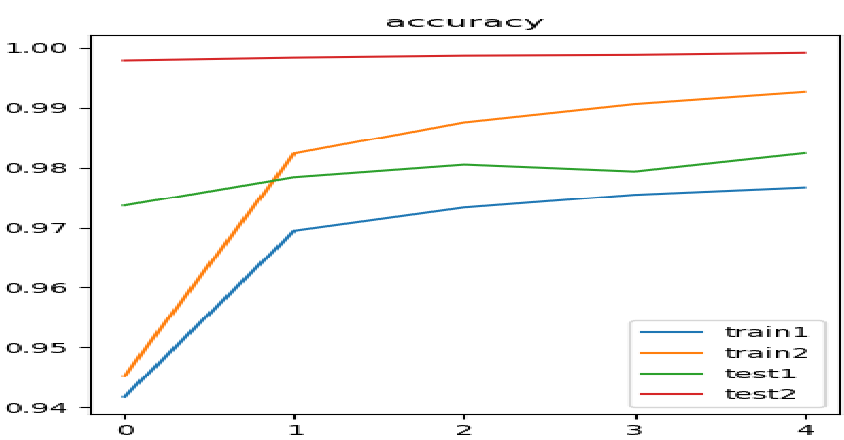

Fig. 5: Accuracy Results

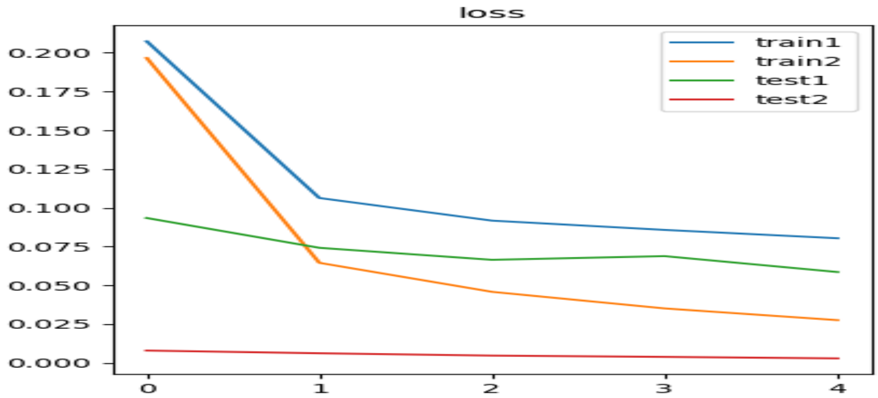

Fig. 6: Loss Results

The prediction results are given in Figure 8. From the figure, it is clear that the proposed model accurately identify the text from the document image.

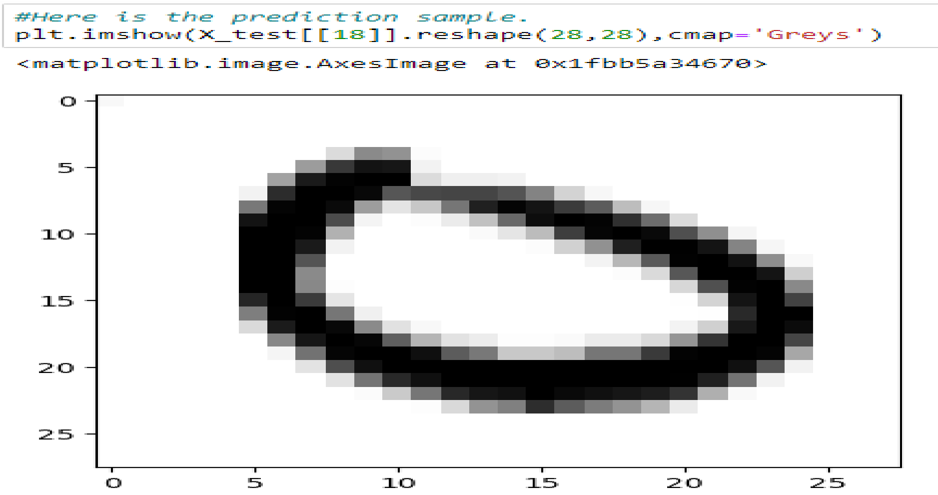

Fig. 7: Output Classified image

The CNN architecture with RMSProp optimizer produced the highest level of recognition accuracy. This network might, however, function more effectively with alternative optimizers. We used the Adam optimizer to conduct experiments to further our investigation. When compared to the Adam optimizer, RMSprop optimizer has more accuracy. The training and testing accuracy of the proposed model is also compared with CNN+LSTM [24], SVM [25], VGG-16 model [26], LSTM and Adaptive Classifier [27], Bi-LSTM [28], and Hybrid PSO-SVM [29]. Compared to all the aforementioned models, the proposed CNN with RMSprop model has the maximum accuracy. From Table 1, it is clear that training and testing accuracy with 'adam' optimizer is 97% and 98%, respectively. Training and testing accuracy with 'rmsprop' optimizer is 99%. The two models are evaluated by using two metrics, loss and accuracy. It is observed that CNN with rmsprop outperformed in terms of accuracy.

V. COMPARATIVE ANALYSIS

Table 1: Comparison of Accuracy

Techniques	Training accuracy	Testing accuracy
CNN + Adam optimizer	97%	98%
CNN + RMSprop optimizer	99%	99%
CNN+LSTM [20]	92.72%	91%
SVM [21]	96.29%	-
VGG-16 model [22]	98.43%	98.16%
LSTM And Adaptive Classifier [23]	98%	98%
Bi-LSTM [24]	96.06%	87.94%
Hybrid PSO-SVM [25]	93%	92.2%

From Table 1, it is clear that training and testing accuracy with 'adam' optimizer is 97% and 98%, respectively. Training and testing accuracy with 'rmsprop' optimizer is 99%. The two models are evaluated by using two metrics, loss and accuracy. It is observed that CNN with rmsprop outperformed in terms of accuracy.

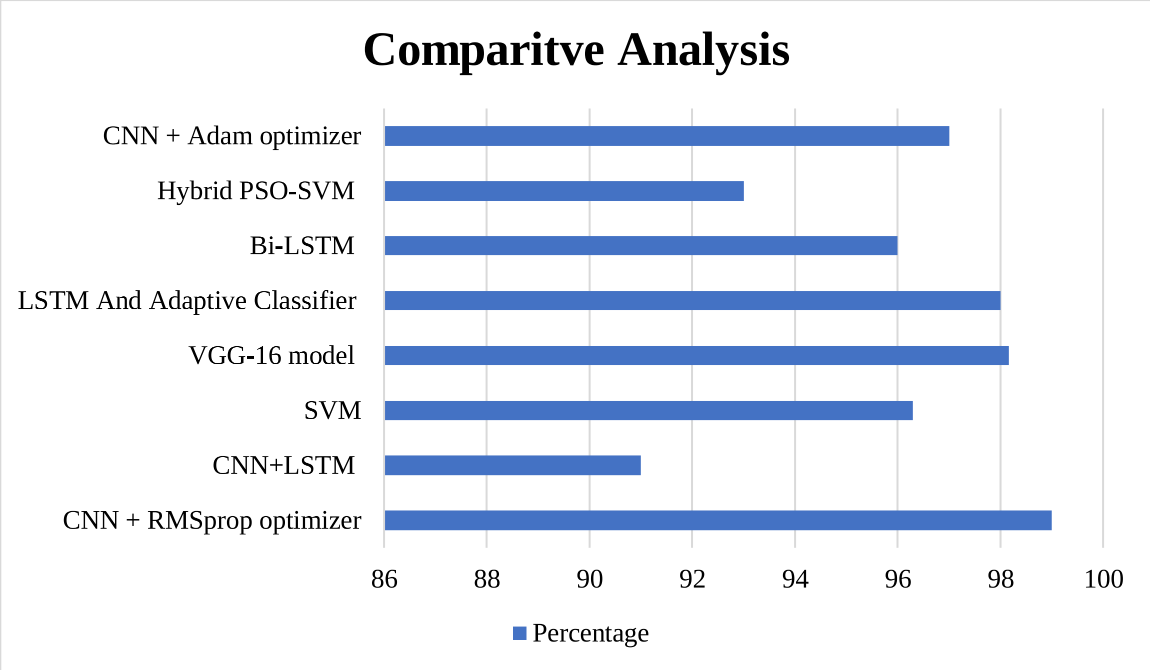

Fig. 8: Accuracy Comparison graph

VI. CONCLUSION

A method for optical character recognition is developed in this article using one-hot encoding-based feature extraction and CNN. At first, the document picture is given a preprocessing treatment in order to get it ready for the feature extraction process. It is decided to carry out one hot encoding-based feature extraction. CNN in two dimensions is utilised to make the final classification of the features. RMSprop is applied in order to perfect the results and make them more accurate. Performance metrics used are training and validation accuracy and loss. Results show that the text in the input image is accurately extracted. As the number of photos that this algorithm was trained and tested on in the local dataset is significantly lower. Because everyone's handwriting is unique, the suggested algorithm will have a poor performance when we test it on different examples of people's writing. In the future, work will be performed to progress the performance of characters that are embedded in environments with complex backgrounds and images that have been distorted.

REFERENCES

[1.] Om Prakash Sharma, M. K. Ghose, Krishna Bikram Shah, "An Improved Zone Based Hybrid Feature Extraction Model for Handwritten Alphabets Recognition Using Euler Number", International Journal of Soft Computing and Engineering, Vol.2, Issue 2, pp. 504-58, May 2012.

[2.] C. C. Tappert, C. Y. Suen, and T. Wakahara, ‘‘The state of the art inonline handwriting recognition,’’ IEEE Trans. Pattern Anal. Mach.Intell.,vol. 12, no. 8, pp. 787–808, Aug. 1990, doi: 10.1109/34.57669

[3.] M. Kumar, S. R. Jindal, M. K. Jindal, and G. S. Lehal, ‘‘Improvedrecognition results of medieval handwritten Gurmukhi manuscripts usingboosting and bagging methodologies,’’ Neural Process. Lett., vol. 50,pp. 43–56, Sep. 2018.

[4.] C. Wolf, J.-M. Jolion, and F. Chassaing, ‘‘Text localization, enhancementand binarization in multimedia documents,’’ in Proc. Object Recognit.Supported User Interact. Service Robots, vol. 2, 2002, pp. 1037–1040.

[5.] J. Pradeep, E. Srinivasan, and S. Himavathi, ‘‘Diagonal based featureextraction for handwritten character recognition system using neuralnetwork,’’ in Proc. 3rd Int. Conf. Electron. Comput. Technol. (ICECT),vol. 4, Apr. 2011, pp. 364–368.

[6.] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber,‘‘Convolutional neural network committees for handwritten characterclassification,’’ in Proc. Int. Conf. Document Anal. Recognit., Sep. 2011,pp. 1135–1139.

[7.] Pritpal Singh and Sumit Budhiraja, “Feature Extraction and Classification Techniques in O.C.R. Systems for Handwritten Gurmukhi Script”. International Journal of Engineering Research and Applications (IJERA), Vol.1, ISSUE 4, pp.1736-1739.

[8.] N. Suguna and Dr. K. Thanushkodi, “An Improved k-Nearest Neighbour Classification Using Genetic Algorithm”, IJCSI, Vol. 7, Issue 4, No 2, July 2010.

[9.] Emanuel Indermuhle, Marcus Liwicki and Horst Bunke, “Recognition of Handwritten Historical Documents: HMM-Adaptation vs. Writer Specific Training”.

[10.] B.V.Dhandra , Gururaj Mukarambi and Mallikarjun Hangarge, “Handwritten Kannada Vowels and English Character Recognition System”, International Journal of Image Processing and Vision Sciences,Vol. 1 Issue1 ,2012.

[11.] S. Antani, L. Agnihotri, Guajarati character recognition, Proceedings of the Fifth International

Conference on Document Analysis and Recognition, 1999, pp. 418–421.
[12.] Noha A. Yousri, Mohamed S. Kamel and Mohamed A. Ismail, “Finding Arbitrary Shaped Clusters for Character Recognition”.
[13.] Faisal Mohammad, Jyoti Anarase, Milan Shingote and Pratik Ghanwat, “Optical Character Recognition Implementation Using Pattern Matching”, International Journal of Computer Science and Information Technologies, Vol. 5(2), 2014.
[14.] Nirmala S Guptha, V. Balamurugan, Geetha Megharaj, Khalid Nazim Abdul Sattar, J. Dhiviya Rose, “Cross lingual handwritten character recognition using long short term memory network with aid of elephant herding optimization algorithm”, Pattern Recognition Letters, Volume 159, Issue C, Jul 2022 pp 16–22.
[15.] Akhtar, Z., Lee, J. W., Attique Khan, M., Sharif, M., Ali Khan, S., & Riaz, N. (Accepted/In press). Optical character recognition (OCR) using partial least square (PLS) based feature reduction: an application to artificial intelligence for biometric identification. Journal of Enterprise Information Management. https://doi.org/10.1108/JEIM-02-2020-0076
[16.] Danish Nadeem and Miss. Saleha Rizvi, “Character Recognition using Template Matching”, Department of computer Science, JMI.
[17.] Rachit Virendra Adhvaryu, “Optical Character Recognition using Template Matching”, International Journal of Computer Science Engineering and Information Technology Research, Vol. 3, Issue 4, Oct 2013.
[18.] M. Ziaratban, K. Faez, F. Faradji, “Language-based feature extraction using template-matching in Farsi/Arabic handwritten numeral recognition”, Ninth International Conference on Document Analysis and Recognition, pp. 297 - 301, 2007.
[19.] Rajib Lochan Das, Binod Kumar Prasad, Goutam Sanyal, "HMM based Offline Handwritten Writer Independent English Character Recognition using Global and Local Feature Extraction", International Journal of Computer Applications (0975 8887), Volume 46 No.10, pp. 45-50, May 2012.