Library Progress International *Print version ISSN 0970 1052*
Vol.44 No.3, Jul-Dec 2024: P.12408-12421 *Online version ISSN 2320 317X*

Original Article *Available online at www.bpasjournals.com*

Leveraging Libraries for a Deep Learning Based Font Recognition Model

Ashish Tripathi[1], Rajnesh Singh[1], Suveg Moudgil[1], Manoj Singhal[2], Girish Kumar Sharma[3], Bhoopendra Dwivedy[1]

[1]SCSE, Galgotias University, Greater Noida, India
[2]Department of Information Technology, G. L. Bajaj Institute of Technology and Management, Greater Noida, India
[3]Computer Science and Application Department, Delhi Skill and Entrepreneurship University, Delhi, India

How to cite this article: Ashish Tripathi, Rajnesh Singh, Suveg Moudgil, Manoj Singhal, Girish Kumar Sharma, Bhoopendra Dwivedy (2024 Leveraging Libraries for a Deep Learning Based Font Recognition Model. *Library Progress International,* 44(3), 12408-12421.

ABSTRACT

Motivation:
Several progresses have been made in the field of text recognition/Optical Character Recognition (OCR) in recent years, but the task of font recognition remains a significant challenge.

ProblemStatement: With a variety of fonts available in numerous shapes and styles, and several fonts with only minor differences in characteristics, font identification becomes complex. The most difficult part is distinguishing the small differences between these similar fonts. Conventional OCR text recognition systems exhibit low recognition rates and several errors, struggling to differentiate between different text fonts. This has made font recognition a persistent challenge. Modern text recognition systems should be able to recognize text and font information across a variety of fonts with high accuracy.

Approach/Method:
The proposed work aims to precisely recognize font shapes and styles of text in images by employing a method based on Convolutional Neural Networks (CNN). This approach is tuned for multi-classification and involves two primary steps. The first step involves splitting the dataset in a 3:1 ratio for training and testing purposes, followed by the training of the CNN-based model. The second step involves feeding the model an image to identify the font. The dataset comprises five classes: Lato, Raleway, Roboto, Sansation, and Walkway, with 100 randomly generated string images per class.

Results and Findings:

The final model achieves an accuracy of 98.93%, with improved precision, recall, and F1 score.

Conclusion:
The proposed CNN-based method demonstrates a high accuracy in font recognition, addressing the challenges of distinguishing between similar fonts and improving the performance of text recognition systems.

Keywords: Font Recognition **,** Convolutional Neural Network (CNN)**,**Optical Character Recognition (OCR)**,**Automatic Font Identification

1. INTRODUCTION

Text is all around us in daily life. Text can be found in books, billboards, and other items like cars and phones. Systems that can be used for a range of challenging tasks, such as image-based machine translation, autonomous cars, and image/video indexing, require the ability to automatically recognize and understand the text from images of natural scenes. The computer vision and document analysis communities have recently become interested in the problem of discovering and identifying text in natural settings. Although optical

character recognition (OCR) for printed documents has been made easier, recognizing text or phrases in images of natural surroundings is still a challenge.

Unlike more traditional textual challenges, where the characters are usually monotone and on a fixed background, Natural-scene images have a wide range of lighting, perspective distortions, image quality, text fonts, and backdrops, among other things. As a result, creating entire systems for these scenarios involves the creation of representations that account for all of these variances. Such systems have taken a lot of work to develop, with top performers incorporating dozens of carefully combined features and processing stages. Recent deep learning research has aimed to develop algorithms that can automatically learn higher level data representations for a variety of tasks. Such systems could be especially useful in situations when specialized characteristics are required yet are difficult to construct by hand. Another potential advantage of these approaches is that we can simply generate huge quantities of features, allowing classification algorithms to attain higher performance. In this research, we'll employ one of these feature learning systems to see how effective these algorithms are at detecting scene text. In other areas, feature learning algorithms have had a series of successes (for example, obtaining excellent performance in visual and audio recognition). Unfortunately, one disadvantage is that these methods are frequently too computationally expensive and especially for large image applications. This work takes on the well-known image-based sequence recognition problem in computer vision. Many visual elements, such as musical notation, handwriting, and scene text usually appear in sequence rather than separately in the real world.

The system often has to predict multiple item labels rather than simply one in order to locate the sequence-like things, in contrast to standard object recognition. Thus, it is natural to frame the problem of object identification as one of sequence recognition. The fact that sequence-like items can come in a wide range of durations is another distinctive quality. English words, for instance, can range in length from two letters, like "hi," to fourteen, like "understandable."

A subset of deep neural networks called recurrent neural network (RNN) models was developed specifically to handle sequences. The advantage of RNN is that it does not need to be aware of the location of each element in a series of object images, which is useful for both training and testing. On the other hand, preprocessing is frequently needed to convert an input object image into a series of image features. The main contribution of the study is a novel neural network model with a network architecture designed for identifying sequence-like objects in images. The proposed neural network model is known as a Convolutional Recurrent Neural Network (CRNN) since it combines Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). For objects that resemble sequences, CRNN offers a number of significant advantages over conventional neural network models: 1) Without the requirement for detailed annotations (such as characters), it can be deduced directly from sequence labels (for instance, words); 2) Without the requirement for manually created features or preprocessing procedures like binarization/segmentation, component localization, etc., it can learn meaningful representations straight from the input image; 3) It can generate a series of labels with the same efficiency as an RNN; 4) It is not bound by the lengths of items that belong in a sequence and only needs height normalization during training and testing; 5) When it comes to scene text (word recognition), it outperforms or is competitive with earlier arts. 6) It requires less storage space than a typical CNN model because it has fewer parameters.

Typography is essential for Graphic Designing. There is always a need for graphics designers to recognize the fonts efficiently in their daily work to get the optimal result. The self-recognition process of those fonts is time-consuming and prone to errors. So, the designers may take a picture of the particular font and seek experts to recognize the font. Several websites that help users search for and identify fonts based on font resemblance are Identifont [1], MyFonts [2], and Fontspring [3]. However, the accuracy of these websites is poor and all of them require time-consuming human interactions and excellent manual image pre-processing. The aforementioned problems might be considerably reduced by choosing and categorizing fonts during the design process. Also, the design process could be made much easier with automated font recognition from an image or photo. According to [5], the Visual Font Recognition (VFR) problem is inherently challenging due to a large number of accessible fonts (hundreds of thousands are available online), the dynamic and open-ended characteristics of font classes, and the subtle and character-dependent variations among fonts, such as weights, slopes, and the letter ends. Furthermore, while the majority of machine learning algorithms are data-driven, it has proven to be very difficult to collect data from real-word for a variety of fonts of different classes. The majority of text images in the real-world lack information related to the font, and font labeling is a laborious task that

requires deep knowledge of many fonts, which most individuals lack.

Deep artificial neural networks or convolutional neural networks can be used to classify images, cluster them based on similarities and recognize objects within scenes. It may be used to recognize faces, people, street signs, cancers, cats & dogs, and a variety of other visual data. The convolutional layer is the foundation of a CNN. The learnable filters (or kernels) that make up the layer's parameters have a limited receptive field but cover the entire depth of the input volume. The proposed model makes use of a CNN as well as many picture-enhancing methods, such as shading, noise, variable character spacing, perspective rotation, blur, and variable aspect ratio. The CNN has been fine-tuned and optimized, which has led to a brief training period for the dataset.

Problem Statement:

Font identification is complex due to the variety of fonts with minor differences in characteristics. Conventional OCR systems exhibit low recognition rates and numerous errors in distinguishing between different text fonts, making font recognition a persistent challenge. High accuracy in recognizing text and font information across various fonts is essential for modern text recognition systems.

Research Contribution

This study proposes a novel approach to font recognition using a Convolutional Neural Network (CNN) tailored for multi-classification tasks. Unlike previous works, our method focuses explicitly on distinguishing between similar fonts with high accuracy. We introduce a comprehensive dataset comprising five font classes (Lato, Raleway, Roboto, Sansation, and Walkway), augmented to enhance model robustness. The CNN model is designed and trained to achieve superior performance in font classification, addressing the limitations of existing systems.

1.1. Overview of the Proposed Work

The suggested model is built on a convolutional neural network (CNN). The model's layers and parameters have been fine-tuned so that it requires very little time to train on the dataset and achieves the highest level of accuracy quickly. The fundamental CNN model has been tweaked by removing layers that were no longer needed and were wasting valuable time. Activation functions, such as ReLU, and additional layers are used to extract important features reliably and effectively without overfitting the model. With the use of max pooling, up-sampling, and dropout layers, the proposed method is refined and optimized to achieve an accuracy of 99.79 %. To understand the loss and accuracy of the suggested strategy, the number of distinct layers has been extensively examined when training data. Convolutional layers, up-sampling layers, max-pooling layers, activation functions, dense layers, and dropout layers have all been optimized to reduce training losses. Additionally, unnecessary layers and functions are deleted, resulting in a reduction in the method's overall execution time without compromising its accuracy. The primary goal of the suggested approach is to recognize the font type which is input by the user with maximum accuracy. Augmentation steps are performed on the loaded dataset. The augmentation steps include:

i. **Noise:** The image from the dataset has noise added to it, which is the unpredictability of an image's brightness or colour.
ii. **Blur:** Blur a picture's blurring, or smoothing, removes "outlier" pixels that could indicate noise in the image. Applying a low-pass filter to an image is an example of blurring.
iii. **Perspective Rotation:** Rotation of image by different angles.
iv. **Shading:** Shading is used to indicate a decrease in image brightness from the centre to the corners, as well as a colour fluctuation across the imaging field, which is common in cameras with small sensors. Shading, as a result, has an impact on image quality by introducing undesirable dark or shaded edges.

Following the augmentation procedures, the dataset is divided into training data and testing data. The dataset divided for training is 75% and for testing is 25%. After this feature extraction is done from the dataset and the model weight is saved. Then classification of the fonts is done on the basis of the saved model weight. The classification is done for the following fonts: Lato, Raleway, Roboto, Sansation, and Walkway.

A wide variety of scene text detection and identification algorithms have been created and published over the years. End-to-end scene text recognition is almost always done in two steps. The first step is to identify and

extract text portions from the input image. Recognizing the textual content and retrieving the extracted text strings from these extracted text sections make up the second stage. Moreover, these methods can be divided into three major groups: (1) Systems for text detection and identification that depend on custom features and human comprehension. (2) Systems that integrate handmade features with deep learning, or two distinct deep networks for the two methods. (3) Systems that identify and recognize text using a single deep neural network rather than in two steps.

To solve the visual font recognition problem, several models have been proposed by researchers on time. In this order, Chen et al. (2014), suggested a scalable method (NCM), that uses the nearest class mean classifier as the base [4]. This method includes the combination of max-margin template selection, local feature metric learning, and local feature embedding to solve the open-ended classification challenges.

The cost of generalizing the new algorithm for new classes and data is minimal. The testing demonstrates that the proposed technique performs admirably on artificial test photos and produces encouraging outcomes on actual test images. Avilés-Cruz et al. [5] (2005) developed a novel optical font recognition method based on global texture analysis, which makes use of statistical methods to identify and classify the font features. The first step in the font recognition procedure is to view the document as a plain picture with one or more different types of fonts. The fourth and third order moments, along with a window analysis, are used by the proposed technique to recover the document's features. Unlike earlier techniques, the identification is not carried out letter by letter. The new method is content independent because it does not call for a study of regional typography.

Eight different kinds of fonts often used in the Spanish language were thoroughly examined. There are four styles for each font type, for a total of 32 font combinations. Font recognition is 100% accurate with clear pictures. Due to the representational strength of massive, multilayer neural networks and recent advancements in unsupervised feature learning, Wang et al. [6] (2012) proposed a system that can train extraordinarily good text detector and character recognizer modules with a similar architecture.

Two modules are combined into a complete, lexicon-driven scene text recognition system that uses readily available techniques to achieve state-of-the-art performance on recognized benchmarks like Street View Text and ICDAR 2003. An end-to-end stacked conditional GAN model that takes into account content, channels, and style with network layers was proposed by Azadi et al. [7] in 2018. The suggested network captures highly styled typefaces observed in actual scenic photographs by transferring the style of the given glyphs to the contents of unseen ones.

Haraguchi et al. [8] (2020) suggested a two-stream CNN-based approach. This approach helps to determine whether the two characters are from the same font or from distinct fonts. The network is trained on 10,000 fonts in various styles. This technique predicts the shape of the fonts using a small number of example images as an input stack. In this work, one network is trained to identify the shape and another network is used to identify the texture of the font.

A technique to improve performance on font recognition with similar-looking categories was put forth by Chen et al. [9] in 2021. In this work, Grad-CAM is used to separate out some of the instance-wise characteristics that contribute to the classifications, and principle component analysis (PCA) is used to demonstrate how it is simpler to distinguish between fonts with comparable convolutional characteristics. They also applied six-fold cross-validation for better accuracy. The model gained 92.27 accuracy in identifying fonts.

Wang et al. [10] proposed a novel technique for precise font identification in real-world images that was created in 2018. They suggested “HE Block” in their proposed work to turn off the readily available function, enabling the entire network to take into account and pick up on more intricate stroke data. Additionally, the HE Block has a training phase and an inference phase, and it may be used in both phases without adding any extra processing overhead.

The key benefit of the suggested model is the avoidance of time-consuming and expensive font annotations for photos in the training dataset. The model is built on CNN, and the implementation of the transfer learning method uses unlabelled data to make better use of the classifier's performance in identifying typefaces in real-world photos.

The rest of the paper discusses motivation and back-ground, which are discussed in section 2, Design and implementation of the proposed model in Section 3. Section 4 includes the result and discussion. Section 5 presents the conclusion and future scope.

2. METHODOLOGY

2.1 Dataset

The dataset for the proposed model is custom-generated and for this, a text recognition data generator has been used. In the work, five types of font classes are chosen and then images are generated per class for the dataset. Random strings of text are generated per image using the same font as per the class. The dataset is divided into a ratio of 3:1 i.e., the data are used for testing purposes in 25% of cases and for training purposes in 75% of cases. The total number of images used is 156 for four different class and fifth class with 160 sample of which are in JPG format and occupy a system space of 9.82 MB. There are 5 different classes of fonts which are in order from the first class i.e., Lato, Raleway, Roboto, Sansation, and Walkway. Each class category contains 20 images. Table 1 shows the dataset distribution [18][19].

Table 1. Dataset distribution

Font Class	Name	Number of Images
0	Lato	156
1	Rateway	156
2	Roboto	156
3	Sansation	156
4	Walkway	160
Total		**784**

Figure 3 Shows the sample image of each font. Each image in the dataset is a randomly generated string from the fonts which are used for classification. The model is then trained on these images and an unseen image is used to predict. Table 2 displays the image sizes included in the dataset. The dimensions are specified in pixels. The classes' minimum and maximum heights are set at 105 pixels. For Class 0 (Lato), the minimum and maximum widths are 86 and 604 pixels, respectively. Class 1 (Raleway) images have widths that are a minimum of 111 pixels and a maximum of 651 pixels [20][21].

Figure 3. Font classes

The minimum and maximum image widths for Class 2 (Roboto) are 110 and 530 pixels, respectively. For Class 3 (Sansation), the minimum and maximum widths are 84 and 638 pixels, respectively. While the maximum and minimum values of width taken for class 4 (Walkway) is 101 and 810 respectively.

Table 2. Font dimensions

1. Classes	2. Width		3. Height	
	4. Minimum Width	5. Maximum Width	6. Minimum Height	7. Maximum Height
8. Lato	9. 86	10. 604	11. 105	12. 105
13. Raleway	14. 111	15. 651	16. 105	17. 105
18. Roboto	19. 110	20. 530	21. 105	22. 105
23. Sansation	24. 84	25. 638	26. 105	27. 105
28. Walkway	29. 101	30. 810	31. 105	32. 105

2.2. Proposed Methodology

2.2.1. Architecture of the Proposed Model

Five Conv2D layers and two Conv2D Transpose layers make up the proposed model's layers. In Figure 4., mnemonic codes have been used which show the arrangement of layers of the proposed model. The description of the mnemonic codes is shown in Table 3 [22][23][24][25].

CO1-BN-RL-MP1-CO2-BN-RL-MP2-COT1-RL-UP1-COT2-RL-UP2-CO3-RL-CO4-RL-CO5-RL-FLA-DEN-DRO-DEN-DRO-DEN-DEN-SO

Table 3. Mnemonics and descriptions

Mnemonic Code	Description
RL	ReLU Activation Function
MP	Maximum Pooling
COT	Convolutional Transpose Layer
UP	Upsampling Layer
CO	Convolutional Layer
DEN	Dense Layer
DRO	Dropout Layer
SO	Softmax Activation Function
FLA	Flatten Layer

The proposed CNN model takes an input of 105 x 105 pixels, where the height and width of the image are 105px and 105px. The input image is passed to a Convolutional – 2D layer with ReLU activation function enabled. The Conv2D layer has a kernel size of 48 x 48 with 64 filters. Batch normalization is also enabled with the layer. The input is then passed through a MaxPool2D layer with a pool size of 2 x 2. Then it is passed through the second Convolutional – 2D layer with a kernel size of 24 x 24, this layer also has ReLU with Batch Normalization enabled. After this, the input goes through a Convolutional – 2D Transpose layer with ReLU enabled with a kernel size of 24 x 24 and with 128 filters.

In Figure 4, the Conv2D Transpose layer is used to perform deconvolution operation on the input. This layer also has UpSampling 2D with the size of 2 x 2 which is used to take 2D inputs. Then the input is passed again on the same configuration of layers (Conv2DT + ReLU and UpSampling 2D) but with a kernel size of 12 x 12 and 64 filters. Then, three times Conv2D layers with ReLU activation functions are used consecutively with a kernel size of 12 x 12 and filter size of 256 in each of the layers. Finally, Dense layers are applied to the stacked layers. The number of units in the first dense layer is 4096 with a Dropout value of 0.5 and a ReLU function.

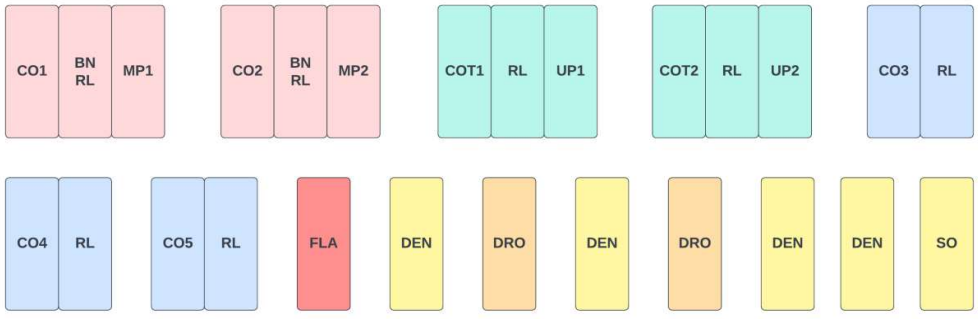

Figure 4. Sequence wise layers of the proposed model

After this, again a dense layer is applied with 4096 units with the same dropout value and a ReLU function. The third dense layer has 2383 units and the final dense layer has 5 units, which is nothing but the number of classes used for the classification. Stacking all these layers together gives a Neural Network which is used to predict inputs.

2.2.2 Proposed Model: Layers and Parameters

In the proposed work, the output shape of the CNN is formed by a convolutional layer that uses a 4D array with the same batch size as the input batch size. The padding, filter, and kernel size parameters that were selected may alter the image's other three dimensions. The second column of the table contains a depiction of the output shape. Height, width, and output filters make up the majority of an output shape's dimensions. Weights and biases are the two types of parameters that each layer in this proposed model contains. All weights and biases together make up the total number of trainable parameters displayed in column three.

2.2.3. Proposed Algorithm

Algorithm 1: Training

1. *def* noise_image()
2. *def* blur_image()
3. *def* affline_rotation()
4. *def* gradient_fill()
5. **Set** path for the dataset
6. *def* conv_label(label):
   if label = 'Lato'
   return 0
   else if label = 'Raleway'
   return 1
   else if label = 'Roboto'
   return 2
   else if label = 'Sansation'
   return 3
   else if label = 'Walway'
   return 4
7. *def* create model ():

   Apply

   Convolutional 2D operation + batch_normalization + ReLU
   Max-pooling
   Convolutional 2D operation + batch_normalization + ReLU
   Max-pooling
   Convolutional 2DTranspose operation + ReLU
   Upsampling2D
   Convolutional 2DTranspose operation + ReLU
   Upsampling2D
   Convolutional 2D operation + ReLU
   Convolutional 2D operation + ReLU
   Convolutional 2D operation + ReLU
   Flatten
   Dense
   Dropout
   Dense
   Dropout
   Dense
   Dense
   Softmax
8. Model Training:
   model.fit(trainX, trainY, validation_data = (testX, testY))
9. Output

Algorithm 2: Classification (Input Image)

1. **Set** input image path
2. Output ← model weight
3. Classification of font ← Output

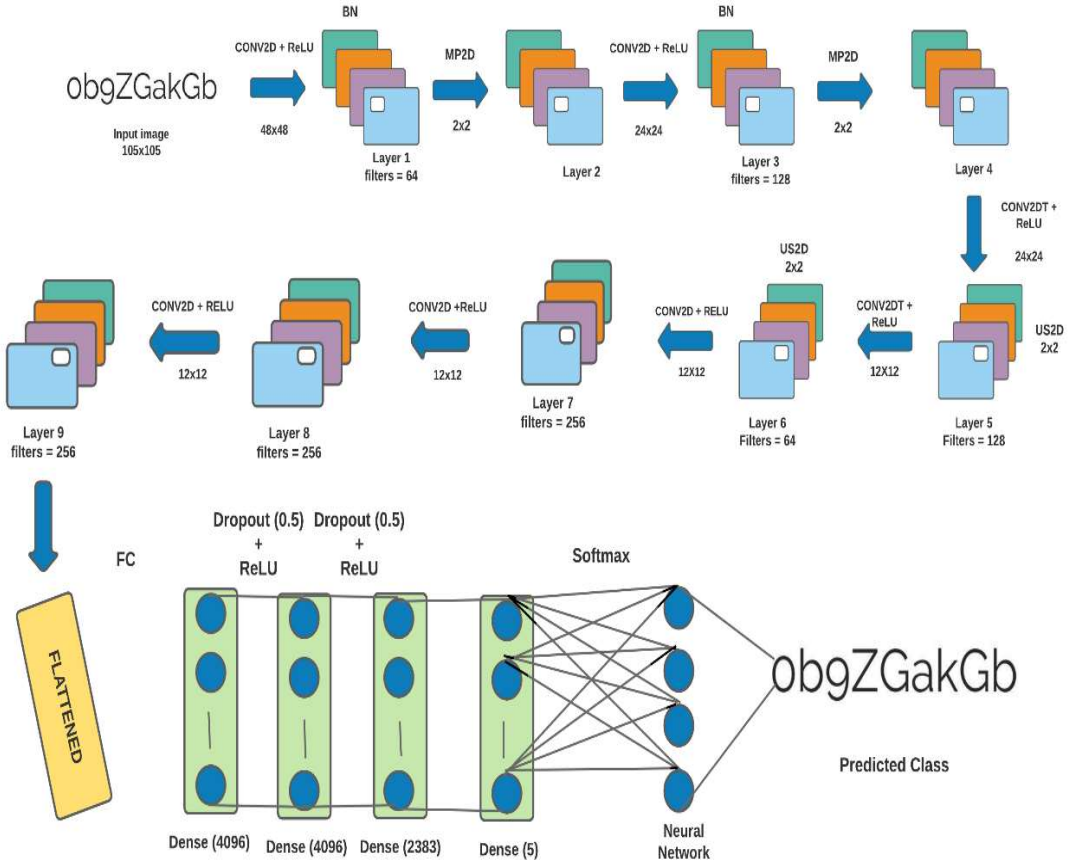

Figure 5. Architecture of the proposed model

3. EXPERIMENT AND ANALYSIS OF RESULT

Figure 6 displays a plot of accuracy over training epochs on the datasets used for training and validation. In the graph, the X-axis is used to show the number of epochs and Y-axis displays the accuracy value. In Figure 7, the loss over training epochs for the training and validation datasets is displayed. The graph's X and Y axes each reflect a different aspect of the data: the number of epochs and the magnitude of the loss, respectively. Model loss is observed to be 0.0038% at 36 epochs with a validation loss of 0.0122%. The overall model and validation accuracy achieved by the proposed model are 98.93%, is 95.95% respectively.

3.1 Confusion Matrix

Since a model's output can be two or more classes, a confusion matrix is a performance-tracking tool that is widely employed in machine learning classification tasks (i.e. binary classification and multiclass classification). The rows of the confusion matrix show actual class instances, while its columns show predicted class instances.

It is also feasible to have columns for actual classes and rows for expected classes. The confusion matrix is useful for assessing the recall, precision, specificity, accuracy, and Area under the ROC Curve (AUC) of a classification model. The name "conflict matrix" (cm) alludes to the fact that it makes it possible for us to rapidly identify the many kinds of mistakes that exist in our classification schemes. Because the correct class is ci, the algorithms should have predicted a sample as ci, instead of cj. In this case of mislabelling, the element cm [i, j] will be increased by one when the confusion matrix is constructed.

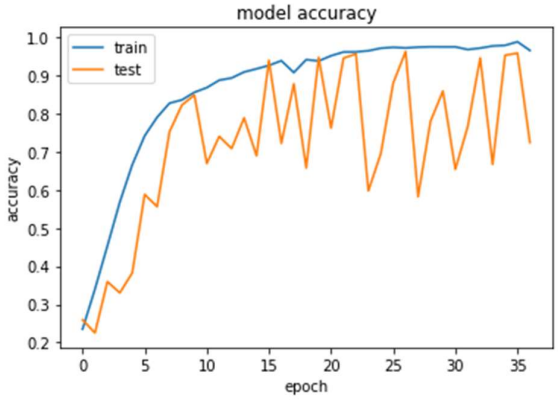

Figure 6. Model accuracy

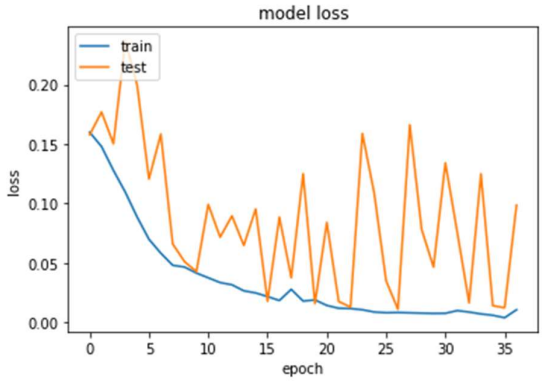

Figure 7. Model loss

3.1.1 Binary Class

The objective of binary classification is to use a classification rule to separate a set of elements into two categories. In the confusion matrix, the negative and positive values represent the binary classification.
The binary or "negative" and "positive" values for the confusion matrix given as follows:

		Predicted	
		Negative	**Positive**
Actual	**Negative**	95	2
	Positive	3	96

The matrix's fields signify the following:

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative (TN)	False Positive (FP)
	Positive	False Negative (FN)	True Positive (TP)

- Positive (P): An observation that is favourable (for example: it is an orange).
- Negative (N) - An observation that is not favourable (for example: it is not an orange).
- True Positive (TP) - The observation and prediction are both positive.
- False Negative (FN) – Positive observation but projected negative result.

- True Negative (TN) - The observation and prediction are both negative.
- False Positive (FP): When an observation is negative but a positive result is predicted.

3.1.2 Multiclass

A confusion matrix that takes into account various classes (i.e. more than two classes). It describes the functionality of a multi-class classification model, much like the binary class confusion matrix does. Figure 8 displays the Multi-Class Confusion matrix for the suggested model. A 5 x 5 confusion matrix has been created because the suggested method uses five classes. The amount of accurate predictions made using the suggested technique is represented by the diagonal cells of the confusion matrix. The inaccurate predictions of the proposed technique are represented by the non-diagonal cells.

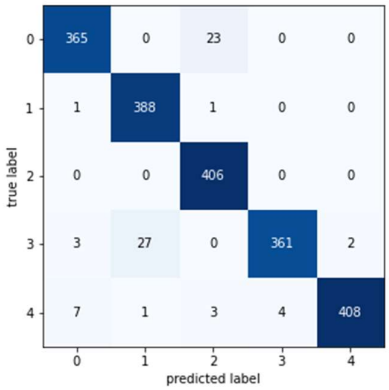

Figure 8. Multiclass Confusion Matrix

3.2 Result Analysis

Table 4 displays the classification report for the suggested model and the graph for the same is shown in Figure 9. The report includes the values for precision, recall, f1 score, and support on five font classes. For class 0 (Lato), the values of precision, recall, and F1-score are 0.97, 0.94, and 0.96 respectively. In the case of Class 1 (Raleway), the model has obtained the values 0.93, 0.99, and 0.96 for precision, recall, and F1-score respectively. Results obtained for class 2 (Roboto) are quite impressive, especially for recall and support where the recall and support values are 1.00 and 406. Also, the precision (0.94) and F1-score (0.97) values are also good. For class 3 (Sansation), the values received are 0.99, 0.92, and 0.95 respectively for precision, recall, and F1-score. Precision, recall, and f1 score values for Class 4 (Walkway) are recorded as 1.00, 0.96, and 0.98 respectively. For all classes support values are recorded as 388, 390, 406, 393, and 423 respectively. Overall the results are quite good and show the strength of the model. The accuracy and average obtained by the proposed model are very impressive and help in the recognition of the fonts in an efficient way.

Table 4. Classification report

Font Class	**Precision**	**Recall**	**F1-Score**	**Support**
Lato	0.97	0.94	0.96	388
Rateway	0.93	0.99	0.96	390
Roboto	0.94	1.00	0.97	406
Sansation	0.99	0.92	0.95	393
Walkway	1.00	0.96	0.98	423
Accuracy			0.98	2000
Macro Average	0.97	0.96	0.97	2000
Weighted Average	0.97	0.96	0.97	2000

3.3 Comparative Analysis

This section shows the comparison between the proposed work and recent research work done for font recognition. Mostly the researchers have used Convolutional Neural Networks and Support Vector Machines as

a base algorithm for their research work and model building. Table 5 shows the comparison between the suggested technique (proposed work) and the other state-of-the-art work based on the size of the data, and accuracy. Also, the visualization of the comparative result is shown in Figure 10.

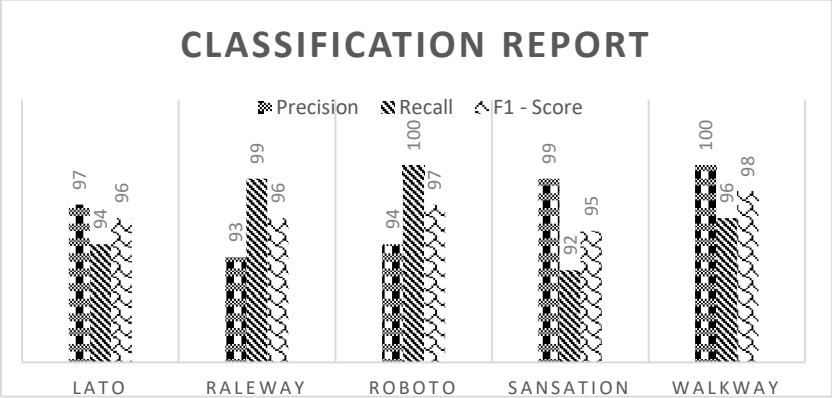

Figure 9. Classification report

Table 5. Comparative analysis with other state-of-the-art models

Author	Model/Approach	Size of Data	Accuracy
Hasan et al. [12]	VGG16	33800	96.23%
Ramanathan et al.[13]	SVM	216	93.54%
Wang et al. [14]	Transfer Learning	200000	93.97%
Mohammed et al. [16]	SVM+RBF	27620	94.82%
Proposed Work	CNN	784	98.93%
Hasan et al. [12]	VGG16	784	94.69%
Ramanathan et al.[13]	SVM	784	93.13%
Wang et al. [14]	Transfer Learning	784	92.91%
Mohammed et al. [16]	SVM+RBF	784	88.93%

In the first paper, the authors gained 96.23% accuracy using transfer learning [12]. They did their research work on Bangla fonts with the size of the dataset being 33800. They proposed a hypothesis based on the VGG16 model for font recognition. In paper [13], Ramanathan et al. used English font in their experimental work. The authors used a total of 216 fonts in their dataset. The proposed model was based on Support Vector Machine (SVM) and gained 93.54% accuracy. Paper [14] shows the research work on Chinese and English fonts. The authors of this paper took the font dataset with a size of 200000 and proposed a transfer learning-based model for the recognition of fonts. They found 93.97% accuracy in their work. In another work, Mohammed et al. [16] developed a model based on SVM with Radial Basis Function (RBF) kernel and provided 94.82% accuracy. They focused on English fonts and used a dataset with a size of 27620 lines of fonts.

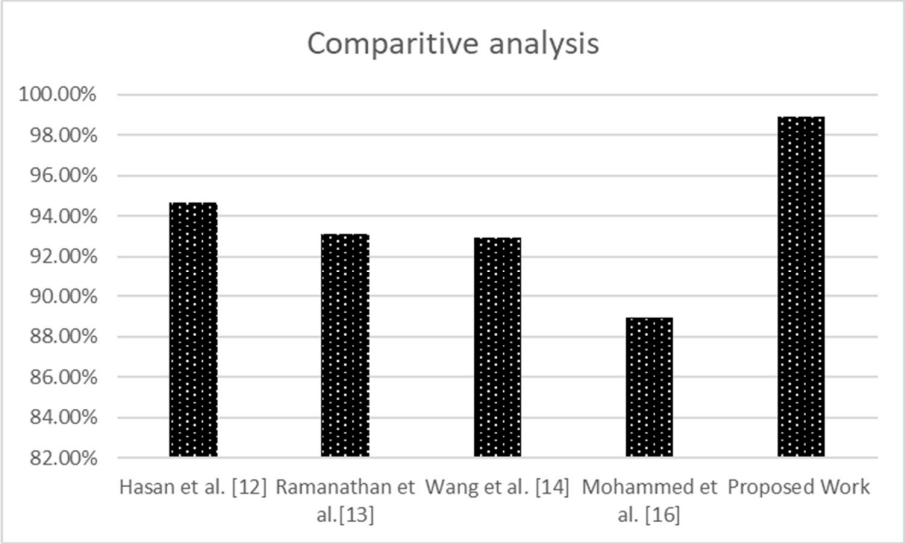

Figure 10. Graphical representation of comparative analysis

In this paper, the suggested model has obtained 98.93% accuracy which comparatively gives higher accuracy as compared to the state-of-the-models shown in table 5. The proposed work employs a Convolutional Neural Network (CNN) for font recognition, achieving an impressive accuracy of 98.93%. This result significantly surpasses the performance of several existing methods in the field. we have implemented state of art work on our dataset. Hasan et al. [12] used the VGG16 architecture and reported an accuracy of 94.69%, while Ramanathan et al. [13] utilized a Support Vector Machine (SVM) achieving 93.13%. Wang et al. [14] applied Transfer Learning techniques and obtained 92.91% accuracy, and Mohammed et al. [16] combined SVM with a Radial Basis Function (RBF) kernel, reaching 88.93%. The superior performance of the proposed CNN-based method highlights its effectiveness for precise font recognition, demonstrating the advantages of custom CNN architectures tailored for specific classification tasks.

.

4. CONCLUSION AND FUTURE SCOPE

Font recognition is a noteworthy work with significant utility when accurately utilized. By analyzing input images using various augmentation techniques like noise, blur, perspective rotation, and shading, the proposed work, based on a modified Convolutional Neural Network (CNN), achieves a final accuracy of 98.93%. The model was trained over 36 epochs, achieving a minimal loss of 0.0038% and a validation loss of 0.0122%, with a validation accuracy of 95.95%. These results indicate that the model is optimized for maximum accuracy with minimal training time and loss. Font recognition can be extended to train and identify many fonts, aiding users in recognizing fonts in different languages such as English, French, and Chinese. The model's high accuracy of 98.93% suggests that with more classes and datasets, its performance can be further enhanced. Implementing the model in an app or web app could facilitate quick font identification for users. Our findings demonstrate that a well-designed CNN can achieve high accuracy in font recognition, significantly improving existing methods. This has several implications: enhancing OCR systems by improving accuracy and reliability, particularly for applications requiring precise font identification like digital archiving and document verification; aiding custom font applications in graphic design, typesetting, and branding; and opening avenues for further research into more extensive datasets, additional font classes, and combining font recognition with other text recognition tasks.

REFERENCE

[1] Typography, Source: https://en.wikipedia.org/wiki/Typography
[2] Identifont, Source: http://www.identifont.com/
[3] Myfonts, Source: https://www.myfonts.com/
[4] G. Chen, J. Yang, H. Jin, J. Brandt, E. Shechtman, A. Agarwala, and T. X. Han. Large-scale visual font recognition. IEEE, 2014.

[5] C. Avilés-Cruz, R. Rangel-Kuoppa, M. Reyes-Ayala, A. Andrade-Gonzalez, and R. Escarela-Perez. High-order statistical texture analysis: font recognition applied. PRL, 26(2):135{145, 2005.

[6] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In ICPR, pages 3304{3308. IEEE, 2012.

[7] Azadi, Samaneh et al. "Multi-content GAN for Few-Shot Font Style Transfer." 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018): 7564-7573.

[8] Haraguchi, Daichi & Harada, Shota & Iwana, Brian & Shinahara, Yuto & Uchida, Seiichi. (2020). Character-Independent Font Identification. 10.1007/978-3-030-57058-3_35.

[9] Chen, Jingchao & Mu, Shiyi & Xu, Shugong & Ding, Youdong. (2021). HENet: Forcing a Network to Think More for Font Recognition. 1-5. 10.1145/3503047.3503055.

[10] Wang, Yizhi & Lian, Zhouhui & Tang, Yingmin & Xiao, Jianguo. (2018). Font Recognition in Natural Images via Transfer Learning. 10.1007/978-3-319-73603-7_19.

[11] https://github.com/Belval/TextRecognitionDataGenerator.

[12] Hasan, S., Rabbi, G., Islam, R., Bijoy, H.I. and Hakim, A., 2022, July. Bangla Font Recognition using Transfer Learning Method. In 2022 International Conference on Inventive Computation Technologies (ICICT) (pp. 57-62). IEEE.

[13] Ramanathan, R., Soman, K.P., Thaneshwaran, L., Viknesh, V., Arunkumar, T. and Yuvaraj, P., 2009, October. A novel technique for english font recognition using support vector machines. In 2009 international conference on advances in recent technologies in communication and computing (pp. 766-769). IEEE.

[14] Wang, Y., Lian, Z., Tang, Y. and Xiao, J., 2018. Font recognition in natural images via transfer learning. In MultiMedia Modeling: 24th International Conference, MMM 2018, Bangkok, Thailand, February 5-7, 2018, Proceedings, Part I 24 (pp. 229-240). Springer International Publishing.

[15] Bychkov, O., Merkulova, K., Dimitrov, G., Zhabska, Y., Kostadinova, I., Petrova, P., Petrov, P., Getova, I. and Panayotova, G., 2020, September. Using neural networks application for the font recognition task solution. In 2020 55th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST) (pp. 167-170). IEEE.

[16] Mohammed, A.J. and Al-Khaffaf, H.S., 2020. Font recognition of English letters based on distance profile features. Science Journal of University of Zakho, 8(2), pp.66-71.

[17] Chen, J., Mu, S., Xu, S. and Ding, Y., 2021, November. HENet: Forcing a Network to Think More for Font Recognition. In 2021 3rd International Conference on Advanced Information Science and System (AISS 2021) (pp. 1-5).

[18] Chaturvedi, Pooja, Ajai Kumar Daniel, and Vipul Narayan. "Coverage prediction for target coverage in WSN using machine learning approaches." Wireless Personal Communications 137.2 (2024): 931-950.

[19] Narayan, Vipul, et al. "A theoretical analysis of simple retrieval engine." Computational Intelligence in the Industry 4.0. CRC Press, 2024. 240-248.

[20] Narayan, Vipul, et al. "A comparison between nonlinear mapping and high-resolution image." Computational Intelligence in the Industry 4.0. CRC Press, 2024. 153-160.

[21] Sandhu, Ramandeep, et al. "Enhancement in performance of cloud computing task scheduling using optimization strategies." Cluster Computing (2024): 1-24.

[22] Sawhney, Rahul, et al. "Ear Biometry: Protection Safeguarding Ear Acknowledgment Framework utilizing Transfer Learning in Industry 4.0." Journal of Electrical Systems 20.3s (2024): 1397-1412.

[23] Gupta, Anuj, et al. "ML-CPC: A Pathway for Machine Learning Based Campus Placement Classification." Journal of Electrical Systems 20.3s (2024): 1453-1464.

[24] Sawhney, Rahul, et al. "An Efficient Scientific Programming Technique for MRI Classification using Deep Residual Networks." Journal of Electrical Systems 20.2s (2024): 241-255.

[25] Narayan, Vipul, et al. "7 Extracting business methodology: using artificial intelligence-based method." Semantic Intelligent Computing and Applications 16 (2023): 123.

BIOGRAPHIES OF AUTHORS

	**Dr. Ashish Tripathi** received his M.Tech degree in CSE, MNNIT Allahabad, in 2012, and a Ph.D. degree in CSE, MNNIT Allahabad, in 2015. He is currently working as an Associate Professor at School of Computing Science and Engineering, Galgotias University, Greater Noida, India. His area of research includes evolutionary computation, ML, DL, Soft Computing, and Cloud Computing.
	**Dr. Rajnesh Singh** received his M.Tech degree in CSE from CDAC Noida, Affiliated to Guru Gobind Singh Indraprastha University, New Delhi in 2006 and PhD degree in CSE from Gautam Buddha University, Greater Noida in 2021. He is currently working as an Associate Professor at School of Computing Science and Engineering, Galgotias University, Greater Noida, India.
	**Dr. Suveg Moudgil** received his M. Tech degree in CSE from Kurukshetra University, Kurukshetra, India, in 2007 and Ph.D. degree in CSE from M.M. (DU), Mullana, Ambala, Haryana, India in the area of Mobile Adhoc Networks in 2018. He is currently working as an Associate Professor at the Department of Computer Science & Engineering, Galgotias University, Greater Noida, India.
	**Dr. Manoj Singhal** received his Ph.D degree in Computer Science from Kurukshetra University, Kurukshetra, India in 2012. He is currently working as a Professor in DIT at GL Bajaj Institute of Technology & Management, Greater Noida, Uttar Pradesh, India.
	**Dr. Girish Kumar Sharma** is working as a Professor in one of the campus of Delhi Skill and Entrepreneurship University in Computer Science and Application Department at Bhai Parmanand DSEU Shakarpur campus -II Delhi (Govt. of NCT of Delhi), India.
	**Dr. Bhoopendra Dwivedy** is currently working as Professor in SCSE, Galgotias University Greater Noida. He has a total experience of more than 21 years of teaching. His areas of research are networks, Vehicular adhoc networks, algorithms and Theoretical computer science.