COMPARATIVE STUDY OF FONT RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS AND TWO FEATURE EXTRACTION METHODS WITH SUPPORT VECTOR MACHINE

Aveen Jalal Mohammed [1]
[1] Department of Computer Science
University of Duhok
Duhok, Iraq
aveen.mohammed@uod.ac

Jwan Abdulkhaliq Mohammed [2]
[2] Department of Computer Science
University of Duhok
Duhok, Iraq
jowan.abdulkhaliq@uod.ac

Amera Ismail Melhum [3]
[3] Department of Computer Science
University of Duhok
Duhok, Iraq
amera_melhum@uod.ac

Abstract **- Font recognition is one of the essential issues in document recognition and analysis, and is frequently a complex and time-consuming process. Many techniques of optical character recognition (OCR) have been suggested and some of them have been marketed, however, a few of these techniques considered font recognition. The issue of OCR is that it saves copies of documents to make them searchable, but the documents stop having the original appearance. To solve this problem, this paper presents a system for recognizing three and six English fonts from character images using Convolution Neural Network (CNN), and then compare the results of proposed system with the two studies. The first study used NCM features and SVM as a classification method, and the second study used DP features and SVM as classification method. The data of this study were taken from Al-Khaffaf dataset [21]. The two types of datasets have been used: the first type is about 27,620 sample for the three fonts classification and the second type is about 72,983 sample for the six fonts classification and both datasets are English character images in gray scale format with 8 bits. The results showed that CNN achieved the highest recognition rate in the proposed system compared with the two studies reached 99.75% and 98.329 % for the three and six fonts recognition, respectively. In addition, CNN got the least time required for creating model about 6 minutes and 23- 24 minutes for three and six fonts recognition, respectively. Based on the results, we can conclude that CNN technique is the best and most accurate model for recognizing fonts.**

Index Terms - Font Recognition, Convolution Neural Network, Support vector machine, Distance profile features, normalized central moments.

I. INTRODUCTION

The purpose of optical character recognition (OCR) is to turn a text picture into a text document that can be searched and edited. However, OCR is not able to retrieve font information. Font is a particular style of text that appears on a page or computer screen. In the field of optical character recognition, character font recognition (CFR) has been studied extensively [1]. Font recognition is an important step in document image analysis that can either be applied as a preprocessing step prior to optical character recognition (OCR) or as a post-processing. The idea of CFR is to get font information from a given text picture in order to assist designers in using it in future designs, or for high performance document recovery in which the typeface should be recognized. Generally, font recognition represents a particular kind of image classification. As a result, characteristics from the image should be retrieved and merged to determine the proper class [2].

Prior work previous to 2012 focused on handcrafting font recognition features. Gabor filters and wavelet transforms have been mainly used by them to extract relevant characteristics. They then fed those characteristics into their selected machine-learning algorithm to assist them with categorization [3]. In the next section, we will see that the majority of the studies focused on two major machine-learning algorithms: Artificial Neural Network (ANN) and Support Vector Machine. Since 2012, however, researchers focused on convolution neural networks (CNN) and support vector machines (SVM) for automated feature creation not just for font recognition but also for medical and generic computer vision issues [4]. This was become possible by the availability of large font datasets and more calculation power supplied by the recent graphic processing units (GPU).

The research problem is that old books and documents are only available in physical form. To make digital files readable on digital devices, such old written materials should be scanned first. OCR saves space and makes documents searchable, but the documents lose their original appearance and feel. If a document has historical importance, keeping the appearance and feel is a desired quality when replicating the document. One solution is to recognize fonts used in the original document and utilize them when recreating the document. Usually, the newly recreated document is in Portable Document Format (PDF) format.

This research specifically aims to present a deep learning approach for English font recognition. The focus is to take English characters images and recognize the font used in the images. The target is to be capable of recognizing three and six English fonts. To perform this target, we propose a system containing two steps: preprocessing and classification. In the first step, the character image is converted from gray scale values (0 to 255) to the range of (0 to 1) to reduce the processing time. In the second step, the images fit to CNN model to train it. In addition, the results of the proposed are compared with the two previous studies. The first study used distance profile features as an input feature to support vector machine (SVM) model as a classification method, and the second study utilized

Normalized central moments (NCM) as input feature to SVM as classification method.

II. RELATED WORK

In this part, we discuss some of the prior studies on font recognition. Adnan Amin [5] used a decision tree generated with C4.5 algorithm to present an algorithm for Arabic font recognition. His that obtained 92% accuracy for 15 different fonts. Zramdini and Ingold proposed a system of font recognition based on connected component statistics that achieved 97.35% accuracy across English text for 10 font classes [6]. Zhu et al. employed Gabor Filters to obtain 99.1% accuracy on text blocks for both English and Chinese text using font recognition as texture identification [7]. In [8], Fractal dimension features were presented to recognize Arabic font which resulted in an accuracy of 98% for 10 font classes. Luqman et al. obtained 96.1% accuracy on 20 fonts in the large scale KAFD dataset using log-Gabor filter features retrieved at several sizes and orientations [9]. Ramanathan et al. [10] introduced a technique for English font identification based on support vector machines with an accuracy of 93.5%.

Recently, deep learning algorithms based on Convolutional Neural Networks (CNN) for font classification have been presented. Wang et al [11] proposed the system of DeepFont that is based on CNN to visualize font recognition from an image and they achieved 80% accuracy. Tao et al. classified single Chinese characters into 7 font classes with 97.77% accuracy using a mix of CNN and 2D RNN models [12]. Pengcheng et al. used deep features derived from a CNN pretrained on natural photos to classify handwritten Chinese characters into 5 calligraphy classes with 95% accuracy [13].

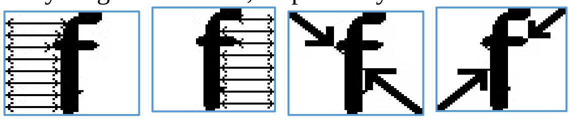
In 2018, [14] created a system to recognize three fonts based on Eigenfaces method, and Decapod and OCRopus software have been used as framework to present the mentioned method because the font recognition should be implemented with other approaches like OCR and the accuracy of proposed system was 97% on 6144 samples. Sakr et al. presented a system to differentiate between 50 Arabic fonts using deep learning method. Their focus is to take a picture containing a text and recognize the font in the picture, the accuracy of the system was 77.2% [15]. Mohammed and Al-Khaffaf [16] presented a system to differentiate three English fonts from character images and they utilized the distance profile as feature extraction method and then feeds the extracted features into support vector machine algorithm. The system obtained 94.82% recognition rate.

In 2022, Mohammadian et al. proposed the first datasets in Persian font recognition and they used CNN as approach to address the issue of Persian font recognition. Their experiment resulted 78% top-1 accuracy on the proposed datasets [17]. In 2023, [18] presented a deep convolutional neural network to generate a mixed training set, specifically employing typefaces such as normal script, cursive script, and seal character as training targets. The model's recognition performance is enhanced further by improving it in three areas: pooling rules, training procedures, and model pruning. It is envisaged that the study presented in its paper would aid in the identification of Chinese characters and the spread of digitalized traditional calligraphy.

III. METHODOLOGY

A. Distance Profile Features

The distance profile feature is a histogram feature that includes left, right, main, and secondary diagonal profile features. For each row, the left distance profile feature is calculated by counting the number of background pixels till the first foreground pixel from left to right. The right distance is calculated in the same way as the left distance, but from right to left. The top left corner and bottom left corner are extracted for the main diagonal, while the top right corner and bottom right corner are tallied for the secondary diagonal. Because the picture size is 51×51, we have 106 features: 51 left features, 51 right features, and 4 diagonal features [19, 20]. Fig. (1a-1d) show the distance profile features for left, right, main and secondary diagonal features, respectively.



(a) (b) (c) (d)

Fig. 1 Distance profile feature. (a) Left distance profile features (b) right distance profile features (c) main diagonal profile features (d) secondary diagonal profile features

B. Normalized Central Moments (NCM)

Normalized central moments has 10 features which consists of(η_{00}, η_{01}, η_{02}, η_{03}, η_{10}, η_{11}, η_{12}, η_{20}, η_{21}, η_{30}), to calculate the following features; We need to go through image moments from order zero to order three [21]. Where Image moment represent as (m_{pq}), the moments of order $(p+q)$of a picture contained binary pixels $f(x,y)$are determined by:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \qquad (1)$$

As demonstrated in Equation (1), (m00) represents the body of area A and the centroid of the image ($\bar{x}$, $\bar{y}$) is obtained from:

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{and} \quad \bar{y} = \frac{m_{01}}{m_{00}} \qquad (2)$$

The central moments that are translation invariant when the location of the object is modified, are obtained from

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y) \qquad (3)$$

Lastly, the normalized central moments (NCM) that are scale and translation invariant are generated from the central moments as follows:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}{}^{\gamma}} \qquad (4)$$

where $\gamma = 1 + \left({}^{p+q}/_{2}\right)$ for $p+q \geq 2$.

C. Support Vector Machine

Support Vector machine (SVM) is a common supervised learning method, it is developed by Vapnik and Cortes. It is effectively utilized in pattern recognition, time series analysis, and classification applications. SVMs can be used on both linear and non-linear data. The SVM employs a kernel function to transform the indivisible input data into a higher-dimensional hyperspace. The purpose of SVM algorithm is to find the decision line for categorizing n-dimensional space into categories, so we can simply put new data in the proper category in the future. This decision line is known as a hyperplane. SVM selects the extreme vectors that aid in the creation of the hyperplane. These extreme vectors are referred to as support vectors, and the method is known as the Support Vector Machine [22]. Consider the diagram below, which shows two distinct categories separated by a decision line or hyperplane:

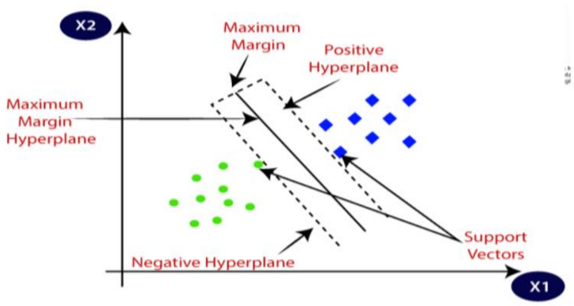


Fig. 2 The classification of SVM

The mapping function, denoted as Φ(x), is the function that carries the nonlinear x data sequence to the higher dimension. In the SVM system, Radial Basis Function (RBF) kernel is utilized to classify the font of English character image using the following equation:

$$K(X_1, X_2) = \exp\left(-\frac{\|X_1 - X_2\|^2}{2\sigma^2}\right) \tag{5}$$

$\|X_1 - X_2\|^2$ is referred as Square Euclidean Distance between two points $X_1$and X_2 and σ is a free parameter which can be utilized to fine-tune the equation.

D. Convolution Neural Network

ConvNets were created to handle data in the form of multidimensional arrays. ConvNets have four typical properties when compared to a fully connected network, such as a multilayer perceptron, namely, local connections, shared weights, pooling, and the use of many layers. Therefore, their architectures typically contain three primary forms of layers: convolutional, pooling, and fully connected layers [23].

1) *Convolutional Layers:* Convolutional layers use several filters to extract features in the immediate area, which are analogous to various feature extractors. Suppose that we have $n^{[l]}$ filters $W^{[l]} \in \mathbb{R}^{3\times3\times n^{[l]}}$ of size 3×3 in the convolutional layer l, where $n^{[l-1]}$ is the number of filters in the previous layer. With stride [1,1], these filters traverse the whole input feature map. The output characteristics are $Y^{[l]}$:

$$Y^{[l]} = f\left(\sum_{n=1}^{n^{[l]}} W^{[l],n} \otimes Y^{[l-1]} + b^{[l]}\right) \tag{6}$$

$f(\hat{A}.)$ represents a non-linear activation function for example a relu or elu activation function. Furthermore, if the convolutional layer is the first layer in the network layer, then $Y^{[l-1]}$ input image $X \in \mathbb{R}^{w\times h\times 1}$. To conserve the invariant size of the input matrices and as much of the margin information as feasible, we pad the matrices' margins with zeros throughout the convolution process, which is known as the same padding.

2) *Pooling Layers:* Pooling layers, also known as subsampling layers, are responsible for feature selection and parameter reduction. The pooling function for every patch of the feature map is divided into two kinds: the first is called maximum pooling which calculates the largest value, and the second is called average pooling which calculates the average value

3) *Fully Connected Layers:* Fully connected layers, such as a multilayer perceptron, are made up of many neurons in every layer. The last pooling layer's feature maps are flattened into a vector, which is needed by fully connected layers. There are three nonlinear activation functions in fully connected layers, such as relu, tanh, and sigmoid that can be employed.

4) *Output Layer:* The output layer, which is also a fully connected layer, produces the final results of classification by multiplying the input with a weight matrix and adding a bias vector, then applying a softmax function [24]. Each prediction is then between the range [0,1], and the outcome is computed:

$$\hat{y}_c = \frac{\exp(z_c)}{\sum_j \exp(z_j)}, \text{ c=1, 2, …. C} \tag{7}$$

where C is the number of classes and z_c is the input of the c^{th} output neuron corresponding to the c^{th} class.

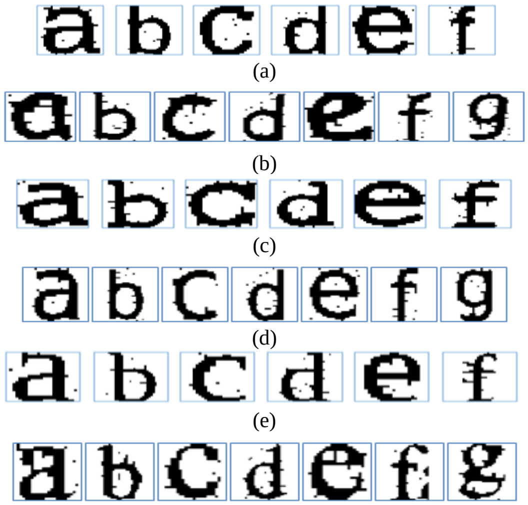
IV. Results

A. Datasets

In this paper, the dataset of sample character glyph images was taken from [25], a 60-page electronic book, where the first three and last three pages has been taken. For this study, two types of datasets have been used: the first type is for the three fonts classification and the second type is for the six fonts classification. For the three fonts classification, the dataset is comprised of 27,620 sample English character images in gray scale format with 8 bits, and the three fonts including (Comic Sans MS (Comic), DejaVu Sans Condensed (DejaVu), Times New Roman (Times)) were used for this type of classification. For the six fonts classification, the dataset is comprised of 72,983 sample English character images in gray scale formats with 8 bits and the six fonts including (Arial, Comic Sans MS (Comic), Courier 10 Pitch, DejaVu Sans Condensed (DejaVu), Liberation Serif (Liberation), Times New Roman (Times)) were utilized in this classification. Both datasets were split into 64% for training, 16% for validation and 20% for testing.

The following sentences show how the dataset was created by [25]. The OCRopus and Decapod open-source software were used to segment the page images into character images in PNG formats. In the second stage, Kanungo et al. algorithm [26] was used to degrade the full image samples. The procedure produced an isolated degraded character picture of size (102x102) pixel with 8bpp. The experiments in this paper employ only the first three and last three pages of the book. Fig.

(2a-2f) depicts samples of letters a, b, c, d, e, f and g that were used for the three fonts classification and the six fonts classification.

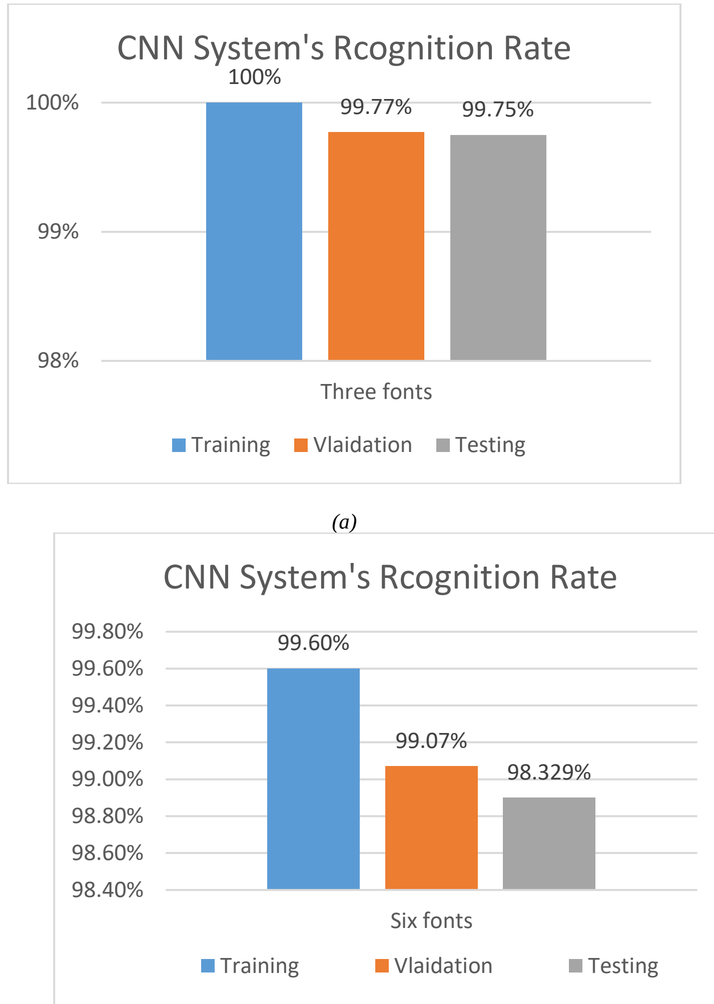


(a) (b) (c) (d) (e) (f)

Fig. 3 Samples of degraded images of letters (a, b, c, d, e, f, and g). (a) Arial, (b) Comic, (c) Courier (d) DejaVu, (e) Liberation, (f) Times.

B. CNN Results

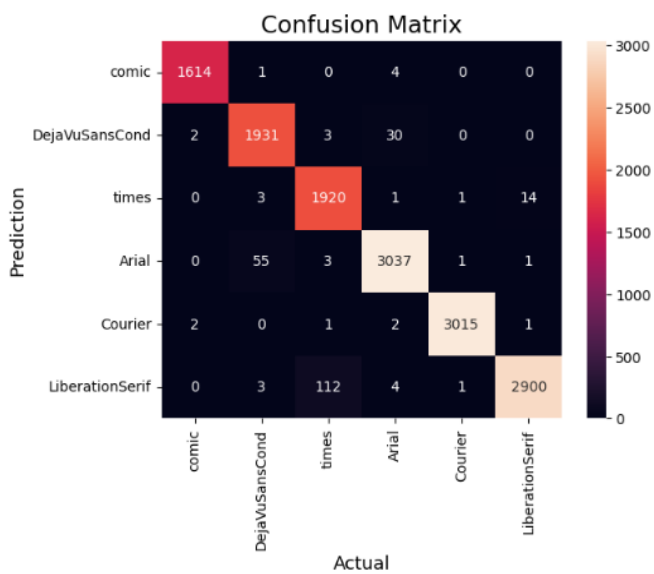
The system is developed using Python programming language on a computer equipped with a Core i7 5500U processor, 2.40 GHz and 8 GB RAM. The proposed system created two models: the first model used a CNN approach to recognize the three English fonts (Comic, DejaVu, Times), and the second model also used the CNN approach to recognize six English fonts (Arial, Comic, Courier, DejaVu, Liberation, Times). Both models consist of a single convolution layer, a single pooling layer, and two fully connected layers. In the convolution layer, the size of the filter is 3×3 with stride 1×1, which means it slides one pixel vertically and one pixel horizontally; and the Relu activation function has been used in this layer. In the pooling layer, the maximum pooling feature map with a size of 2×2 has been used. The Relu activation function has been also used in the fully connected layer. However, each model has a different number of outputs where the first model has three outcomes while six results for the second model in the connected layers. Both models has trained with three epochs and the batch size is 32. The system's recognition rate for the three-font recognition was 99.75% while it is slightly low for the six-font recognition reached 98.329% of the test samples, see Fig. 4.



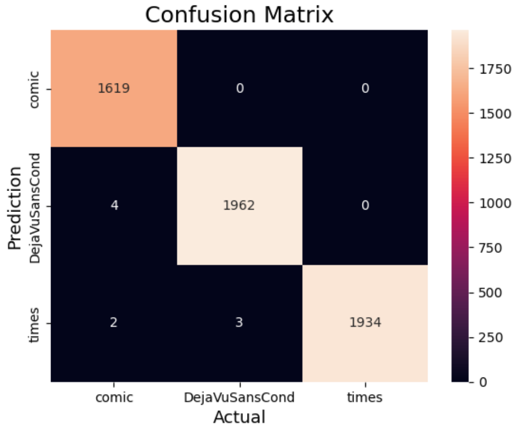
(b)

Fig. 4 The recognition rate of (a) three fonts (b) six fonts.

The below figure presents the confusion matrix for the three and six fonts.



(a)



(b)

Fig. 5 The confusion matrix of (a) three fonts (b) six fonts.

C. *Implementation On User Interface*

This part presents a GUI (Graphical User Interface) that has been created for implementing the proposed font recognition system as shown in Fig 6. This user-friendly interface consists of four menus: File, Create CSV, Training and Testing. For the first time using the system; we need to train the system.

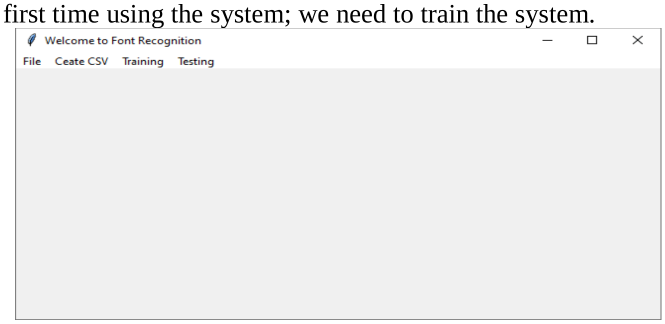
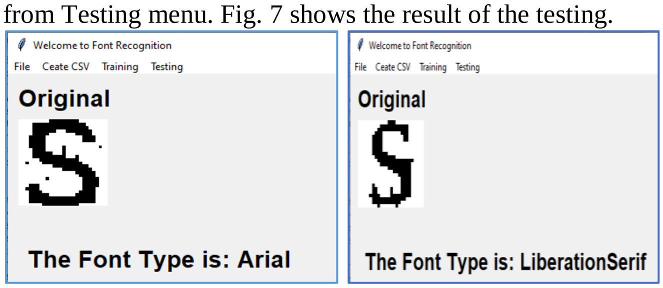


Fig. 6 The font recognition system using GUI.

First you will go to Create CSV; which has three options: training, validation and testing, here a CSV file created to save the path of training, validation and testing images separately. After the path of training and validation data is specified, now the system is able to train the model through the third menu called Training. The fourth menu is used to test the system, which has two options: CNN model, Model Accuracy. The CNN model is used to recognize the font of a single image; by doing the following steps: first go to file menu and then chose open to select an image to be tested, then select CNN model from Testing menu. Fig. 7 shows the result of the testing.



(a) (b)

(e) (f)

Fig. 7 The results of font recognition for (a) Arial, (b) Liberation, (c) Comic, (d) Courier (e) Déjà vu, and (f) Times.

V. Discussion

This part will discuss the proposed system with the two previous studies. The two studies have used two feature extraction algorithms (DP and NCM) with SVM and RBF as a kernel separately. For DP, the gamma and C pair of (1/10, 7); while For NCM, the gamma and C pair of (1/26, 35) has been used. Fig. 8 shows the result of the all experiments. In the three fonts recognition, there is a slide difference between the recognition rates of the model [14] and [16] which are (94.82% and 94.90%) respectively, while the CNN model has the highest recognition rate reached (99.75%). For the six fonts recognition, [16] model shows the least recognition rate; which is 88.57%, while the CNN model also recorded the best rate reached (98.329%).

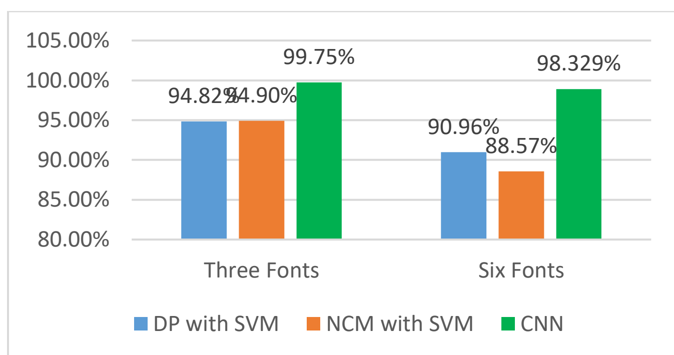


Fig. 8 The recognition rate of CNN and the two studies

Another comparison is based on the required time for creating a model. Table 1 depicts the three experiments for each model. For the three fonts recognition, the NCM with SVM model have taken between 23- 24 minutes to be trained and tested which is the longest time than others models, while the

CNN Model has recorded the least time to be trained and tested about 6 minutes. For the six fonts recognition, the NCM with SVM model have also recorded the longest time around 1 hour while CNN model has taken least time than others about 24 minutes.

The reason of making CNN better than the mentioned methods is that CNN technique can be used to recognize fonts without the need for additional pre-processing steps such as feature extraction, normalization, etc. However, NCM has a lot of multiplication and division operations; and this takes a lot of processing time and DP has a lot of features that makes SVM takes a lot of time to create hyper-plane.

TABLE I
THE DURATION TIME FOR EACH MODE

Models	Experiments	3 fonts	6 fonts
Distance Profile (DP) with SVM	Ex 1	00:14:59	00:39:51
	Ex 2	00:13:40	00:36:31
	Ex 3	00:13:40	00:36:36
Normalized Central Moments (NCM) with SVM	Ex 1	00:24:35	01:05:46
	Ex 2	00:23:29	01:02:14
	Ex 3	00:23:27	01:01:55
CNN	Ex 1	0:06:36	0:24:08
	Ex 2	0:06:30	0:23:16
	Ex 3	0:06:27	0:24:07

VI. CONCLUSION

The paper presents research on the study of font recognition by building models for categorizing English language fonts into three and six fonts using CNN. The focus of study is to take an image of character and recognize the font in the image. This research focuses on detection and classification of character images with three or six fonts. The aim is to be able to recognize three or six fonts. To achieve this aim, the proposed deep neural network was trained and tested using Al-Khaffaf dataset [21]. Then, the results of proposed font recognition system were compared with the two studies. The first study used NCM features and SVM as a classification method, and the second study used DP features and SVM as classification method. Experiment results showed that the CNN system has achieved the best recognition accuracy than other studies for recognizing the three and six English fonts reached 99.75% and 98.329% recognition rate, respectively. There are two main reasons that make CNN better than other techniques: first, it can extract high-level features such as shapes and objects; it also finds the position of the shape in the image. Second, CNN has a pooling layer, this layer helps to reduce the number of features, as a result, reduces the computational time.

However, we also note that our experiment is limited to only one dataset with three and six fonts. In the future, we plan to run an experiment with two or three datasets added to Al-Khaffaf dataset [25] and make a comparison with them with more fonts.

REFERENCES

[1] Sharada, B. A Convolutional Autoencoder based Keyword Spotting in Historical Handwritten Devanagari Documents. In 2022 International Conference on Inventive Computation Technologies (ICICT) 2022, 356-362. IEEE.

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

[3] Isma'eel, H. A.; Sakr, G. E.; Habib, R. H.; Almedawar, M. M.; Zgheib, N. K.; Elhajj, I. H. Improved accuracy of anticoagulant dose prediction using a pharmacogenetic and artificial neural network-based method. European journal of clinical pharmacology 2014. 70, 265-273.

[4] Mohammadian, M.; Maleki, N.; Olsson, T.; Ahlgren, F. Persis: A Persian Font Recognition Pipeline Using Convolutional Neural Networks. In 2022 12th International Conference on Computer and Knowledge Engineering (ICCKE) 2022,196-204. IEEE.

[5] Amin, A. Recognition of printed Arabic text based on global features and decision tree learning techniques. Pattern recognition 2000, 33, 8, 1309-1323.

[6] Zramdini, A.; Ingold, R. Optical font recognition using typographical features. IEEE Transactions on pattern analysis and machine intelligence 1998, 20, 8, 877-882.

[7] Zhu, Y.; Tan, T.; Wang, Y. Font recognition based on global texture analysis. IEEE Transactions on pattern analysis and machine intelligence 2001, 23, 10, 1192-1200.

[8] Moussa, S. B.; Zahour, A.; Benabdelhafid, A.; Alimi, A. M. New features using fractal multi-dimensions for generalized Arabic font recognition. Pattern Recognition Letters 2010, 31, 5, 361-371.

[9] Luqman, H.; Mahmoud, S. A.; Awaida, S. KAFD Arabic font database. Pattern Recognition 2014, 47, 6, 2231-2240.

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

[11] Wang, Z.; Yang, J.; Jin, H.; Shechtman, E.; Agarwala, A.; Brandt, J.; Huang, T. S. Deepfont: Identify your font from an image. In Proceedings of the 23rd ACM international conference on Multimedia 2015, 451-459.

[12] Tao, D.; Lin, X.; Jin, L.; Li, X. Principal component 2-D long short-term memory for font recognition on single Chinese characters. IEEE transactions on cybernetics 2015, 46, 3, 756-765.

[13] Pengcheng, G.; Gang, G.; Jiangqin, W.; Baogang, W. Chinese calligraphic style representation for recognition. International Journal on Document Analysis and Recognition (IJDAR) 2017, 20, 59-68.

[14] Al-Khaffaf, H; Musa, N. Optical english font recognition in document images using eigenfaces. Revista Innovaciencia 2018, 6, 1, 1-11.

[15] Sakr, G.; Mhanna, A.; Demerjian, R. Convolution Neural Networks for Arabic Font Recognition. In Proceedings of the 2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Sorento, Italy, 26–29 November 2019; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA 2019, 128–133.

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

[17] Mohammadian, M; Maleki, N; Olsson, T; Ahlgren, F.; Persis: A Persian Font Recognition Pipeline Using Convolutional Neural Networks. 2022 12th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, Islamic Republic of 2022, 196-204, doi: 10.1109/ICCKE57176.2022.9960037.

[18] Zhang, X; Research on Applying Dense Convolutional Neural Network in Chinese Character Font Recognition. 2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China, 2023, 1589-1593, doi: 10.1109/ICETCI57876.2023.10176490.

[19] Mohammed, A. J.; Al-Khaffaf, H. S. Font Recognition of English Letters Using Normalized Central Moments Features. In 2020 International Conference on Computer Science and Software Engineering (CSASE) 2020, 249-254.

[20] Al-Sheikh, I. S.; Mohd, M. A. S. N. I. Z. A. H.; Warlina, L. A review of arabic text recognition dataset. Asia-Pacific J. Inf. Technol. Multimedia 2020, 9, 1, 69-81.

[21] Moung, E. G.; Hou, C. J.; Sufian, M. M.; Hijazi, M. H. A.; Dargham, J. A.; Omatu, S. Fusion of moment invariant method and deep learning algorithm for COVID-19 classification. Big Data and Cognitive Computing 2021, 5, 4, 74.

[22] Tanveer, M.; Rajani, T.; Rastogi, R.; Shao, Y. H.; Ganaie, M. A. Comprehensive review on twin support vector machines. Annals of Operations Research 2022, 1-46.

[23] Zhang, X.; Wu, F.; Li, Z. Application of convolutional neural network to traditional data. Expert Systems with Applications 2021, 168, 114185.

[24] Chunman, Y.; Cheng, W. Development and application of convolutional neural network model. Journal of Frontiers of Computer Science & Technology 2021, 15, 1, 27.

[25] Al-Khaffaf, H. S.; Shafait, F.; Cutter, M. P.; Breuel, T. M. On the performance of Decapod's digital font reconstruction. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012) 2012, 649-652. IEEE.

[26] Kanungo, T.; Haralick, R. M.; Baird, H. S.; Stuezle, W.; Madigan, D. A statistical, nonparametric methodology for document degradation model validation. IEEE Transactions on Pattern Analysis and Machine Intelligence 2000, 22,11, 1209-1223.