Muthana J. Mohammed, Emad A. Mohammed and Mohammed S. Jarjees*

Recognition of multifont English electronic prescribing based on convolution neural network algorithm

https://doi.org/10.1515/bams-2020-0021
Received April 8, 2020; accepted August 17, 2020; published online September 14, 2020

Abstract: The printed character recognition is an efficient and automatic method for inputting information to a computer nowadays that is used to translate the printed or handwritten images into an editable and readable text file. This paper aims to recognize a multifont and multisize of the English language printed word for a smart pharmacy purpose. The recognition system has been based on a convolution neural network (CNN) approach where line, word, and character are separately corrected, and then each of the separated characters is fed into the CNN algorithm for recognition purposes. The OpenCV open-source library has been used for preprocessing, which can segment English characters accurately and efficiently, and for recognition, the Keras library with the backend of TensorFlow has been used. The training and testing data sets have been designed to include 23 different fonts with six different sizes. The CNN algorithm achieves the highest accuracy of 96.6% comparing to the other state-of-the-art machine learning methods. The higher classification accuracy of the CNN approach shows that this type of algorithm is ideal for the English language printed word recognition. The highest error rate after testing the system using English electronic prescribing written with all proposed font-types is 0.23% in Georgia font.

Keywords: convolution neural network; deep learning; electronic prescribing; printed character recognition.

Introduction

A pharmacy provides a direct link between the patients and their manner of cure in the health-care system. Any pharmaceutical system should promote both safety and health. These processes are not easy to perform and more prone to errors, particularly when only manpower is used. The risk of errors can harm the patient, and it can lead to death in some cases [1, 2].

Dispensing is a complex process that involves a pharmacist read the prescription and fetch medication from shelves. During the dispensing process, errors may arise at any of these two stages. In the pharmacies community of England and Wales, more than 600 million prescription products are dispensed each year. A prospective study in 35 pharmacies community over a four-week period in the UK (United Kingdom) was implemented. Pharmacists registered details about all incidents during the dispensing process. Within the study period, 1,25,395 prescribed products were dispensed and 330 incidents were registered relating to 310 prescriptions. Moreover, 280 (84.8%) incidents were classified as a near miss (where the error was detected before the patient was given the medicine) while the remaining 50 (15.2%) were classified as dispensing errors (where the patient was given a wrong medicine). Most of the incidents were caused either by misreading the prescription (24.5%), choosing the previous medication from the patient's database record on the pharmacy computer (11.4%) similar medication names (16.8%) or similar packaging (7.6%) [3].

Additionally, in the United States, the estimated numbers of medical errors in hospitals have significantly increased from 98,000 to 40,0000 [4]. To overcome these human errors, several studies have been conducted using printed character recognition, which plays an important role in analyzing printed and handwritten document images [5].

Recently, the recognition of English printed word has become essential in many fields of computer applications. Several algorithms on text recognition have been studied. Patil and Mane [6] provided a review of handwritten and typewritten recognition based on template match, fuzzy logic, and neural network. Zanwar et al. [7] suggested two phases for character recognition, the first phase is the training phase that includes filtering process, feature extraction, and feature optimization. The normalization of the image has been performed by filtering process, while the feature extraction has been obtained by independent

*Corresponding author: Mohammed S. Jarjees, Technical Engineering College, Northern Technical University, Mosul, Iraq, E-mail: mohammed.s.jarjees@ntu.edu.iq
Muthana J. Mohammed and Emad A. Mohammed, Technical Engineering College, Northern Technical University, Mosul, Iraq, E-mail: engmothanajasem@gmail.com (M.J. Mohammed), e.a.mohammed@ntu.edu.iq (E.A. Mohammed)

component analysis. The feature optimization was produced by particle swarm optimization. A backpropagation neural network has also been used for classification in the second phase. Ali et al. [8] proposed wavelet compression for character recognition. Wavelet compression extracted the important coefficient that will be used as a vector matrix for calculating the Euclidean distance. Euclidean distance between the coefficient of testing image and data set from training images has been calculated. The total recognition rate using a data set of 26 characters each with eight different font rate was 78.8%. Siriteerakul [9] suggested classification of mixed Thai-English character based on histogram of oriented gradient as feature extraction from the image to creating a descriptor vector for each image in the data set. Set of descriptor vectors were collected for training the support vector machine. The achieved accuracy using data set that contained 5,41,440 images for English characters and 62,784 images of Thai characters with 142 classes is 97%. Janrao and Dighe [10] used a diagonal approach for feature extraction and K-nearest neighbor (KNN) algorithm, as well as linear vector quantization (LVQ) for recognition of handwritten English characters. Each character image is resized into 90×60 pixel and then the image is separated into 54 regions each with 10×10 pixels. Each region has 19 diagonal lines. These lines are averaged and stored in each region that will continue until the 54 regions.

The accuracy has been obtained using two data sets, the center of excellence for document analysis and recognition database data set, and a data set that were created by different writers are 100%, 93.65%, 77.80% and 88.29% from the KNN and LVQ, respectively. Dash and Nayak [11] suggested the recognition of the handwritten English character using artificial neural network. The experiment was carried out using ‘C’ language. The maximum accuracy obtained using 10 samples of each small and capital of the English alphabets is 92.59%. Rahiman et al. [12] suggested a way for the recognition of printed documents in Malayalam and English based on comparing each character with the image in database after resizing. They start by several preprocessing steps that includes lines, words, characters segmentation, and resizing character into 16×16 bit map. An accuracy of 87.25% is achieved using 124 images. Prasad and Sanyal [13] used two stages for handwritten English character recognition. The first stage covers preprocessing steps including thickening, threshold, median filtering, thinning, and extraction of local and global features. The second stage involves the recognition of the English characters using hidden Markov model. The data set that has been used in this study composed of 13,000 samples generated from 100 writers. An average recognition rate obtained using 2600 samples was 93.24%. Seeri et al. [14] used the probabilistic neural network (PNN) for the recognition of the English characters (digits, upper case letters, and lower case letters) in natural scene images. The projection profile was used to segment each character in the natural scene images. Each of these characters was fed to the PNN for recognition. Two thousand six hundred twenty-eight samples of the international conference on document analysis and recognition database have been used to train the PNN. An accuracy achieved using 2026 samples was 79.07%. Nasser and Karim [15] proposed a system for the recognition of English characters based on the bag of visual words (BOVWs). They used Haar filter to transform the character image into the frequency domain. Then, to detect the interest points, the feature from accelerated segment test corner detection has been used. Speeded up robust features descriptor was also used to describe those points. After that, they used the clustering algorithm of moving k-means to obtain BOVWs and then build vocabulary. For character recognition, they compared vocabulary with all vocabulary databases using the Manhattan Distance Measure. Ansari [16] suggested the recognition system for the character written on a vehicle Indian number plate using a template matching. Two stages for character recognition are involved. In the primary stage, the image of the license plate is captured using a digital camera. Then, the input image is converted into a grayscale image. After that, morphological operations such as dilation and erosion are performed to enhance the quality of the image for better recognition. Finally, each character on the image of the number plate is segmented. The segmentation was performed on the basis of connected components. The second stage involves the recognition of each separated character using correlation between segmented character and the templates of the English alphabets in the database.

In this paper, the recognition of multifont English electronic prescribing based on the convolutional neural network (CNN) algorithm has been proposed. CNN combines the automatic features extraction and classification layers. The proposed system contains several preprocessing steps to extract each character. The flow diagram of the English text recognition proposed system is shown in Figure 1.

This paper is organized as follows: the background and state-of-the-art works are presented in Section 1, the segmentation process and the proposed convolution neural networks (CNN) architecture is explained in Section 2, experimental results are provided in Section 3 and finally conclusion is given.

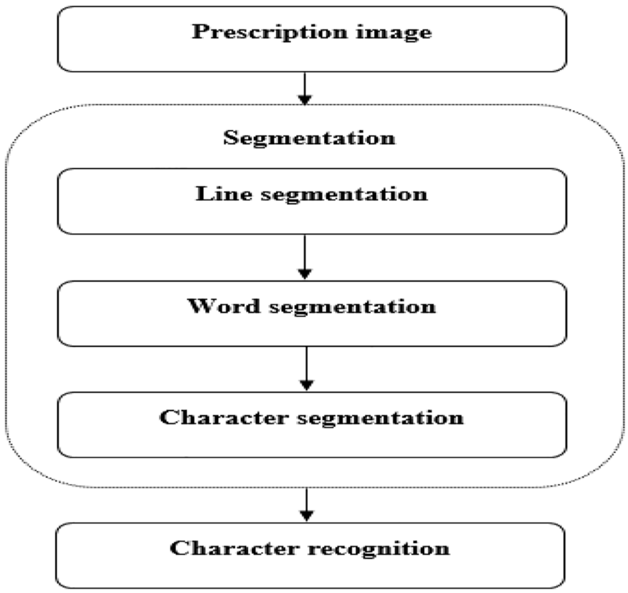

Figure 1: The flow diagram of proposed system.

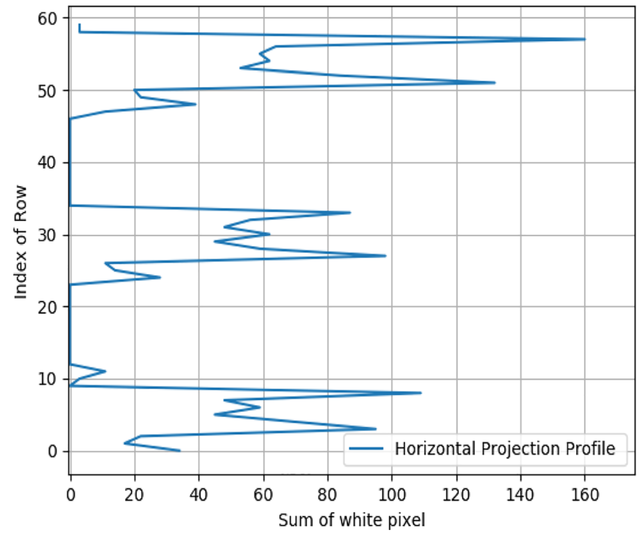

Figure 2: Horizontal projection profile.

Materials and methods

The segmentation process and the proposed convolution neural networks (CNN) architecture have been explained in this section.

Segmentation process

Segmentation is a process of splitting the printed English document into three levels: line, word, and character segmentation. A binary image has two-pixel values, background pixels are represented by '1' and word text pixels are represented by' 0'. For input image segmentation, the binary image has been inverted; therefore the background pixels are with 0-pixel value and foreground with 1-pixel value [17]. The recognition accuracy is greatly dependent on the accuracy of segmentation [18].

Text-line segmentation: In printed English language, the height of whole text lines is practically the same. There is also existed with space between two text lines that have values of only background pixels (0). Thus, to find the space line between two text lines, horizontal projection profile (HPP) has been calculated. The image is scanned row by row and the sum of the whole white pixels in each horizontal row has been calculated. If the sum has been found 0, it implies all the pixels in that horizontal row are background pixels, and this row has been considered as an empty row. The HPP of image text composed from three lines has been illustrated in Figure 2. HPP is given in Eq. (1) [19].

$$HPP = \sum_{0}^{m-1} f(x,y) \tag{1}$$

Word segmentation: For word segmentation, every extracted text from text line segmentation is processed. The common approach used for word and character segmentation is a vertical projection profile (VPP) [20]. Each text line is scanned column by column, and the sum of the whole white pixels is calculated. English characters in each word have been separated by blank space with a width less than a blank space between words. Using this space information, word segmentation has been performed. If space-counter is greater than the threshold, the word will be detected. VPP of image text composed from three words has been illustrated in Figure 3. VPP is given in Eq. (2) [19].

$$VPP = \sum_{0}^{n-1} f(x,y) \tag{2}$$

Character segmentation: Character segmentation is the final level for text-based image segmentation. The characters of each English word are separated in the same manner of word segmentation, and VPP has been calculated and the space between characters are detected.

Convolution neural network

Convolutional Neural Network also known as ConvNet or CNN is a deep learning approach that consists of several numbers of layers. CNN's have excellent applications such as segmentation [21], object detection [22], and face recognition [23]. LeNet was the first CNN practical architecture used for handwritten digit recognition in 1998 [24]. The parameters number of CNN models has been widely reduced by using share weights and biases compared with other neural network algorithms [25]. Whatever its form or sophistication, the CNN structure consists mainly of five layers: an input layer, a convolution layer, a pooling layer, a fully connected hidden layer, and a softmax layer [26].

The input layer is an input of the neural network; it represents all matrix pixels of the input image, the RGB input image is typically used in convolutional neural network architecture. Convolution layers is an important layer in deep convolution neural network. The objective of the convolution layer uses convolution operation (*) around the image to extract the main feature maps. The element concern that implementation of the convolution operation in the convolution layer is called Kernel or Filter, which is a square matrix lesser than an image.

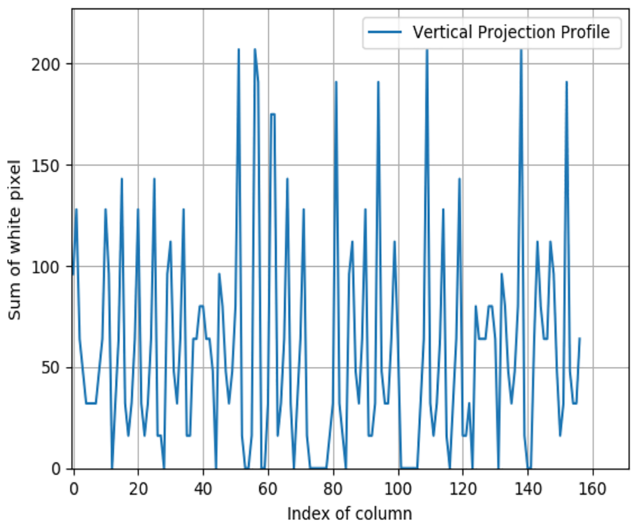

Figure 3: Vertical projection profile.

The zone in the image that the kernel is applied is called the local receptive field [27]. The convolution performs dot products between the kernel with a patch of the image. The formula of the convolution operation per filter is shown in Eq. (3) [28].

$$\sum_i w_i x_i + \mathrm{b} \tag{3}$$

where W is a weight of the filter, b is a bias of the filter, and x is a patch of the image.

The rectified linear unit (ReLU) has been used after every convolution layer and fully connected layers (except the last fully connected layer) in the CNN architecture. CNN using ReLU function train multiple times faster than using other activation functions such as Tanh or sigmoid functions [29]. The ReLU activation function is shown in Eq. (4) [30].

$$f(x) = \max(0\,;\,x) \tag{4}$$

The pooling layer is also called the subsampling layer, which is used to reduce the size of the input data; thus the trainable parameters and computations of the network will be reduced. Another benefit of pooling layers is the generalization of the resultant feature maps from the convolution layer that causes the classification invariant to distortion effects and orientation changes. There are several ways of pooling in the deep CNN, but max pooling has been mostly used.

The convolutional and pooling layers are followed by one or more fully connected hidden layers, where the neurons in the previous layer will be connected to all the neurons of the first fully connected layers. The feature maps of the image, trained by the preceding layers, are combined to perform the classification of the images. The number of outputs neuron in the last fully connected hidden layer is equal to the number of the classes in the input images.

The softmax activation function appears in the final output layer of some deep learning architecture. The softmax function is used to calculate the probability of distribution from a vector of real numbers. The softmax function generates an output within range values between 0 and 1, where the sum of all probability equals to 1. The probabilities of each class in the output of CNN models are returned by it, with the highest probability in the target class. Both the sigmoid and softmax activation functions are used in deep learning architecture, but the softmax activation function is used in multiclass classification tasks, while the sigmoid activation function is used in binary classification tasks [31]. The softmax activation function is shown in equation 5 [32].

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{5}$$

Experiment platform

The experiments of the English character recognition have been implemented on a Laptop PC (64 bit Operating system, x64-based processor,8 GB RAM, Intel(R) Core(TM) i5-3210 processor) with Open CV, Keras and TensorFlow python libraries and Jupyter notebook environment.

Proposed system

After the segmentation process is finished and each character is isolated, the CNN model will be ready to recognize each character. The architecture of the proposed CNN model is illustrated in Figure 4. It consists of six layers. CNN takes an input image with a fixed size. Therefore, all training images are resized to 32×32 pixels. CNN layers divide into two parts, the first three layers perform feature extraction and the last three hidden layers are responsible for feature classification.

The first layer (L1) is a convolution layer that has 32 filters each with the size of 3×3 and padding the 'SAME' value. The convolution operation is implemented in this layer by convoluted the filters with the input image and 32@32×32 output feature maps are obtained. Nine trainable weights plus a trainable bias, therefore this layer has 896 $(3 \times (32 \times 9) + 32)$ trainable parameters.

The second layer (L2) is also a convolution layer. It contains 32 kernels. This layer differs from the first layer (L1) by adding max pooling and dropout layers. Max pooling based on nonoverlapping with the size of 2×2 kernel in the x and y directions which produces 32 feature maps each of size 16×16. [34] For regularization and to control over fitting, the dropout layer has been added with a probability of 0.5. Therefore, 9,248 trainable parameters are produced in this layer.

The third layer consists of three layers as in (L2) by using 64 kernels with the size of 3×3 in the convolution layer, 64 feature maps of each size 14×14 are created. These are further reduced to the size 7×7 in the max pooling layer, and 64 feature maps are obtained by this layer (L3). Therefore, 18,496 trainable parameters are produced in this layer.

The last three layers are responsible for features classification. The first hidden layer has been composed of 512 neurons. The second hidden layer contained 256 neurons. The third layer has been composed of 52 neurons, indicating the number of classes of the input images. The CNN model architecture has 1,779,476 trainable parameters. The proposed system has been trained using Adam optimizer with a learning rate of 0.0001, decay of 0.000003125 and a batch size of 32. Categorical cross-entropy has been used to calculate the loss function, then the trainable parameter of the network has been updated to minimize the prediction loss. In Table 1 are shown the whole parameter of the proposed network.

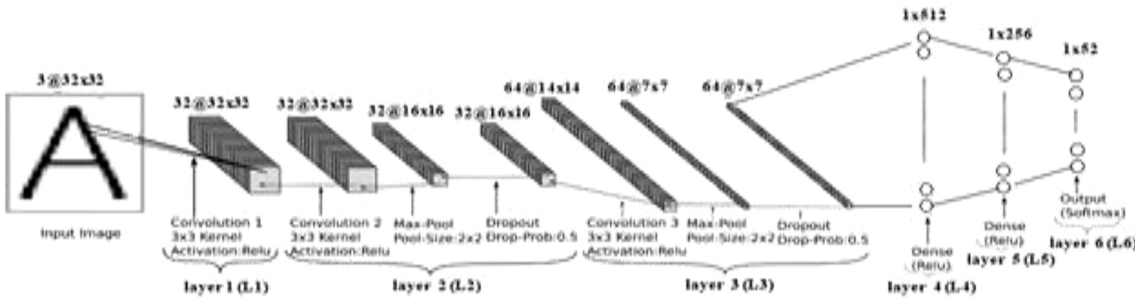

Figure 4: The proposed CNN model [33].

Table 1: The architecture parameters of the proposed CNN.

Layer (type)	No. Kernel	Kernel size	Output shape	No. parameters
Input	-	-	(3,32,32)	0
Conv1(Conv2D)	32	3 × 3	(32,32,32)	896
Conv1(Conv2D)	32	3 × 3	(32, 32, 32)	9248
Pooling1(MaxPooling2D)	-	2 × 2	(16, 16, 32)	0
Drop1(Dropout)	-	-	(16, 16, 32)	0
Conv3(Conv2D)	64	3 × 3	(14, 14, 64)	18496
Pooling2(MaxPooling2D)	-	2 × 2	(7, 7, 64)	0
Drop2(Dropout)	-	-	(7, 7, 64)	0
Flatten(Flatten)	-	-	(3136)	0
Dense1(Dense)	-	-	(512)	1606144
Dense2(Dense)	-	-	(256)	131328
Dense3(Dense)	-	-	(52)	13364
Output(Softmax)	-	-	(52)	0

Proposed data set model

The data set, which has been used in this paper, is created with different fonts and sizes. It is composed of 276 images of each English character that are in font-type (Georgia, Lucida Sans Typewriter, Cambria, Garamond, Arial, Palatino Linotype, Microsoft Sans Serif, New century schoolbook, Calibri, Arial Rounded MT Bold, Bodoni MT, Rockwell, Century Schoolbook, Book Antiqua, Tahoma, Yu Gothic UI Light, Simplified Arabic Fixed, Agency FB, Bahnschrift Light Condensed, Candara Light, Eras Light ITC, Imprint MT Shadow, Haettenschweiler). Each English character data set has six different sizes and bolds. Therefore, the total number of character images is 14,352 in a data set. All images in data set are resized with 32 × 32 pixels.

Results and discussion

Feature visualizing is used to display the feature maps that are the output of the different hidden convolution layers in a model network. This provides an overview of how the input has been decomposed into the various learned filters by the network. In this experiment, the output of different convolution layers in the proposed CNN model have been shown.

Figure 5 shows the visualization feature maps of convolution layers used in the proposed model with an input image of ‘A’ character. It can be seen that the feature maps at first and second convolution layer are acting as various edge detectors. It can be also noted that the feature maps at the third convolution layer give less information about the actual input because the feature maps become more sparse.

The data set has been randomly shuffled and divided into a two-part training set with 11,190 images and validation with 2798 images. Throughout the training process, the model has achieved a validation accuracy of 96.6%. The accuracy rate has been calculated using Eq. (6) [35].

Figure 6 illustrates the accuracy of both training and validation. It can be seen that the accuracy of the training model begins at about 0.70 or 70% and continues to

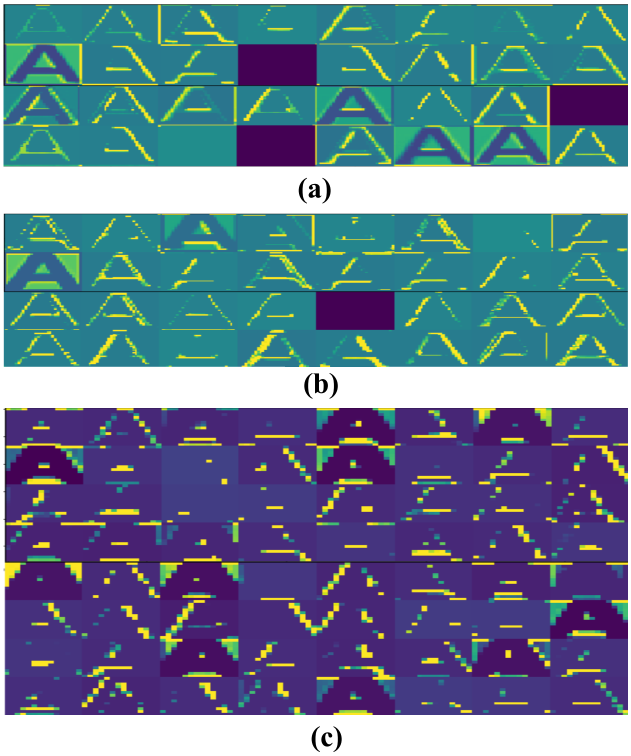

Figure 5: Feature visualization using ‘A’ character for our proposed model, a-b-c the Output feature maps of first, second, and third convolution layers in the CNN model, respectively.

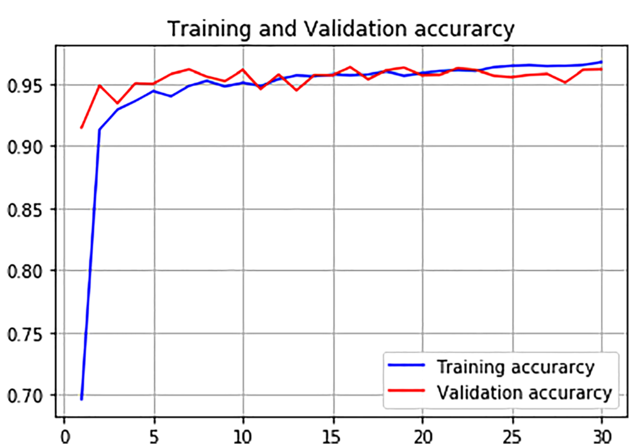

Figure 6: The training and validation accuracy of the proposed model.

increase with each epoch as the model has been trained. Figure 7 shows the amount of loss achieved at the end of the epoch. It can be seen that the proposed model produces a large classification rate and a small loss rate. The proposed model confusion matrix on validation samples of the printed English character images is illustrated in Figure 8.

The proposed model produces some misclassified characters due to the character is written in different fonttype. Therefore, most of them appear in similar forms, for

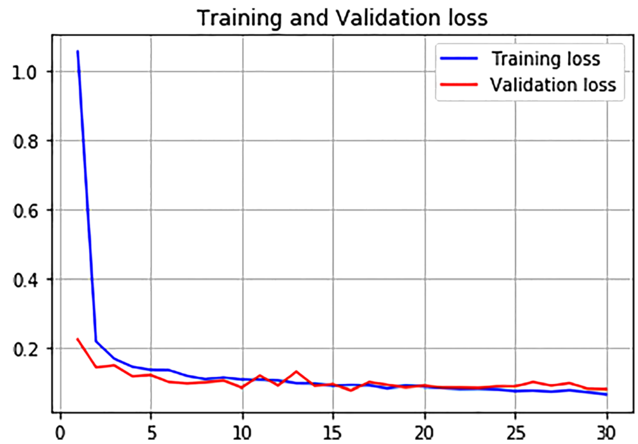

Figure 7: Training and validation losses of our proposed model.

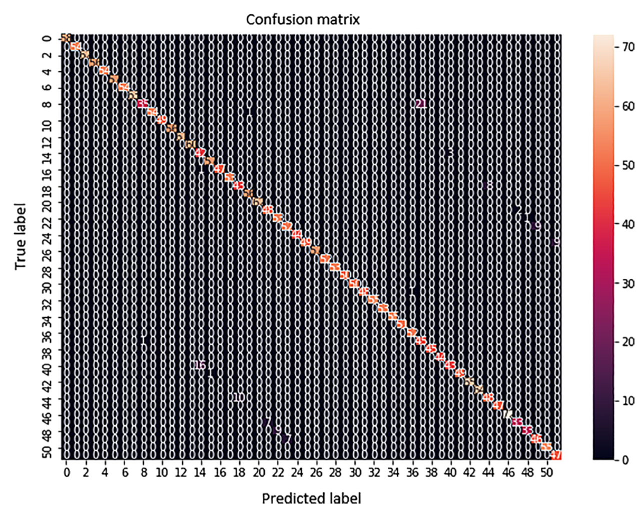

Figure 8: Confusion matrix for a subset of 52 classes (capital and small) of English characters, 0 corresponding to ‘A’ and 51 corresponding to ‘z’.

example, a capital letter of (I) character in fonts (Arial, Microsoft Sans Serif, Calibri, Arial Rounded MT Bold, Yu Gothic UI Light and Agency FB) is almost similar to form of a small letter (l) character in our proposed data sets, and other characters are classified as wrong, but these mistakes in the classification give its correct result, for example, that some small letter of (v) are classified in the class of (V), as well as a letter of (S), which are classified in the class of (s).

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \times 100\% \qquad (6)$$

The efficiency of the proposed model is evaluated by doing a comparison with other methods using for the recognition of multifont characters. Table 2 shows the comparison between the state-of-art systems and the proposed system in terms of method, number of font style, size of training and testing dataset, and accuracy.

Table 2: The comparison between the state-of-the-art systems and the proposed system in terms of method, no. of font style, size of training and testing data set, and accuracy.

Reference	Method	No. of font style used for upper/lower letters	Training data set size	Testing data set size	Accuracy (%)
[36]	KNN	10 Upper/lower	512	260	75.24%
[36]	SVM	10 Upper/lower	512	260	80%
[37]	SVM + RBF	3 Upper/lower	17677	5524	94.82%
[38]	SOM	24 Upper	6162	–	96.5%
[39]	Inception V3 model	–	53342	–	90.6%
Proposed model	CNN	23 Upper/lower	11190	2798	96.6%

SVM, support vector machine; KNN, k-nearest neighbor.

Conclusion

Character recognition is a key step toward artificial intelligence and computer vision field. This paper focused on the recognition of printed English characters in smart pharmacies using deep learning CNN. The proposed model has been obtained an accuracy of 96.6% after training with few numbers of epochs and without using a large training data set. Most of the errors are due to the character that has been written in different font-type, and some of these characters appear in similar forms. The benefits of the proposed method are being more resistant to the font type and requiring a smaller number of training samples when compared with other approaches in the literature. As future work, the proposed model will be expanded for word recognition rather than character recognition.

Research funding: There is no research funding for this study.

Author contributions: All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.

Competing interests: The funding organization(s) played no role in the study design; in the collection, analysis, and interpretation of data; in the writing of the report or in the decision to submit the report for publication.

Informed consent: Informed consent was obtained from all individuals included in this study.

Ethical approval: The conducted research is not related to either human or animal use.

References

1. Beso A, Franklin BD, Barber N. The frequency and potential causes of dispensing errors in a hospital pharmacy. Pharm World Sci 2005;27:182–90.
2. James KL, Barlow D, McArtney R, Hiom S, Roberts D, Whittlesea C. Incidence, type and causes of dispensing errors: a review of the literature. Int J Pharm Pract 2009;17:9–30.
3. Ashcroft DM, Quinlan P, Blenkinsopp A. Prospective study of the incidence, nature and causes of dispensing errors in community pharmacies. Pharmacoepidemiol Drug Saf 2005;14:327–32.
4. Boyd AM, Chaffee BW. Critical evaluation of pharmacy automation and robotic systems: a call to action. Hosp Pharm 2019;54:4–11.
5. Parameshwarppa S. Handwritten Kannada characters recognition using curvelet transform. Int J Comput Appl 2015: 12–16. https://doi.org/10.1109/erect.2015.7499039.
6. Patil JM, Mane AP. Multi font and size optical character recognition using template matching. Int J Emerg Technol Adv Eng 2013;3:504–6.
7. Zanwar SR, Narote AS, Narote SP. English character recognition using robust back propagation neural network. In: Communications in Computer and Information Science. Singapore: Springer; 2019, vol 1037:216–27 pp.
8. Ali H, Ali Y, Hossain E, Sultana S. Character recognition using wavelet compression. In: Proceedings 2010 13th International Conference and Computer and Information Technology ICCIT 2010; 2010:452–7 pp. https://doi.org/10.1109/ICCITECHN.2010.5723900.
9. Siriteerakul T. Mixed Thai-English character classification based on histogram of oriented gradient feature. In: Proceedings 12th International Conference on Document Analysis Recognition, ICDAR; 2013:847–51 pp. https://doi.org/10.1109/ICDAR.2013.173.
10. Janrao RR, Dighe MDD. Handwritten English character recognition using lvq and knn. Int J Eng Sci Res Technol 2016;5:904–12.
11. Dash T, Nayak T. English character recognition using artificial neural network. In: Proceedings of the National Conference AIRES-2012, Andhra University; 2013:7–9 pp.
12. Rahiman MA, Adheena CV, Anitha R, Deepa N, Kumar GM, Rajasree MS. Bilingual OCR system for printed documents in Malayalam and English. In: ICECT 2011 – 2011 3rd International Conference on Electronics Computer Technology; 2011, vol 3:40–5 pp.
13. Prasad BK, Sanyal G. A model approach to off-line English character. Int J Sci Res Publ 2012;2:2–7.
14. Seeri SV, Pujari JD, Hiremath PS. PNN based character recognition in natural scene images. Bonfring Int J Softw Eng Soft Comput 2016;6:109–13.
15. Nasser EF, Karim AA. English character recognition from video stream based on bag of visual words (BOVW). J Madent Alelem Coll 2018;10:12–29.
16. Ansari NN. License number plate recognition using template matching. Int J Comput Trends Technol 2016;35:175–8.
17. Kaur I, Singh KJ. Printed text recognition system for multi-script image. Int J Signal Process Syst 2016;4:411–16.

18. Mohammed AZ, Mohammed EA, Aaref AM. Real-time driver awareness detection system. IOP Conf Ser Mater Sci Eng 2020; 745:1–13.
19. Ram Mohan Rao GIRK, Ramesh B. Font and size identification in Telugu printed document. Int J Eng Res Dev 2013;6:92–110.
20. Priyanka N, Pal S, Mandal R. Line and word segmentation approach for printed documents. RTIPPR Int J Comput Appl IJCA 2010;1:30–6.
21. Hariharan B, Arbelaez P, Girshick R, Malik J. Object instance segmentation and fine-grained localization using hypercolumns. IEEE Trans Pattern Anal Mach Intell 2017;39:1–14.
22. Cai Z, Vasconcelos N. Cascade R-CNN: delving into high quality object detection. Cvpr 2018:6154–62.
23. Ramaiah NP, Ijjine EP, Mohan CK. Illumination invariant thermal face recognition using convolutional neural network. In: 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES). Kozhikode, India: IEEE; 2015. https://doi.org/10.1109/SPICES.2015.7091490.
24. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998;86:2278–324.
25. Goodfellow I, Courville A. Deep learning. MIT Press; 2016. http://www.deeplearningbook.org.
26. Lin G, Shen W. Research on convolutional neural network based on improved Relu piecewise activation function. Procedia Comput Sci 2018;131:977–84.
27. Wani MA, Bhat FA, Afzal S, Khan AI. Advances in deep learning. Kacprzyk J, editor. Studies in Big Data. India: Springer Singapore; 2020. Chapter 2; Basics of supervised deep learning; 13–29 pp.
28. Cristina M, Neri G, Osiris O, Villegas V, Guadalupe V, Sánchez C. A convolutional neural network for handwritten digit recognition 2 materials and methods. Int J Comb Optim Probl Informatics 2020; 11:97–105.
29. Krizhevsky A, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems; 2012:1097–105 pp.
30. Chung H, Lee SJ, Park JG. Deep neural network using trainable activation functions. In: Proceedings of the International Joint Conference on Neural Networks; 2016, vol 2016:348–52 pp.
31. Nwankpa CE, Ijomah W, Gachagan A, Marshall S. Activation functions: comparison of trends in practice and research for deep learning [Internet]. Glasgow, UK, arXiv; 2018. Available from: http://arxiv.org/abs/1811.03378v1.
32. Kouretas I, Paliouras V. Simplified hardware implementation of the softmax activation function. In: 2019 8th International Conference on Modern Circuits and Systems Technologies, MOCAST 2019. https://doi.org/10.1109/MOCAST.2019.8741677.
33. Benaddy M, El Meslouhi O, Es-saady Y, Kardouchi M. Handwritten tifinagh characters recognition using deep convolutional neural networks. Sens Imaging 2019;20:1–17.
34. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors [Internet]. Canada, arXiv; 2012. Available from: http://arxiv.org/abs/1207.0580.
35. Alla S, Adari SK. Beginning anomaly detection using python-based deep learning. Apress. California: Springer; 2019: 319–60 pp. Appendex A, Intro to keras.
36. Bharath V, Rani NS. A font style classification system for English OCR. In: Proceedings 2017 International Conference Intelligent Computing and Control. I2C2 2017; 2017:1–5 pp. https://doi.org/10.1109/I2C2.2017.8321962.
37. Mohammed AJ, Al-Khaffaf HSM. Font recognition of English letters based on distance profile features. Sci J Uni Zakho 2020;8:66–71.
38. Samadiani N, Hassanpour H. A neural network-based approach for recognizing multi-font printed English characters. J Elect Sys Inform Technol 2015;2:207–18.
39. Wei TC, Sheikh U, Rahman AAHA. Improved optical character recognition with deep neural network. In: Proceedings – 2018 IEEE 14th International Colloquium on Signal Processing and its Applications, CSPA 2018; 2018:245–9 pp. https://doi.org/10.1109/CSPA.2018.8368720.