

DEEP LEARNING METHOD FOR HANDWRITING RECOGNITION

Ayşe Ayvacı^{1,*}, Abdullah Erdal Tümer^{2,3}

¹Necmettin Erbakan University, Institute of Natural Sciences, Department of Computer Engineering, Konya/Turkey

²Necmettin Erbakan University, Faculty of engineering and architecture, Department of Computer Engineering, 42195, Konya/Turkey

³Kyrgyz – Turkish Manas University, Bishkek, Kyrgyzstan

ABSTRACT

The advancement of technology nowadays resulted into documents, such as forms and petitions, being filled out in computer and digital environment. Yet in some cases, documents are still preserved in traditional style, on print. Due to its distinct proportions, however, its storage, sharing and filing has become a complication. The relocation of these written documents to digital environment is therefore of great significance. In this view, this study aims to explore methodologies of digitizing handwritten documents. In this study, the documents converted to image format were pre-processed using image processing methods. These operations include dividing lines of the document into image format, dividing into words which then divided into characters, and finally, a classification operation on the characters. As classification phase, one of the deep learning methods is the Convolution Neural Network method is used in image recognition. The model was trained using the EMNIST dataset, and in the character, dataset created from the documents at hand. The dataset created had a success rate of 87.81%. Characters classified as finishers are sequentially combined and the document is transferred to the computer afterwards.

ARTICLE INFO

Research article

Received: 3.01.2021

Accepted: 28.06.2021

Keywords:

Character recognition, convolutional neural network, deep learning, handwriting recognition, image processing

*Corresponding author

1. Introduction

Writing process is a complex activity consisting of cognitive, kinesthetic, and perceptual-motor components [1]. Writing is the written form of the words, feelings, and thoughts in the language. Handwriting is unique for each individual. It is a tool developed long ago to expand human knowledge and facilitate communication [2].

As technology advances, numerous documents are filled in the digital environment. Yet there are still transactions which are still being done on paper. The task of transferring these written documents to computer can be done by one or more people. In this circumstance, handwriting recognition

systems have been developed to transfer written documents automatically to computer. Handwriting recognition is the process of defining and making sense out of letters, numbers, and symbols that are written by hand on papers, tablets, and smartphones, using computer systems.

There are two methods available in handwriting recognition systems - offline method and online method. In the offline method (Figure 1a), the writing is used as a picture. It contains the spatial brightness of the image [2]. On the other hand, the online method (Figure 1b) contains spatial-timer values of characters.

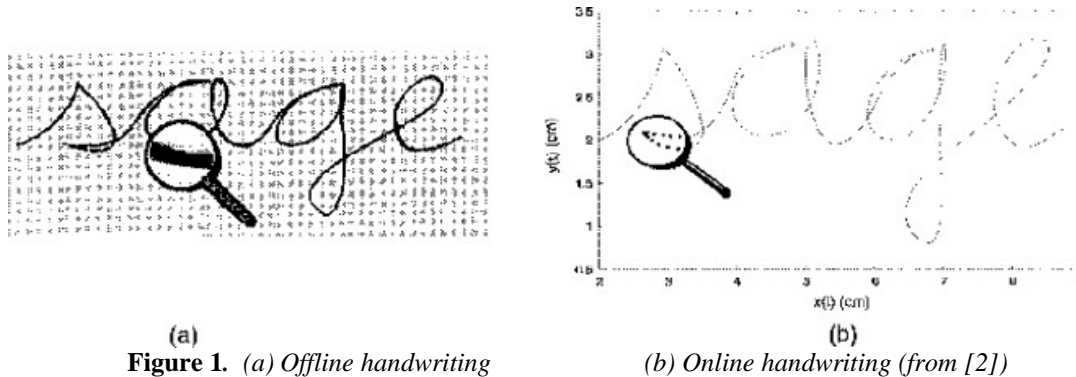


Figure 1. (a) Offline handwriting

(b) Online handwriting (from [2])

The character recognition rates of online methods are more successful than offline methods. Offline methods have 95%, 85%, and 78% recognition for 10, 100, 1000 words, and above respectively [2, 3]. In online methods, the success rate decreases as the number of words requested to be recognized increases. An accuracy rate of 80% is achieved at 21,000 words and above [2, 4].

The convenience of character recognition technology can be utilized in several business areas:

- Checks sent to banks are automatically recognized and the necessary account transactions are performed electronically;
- Bringing doctors' handwriting to an understandable format. In a study conducted on this subject, 500 prescription samples taken from doctors were identified handwriting recognition and keywords from the doctor's handwriting. With these specified words, it is understood what the desired drug is in the prescription report written by the doctor. Hidden Markov Model was used as the model in the project. Multilayer perceptron (MLP) was also used to increase performance [5]
- Recognition of addresses on letters and automatic parsing by postal code. In the project carried out by CEDAR for USPS (United States Postal Service), the aim is to correctly separate envelopes according to the address where letters will automatically go in post offices [6].

Handwriting characters are more difficult to distinguish than optical characters. Consequently, accuracy rates in current studies are less than 90%. Some of difficulties in handwriting recognition include:

- Having too many different fonts, numbers, and symbols
- Difference in writing style
- Cursive or interconnected writing of letters
- Pencil or paper used

- Difference of letters' shapes and sizes depending on individual's writing style and speed.

Despite these challenges, researchers have conducted many studies for more successful character recognition. For instance, Fanany [7] used a workflow algorithm and a machine learning model to recognize handwriting characters in a form document. Convolutional Neural Network (CNN) is used for feature extraction of characters and Support Vector Machines (SVM) is used for classification. The study presents 83.37% accuracy in ten different test form documents.

Mahapatra, Choudhury, Karsh [8], in another study, used K-Nearest Neighbor (KNN), SVM-based classifiers such as CNN, hybrid KNN-SVM and ν -SVM on different handwriting datasets such as EMNIST, Devanagari handwriting character and Kannada-MNIST. The successes of the developed models have been compared. As a result, EMNIST, Devanagari Handwritten Character, and Kannada-MNIST achieved a success rate of 89.02%, 86.67%, and 95.3%, respectively.

Saha and Jaiswal [9] also tested the EMNIST and UCI Devanagari datasets with the CNN classification model. In this model, success rate of EMNIST dataset was 79.3% while it is 93% for the UCI Devanagari dataset.

Cohen, Afshar, Tapson, and Van Schaik [10], evaluated the EMNIST dataset with an OPIUM- based classifier. The model achieved success rate of 78.02% in the EMNIST balanced dataset.

Baykal, Aktaş and Yildiz [11], created an active offline verification system by selecting the signature characteristic, which is one of the biometric characteristics.

In the study, recognition of handwriting characters was performed offline. The current study, unlike the other studies in the literature, did not use a separate algorithm to extract features on images. Instead of a separate algorithm, a single

method was used to both extract and classify character traits using the CNN method. The CNN method was used for classification and feature extraction. The Balanced part of the EMNIST [12] was used to train the model. The dataset used in the study was drawn from the Kaggle data store (from <https://www.kaggle.com/crawford/emnist>). EMNIST Balanced dataset consists of 131,600 characters and 47 classes. The model trained with the EMNIST Balanced dataset, was also tested on an independently generated dataset.

The study consists of two stages: pre-processing and classification. The following sections consist of pre-processing, classification, experimental study and result consecutively.

2. Pre-Processing

Having characters of the same size and shape affects the success rate in recognition. Therefore, preprocessing was applied to the letters and numbers in the data and test set. Pre-machining is done in two ways:

- Pre-processing:* With this process, the text on the paper is separated into characters. The separated characters are then tested correctly. For this, following operations are performed:
 - Letters on paper are loaded into the computer in picture format. The installation process is done with the help of a scanner or camera. An original image is given in Figure 2.

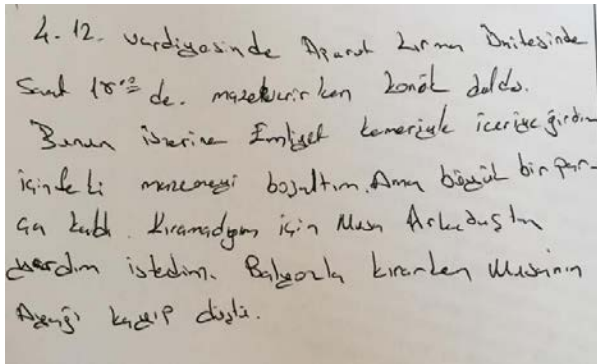


Figure 2. Original image

- The image is first converted into a gray-level image. Figure 3 is an example of an image that is converted into grayscale.



Figure 3. Grayscale image

The grayscale image is converted into a black-and-white image by applying the threshold. Afterwards, the image looks as in Figure 4.



Figure 4. Black-and-white image

Through the contour extraction process in the black-and-white image, the lines are determined by drawing a curve that connects all continuous points that have the same color and density along the borderlines. Fig. 5 also shows an image divided into lines.

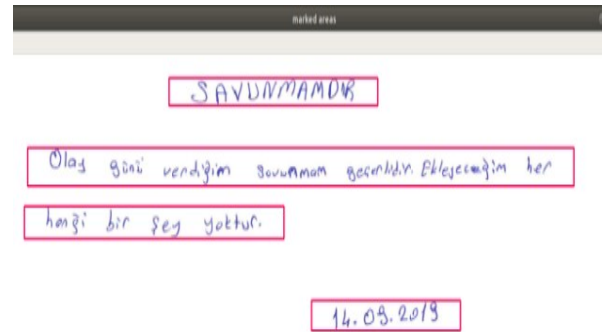


Figure 5. Image divided into rows

Application of re-contour extraction process of lines resulted into division of document into words. Figure 6 includes an image divided into sentences.

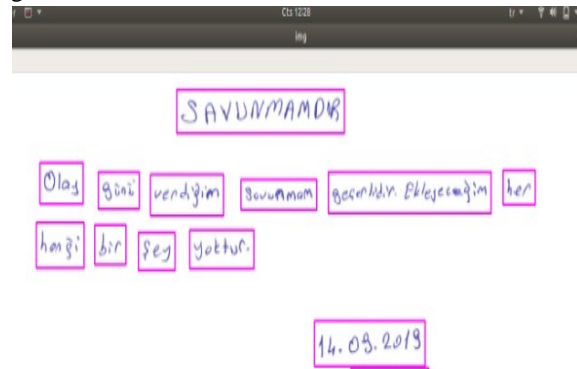


Figure 6. Image divided into words

- The reserved words over the contour extraction process are completed. Throughout this process, the boundaries of the character within the words are determined
- Over characters whose boundaries are set, the ROI (area of interest) operation is performed. This process is the process of obtaining a new image that has only character

boundaries according to the pixel values of the corresponding region. The resulting new images are 28x28 pixels. The new images obtained are characters in black and white format, as in Figure 7.



Figure 7. Characters allocated to test

- ii. *Normalization:* After pre-processing, normalization is applied to the EMNIST dataset. Through this process, the images are brought to the same size and type.

3. Classification

In this study, a well-known deep learning algorithm CNN is used. CNN algorithms are applied in many different areas such as natural language processing (NLP) [13, 14], biomedical processing [15, 16], especially image [17, 18] and audio processing [19, 20]. It is a deep learning algorithm that has the best classification success, especially in the field of image processing. CNN processes the image in various layers. The layer structure of the CNN model is shown in Figure 8. These layers [21] are:

- Convolutional layer- Which is used to determine properties;
- Non-Linearity layer- Which introduces non-linearity to the system;
- Pooling (Down-sampling) layer - Which reduces the number of weights and controls compliance;
- Flattening layer – Which prepares data for the classical neural networks; and
- Fully-Connected layer – Which is the standard neural network used in classification.

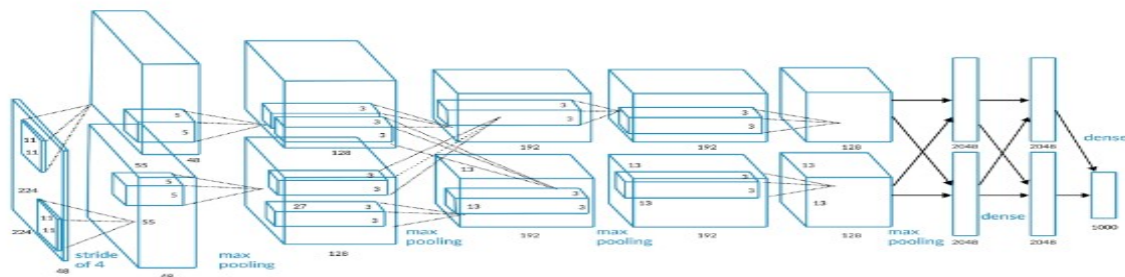


Figure 8. CNN's layer structure (from www.missinglink.ai)

The Convolution layer is used to detect the properties of the picture. This layer uses a low or high- level filter to extract properties. These filters are usually matrices of odd numbers. By moving the filter over the image and using matrix multiplication, its properties are detected. In Figure 9, the example of convolution filter and its application are shown. After filters, zero values are added in order for the image to not lose its original size.

The Non-linearity layer comes after convolution layers. Since all layers can be linear functions, the

Neural Network acts as a single perceptron, which means that the results are calculated linearly. In this layer, one of the activation functions is used. The best results about the speed of neural network training are the Rectifier (Relu) function, therefore this function was used. The Relu function is shown in Equation 1 below:

$$RELU\ Func. = -f(x) = max \tag{1}$$

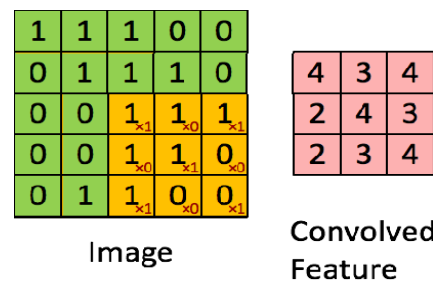


Figure 9. Convolution filter (form www.medium.com/@tuncerergin)

The Pooling layer is inserted between successive convolutional layers. This layer is designed to reduce the spatial size of the representation and the number of parameters and calculations within the network. The most used is max-pooling. In max-pooling, a filter is created first. This filter moves around the image and takes the largest

number in the area it occupies. Figure 10 shows an example of a max-pooling process.

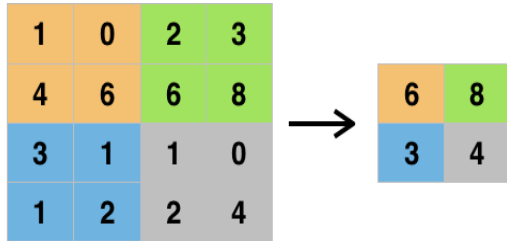


Figure 10. Max-pooling process (from www.medium.com/@tuncerergin)

The Flattening layer, the last layer, prepares the input data for the Fully Connected Layer. Neural networks take inputs from a one-dimensional array. The data in a network is in a state of matrices, that was converted into a one-dimensional array, in convolutional and pool layers. Figure 11 shows an example of the last layer translated into a one-dimensional array.

The Fully Connected Layers take data from the flattening layer and performs learning through the Neural network. During the training, the dropout method (shown in Figure 12) is used to prevent underfitting, unnecessary memorization, and overfitting proms. Decreasing the number of connections within the network improves training performance.

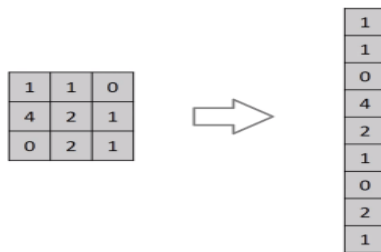


Figure 11. Flattening layer (from www.medium.com/@tuncerergin)

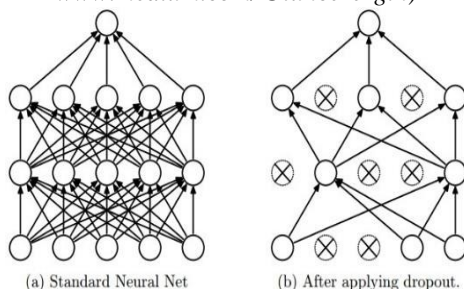


Figure 12. Dropout layer (from www.medium.com/@tuncerergin)

4. Experimental Work

The study, a computer with a working i5 processor with 8 GB of RAM is also written in Python

2.6. The OpenCV library was used for pre-processing. The Keras Library was used in deep learning. The study consisted of two phases. The first stage is the acquisition of characters, the creation of a CNN algorithm to classify the characters that we get in the second stage, and the training and testing of characters.

In the first stage, the letters written on paper are first thrown into the computer environment in PDF format. The document in PDF format is converted to JPG format, which is the image format. Images 2 using OpenCV library. the pre-processing steps described in the section were performed, and thus the articles written in the image were separated the first line by line, the words for each line were separated sequentially, each separated word was divided into characters, and the separation of characters in the image-shaped text was performed. The character is normalized and the character images are translated into black-and-white (binary system).

In the second stage, model training was first performed to classify the characters. The Balanced part of the EMNIST dataset was used as the dataset for the training of the model, and Convolutional neural networks were used as the deep learning architecture. To create the model; after creating layers of the CNN algorithm, the model was trained on the EMNIST dataset 2 times according to different epochs and batch size values. The trained models were tested using the detected letters in the documents using the image processing method. Figure 13 shows block diagram of the study.

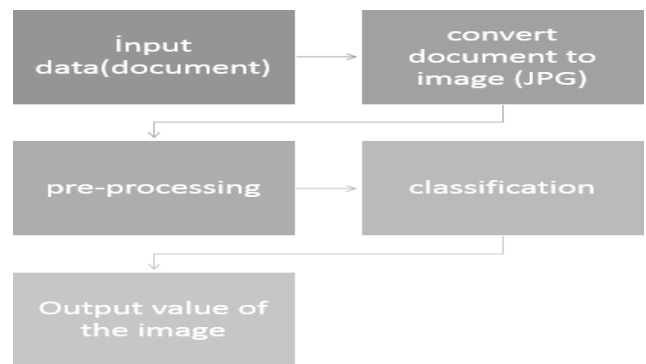


Figure 13. Block diagram of the study

This study used handwritten letters and numbers consisting of 112,799 training sets and 18,799 test sets in 28x28 pixels sizes found in the EMNIST Balanced dataset. The 10% of the training set is reserved for validation. In the training of the developed model, 4 CONV2D layers, 4 Max-Pooling layers, 1 Flatten layer, 1 Dropout layer and 2 Dense layers were used. Layers of the model created are shown in Figure 14 and Figure 15.

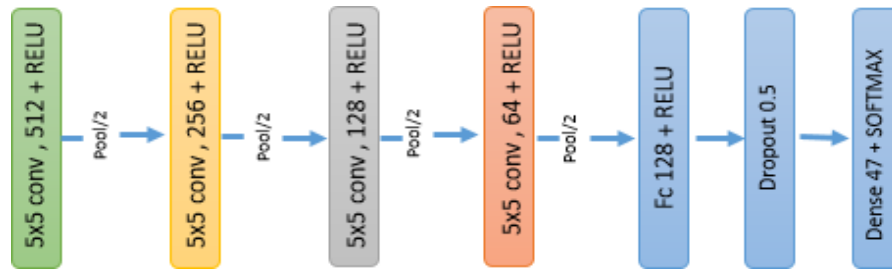


Figure 14. CNN's block diagram used in the study

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 28, 28, 512)	13312
max_pooling2d_1 (MaxPooling2)	(None, 14, 14, 512)	0
conv2d_2 (Conv2D)	(None, 14, 14, 256)	3277056
max_pooling2d_2 (MaxPooling2)	(None, 7, 7, 256)	0
conv2d_3 (Conv2D)	(None, 7, 7, 128)	819328
max_pooling2d_3 (MaxPooling2)	(None, 3, 3, 128)	0
conv2d_4 (Conv2D)	(None, 3, 3, 64)	73792
max_pooling2d_4 (MaxPooling2)	(None, 1, 1, 64)	0
flatten_1 (Flatten)	(None, 64)	0
dense_1 (Dense)	(None, 128)	8320
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 47)	6063
Total params: 4,197,871		
Trainable params: 4,197,871		
Non-trainable params: 0		

Figure 15. Description of the layers of the model used in the study.

The Model was trained twice, and the trainings lasted about 20 hours. The first training was performed with 5 epoch and 512 batch size, and the second training was performed with 10 epoch and 1024 batch size. Result of these two pieces of training showed that the second training with a rate of 87.81%

success is the best score recorded. The values of the two pieces of training are also given in Table 1.

According to the model's 10 epoch and batch size 1024, its graphics are shown in Figure 16 and Figure 17.

Table 1. Study results

Epochs/ batch_size	Acc	Loss	Val_acc	Val_loss	Test_acc	Test_loss
5/512	%84.64	%37	%85.30	%35	%86	%36
10/1024	%87.81	%34	%87.61	%34	%87	%35

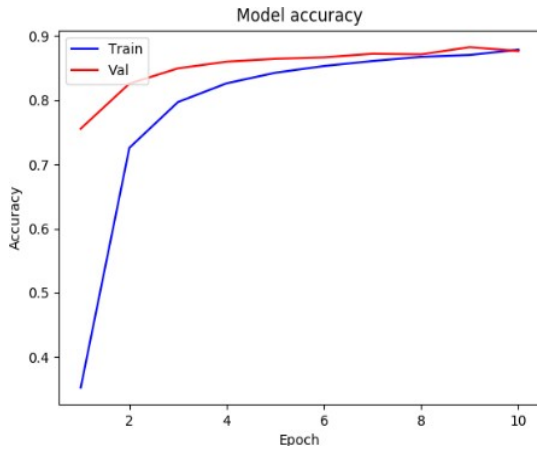


Figure 16. Success graph of the model.

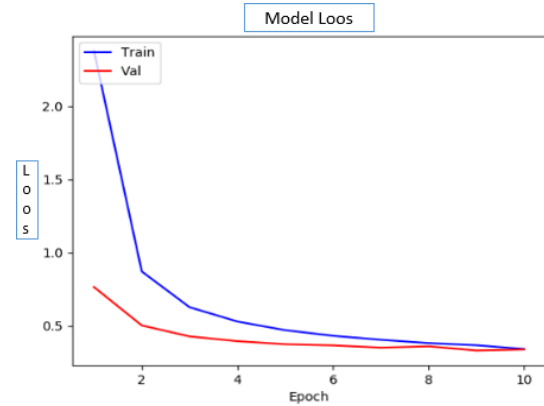


Figure 17. Error graph of the model

In this study, the independently created datasets were tested separately in both two trained models. The data tested has values of 28x28 pixels. Figure 18 shows the results of the characters that are obtained from handwritten documents in trained models.

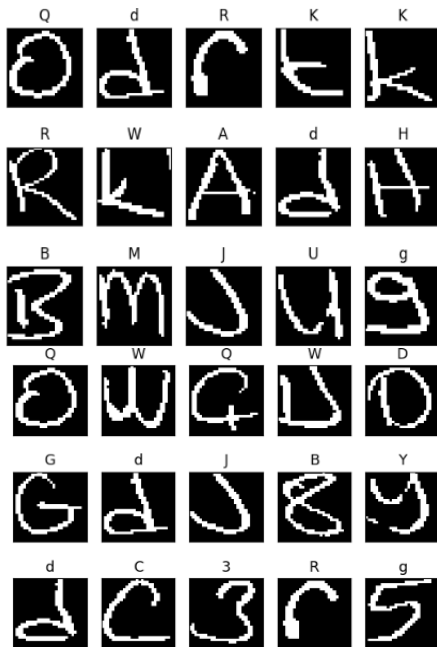


Figure 18. The outputs of the characters tested in the model.

5. Conclusions

This paper primarily discussed the recognition of characters from handwritten documents to computer environment. By means of encoding, these characters were recorded into

computer media. For this purpose, the CNN model has been created throughout the operation. Pre-processing, slicing, and character recognition were then performed altogether as part of the procedure. At the end of the process, the obtained characters - 10 epoch and 1024 batch size trained models – both achieved a success rate of 87.81%.

The success rates of studies conducted using the Balanced part of the EMNIST dataset and different classifier algorithms are given in Table 2. Looking at Table 2, the method we used was more successful than other methods.

Table 2. Success rates on the EMNIST dataset for CNN, Linear Classifier, and OPIUM- based Classifier

Algorithm Name	Success Rate
OPIUM- based	78.02%
Linear Classifier	50.93%
CNN	87.81%

In reference to data in Table 1, it is foreseen that a more successful model can be obtained by increasing the values used in epoch and batch size. For this reason, future studies and its corresponding tests and trainings is recommended to carry out increase on the number of layers in order to improve the success rate of handwriting recognition, and the algorithm as well.

References

[1] B. Engel-Yeger, L. Nagauker-Yanuv, and S. Rosenblum, "Handwriting performance, self-reports, and perceived self-efficacy among children with dysgraphia," *American Journal of Occupational*

- Therapy*, vol. 63, no. 2, pp. 182-192, 2009.
- [2] R. Plamondon, and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 1, pp. 63-84, 2000.
- [3] S. N. Srihari, "High-performance reading machines," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1120-1132, 1992.
- [4] G. Seni, R. K. Srihari, and N. Nasrabadi, "Large vocabulary recognition of on-line handwritten cursive words," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 7, pp. 757-762, 1996.
- [5] P. P. Roy, A. K. Bhunia, A. Das *et al.*, "Keyword spotting in doctor's handwriting on medical prescriptions," *Expert Systems with Applications*, vol. 76, pp. 113-128, 2017.
- [6] S. N. Srihari, S.-H. Cha, H. Arora *et al.*, "Handwriting identification: Research to study validity of individuality of handwriting and develop computer-assisted procedures for comparing handwriting," *Technical Report CEDAR-TR-01-1*, 2001.
- [7] Fanany, M. I. Handwriting recognition on form document using convolutional neural network and support vector machines (CNN-SVM). In 2017 5th international conference on information and communication technology (ICoIC7) (pp. 1-6). IEEE. (2017, May).
- [8] D. Mahapatra, C. Choudhury, and R. K. Karsh, "Mahapatra, D., Choudhury, C., & Karsh, R. K. Handwritten character recognition using knn and svm based classifier over feature vector from autoencoder. In International Conference on Machine Learning, Image Processing, Network Security and Data Sciences (pp. 304-317). Springer, Singapore. 2020, July.
- [9] P. Saha, and A. Jaiswal, "Handwriting Recognition Using Active Contour," *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, pp. 505-514: Springer, 2020.
- [10] S. A. Gregory Cohen, Jonathan Tapson, and Andre van Schaik, "EMNIST: an extension of MNIST to handwritten letters," 2017.
- [11] Baykal, B., Aktaş, T. Ö., & Yildiz, O. "Makine Öğrenmesi Yöntemleri İle Otomatik Çevrimdışı İmza Tanıma ve Doğrulama Sistemi". In 2017 International Artificial Intelligence and Data Processing Symposium (IDAP) (pp. 1-5). IEEE. 2017, September.
- [12] NIST, "The EMNIST Dataset," 2017, April.
- [13] Sun, B., Yang, L., Zhang, W., Dong, P., Young, C., Dong, J., & Lin, M. Demonstration of applications in computer vision and nlp on ultra power-efficient cnn domain specific accelerator with 9.3 tops/watt. In 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) (pp. 611-611). IEEE. (2019, July).
- [14] Akhtyamova, L., Ignatov, A., & Cardiff, J. A Large-scale CNN ensemble for medication safety analysis. In International Conference on Applications of Natural Language to Information Systems (pp. 247-253). Springer, Cham. 2017, June.
- [15] M. Cho, J. Ha, C. Park *et al.*, "Combinatorial feature embedding based on CNN and LSTM for biomedical named entity recognition," *Journal of Biomedical Informatics*, vol. 103, pp. 103381, 2020.
- [16] Momeni, A., Thibault, M., & Gevaert, O. Dropout-enabled ensemble learning for multi-scale biomedical data. In International MICCAI Brainlesion Workshop (pp. 407-415). Springer, Cham. 2018, September.
- [17] M. Amin-Naji, A. Aghagolzadeh, and M. Ezoji, "Ensemble of CNN for multi-focus image fusion," *Information fusion*, vol. 51, pp. 201-214, 2019.
- [18] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep CNN with batch renormalization," *Neural Networks*, vol. 121, pp. 461-473, 2020.
- [19] Z. Mushtaq, S.-F. Su, and Q.-V. Tran, "Spectral images based environmental sound classification using CNN with meaningful data augmentation," *Applied Acoustics*, vol. 172, pp. 107581, 2020.
- [20] Y. Su, K. Zhang, J. Wang *et al.*, "Environment sound classification using a two-stream CNN based on decision-level fusion," *Sensors*, vol. 19, no. 7, pp. 1733, 2019.
- [21] T. Ergin, "Convolutional Neural Network (ConvNet yada CNN) nedir, nasıl çalışır?," 2018, October