Khazar Journal of Science and Technology
Khazar University Press 2022
Volume 6 №2 2022, 32-55
DOI: 10.5782/2520-613.202.6.2.32

Recognition of Handwritten Azerbaijani Letters using Convolutional Neural Networks

Heydar Rzayev, Amir Masoud Rahmani, Behnam Kiani Kalejahi

Faculty of Engineering and Applied Sciences,
Khazar University, Azerbaijan
Corresponding author: heydar.rzayev@khazar.org

Abstract

Technology advancements have made it possible to fill out documents such as petitions and forms electronically. However, in some circumstances, hard copies of documents that are difficult to share, store, and save due to their rigid dimensions are still used to preserve documents in the conventional manner. It is crucial to convert these written documents into digital media because of this. From this view point, this goal of this study is to investigate various methods for the digitalization of handwritten documents. In this study, image processing methods were used to pre-process the documents that were converted to image format. These operations include splitting the image format of the document into the lines, separating them into words and characters, and then classifying the characters. Convolutional Neural Networks, which is used for image recognition, is one of the deep learning techniques used in classification. The Extended MNIST dataset and the symbol dataset created from the pre-existing documents are used to train the model. The success rate of the generated dataset was 88.72 percent.

Keywords: Image processing; character recognition; handwriting recognition; deep learning; convolutional neural network.

Introduction

In the modern era, with the advancement of technology, more and more documents are being filled in the computer environment. Nevertheless, there are many documents that are still on the paper. Transferring these to the digital environment is done with the assignment of one or more people. To eliminate manual work, handwriting recognition systems have been developed to automatically transfer such documents to the digital environment. Handwriting recognition is the identification of letters, numbers and symbols written on media such as paper, tablet and phone by computer systems and making them meaningful.

Handwritten characters are more difficult to distinguish than optical characters. Therefore, the accuracy rates in current studies are below 90%. There are many variations that affect the recognition of handwriting (e.g., natural variations, word size, defects in the image). These variations cause some difficulties in the recognition of words or characters. Despite these difficulties, many studies have been carried out and continue to be conducted in this area.

Fortunately, in some languages such as Chinese, Arabic and Japanese, the handwriting recognition systems are quite good. However, on the contrary, there are not 2 enough studies in the field of Azerbaijani handwriting recognition. The level of success in the studies carried out is quite low.

There are two methods in handwriting recognition systems: online and offline. In the online method, the two-dimensional coordinates of the writer's strokes are used while writing on tablets or special screens. The second method is called the offline method, where the text is used as a picture. This is a well-known problem, and many methods can be used to recognize the image.

In this study, offline handwritten texts were translated into picture format, and the recognition process was carried out. Since Azerbaijani has an infinite dimensional vocabulary, character-based recognition was used in the study. Instead of a separate algorithm, both the feature extraction and recognition processes of the characters were done with a single method, which is the Convolutional Neural Network (CNN) method.

The first step is to create a model for recognition and then train this model. A model has been created, and training has been made for the recognition of the characters. The EMNIST (NIST, 2017, April) data set was used in the training of the model which is obtained from the Kaggle site. The EMNIST dataset contains a large number of handwritten letters and numbers. All the characters in the EMNIST dataset were trained in 2D with the CNN method. The data is given as a direct input to the CNN model without any feature extraction on the characters beforehand. The distinctive features of the characters were made with the CNN method during the training of the model, and the recognition was carried out in this way.

The second stage is to obtain the characters from the documents and perform the recognition process on these characters. The data used in the recognition process was created from documents written by many people around us. Then, a preprocessing step was done on these documents, and their characters were separated. And then a recognition process was performed using the model trained with the CNN method.

Handwriting Recognition

Handwriting recognition is the conversion of handwritten text on other devices, such as paper documents, photographs, and touch screens, into an understandable digital format. It is examined under two main headings as online and offline handwriting recognition. The online handwriting recognition is the process of converting handwritten text into a digital format while the user is writing. The offline handwriting recognition is the process of converting handwritten text into digital format after the user has finished writing.

Online Handwriting Recognition

Online handwriting recognition (OHWR) is a system for the automatic processing of writing using a hand or pen by tablets or special screens. In OHWR, the size of the vocabulary has a direct impact on the performance of the system. As the number of words increases, the recognition speed and performance of the system decrease (Bilgin Taşdemir, 2018).

Many methods and approaches have been found about this system since the beginning of the 1960s. These methods can be studied in three classes: understanding of text written with a first-class screen and stylus. Verification of the person's identity from signatures in second-class handwriting. Third grade is to use the neuromotor features of handwriting to design systems for education and rehabilitation (Plamondon & Srihari, 2000).

In OHWR systems segmentation process is used to get the basic structure of character recognition. Two methods are used. The first method is line detection (Hennig et al., 1996). Thanks to this method, word segmentation and adjustment of parts that do not contain text are done. The second method focuses on splitting the input into individual characters or even sub-characters. The biggest problem with splitting words is determining the beginning and end of individual characters. The most common approaches used for this are unsupervised learning (Hébert et al., 1999), (Plamondon & Srihari, 2000) and data-based knowledge-based methods (Hennig et al., 1997).

Offline Handwriting Recognition

Offline handwriting recognition is a system that detects characters on a document. When using this system, first the document is digitized by transferring it to the computer environment. Then, the document is divided into paragraphs, sentences and words, respectively (Şekerci, 2007)0. Offline handwriting recognition is divided into a holistic or analytical strategy:

- Holistic Strategy: It is processed directly on the word without slicing. The longer the word, the lower the recognition rate.

- Analytical Strategy: It is processed on the document after slicing it to paragraphs, sentences, words, or characters. This strategy in character recognition follows a certain sequence of operations. These are (Srihari et al., 2001):

  - Pre-processing
  - Segmentation
  - Feature extraction
  - Classification
  - Post-processing

Pre-processing

The pre-processing stage consists of the operations such as normalization, noise removal, and reference line determination. Noise-removal can be achieved by techniques such as filters, noise models, and morphological operations. Normalization is the process of standardizing characters by removing differences in typefaces. Detecting and using reference line is also important because it prevents some characters from mixing with each other. For example, while the characters 'g' and '9' may be confused with each other, when their position relative to the reference line is considered, the confusion of these two characters with each other will be eliminated (Steinherz et al., 1999), (Şekerci, 2007).

Segmentation

The segmentation phase is the process of separating words into letters, characters and numbers. One popular method is to split the document first into lines, then words and lastly into characters. Another method used is to first divide the document into as small pieces as possible and then combine them using the Hidden Markov Models method (Hidden Markov Model-HMM) (Arica & Yarman-Vural, 2001), (Şekerci, 2007).

Feature extraction

The feature extraction is an important part of the handwriting recognition process. It is used to determine which pixels are relevant and which aren't. Different methods can be used at this stage. These methods are: applying transformations such as Fourier and Wavelet, histogram or projection-based methods, or defining letters as a set of simple shapes such as lines, 8 curves, and corners (Arica & Yarman-Vural, 2001).

Classification

After the feature extraction, we classify characters or numbers with the data we have. Many algorithms have been used for the classification process:

- K-NN
- libSVM
- Artificial Neural Networks (ANN)
- Convolutional Neural Networks (CNN)
- Bayesian Classifier

K-NN, LibSVM, and Artificial Neural Network algorithms are algorithms used for character recognition treated as an analytical strategy. CNN is the algorithm used both as a holistic strategy and for character recognition.

K-NN

K-Nearest Neighbours (K-NN) algorithm is a machine learning algorithm used in clustering problems. The performance of the algorithm is affected by the number of k clusters, the values of the initially selected cluster centres and the similarity measurement criteria (Zouhal & Denoeux, 1998). The choice of the value of k is important in the K-NN algorithm. The K value is usually chosen as 1. Having a K value of 1 may cause problems in recognizing similar characters. An example of this is the recognition of the letters 'e' and 'c'. Therefore, the K value is chosen as 3 for the clustering of lowercase letters and as 5 for the clustering of uppercase letters (Bektaş et al., 2016).

libSVM

It is a library developed for Support Vector Machines (SVM). Unlike SVM, which is used to estimate two-class data, it allows the classification of multi-class data (Bektaş et al., 2016).

ANN

ANNs mimic the structure of a living organism's nervous system. The process involves receiving numerous pieces of information, analysing them individually and gauging their relative importance in relation to others—then making decisions based on these factors by choosing an appropriate output action. It is used for solving complex problems that cannot be solved by conventional, linear approaches alone. A simple ANN model is shown in Figure 1.

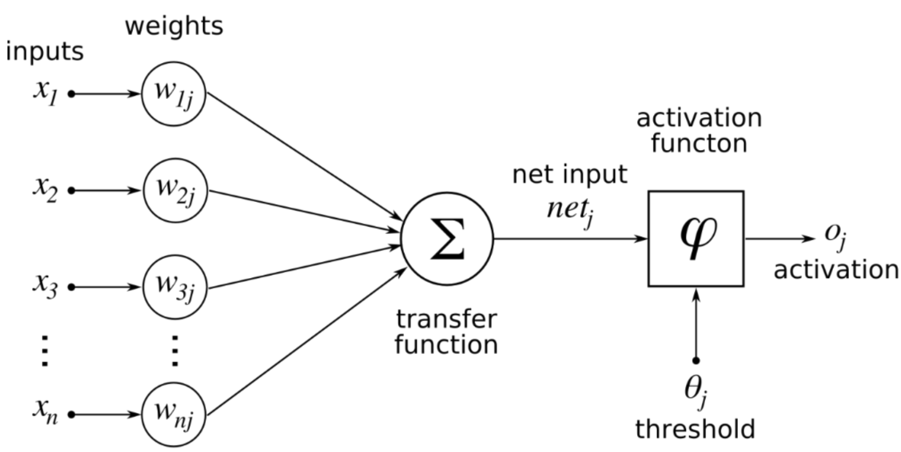

Figure 1. ANN

CNN

Convolution Neural Network (CNN) is a deep learning architecture. The algorithm takes place in two stages. In the first stage, the image, which is given as the input value to the algorithm, is processed in the convolution layer and the important features of the image are extracted in this layer and thrown into a matrix. In the second stage, classification is performed on the matrix obtained in the convolution layer by using the multiple artificial neural network algorithm. The character recognition steps of the CNN algorithm are shown in Figure 2.

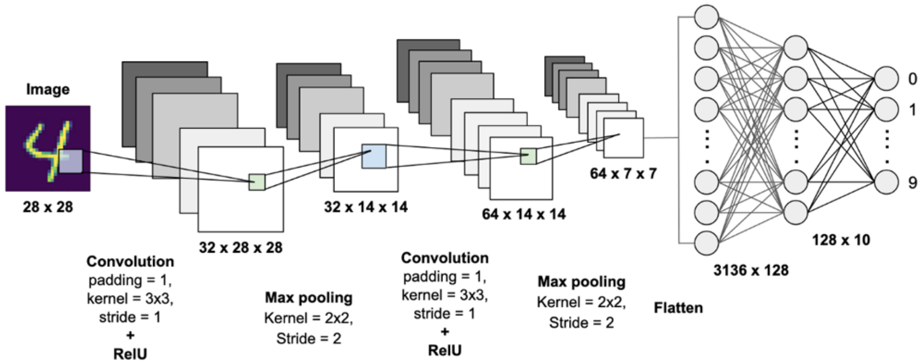

Figure 2. Character Recognition with CNN

Bayesian Classifier

Naive Bayes classifier is a classification method developed by taking Bayes probability theorem as an example. It is an approach that calculates the probability that a new data belongs to which of the existing classes, using the entire dataset for a new classification.

The probabilities of a new data sample X with no class value are calculated using Eq. (1) for each class using the Bayesian classifier. Considering that there are N classes, $S = (s_1, s_2, \ldots, s_n)$. For each data in the dataset, data m-dimensional feature vectors; It is shown as $X = (x_1, x_2, \ldots, x_m)$.

$$P\left(\frac{s_i}{X}\right) = \frac{P\left(\frac{X}{s_i}\right) * P(s_i)}{P(X)} \qquad (1)$$

The class that X data belongs to is found by using Eq. (2).

$$X \in s_i \rightarrow P\left(\frac{s_i}{X}\right) = maxP\left(\frac{s_i}{X}\right); i \in |S| \qquad (2)$$

Post-processing

Post-processing is used to increase the accuracy of classification. There are many methods for verification. One of them is the verification of letters using databases containing combinations of 2 or 3 letters. Another method is to increase the accuracy of the sentence by using a high-level formal grammatical model. The main idea behind the post-processing is to improve the accuracy of the result by using additional information from other sources such as additional datasets or knowledge about the problem domain.

Material and Method

Technical Information

The hardware information of the computer used for the training and testing of the model developed in the study is given in Table 1.

Table 1. Computer hardware information

Processor:	Intel Core i5-7360U, 2.30 GHz, 4 MB
Graphics card:	Intel Iris Plus Graphics 640
Memory:	8 GB
Persistent memory:	1 TB
Operating system:	Ubuntu 14 LTS

Python 2.7 was used as the programming language in the software part. OpenCv Library was used for image processing, Tensorflow and Keras libraries were used for artificial intelligence.

Dataset

Training dataset

The data set used in the training of the model was taken from the site called Kaggle. The EMNIST dataset contains handwritten letters and numbers consisting of 28 x 28 pixel images. There are six different panels provided in this dataset. A brief summary of the dataset (NIST, 2017, April) is as follows:

- EMNIST ByClass: 814,255 characters. 62 class value.
- EMNIST ByMerge: 814,255 characters. 47 class value.
- EMNIST Balanced: 131,600 characters. 47 class value.
- EMNIST Letters: 145,600 characters. 26 class value.
- EMNIST Digits: 280,000 characters. 10 class value.
- EMNIST MNIST: 70,000 characters. 10 class value.

Looking at Table 2, a data summary of six different partitions of the EMNIST dataset is given. Looking at the chart, how much data is reserved for testing and training, the total number of data and its availability for validation are given.

Table 2. EMNIST dataset and organization

Name	Category	Training	Test	Verification	Total
BY_CLASS	62	697,932	116,323	No	814,255
BY_MERGE	47	697,932	116,323	No	814,255
BALANCED	47	112,800	18,800	Yes	131,600
DIGITS	10	240,000	40,000	Yes	280,00
LETTERS	37	88,800	14,800	Yes	103,600
MNIST	10	60,000	10,000	Yes	70,000

Test dataset

This data set was created with documents written by many people around us. It is a data set created for recognizing handwritten characters on paper. The dataset contains 10 documents. These documents are handwritten by different people and transferred to the computer environment by scanner or camera.

Handwriting Recognition with Developed Deep Learning Algorithm

The processes performed in this study are divided into the following steps. The applications in each step are developed using the Python language. The steps of the developed application are shown in Figure 3.

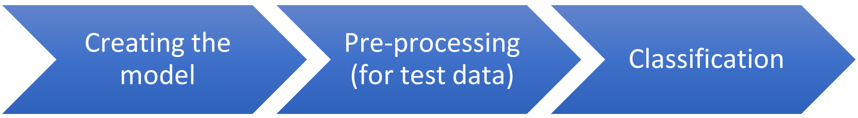

Figure 3. Steps of the developed application

Creating the model

In the study, the deep learning method was used to classify the characters. A model was created using the Convolutional Neural Network architecture of deep learning and the classification process was made using this model.

Deep Learning

Deep learning performs feature extraction with the help of multiple nonlinear layers. All consecutive layers take the output of the previous layer as input (Deng & Yu, 2014), (Şeker et al., 2017). Deep learning is basically a method based on learning based on the values of the pixels that best represent the data. In other words, it works by extracting the most important traits of the character in the image and classifies it by using these traits. This method is called the feature extraction method. In deep learning, different algorithms are used to extract the best pixels on the image instead of manually extracting data.

Looking at the history of deep learning, the first algorithm for supervised deep-feed multilayer perceptrons was introduced by Ivakhnenko and Lapa in 1965 (Ivakhnenko & Lapa, 1966). In this study, the best features in the layers are determined by statistical methods and sent to the next layer. Backpropagation is not used to train the networks end-to-end.

Second, deep learning architecture was proposed by Fukushima in 1979. It was developed by being inspired by the nervous system of vertebrates. Fukushima's networks, similar to today's networks, contain multiple bending and pool layers (Fukushima et al., 1983).

Although deep learning architectures have emerged in previous years, a successful deep neural network application has been developed for the first time by Yann LeCun et al. on mailbox texts (LeCun et al., 1989), (Şeker et al., 2017). After this work, Yann LeCunn applied convolutional networks together with backpropagation to classify handwritten digits (MNIST) using the “LeNet” network (Le Cun et al., 1989). With this study, the first operations in character recognition were started.

Today, with the developing technology, the interest in this field has increased. The term deep learning was first introduced by Igor Aizenberg et al. in 2000 (Aizenberg et al., 2000).

Later, in this area, Geoffrey Hinton described how to train a multilayer feedforward neural network and how to do a supervised back propagation method in his article in 2006 (Hinton, 2007)0.

With the development of computers, the acceleration of GPU and CPU, and training of deep networks without pre-training have emerged. Ciresan et al. used this method in applications such as traffic signs, medical imaging and character recognition (Ciresan et al., 2011).

Krishevsky, Sutskever and Hinton used similar architectures in 2012 and in their work using GPU, they developed the "dropout" layer to reduce memorization and used this method in their applications (Hinton et al., 2012)0.

From the past to the present, many applications have been developed with deep learning and continue to be developed. In this process, a lot of work has been done in the field of deep learning and different architectures have been developed. There are 6 architectures of deep learning. These are:

- Convolutional Neural Networks
- Recurrent Neural Networks
- Long / Short Term Memory
- Restricted Boltzmann Machines
- Deep Belief Networks
- Autoencoders and Denoising Autoencoders

Convolutional Neural Networks: Convolutional neural networks (CNN) are a different type of multi-layer perceptron (MLP). The first CNN network in history is the LeNet architecture founded by Yann LeCun in 1988 (Le Cun et al., 1989).

CNN algorithms include image and audio processing (Mushtaq et al., 2020), (Su et al., 2019), natural language processing (NLP) (Akhtyamova et al., 2017), (Sun et al., 2019)0, biomedical image processing (Cho et al., 2020), (Momeni et al., 2018). It is a deep learning algorithm that has the best classification success, especially in the field of image processing.

- CNN architecture processes data in various layers. The layer structure of the CNN model is as in Figure 4. These layers (Ergin, 2018, October) are:
- Convolutional layer – Used to extract features on the image.
- Non-Linearity layer – ReLU layer - It is the layer where the activation process is located. It uses a nonlinear function in the system.
- Pooling (Down-sampling) layer – Size reduction and compatibility check
- Flattening layer – Prepares one-dimensional vector data for Standard Neural Network.
- Fully-Connected layer – Uses Standard Neural Network for classification.

The Convolution layer is used to detect features on the image. This is done by using low-level or high-level filters that are smaller than the size of the image to extract the feature on the image in this layer. These filters are usually matrices of odd numbers. By moving these filters over the image and using matrix multiplication, features are tried to be detected. Zero values are added so that the picture does not lose its original size after the filters.

The Non-Linearity layer comes after the Convolution layers. This layer is also known as the activation layer and one of the activation functions is used. The purpose of this layer is to prevent the model from learning negative values or failing to grasp some features by using activation functions. Usually as an activation function; Nonlinear functions such as ReLU, tanh and sigmoid are used. Since the Rectifier (ReLU) function gives the best results in terms of speed, this function is mostly used in training.

The pooling layer is added between successive convolution layers. This layer is used to reduce its dimensionality. Thanks to this layer, unnecessary features are removed and focus on important features is ensured. In CNN models, two different pooling techniques are generally used as Max (Maximum) and Average (Average). In the first technique, Maxpooling, a filter is first created and this filter is moved over the picture and takes the largest number in the area it covers. In the second technique, Averagepooling, a filter is created again and this filter is moved over the picture to get the average number of pixels in the area it covers.

The Flattening layer is the layer that prepares the input data for the Fully Connected Layer. It takes the input data of neural networks as a one-dimensional array. These data are the data in the matrices from the Convolution and Pooling layers. The flattening layer converts this data to a one-dimensional array.

The Fully Connected Layer receives the data as processed in the Flattening layer and converts it into a one-dimensional array and performs the learning process with the classical neural network method. During the training, the Dropout method is used to prevent problems such as under fitting, unnecessary memorization and over fitting.

Training the Created Model

For the deep learning algorithm in the classification step, an application developed in Python using the Keras library was made. The Balanced part of the EMNIST dataset was used for training. The layers of the model are also created using the Convolutional neural network architecture. Layers of the developed model; It consists of 4 CONV2D layers, 4 MaxPooling2 layers, 1 Flatten layer, 1 Dropout layer and 2 Dense layers. The output of the layer structure of the model is shown in the application developed in Figure 4.

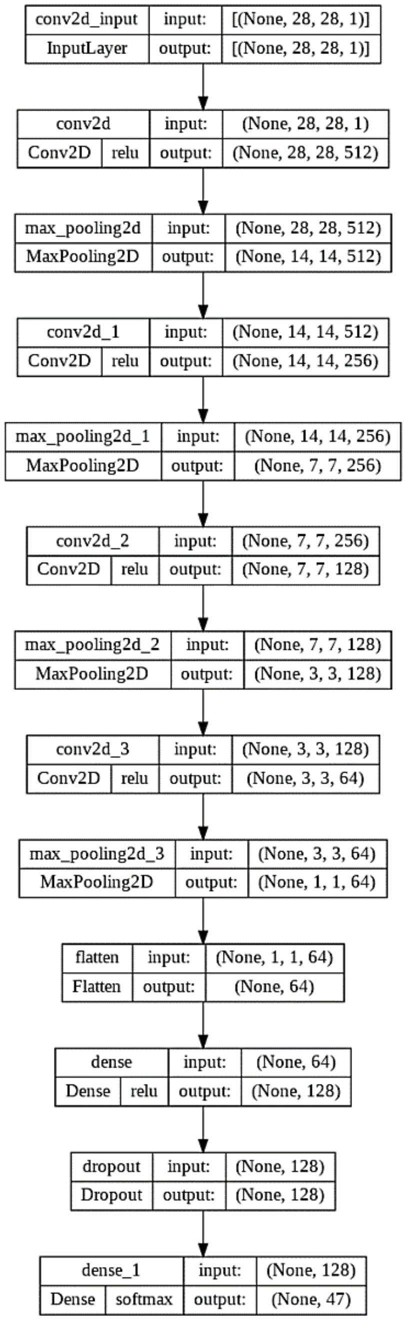

Figure 4. Description of the layers of the model used in the study

The descriptions of the layers of the model used in this study are given below:

- CONV2D: It is the Convolution layer. It is the layer used to detect features on the image.

- MaxPooling2: The pooling layer is added between successive convolution layers. This layer is used to reduce dimensionality.

- Flatten: The Fatting layer is the layer that prepares the input data for the Fully Connected layer.

- Dropout: It is used to prevent under fitting, unnecessary memorization and over fitting problems during training.

- Dense: It is the Fully Connected layer.

The visual representation of the created model is shown in Figure 5.

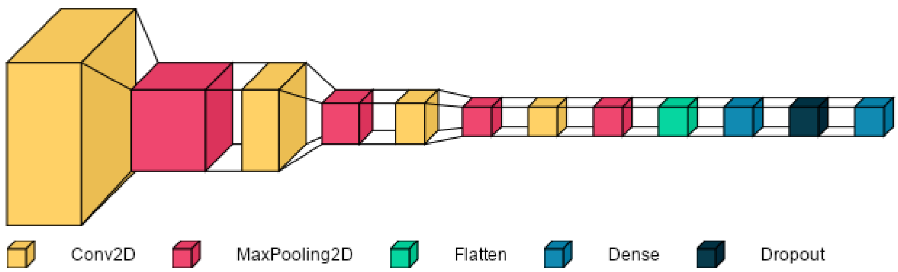

Figure 5. Visual representation of the model

The training of the model was done using the Balanced part of the EMNIST dataset. The model was trained 4 times according to different epochs and batch_size values. 10% of the training set is reserved for validation. The epochs and batch_size values used in the training of the model are as follows;

- 5 epochs and 512 batch_size

- 5 epochs and 1024 batch_size

- 10 epochs and 512 batch_size
- 10 epochs and 1024 batch_size

Pre-processing

In this study, a character-based handwriting recognition system has been developed. For this reason, some operations were performed to separate the documents in the data set created for the test into their characters. An application has been developed using the OpenCV Library for pre-processing. Pre-processing steps are explained in order. These steps are:

1. The texts on the paper were saved in the computer environment in a ".png" format after being scanned with the help of a scanner or camera.

2. The original image was first converted to a greyscale image.

3. The greyscale image is converted to a binary image, that is, to a black-and-white image, by applying the threshold function.

4. Lines were determined by applying morphological transformations on the black-and-white image.

5. The document is divided into words by applying the morphological transformation process on the lines again.

6. The normalization process was applied to the data set created with the existing characters.

At this stage, the application of the steps explained above was developed using the Python programming language. This source code was run with the help of Algorithm 1 given in Table 3; All documents in the test dataset are separated into characters.

Table 3. Pre-processing algorithm pseudocode

Algorithm 1: (Pre-processing)
Input: N images in the data set **Output:** Dataset containing black and white toned character images
Step 1. Get Started **Step 2.** Loop (N images): **1.** Upload the image from the folder. **2.** Convert the image to a grey scale. **3.** Convert the grey scale image to a black and white image. **4.** Determine the rows by applying a morphological operation.
Step 3. Loop (As many as detected rows)**:** **1.** Separate the image into words by applying the dilation process on the image that is divided into lines.
Step 4. Loop (As many as detected words)**:** **1.** Identify characters by applying morphological operations and contour subtraction in the image divided into words. **2.** Perform ROI operation on the characters that have been located **3.** Apply normalization to character images.
Step 5. Finish

The block diagram of the preprocessing step of the study is given in Figure 6.

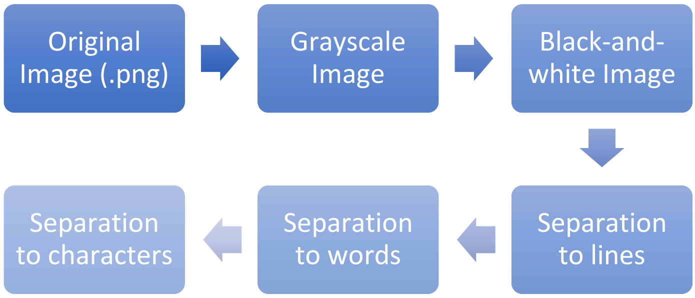

Figure 6. Block diagram of pre-processing

At this stage, classification was performed on the pre-processed characters. The classification application was developed using the Python language. In practice, the documents in the test dataset are separated into characters by applying the pre-processing step. Then, the prediction processes of the models were made using these characters. The block diagram of these steps is shown in Figure 7. Using the model's "predict" function, predictions were made on the characters.

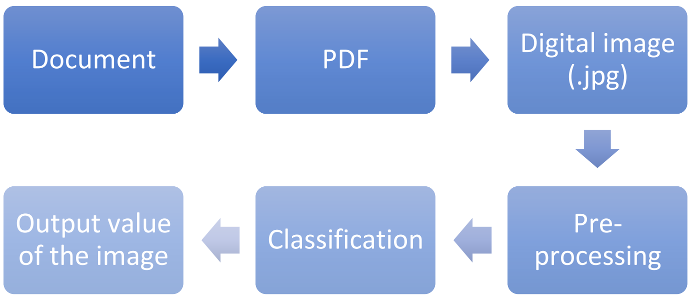

Figure 7. Block diagram of classification

Results

Epoch and 512 Batch_size

The first training of the model was performed using the EMNIST dataset and 5 epochs and 512 batch_size values. The desired success in training has not been achieved.

Epoch and 1024 Batch_size

The model was trained a second time using the EMNIST dataset with 5 epochs and 1024 batch_size values. The success (accuracy) and loss (loss) graph of the training is given in Figure 8 and Figure 9.

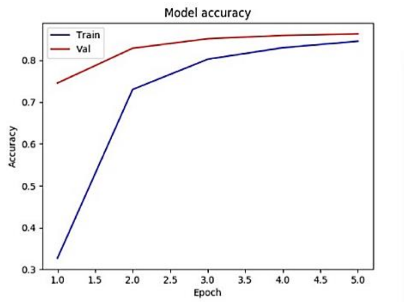

Figure 8. The accuracy graph of the model after training with 5 epochs and 1024 batch_size

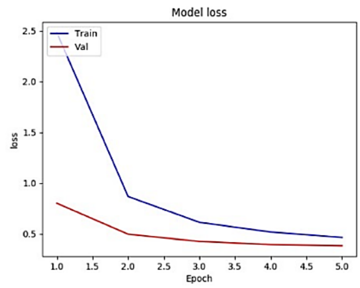

Figure 9. The loss graph of the model after training with 5 epochs and 1024 batch_size

Epoch and 512 Batch_size

The graphs of success and loss rates according to the 10 epoch and 512 batch_size values of the model are given in Figure 10 and Figure 11.

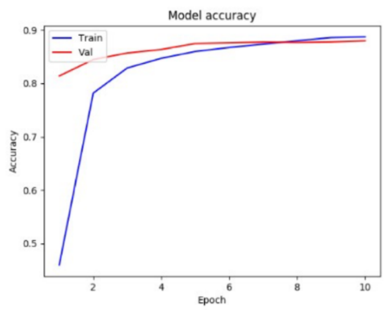

Figure 10. The accuracy graph of the model after training with 10 epochs and 512 batch_size

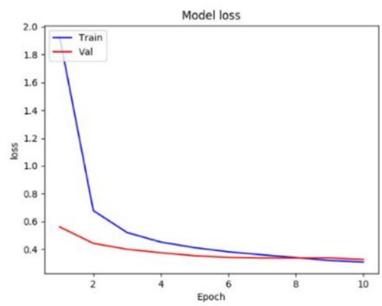

Figure 11. The loss graph of the model after training with 10 epochs and 512 batch_size

Epoch and 1024 Batch_size

The graphs of success and loss rates according to the 10 epoch and 1024 batch_size values of the model are given in Figure 12 and Figure 13.

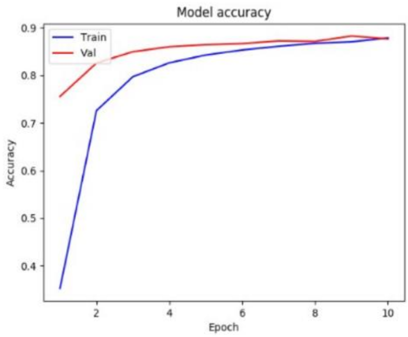

Figure 12. The accuracy graph of the model after training with 10 epochs and 1024 batch_size

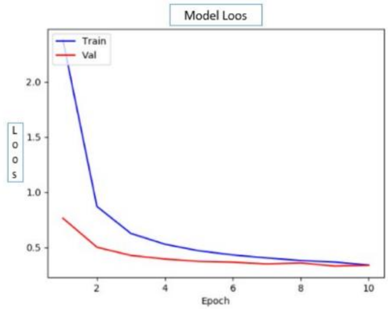

Figure 13. The loss graph of the model after training with 10 epochs and 1024 batch_size

Conclusion and Recommendations

Conclusion

In the study, a character recognition application was carried out by using a deep learning algorithm on handwritten character images. Since Azerbaijani has an infinite dimensional vocabulary, a character-based recognition process was carried out in the study. In the study, instead of only the border regions of the characters, a 2-dimensional classification process was made as a whole. Convolutional Neural Network algorithm, which is a deep learning architecture, is used for character recognition.

The study was carried out in two stages. The first stage is the creation and training of a model for the classification of characters. The second step is that the trained model performs the recognition on the handwriting. The operations carried out in these two stages are given in detail in section 2.

In the study, the model was created according to the Convolutional Neural Network architecture and was made using the Balanced part of the EMNIST dataset for its training.

The model, which was created according to the Convolution Neural Network architecture, was trained on the EMNIST dataset according to 4 different values. The learning results of the model are given in Table 4.

Table 4. The results according to the four different trainings of the model

Epochs/ batch_size	Acc	Loss	Val_acc	Val_loss	Test_acc	Test_loss
5/512	84.64%	37%	85.30%	35%	86%	36%
5/2014	84.54%	46%	86.29%	38.40%	86.48%	35.81%
10/512	88.72%	30.78%	87.98%	32.68%	87.50%	33.42%
10/1024	87.81%	34%	87.61%	34%	87%	35%

By training the model according to different epoch and batch_size values, the most successful model for character classification has been tried to be created. When Table 4 is examined, the increase in the epoch value in the training made according to different epoch values with the same batch_size values also increased the success of the model. However, the increase in the batch_size value in the training made according to different batch_size values with the same epoch values caused the success rate of the model to decrease.

Looking at the chart, the most successful training was done according to 10 epoch and 512 batch_size values with a rate of 88.72%.

After the training of the model, testing was done on handwritten documents on paper. At this stage, the handwritings on the paper were separated into their characters by going through the pre-processing described in section 2. A recognition process was performed on the separated characters according to the most successful training of the model.

Recommendations

In this study, character-based handwriting recognition was performed using the Convolutional Neural Network algorithm. A more successful recognition process can be done by using different deep learning algorithms and parameters on the data set used.

In future studies, a different recognition process will be performed on the same data set by using deep learning architectures in another way. And in order to increase the accuracy of classification, using databases containing combinations of 2 or 3 letters, the letters will be verified, the characters will be combined and the document will be transferred to the computer environment as a whole.

References

Aizenberg, I. N., Aizenberg, N. N., & Vandewalle, J. (2000). Multiple-Valued threshold logic and multi-valued neurons Multi-Valued and Universal Binary Neurons (pp. 25-80): Springer.

Akhtyamova, L., Ignatov, A., & Cardiff, J. (2017). A Large-scale CNN ensemble for medication safety analysis. Paper presented at the International Conference on Applications of Natural Language to Information Systems.

Arica, N., & Yarman-Vural, F. T. (2001). An overview of character recognition focused on off-line handwriting. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 31(2), 216-233.

Bektaş, B., Babur, S., Turhal, U., & Köse, E. (2016). Makine öğrenmesi yardimiyla optik karakter tanima sistemi optical character recognition system via machine learning. 5th International Printing Techologies Symposium İstanbul Üniversitesi - İstanbul - 04-05 Kasim / November.

Bilgin Taşdemir, E. F. (2018). A large vocabulary online handwriting recognition system for Turkish.

Cho, M., Ha, J., Park, C., & Park, S. (2020). Combinatorial feature embedding based on CNN and LSTM for biomedical named entity recognition. Journal of Biomedical Informatics, 103, 103381.

Ciresan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2011). Convolutional neural network committees for handwritten character classification. Paper presented at the 2011 International Conference on Document Analysis and Recognition.

Deng, L., & Yu, D. (2014). Deep learning: methods and applications. Foundations and trends in signal processing, 7(3–4), 197-387.

Fukushima, K., Miyake, S., & Ito, T. (1983). Neocognitron: A neural network model for a mechanism of visual pattern recognition. IEEE transactions on systems, man, and cybernetics(5), 826-834.

Hébert, J.-F., Parizeau, M., & Ghazzali, N. (1999). Learning to segment cursive words using isolated characters. Paper presented at the Proc. of the Vision Interface Conference.

Hennig, A., Sherkat, N., & Whitrow, R. (1996). Zone estimation for multiple lines of handwriting using approximating spline functions. Paper presented at the Proceedings of the Fifth International Workshop on Frontiers in Handwriting Recognition IWFHR-5.

Hennig, A., Sherkat, N., & Whitrow, R. J. (1997). Recognising letters in on-line handwriting using hierarchical fuzzy inference. Paper presented at the Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on.

Hinton, G. E. (2007). Learning multiple layers of representation. Trends in cognitive sciences, 11(10), 428-434.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

Ivakhnenko, A. G. e., & Lapa, V. G. (1966). Cybernetic predicting devices: PURDUE UNIV LAFAYETTE IND SCHOOL OF ELECTRICAL ENGINEERING.

Le Cun, Y., Jackel, L. D., Boser, B., Denker, J. S., Graf, H. P., Guyon, I., Hubbard, W. (1989). Handwritten digit recognition: Applications of neural network chips and automatic learning. IEEE Communications Magazine, 27(11), 41-46.

Le Cun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Handwritten digit recognition with a back-propagation network. Paper presented at the Proceedings of the 2nd International Conference

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), 541-551.

Momeni, A., Thibault, M., & Gevaert, O. (2018). Dropout-enabled ensemble learning for multi-scale biomedical data. Paper presented at the International MICCAI Brainlesion Workshop.

Mushtaq, Z., Su, S.-F., & Tran, Q.-V. (2020). Spectral images based environmental sound classification using CNN with meaningful data augmentation. Applied Acoustics, 172, 107581.

Plamondon, R., & Srihari, S. N. (2000). Online and off-line handwriting recognition: a comprehensive survey. IEEE Transactions on pattern analysis and machine intelligence, 22(1), 63-84.

Şeker, A., Diri, B., & Balık, H. H. (2017). Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. Gazi Mühendislik Bilimleri Dergisi (GMBD), 3(3), 47- 64.

Şekerci, M. (2007). Birleşik ve eğik Türkçe el yazısı tanıma sistemi.

Srihari, S. N., Cha, S.-H., Arora, H., & Lee, S. (2001). Handwriting identification: Research to study validity of individuality of handwriting and develop computer-assisted procedures for comparing handwriting. Technical Report CEDAR-TR-01-1.

Steinherz, T., Rivlin, E., & Intrator, N. (1999). Offline cursive script word recognition– a survey. International Journal on Document Analysis and Recognition, 2(2-3), 90-110.

Su, Y., Zhang, K., Wang, J., & Madani, K. (2019). Environment sound classification using a two-stream CNN based on decision-level fusion. Sensors, 19(7), 1733.

Sun, B., Yang, L., Zhang, W., Dong, P., Young, C., Dong, J., & Lin, M. (2019). Demonstration of Applications in Computer Vision and NLP on Ultra Power- Efficient CNN Domain Specific Accelerator with 9.3 TOPS/Watt. Paper presented at the 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW).

Zouhal, L. M., & Denoeux, T. (1998). An evidence-theoretic k-NN rule with parameter optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 28(2), 263-271.