Available online at www.dergipark.org.tr/en

INTERNATIONAL ADVANCED RESEARCHES and ENGINEERING JOURNAL

Journal homepage: www.dergipark.org.tr/en/pub/iarej

e-ISSN: 2618-575X

International Open Access

Volume 04
Issue 03

December, 2020

Research Article

Turkish sign language digits classification with CNN using different optimizers

Onur Sevli [a,*] ***and Nazan Kemaloğlu*** [b]

[a]*Burdur Mehmet Akif Ersoy University, Faculty of Engineering, Computer Engineering Department, 15030, Burdur, Turkey*
[b]*Burdur Mehmet Akif Ersoy University, Center of Information Technologies, 15030, Burdur, Turkey*

ARTICLE INFO

Article history:
Received 08 March 2020
Revised 25 May 2020
Accepted 10 July 2020

Keywords:
Convolutional neural networks
Sign language recognition
Turkish sign language

ABSTRACT

Sign language is a way for hearing-impaired people to communicate among themselves and with people without hearing impairment. Communication with the sign language is difficult because few people know this language and the language does not have universal patterns. Sign language interpretation is the translation of visible signs into speech or writing. The sign language interpretation process has reached a practical solution with the help of computer vision technology. One of the models widely used for computer vision technology that mimics the work of the human eye in a computer environment is deep learning. Convolutional neural networks (CNN), which are included in deep learning technology, give successful results in sign language recognition as well as other image recognition applications. In this study, the dataset containing 2062 images consisting of Turkish sign language digits was classified with the developed CNN model. One of the important parameters used to minimize network error of the CNN model during the training is the learning rate. The learning rate is a coefficient used to update other parameters in the network depending on the network error. The optimization of the learning rate is important to achieve rapid progress without getting stuck in local minimums while reducing network error. There are several optimization techniques used for this purpose. In this study, the success of four different training and test processes performed with SGD, RMSprop, Adam and Adamax optimizers were compared. Adam optimizer, which is widely used today with its high performance, was found to be the most successful technique in this study with 98.42% training and 98.55% test accuracy.

1. Introduction

Sign language is a unique communication method that enables hearing-impaired people to communicate with each other and people with no hearing impairment. Contrary to popular belief, there are no universal patterns of sign language that a limited number of people know today. Each society has its national sign language and differs from nation to nation [1]. This makes communication difficult. Hearing and speech impaired people have less writing skills, so choosing to write instead of sign language is not comfortable [2]. Sign languages are translated into spoken or written languages by other people who know that sign language. However, it becomes difficult due to the low number of people who know sign language and differences between sign languages. It is necessary to translate sign languages among themselves and into normal speaking and writing languages practically and accurately.

Sign language translation process based on seeing with the human eye and interpreting has become feasible by computers with technological developments. This method, also known as computer vision, is a technology that provides fast results by imitating the work of the human eye in a computer environment [3]. The most preferred of the models using this technology today is deep learning, which is a sub-branch of machine learning. Deep neural networks are mathematical models of the human nervous system and consist of millions of configurable parameters. Convolutional neural networks (CNN), which is a sub-branch of deep learning, produce extremely successful results in image recognition and classification problems

* Corresponding author. Tel.: +90-248-213-4130
E-mail addresses: onursevli@mehmetakif.edu.tr *(O. Sevli)*, nkemaloglu@mehmetakif.edu.tr *(N. Kemaloğlu)*
ORCID: 0000-0002-8933-8395 *(O. Sevli)*, 0000-0002-6262-4244 *(N. Kemaloğlu)*
DOI: 10.35860/iarej.700564

and have been applied successfully in the recognition of human movements in recent years. The number of studies using CNN in the field of sign language recognition has been increasing recently.

Bheda and Radpour proposed a classifier CNN model for American Sign Language and achieved 82.5% accuracy [4]. Koller et al. presented a deep learning model that interprets sign language mouth shapes [5]. Huang et al. proposed a 3D CNN model for sign language recognition [6]. Pigou et al. achieved 91.7% accuracy in their CNN-based classification study for Italian sign language [7]. Hasan and Ahmad achieved 96.4% accuracy in their machine learning-based study on the Bengal sign language database consisting of 11 digits and 16 words [8]. Agarwal and Thakur classified the sign language dataset consisting of digits 0-9 using support vector machines [9]. Oyewole et al. used Principal Component Analysis and Artificial Neural Networks in their classification study for Nigerian sign language [10]. Besides, previous studies include classifications for Chinese [11], Korean [12], Albanian [13], Arabic [14-15], Mexican [16] and Tamil [17] sign languages.

Aran et al .; developed a tool that teaches Turkish sign language. The system can recognize complex signs including both hand gestures and head gestures and expressions, and verbal and animated feedback is provided to the user. In performance tests, it was observed that the system showed 99% success in recognizing hand signals and 85% in recognizing head and facial expressions [21].

Beşer et al. classified the sign language dataset consisting of digits 0-9 that created by Ankara Ayrancı Anatolian High School students. The data was divided into 70% education and 30% test. In the study using Capsule Neural Network for classification, a success of 94.52% was achieved [22].

Particle swarm optimization, artificial bee colony, and genetic algorithm were used for classification in the study with the same data set. As a result of the experimental studies, it was seen that the artificial bee colony gave the highest accuracy with a success of 98.09% [23].

In this study, a CNN model was developed for Turkish sign language analysis, and the dataset created by Ankara Ayrancı Anatolian High School students was classified. The dataset contains 2062 images of 10 different digits between 0 and 9. Four different optimizers were used to optimize the learning rate used in updating the parameters of the CNN network model, and the accuracy values of the four separate learning processes were compared. The following sections describe data set features, CNN architecture, learning rate optimization methods, experimental studies, and results.

2. Material and Method

2.1 Dataset

The dataset was created by 218 students studying at Ankara Ayrancı Anadolu High School and shared publicly [18]. The dataset contains 2062 images consisting of 10 digits between 0 and 9. Image examples from the data set are shown in Figure 1.

The main features of the dataset:

- Image sizes: 64x64
- Color space: Grayscale
- Number of classes: 10 (Digits: 0-9)

Figure 1. Image examples from the dataset

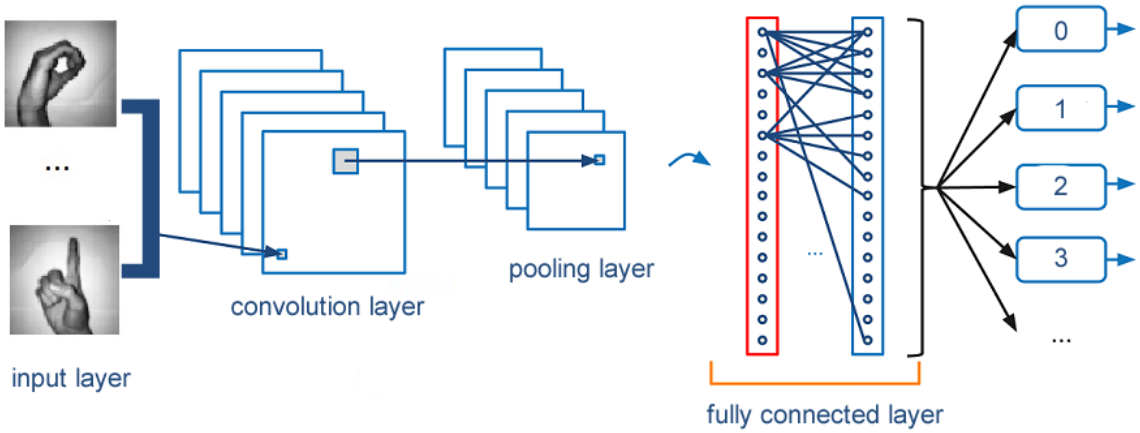

Figure 2. The general structure of CNN

2.2 Convolutional Neural Networks (CNN)

Machine learning is a sub-branch of artificial intelligence and it is a technique that enables machines to produce new solutions based on previous solutions. Artificial neural networks are mathematical models based on the functioning of the human brain. Deep learning is the processing of data in deep neural networks for feature extraction. One of the deep learning methods, CNN is used in computer vision, classification, recognition, regression, and many other fields. They are deep neural networks that have gained popularity with their success in image classification.

CNN is a computer vision technique that allows us to automatically extract and define features from images. It contains input, convolution, pooling, fully connected, and prediction layers. The general structure of CNN is shown in Figure 2.

The input layer consists of images to be processed. Convolution is the process of applying special filters on the image pixels and multiplying the input pixel values by the filter values. The image features are extracted with the help of the filters applied to the input.

Pooling is the reduction process performed on the input image matrix. It is carried out by taking the average or maximum values. A large number of pixels in the pool are reduced to a single value. The size of the image is reduced by moving the pool over all the image pixels. In this way, the image can be processed more efficiently. The average and maximum pooling processes with a 2x2 pool on the 4x4 input matrix are shown in Figure 3.

One of the other important parameters in neural networks is the activation function. The activation function increases the nonlinearity of the inputs and produces a stable result for the next layer. An appropriate activation function should be selected according to the procedure performed on the neural network.

The fully connected layer provides the transition to the neural network that will make the classification. The number of neurons in the output of the classification layer must be the same as the number of classes in the dataset.

2.3 Learning Rate Optimization

One of the methods used to increase the CNN model performance is learning rate optimization. The learning rate is a coefficient used in updating the network parameters depending on the amount of error that occurs in the learning process of the network. If the learning rate is too small, network parameters are updated in very small steps and the process takes a long time. The high learning rate can cause the network to miss the optimum point that minimizes the error. Therefore, learning rate optimization is very important.

The learning rate is a dynamically adjusted parameter depending on the situation. Different optimization algorithms are used to optimize the learning rate. Stochastic Gradient Descent (SGD), RMSprop, Adam, and AdaMax are some of these optimizers.

SGD performs a parameter update for each training sample. It is usually a fast technique. It makes an update at every step. The high variance caused by frequent updates causes oscillations in the loss function. While this helps to find better local minimums, it can sometimes lead to missing the global minimum.

RMSprop is a gradient-based optimization technique. It offers a solution to the problem of reducing the learning rate excessively.

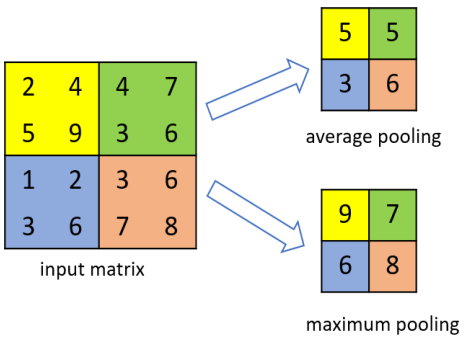

Figure 3. The pooling process

Instead of using all the values obtained from the squares of the past slopes, it restricted the amount [19]. RMSprop helps to reduce oscillations in the loss function.

Adam [20], is an adaptive learning rate optimization technique designed specifically to train deep neural networks. Adam is a combination of RMSprop and SGD, which are two other methods used in this field, with momentum. It is widely used because of the performance it provides.

AdaMax is the version of Adam optimizer that replaces the root mean square (RMS) feature with the infinity norm of previous gradients. AdaMax is generally suitable for infrequent parameter updates and noisy gradients.

Optimizers can perform differently for various situations, such as the data set used, the structure of the model created, and the classification type. It is necessary to choose the optimizer that gives the fastest and most accurate results according to the processed data and the model used.

3. Experimental Study

In this study, the performance of the CNN model created was evaluated comparatively using four different optimizers, namely SGD, RMSprop, Adam, and AdaMax, on the dataset consisting of digits in the Turkish sign language. 2602 images used were divided into 80% training and 20% test set. There were 1649 images in the training set and 413 images in the test set.

In this study, all models were coded with Python and run with GPU acceleration. The study was carried out on hardware with 32 GB RAM, Intel Core i7-9750H processor, NVIDIA GeForce RTX 2070 graphics card.

3.1 The CNN Model

The CNN model created consists of 16 layers. The first layer is the input layer containing a total of 4096 pixels in 64x64 size for each image. The second layer is the convolution layer and contains 8 filters of 5x5 size. The third layer is the pooling layer with a pool size of 2x2. The fourth layer is the dropout layer that makes 25% of the neurons randomly passive. The fifth layer is the convolution layer and contains 16 filters of 3x3 size. The sixth layer is the pooling layer with a pool size of 2x2. The seventh layer is the dropout layer that makes 25% of the neurons randomly passive. The eighth layer is the convolution layer and contains 32 filters of 3x3 size. The ninth layer is the pooling layer with a pool size of 2x2. The tenth layer is the dropout layer that makes 25% of the neurons randomly passive. The eleventh layer is the convolution layer and contains 64 filters of 3x3 size. The twelfth layer is the pooling layer with a pool size of 2x2. The thirteenth layer is the dropout layer that makes 25% of the neurons randomly passive. The ReLU activation function was used in all mentioned layers. The maximum pooling method was used in all the pooling layers.

The fully connected layer provides the transition to a 3-layer artificial neural network. The first layer consists of 128, the second layer 64, and the last layer, the output layer, consists of 10 neurons. The ReLU activation function was used in the first two layers, the softmax activation function was used in the output layer.

The summary of the CNN model created is shown in Table 1.

The model includes a total of 164,618 trainable parameters.

3.2 Training and Results

In a neural network, the difference between predicted and actual values is called the error rate. Network parameters are updated depending on the error rate and in this way, the most accurate result is tried to be reached. The update parameters are multiplied by the learning rate to quickly reduce the error during the training process. The learning rate should be changed dynamically depending on the current error rate in the network. There are several optimization methods used for this job and it is important to choose the most ideal one.

The developed CNN model was trained four times, each using a different optimizer namely SGD, RMSprop, Adam, and AdaMax. Each training process took 100 epochs. The input batch size was 250 images. Accuracy and losses at the end of each training were reported.

SGD optimizer: As a result of 100 epochs, the training accuracy achieved with the SGD optimizer was 13% and the test accuracy was 7.7%. The confusion matrix obtained at the end of the training is shown in Figure 4.

Table 1. Summary of the CNN model created

Layer	Output Shape	Parameter
conv2d_1 (Conv2D)	(None, 64, 64, 8)	208
max_pooling2d_1	(None, 32, 32, 8)	0
dropbox_1 (Dropout)	(None, 32, 32, 8)	0
conv2d_2 (Conv2D)	(None, 32, 32, 16)	1168
max_pooling2d_2	(None, 16, 16, 16)	0
dropout_2 (Dropout)	(None, 16, 16, 16)	0
conv2d_3 (Conv2D)	(None, 16, 16, 32)	4640
max_pooling2d_3	(None, 8, 8, 32)	0
dropout_3 (Dropout)	(None, 8, 8, 32)	0
conv2d_4 (Conv2D)	(None, 8, 8, 64)	18496
max_pooling2d_4	(None, 4, 4, 64)	0
dropout_4 (Dropout)	(None, 4, 4, 64)	0
flatten_1 (Flatten)	(None, 1024)	0
dense_1 (Dense)	(None, 128)	131200
dense_2 (Dense)	(None, 64)	8256
dense_3 (Dense)	(None, 10)	650

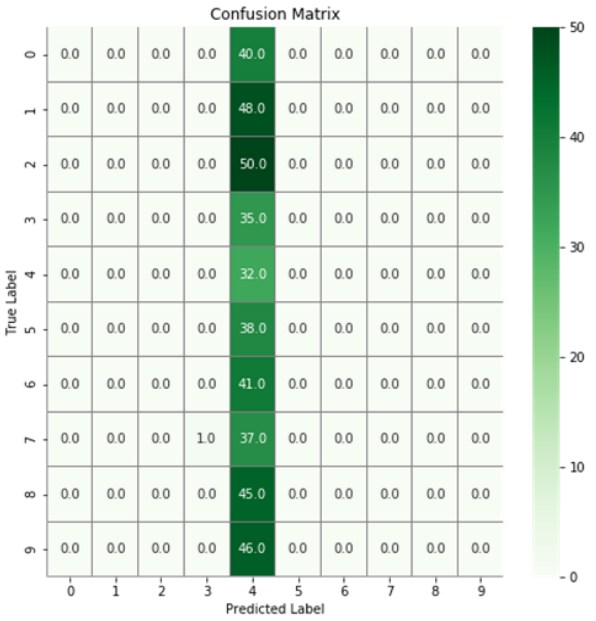

Figure 4. The confusion matrix of the SGD optimizer

As seen from the confusion matrix, the classification success of the model was extremely low. The accuracy and the loss graphics obtained with the SGD optimizer are shown in Figures 5 and 6.

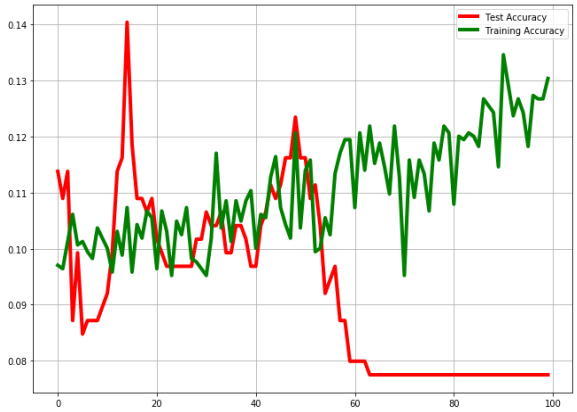

Figure 5. The accuracy graphics of the SGD optimizer

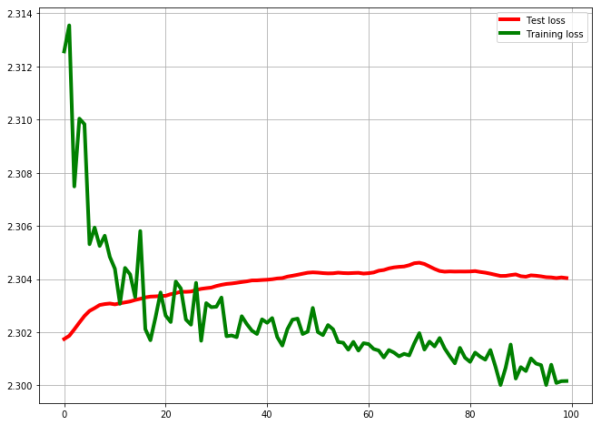

Figure 6. The loss graphics of the SGD optimizer

High variance arising from the frequent updating of the SGD optimizer is seen as an oscillation in the graphics. SGD often got stuck in local minimums. As a result of 100 epochs, it could not raise the accuracy and reduce the error quickly. Its performance was insufficient compared to the other optimizers used with an equal number of epochs.

RMSprop: As a result of 100 epochs, the training accuracy achieved with the RMSprop optimizer was 95.45% and the test accuracy was 96.85%. The confusion matrix obtained at the end of the training is shown in Figure 7.

As seen from the matrix, the model made quite accurate predictions. The accuracy and the loss graphics obtained with the RMSprop optimizer are shown in Figures 8 and 9.

RMSprop significantly increased success compared to SGD. There were still oscillations in the graphics, but they were greatly reduced. RMSprop quickly reduced the error while increasing accuracy.

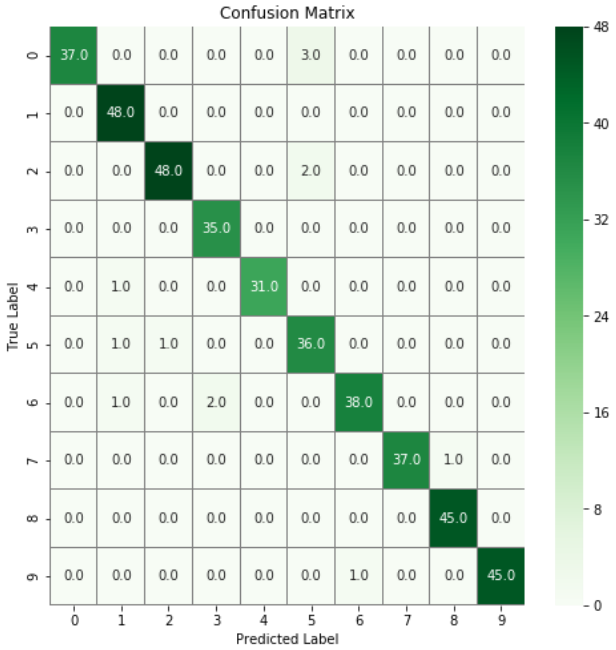

Figure 7. The confusion matrix of the RMSprop optimizer

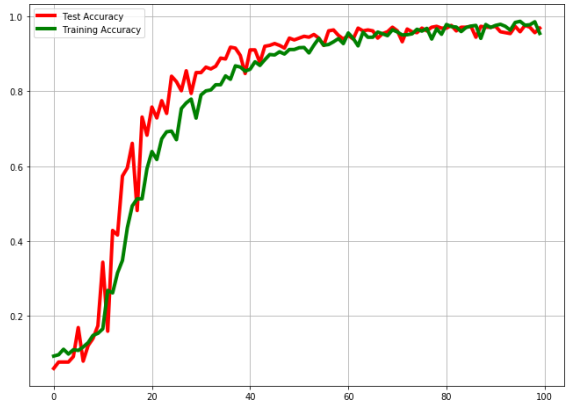

Figure 8. The accuracy graphics of the RMSprop optimizer

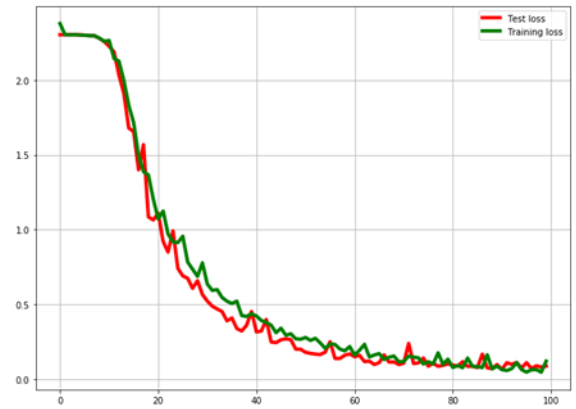

Figure 9. The loss graphics of the RMSprop optimizer

Adam: As a result of 100 epochs, the training accuracy achieved with the Adam optimizer was 98.42% and the test accuracy was 98.55%. The confusion matrix obtained at the end of the training is shown in Figure 10.

As can be seen from the matrix, the discrimination ability of the model with the Adam optimizer was quite high and the model made highly accurate predictions. The accuracy and the loss graphics obtained with the Adam optimizer are shown in Figures 11 and 12.

As seen from the graphics, the accuracy of the model with the Adam optimizer increased faster than the other optimizers. The error showed a faster descent with fewer oscillations.

Adamax: As a result of 100 epochs, the training accuracy achieved with the Adamax optimizer was 89.81% and the test accuracy was 91.53%. The confusion matrix obtained at the end of the training is shown in Figure 13.

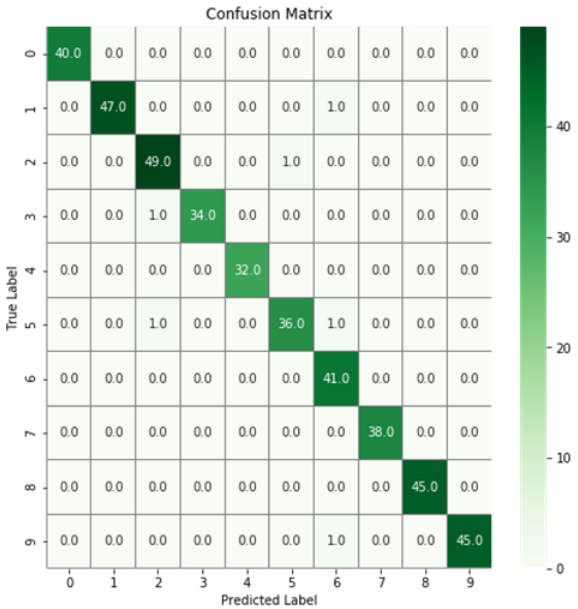

Figure 10. The confusion matrix of the Adam optimizer

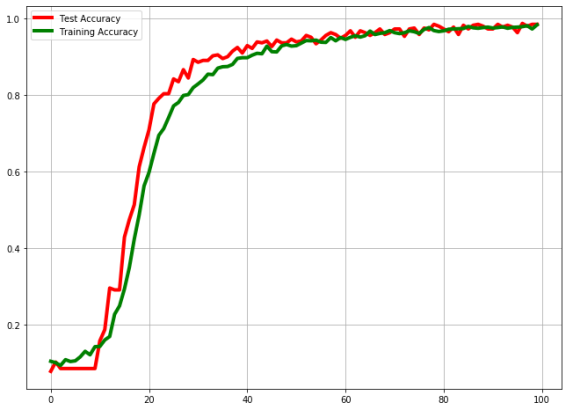

Figure 11. The accuracy graphics of the Adam optimizer

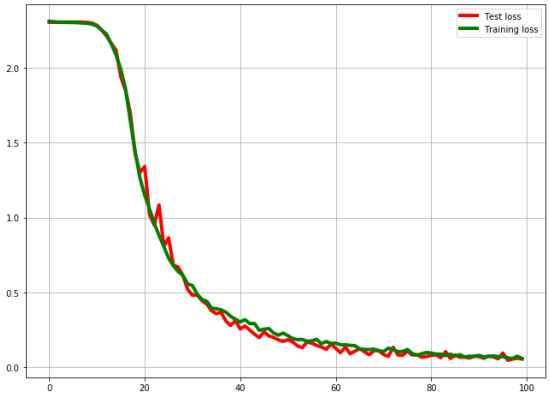

Figure 12. The loss graphics of the Adam optimizer

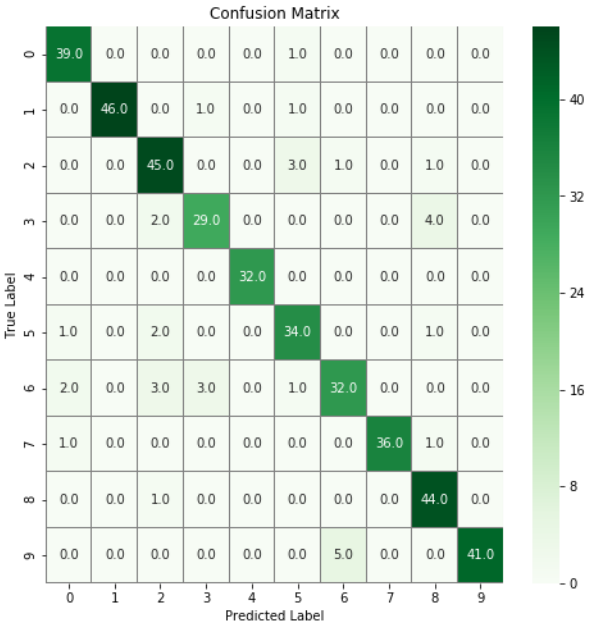

Figure 13. The confusion matrix for the Adamax optimizer

According to the matrix, Adamax had a successful discrimination and classification accuracy, although it was lower than Adam and RMSprop. The accuracy and the loss graphics obtained with the Adamax optimizer are shown in Figures 14 and 15.

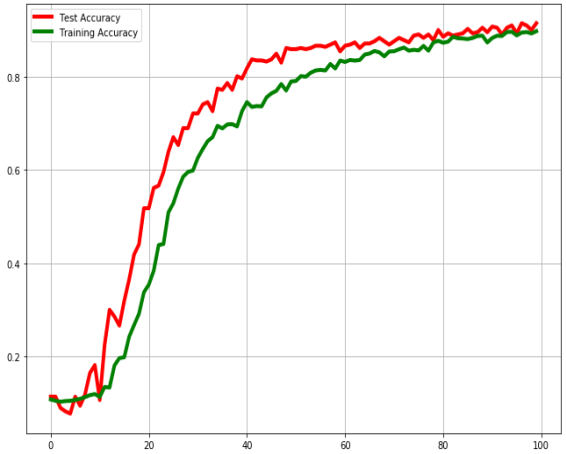

Figure 14. The accuracy graphics for the Adamax optimizer

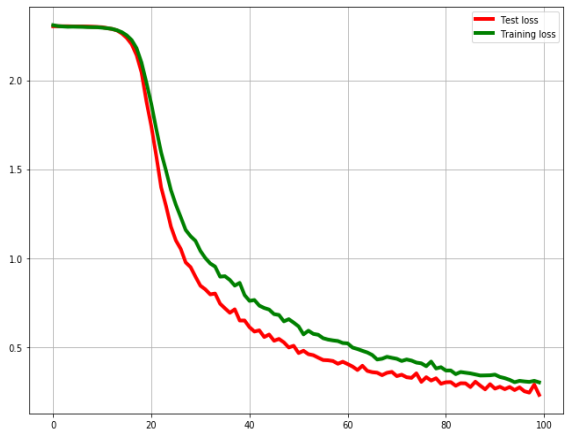

Figure 15. The loss graphics for the Adamax optimizer

Table 2. Accuracy values of the CNN model with different optimizers

Optimizer	Training Accuracy	Test Accuracy
SGD	%13	%7.7
RMSprop	%95.45	%96.85
Adam	%98.42	%98.55
Adamax	%89.81	%91.53

According to the graphs, although Adamax provides partially lower accuracy compared to Adam and RMSprop, the graphics oscillated less than RMSprop. Adamax has reached similar accuracy levels in a longer period than Adam.

As a result of 100 epochs, the training and test accuracies of the CNN model created with different optimizers are summarized in Table 2.

The Adam optimizer provided higher accuracy in both training and test stages than the other three optimizers used. The Adam optimizer, which is widely used in the field of deep learning due to its performance, also showed the highest performance in this study. There was also consistency between the training and test accuracies of Adam optimizer. The SGD optimizer got stuck in local minimums at the stage of minimizing the error with frequent updates and performed lower than other optimizers at equal epoch values. The RMSprop optimizer provided a close performance to Adam and less oscillation than SGD. The Adamax optimizer performed lower than RMSprop in terms of accuracy values but it was more consistent with fewer oscillations.

4. Conclusions

Sign language plays an important role in the communication of hearing-impaired people among themselves and with people without hearing impairments. However, there are problems in communication due to the differences between sign languages and the low number of people who know sign language. Sign language interpretation is done by converting visible body signs into speech or written language. Computer vision applications imitating the human eye offer a technological solution in this field.

CNN, which is a deep learning technology, has great success, especially in image classification. CNN models are trained with existing images, allowing them to predict different situations in the future. To reduce the estimation error, the parameters used in the CNN network need to be finely adjusted. One of these parameters is the learning rate. The learning rate is a coefficient used to update network parameters depending on the error that occurs in a neural network. Optimization is required so that the learning rate does not get stuck in the local minimum points while reducing the network error. There are various optimizers used for this purpose that perform different performances on different models and datasets.

With the CNN model developed in this study, 2062 images consisting of the digits in Turkish sign language were classified. The CNN model was trained four times using four different optimizers and the performance of each stage was evaluated. The optimizers used are SGD, RMSprop, Adam, and Adamax. In terms of training and test accuracy, Adam optimizer performed best. The CNN model we have established has achieved a remarkable success among the studies conducted on the same dataset with a 98.55% success rate. In the following studies, a larger database including gestures related to Turkish sign language will be created and model success will be tested with a similar CNN model and different optimizers.

Declaration

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article. The author(s) also declared that this article is original, was prepared in accordance with international publication and research ethics, and ethical committee permission or any special permission is not required.

References

1. Oral, A. Z., *Türk işaret dili çevirisi.* 2016, Ankara.
2. Van Herreweghe, M. *Prelinguaal dove jongeren en nederlands: een syntactisch onderzoek.* 1996. PhD Thesis. Ghent University.
3. Alkoffash, M. S., Bawaneh, M. J., Muaidi, H., Alqrainy, S., and Alzghool, M. A *survey of digital image processing techniques in character recognition*. International Journal of Computer Science and Network Security (IJCSNS), 2014. 14(3): p. 65.
4. Bheda, V., and Radpour, D., *Using deep convolutional networks for gesture recognition in American sign language.* arXiv preprint arXiv:1710.06836, 2017.
5. Koller, O., Ney, H., and Bowden, R., *Deep learning of mouth shapes for sign language.* In Proceedings of the IEEE International Conference on Computer Vision Workshops, 2015. p. 85-91.
6. Huang, J., Zhou, W., Li, H., and Li, W., *Sign language recognition using 3d convolutional neural networks.* In 2015 IEEE international conference on multimedia and expo (ICME), 2015. p. 1-6.
7. Pigou, L., Dieleman, S., Kindermans, P. J., and Schrauwen, B., *Sign language recognition using convolutional neural networks.* In European Conference on Computer Vision, 2014. p. 572-578.
8. Hasan, S. K., and Ahmad, M., *A new approach of sign language recognition system for bilingual users.* In 2015 International Conference on Electrical & Electronic Engineering (ICEEE), 2015. p. 33-36.
9. Agarwal, A., and Thakur, M. K., *Sign language recognition using Microsoft Kinect.* In 2013 Sixth International Conference on Contemporary Computing (IC3), 2013. p 181-185.
10. Oyewole, O. G., Nicholas, G., Oludele, A., and Samuel, O., *Bridging Communication Gap Among People with Hearing Impairment:, An Application of Image Processing and Artificial Neural Network*. International Journal of Information and Communication Sciences, 2018. 3(1): p. 11.
11. Wang, C., Gao, W., and Xuan, Z., *A real-time large vocabulary continuous recognition system for chinese sign language.* In Pacific-Rim Conference on Multimedia, 2001. p. 150-157.
12. Kim, J. S., Jang, W., and Bien, Z., *A dynamic gesture recognition system for the Korean sign language (KSL)*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 1996. 26(2): p. 354-359.
13. Gani, E., & Kika, A., *Albanian Sign Language (AlbSL) Number Recognition from Both Hand's Gestures Acquired by Kinect Sensors*. arXiv preprint arXiv:1608.02991, 2016.
14. Assaleh, K., and Al-Rousan, M., *Recognition of Arabic sign language alphabet using polynomial classifiers*. EURASIP Journal on Advances in Signal Processing, 2005. 13: p. 507614.
15. Assaleh, K., Shanableh, T., Fanaswala, M., Bajaj, H., and Amin, F., *Vision-based system for continuous Arabic Sign Language recognition in user dependent mode*. In 2008 5th International Symposium on Mechatronics and Its Applications, 2008. p. 1-5.
16. Solís, F., Martínez, D., and Espinoza, O., *Automatic mexican sign language recognition using normalized moments and artificial neural networks*. Engineering, 2016. 8(10): p. 733-740.
17. Rajam, P. S., and Balakrishnan, G., *Recognition of tamil sign language alphabet using image processing to aid deaf-dumb people.* Procedia Engineering, 2012. 30: p. 861-868.
18. Turkey Ankara Ayrancı Anadolu High School's Sign Language Digits Dataset, *https://www.kaggle.com /ardamavi/sign-language-digits-dataset*. Web. 10 Jan 2020.
19. Gazel, S. E. R., and batı, C. T., *Derin Sinir Ağları ile En İyi Modelin Belirlenmesi: Mantar Verileri Üzerine Keras Uygulaması.* Yüzüncü Yıl Üniversitesi Tarım Bilimleri Dergisi, 29(3): p. 406-417.
20. Kingma, D. P., and Ba, J., *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980, 2014.
21. Aran, O., Keskin, C., and Akarun, L., *Sign language tutoring tool*. In 2005 13th European Signal Processing Conference, 2005. pp. 1-4.
22. Beşer, F., Kizrak, M. A., Bolat, B., and Yildirim, T., *Recognition of sign language using capsule networks*. In 2018 26th Signal Processing and Communications Applications Conference (SIU), 2018. p. 1-4.
23. Ozcan, T., and Basturk, A., *Transfer learning-based convolutional neural networks with heuristic optimization for hand gesture recognition*. Neural Computing and Applications, 2019. 31(12): p. 8955-8970.