Bitlis Eren Üniversitesi Fen Bilimleri Dergisi
BİTLİS EREN UNIVERSITY JOURNAL OF SCIENCE
ISSN: 2147-3129/e-ISSN: 2147-3188
VOLUME: 12 NO: 3 PAGE: 746-763 YEAR: 2023
DOI:10.17798/bitlisfen.1294417

CNNTuner: Image Classification with A Novel CNN Model Optimized Hyperparameters

Halit ÇETİNER[1*], Sedat METLEK[2]

[1]*Isparta University of Applied Sciences, Vocational School of Technical Sciences, Isparta, Turkey*
[2]*Burdur Mehmet Akif Ersoy University, Vocational School of Technical Sciences, Burdur, Turkey*
(ORCID: 0000-0001-7794-2555) (ORCID: 0000-0002-0393-9908)

Keywords: CNN, Deep learning, F-MNIST, Keras Tuner, Parameter optimization.

Abstract
Today, the impact of deep learning in computer vision applications is growing every day. Deep learning techniques apply in many areas such as clothing search, automatic product recommendation. The main task in these applications is to perform the classification process automatically. But, high similarities between multiple apparel objects make classification difficult. In this paper, a new deep learning model based on convolutional neural networks (CNNs) is proposed to solve the classification problem. These networks can extract features from images using convolutional layers, unlike traditional machine learning algorithms. As the extracted features are highly discriminative, good results can be obtained in terms of classification performance. Performance results vary according to the number of filters and window sizes in the convolution layers that extract the features. Considering that there is more than one parameter that influences the performance result, the parameter that gives the best result can be determined after many experimental studies. The specified parameterization process is a difficult and laborious process. To address this issue, the parameters of a newly proposed CNN-based deep learning model were optimized using the Keras Tuner tool on the Fashion MNIST (F-MNIST) dataset containing multi-class fashion images. The performance results of the model were obtained using the data separated according to the cross-validation technique 5. At the same time, to measure the impact of the optimized parameters on classification, the performance results of the proposed model, called CNNTuner, are compared with state-of-the-art (SOTA) studies.

1. Introduction

The fashion industry is an industry that operates in many important areas, from unworn waste clothes to the creation of online store catalog images, especially in the field of sustainable fashion, which is the development of useful products. Deep learning approaches are used in different problems such as pose estimation, portrait graphic creation, garment segmentation, and garment recognition in the field of fashion [1]–[3]. Since these methods are more successful in automatic feature extraction and obtaining strong distinctive features, unlike classical machine learning methods, there is a tendency towards these methods. Security forces also have difficulties in recognizing and classifying clothes in suspicious situations where there is no clue [1]. In addition, trainable clothing search and classification systems can be created in accordance with the account profile in e-commerce-based systems, which are quite common today [3]. With the help of the specified features, it is aimed to create strong marketing

[*]Corresponding author: *halitcetiner@isparta.edu.tr*

Received: 09.05.2023, Accepted: 13.08.2023

strategies by automatically bringing the dresses suitable for the account profile. Along with these purposes and motivations, it becomes important to automatically classify clothes according to their types and shapes in line with very different purposes and targets. The problem of classifying fashion dresses is a complex task because of the multiplicity of labels characterizing the garment type, the richness of the garment features, and the similarity of the garments. Differences in camera shooting angle, lighting, and background differences are effective in deepening the classification problem [4]. In addition, high similarities in similar clothing classes such as trousers and tights make classification difficult. Although clothes recognition and classification, which is easy for humans, can be performed automatically by a computer with high performance, algorithms that work with high accuracy are needed.

Computer algorithms, on the other hand, are getting stronger day by day with powerful artificial intelligence libraries such as Tensorflow to extract meaningful information from the ever-increasing volume of images through search engines and social networks. In addition to supporting the Central Processing Unit (CPU), these libraries also support graphics processing units with high computing power, such as the Graphics Processing Unit (GPU). GPU-based codes enable rapid processing of thousands of images. Due to the limitations of classical machine learning algorithms in processing large amounts of image data, there has been a trend towards deep neural networks such as CNNs [1], [5]. There is also an increase in electronic transactions, where most transactions are carried out electronically and most of them are controlled in real time over the internet. Time series created in electronic transactions are analyzed through statistical and mathematical analysis. However, manual adjustment of the algorithm parameters used in the analysis is a difficult, tedious, laborious, and time-consuming problem. In this problem, the layer adjustments of the proposed deep learning architectures should be made automatically for the automatic classification of clothing images. In this article, the Keras Tuner tool has been used to automatically adjust the parameters of the model called CNNTuner.

Recognition and classification of images in the field of fashion using models based on deep learning architectures have many different advantages. Using these methods, users will be able to perform many different tasks in a short period of time, such as searching for related clothing and identifying the suspect type of clothing. Performing these tasks by hand is a tedious, time-consuming and exhausting process. New classification methods based on deep learning can be proposed to improve the performance of experts in searching for clothes and finding related clothes.

There are a number of issues that need to be considered when using deep learning methods for clothing classification. Firstly, due to differences in perspective, the same outfit can be described as different and different outfits as the same [1]. Secondly, clothes can be deformed as a result of washing, stretching or folding after washing [2], [3]. The third is the camera angle, lighting differences, diffuse background and shadows that are generally encountered in computer vision methods [12]. In this study, a dataset consisting of images with regular backgrounds, taken from the same angles and undeformed was determined by investigating the three main problems mentioned. In multiclass classification, which is used instead of binary classification, it is a difficult task to reduce the classification error [1]. In this study, parameter optimization of the proposed CNN model was carried out in order to reduce the multiclassification error. In order to determine the effect of parameter optimization on classification, it was compared with non-optimized parameters. In multiclass classification problems, it can be difficult to distinguish similar classes. In multiclass classification problems, the softmax activation function used in the last layer of CNN models shows the probability value of each category. In multiclass classification, the difference between these probability values may be small.

The main contributions of this article to the literature are given below.

- A new CNN model named CNNTuner has been proposed for garment classification with multiple classification problems.

- Determining the best parameters with experimental studies is a long and tiring process. This process has been accelerated by CNNTuner.

- By providing parameter optimization of the CNNTuner model, the multiple classification error has been reduced.

The remainder of the article is organized as follows. In the second section, detailed information is given about the selected dataset and the proposed CNN model. In the third section, the effect of parameter optimization on the proposed CNN model is introduced together with the performance criteria. In the last section, the article concluded by giving information about future studies.

2. Literature Review

In the literature, classification tasks have been performed with optimized parameters using different convolution-based architectures on the F-MNIST dataset [13]. Two different CNN models have been developed by Greeshma and Sreekumar to classify fashion garments [13]. The first is a model with two convolutional layers, while the second is a model with four convolutional layers. The studies show that the training process is performed with 40 and 60 epoch steps. The Adam and Adadelta optimisation algorithms used 128 and 64 values as the batch size. It was found that the best parameter optimisation result was obtained from the model with the four-convolution layer Adam optimisation algorithm and a batch size value of 64. In terms of epochs, it achieved the best test accuracy value of 60 epochs. The specified epoch value is a very high value. In this article, an attempt is made to obtain better results using the model with the best optimisation parameter with a lower epoch value.

Bhatnagar et al. developed three different CNN models to defend the F-MNIST dataset [14]. The most successful models they have developed have used residual skip connections. It is stated that the learning process is accelerated by combining these connections with the batch normalization connections.

3. Material and Methods

3.1. Material

The accuracy metric shown in Equation 1 is the ratio of the predicted accuracy of the proposed model divided by the total predictions. Equation 2 shows the harmonic mean of the precision and recall metrics expressed in Equation 3 and Equation 4. The precision metric

The F-MNIST database contains 70,000 fashion images with a width and height of 28 pixels [15]. All images are in grayscale image and there are 10 categories. Each category contains 7,000 images. The training set consists of 60,000 images, while the test set consists of 10,000 images. The 10 categories are "T-shirt/top", "Trouser", "Pullover", "Dress", "Coat", "Sandal", "Shirt", "Sneaker", "Bag", "Ankle boot". Examples of images for each of these categories are shown in Figure 1. Since the separated training data were balanced, no data augmentation or balancing method was applied.

3.2. Evaluation Metrics

This section presents the model evaluation metrics constructed by performing parameter optimization. The metrics of precision, recall, F1 score, F2 score, specificity and accuracy were used to evaluate the model. False Negative (FN), True Negative (TN), False Positive (FP), True Positive (TP) markers were used in the formulation of these metrics. TP represents the accuracy of the proposed model's predicted class output and actual class output. FP means that the predicted value of the proposed model is correct while in reality it is incorrect. The FN marker represents that the proposed model predicts an incorrect output while the actual output is correct. Finally, TN means that the proposed model predicts the wrong output, while in the actual situation it is wrong [16].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$F1 = 2x\frac{PrecisionxRecall}{Precision+Recall} \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$Recall\ (Sensitivity) = \frac{TP}{TP+FN} \tag{4}$$

$$F2 = \frac{5xPrecisionxRecall}{4x(Precision+Recall)} \tag{5}$$

$$Specificity = \frac{TN}{TN+FP} \tag{6}$$

in Equation 3 is the number of correct predictions of the proposed model divided by the total number of predictions. In Equation 4, the recall formula shows the successful prediction status. Equation 5 increases the importance of recall performance while decreasing precision.

Equation 6 shows how effective the proposed model is in identifying negative tags.

3.3. Keras Tuner

Keras is a deep learning library that runs on top of artificial intelligence libraries such as Tensorflow. Optimization algorithms help to maximize performance by minimizing the objective function [13]. There are two separate processes in deep learning models: training and testing. In the training process, a low number of losses is a measure of whether performance is good or not [17]. The lower the losses, the more successful the proposal. The Adam optimization method is used to optimize the training of the proposed model [18].

3.4. CNN

CNN algorithms are widely used for classification of image data [19]–[21]. CNN consists of interconnected layers with different characteristics. These layers go from the input layer to the classification layer. The layers contain neurons that learn different parameters such as weights. The architecture of the CNN is not fully interconnected as in classical neural networks. Only the last layer is fully connected, partially avoiding the problem of over-fitting and the waste of full connections. Simultaneously, CNN has specialized layers for classical machine learning data reduction and feature extraction steps. While the feature mapping is performed by convolution layers, the dimensionality reduction of the feature maps is performed by the pooling layer. The task of understanding and classifying the values in the feature map is performed by the fully connected layer and the classification layer with activation function [22]. In general, binary classification tasks are performed with the sigmoid activation function, while the softmax activation function is used for multiple classification problems.

Popular CNN algorithms are known as AlexNet, ZFNet, VGGNet, GoogleNet and ResNet. AlexNet consists of 5 convolutional layers, pooling, dropout and 3 fully connected layers [23]. In addition to the classification layer, it uses the ReLU function for non-linear functions. ZFNet is designed to improve the performance of the AlexNet architecture [24].

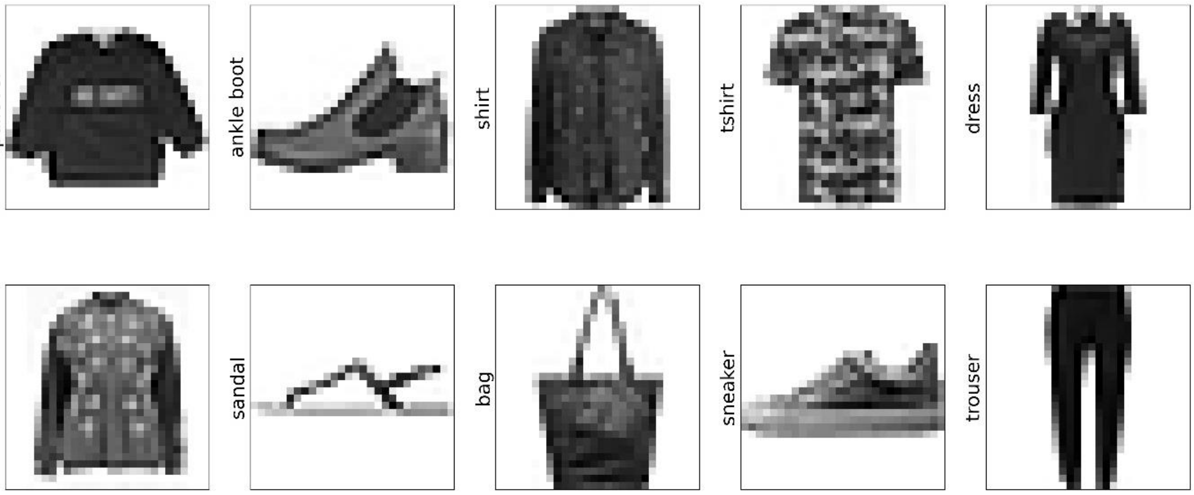

Figure 1. Sample images from classes in the dataset

The main reason why it is better is seen in the size of its parameters and filters [24]. Parameter optimization is also important in the development of popular CNN architectures, and was one of the main motivations for writing this paper. Instead of using windows with a width and height of 11 pixels, ZFNet uses windows with a width and height of 7 pixels, allowing more information to be retained. VGGNet, uses filters with a width and height of 3 pixels. In the pooling layer, it uses windows with a width and height of 2 pixels. To increase the volume depth, VGGNet has doubled the number of filters after each pooling layer [25]. GoogleNet is capable of processing at a number of different scales in parallel, with filters of different sizes [26]. It implements maximum pooling in window sizes with the same width and height for each of its parallel connections. The network consists of three convolutional layers, followed by 9 inception layers, including two convolutional layers and one fully connected layer. It has a total of 22 layers. ResNet includes shortcut module connections with an identity connection that can bypass the weight layers designed to solve the gradient problem [26], [27]. Popular CNN architectures can be used as a basis to provide suitable solutions for different problems, or, as in this study, models can be constructed by combining basic CNN layers with optimized parameters.

3.5. Proposed Model

Deep learning methods are widely used in the literature to solve a wide variety of problems [6]–[8]. Pre-trained weighted datasets such as ImageNet and COCO have helped explore the power of deep learning. The COCO dataset is a large dataset consisting of 381 thousand images that can be used in 91 different object detection and segmentation tasks [9]. ImageNet, on the other hand, is a dataset with 1.2 million training sets with 1000 object categories [10]. CNN algorithms, which have been used since 1980, have increased their popularity in ImageNet competitions [11]. Because CNN architectures can provide solutions for different problems and features with high distinctiveness, it formed the basis of the model proposed in this study. For these reasons, the automatic classification of fashion clothing images is carried out with a model called CNNTuner. At the same time, the parameters of this model are determined automatically, preventing the time spent on parameter determination.

The proposed model is a new 15-layer architecture developed on the basis of convolutional layers. Keras Tuner tool, one of the popular parameter optimization algorithms, is integrated into this architecture. In order for this tool to work with different filtering and activation options, each step has been controlled and run. The parameters of each convolution, maximum pooling and activation functions used in the model were determined by the Keras Tuner tool instead of being randomly selected. In the CNNTuner model, the details of which are given below, in order for the layers to work harmoniously with each other, any parameter selected at each step should not prevent the model from being compiled. For this purpose, many different experimental studies have been carried out and the results are presented in the study. The proposed model is detailed in the next step of the article.

A new CNN model called CNNTuner is proposed to measure the effect of optimized CNN parameters on classification. Fashion images with dimensions of 28x28x1 pixels were used as input for the classification of the F-MNIST dataset. Grayscale image was used as the input color format. Some of the images in the format used are shown in Figure 1. Figure 2 shows the tested hyperparameters and layers of the CNNTuner model. The aim of this article is to optimize the parameters in the CNNTuner model structure in order to obtain the best performance results, without having to be tested by constantly changing them. In the model realized for this purpose, 28x28x1 pixels dimensions images are taken as input in the first layer. In the 2nd layer, the parameter of the convolution layer is set as one of 8 filters from 32 filters to 256 filters. At the same time, instead of using a fixed window size, the optimizing algorithm will prefer to use one of the 3 or 5 pixel window sizes. In the 3rd layer, one of the frequently used ReLU, eLU, seLU, tanh activation functions are defined to be selected.

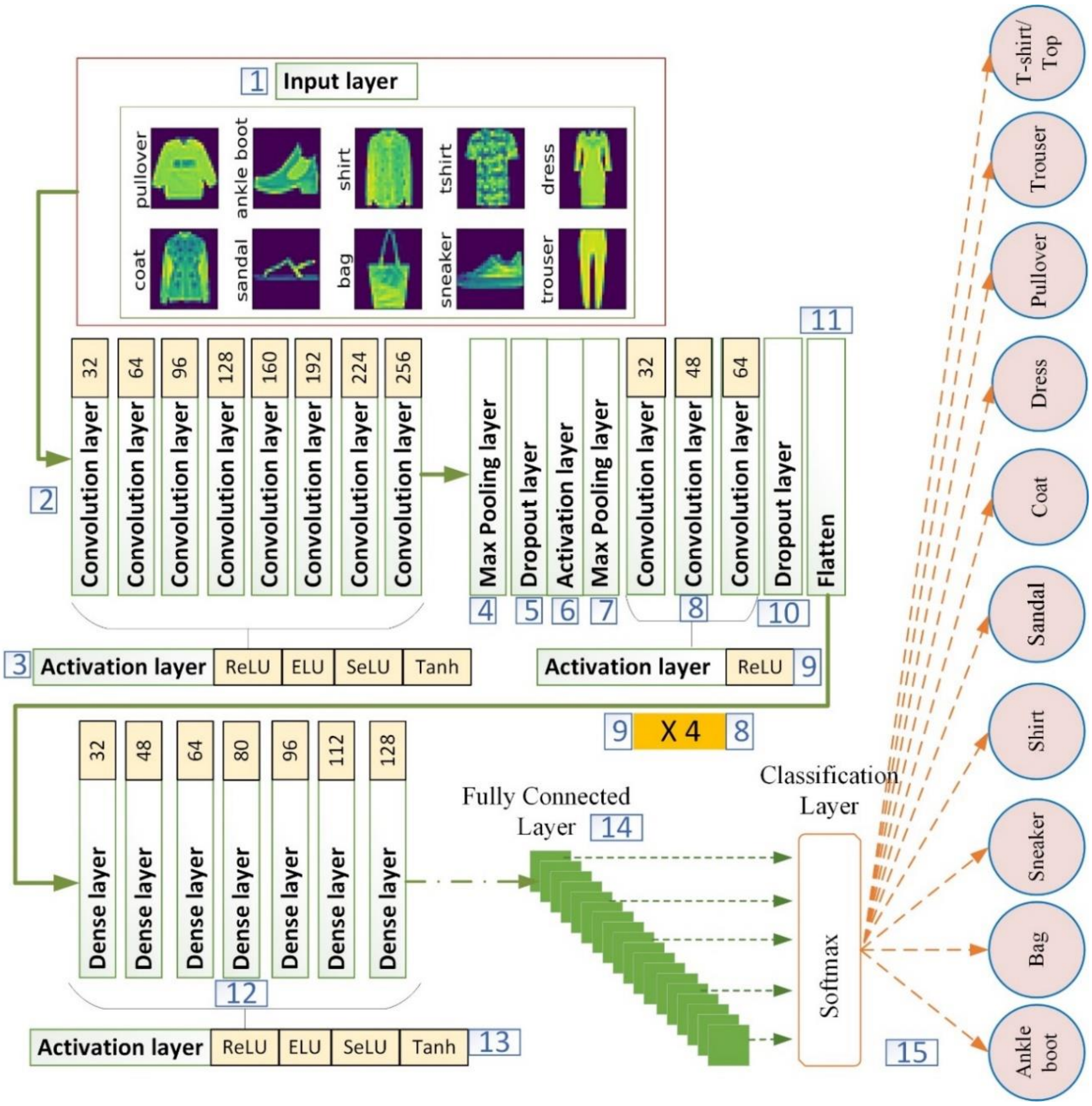

Figure 2. Proposed model with hyperparameter tested

In the 4th layer of the proposed model, a maximum pooling layer with a width and height value of 1 pixel is defined. At this point, since the number of filters coming from the previous layers is not clear, a layered structure has been created that differs from the usual fixed-definition CNN architecture. No matter what convolution filter comes, a model has been created in which the deep learning model will be built without errors. In the 5th layer, the dropout layer, which performs the neuron dropout process that prevents overlearning at a rate of 0.9, is applied.

A constant ReLU activation function is defined in the 6th layer. A maximum pooling layer with a width and height of 2 pixels is defined in the 7th layer of the proposed model. In layers 8 and 9, the convolution layer and the ReLU activation function layers are repeated together 4 times. In layer 8, one of the 8 filters from 32 filters to 256 filters is determined as the parameter of the convolution layer. A window size of 3 pixels in width and height has been applied to the selected convolution layer. The ReLU activation function was used in the 9th layer. In the 10th layer, a dropping layer was added, which causes neurons to drop out at a rate of 0.9. In the 11th layer, the Flatten layer has been applied, which reduces the data from the layers to a single dimension. In the 12th layer, a dense layer is defined, which increases by 16 neurons from 32 neurons to 128 neurons. The optimization algorithm has selected one of the specified parameters. In the 13th layer, one of the activation functions ReLU, eLU, seLU and tanh is defined to be selected. The 14th layer is a fully connected layer, which provides a full connection to the features from the previous layers. In the 15th layer, the model is completed with a classification layer with a softmax activation function with a total of 10 classes from the F-MNIST dataset.

In this paper, one of the most important points to emphasize is the selection of one of the many different layer parameter options. The Keras Tuner optimization tool was used instead of manual selection. The best parameters were determined through 100 different iterations. Thanks to the optimization method, which increases the efficiency of the proposed model, the best result is obtained both in terms of time and performance. Both accuracy and confusion matrix information are presented to measure the performance of the results. A total of 8 different filters were applied in the 2nd layer of the proposed model. Another point to emphasize is that the resulting layered structure should be adjusted so that there is no size mismatch. It is very important to set the application to work regardless of the number of filters selected. A detailed representation of the above parameters is shown in Figure 2. If desired, the model applied to the F-MNIST dataset can be tested on a variety of datasets with a low success rate in the literature. In this case, however, it may be necessary to modify the proposed model. Not all problems can be solved with a single deep learning model.

4. Results

Parameter optimization saves the researcher the lengthy process of determining appropriate parameters. In this context, the present study makes a unique contribution to the literature on the subject. In CNN models, there are pre-trained architectures as well as structures formed by convolution, pooling, batch normalization, dropout, fully connected, classification layers coming one after another in a certain shape and structure. There are two different ways to build CNN-based deep learning models. In this study, the second type of CNN structure is targeted by combining the basic layers of the CNN structure. In such models, researchers use experimental studies to determine parameters such as the number of filters, kernel size, and number of layer repetitions to achieve optimal training and testing performance. These procedures will speed up the parameter determination processes of the researchers due to the convenience they provide in determining the parameters. According to the cross validation method, recall, F1 score, F2 score, specificity, accuracy performance values were also examined in detail. At this stage, the training data of each Fold number of cross validation were tested separately. The Fold values of cross validation are Fold1, Fold2, Fold3, Fold4, and Fold5, respectively.

Table 1 shows the performance results that were obtained according to the Fold1 option on a class basis in the F-MNIST database. The results of each class with precision, recall, F1 score, F2 score and specificity metrics are shown. While the Trouser and Bag classes give the highest results, the results obtained from the Shirt and Coat classes are the lowest performance results. In addition to the performance results given in Table 1, the accuracy results obtained using the accuracy formula are shown in Figure 3. Class-wise accuracy values are given as well as the classification accuracy obtained collectively. According to the Fold1 option, the proposed model has an average accuracy of 95.84%.

Table 1. Fold1 performance result of the proposed model

Class name	Precision	Recall	F1 Score	Specificity	F2 score
T-shirt/top	0.94	0.96	0.95	0.95	0.96
Trouser	1.00	1.00	1.00	1.00	1.00
Pullover	0.93	0.93	0.93	0.94	0.94
Dress	0.94	0.97	0.95	0.98	0.96
Coat	0.90	0.93	0.91	0.97	0.93
Sandal	1.00	0.99	0.99	0.99	1.00
Shirt	0.92	0.84	0.88	0.85	0.91
Sneaker	0.97	0.99	0.98	0.99	0.99
Bag	1.00	1.00	1.00	1.00	1.00
Ankle boot	0.99	0.98	0.98	0.99	0.99

Table 2 presents the performance results obtained according to the Fold2 option on a class basis in the F-MNIST database. While the class of Trousers and the class of Sandals give the highest results, the results obtained from the class of pullover and the class of shirt are the lowest performance results. Figure 4, which is shared in integration with the performance metrics given in Table 2, shows the confusion matrix values obtained according to the same Fold value. According to the class-based accuracy values shown in Figure 4, accuracy values of 98.26%, 99.79%, 93.99%, 98.57%, 95.99%, 100%, 94.29%, 99.58%, 99.78%, 98.83% were achieved in T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot classes.

Table 3 presents the performance results obtained according to the Fold3 option on a class basis. Trouser, Sandal, Sneaker, Bag and Ankle boot classes gave higher results than the other classes. In this Fold3 option, there was no class with low results in general. In addition to the performance results given in Table 3, the accuracy results obtained using the accuracy formula are shown in Figure 5. Class-wise accuracy values are given as well as the classification accuracy obtained collectively. According to the Fold3 option, the proposed model has an average accuracy of 99.13%.

According to the Fold3 results given in Figure 5, T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle Boot classes reached 98.62%, 100%, 98.29%, 98.60%, 97.38%, 100%, 98.47%, 99.79%, 99.78%, 100% accuracy values respectively. It is seen that the highest error is obtained from the Coat class.

Figure 6 shows the confusion matrix results obtained according to the Fold4 option. T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot classes reached 98.05%, 99.79%, 95.24%, 98.83%, 98.62%, 99.78%, 96.16%, 100%, 99.78%, 99.79% accuracy values respectively. It is seen that the highest error is obtained from the Pullover class.

Table 2. Fold2 performance result of the proposed model

Class name	Precision	Recall	F1 Score	Specificity	F2 score
T-shirt/top	0.98	0.98	0.98	0.99	0.99
Trouser	1.00	1.00	1.00	1.00	1.00
Pullover	0.94	0.96	0.95	0.98	0.97
Dress	0.99	0.97	0.98	0.98	0.99
Coat	0.96	0.96	0.96	0.97	0.98
Sandal	1.00	1.00	1.00	1.00	1.00
Shirt	0.94	0.96	0.95	0.97	0.97
Sneaker	1.00	0.99	0.99	1.00	1.00
Bag	1.00	0.99	0.99	1.00	1.00
Ankle boot	0.99	1.00	0.99	1.00	1.00

Table 3. Fold3 performance result of the proposed model

Class name	Precision	Recall	F1 Score	Specificity	F2 score
T-shirt/top	0.99	0.99	0.99	0.99	1.00
Trouser	1.00	1.00	1.00	1.00	1.00
Pullover	0.98	0.98	0.98	0.98	0.99
Dress	0.99	0.99	0.99	0.99	1.00
Coat	0.97	0.99	0.98	1.00	0.99
Sandal	1.00	1.00	1.00	1.00	1.00
Shirt	0.98	0.96	0.97	0.96	0.98
Sneaker	1.00	1.00	1.00	1.00	1.00
Bag	1.00	1.00	1.00	1.00	1.00
Ankle boot	1.00	1.00	1.00	1.00	1.00

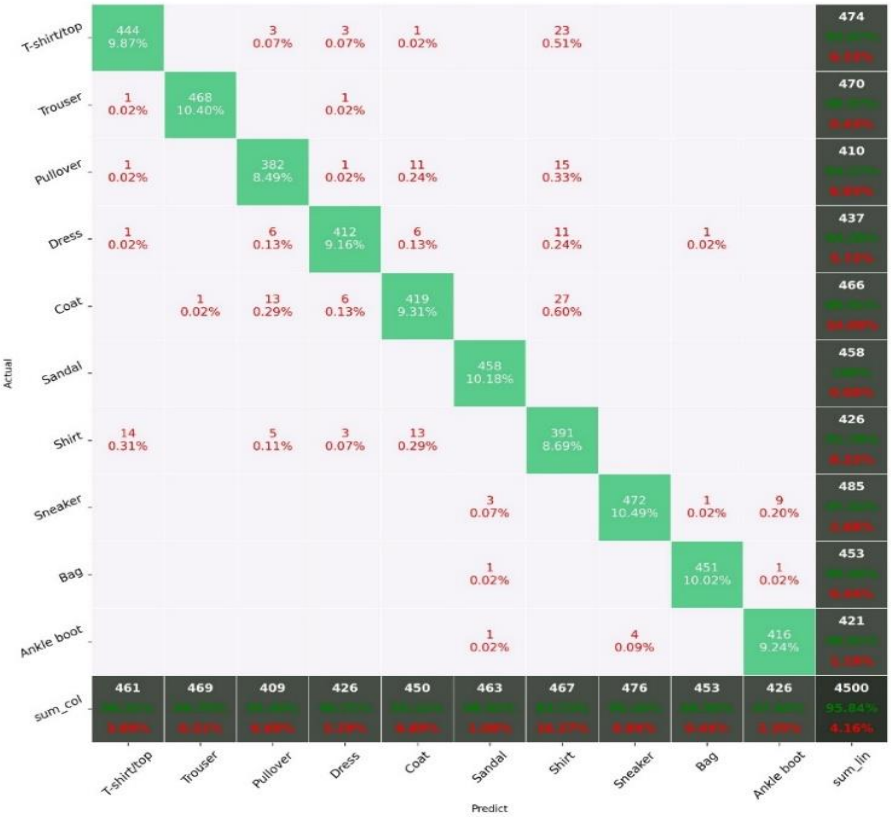

Figure 3. Fold1 confusion matrix results

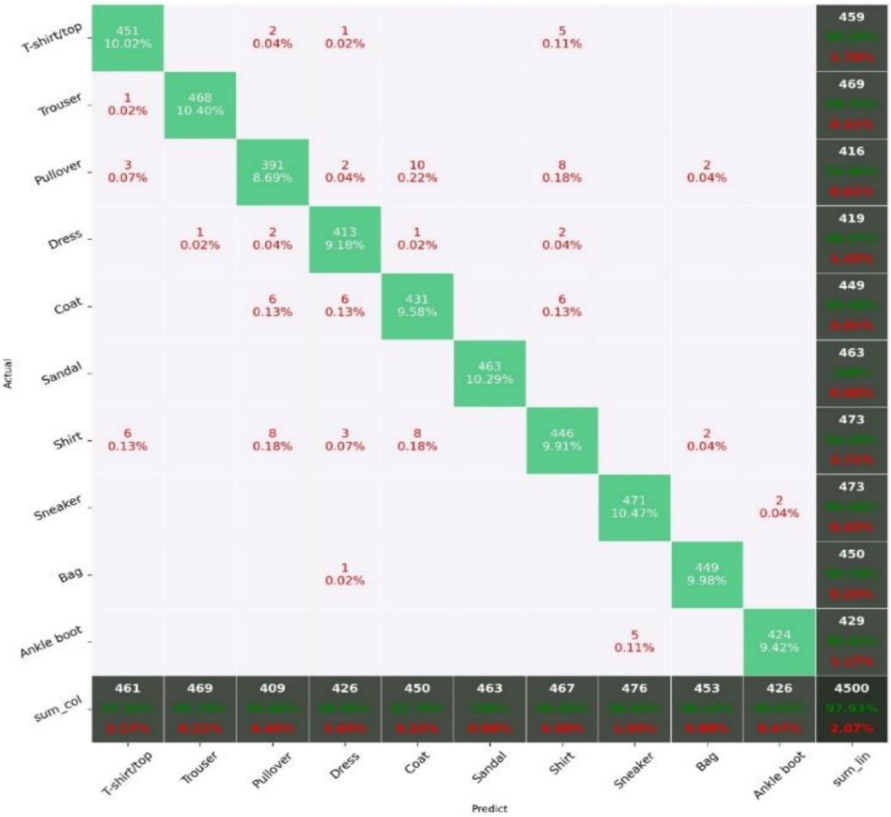

Figure 4. Fold2 confusion matrix results

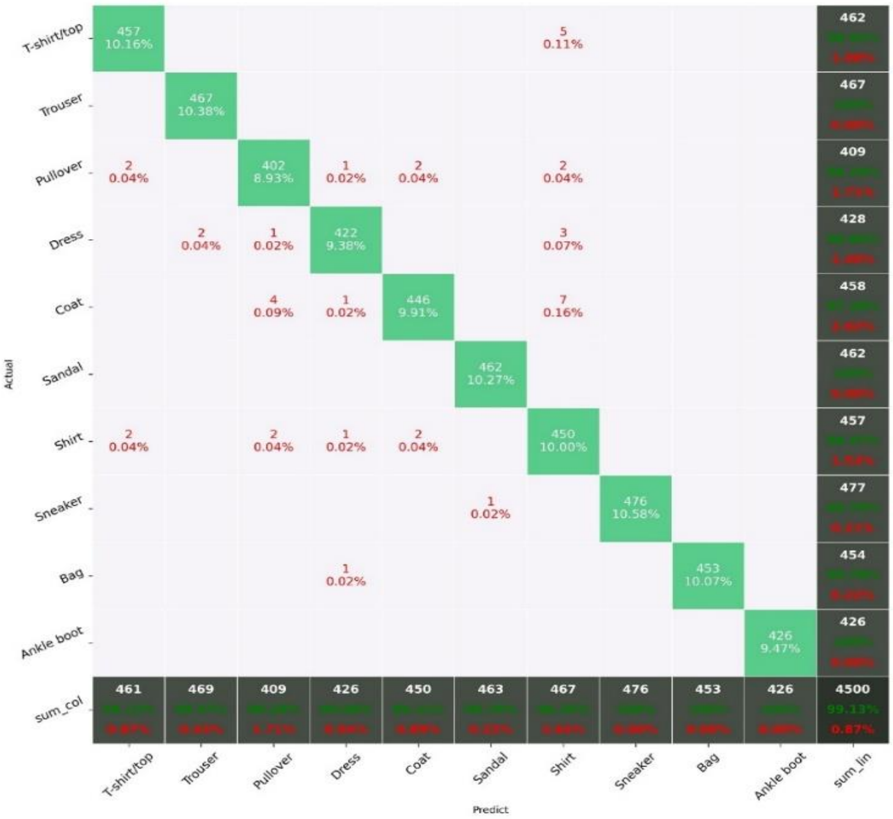

Figure 5. Fold3 confusion matrix results

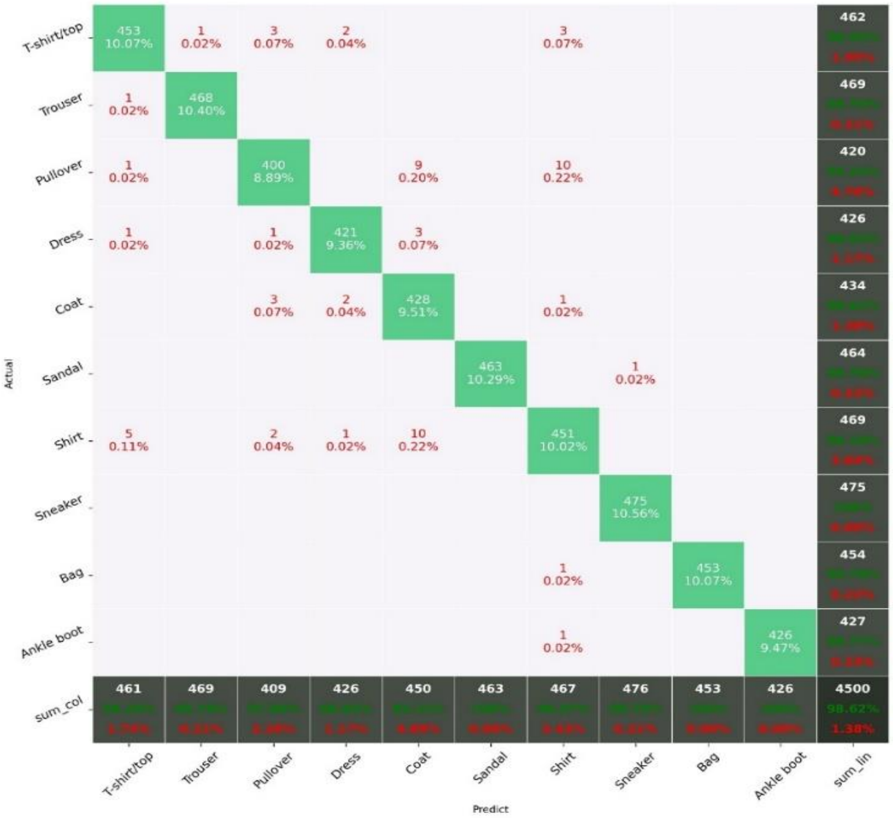

Figure 6. Confusion matrix results obtained according to the Fold4 option

Table 4 shows the performance results obtained according to the Fold4 option. Although it is similar to Fold3 results, there was no class with low results. The classes with very good results are Trouser, Sandal, Sneaker, Bag, Ankle boot. The confusion matrix results obtained according to the Fold5 option are given in Figure 7. According to these results, T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot classes gave 97.85%, 100%, 98.05%, 99.30%, 98.68%, 100%, 98.69%, 99.79%, 99.34%, 100% accuracy performance results. Among the classes, the T-shirt/top class gave the highest error. When the results are analyzed, it is determined that Fold5 provides superior performance than other Fold values. This is shown in Table 6.

Table 4. Fold4 performance result of the proposed model

Class name	Precision	Recall	F1 Score	Specificity	F2 score
T-shirt/top	0.98	0.98	0.98	0.99	0.99
Trouser	1.00	1.00	1.00	1.00	1.00
Pullover	0.95	0.98	0.97	0.99	0.98
Dress	0.99	0.99	0.99	0.99	1.00
Coat	0.99	0.95	0.97	0.95	0.99
Sandal	1.00	1.00	1.00	1.00	1.00
Shirt	0.96	0.97	0.96	0.98	0.98
Sneaker	1.00	1.00	1.00	1.00	1.00
Bag	1.00	1.00	1.00	1.00	1.00
Ankle boot	1.00	1.00	1.00	1.00	1.00

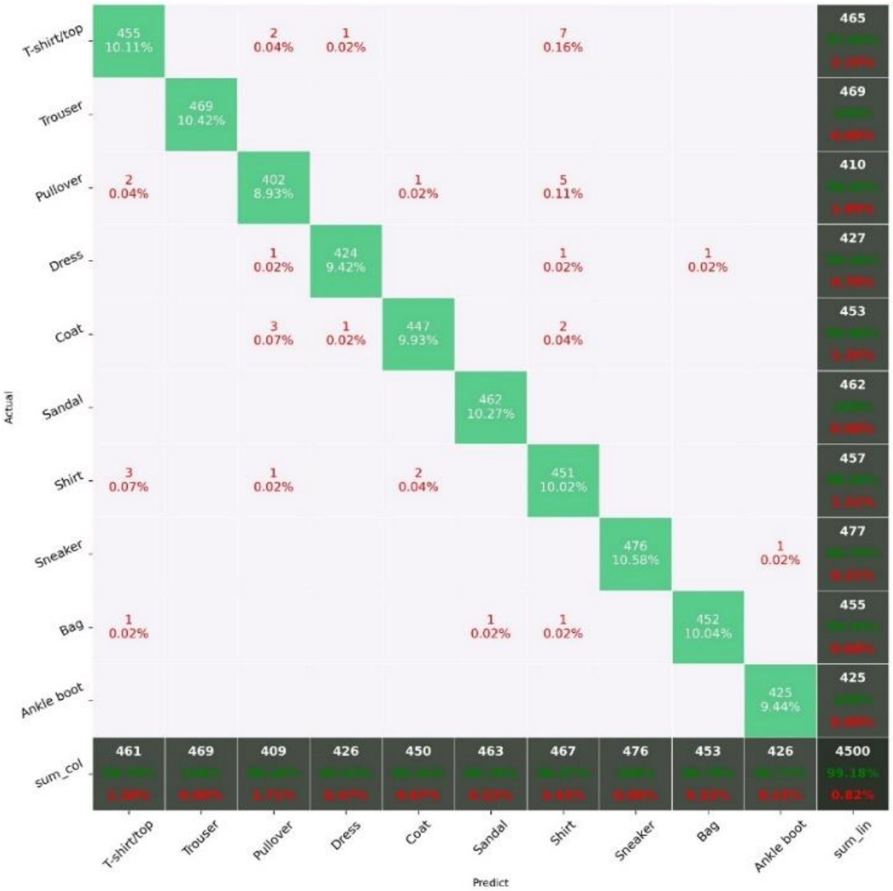

Figure 7. Confusion matrix results according to Fold5 option

Table 5. Fold5 performance result of the proposed model

Class name	Precision	Recall	F1 Score	Specificity	F2 score
T-shirt/top	0.98	0.99	0.98	0.99	0.99
Trouser	1.00	1.00	1.00	1.00	1.00
Pullover	0.98	0.98	0.98	0.98	0.99
Dress	0.99	1.00	0.99	0.99	1.00
Coat	0.99	0.99	0.99	0.99	0.99
Sandal	1.00	1.00	1.00	1.00	1.00
Shirt	0.99	0.97	0.98	0.98	0.99
Sneaker	1.00	1.00	1.00	1.00	1.00
Bag	0.99	1.00	1.00	1.00	1.00
Ankle boot	1.00	1.00	1.00	1.00	1.00

The results shown in Table 6 are different representations of the confusion matrix results given between Figure 3 and Figure 7. In addition to these, the performance results obtained are reinforced with graphical outputs.

Table 6 presents the training and validation results for each Fold value. Fold5 is preferred because the difference in success rate between the training and validation results is quite small. The model performance graphs plotted over Fold5 are presented below.

Figure 8-11 show the train accuracy, train loss, validation accuracy, validation loss graphs of the proposed model respectively. They have been zoomed in to make the graphs between the fold values clearer and closer together. In each model performance graph, the graph outputs between the last two epochs are zoomed and shown in a separate box. Train accuracy performance graphs of all Fold values are shown in Figure 8. According to the graph features shown in Figure 8, the Fold5 accuracy value is superior to the other Fold values. The difference between Fold1 train accuracy performance graph and Fold5 is higher than other Fold values. In Figure 9, the loss graphs of the Fold values whose train accuracy values are given in Figure 8 are plotted. At 20 epochs, the smallest loss value obtained was 0.0080.

Figure 10 shows the validation accuracy performance graphs. There is a significant difference between the Fold values. The most successful result is obtained from Fold5, while the lowest result is obtained from Fold1. In the same parallel, the highest loss is obtained from Fold1, while the lowest loss is obtained from Fold5.

Table 6. Performance results of train and validation of the proposed model

	Train		Validation	
	Accuracy	Loss	Accuracy	Loss
Fold1	0.9590	0.1111	0.9584	0.1241
Fold2	0.9942	0.0186	0.9793	0.0856
Fold3	0.9995	0.0023	0.9913	0.0213
Fold4	0.9949	0.0149	0.9862	0.0246
Fold5	0.9999	0.0080	0.9918	0.0189

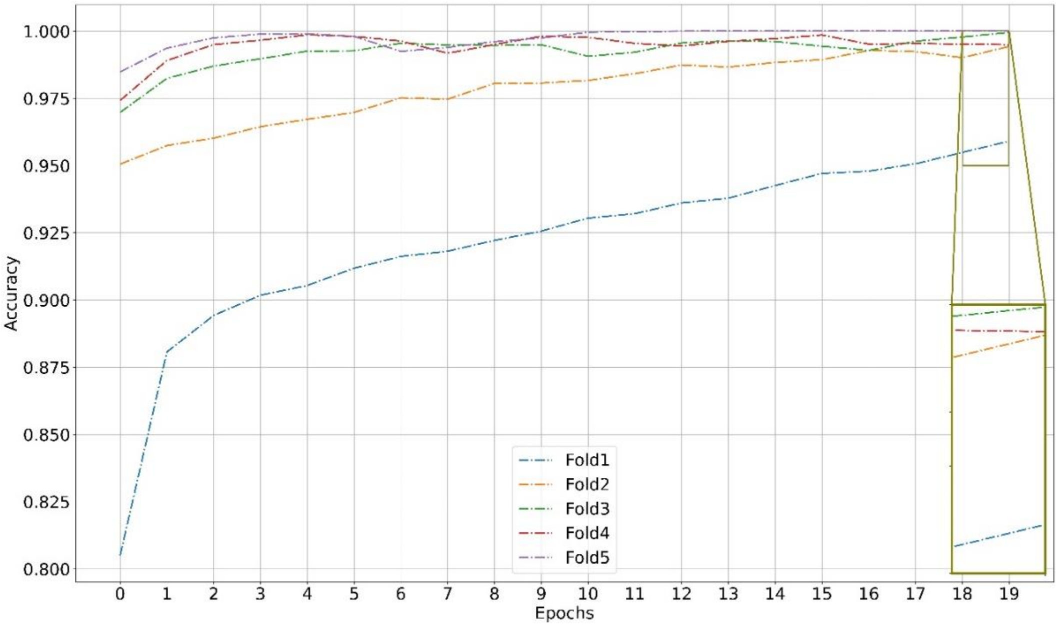

Figure 8. Train accuracy performance graph of the proposed model

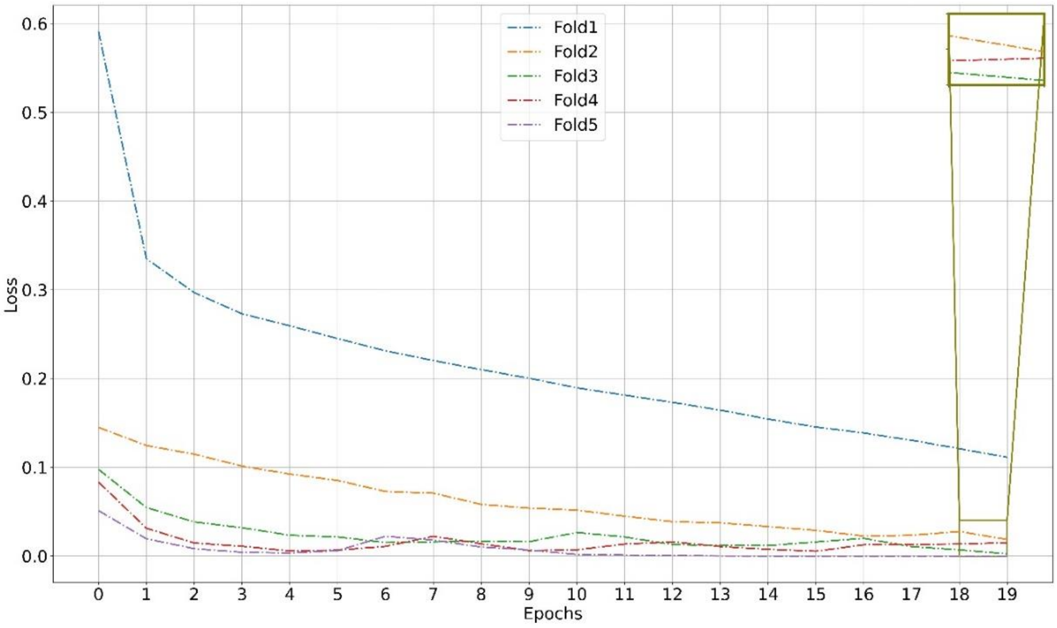

Figure 9. Graph of the performance of the proposed model for train losses

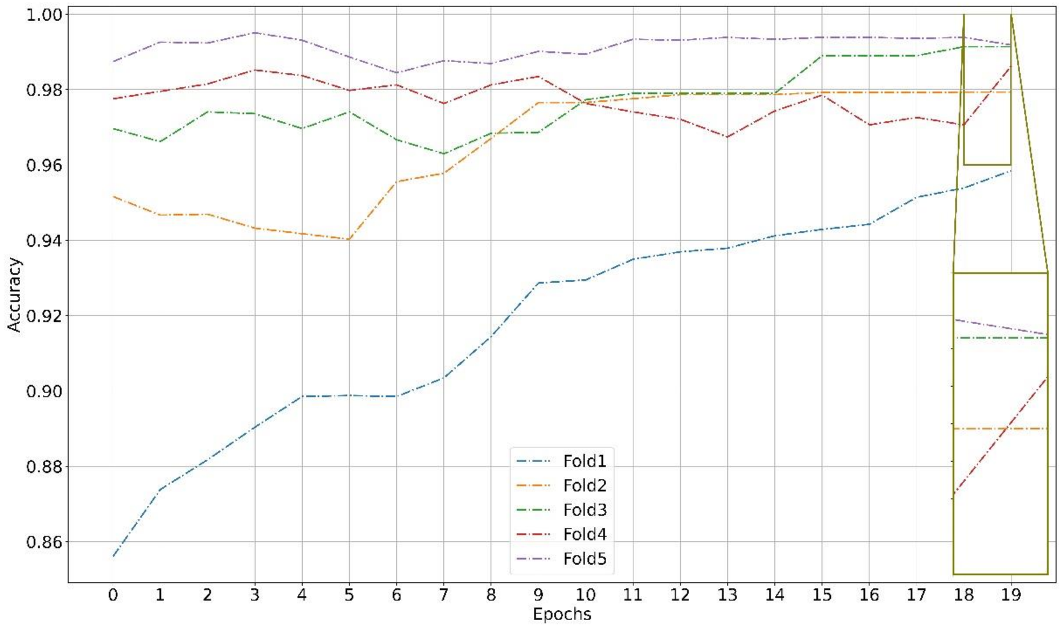

Figure 10. Graph showing the proposed model's performance in validating accuracy

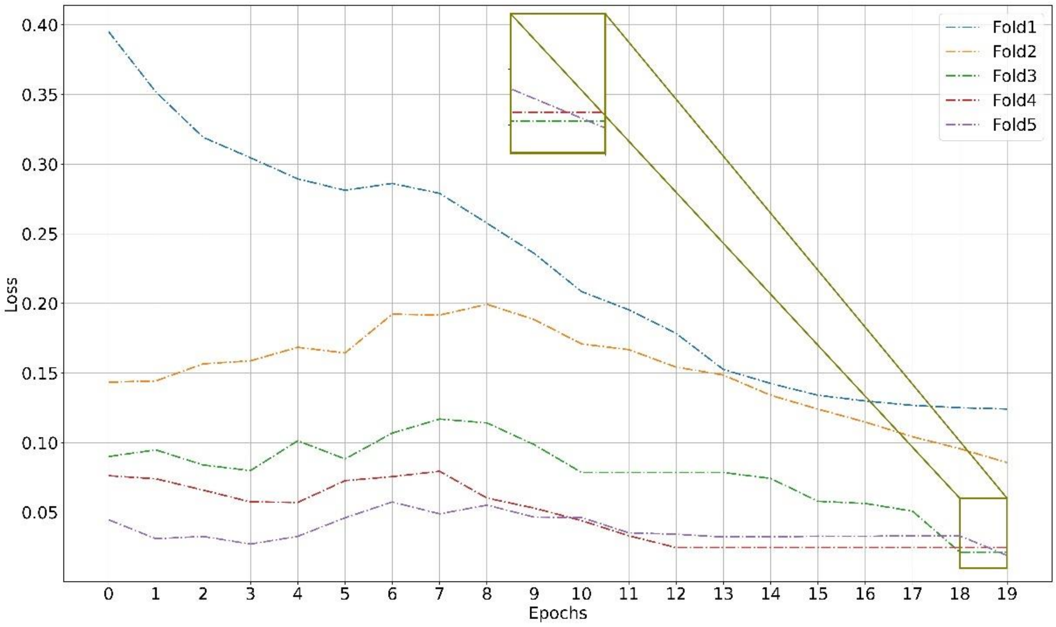

Figure 11. Validation loss performance graph of the proposed model

5. Discussion

In this section, state-of-the-art (SOTA) studies using the same dataset are discussed and compared. In terms of mean accuracy, recall, precision, F1 score, F2 score, and specificity metrics, recent studies on the F-MNIST dataset were compared with CNNTuner.

The proposed model should work in harmony with different filter and activation options. In order for this to be adjusted and the proposed model to work with the Keras Tuner tool, the layer output sizes are provided to work without error. Experimental studies have been carried out more than once in the realization of this. As a result of the evaluations, options other than the parameters selected by the Keras Tuner tool were selected and the performance results were examined. When these results were evaluated, a performance difference between 4% and 10% was determined.

The comparison in Table 7 shows that, in general, the effect of a new activation function or the contribution of residual connections to classification is observed. Although residual connections and activation functions are effective in model performance, this study shows that the performance impact of an accurate deep learning model with appropriate parameters is higher. The abbreviation DNN in Table 7 stands for Deep Neural Network. The DeDNN-oReLU model is parameter optimized by genetic algorithm with oReLU activation function [28]. DNN-ReLU, Residual DNN, DNN-NOM models, like DNN-oReLU, consist of combining different activation types with the DNN method [29].

oReLU represents the nonlinear activation function [28]. The DNN-NOM model, using the keyword Nonlinear Optical Materials (NOM), represents a neural network based on nonlinear optical materials. The Diffractive Processing Unit (DPU) represents the diffractive processing unit that can be configured into different types of models on a large scale.

Bhatnagar et al. developed the Support Vector Classifier (SVC), CNN+BatchNorm, CNN+BatchNorm+Skip and CNN2 models [14]. Each model has different aspects. SVC is used as a classifier in the SVC model. The model called CNN2 represents the CNN model with 2 convolutional layers. Skip in CNN+BatchNorm+Skip refers to skip connection structures. Shan et al. conducted experimental studies on the LeNet-DRLU model with DRLU activation function developed based on the ReLU activation function using the F-MNIST database [32]. Residual Capsule Network (RCNet) is a vector-based network that combines the capsule network structure with residual hops, and the F-MNIST dataset is used to measure the performance of the study. It is seen that 92.91% accuracy rate was achieved in the experimental studies performed on the specified dataset. Figure 12 shows the prediction results obtained using random images on the test data. When examining these results, the actual class label of the model given as input is shown on the y-axis, while the predicted class of the model is shown at the top of the axis.

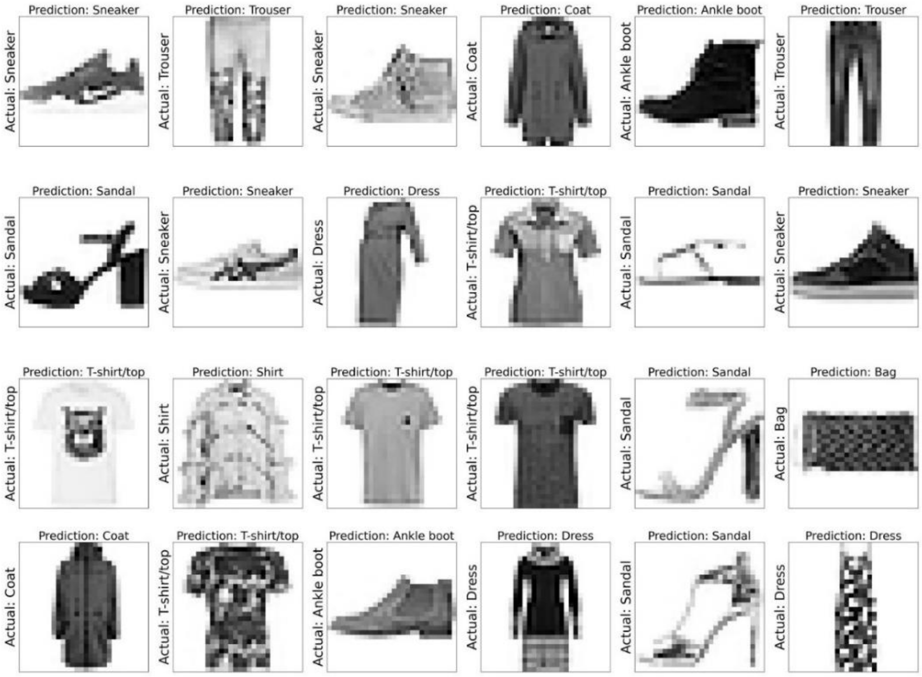

Figure 12. Prediction results of the CNNTuner model

Table 7. SOTA studies in the literature using the same dataset

References	Model	F1 score	Recall	Precision	Specificity	F2 score	Accuracy (%)
[28]	DNN-oReLU	-	-	-	-	-	87.85
[29]	DNN-ReLU	-	-	-	-	-	85.05
[30]	DPU	-	-	-	-	-	84.60
[31]	Residual DNN	-	-	-	-	-	88.40
[29]	DNN-NOM	-	-	-	-	-	88.26
[14]	SVC	-	-	-	-	-	89.70
[14]	CNN2+Batch Norm	-	-	-	-	-	92.22
[14]	CNN2+ BatchNorm+Skip	-	-	-	-	-	92.54
[14]	CNN2	-	-	-	-	-	91.17
[32]	LeNet-DRLU	-	-	-	-	-	92.00
[33]	RCNet	-	-	-	-	-	92.91
Our method	CNNTuner	0.99	0.99	0.99	0.99	0.99	99.18

6. Conclusion

CNN structures with convolution-based automatic feature extraction capability of deep learning can be effectively used in garment retrieval, automatic garment labeling and garment classification. With this motivation, a model called CNNTuner was proposed to be used on a publicly available dataset to classify clothing images. The classification process was applied to the grayscale F-MNIST dataset consisting of 10 classes of 28x28 pixel size with 50,000 training and 10,000 test images. The parameters of each convolution, maximum pooling and activation functions used in the model are determined by the Keras Tuner tool instead of being randomly selected.

The training and test data separated according to the cross validation 5 technique resulted in Fold1, Fold2, Fold3, Fold4 and Fold5. Among these values, the Fold with higher accuracy and lower loss was selected and test predictions were obtained. The search for the right parameter in deep learning models can be a time-consuming process. Parameters selected using the Keras Tuner tool should work seamlessly on the model. When either the activation or filter options are selected, the next layer selected must be compatible with the filter dimensions. For these reasons, the model has been designed to ensure this compatibility.

In future studies, optimized parameters can be used to improve the performance of image segmentation algorithms such as Mask RCNN, UNET and YOLO, which have a long training process. The purpose of this study is to contribute to the literature in this area.

Contributions of the authors

All authors contributed equally to the study.

Conflict of Interest Statement

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The study is complied with research and publication ethics.

References

[1] Y. Seo and K. Shin, “Hierarchical convolutional neural networks for fashion image classification,” *Expert Syst. Appl.*, vol. 116, pp. 328–339, 2019, doi: 10.1016/j.eswa.2018.09.022.

[2] S. G. Eshwar, G. G. P. J, A. V Rishikesh, N. A. Charan, and V. Umadevi, “Apparel classification using Convolutional Neural Networks,” in *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*, 2016, pp. 1–5. doi: 10.1109/ICTBIG.2016.7892641.

[3] K. Hara, V. Jagadeesh, and R. Piramuthu, “Fashion apparel detection: The role of deep

convolutional neural network and pose-dependent priors," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–9, 2016. doi: 10.1109/WACV.2016.7477611.

[4] M. Kayed, A. Anter, and H. Mohamed, "Classification of garments from fashion MNIST dataset using CNN LeNet-5 architecture," in *2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)*, pp. 238–243, 2020,. doi: 10.1109/ITCE48509.2020.9047776.

[5] S. Metlek, "A new proposal for the prediction of an aircraft engine fuel consumption: a novel CNN-BiLSTM deep neural network model," *Aircr. Eng. Aerosp. Technol.*, vol. 95, no. 5, pp. 838–848, Jan. 2023, doi: 10.1108/AEAT-05-2022-0132.

[6] A. Kishwar and A. Zafar, "Fake news detection on Pakistani news using machine learning and deep learning," *Expert Syst. Appl.*, vol. 211, p. 118558, 2023, doi: 10.1016/j.eswa.2022.118558.

[7] M. S. Khan, N. Tafshir, K. N. Alam, A.R. Dhruba, M. M. Khan, A. A. Albraikan, and F. A. Almalki, "Deep learning for ocular disease recognition: an ınner-class balance," *Comput. Intell. Neurosci.*, vol. 2022, 2022.

[8] H. Çetiner and B. Kara, "Recurrent neural network based model development for wheat yield forecasting," *J. Eng. Sci. Adiyaman Univ.*, vol. 9, no. 16, pp. 204–218, 2022, doi: 10.54365/adyumbd.1075265.

[9] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, and D. Novotny, "Common objects in 3d: large-scale learning and evaluation of real-life 3d category reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10901–10911, 2021.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[11] M. A. Morid, A. Borjali, and G. Del Fiol, "A scoping review of transfer learning research on medical image analysis using ImageNet," *Comput. Biol. Med.*, vol. 128, p. 104115, 2021, doi: 10.1016/j.compbiomed.2020.104115.

[12] X. Wang and T. Zhang, "Clothes search in consumer photos via color matching and attribute learning," in *Proceedings of the 19th ACM international conference on Multimedia*, pp. 1353–1356, 2011.

[13] K. V Greeshma and K. Sreekumar, "Hyperparameter optimization and regularization on fashion-MNIST classification," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2, pp. 3713–3719, 2019.

[14] S. Bhatnagar, D. Ghosal, and M. H. Kolekar, "Classification of fashion article images using convolutional neural networks," in *2017 Fourth International Conference on Image Information Processing (ICIIP)*, pp. 1–6, 2017. doi: 10.1109/ICIIP.2017.8313740.

[15] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv Prepr. arXiv1708.07747*, 2017.

[16] H. Çetiner, "Cataract disease classification from fundus images with transfer learning based deep learning model on two ocular disease datasets," *Gumushane University Journal of Science and Technology*, vol. 13, no. 2, pp. 258–269, Jan. 2023, doi: 10.17714/gumusfenbil.1168842.

[17] S. Suganyadevi, V. Seethalakshmi, and K. Balasamy, "A review on deep learning in medical image analysis," *Int. J. Multimed. Inf. Retr.*, vol. 11, no. 1, pp. 19–38, 2022, doi: 10.1007/s13735-021-00218-1.

[18] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," *Int. Conf. Learn. Represent.*, Dec. 2014.

[19] Vijayalakshmi A and Rajesh Kanna B, "Deep learning approach to detect malaria from microscopic images," *Multimed. Tools Appl.*, vol. 79, no. 21–22, pp. 15297–15317, Jun. 2020, doi: 10.1007/s11042-019-7162-y.

[20] S. Zhang, W. Huang, and C. Zhang, "Three-channel convolutional neural networks for vegetable leaf disease recognition," *Cogn. Syst. Res.*, vol. 53, pp. 31–41, 2019, doi: 10.1016/j.cogsys.2018.04.006.

[21] S. Zhang, S. Zhang, C. Zhang, X. Wang, and Y. Shi, "Cucumber leaf disease identification with global pooling dilated convolutional neural network," *Comput. Electron. Agric.*, vol. 162, pp. 422–430, 2019, doi: 10.1016/j.compag.2019.03.012.

[22] N. Saxena, V. Sharma, R. Sharma, K. K. Sharma, and S. Gupta, “Design, modeling, and frequency domain analysis with parametric variation for fixed-guided vibrational piezoelectric energy harvesters,” *Microprocess. Microsyst.*, vol. 95, p. 104692, Nov. 2022, doi: 10.1016/j.micpro.2022.104692.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012.

[24] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *European conference on computer vision*, pp. 818–833, 2014.

[25] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv Prepr. arXiv1409.1556*, 2014.

[26] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” 2017.

[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Jun. 2016. doi: 10.1109/CVPR.2016.90.

[28] C. Dong, Y. Cai, S. Dai, J. Wu, G. Tong, W. Wang, Z. Wu, H. Zhang, and J. Xia, “An optimized optical diffractive deep neural network with OReLU function based on genetic algorithm,” *Opt. Laser Technol.*, vol. 160, p. 109104, 2023, doi: 10.1016/j.optlastec.2022.109104.

[29] Y. Sun, M. Dong, M. Yu, L. Lu, S. Liang, J. Xia, and L. Zhu, “Modeling and simulation of all-optical diffractive neural network based on nonlinear optical materials,” *Opt. Lett.*, vol. 47, no. 1, pp. 126–129, 2022, doi: 10.1364/OL.442970.

[30] T. Zhou, X. Lin, J. Wu, Y. Chen, H. Xie, Y. Li, J. Fan, H. Wu, L. Fang, and Q. Dai, “Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit,” *Nat. Photonics*, vol. 15, no. 5, pp. 367–373, 2021, doi: 10.1038/s41566-021-00796-w.

[31] H. Dou, Y. Deng, T. Yan, H. Wu, X. Lin, and Q. Dai, “Residual D2NN: training diffractive deep neural networks via learnable light shortcuts,” *Opt. Lett.*, vol. 45, no. 10, pp. 2688–2691, 2020, doi: 10.1364/OL.389696.

[32] C. Shan, A. Li, and X. Chen, “Deep delay rectified neural networks,” *J. Supercomput.*, vol. 79, no. 1, pp. 880–896, 2023, doi: 10.1007/s11227-022-04704-z.

[33] J. Zhang, Q. Xu, L. Guo, L. Ding, and S. Ding, “A novel capsule network based on deep routing and residual learning,” *Soft Comput.*, 2023, doi: 10.1007/s00500-023-08018-x.