International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 41, No. 5, October, 2024, pp. 2221-2231
Journal homepage: http://iieta.org/journals/ts

Effects of Different Pre-Training Algorithms on Video Classification Results

Elif Akarsu*, Ibrahim Yucel Ozbek, Tevhit Karacali

Department of Electrical-Electronics Engineering, Ataturk University, Erzurum 25050, Turkey

Corresponding Author Email: elif.akarsu@atauni.edu.tr

https://doi.org/10.18280/ts.410501

Received: 20 December 2023
Revised: 29 March 2024
Accepted: 15 June 2024
Available online: 31 October 2024

Keywords:
video classification, pre-training algorithm, Bi-LSTM, t-SNE, feature histogram

ABSTRACT

The dataset used in the study consists of 600 videos obtained by us from social media (Instagram). The data set includes 6 different video files. These files contain videos of different subjects and such as dance, driving car, fitness, make-up, mukbang. Dataset, containing 600 videos, 100 videos in each class. Additionally, the length of each video is approximately 1000 and the number of frames varies accordingly. Additionally, the videos are in mp4 format and received and processed in this format. All frames were used in classification. Because video combines multiple picture frames to create a series of images, classification is performed on this image data. The image feature information of these frames was extracted with the help of three different pre-training algorithms. These features include all features in the framework. These algorithms; ResNet50, ResNet101, GoogleNet. It is possible to collect data with different algorithms during the pre-training phase, observe the results clearly with different algorithms, and classify them in MATLAB by taking 1000 features from each algorithm. To check whether the features obtained from each pre-training algorithm were classifiable, before proceeding to the classification stage, the videos were found to be classifiable by t-SNE and pixel analysis. After analyzing the features of three different pre-training algorithms, classification was started. In the classification phase, the Bi-LSTM classifier was used because it is a time-dependent function and classifies all the data after reviewing it in detail. In conclusion; The effects of videos belonging to these six different classes on the classification accuracy of the results obtained with the same parameters but different pre-training algorithms were investigated.

1. INTRODUCTION

This article focuses on a classification algorithm consisting of 600 videos and 6 different categories. The videos used in the classification were collected by us from social media, taking into account certain standards. The main purpose of collecting videos instead of using a ready-made data set is to ensure that the study is original. However, at this point, collecting 600 videos and eliminating those that are not suitable for classification poses a challenge in terms of both time loss and providing sufficient space for data storage. Additionally, another difficulty is that unnecessary data in video frames that are not carefully selected can have negative effects on the content of the extracted features and therefore on classification. However, since the data set collected by us is original and can have a positive impact on the video classification study, we focused on evaluating the results obtained in the classification after collecting the data, sorting it in detail and removing unnecessary data. At this point, each resulting video consists of image frames. These picture frames include many features. Features are obtained from neural networks with different layers and structures.

Neural networks are the key factor in video classification applications. Therefore, neural networks are a powerful tool for video, audio, and image classification. With neural networks, data can be learned, predictions can be made about the data, and it is possible to classify the data according to their patterns. Additionally, generalizations can be made as well as classification [1]. This allows a variety of operations to be performed, such as image processing, speech recognition, and decision-making. The most important feature of neural networks is the ability to learn from input data; This proves that the more examples, the more data it can learn with the generalization feature of neural networks. Predictions can be made on new and comprehensive data with the features learned as a result of training. This process makes it useful in a wide variety of applications where the system must be restored in response to new and unforeseen situations [2].

Neural networks also can learn from raw input data, allowing them to automatically extract features. This allows the neural network to recognize images. Neural networks have numerous advantages and are used in many applications, starting from simple data processing to the classification of complex data. Therefore, the network structure may vary depending on the complexity of the problem to be solved [3]. It inputs input data into the layer and then passes it through one or more hidden layers, and then passes it to the next layer. and ultimately produces an output. The training process involves an input dataset and multiple corresponding outputs; The main goal here is to minimize the difference between predicted and actual results [4]. thus, it may be possible to make predictions about new and unseen data. This is where

Deep Convolutional Neural Networks come into play. CNNs are a special form of deep learning algorithms. Convolutional neural networks make it easier to solve unsolved problems in traditional machine learning techniques. Moreover, its only advantage is not classification, it also eliminates the need to manually obtain features by extracting the desired feature from the raw image [5]. Image Processing applications often require more than image classification. Therefore, classifying multiple images after they come together in a moving structure is a difficult but fruitful scientific problem that promises many applications [6]. Our aim in examining this problem is to develop a system that can be added to an ordinary image-based surveillance system and can detect and classify objects in real time. Considering the abundance and complexity of time-dependent video information, the algorithms and methods to be used in this real-time system must work quickly and reliably [7]. In addition, regarding the unique aspects of the study, it creates unique value compared to other video classification studies, since dimension reduction analysis is performed before classification and the data set used is unique and is subject to a more supervised classification using the Bi-LSTM classifier.

In the literature review of the study, firstly the data sets to be used in classification were examined and it was seen that ready-made data was generally used. Since this situation will negatively affect the originality of the study to be carried out and considering that ready-made data has been used many times before, the situation of creating an original data set has arisen. After collecting the necessary data and removing the unnecessary ones, three different pre-training algorithms, which are widely used in video classification studies in the literature, were used to obtain the features of the video frames. However, it is not very common to compare the three pre-training algorithms used in this study with each other. These 3 different pre-training algorithms have a wide network structure and have an important place in providing more accurate information. Therefore, GoogleNet, ResNet50, and ResNet 101 were chosen as pre-training algorithms. When it comes to the classification stage, video classification with LSTM has come to the fore in the studies in the literature because it is time-dependent, the number of network layers can be adjusted and it is very successful in multi-class data classification. However, in this study, the Bi-LSTM classification network was used because the data was examined in more detail starting from the classification stage and it has more unique aspects, is less used in the literature, and contains the advantages of LSTM.

2. METHODOLOGY

In this study, 600 videos collected from different accounts on social media (Instagram) are used. These videos are in 6 different classes and there are 100 videos in each class. These include videos titled mukbang, fitness, makeup, dance, swimming, and driving car. Since the videos are collected from different accounts, different people perform the same action. Thus, a wider range of data is obtained. The size of each video is 720*720, and each video consists of approximately 1000 frames. Each video contains 30 frames per second. Certain factors were taken into account to include 600 videos in the study. The videos consist of 6 different classes. The reason for choosing these 6 classes is that the effects of performing different actions in completely different areas are investigated and this number is sufficient and necessary to describe many actions. In other words, it was aimed to investigate the effect of different actions such as eating, applying make-up, and driving on classification. During the classification phase, factors such as the fact that videos in each category contain clear visuals of the action performed, videos with unnecessary details are not included in the classification, image sizes are compatible with each other, and there are videos of more than one type of action in the same category in the data set are taken into account so that learning is sufficient in every class. For example, when choosing makeup videos, the goal is to introduce different types of actions to the network, such as applying lipstick, applying mascara, applying foundation. or performing the same action by more than one person and in more than one way makes it easier to obtain more detailed data. Thus, the resulting data set contains different details. In the first stage, pre-training algorithms come into play to obtain features from the video collection stage.

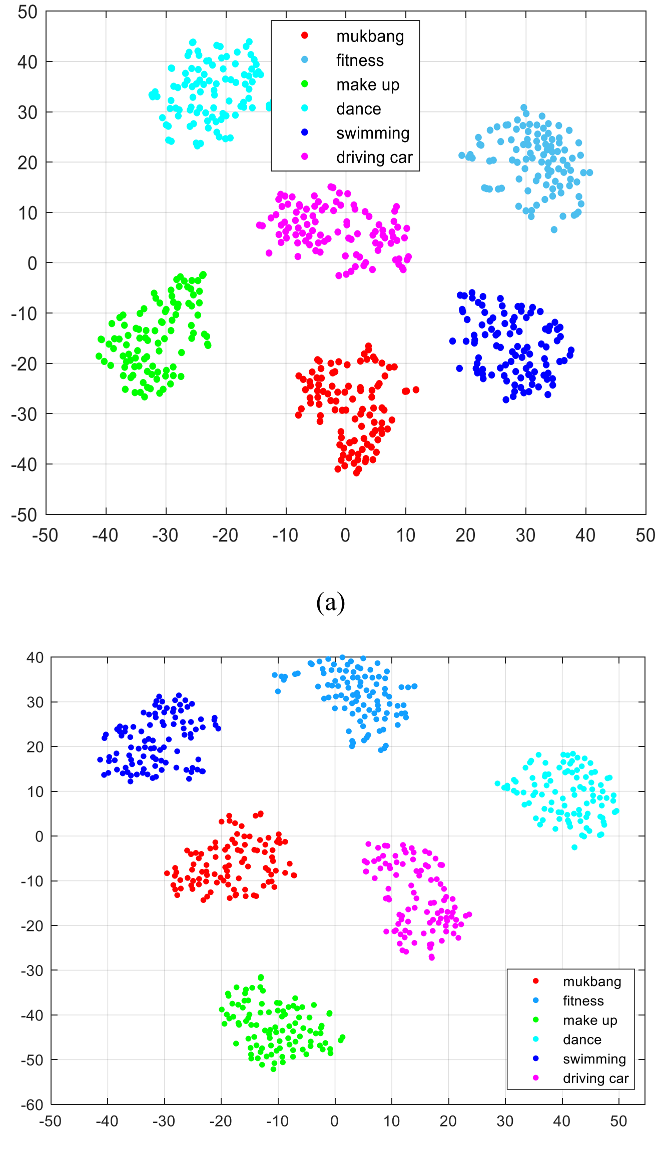
Three different pre-training networks consisting of ResNet50, ResNet101, and GoogleNet, were used to obtain information from each framework. The main reason for using these three pre-training networks is the large number of layers in different numbers and the different depths of the 3 different networks. It was wanted to investigate what the effect of having different networks that have many layers and varying depths would be on classification. Then, the features obtained with pre-training algorithms were reduced in size.

GoogleNet; It has 22 deep layers. Regardless of the initial size of the images entering the layers, they are converted to 224*224*3 by the algorithm. And it has a total of 144 layers and 170 connections [8]. ResNet 50 has 50 deep layers. Regardless of the initial size of the images entering the layers, they are converted to 224*224*3 by the algorithm and it has a total of 177 layers and 192 connections [9]. ResNet 101; It has 101 deep layers. Regardless of the initial size of the images entering the layers, they are converted to 224*224*3 by the algorithm and it has a total of 347 layers and 379 connections [10]. In other words, there are three different network structures with different depth, connections and layer numbers. At the classification stage, 80% of the features obtained are allocated for training and 20% for testing, this is done using a 5-fold. All data is multiplied by 5 tests and training data are divided by each multiplier. During this division process, different data is trained on each fold.

2.1 t-SNE analysis

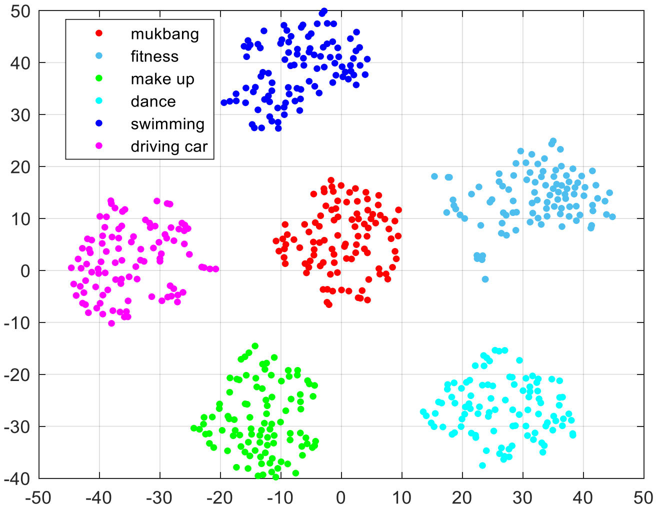
t-SNE is a linear technique used to reduce feature dimension [11]. It is especially used for visualizing high-dimensional feature matrices. It is widely applied in image processing and audio processing. t-SNE expresses the closeness and distance of data probabilities. Distances in space are expressed with a Gauss distribution, while distances in embedded space are expressed by t-distributions [12]. This makes the structure of t-SNE sensitive. t-SNE allows the structure to be revealed very broadly on a single map. Multiple, distinct, manifolds or data sets emerge. Collecting some points in the center reduces the total. While Local Linear Embedding and robustities are suitable for extracting a single continuous low-dimensional manifold, t-SNE extracts clustered groups of local samples by focusing on recorded local data [13]. The ability to group samples locally can be useful for visually resolving a data setup that contains multiple manifolds at once, as in a digital data setup. The most current and best solution for reducing dimensions is the t-SNE technique [14]. Therefore, the t-SNE dimensionality reduction algorithm was used in this study.

Dimension reduction analysis is a data visualization problem, which is the visualization of these reduced dimensions as a result of reducing the features obtained from the frames of each video into two dimensions. The main idea of the t-SNE algorithm is to find a low-dimensional representation that preserves the distances between points as much as possible. t-SNE starts with an arbitrary low-dimensional representation for each data point and tries to keep points close together and points far apart in the original space. t-SNE gives more importance to points that are close together rather than maintaining distance between points that are far apart. Due to all these advantages, 1000 features were extracted from each image, frame in the videos for all three pre-training algorithms. The extracted features were visualized by reducing them to two dimensions [15]. Results in Figure 1; It showed that after the dimensions of the extracted features were reduced and visualized, 100 videos from 6 classes were grouped within themselves. It was also observed that each of the 600 videos could be classified within itself.



(a)

(b)



(c)

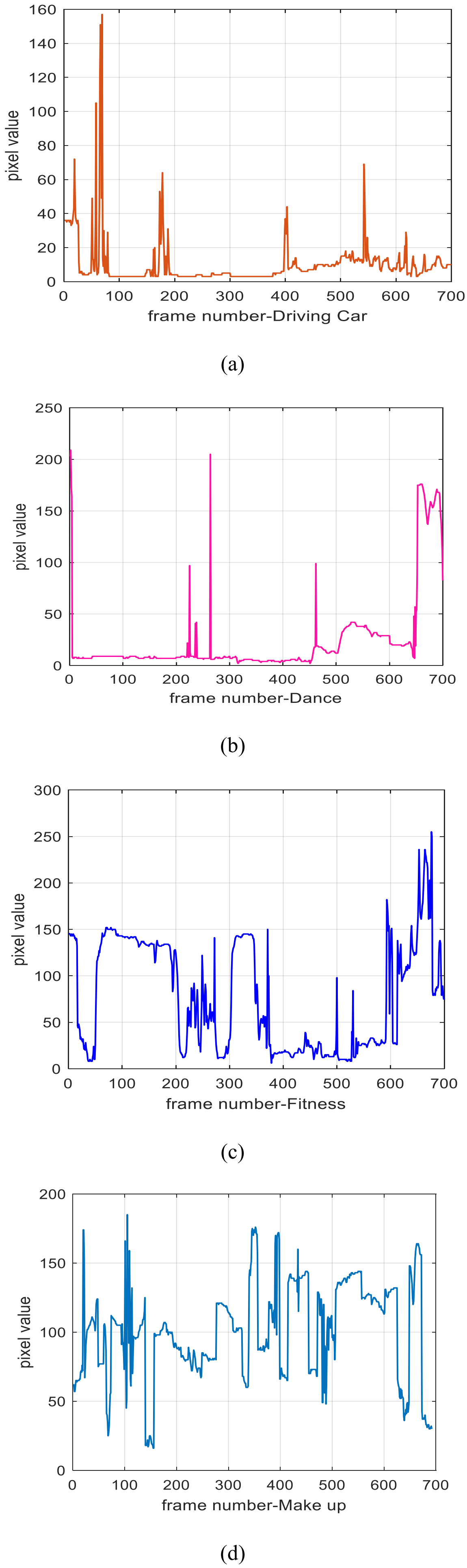
Figure 1. t-SNE analysis results (a) ResNet101, (b) ResNet50, (c) GoogleNet

2.2 Pixel analysis

Pixel analysis is a numerical evaluation of the density of the pixels that make up the image. That is, there is a numerical value corresponding to the pixel position. Therefore, these values are also very effective in classification. because one of the parameters that make up the obtained features is pixel values. Pixel analysis is the graphing of the values as a result of dividing each video into images and taking the average of the values corresponding to the pixel information taken in the same region of the image frames. This reveals that the mathematical results corresponding to pixels are of great importance in the feature determination process, as they create different and very specific images. Hence, pixel analysis is another data visualization technique used to check whether the collected data is suitable for classification. Thus, a point close to the center of the 100 videos in each class was selected and the pixel values corresponding to this point were analyzed. First of all, each video is divided into individual images and the images are stored for each video.

To select the pixel to be analyzed, the pixel (692, 832) was selected from the center of the images taken from the video using the image analyzer from the MATLAB applications section. and the values corresponding to this pixel were kept in the Excel file with .csv extension, and then the graphics in picture 2 were obtained by averaging the pixel values obtained for 100 videos. In other words, the pixel values of 100 videos in the first class were averaged, and the average values of 100 videos in the second class were taken.

This process was done for 6 classes, that is, 600 videos. The values corresponding to the pixels of the first 700 frames are shown for 6 different classes in Figure 2. The vertical axis represents the pixel value and the horizontal axis represents the frame number. Values range from 0 to 250 for each frame. The fact that the results are different for each class is a desirable situation and constitutes one of the components in distinguishing the videos.

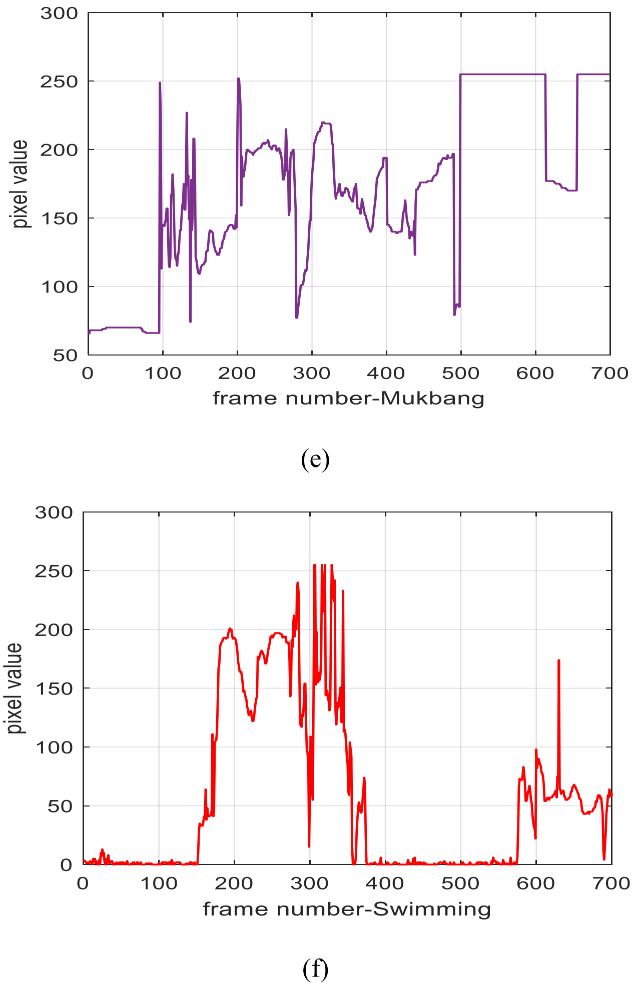


(a)

(b)

(c)

(d)



(e)

(f)

Figure 2. Pixel analyzes of 6 different classes

Figure 2 shows the average pixel analysis of 6 different classes. It is seen that quite distinct and characteristic results are obtained for each class. Thus, it can be seen that conditions such as color and intensity of the videos to be classified will be very effective in extracting features.

2.3 Video classification

The first step in video classification is to line up each image frame that makes up the video. Then, features are retrieved from each of the sequenced frames. The number of these feature retrievals is 1000 for each of the three pre-training algorithms [16]. 1000 features are extracted from the FC1000 layer of ResNet 50 and ResNet 101 and the loss-3 classifier layer of GoogleNet. In the last stage, it is sent to the classifier and classified.

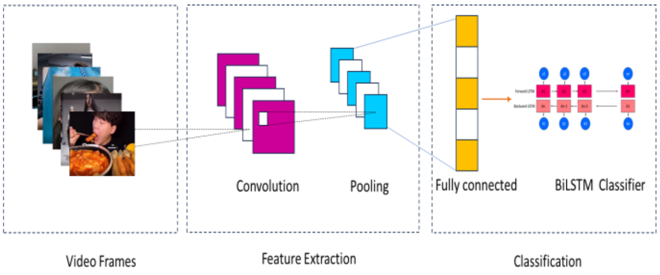


Figure 3. Algorithm structure from video classification

In Figure 3, we see a video classification algorithm. First of all, the frames of the videos belonging to each of the 6 classes we have are arranged one after the other. Consecutive frames are sent to a pre-training algorithm so that their features can be extracted. The obtained features are classified in MATLAB by sending them to the Bi-LSTM classifier. The Bi-LSTM classifier is used because it is a time-dependent classifier. Thus, it is a widely used algorithm for the classification of video in close pattern with time information. Let the attribute entering the network be denoted by X, and let the resulting label value be denoted by Y. Our first goal is to find the f function that provides the relationship between X and Y. Our second goal is to predict Y data in response to new X data. The function obtained from here is it is nonlinear and has an iterative solution.

$$hw(x) = (w0*x1/w1+x2)+w3*x3 \tag{1}$$

$$\mathrm{Y} = hw(x)+\dot{\varepsilon} \tag{2}$$

$$\dot{\varepsilon} = \mathrm{Y}-hw(x) \tag{3}$$

Here $hw(x)$ constitutes the prediction feature, while $\dot{\varepsilon}$ represents the error data. The combination of prediction and error results in recorded actual values. The extracted features are turned into a matrix and sent to the Bi-LSTM classifier. Choosing the classification parameters correctly is important in order to avoid overfitting or underfitting of the system and to ensure that the modal gives correct results. One of the important parameters here, MiniBatchSize, is the size of the minibatch to be used for each training iteration, specified as a comma-separated pair of a positive integer.

Additionally, the epoch number expresses how much of all the training data is exposed to the network during the training process. The learning rate determines how fast the network can learn from data. This rate affects how quickly the network can adapt to new information and how quickly it can forget existing information. Low learning rates cause the network to learn slowly. The classification process was carried out with the parameters in Table 1 for all three pre-training algorithms.

Classification is done in 5 folds, the purpose of doing this is to review all the data by dividing it into 5 parts and to prevent any data that does not fall into the classification. In the context of Machine Learning (ML), Cross-Validation is a critical statistical technique used to evaluate the performance and generalizability of a particular predictive model or algorithm [17]. This method makes a model capable of performing extremely well on training data while also trying to minimize problems such as overfitting that occur if it performs poorly on new data [18]. Given the important role predictive models such as language classification and image classification play in AI applications, cross-validation is an essential component of the model evaluation process that ensures high-quality performance across different datasets and scenarios. Figure 4 shows the visual structure of cross-validation.

Table 1. Classification parameters used in feature classification

Classification Parameter	Value
MiniBatchSize	93
MaxEpoch	25
InitialLearnRate	0.01
Optimizer	adam
GradientThreshold	25

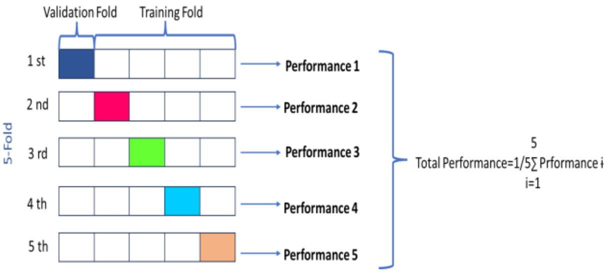
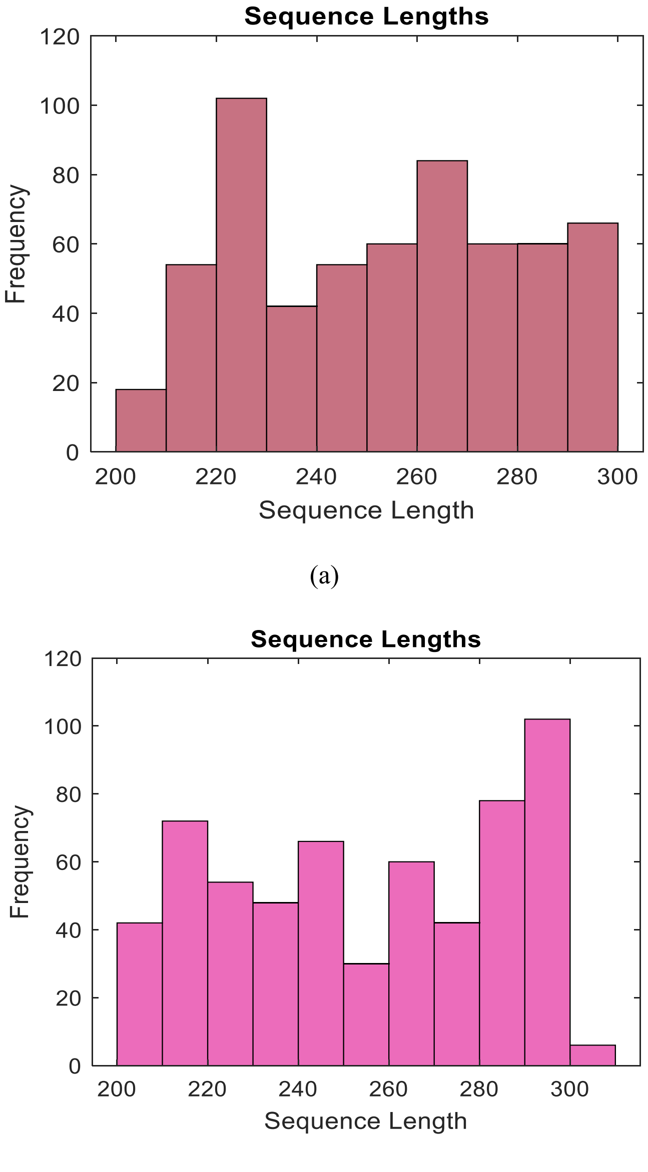


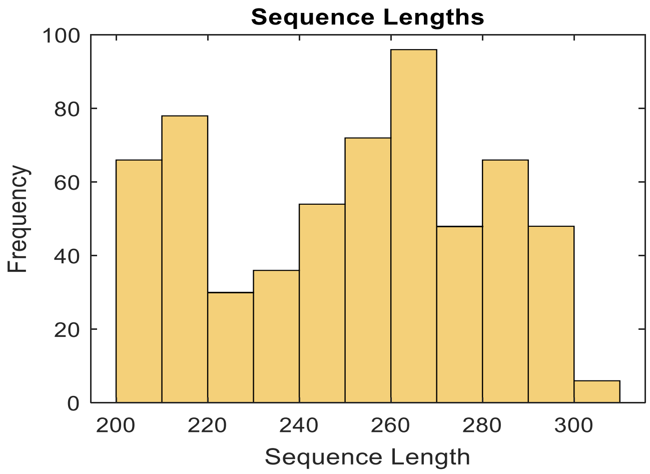
Figure 4. The schematic structure of cross-validation

It involves splitting the existing dataset into two or more distinct subsets called "folds." The train data is processed for the number of floors here and tested on the remaining floors. By repeating this process many times. In this way, a more accurate and consistent model will be created. It is 5-fold cross-validation where the data is divided into 5 equal subsets and the network is trained and tested 5 times using a dissimilar subset as test data each time. After all iterations are completed, the results are averaged to determine the final model performance [19]. In addition, the features taken in the pre-training algorithms are sent to the Bi-LSTM classifier in certain proportions and amounts.



(a)

(b)



(c)

Figure 5. Feature distribution histograms (a) ResNet101, (b) ResNet50, (c) GoogleNet

If overfitting occurs, the training error decreases and the test data increases. Moreover, the data is not learned but memorized. For this reason, when the test data arrived, it was not similar to the train data they had memorized, so it could not be classified correctly. It is possible to talk about high variance. The main reason for this is to prevent overfitting by introducing too much data into the network at once. Figure 5 shows the orientation histograms of the sent features in size 1*144. As can be seen, it is sent to the network in series between 200-300. The vertical axis (frequency axis) of the graphics in picture 3 expresses how large series of attributes are sent to the network. The horizontal axis represents the number of attributes sent in each cycle. This adjustability allows attributes to be sent in quantities that the network can handle. Thus, all data is reviewed. It also prevents problems such as overfitting.

In the classification phase, Bi-LSTM classification algorithm consisting of 780 hidden layers and 0.5 dropout is used. Bi-LSTM is an algorithm consisting of a bidirectional LSTM classifier [20]. Bidirectional LSTM is composed of two different LSTM layers that process input data forward and backward. As seen in Figure 6, both outputs of these two layers are used to give the final outputs. Additionally, LSTM often encounters poor performance due to the vanishing gradient problem. To overcome this problem, a two-way LSTM structure has been proposed. In the classification algorithm structure in one-way forward LSTM, only previous features are reviewed, but subsequent features are not taken into account. However, with the Bidirectional LSTM classifier, it is possible to use previous and future features effectively in feature classification. Unlike the LSTM network, the Bi-LSTM network has two equidistant structures in the forward and reverse directions [21]. Additionally, the number of layers and dropout value can be changed to enable classification with the Bi-LSTM classifier. Bi-LSTM can be expressed as the following equations; st is the hidden state output; and G is the weight matrix. Also, G_{ys} is a weight connecting input y to hidden layer s. b, bias; tanh is the activation function.

$$s_t^f = \tanh\left(G_{ys}^f y_t + G_{ss}^f S_{t-1}^f + b_s^f\right) \tag{4}$$

$$s_t^b = \tanh\left(G_{ys}^b y_t + G_{ss}^b s_{t+1}^b + b_s^b\right) \tag{5}$$

$$x_t = G_{sx}^f S^f{}_t + G^b{}_{sx} s^b{}_t + b_x \tag{6}$$

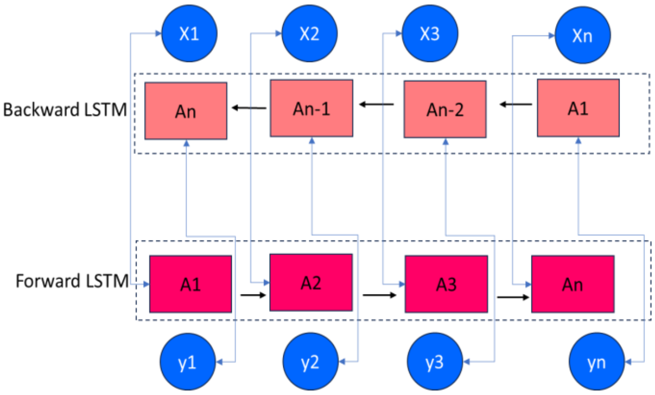
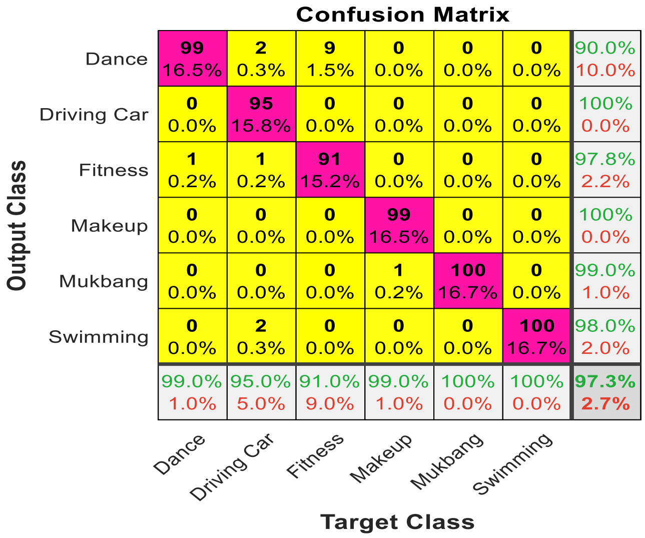


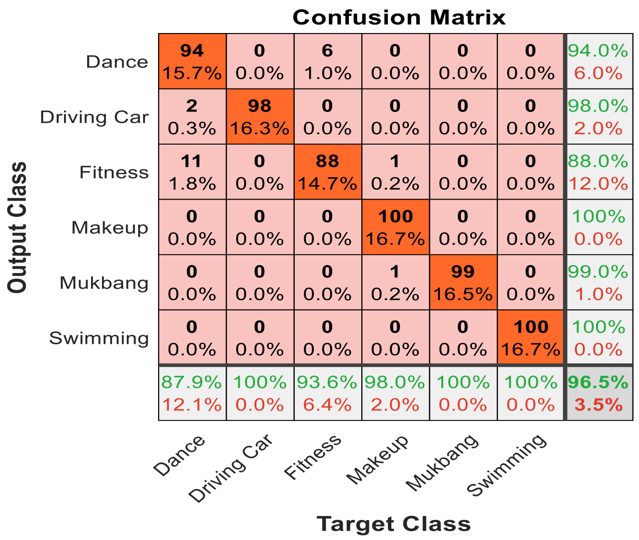
Figure 6. Structure of bidirectional LSTM

3. RESULTS

All networks were re-trained, allowing us to obtain as many features as we wanted. With this method, it is possible to obtain any number of features from any desired location. Then, all features were classified using the Bi-LSTM classifier in 5 cycles [22]. Thus, the features obtained from different pre-training algorithms were classified separately and the results were compared. The videos used in classification consist of 6 different classes. 100 videos in each class, 600 videos in total, were subjected to classification. And classification results are obtained for each algorithm. For each of ResNet50, ResNet101 and GoogleNet, 1000 features were obtained from the last fully connected layer. 1000 features appear to be a sufficient amount to detect the features of a picture frame. After the features were obtained, the classification step for each algorithm was performed 5-fold. The purpose of this is to obtain more reliable results and to classify in a controlled way without introducing too much data into the network at once. In each cycle, testing and training data are randomly split. This process is done 5 times in 5 layers [23]. This way, all data is reviewed, overfitting is prevented, and the most necessary information is obtained. As a result of the classification, the confusion matrices in the classification results of 3 different pre-training algorithms are shown in Figure 7.

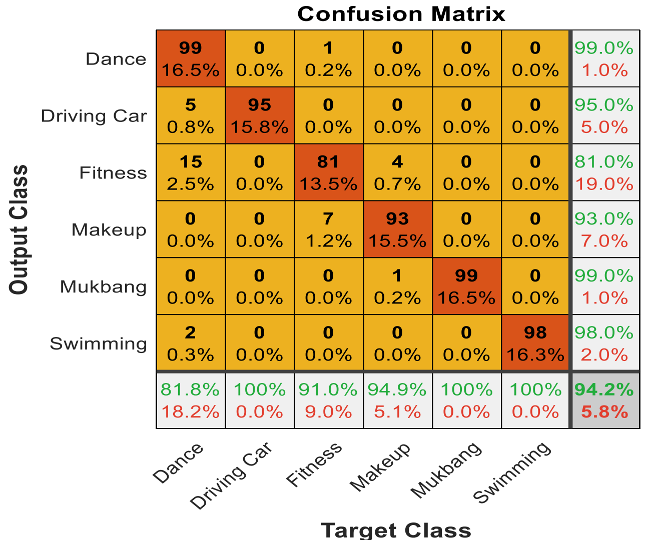


(a)



	Dance	Driving Car	Fitness	Makeup	Mukbang	Swimming	
Dance	94 15.7%	0 0.0%	6 1.0%	0 0.0%	0 0.0%	0 0.0%	94.0% 6.0%
Driving Car	2 0.3%	98 16.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.0% 2.0%
Fitness	11 1.8%	0 0.0%	88 14.7%	1 0.2%	0 0.0%	0 0.0%	88.0% 12.0%
Makeup	0 0.0%	0 0.0%	0 0.0%	100 16.7%	0 0.0%	0 0.0%	100% 0.0%
Mukbang	0 0.0%	0 0.0%	0 0.0%	1 0.2%	99 16.5%	0 0.0%	99.0% 1.0%
Swimming	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 16.7%	100% 0.0%
	87.9% 12.1%	100% 0.0%	93.6% 6.4%	98.0% 2.0%	100% 0.0%	100% 0.0%	96.5% 3.5%

(b)



	Dance	Driving Car	Fitness	Makeup	Mukbang	Swimming	
Dance	99 16.5%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	99.0% 1.0%
Driving Car	5 0.8%	95 15.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	95.0% 5.0%
Fitness	15 2.5%	0 0.0%	81 13.5%	4 0.7%	0 0.0%	0 0.0%	81.0% 19.0%
Makeup	0 0.0%	0 0.0%	7 1.2%	93 15.5%	0 0.0%	0 0.0%	93.0% 7.0%
Mukbang	0 0.0%	0 0.0%	0 0.0%	1 0.2%	99 16.5%	0 0.0%	99.0% 1.0%
Swimming	2 0.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98 16.3%	98.0% 2.0%
	81.8% 18.2%	100% 0.0%	91.0% 9.0%	94.9% 5.1%	100% 0.0%	100% 0.0%	94.2% 5.8%

(c)

Figure 7. Confusion matrix results (a) ResNet101, (b) ResNet 50, (c) GoogleNet

Values located on the diagonal of the confusion matrix are correctly predicted values. However, values outside the diagonals appear as false values. As this value decreases, accuracy increases. When the accuracy matrices were examined, the highest accuracy value was obtained from ResNet101. Then, it is seen that the lowest accuracy is achieved with ResNet50 and GoogleNet. The reason why the highest accuracy is achieved with ResNet 101 is that the number of layers is higher than the other two and the depth of the network is high. It has been observed that as the depth of the network and the number of layers decrease, the classification accuracy also decreases. Accuracy is the ratio of how much of the total number of data is classified correctly. The primary priority of the accuracy value is to send the attribute data to be classified for classification in balanced and accurate quantities. The accuracy of the predictions made by the classifier can be adjusted in advance. Figure 8 shows the accuracy values obtained for all three pre-training algorithms.

$$Accuracy = (\mathrm{TP} + \mathrm{TN}) / TotalNumberofSamples \qquad (7)$$

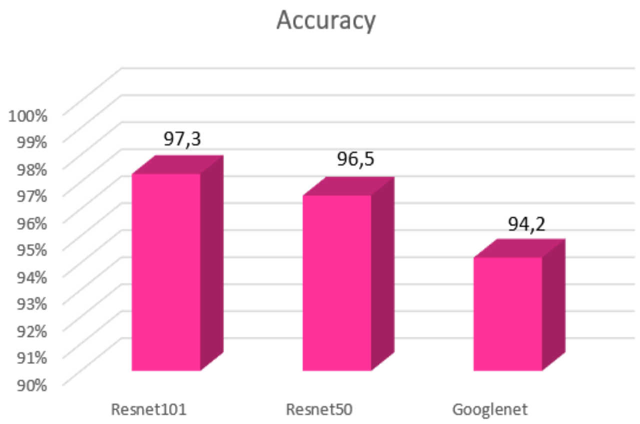


Figure 8. Accuracy results of three pre-training algorithms

Hypothesis testing is the determination of the accuracy of a hypothesis within a statistical confidence interval. The methods used for statistics, there are two possible types of statistical analysis when testing hypotheses error is mentioned. These errors are called Type 1 and Type 2 errors. Type-I Error; Although the sample results (Measured or Perceived) are true, that is, it is accepted as false and rejected. Type I errors cause false positives (False Positive -FP) [24]. Type-II Error: Although the sample results (Measured or Perceived) are wrong, the hypothesis is the acceptance of Truth in reality. Type II errors lead to false negatives (False negative-FN) is equivalent [24].

Based on this; The classification of actually true (actual positive) and correctly predicted (predicted positive) values is called TRUE POSITIVE (TP) [25]. Not actually accurate and not correctly predicted values are called TRUE NEGATIVE (TN) [25].

The classification of actually inaccurate (actual negative) and correctly predicted (predicted positive) values is called FALSE POSITIVE (FP). The classification of values that are actually true (actual positive) and not predicted correctly (predicted negative) is called FALSE NEGATIVE (FN). These explained terms are positioned and presented in Table 2. We can obtain accuracy, recall, precision, and F1 score values with the help of the table.

Table 2. Basic explanation of the confusion matrix

Positive	Negative
TP	FP
FN	TN

The desired situation as a result of the evaluation obtained in this way; By using a correctly constructed model (appropriately selected feature and classifier pair in this study), the magnitudes in the main diagonal can be ensured to be as large as possible. It is possible to easily visually evaluate model performance, especially when all cells in this matrix are normalized by dividing by the relevant "target sample" numbers in the data set.

Precision: A good classifier should have a sensitivity of 1. When the numerator and denominator are equal, it becomes 1. As the denominator increases, the precision value decreases [26].

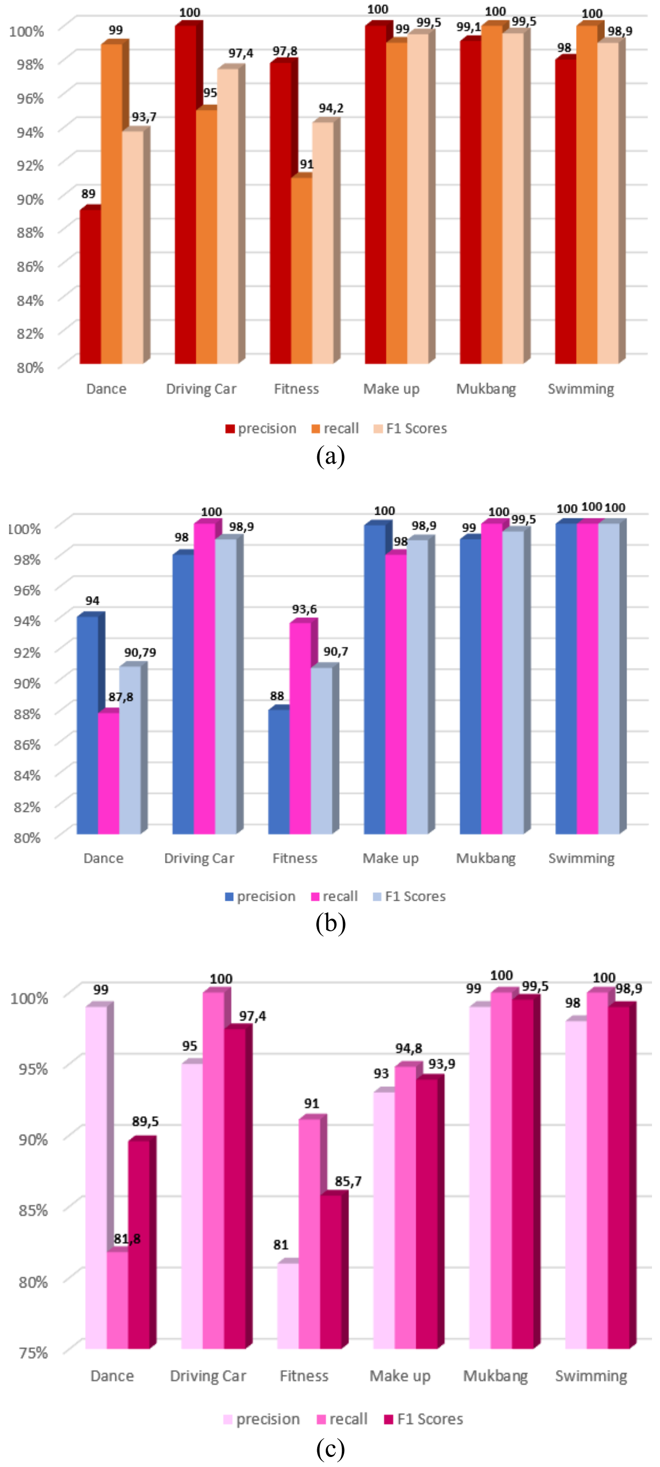
$$Precision = \mathrm{TP} / \mathrm{TP} + \mathrm{FP} \qquad (8)$$

Recall: The recall value of a good classifier should be 1. Denominator value such as precision value. As it grows, the Recall value decreases [26].

$$Recall = \mathrm{TP} / \mathrm{TP} + \mathrm{FN} \tag{9}$$

F1 Score: It is the harmonic mean of recall and precision values. It is a combination of precision and recall. If the Precision and Recall values are large, the F1 score is also large [26].

$$\text{F1-Score}= 2^{*} \text{ recall } {}^{*}\text{precision / recall +precision} \tag{10}$$



(a)

(b)

(c)

Figure 9. Precision, recall, F1 score calculation results (a) ResNet101, (b) ResNet50, (c) GoogleNet

Calculations were made to obtain more detailed information from each of the confusion matrices obtained as a result of the classification of all three pre-training algorithms. It will be possible to see the analysis of the classification made as a result of these calculations in more detail. Based on this, parameters such as True positive rate and true negative rate, specificity were found and compared for all three different algorithms and 6 different classes. It can also be expressed more simply as the ratio of True Positive to False Positive. True positive value, True Positive value; It is found as the ratio of the sum of the True Positive value and the False Negative value. This ratio is also called sensitivity. Sensitivity, recall, and TPR mean the same thing.

$$\mathrm{TPR} = sensitivity = Recall = \mathrm{TP} / \mathrm{TP} + \mathrm{FN} \tag{11}$$

Specificity is expressed as the ratio of true negative results to the sum of false positive and true negative results, and it is desired to be high because it expresses how high the selectivity of the classification process is of all the examples in the data set that are in the negative class, it is true that they are in the negative class. It is the number of examples that indicate that it was predicted in some way.

$$\text{Specificity=TN/FP+TN} \tag{12}$$

Additionally, the False Positive rate can also be calculated as 1-Selectivity. It is the opposite of selectivity. It is required to be quite low. It is equal to the ratio of false positive values to the false positive and true negative. The representation of this will be as

$$FalsePositiveRate = \mathrm{FP} / (\mathrm{FP} + \mathrm{TN}) = 1 - Specificity \tag{13}$$

In Figure 9, the precision, recall, F1 score calculation results of 3 different pre-training algorithms are presented graphically, as well as the results of the TPR values obtained in Figure 10, and in Figure 11, the specificity values are calculated and presented in graphs.

When the average value of the Specificity of all three algorithms is examined, the results of ResNet 101 and 50 are respectively 99.4% and 99.3%. This means that we can say that the selectivity of both algorithms is the same, but the selectivity value of GoogleNet is 98%, It was found to be 6, which is a value very close to the other two. As a result, the selectivity of all three pre-training algorithms gave very successful results. Accordingly, FPR gave very low results for all three pre-training algorithms, which is desirable. In Figure 12, the obtained FPR values are presented graphically and compared.

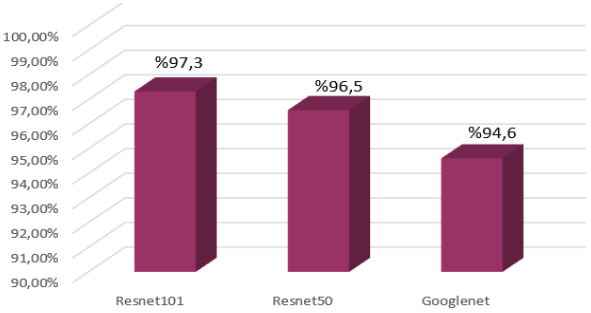


Figure 10. TPR values of ResNet101, ResNet50 and GoogleNet

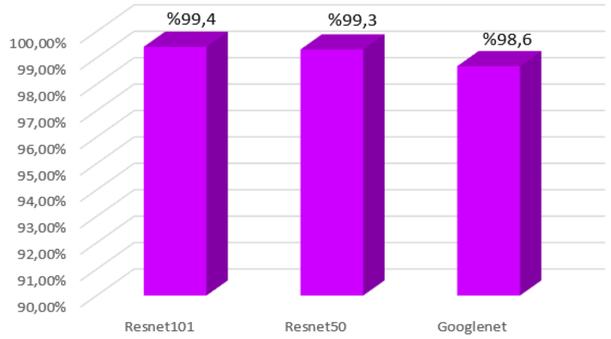


Figure 11. Specificity values of ResNet101, ResNet50 and GoogleNet

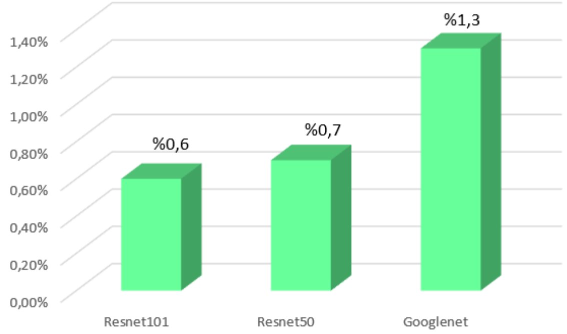
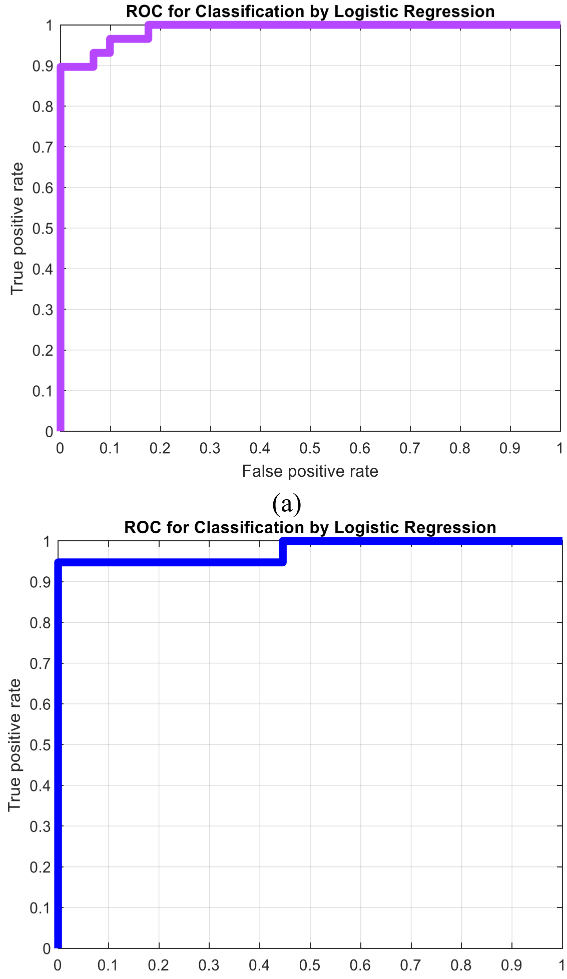
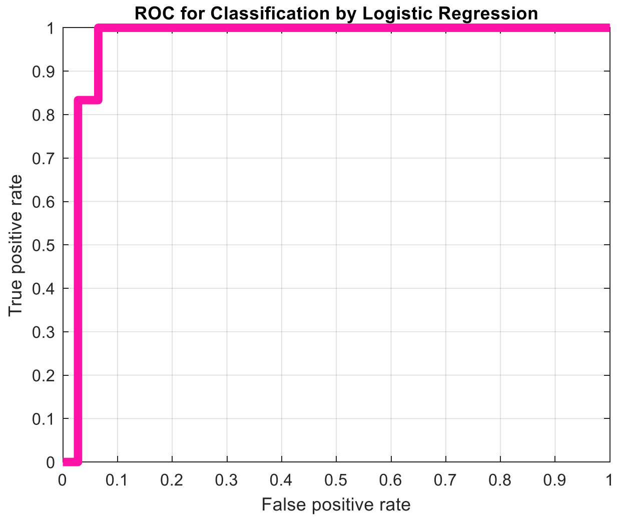


Figure 12. FPR values of ResNet101, ResNet50 and GoogleNet



(a)

(b)



(c)

Figure 13. ROC analyses of ResNet101, ResNet50 and GoogleNet

Finally, ROC analyses of ResNet101, resent50, and GoogleNet pre-training algorithms were performed. ROC tells us how well the model can distinguish between a true positive rate and a false positive rate. AUC gives the area under the ROC curve, it is between 0 and 1, if it is 0, all predictions are wrong. The true positive rate briefly shows how many of them we predicted as positive when the situation was positive and the false positive rate shows how many of them we predicted as positive when the situation was negative. That is, the more the underlying area, the better the classifier can discriminate. In Figure 13, the ROC curve is drawn by taking the average of 6 classes for three algorithms. The area under the Roc curve was found to be 0.987 in Figure 13 (a), 0.976 in Figure 13 (b), and 0.966 in Figure 13 (c). All three values are quite high. However, there are very small differences within themselves. This indicates that the classifier makes a correct discrimination.

4. DISCUSSION

The dataset with 6 classes and 100 videos in each class, collected from different accounts on Instagram, was created by us. Since the video durations on Instagram are specific, the frame numbers of the videos in the dataset are almost close to each other. While collecting data, attention was paid to the image quality and size of the videos. Moreover; It is important that there are different people performing the same action in the videos. Because what is intended to be done here is to ensure that the action taken is learned in detail by the network. For example; In mukbang videos, different people eat and the foods eaten differ from each other. These differences allow us to access more detailed and diverse information about the videos and in make-up videos, different people wear make-up and the make-up material applied to the face varies. In some videos, lipstick, eye shadow, foundation, and more than one make-up material are used together. After data was collected, 1000 features were obtained from each frame with 3 different pre-training algorithms (ResNet101, ResNet50, GoogleNet). By performing dimensionality reduction (t-SNE) analysis of these features, 600 videos belonging to 6 classes could be classified and successful results were obtained for the features obtained from all 3 pre-training algorithms. In addition, the average values of 100 videos in 6 different classes of the pixels taken from a point close to the center of the images were evaluated and it was observed that the values of each video gave quite different results. This difference has a positive effect on the distinguishability of the videos. After feature and pixel analysis, in the video classification stage, a Bi-LSTM classifier was used. The number of layers of the Bi-LSTM classifier can be increased or decreased at the desired rate, and overfitting can be prevented by making dropouts at the desired rate [27]. It was preferred due to these advantages. As a result of classification, the most successful results were obtained from the ResNet101 algorithm. Both the depth of the network and the number of layers are greater than the other two algorithms. Positive results have also been observed. It is understood from this that as the structure of the network improves, the success of the system and the received attributes increases [28]. Then, accuracy values of ResNet50 and GoogleNet were obtained. From the results obtained, it is seen that as the depth of the network and the number of layers increases, the obtained features are more accurate and more selected, as well as the accuracy and other parameters obtained. As a result, in this study, the effects of different feature extraction algorithms with the same dataset and the same classification parameters on the system outputs were investigated.

5. CONCLUSION

In this study, we worked on the video dataset we created. A dataset of 600 videos, collected from different social media accounts in 6 different categories, was used. Since the duration of the videos uploaded to Instagram is specific, the duration and number of frames of almost every video are very close to each other. This minimizes data loss when combining the feature matrices of the videos one under the other. That is, in order for the data to be sent to classification, the size of the features of the 100 videos in each class must be the same. Features were obtained in this way by using three different pre-training algorithms. The results of three different pre-training algorithms were obtained. At this point, it was requested to observe the effect of the different data obtained. The first layers have low-level features, while the last layers contain detailed information. Based on this feature, it is aimed to acquire more successful results by taking features from the last layer [29]. ResNet101 gave the most accurate test results with 97.3% accuracy. then ResNet 50 gave 96.5%; GoogleNet gave 94.2% accuracy results. Based on all these results, we need to be able to test how accurate the established model is, since the data set has been collected and classified by us. Therefore, to test the accuracy of the model, it was classified 5 times using the 6-class Anomaly-Detection-Data Set-UCF obtained from Kaggle with the same pre-training and classification algorithms. Classifying the same parameters and the same algorithms using the readily available dataset allows us to test the accuracy of the established model. When we look at the classification results, ResNet 101 is 98.3%, ResNet 50 is 97.3%, and GoogleNet is 96.2%. According to these results, it was observed that they were related to the data set we collected, the results gave high accuracy, and no overfitting or underfitting occurred. Moreover, precision and recall results were also evaluated. Recall and Precision values were calculated by making calculations with the accuracy matrices obtained as a result of the classification and their relationships with the obtained accuracy were examined. In addition, histograms of the feature matrices included in the classification, containing the information to be sent to the classifier, were created and the amount of feature series sent to the classification was kept at a certain rate to prevent overfitting.

6. ETHICAL CONSIDERATIONS

All video files used in the research were taken from public accounts. These are video files that can be accessed and shared by the whole world without any confidentiality. So it is possible to take these videos and use them as desired. The videos used are taken from public blog accounts are for general information purposes and are not confidential. Additionally, no videos from any private or individual accounts were included in the research. No one's personal posts were used in this study.

REFERENCES

[1] Ziane, A., Douaoui, A., Yahiaoui, I., Pulido, M., Larid, M., Gulakhmadov, A., Chen, X. (2022). Upgrading the salinity index estimation and mapping quality of soil salinity using artificial neural networks in the lower-cheliff plain of Algeria in north Africa: Amélioration de l'estimation de l'indice de salinité et de la qualité de la cartographie de la salinité des sols en utilisant les réseaux de neurones artificiels dans la plaine du Bas Cheliff au Nord de l'Algérie. Canadian Journal of Remote Sensing, 48(2): 182-196. https://doi.org/10.1080/07038992.2021.2010523

[2] Borovykh, A., Oosterlee, C.W., Bohté, S.M. (2019). Generalization in fully-connected neural networks for time series forecasting. Journal of Computational Science, 36: 101020. https://doi.org/10.1016/j.jocs.2019.07.007

[3] Unsvåg, E.F., Gambäck, B. (2018, October). The effects of user features on Twitter hate speech detection. In Proceedings of the 2nd Workshop on Abusive Language Online (ALW2). Brussels, Belgium. Association for Computational Linguistics. Proceedings of the Workshop, Co-Located with EMNLP, pp. 75-85. https://doi.org/10.18653/v1/w18-5110

[4] Xu, D., Shi, Y., Tsang, I.W., Ong, Y.S., Gong, C., Shen, X. (2019). Survey on multi-output learning. IEEE Transactions on Neural Networks and Learning Systems, 31(7): 2409-2429. https://doi.org/10.1109/TNNLS.2019.2945133

[5] Taye, M.M. (2023). Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions. Computation, 11(3): 52. https://doi.org/10.3390/computation11030052

[6] Nazir, S., Kaleem, M. (2023). Federated learning for medical image analysis with deep neural networks. Diagnostics, 13(9): 1532. https://doi.org/10.3390/diagnostics13091532

[7] Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S., Dehmer, M. (2020). An introductory review of deep learning for prediction models with big data. Frontiers in Artificial Intelligence, 3: 4. https://doi.org/10.3389/frai.2020.00004

[8] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9. https://doi.org/10.1109/CVPR.2015.7298594

[9] Zhang, L., Bian, Y., Jiang, P., Zhang, F. (2023). A transfer residual neural network based on ResNet-50 for detection of steel surface defects. Applied Sciences, 13(9): 5260. https://doi.org/10.3390/app13095260

[10] Constantinou, M., Exarchos, T., Vrahatis, A.G., Vlamos, P. (2023). COVID-19 classification on chest X-ray images using deep learning methods. International Journal of Environmental Research and Public Health, 20(3): 2035. https://doi.org/10.3390/ijerph20032035

[11] Devassy, B.M., George, S. (2020). Dimensionality reduction and visualisation of hyperspectral ink data using t-SNE. Forensic Science International, 311: 110194. https://doi.org/10.1016/j.forsciint.2020.110194

[12] Kimura, M. (2021). Generalized t-SNE through the lens of information geometry. IEEE Access, 9: 129619-129625. https://doi.org/10.1109/ACCESS.2021.3113397

[13] Chipman, H.A., Gu, H. (2005). Interpretable dimension reduction. Journal of Applied Statistics, 32(9): 969-987. https://doi.org/10.1080/02664760500168648

[14] Ma, S., Bi, W., Liu, X., Li, S., Qiu, Y., Huang, C., Lv, R., Yin, Q. (2022). Single-cell sequencing analysis of the db/db mouse hippocampus reveals cell-type-specific insights into the pathobiology of diabetes-associated cognitive dysfunction. Frontiers in Endocrinology, 13: 891039. https://doi.org/10.3389/fendo.2022.891039

[15] Diykh, M., Miften, F.S., Abdulla, S., Deo, R.C., Siuly, S., Green, J.H., Oudahb, A.Y. (2022). Texture analysis based graph approach for automatic detection of neonatal seizure from multi-channel EEG signals. Measurement, 190: 110731. https://doi.org/10.1016/j.measurement.2022.110731

[16] Li, K., Chen, Y., Liu, J., Mu, X. (2022). Survey of deep learning-based object detection algorithms. Jisuanji Gongcheng/Computer Engineering, 48(7). https://doi.org/10.19678/j.issn.1000-3428.0062725

[17] Coelho Ribeiro, L.A., Bresolin, T., Rosa, G.J. de M., Rume Casagrande, D., Danes, M. de A.C., Dórea, J.R.R. (2021). Disentangling data dependency using cross-validation strategies to evaluate prediction quality of cattle grazing activities using machine learning algorithms and wearable sensor data. Journal of Animal Science, 99(9). https://doi.org/10.1093/jas/skab206

[18] Devasahayam, S. (2023). Deep learning models in Python for predicting hydrogen production: A comparative study. Energy, 280: 128088. https://doi.org/10.1016/j.energy.2023.128088

[19] Moreno-Torres, J.G., Sáez, J.A., Herrera, F. (2012). Study on the impact of partition-induced dataset shift on $ k $-fold cross-validation. IEEE Transactions on Neural Networks and Learning Systems, 23(8): 1304-1312. https://doi.org/10.1109/TNNLS.2012.2199516

[20] Schuster, M., Paliwal, K.K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11): 2673-2681. https://doi.org/10.1109/78.650093

[21] Liu, G., Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing, 337: 325-338. https://doi.org/10.1016/j.neucom.2019.01.078

[22] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X. (2016). Improved techniques for training gans. Advances in Neural Information Processing Systems, 29.

[23] Simon, R.M., Subramanian, J., Li, M.C., Menezes, S. (2011). Using cross-validation to evaluate predictive accuracy of survival risk classifiers based on high-dimensional data. Briefings in Bioinformatics, 12(3): 203-214. https://doi.org/10.1093/bib/bbr001

[24] Lieberman, M.D., Cunningham, W.A. (2009). Type I and Type II error concerns in fMRI research: Re-balancing the scale. Social Cognitive and Affective Neuroscience, 4(4): 423-428. https://doi.org/10.1093/scan/nsp052

[25] Wang, Y., Lu, Z. (2023). Automatic segmentation of pneumothorax in chest radiographs based on dual-task interactive learning method. In Proceedings of the 2023 5th International Conference on Image, Video and Signal Processing, pp. 51-58. https://doi.org/10.1145/3591156.3591163

[26] Metzler, H., Baginski, H., Niederkrotenthaler, T., Garcia, D. (2022). Detecting potentially harmful and protective suicide-related content on Twitter: Machine learning approach. Journal of Medical Internet Research, 24(8): e34705. https://doi.org/10.2196/34705

[27] Garbin, C., Zhu, X., Marques, O. (2020). Dropout vs. batch normalization: An empirical study of their impact to deep learning. Multimedia Tools and Applications, 79(19): 12777-12815. https://doi.org/10.1007/s11042-019-08453-9

[28] Kaya, M., Bilge, H.Ş. (2019). Deep metric learning: A survey. Symmetry, 11(9): 1066. https://doi.org/10.3390/sym11091066

[29] Jiao, Z., Gao, X., Wang, Y., Li, J. (2016). A deep feature based framework for breast masses classification. Neurocomputing, 197: 221-231. https://doi.org/10.1016/j.neucom.2016.02.060