

## Article

# Deep Learning and Kurtosis-Controlled, Entropy-Based Framework for Human Gait Recognition Using Video Sequences

Muhammad Imran Sharif <sup>1</sup>, Muhammad Attique Khan <sup>1</sup>, Abdullah Alqahtani <sup>2</sup>, Muhammad Nazir <sup>1</sup>, Shtwai Alsubai <sup>2</sup>, Adel Binbusayyis <sup>2</sup> and Robertas Damaševičius <sup>3,\*</sup>

<sup>1</sup> Department of Computer Science, HITEC University Taxila, Taxila 47080, Pakistan; Imransharif@ciitwah.edu.pk (M.I.S.); attique@ciitwah.edu.pk (M.A.K.); muhammad.nazir@hitecuni.edu.pk (M.N.)

<sup>2</sup> College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia; Aq.alqahtani@psau.edu.sa (A.A.); Sa.alsubai@psau.edu.sa (S.A.); a.binbusayyis@psau.edu.sa (A.B.)

<sup>3</sup> Faculty of Applied Mathematics, Silesian University of Technology, 44-100 Gliwice, Poland

\* Correspondence: robertas.damasevicius@polsl.pl

**Abstract:** Gait is commonly defined as the movement pattern of the limbs over a hard substrate, and it serves as a source of identification information for various computer-vision and image-understanding techniques. A variety of parameters, such as human clothing, angle shift, walking style, occlusion, and so on, have a significant impact on gait-recognition systems, making the scene quite complex to handle. In this article, we propose a system that effectively handles problems associated with viewing angle shifts and walking styles in a real-time environment. The following steps are included in the proposed novel framework: (a) real-time video capture, (b) feature extraction using transfer learning on the ResNet101 deep model, and (c) feature selection using the proposed kurtosis-controlled entropy (KcE) approach, followed by a correlation-based feature fusion step. The most discriminant features are then classified using the most advanced machine learning classifiers. The simulation process is fed by the CASIA B dataset as well as a real-time captured dataset. On selected datasets, the accuracy is 95.26% and 96.60%, respectively. When compared to several known techniques, the results show that our proposed framework outperforms them all.

**Keywords:** gait recognition; deep learning; feature selection; classification; video understanding



check for updates

**Citation:** Sharif, M.I.; Khan, M.A.; Alqahtani, A.; Nazir, M.; Alsubai, S.; Binbusayyis, A.; Damaševičius, R. Deep Learning and Kurtosis-Controlled, Entropy-Based Framework for Human Gait Recognition Using Video Sequences. *Electronics* **2022**, *11*, 334. <https://doi.org/10.3390/electronics11030334>

Academic Editor:  
Abdeldjalil Ouahabi

Received: 28 December 2021

Accepted: 18 January 2022

Published: 21 January 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Human gait recognition (HGR) [1] is a biometric application used to solve human recognition's behavioral characteristics from a distance. A few other biometrics, such as handwriting, face [2], iris [3], ear [4], electroencephalography (EEG) [5], etc., are also useful for identifying an individual in a defined vicinity [6]. Gait recognition is critical in security systems. In this modern technological era, we require an innovative and up-to-date biometric application; thus, gait is an ideal approach for identifying individuals. The primary advantage of gait recognition over other biometric techniques is that it produces desirable results while avoiding identification from low-resolution videos [7]. Each individual has a few unique characteristics, such as walking style, speed, clothes variation, carrying conditions, and variation in angles [8].

Individual body gestures or walking styles are used to detect human gait features because each subject has a unique walking style. Each subject's walking style varies depending on the situation and the type of clothing he is wearing [9]. Additionally, when an individual holds a bag, the features are changed [10]. Gait recognition is divided into two approaches: model-based and model-free, also known as the holistic model [11]. The

model-based method requires extravagant computing costs. However, in the free-model-based technique, we can detect suspicious activity through preprocessing and segmentation techniques [12]. The major part of the process is to detect the same person in a different environment because a person's body language varies in different situations (for example, carrying a bag and moving fast) [13].

Many gait-recognition techniques are presented in the literature using machine learning (ML) [14,15]. ML is an important research area; it is utilized in several applications, such as human action recognition [16–18], image processing [19,20], and recently, COVID-19 diagnostics [21]. A simple gait-recognition method involves a few essential steps, such as preprocessing original video frames, segmenting the region of interest (ROI), extracting features from ROI, and classifying extracted features using classification algorithms [22]. Researchers use thresholding techniques to segment the ROI after enhancing the contrast of video sequences in the preprocessing step. This step is critical in traditional approaches because the features are only extracted from these segmented regions. This procedure, however, is complicated and unauthentic.

Deep learning has had a lot of success with human gait recognition in recent years. The convolutional neural network (CNN) is a type of deep learning model that is used for several processes, such as gait recognition [23], action recognition [24], medical imaging [25], and others [26,27]. A simple CNN model consists of a few important layers, such as convolutional, pooling, batch normalization, ReLu, GAP, fully connected, and classification layers [28,29]. Figure 1 depicts a simple architecture. A group of filters is adapted in the convolutional layer to extract some important features of an image, such as edges and shape. The non-linear conversion is carried out by the ReLu layer, also known as the activation layer. The batch-normalization layer minimizes the number of training epochs, whereas the pooling layers solve the overflow problem. The fully connected layer extracts the image's deep features, also known as high-level features, and classifies them in the final step using the softmax function [30,31].

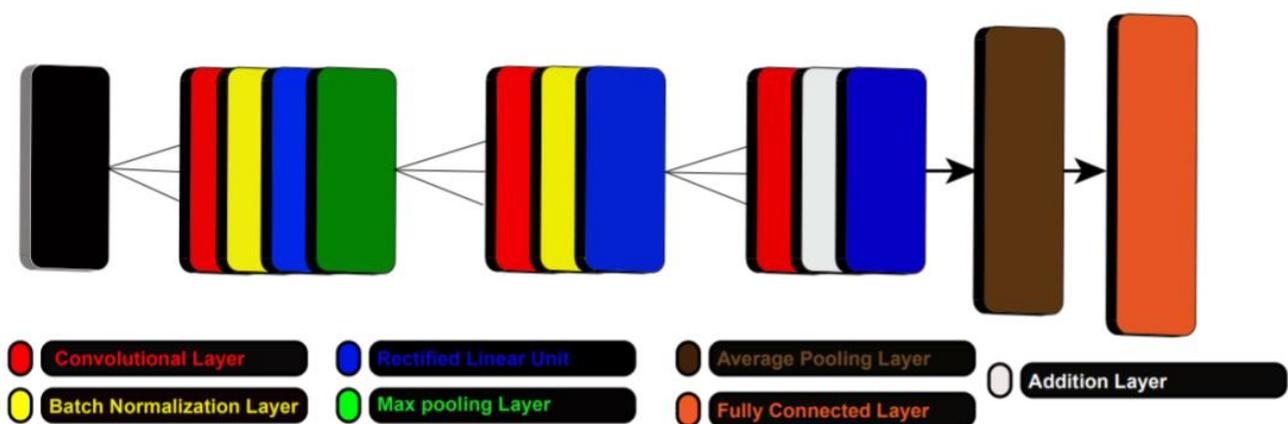


Figure 1. A simple CNN architecture for gait recognition.

Our major contributions in this work are:

- A database captured in the real-time outdoor environment using more than 50 subjects. The captured videos include a high rate of noise and background complexity.
- Refinement of the contrast of extracted video frames using the 3D box filtering approach and then fine-tuning of the ResNet101 model. The transfer-learning-based model is trained on real-time captured video frames and extracted features.
- A kurtosis-based heuristic approach is proposed to select the best features and fuse them in one vector using the correlation approach.
- Classification using multiclass one against all-SVM (OaA-SVM) and comparison of the performance of the proposed method on different feature sets.

This paper is organized as follows: previous techniques are discussed in Section 2. Section 3 describes the proposed method, such as frame refinement, deep learning, feature selection, and classification. Results of the proposed technique are discussed in Section 4. Finally, the conclusion and future directions are presented in Section 5.

## 2. Related Work

There are several methods for HGR using deep learning [32]. The authors focused on deep-learning-based methods in these methods, but the majority of them focused on selecting important features and feature fusion. The experimental process used several gait datasets, such as CASIA A, CASIA B, and CASIA C [33]. In this work, our focus is on the real-time recorded dataset and CASIA B datasets. CASIA B is a famous dataset and is mostly utilized for gait recognition.

Wang et al. [34] presented a novel gait-recognition method using a convolutional LSTM approach named Conv-LSTM. They performed a few steps to complete the gait-recognition process. At the start, they presented GEI frame by frame and then expanded its volume to relax the constraint of the gait cycle. Later on, they performed an experiment to analyze the cross-covariance of one subject. After that, they design a Conv-LSTM model for final gait recognition. The experiments were performed on the CASIA B dataset and OU-ISIR datasets. On the CASIA B dataset, they achieved an accuracy of 93%, and 95% on OU-ISIR, respectively.

Arshad et al. [6] presented a new approach for HGR. In this approach, two deep neural networks were used with the FECS selection method. Two pre-trained models, VGG19 and AlexNet, were used for feature extraction. The extracted features were refined in the later step using entropy and skewness vectors. Four datasets were used for the experimental process, CASIA A, CASIA B, CASIA C, and AVAMVG. On these datasets, they achieved the accuracy of 99.8%, 99.7%, 93.3%, and 92.2%, respectively.

Mehmood et al. [22] presented an automated deep-learning-based system for HGR under multiple angles. Four key steps were performed, preprocessing, feature extraction, feature selection, and classification. They extracted deep-learning features and applied the firefly algorithm for feature optimization. The experiments were performed on a widely available CASIA B dataset and achieved notable accuracy.

Anusha et al. [35] presented a technique for HGR based on multiple features. They extracted low-level features through spatial, texture, and gradient information. Five databases were used for the experimental process, CASIA A, CASIA B, OU-ISIR D, CMU MoBo, and KTH video datasets. These all datasets were tested on different angles and achieved improved performance.

Sharif et al. [14] presented a method for HGR based on accurate ROI segmentation and multilevel features fusion. Several clothing and carrying conditions were considered for the experimental process and achieved an accuracy of 98.6%, 93.5%, and 97.3%, respectively. A PoseGait model was presented by Liao et al. [13] for HGR. They considered the problem of drastic variations of individuals. The 3D models were used for capturing data from different angles. The 3D image was defined as 3D coordinates of the human body joints.

Wu et al. [36] presented a graph-based approach for multiview HGR. The Spiderweb-graph-based approach was applied to capture the data in a single view and then connect to the other view of gait data concurrently. Memory and capsule modules were used for the trajectory view of each gait as well as STF extraction. The experiments were performed on three challenging gait datasets, SDUgait, OU-MVLP, and CASIA B, and achieved an accuracy of 98.54%, 96.91%, and 98.77%, respectively. Arshad et al. [37] focused on the feature-selection approach to improve the performance of HGR. They used the Bayesian model and quartile-deviation approaches for enhanced feature vectors.

In conclusion, the previous studies focused primarily on the selection of the most relevant features. They used CNN to extract features and then combined the results with data from a few other channels. They did not, however, focus on real-time gait recognition

due to the complexity of the system design. In this paper, we focus on real-time gait recognition in an outdoor setting.

### 3. Proposed Methodology

The proposed method for real-time human gait recognition is presented in this section with detailed mathematical modeling and visual results. The following steps are involved in the proposed framework: video preprocessing, deep learning features extraction using transfer learning, kurtosis-based features selection, correlation-based features fusion, and finally, one-against-all-SVM (OaA-SVM)-based classification. The main architecture diagram is shown in Figure 2. The proposed method is executed in a sequence, and at the end of the execution, it returns a labeled output and numerical results in the form of recall rate, precision, accuracy, etc. The details of each step are given below in the following subsections.

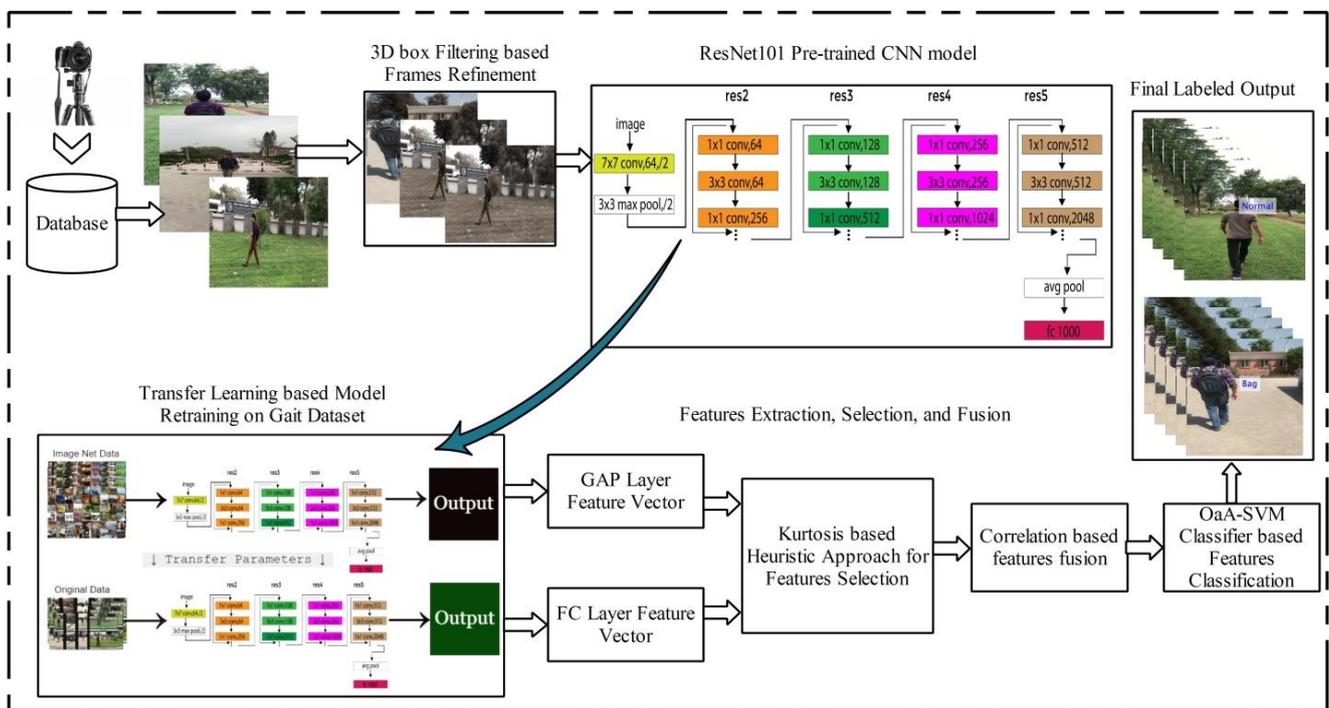


Figure 2. Proposed architecture of real-time human gait recognition using deep learning.

#### 3.1. Videos Preprocessing

One of the most prominent parts of digital image processing is to enhance the quality of visual information of an image. This technique helps to remove the noise and messy information from the image and make it more readable [38]. In the area of machine learning, the algorithm requires better information about an object to learn a good model. In this work, we initially process with videos, and in a later stage, convert them into frames and label them according to the actual class. Then, we resize images into a dimension of  $R \times C \times ch$ , where  $R$  denotes row pixels,  $C$  denotes column pixels, and  $ch$  represent number of channels, respectively. We set  $R = C = 224$  and  $ch = 3$ .

The real-time videos are recorded into four different angles, and each angle consists of two gait classes—walk while carrying a bag and normal walk without any bag. The four angles are  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ , and  $360^\circ$ . The original frames also captured some noise during the video recording. For this purpose, we used the 3D box filter, which is a perfect choice. The detail of 3D box filtering is given in Algorithm 1.

---

**Algorithm 1:** Data Augmentation Process.

---

**Input:** Original video frame  $\leftarrow \phi(r, c)$ .

**Output:** Improved video frame  $\leftarrow \tilde{\phi}(r, c)$ .

**Step 1:** Load all video frames  $\leftarrow \Delta(\phi)$ .

**for:** 1 : to  $\leftarrow Nframes$

**Step 2:** Calculate filter size.

- $\tilde{F}_{abc\tilde{h}} = \sum_{r=1}^4 \sum_{c=1}^4 \sum_{ch=1}^3 (F_{rc} \tilde{c}h)$

**Step 3:** Perform padding.

- $Padding = \frac{\tilde{F}_{abc\tilde{h}} - 1}{2}$

- where horizontal padding is performed.

**Step 4:** Update padding window.

- $Upd(Padding) = \frac{\sum(Padding)}{t}$

- where  $t = 2$

**Step 5:** Perform normalization.

- $Normalization = \frac{1}{\prod(Upd(Padding))}$

- $\prod \leftarrow$  product of padding image.

**end for**

---

The visual effects are also shown in the main flow, Figure 2. Here, the frames are given before and after the filter processing. The improved frames are utilized in the next step for model learning.

### 3.2. Convolutional Neural Network

Deep learning emerged recently in the field of computer vision and has since spread to other fields. Convolutional Neural Network (CNN) is a deep learning approach that won a competition for image classification using ImageNet in 2012 [39]. In deep learning, images are directly passed in the model without segmentation; therefore, it is called image-based ML. A simple CNN consists of convolutional layer, ReLu layer, dense layer, pooling layer, and a softmax layer. Visually, simple architecture is shown in Figure 1. In a CNN, the input image is passed to the network in a fixed size. A convolutional operation is performed in the convolutional layer and in the output weights, and bias vectors are generated. Back-propagation is used in the CNN for training the weights. Mathematically, convolutional operation is formulated as follows:

$$\hat{Y}_C = \sum_{k=1}^C \psi_{k'k} * x_k, \forall C', \tag{1}$$

where  $\hat{Y}_C$  is convolutional layer output,  $C$  denotes the convolutional kernels,  $x$  is composed of channels  $ch$ ,  $*$  denotes the convolutional operation,  $\psi$  represents convolutional kernel and is defined by  $\psi \in \mathbb{R}^{k \times k \times C}$ . To increase the nonlinearity of a network, ReLu layer is applied and defined as follows:

$$\hat{Y}_{\tilde{r}} = \max(0, x), \tag{2}$$

Another layer named maxpooling is employed to reduce the dimension of extracted features. This layer is formulated as follows:

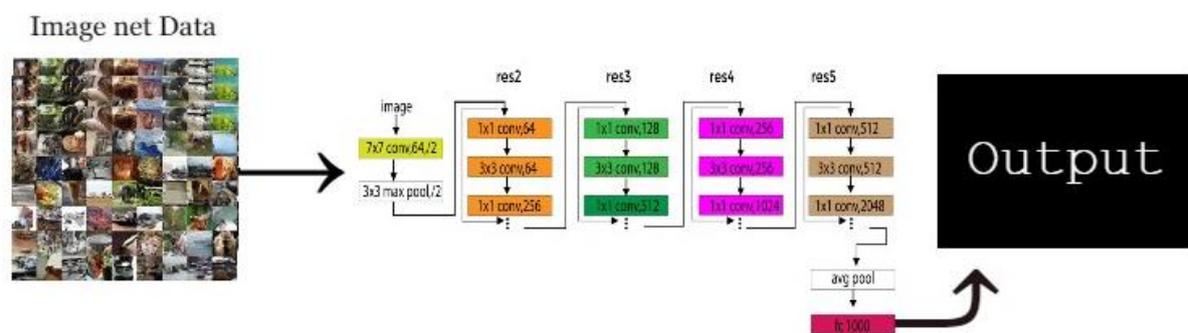
$$\hat{Y}_{max}(i, j, c) = \max_{u,v \in N} x + u, j + v, c, \forall i, j, c, \tag{3}$$

where  $\hat{Y}_{max}$  denotes the max pooling operation,  $x$  denotes the input weight matrix, and  $x \in \hat{Y}_C$ ,  $u$ ,  $j$ , and  $v$  are static parameters. The filter of this layer is normally fixed as  $3 \times 3$ , denoted by  $u$ , and stride is 2, denoted by  $v$ . An important layer in a CNN is fully connected

layer. In this layer, all neurons are connected to each other, and resultant output is obtained in a 1D matrix. Finally, the extracted features are classified in the softmax layer.

### 3.3. Deep Features Extraction

Feature extraction is the process of reducing original images into small number of important information groups. These groups are called features. The features have few important characteristics such as color, shape, texture, and location of an object. In this proposed technique, we utilized a pre-trained deep learning network named ResNet101. Originally, this model consists of total of 101 layers and was trained on ImageNet dataset. A visual flow is shown in Figure 3. Note that the filter size of first convolutional layer is  $7 \times 7$ , stride is 2, and number of channels is 64. In the next, a maxpooling layer is applied for filter size  $3 \times 3$ , and stride is 2. Five residual layers are included in this network, and lastly, global average pool layer and FC layer are added following a softmax layer.



**Figure 3.** Original ResNet101 pre-trained CNN model for image classification.

**Training Model using Transfer Learning:** In this work, we use ResNet101 pre-trained CNN model for features extraction [40]. We trained this model using transfer learning, as visually shown in Figure 4 [41]. Note that parameters of original ResNet101 pre-trained model are transferred and trained on the real-time captured gait database. We remove the last layer from modified ResNet101 and extract features. The features are extracted from global average pool (GAP) and FC layers. Mathematically, TL can define as follows: given a source domain  $S_D$  and learning task  $L_t$ , a target domain  $T_D$  and learning task  $L_{\bar{t}}$ , TL improves the learning of a target predictive function  $F_t(\cdot)$  in the  $T_D$  using the knowledge in  $S_D$  and  $L_t$ , where  $S_D \neq T_D$  or  $L_t \neq L_{\bar{t}}$ . After this process, we obtain two feature vectors from GAP and FC layers of dimensions  $N \times 2048$  and  $N \times 1000$ , respectively.

### 3.4. Kurtosis-Controlled, Entropy-Based Feature Selection

Feature selection (FS) is the process of selecting the most prominent features from the original feature vector. In the FS, the value of features is not updated, and features are selected in the original form [42]. The main motivation behind this step is to obtain the most optimal information of an image and discard the redundant and irrelevant data. In this work, we proposed a new feature selection technique named kurtosis-controlled OaA-SVM. The working of this technique is given in Algorithms 2 and 3 for both vectors.

**Description:** Consider that we have two extracted deep feature vectors denoted by  $V_1$  and  $V_2$ , where the dimension of each vector is  $N \times 2048$  and  $N \times 1000$ , respectively. In the first step, features are initialized and processed in the system until all features are passed. In the next step, kurtosis is computed for all features in the pair, where window size and stride were  $2 \times 2$  and 1. A newly generated vector is obtained, which is evaluated in the next step by fitness function. In the fitness function, we select one vs. all SVM classifier that classifies the feature and, in the output, returns an error rate.

This process continues until the error rate is minimized and stops when error increases for the next iteration. By following this, we obtain two feature vectors in the output of dimension  $N \times 600$ , as discussed in Algorithms 1 and 2.

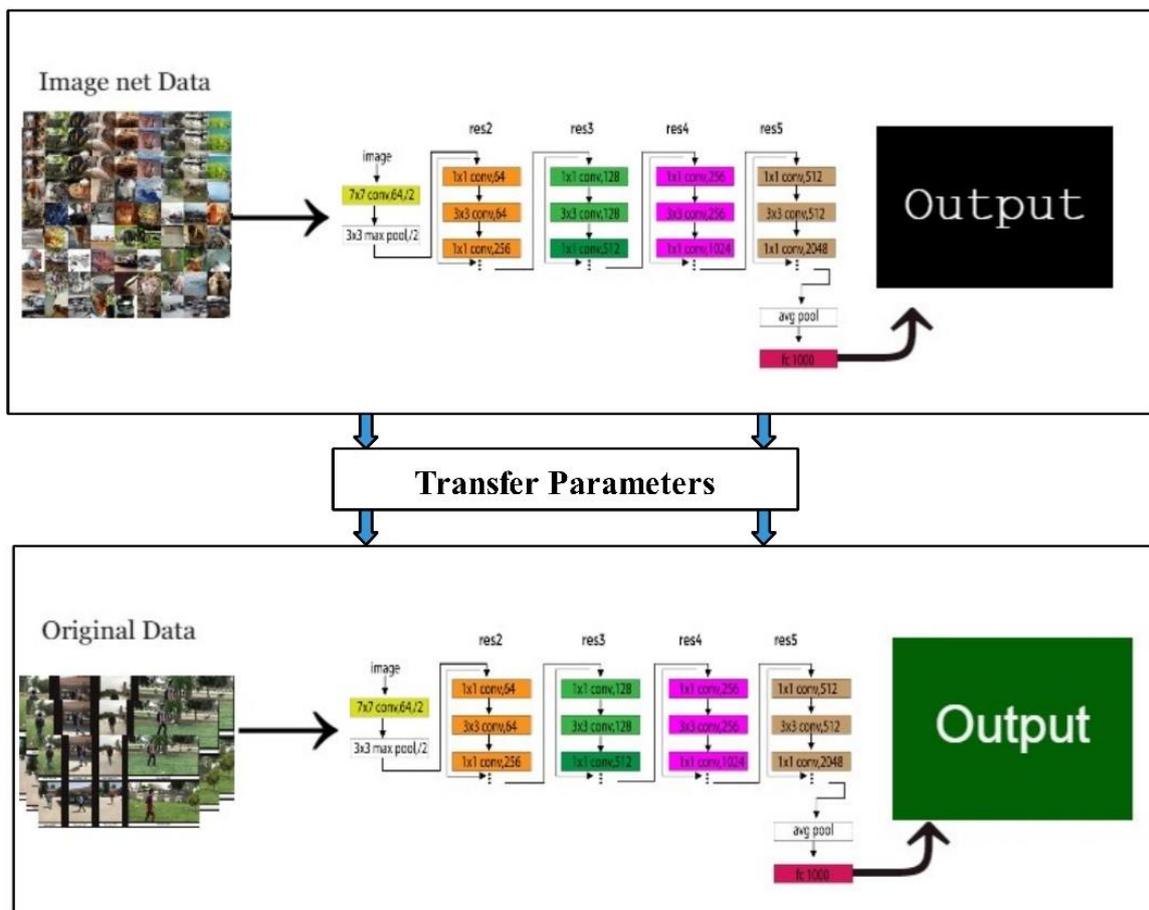


Figure 4. Transfer-learning-based model retraining on real-time captured video frames for HGR.

Finally, we fused both selected vectors by employing correlation-based approach. In this approach, features of both vectors are paired as  $i(1)$  and  $j(1)$ . After that, the correlation coefficient is computed, and higher-correlated feature pairs are added to the fused vector. Mathematically, it is defined as follows:

$$r_{v1v2} = \frac{\sum(v1_i - \bar{v1})(v2_j - \bar{v2})}{\sqrt{(\sum(v1_i - \bar{v1})^2 \sum(v2_j - \bar{v2})^2)}} \tag{4}$$

where  $v1 \in V_1, v2 \in V_2, r_{v1v2}$  is a correlation coefficient among two features  $v1_i$  and  $v2_j$ ,  $v1_i$  is  $i$ th features of  $V_1$ , and  $v2_j$  is  $j$ th feature of  $V_2$ , respectively. The notation  $\bar{v1}$  is mean value of feature vector  $V_1$  and  $\bar{v2}$  is mean value of feature vector  $V_2$ .

In this approach, we choose those features for fused vector which have correlation value near to 1 or greater than 0. This means we only selected positively correlated features for final fused vector. This procedure is performed for all features in both vectors and in the output; we obtained a resultant features vector of length  $N \times f_k$ , where  $f_k$  denotes the length of fused feature vector.

Finally, the features in the fused vector are classified using OaA-SVM.

**Algorithm 2:** Features selection for deep learning model 1.

**Input:** Feature vector  $V_1$  of dimension  $N \times 2048$ .

**Output:** Selected feature  $V_s$  of dimension  $N \times 600$ .

**Step 1:** Features initialization.

for  $i \leftarrow 1$  to  $N$  //  $N = 2048$

**Step 2:** Compute kurtosis of each feature pair.

$$- \text{Kurtosis} = \frac{\frac{\sum_{i=1}^N (V_i - \bar{V})^4}{N}}{\left[ \frac{\sum_{i=1}^N (V_i - \bar{V})^2}{N} \right]^2} \quad // V_i \text{ is input features, } \bar{V} \text{ is mean feature value, and } N \text{ denotes}$$

the total features.

- Window size is  $2 \times 2$  //  $\sigma$  is a standard deviation.

- Stride is 1.

**Step 3:** Newly generated kurtosis vector  $\tilde{V}$  //  $\tilde{V}$  is kurtosis vector.

**Step 4:** Perform fitness function.

- OaA-SVM classifier.

- Evaluate features.

- Compute error rate.

**Step 5:** Repeat steps 2, 3, and 4 until error rate is minimized.

end for

**Algorithm 3:** Features selection for deep learning model 2.

**Input:** Feature vector  $V_2$  of dimension  $N \times 1000$ .

**Output:** Selected feature  $V_s$  of dimension  $N \times 600$ .

**Step 1:** Features initialization.

for  $i \leftarrow 1$  to  $N$  //  $N = 1000$

**Step 2:** Compute kurtosis of each feature pair.

$$- \text{Kurtosis} = \frac{\frac{\sum_{i=1}^N (V_i - \bar{V})^4}{N}}{\left[ \frac{\sum_{i=1}^N (V_i - \bar{V})^2}{N} \right]^2} \quad // V_i \text{ is input features, } \bar{V} \text{ is mean feature value, and } N \text{ denotes}$$

the total features.

- Window size is  $2 \times 2$ .

- Stride is 1.

**Step 3:** Newly generated kurtosis vector  $\tilde{V}$  //  $\tilde{V}$  = random vector size.

**Step 4:** Perform fitness function.

- OaA-SVM classifier.

- Evaluate features.

- Compute error rate.

**Step 5:** Repeat steps 2, 3, and 4 until error rate is minimized.

end for

3.5. Recognition

The one against all SVM (OaA-SVM) is utilized in this work for features classification [43]. The OaA-SVM approach is usually to determine the separation of one gait action with other listed gait action classes. Mostly, this classification is used for unknown patterns to generate the maximum results.

Here, we have N-class problem, and D represents the training samples:  $\{f_{k1}, l_1\}, \dots, \{f_{kD}, l_D\}$ . Here,  $f_{ki} \in R^b$  represents b-dimensional feature vector, where  $l_i \in \{1, 2, \dots, B\}$  is the corresponding class label. This approach usually constructs the binary SVM classifiers. The  $i$ th represent all the positive labels of training class and the other negative labels. Then,  $i$ th SVM solves the problems with the following yield-decision function:

$$f_i(s) = h_i^T \Phi(f_{ki}) + \beta_i, \tag{5}$$

Here,  $f_i(s)$  is the yield-decision function,  $h_i^T$  represents weight values,  $f_{k1}$  denotes input features, and  $\beta_i$  denotes the bias values. The minimization function is applied on weight values as:

$$\text{Minimize : } M(h, \theta_j^i) = \frac{1}{2} \|h_i\|^2 + C \sum_{i=1}^N \theta_j^i, \tag{6}$$

$$\text{Subject to : } l_j(h_i^T \Phi)(f_{ki}) + \beta_i \geq 1 - \theta_j^i, \theta_j^i \geq 0, \tag{7}$$

where  $M$  is a minimization parameter,  $C$  denotes the output class,  $\theta_j^i$  is between-class distance,  $l_j = 1$  if  $l_j = i$ , and  $l_j = -1$  otherwise. A sample  $l$  is classified as in class  $i^*$ .

$$i^* = \arg \max_{i=1, \dots, B} f_{ki} = \arg \max_{i=1, \dots, B} (h^T \Phi)(f_{ki}) + \beta_i, \tag{8}$$

Resultant visual results are also shown in Figure 5, in which the classifier returned a labeled output such as walking with bag and normal walk. The numerical results are also computed and described in Section 4.



Figure 5. Proposed system predicted labeling results.

## 4. Results

### 4.1. Datasets

The results of the proposed automated system for gait recognition are presented in this section. The proposed framework is tested on a real-time captured dataset and CASIA B dataset. In the real-time dataset, a total of 50 students are included, and for each student, we recorded 8 videos in 4 angles, 90°, 180°, 270°, and 360°. For each angle, two videos are recorded—one while wearing a bag and one without wearing a bag. A few samples are shown in Figure 6. For the CASIA B dataset [44], we only consider a 90° angle for evaluation results. In this dataset, three different covariant factors are included, normal walking, walking with a bag, and walking while wearing a coat, as shown in Figure 7.



Figure 6. Sample image frames of real-time video sequences.

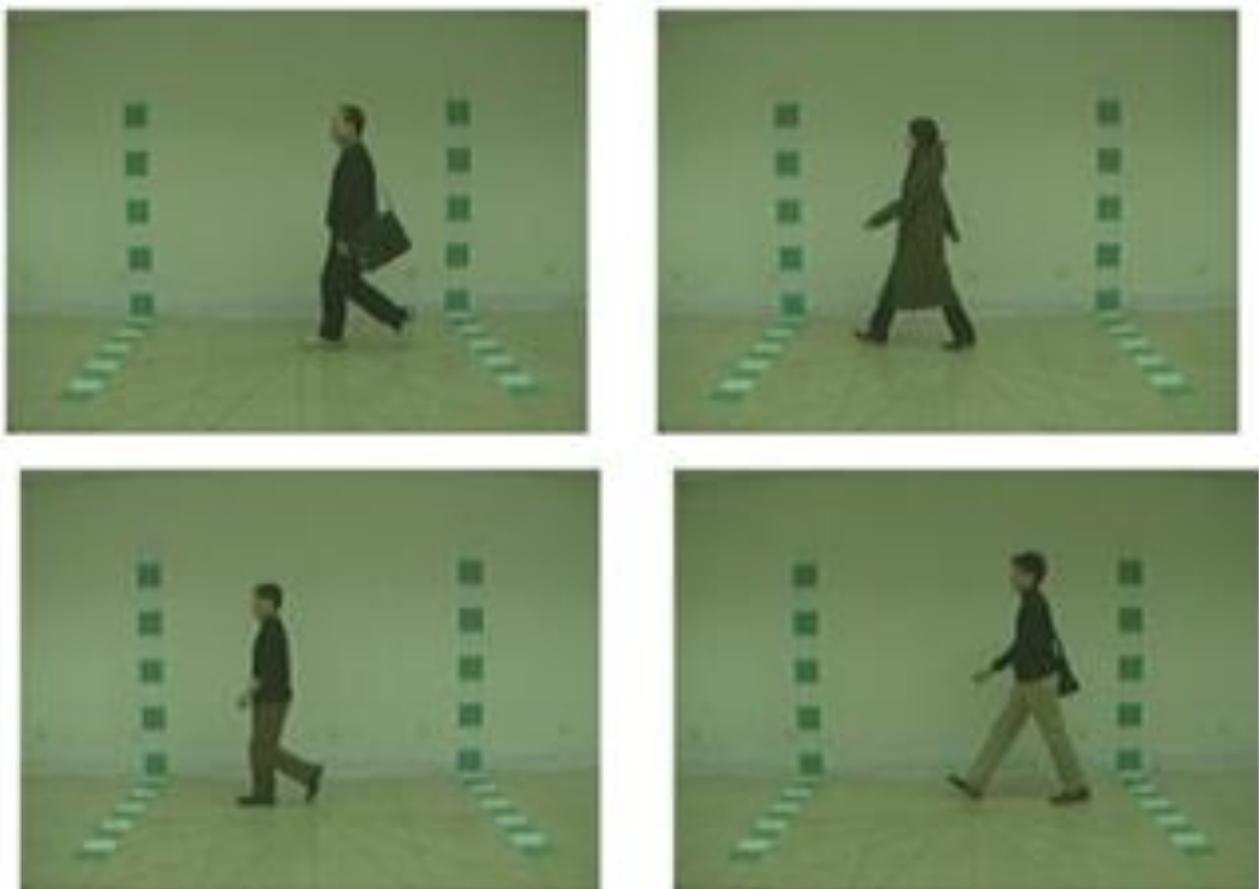


Figure 7. Sample frames collected from CASIA B dataset [44].

#### 4.2. Experimental Setup

We utilized 70% video frames for training the proposed framework, and the remaining 30% were utilized for the testing. In the training process, a group of hyper parameters was employed, where the learning rate was 0.001, epochs was 200, mini batch size was 64,

the optimization algorithm was Adam, the activation function was sigmoid, the dropout factor was 0.5, momentum was 0.7, the loss function was cross entropy, and the learning rate schedule was Piecewise. Several classifiers were utilized for validation, such as SVM, K-Nearest Neighbor, decision trees, and Naïve Bayes. The following parameters were used for the analysis of selected classifiers: precision, recall, F1 Score, AUC, and classification time. All results were computed using K-Fold validation, whereas the value of K was 10. The simulations of this work were conducted in MATLAB2020a using a dedicated personal desktop computer with 256 GB SSD, 16 GB RAM, and 16 GB memory graphics card.

#### 4.3. Real-Time Dataset Results

In this section, we present our proposed method's results on real-time video sequences. The results are computed in three different scenarios—normal walking results for all selected angles, walking while carrying a bag for all three angles, and finally, classification of normal walking and walking while carrying a bag. Table 1 shows the results of a normal walk for the four selected angles of 90°, 180°, 270°, and 360°, respectively. The OaA-SVM classifier achieved superior performance compared to other listed classification algorithms. The achieved accuracy by OaA-SVM is 95.75%, whereas the precision rate is 95.25%, F1-Score is 95.98%, AUC is 1.00, and FPR is 0.0125, respectively. The Medium KNN also gives better performance and achieves an accuracy of 95%. The recall and precision rates are 95% and 95.50%, respectively. The worst accuracy noted for this experiment is 48.75% on the Kernel Naïve Bayes classifier. The change in accuracy among all classifiers shows the authenticity of this work. The performance of OaA-SVM can be further verified in Figure 8. The prediction accuracy for the 270° angle is maximum, whereas the correct prediction performance for 360° is 86%. In the latter, the recognition time is also tabulated in Table 1 and shows that the Fine Tree classifier executed much faster compared to the other listed classifiers. The execution time of Fine Tree is 19.465 (s). The variation in execution time is also shown in Figure 9.

**Table 1.** Proposed gait-recognition results using real-time captured dataset for normal walk on selected angles.

Classifiers	Performance Measures						
	Recall (%)	Precision (%)	F1 Score (%)	AUC	FPR	Accuracy (%)	Time (s)
OaA-SVM	95.75	96.25	95.98	1.00	0.0125	96.0	204.050
Cubic KNN	94.25	94.75	94.48	1.00	0.0175	94.5	366.480
Medium KNN	95.00	95.50	95.24	1.00	0.0175	94.9	175.870
Bagged Trees	93.50	93.75	93.62	0.99	0.0200	93.5	345.750
CG-SVM	90.25	90.50	90.36	0.99	0.0325	90.1	189.300
Fine Tree	86.75	86.50	86.62	0.92	0.0425	86.7	19.565
Medium Tree	85.50	85.75	85.62	0.92	0.0475	85.4	39.907
Naïve Bayes	84.50	87.25	85.84	0.90	0.0500	84.3	34.035
Coarse KNN	61.50	72.50	66.54	0.92	0.1275	61.5	189.760
FG-SVM	61.00	84.75	70.92	0.91	0.1300	60.9	132.050
Kernel Bayes	48.75	63.50	50.54	0.73	0.1725	48.5	582.710

Table 2 shows results of walking while carrying a bag at 90°, 180°, 270°, and 360° angles, respectively. The OaA-SVM classifier attained better accuracy compared to other classification algorithms. For OaA-SVM, the accuracy is 96.5%, whereas the precision rate is 97%, F1-Score is 96.7%, AUC is 1.00, and FPR is 0.01, respectively. The KNN series classifiers, such as Cubic KNN and Medium KNN, give the second-best accuracy of 95.1%. Naïve Bayes classifier achieves an accuracy of 83.7%. The recall and precision rates of the Naïve Bayes classifier are 83.7% and 86.7%, respectively. The Kernel Naïve Bayes, FG-SVM, and Coarse KNN classifiers do not perform well. They achieve accuracies of 48.6%, 61.3%, and 63%, respectively.

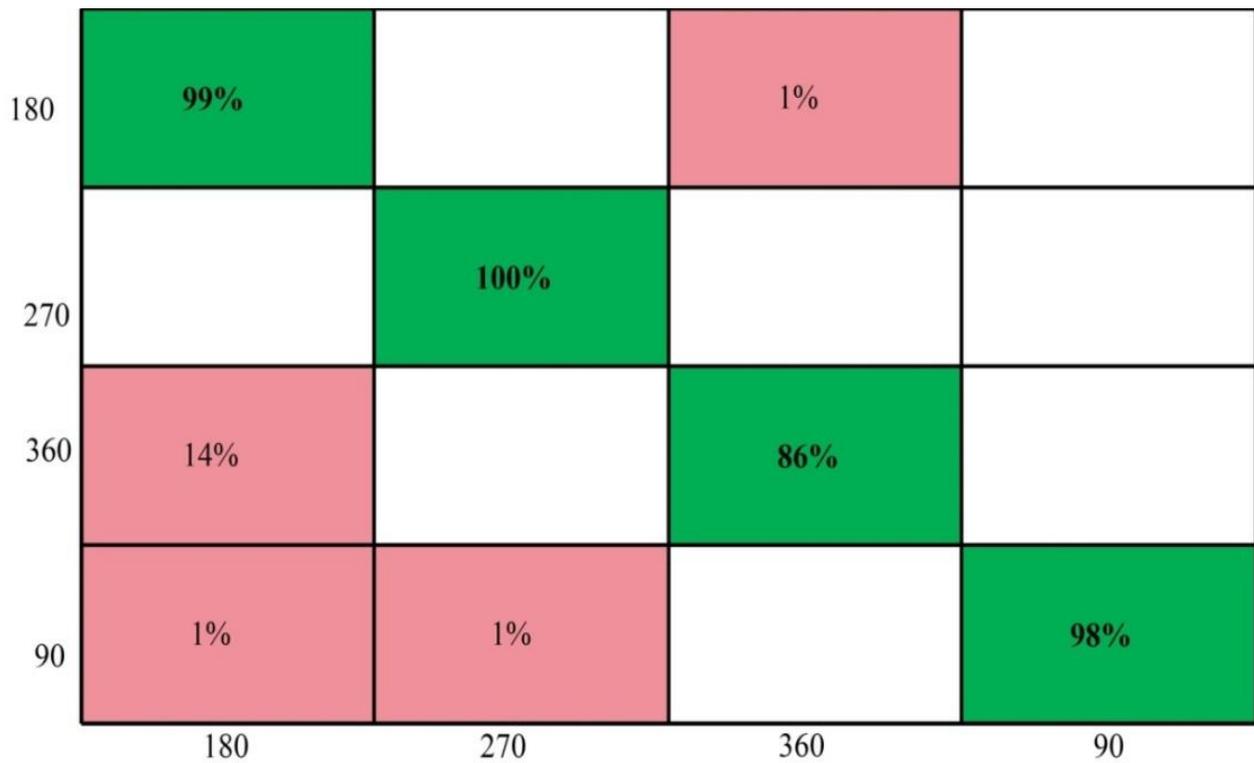


Figure 8. Confusion matrix of OaA-SVM for normal walk on different selected angles.

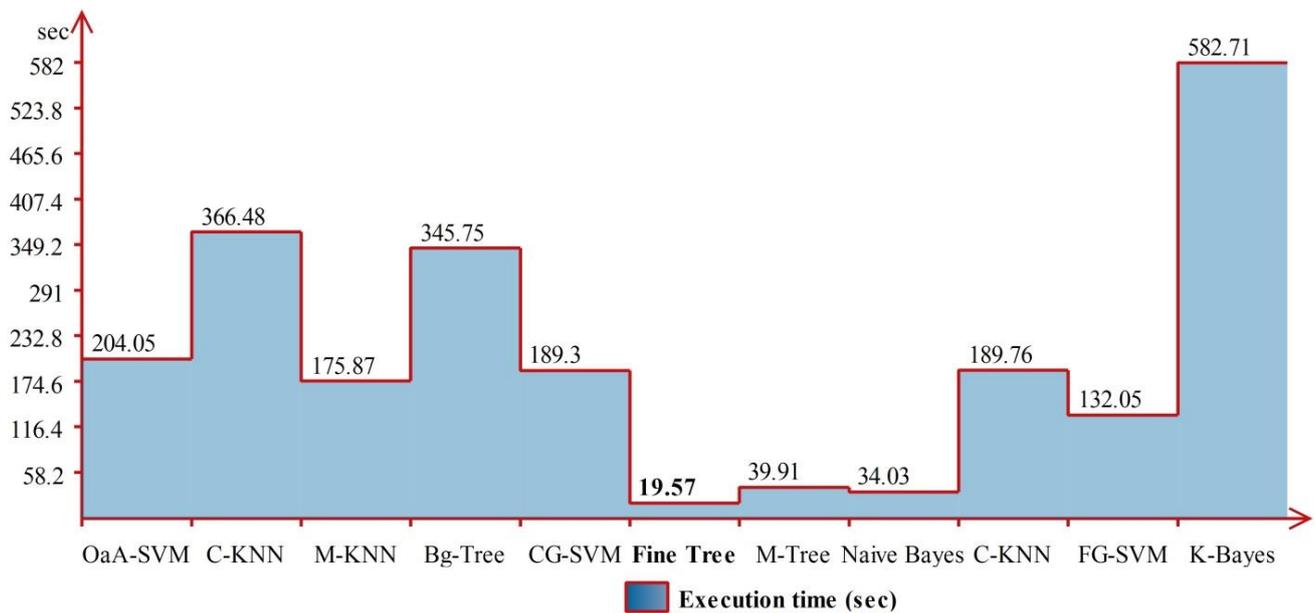
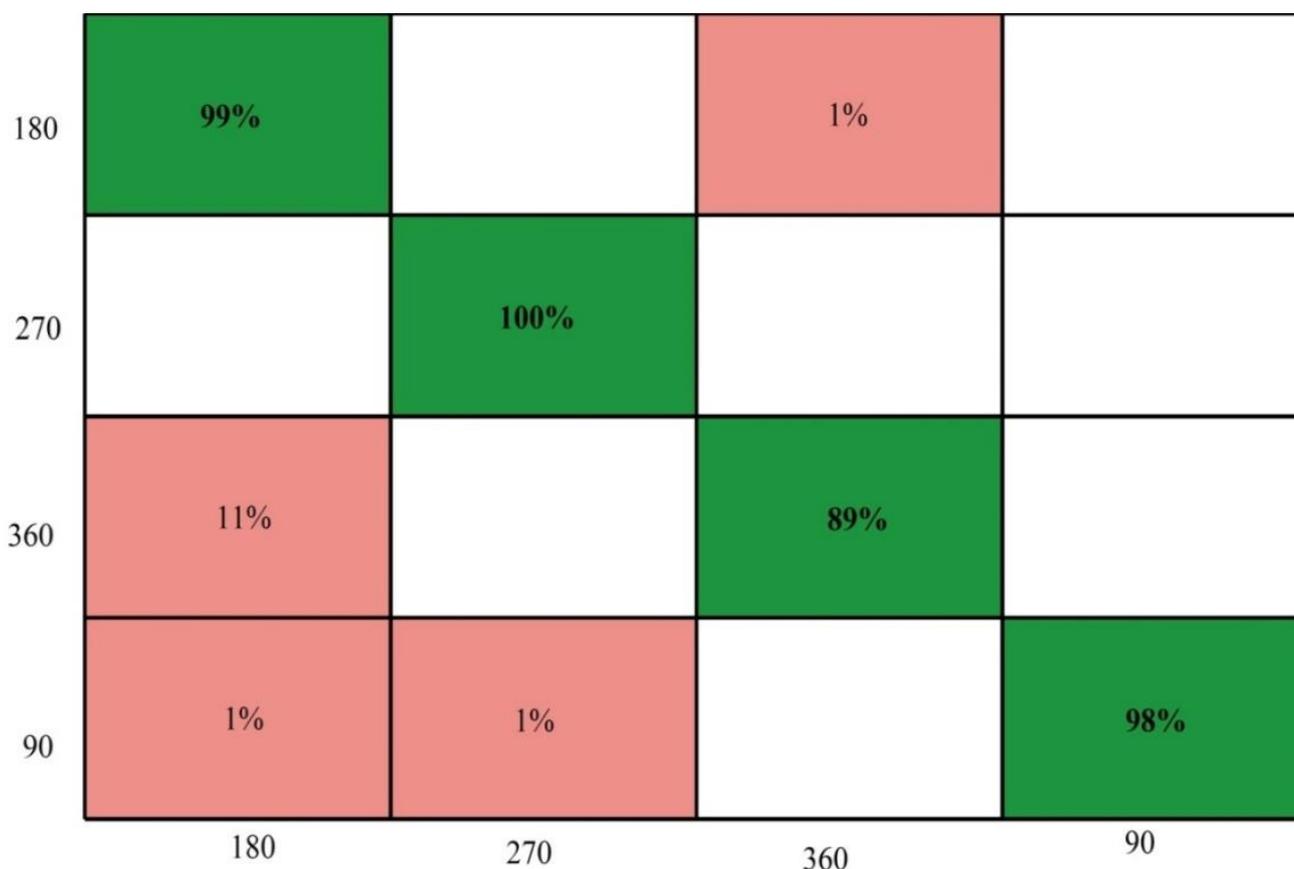


Figure 9. Variation in computational time for selected classifiers on Real time dataset.

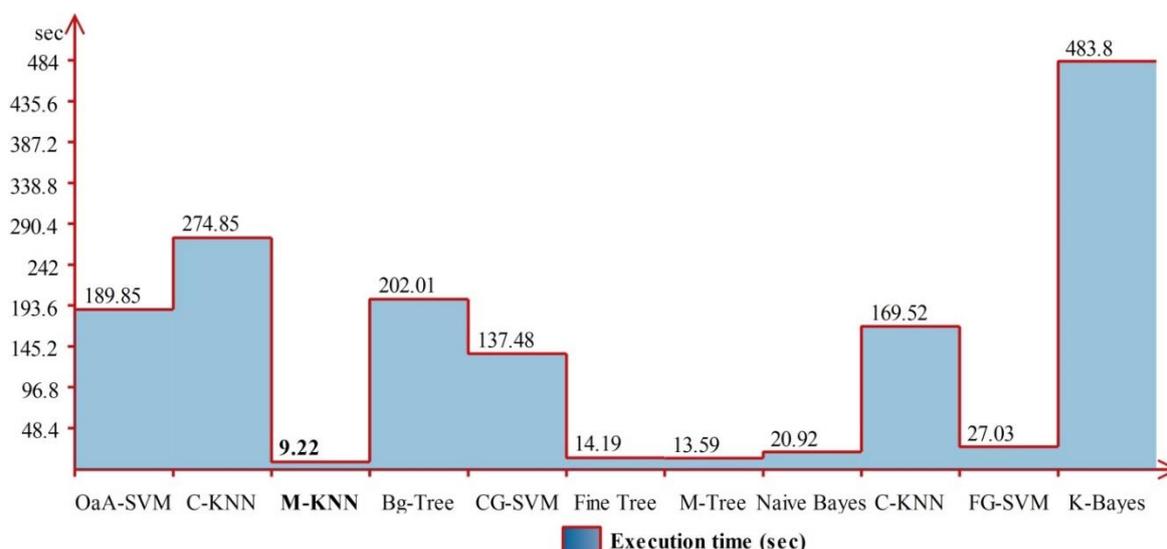
**Table 2.** Proposed gait-recognition results using real-time captured dataset for walking while carrying a bag at selected angles.

Classifiers	Performance Measures						
	Recall (%)	Precision (%)	FI Score (%)	AUC	FPR	Accuracy (%)	Time (s)
OaA-SVM	96.5	97.0	96.7	1.00	0.01	96.6	189.850
Cubic KNN	95.0	95.2	95.1	1.00	0.01	95.1	274.850
Medium KNN	95.0	95.2	95.1	1.00	0.01	95.1	9.219
Bagged Trees	94.7	95.0	94.8	0.99	0.01	94.9	202.010
CG-SVM	90.5	91.0	90.7	0.99	0.03	90.5	137.480
Fine Tree	87.7	81.0	84.2	0.93	0.04	87.6	14.190
Medium Tree	86.7	87.0	86.8	0.93	0.04	86.8	13.590
Naïve Bayes	83.7	86.7	85.2	0.89	0.05	83.7	20.920
Coarse KNN	63.0	74.0	68.0	0.92	0.12	63.0	169.521
FG-SVM	61.0	84.7	70.9	0.91	0.13	61.3	27.030
Kernel Bayes	48.7	60.2	53.8	0.75	0.17	48.6	483.800

The performance of OaA-SVM can be further verified through Figure 10, which shows that the prediction accuracy for the 270° angle is the maximum (100%), whereas correct prediction performance for 360° is 89%. The execution time during the recognition process is also tabulated in Table 2 and shows that the Medium Tree classifier executed much faster compared to other listed classifiers. The execution time is also plotted in Figure 11, which shows a huge variation due to the classifier’s complexity.



**Figure 10.** Confusion matrix of OaA-SVM for walking while carrying a bag at different selected angles.



**Figure 11.** Variation in computational time for selected classifiers of 4 selected angles on Real time dataset.

Table 3 represents the classification results of a normal walk and walking while carrying a bag. The purpose of this experiment is to analyze the performance of the proposed algorithm for the binary class classification problem. Table 3 shows that the OaA-SVM attained an accuracy of 96.4%. The other calculated measures, such as recall rate, precision rate, f1-Score, AUC, and FPR values, are 96.5%, 96.5%, 96.5%, 1.00, and 0.03, respectively. Medium KNN gives the second-best classification performance of 96.1%, whereas the recall and precision rates are 96% and 96%, respectively. The following classifiers, Naïve Bayes, Coarse KNN, Kernel Naïve Bayes, and FG SVM, do not perform well and achieve an accuracy of 67.8%, 79.3%, 73.5, and 79.6%, respectively. The confusion matrix also provided and illustrated in Figure 12 shows that the correct prediction rate of both classes is above 90%. In the latter, the execution time is also plotted for this experiment and illustrated in Figure 13, which shows that the Fine Tree execution time is better compared to other algorithms.

**Table 3.** Recognition results for all datasets with kurtosis function on 600 predictions on original deep-feature vector extracted from ResNet 101.

Classifiers	Performance Measures						
	Recall (%)	Precision (%)	F1 Score (%)	AUC	FPR	Accuracy (%)	Time (s)
OaA-SVM	96.5	96.5	96.5	1.00	0.03	96.4	129.830
Cubic KNN	95.0	96.0	95.7	1.00	0.04	95.6	278.980
Medium KNN	96.0	96.0	96.0	1.00	0.04	96.1	106.850
Bagged Trees	90.5	90.5	90.5	0.97	0.09	90.3	209.610
CG-SVM	82.0	82.5	82.2	0.91	0.18	82.2	102.260
Fine Tree	80.0	80.0	80.0	0.81	0.20	79.8	11.954
Medium Tree	76.0	76.5	76.2	0.80	0.24	76.3	30.888
Naïve Bayes	68.0	69.0	68.4	0.78	0.32	67.8	18.767
Coarse KNN	79.0	79.5	79.2	0.88	0.21	79.3	117.990
FG-SVM	79.5	85.5	82.3	0.98	0.41	79.6	74.026
Kernel Bayes	73.0	74.5	73.7	0.84	0.27	73.0	254.580

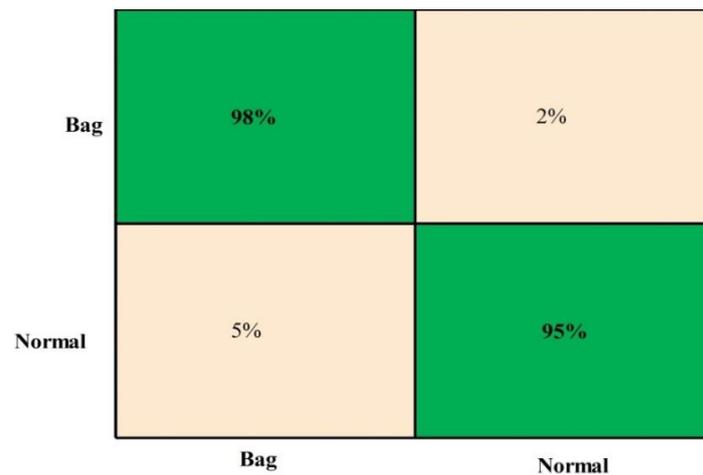


Figure 12. Confusion matrix of OaA-SVM for normal walking and walking while carrying a bag classification.

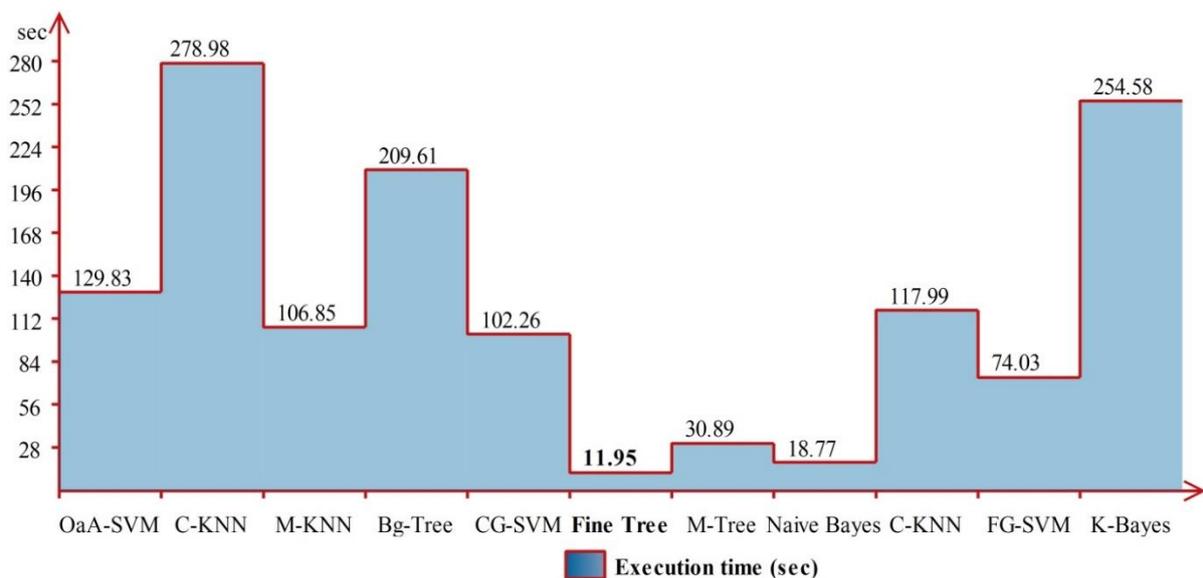


Figure 13. Execution time of proposed method on real-time dataset (normal walking and walking while carrying a bag).

4.4. CASIA B Dataset Results at a 90° Angle

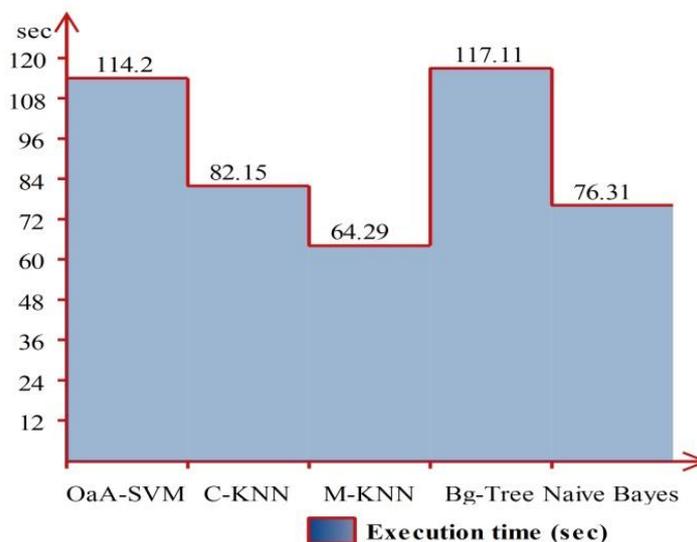
CASIA B dataset results are presented in this subsection. For experimental results, we only selected a 90° angle because most of the recent studies used this angle. Numerical results are given in Table 4. The OaA-SVM gives better results among all other implemented classifiers. The results are computed with different feature vectors, such as original global average pool (GAP) layer features, fully connected layer features (FC), and the proposed approach. For GAP-layer features, the achieved accuracy is 90.22%, and the recall rate is 90.10%, whereas the FC layer gives an accuracy of 88.64%. The proposed method gives an accuracy of 95.26%, where the execution time is 176.4450 (s). Using Cubic KNN, the attained accuracy is 93.60%, which is the second-best performance after the OaA-SVM classifier. Table 5 presents the confusion matrix of the OaA-SVM classifier performance using the proposed scheme. In this table, it is illustrated that the corrected prediction accuracy of normal walking is 94%, walking while wearing a coat (W-Coat) is 95%, and walking while carrying a bag is 97%. In addition, the computational time of all classifiers is plotted in Figure 14, which shows that Medium KNN efficiency is better than all other classifiers in terms of the computational cost.

**Table 4.** Proposed recognition results on CASIA B dataset, where selected angle is 90°.

Classifier	Features			Measures		
	GAP	FC	Proposed	Recall (%)	Accuracy (%)	Time (s)
OaA-SVM	✓	✓		90.10	90.22	242.4426
				88.52	88.64	176.4450
			✓	95.10	95.26	114.2004
Cubic KNN	✓	✓		84.42	84.54	165.5994
				83.60	83.98	111.2011
			✓	93.60	93.60	82.1460
Medium KNN	✓	✓		83.40	83.48	151.0014
				84.80	84.76	104.1446
			✓	93.40	93.46	64.2914
Baggage Tree	✓	✓		85.10	85.16	256.1130
				84.14	84.33	201.0148
			✓	87.50	87.45	117.1106
Naïve Bayes	✓	✓		71.10	71.04	171.2540
				74.94	74.82	104.3360
			✓	79.30	79.30	76.3114

**Table 5.** Confusion matrix of OaA-SVM for CASIA B dataset.

Gait Name	Gait Name		
	Normal Walk	W-Coat	W-Bag
Normal Walk	94%	4%	2%
W-Coat	3%	95%	2%
W-Bag	1%	2%	97%



**Figure 14.** Execution time of proposed method for CASIA B dataset (90° angle).

4.5. Discussion

A detailed discussion on results and comparison is conducted in this section. As shown in Figure 1, the proposed method has a few important steps, such as video preprocessing, deep-learning feature extraction, feature selection through the kurtosis approach, the fusion of selected features through the correlation approach, and finally, the OaA-SVM-based feature classification. The proposed method is evaluated on two datasets. One is recorded in the real-time environment, and the second is CASIA B. The real-time captured dataset

results are given in Tables 1–3. In Table 1, results are presented for normal walking under four different angles, whereas in Table 2, results are given for walking while carrying a bag. The classification of the binary class problem is also conducted, and results are tabulated in Table 3. The OaA-SVM outperformed in all three experiments and can be verified through Table 1. Additionally, we used the CASIA B dataset for the validation of the proposed technique. The results are listed in Table 4 and verified through Table 5.

We also performed experiments on different feature sets to confirm the authenticity of the proposed heuristic-feature-selection approach. For this purpose, we select several feature vectors, such as 300 features, 400 features, 500 features, 600 features, 700 features, and a complete feature vector. Results are given in Table 6 for both datasets. This table shows that the results of 600 features are much better compared to other feature sets. The results increase initially, but after 600 features, accuracy is degraded by approximately 1%. For all features, the accuracy difference is almost 4%. Based on this table, we can say that the proposed technique achieves a significant performance on 600 features.

**Table 6.** Analysis of proposed feature-selection framework on numerous feature sets. \*The OvA-SVM is employed as a classifier for this table.

Dataset	Accuracy (%) on Feature Sets					
	300 Features	400 Features	500 Features	600 Features	700 Features	All Features
Real-time (normal walking)	93.70	94.24	95.35	96.00	95.70	93.04
Real-time (walking while carrying a bag)	94.10	94.90	95.80	96.60	96.32	92.10
Real-time (normal walking vs. walking while carrying a bag)	92.90	93.72	95.30	96.40	96.14	93.50
CASIA B Dataset	92.96	93.40	93.85	95.26	95.10	92.64

The comparison of the proposed method is also listed in Table 7. In this table, we only added those techniques in which the CASIA B dataset was used. For a real-time dataset, the comparison with recent techniques is not fair. This table shows that the previous best-reported accuracy on this dataset was 93.40% [6]. In this work, we have improved this accuracy by almost 2% and reached 95.26%.

**Table 7.** Comparison with existing techniques for CASIA B dataset.

Reference	Year	Dataset	Accuracy (%)
[45]	2015	CASIA B	86.30
[46]	2017	CASIA B	90.60
[37]	2019	CASIA B	87.7
[6]	2020	CASIA B	93.40
	Proposed	CASIA B	95.26
	Proposed	Real-time	96.60

Normally, researchers employ metaheuristic techniques, such as a genetic algorithm, PSO, ACO, and BCO [47]. These techniques have consumed too much time during the selection process. However, the proposed feature-selection approach is based on the single-kurtosis value activation function and is executed fast compared to the GA, PSO, BCO, Whale, and ACO on both the selected dataset, as shown by time plotted in Figures 15 and 16. These figures show that the proposed feature-selection approach is more suitable for gait recognition in terms of computational time than the metaheuristic techniques.

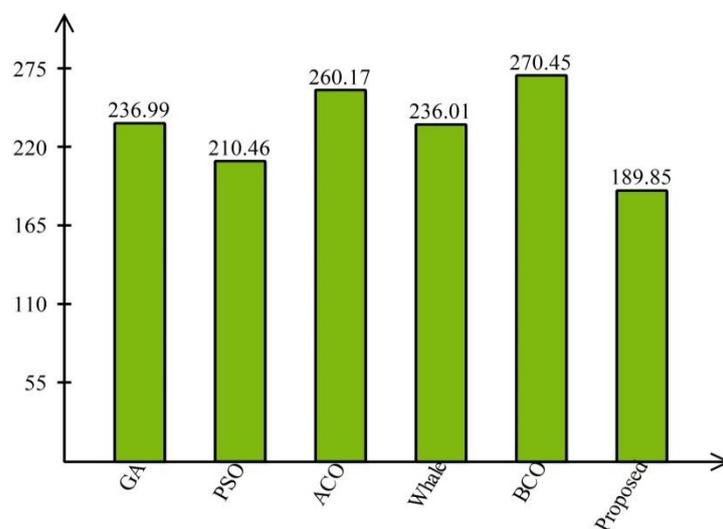


Figure 15. Time comparison for the real-time dataset.

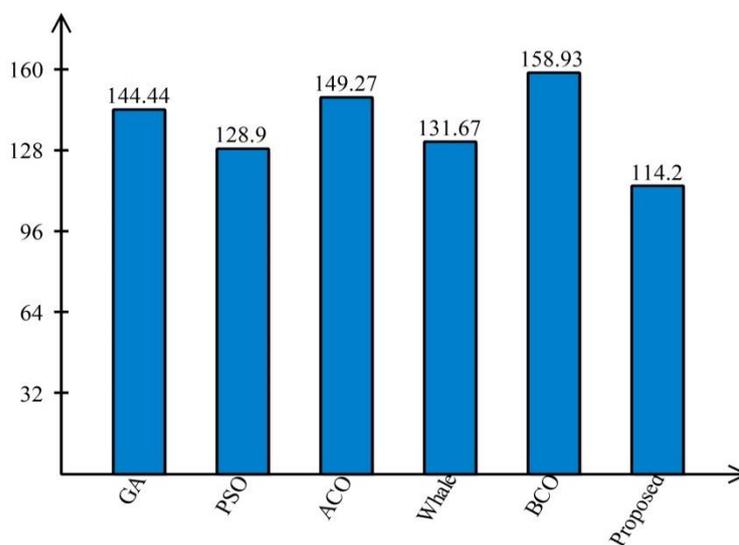


Figure 16. Time comparison on the CASIA B dataset.

A detailed analysis is also conducted through confidence intervals. For this purpose, the proposed framework is executed 500 times for both datasets. For the real-time dataset, we obtained three accuracy values—maximum (96.8%), minimum (95.3%), and average (96.05%). Based on these values, the standard deviation and standard error mean (SEM) are computed and obtain values of 0.75 and 0.5303. Similarly, for the CASIA B dataset, the standard deviation and standard error mean value are 0.56 and 0.3959, respectively. Using standard deviation and standard error mean values, the margin of error is computed, as plotted in Figures 17 and 18. From these figures, it is shown that the proposed framework accuracy is consistent after the number of selected iterations. Moreover, the performance of the proposed framework (OvA-SVM) is also analyzed with a few other classifiers, such as Softmax, ELM, and KELM, as illustrated in Figure 19. From this figure, it can be noted that the proposed framework shows better performance.

Confidence Level	Margin of Error	Error Bar
68.3%, $\sigma_{\bar{x}}$	96.05 $\pm$ 0.53 ( $\pm$ 0.55%)	
90%, 1.645 $\sigma_{\bar{x}}$	96.05 $\pm$ 0.872 ( $\pm$ 0.91%)	
95%, 1.960 $\sigma_{\bar{x}}$	96.05 $\pm$ 1.039 ( $\pm$ 1.08%)	
99%, 2.576 $\sigma_{\bar{x}}$	96.05 $\pm$ 1.366 ( $\pm$ 1.42%)	
99.9%, 3.291 $\sigma_{\bar{x}}$	96.05 $\pm$ 1.745 ( $\pm$ 1.82%)	
99.99%, 3.891 $\sigma_{\bar{x}}$	96.05 $\pm$ 2.064 ( $\pm$ 2.15%)	
99.999%, 4.417 $\sigma_{\bar{x}}$	96.05 $\pm$ 2.342 ( $\pm$ 2.44%)	
99.9999%, 4.892 $\sigma_{\bar{x}}$	96.05 $\pm$ 2.594 ( $\pm$ 2.70%)	

Figure 17. Confidence interval based analysis of proposed method on real-time dataset.

Confidence Level	Margin of Error	Error Bar
68.3%, $\sigma_{\bar{x}}$	94.7 $\pm$ 0.396 ( $\pm$ 0.42%)	
90%, 1.645 $\sigma_{\bar{x}}$	94.7 $\pm$ 0.651 ( $\pm$ 0.69%)	
95%, 1.960 $\sigma_{\bar{x}}$	94.7 $\pm$ 0.776 ( $\pm$ 0.82%)	
99%, 2.576 $\sigma_{\bar{x}}$	94.7 $\pm$ 1.02 ( $\pm$ 1.08%)	
99.9%, 3.291 $\sigma_{\bar{x}}$	94.7 $\pm$ 1.303 ( $\pm$ 1.38%)	
99.99%, 3.891 $\sigma_{\bar{x}}$	94.7 $\pm$ 1.541 ( $\pm$ 1.63%)	
99.999%, 4.417 $\sigma_{\bar{x}}$	94.7 $\pm$ 1.749 ( $\pm$ 1.85%)	
99.9999%, 4.892 $\sigma_{\bar{x}}$	94.7 $\pm$ 1.937 ( $\pm$ 2.05%)	

Figure 18. Confidence-interval-based analysis of proposed method on CASIA B dataset.

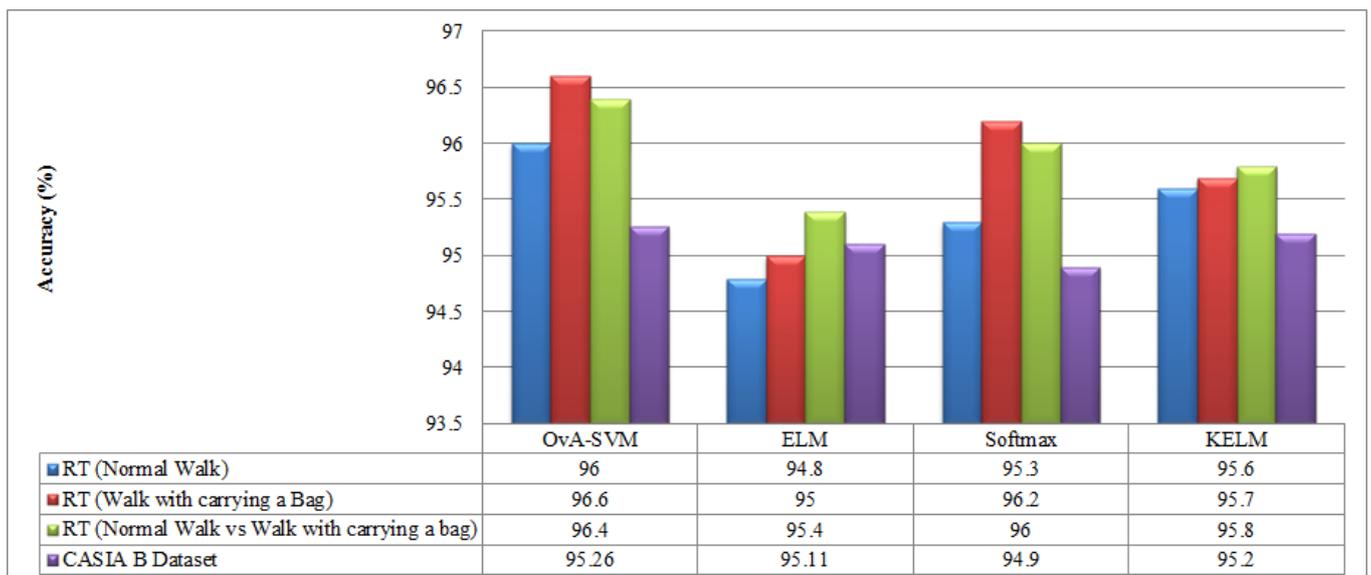


Figure 19. Comparison of OvS-SVM classifier with other classification algorithms for gait recognition.

The reason for the selected hyper parameters is as follows. Normally, the researchers employed gradient descent [48] as an optimization function, but due to the complex nature

of the selected pre-trained model Resnet101 (residual blocks), the ADAM optimizer [49] can work better. The initial learning rate is normally 0.010, but in this work, the selected dataset dimension is high; therefore, the learning rate of 0.001 is suitable. Mini batch size is always selected based on the machine (desktop computer), but we do not have enough resources for the execution on a mini batch size of 128; therefore, we selected a value of 64. The maximum epochs are normally 70–100, but in this work, due to the higher number of video frames, we obtained a better training accuracy after 150 epochs. A few training results that were noted during the training process are given in Tables 8 and 9 below.

**Table 8.** Training accuracy on selected hyper parameters using real-time collected dataset.

Epochs	Accuracy (%)	Error (%)	Time (min)
20	83.5	16.5	221.6784
40	87.9	12.1	375.7994
60	90.2	9.8	588.7834
80	92.6	5.4	792.5673
100	93.9	5.1	875.1247
150	96.8	1.2	988.0045
200	98.1	0.4	1105.5683

**Table 9.** Training accuracy on selected hyper parameters using CASIA B dataset.

Epochs	Accuracy (%)	Error (%)	Time (min)
20	81.4	18.6	174.8957
40	84.6	15.4	292.0645
60	88.0	12	411.4756
80	90.2	9.8	581.8322
100	91.6	8.4	695.4570
150	94.3	5.7	808.5334
200	97.5	2.5	981.6873

## 5. Conclusions

Human gait recognition is an active research domain based on an important biometric application. Through gait, the human walking style can be easily determined in the video sequences. The major use of human gait recognition is in video surveillance, crime prevention, and biometrics. In this work, a deep-learning-based method is presented for human gait recognition in a real-time environment and for offline publicly available datasets. Transfer learning is applied for feature extraction and then selects robust features using a heuristic approach. The correlation formulation is applied for the best-selected feature fusion. In the end, multiclass OaA-SVM is applied for the final classification. The methodology is evaluated on a real-time captured database and CASIA B dataset, and it achieves an accuracy of 96.60% and 95.26%, respectively.

We can conclude that the preprocessing step before the model's learning reduces the error rate. This step further shows strength in selecting the best features, but some features are discarded, which are essential for classification. The kurtosis-controlled entropy (KcE) is a new heuristic feature-selection technique that is executed in less time than the metaheuristic techniques. Another new technique named correlation-formulation-based fusion is used in this work for the best feature fusion. We compare the results of this method to existing methods such as PCA and LDA, and our newly proposed selection technique gives better results and has a shorter computational time for real-time video sequences. Moreover, the fusion process, through correlation formulation, increased the information of the human walking style that later helped the improved gait-recognition accuracy.

The drawback of this work is as follows: in the correlation-formulation-based feature fusion step, the method added some redundant features that later increased the computa-

tional time and reduced the recognition accuracy slightly. In the future, we shall improve the fusion approach and increase the database by adding data from more subjects.

**Author Contributions:** Conceptualization, M.I.S., M.A.K. and M.N.; methodology, M.I.S., M.A.K. and M.N.; software, M.I.S., M.A.K. and M.N.; validation, A.A., S.A. and A.B.; formal analysis, A.A., S.A. and A.B.; investigation, A.A., S.A. and A.B.; resources, A.A., S.A. and A.B.; data curation, M.A.K. and R.D.; writing—original draft preparation, M.A.K., M.I.S. and M.N.; writing—review and editing, R.D. and A.A.; visualization, R.D. and S.A.; supervision, R.D.; project administration, R.D. and A.B.; funding acquisition, R.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data used in this study are available from: <https://drive.google.com/file/d/1g0UBj35Gu0HyjGhdid1NMobn2mPk8PyN/view?usp=sharing> (accessed on 20 December 2021).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Saleem, F.; Khan, M.A.; Alhaisoni, M.; Tariq, U.; Armghan, A.; Alenezi, F.; Choi, J.-I.; Kadry, S. Human Gait Recognition: A Single Stream Optimal Deep Learning Features Fusion. *Sensors* **2021**, *21*, 7584. [[CrossRef](#)] [[PubMed](#)]
- Bendjillali, R.I.; Beladgham, M.; Merit, K.; Taleb-Ahmed, A. Improved Facial Expression Recognition Based on DWT Feature for Deep CNN. *Electronics* **2019**, *8*, 324. [[CrossRef](#)]
- Adjabi, I.; Ouahabi, A.; Benzaoui, A.; Taleb-Ahmed, A. Past, Present, and Future of Face Recognition: A Review. *Electronics* **2020**, *9*, 1188. [[CrossRef](#)]
- Olanrewaju, L.; Oyebiyi, O.; Misra, S.; Maskeliunas, R.; Damasevicius, R. Secure ear biometrics using circular kernel principal component analysis, Chebyshev transform hashing and Bose–Chaudhuri–Hocquenghem error-correcting codes. *Signal Image Video Process.* **2020**, *14*, 847–855. [[CrossRef](#)]
- Rodrigues, J.D.C.; Filho, P.P.R.; Damasevicius, R.; de Albuquerque, V.H.C. EEG-based biometric systems. In *Neurotechnology: Methods, Advances and Applications*; The Institution of Engineering and Technology: London, UK, 2020; pp. 97–153. Available online: [https://www.researchgate.net/publication/340455635\\_Neurotechnology\\_Methods\\_advances\\_and\\_applications](https://www.researchgate.net/publication/340455635_Neurotechnology_Methods_advances_and_applications) (accessed on 17 January 2022).
- Arshad, H.; Khan, M.A.; Sharif, M.I.; Yasmin, M.; Tavares, J.M.R.S.; Zhang, Y.-D.; Satapathy, S.C. A multilevel paradigm for deep convolutional neural network features selection with an application to human gait recognition. *Expert Syst.* **2020**, *20*, 1–21. [[CrossRef](#)]
- Sokolova, A.; Konushin, A. Pose-based deep gait recognition. *IET Biom.* **2018**, *8*, 134–143. [[CrossRef](#)]
- Wang, L.; Li, Y.; Xiong, F.; Zhang, W. Gait Recognition Using Optical Motion Capture: A Decision Fusion Based Method. *Sensors* **2021**, *21*, 3496. [[CrossRef](#)]
- Kim, H.; Kim, H.-J.; Park, J.; Ryu, J.-K.; Kim, S.-C. Recognition of Fine-Grained Walking Patterns Using a Smartwatch with Deep Attentive Neural Networks. *Sensors* **2021**, *21*, 6393. [[CrossRef](#)]
- Hwang, T.-H.; Effenberg, A.O. Head Trajectory Diagrams for Gait Symmetry Analysis Using a Single Head-Worn IMU. *Sensors* **2021**, *21*, 6621. [[CrossRef](#)]
- Khan, M.H.; Li, F.; Farid, M.S.; Grzegorzec, M. Gait recognition using motion trajectory analysis. In *Proceedings of the International Conference on Computer Recognition Systems, Polanica Zdroj, Poland, 22–24 May 2017*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 73–82.
- Manssor, S.A.; Sun, S.; Elhassan, M.A. Real-Time Human Recognition at Night via Integrated Face and Gait Recognition Technologies. *Sensors* **2021**, *21*, 4323. [[CrossRef](#)]
- Liao, R.; Yu, S.; An, W.; Huang, Y. A model-based gait recognition method with body pose and human prior knowledge. *Pattern Recognit.* **2019**, *98*, 107069. [[CrossRef](#)]
- Sharif, M.; Attique, M.; Tahir, M.Z.; Yasmim, M.; Saba, T.; Tanik, U.J. A Machine Learning Method with Threshold Based Parallel Feature Fusion and Feature Selection for Automated Gait Recognition. *J. Organ. End User Comput.* **2020**, *32*, 67–92. [[CrossRef](#)]
- Priya, S.J.; Rani, A.J.; Subathra, M.S.P.; Mohammed, M.A.; Damaševičius, R.; Ubendran, N. Local Pattern Transformation Based Feature Extraction for Recognition of Parkinson’s Disease Based on Gait Signals. *Diagnostics* **2021**, *11*, 1395. [[CrossRef](#)] [[PubMed](#)]
- Khan, S.; Khan, M.A.; Alhaisoni, M.; Tariq, U.; Yong, H.-S.; Armghan, A.; Alenezi, F. Human Action Recognition: A Paradigm of Best Deep Learning Features Selection and Serial Based Extended Fusion. *Sensors* **2021**, *21*, 7941. [[CrossRef](#)]
- Khan, M.A.; Zhang, Y.-D.; Khan, S.A.; Attique, M.; Rehman, A.; Seo, S. A resource conscious human action recognition framework using 26-layered deep convolutional neural network. *Multimed. Tools Appl.* **2021**, *80*, 35827–35849. [[CrossRef](#)]
- Khan, M.A.; Javed, K.; Khan, S.A.; Saba, T.; Habib, U.; Khan, J.A.; Abbasi, A.A. Human action recognition using fusion of multiview and deep features: An application to video surveillance. *Multimed. Tools Appl.* **2020**, 1–27. [[CrossRef](#)]

19. Zebari, D.A.; Ibrahim, D.A.; Zeebaree, D.Q.; Haron, H.; Salih, M.S.; Damaševičius, R.; Mohammed, M.A. Systematic Review of Computing Approaches for Breast Cancer Detection Based Computer Aided Diagnosis Using Mammogram Images. *Appl. Artif. Intell.* **2021**, *1*, 1–47. [[CrossRef](#)]
20. Kassem, M.; Hosny, K.; Damaševičius, R.; Eltoukhy, M. Machine Learning and Deep Learning Methods for Skin Lesion Classification and Diagnosis: A Systematic Review. *Diagnostics* **2021**, *11*, 1390. [[CrossRef](#)]
21. Kumar, V.; Singh, D.; Kaur, M.; Damaševičius, R. Overview of current state of research on the application of artificial intelligence techniques for COVID-19. *PeerJ Comput. Sci.* **2021**, *7*, e564. [[CrossRef](#)]
22. Mehmood, A.; Khan, M.A.; Sharif, M.; Khan, S.A.; Shaheen, M.; Saba, T.; Riaz, N.; Ashraf, I. Prosperous Human Gait Recognition: An end-to-end system based on pre-trained CNN features selection. *Multimed. Tools Appl.* **2020**, *11*, 1–21. [[CrossRef](#)]
23. Anusha, R.; Jaidhar, C.D. Clothing invariant human gait recognition using modified local optimal oriented pattern binary descriptor. *Multimed. Tools Appl.* **2020**, *79*, 2873–2896. [[CrossRef](#)]
24. Khan, M.A.; Akram, T.; Sharif, M.; Muhammad, N.; Javed, M.Y.; Naqvi, S.R. Improved strategy for human action recognition; experiencing a cascaded design. *IET Image Process.* **2020**, *14*, 818–829. [[CrossRef](#)]
25. Kadry, S.; Rajinikanth, V.; Taniar, D.; Damaševičius, R.; Valencia, X.P.B. Automated segmentation of leukocyte from hematological images—A study using various CNN schemes. *J. Supercomput.* **2021**, *1*–21. [[CrossRef](#)]
26. Tanveer, M.; Rashid, A.H.; Ganaie, M.; Reza, M.; Razzak, I.; Hua, K.-L. Classification of Alzheimer’s disease using ensemble of deep neural networks trained through transfer learning. *IEEE J. Biomed. Health Inform.* **2021**. [[CrossRef](#)]
27. Khan, M.Z.; Saba, T.; Razzak, I.; Rehman, A.; Bahaj, S.A. Hot-Spot Zone Detection to Tackle Covid19 Spread by Fusing the Traditional Machine Learning and Deep Learning Approaches of Computer Vision. *IEEE Access* **2021**, *9*, 100040–100049. [[CrossRef](#)]
28. Rehman, A.; Naz, S.; Razzak, M.I.; Hameed, I.A. Automatic Visual Features for Writer Identification: A Deep Learning Approach. *IEEE Access* **2019**, *7*, 17149–17157. [[CrossRef](#)]
29. Alyasseri, Z.A.A.; Al-Betar, M.A.; Abu Doush, I.; Awadallah, M.A.; Abasi, A.K.; Makhadmeh, S.N.; Alomari, O.A.; Abdulkareem, K.H.; Adam, A.; Damasevicius, R.; et al. Review on COVID-19 diagnosis models based on machine learning and deep learning approaches. *Expert Syst.* **2021**, e12759. [[CrossRef](#)]
30. Castro, F.M.; Marín-Jiménez, M.J.; Guil, N.; De La Blanca, N.P. Multimodal feature fusion for CNN-based gait recognition: An empirical comparison. *Neural Comput. Appl.* **2020**, *32*, 14173–14193. [[CrossRef](#)]
31. Rawat, W.; Wang, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Comput.* **2017**, *29*, 2352–2449. [[CrossRef](#)]
32. Bari, A.S.M.H.; Gavrilova, M.L. Artificial Neural Network Based Gait Recognition Using Kinect Sensor. *IEEE Access* **2019**, *7*, 162708–162722. [[CrossRef](#)]
33. Zheng, S.; Zhang, J.; Huang, K.; He, R.; Tan, T. Robust view transformation model for gait recognition. In Proceedings of the 2011 18th IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 2073–2076.
34. Wang, X.; Yan, W.Q. Human Gait Recognition Based on Frame-by-Frame Gait Energy Images and Convolutional Long Short-Term Memory. *Int. J. Neural Syst.* **2019**, *30*, 1950027. [[CrossRef](#)] [[PubMed](#)]
35. Anusha, R.; Jaidhar, C.D. Human gait recognition based on histogram of oriented gradients and Haralick texture descriptor. *Multimed. Tools Appl.* **2020**, *79*, 8213–8234. [[CrossRef](#)]
36. Zhao, A.; Li, J.; Ahmed, M. SpiderNet: A spiderweb graph neural network for multi-view gait recognition. *Knowl. Based Syst.* **2020**, *206*, 106273. [[CrossRef](#)]
37. Arshad, H.; Khan, M.A.; Sharif, M.; Yasmin, M.; Javed, M.Y. Multi-level features fusion and selection for human gait recognition: An optimized framework of Bayesian model and binomial distribution. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 3601–3618. [[CrossRef](#)]
38. Ferroukhi, M.; Ouahabi, A.; Attari, M.; Habchi, Y.; Taleb-Ahmed, A. Medical Video Coding Based on 2nd-Generation Wavelets: Performance Evaluation. *Electronics* **2019**, *8*, 88. [[CrossRef](#)]
39. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [[CrossRef](#)]
40. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv* **2016**, arXiv:1602.07261.
41. Torrey, L.; Shavlik, J. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*; IGI Global: Hershey, PA, USA, 2010; pp. 242–264.
42. Naheed, N.; Shaheen, M.; Khan, S.A.; Alawairdhi, M.; Khan, M.A. Importance of Features Selection, Attributes Selection, Challenges and Future Directions for Medical Imaging Data: A Review. *Comput. Model. Eng. Sci.* **2020**, *125*, 315–344. [[CrossRef](#)]
43. Liu, Y.; Zheng, Y.F. One-against-all multi-class SVM classification using reliability measures. In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005; pp. 849–854.
44. Arai, K.; Andrie, R. Gait recognition method based on wavelet transformation and its evaluation with chinese academy of sciences (casia) gait database as a human gait recognition dataset. In Proceedings of the 2012 Ninth International Conference on Information Technology–New Generations, Las Vegas, NV, USA, 16–18 April 2012; pp. 656–661.
45. Arora, P.; Hanmandlu, M.; Srivastava, S. Gait based authentication using gait information image features. *Pattern Recognit. Lett.* **2015**, *68*, 336–342. [[CrossRef](#)]
46. Castro, F.M.; Marín-Jiménez, M.J.; Mata, N.G.; Muñoz-Salinas, R. Fisher motion descriptor for multiview gait recognition. *Int. J. Pattern Recognit. Artif. Intell.* **2017**, *31*, 1756002. [[CrossRef](#)]

47. Agrawal, P.; Abutarboush, H.F.; Ganesh, T.; Mohamed, A.W. Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009–2019). *IEEE Access* **2021**, *9*, 26766–26791. [[CrossRef](#)]
48. Amari, S.-I. Backpropagation and stochastic gradient descent method. *Neurocomputing* **1993**, *5*, 185–196. [[CrossRef](#)]
49. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015.