applied sciences

MDPI

Article

VGG16-MLP: Gait Recognition with Fine-Tuned VGG-16 and Multilayer Perceptron

Jashila Nair Mogan, Chin Poo Lee *, Kian Ming Lim and Kalaiarasi Sonai Muthu

Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia; 1121116804@student.mmu.edu.my (J.N.M.); kmlim@mmu.edu.my (K.M.L.); kalaiarasi@mmu.edu.my (K.S.M.)
* Correspondence: cplee@mmu.edu.my

Abstract: Gait is a pattern of a person's walking. The body movements of a person while walking makes the gait unique. Regardless of the uniqueness, the gait recognition process suffers under various factors, namely the viewing angle, carrying condition, and clothing. In this paper, a pre-trained VGG-16 model is incorporated with a multilayer perceptron to enhance the performance under various covariates. At first, the gait energy image is obtained by averaging the silhouettes over a gait cycle. Transfer learning and fine-tuning techniques are then applied on the pre-trained VGG-16 model to learn the gait features of the attained gait energy image. Subsequently, a multilayer perceptron is utilized to determine the relationship among the gait features and the corresponding subject. Lastly, the classification layer identifies the corresponding subject. Experiments are conducted to evaluate the performance of the proposed method on the CASIA-B dataset, the OU-ISIR dataset D, and the OU-ISIR large population dataset. The comparison with the state-of-the-art methods shows that the proposed method outperforms the methods on all the datasets.

Keywords: gait; gait recognition; deep learning; pre-trained model; multilayer perceptron

Citation: Mogan, J.N.; Lee, C.P.; Lim, K.M.; Muthu, K.S. VGG16-MLP: Gait Recognition with Fine-Tuned VGG-16 and Multilayer Perceptron. *Appl. Sci.* **2022**, *12*, 7639. https://doi.org/10.3390/app12157639

Academic Editor: Hanatsu Nagano

Received: 27 June 2022
Accepted: 19 July 2022
Published: 29 July 2022

1. Introduction

Identifying an individual based on the way they walk is known as gait recognition. The key point that facilitates gait recognition is the changes that take place in the body parts when a person is walking. Gait recognition concerns robust to low-resolution images, which were taken from a far distance. Moreover, gait is difficult to imitate as the body movements while walking are different for every person. Unlike other biometric modalities, cooperation of an individual is not needed to perform gait recognition. Nevertheless, the gait recognition process can be affected by the involvement of variations, namely the viewing angle, carrying condition, and clothing.

In the early days, handcrafted methods were proposed to solve the aforementioned issue. The handcrafted methods can be categorized as model-based and model-free methods. Model-based methods require a walking human skeleton model to learn the gait features. In contrast, model-free methods extract the features from the human gait silhouettes. Due to this difference, the computational cost is higher for model-based methods than model-free methods. Other than that, the model-free methods are easy to execute. Several handcrafted methods managed to achieve high accuracy when the variations are engaged. However, the salient information could be ignored in handcrafted methods as these methods only extract the features, which are manually defined.

Deep learning methods are popular among the researchers as the methods are capable of learning the most discriminative features by themselves. The deep learning methods extract a large number of complex features, which highly impacts the accuracy. Among the deep learning methods, the incorporation of a pre-trained convolutional neural network (CNN) model has become trendy. This is because the pre-trained model was learned from a large dataset to obtain prominent features. The pre-trained models that are mostly

being utilized for gait recognition are ResNet [1], DenseNet [2], AlexNet [3], and VGG [4]. Integration of the pre-trained models using transfer learning and fine-tuning techniques improves the accuracy on a large scale. Nevertheless, not many researchers have applied the pre-trained models in their work.

Consequently, the incorporation of a pre-trained model and a multilayer perceptron is proposed in this paper to achieve better accuracy under various factors. The method first averages the gait silhouettes over a gait cycle to attain the gait energy image (GEI). A pre-trained VGG-16 model is then employed to extract the gait features of the obtained GEIs. The multilayer perceptron is used to identify the association between the features and the subjects. Finally, the subjects are classified based on the features in the classification layer.

The main contributions of this work are specified below:

- The extraction of deep features by exploiting the pre-trained VGG-16 model using transfer learning and fine-tuning techniques.
- The correlation of the obtained gait features and the associated subject is determined by utilizing the multilayer perceptron.
- The generalization capability of the proposed method was assessed on CASIA-B, OU-ISIR D, and OU-LP datasets.

2. Related Works

The handcrafted methods encode the predetermined low-level features, while the deep-learning-based methods extract both low-level and intricate features. Over the years, numerous methods were presented to solve the gait recognition problem.

2.1. Handcrafted Approach

Handcrafted methods can be grouped as model-based methods and model-free methods. Model-based methods [5–9] utilize a human model to obtain the gait features, namely limb joint angles and length between joints. Ahmed et al. (2014) [10] extracted horizontal distance features and vertical distance features in a gait cycle by using Kinect sensors. Four features were chosen for horizontal distance features, while six features were chosen for vertical distance features. In Sattrupai and Kusakunniran (2018) [11], the dense trajectory technique was utilized to capture the gait features. The obtained features were grouped using the k-means clustering process. The subject classification was performed using the k-nearest neighbor technique.

Sun et al. (2018) [12] selected static features and dynamic features based on the changes occurring while walking. The features that do not change were taken as static features, while features that changed were considered as dynamic features based on a walking skeleton model. Both the features were then fused together, which produced better accuracy than a single feature. Sharif et al. (2020) [13] extracted shape features using histograms of oriented gradients (HOGs), geometric features, and texture features with local binary patterns (LBPs). Principal component analysis (PCA) was applied on the features for feature reduction. A support vector machine (SVM) was employed to perform the subject classification.

On the other hand, the model-free methods [14–22] extract features from the gait silhouettes without needing any skeleton model. Lishani et al. (2014) [23] segmented the GEI into three parts, namely the top, middle, and bottom, to capture the gait information. Other than that, Haralick texture features were extracted from GEI as the features are accentuated on textures. Subsequently, Lee et al. (2015) [24] studied pixel-wise binary patterns for every gait cycle, known as transient binary patterns (TBPs). The pixel-wise TBPs were converted into region-level blocks and then made into histograms. All the histograms were combined into global TBPs, denoting the motion information and spatial location. In Alvarez and Sahonero-Alvarez (2018) [25], the head and feet parts of a human body in GEI were considered as features. The feature extraction and feature reduction were accomplished using the PCA algorithm. Khan et al. (2019) [26] utilized the optical flow field to extract the dense trajectories from the gait sequence. Three local descriptors were

calculated to extract the motion and appearance information, namely HOGs, histograms of optical flow (HOFs), and motion boundary histogram (MBHs). Mogan et al. (2020) [27] convolved pre-learned filters with gait video frames, producing a set of feature maps. All the feature maps were segmented into several regions and the gradient of each pixel was calculated. The obtained gradients were then constructed into a final histogram.

2.2. Deep Learning

The main feature of the deep learning methods is the layered structure, which is arranged in a tiered structure. The layers near to the input extract low-level features such as texture and edges. The intricacy of the feature extraction grows with the layers. The obtained low-level features are then merged to produce more intricate representation. CNNs are the most preferred method among the other deep learning methods as CNN manages to capture the significant features from an image. Yeoh et al. (2016) [28] explored the application of CNN for clothing invariant gait recognition. The CNN was made up of three convolutional layers, two fully connected layers, and a Softmax layer. Shiraga et al. (2016) [29] presented a network called GEINet, which consists of two sequential triplets in the convolution layer, the pooling layer, and the normalization layer, as well as two fully connected layers and a classifier layer. As the input of the network, the GEI was obtained by averaging the silhouettes over a gait cycle. All the convolution layers and the first fully connected layer utilized the ReLU activation function. Max pooling was used in the pooling layer, while local response normalization was employed in the normalization layer. The Softmax function was applied in the classifier layer. Similarly, Alotaibi and Mahmood (2017) [30] developed a deep CNN, which accepts GEI as the input of the network. The network was made up of four convolution layers, eight pooling layers, and a fully connected layer. The Tanh function was utilized as the activation function in the convolution layers. The max-pooling algorithm was applied in the pooling layers. The subjects were classified using the Softmax function. SGD was employed as the optimizer to reduce the cost function. Min et al. (2019) [31] proposed a deep CNN network that contains four convolution layers, four max-pooling layers, one fully connected layer, and a classifier layer. The Adam optimizer was used to speed up the network convergence. The leaky ReLU activation function was utilized in the convolution and fully connected layers. The classification was performed using the Softmax function. Tong et al. (2017) [32] presented a triplet CNN-based network where the weights are shared among the three networks. The triplet network accepts triplet inputs as positive, query, and negative. The triplet loss function was used to train the network. Later, Tong et al. (2018) [33] proposed a deep neural network, which consists of a temporal feature network (TFN) and a spatial feature network (SFN). The low-level features were captured by TFN, while the spatial features were extracted by SFN.

Aung and Pluempitiwiriyawej (2020) [34] developed a network consisting of eight convolution layers, five pooling layers, two fully connected layers, and a classifier layer. The ReLU activation function was utilized in the convolution and fully connected layers. Max pooling was applied in the pooling layers. The classification was performed using the Softmax function. SGD was employed as the network optimizer. Zhu et al. (2020) [35] constructed a network with three parallel convolution layers and a shared fully connected block, which accepts binarized silhouettes as inputs. The negative log-likelihood loss function was applied to train the network. Su et al. (2020) [36] constructed a network comprising six convolution layers, three max-pooling layers, and five fully connected layers. In order to learn all the possible information from both positive and negative samples, a loss function named center-ranked loss was proposed. Balamurugan et al. (2021) [37] presented 11 layers network for gait recognition problem. The network comprises four convolution layers, four pooling layers, two fully connected layers, and a classifier layer. The ReLU activation function was applied in the convolution and fully connected layers. Max pooling was used in the pooling layers. The Softmax function was employed to classify the subjects. The stochastic gradient decent with momentum was utilized as the

optimizer. Subsequently, Han et al. (2022) [38] investigated the use of angular Softmax loss and triplet loss to learn the distinguishable features and make the obtained features more distinctive. GaitSet [39] was utilized as the backbone network to study the effects of the loss functions. Elharrouss et al. (2021) [40] developed two separate CNN models to perform angle estimation and gait recognition. The angle of the image captured was detected and fed into the second CNN model to identify the gait.

Several existing works applied transfer learning techniques to learn gait features using a pre-trained network. Arshad et al. (2020) [41] proposed a method where pre-trained VGG-19 and AlexNet models were used to extract the gait features. In order to choose the finest sets of the acquired features, entropy and skewness vectors were calculated. Mehmood et al. (2020) [42] employed a pre-trained DenseNet-201 model without any fine-tuning for gait recognition under different view angles. One against the all multi-support vector machine was utilized for the subject classification. Similarly, Ambika and Radhika (2021) [43] utilized a pre-trained DenseNet-201 model, where the classifier layer was tweaked to use the Softmax function with ten neurons as the work involves ten subjects' classification. Mogan et al. (2022) [44] integrated pre-trained DenseNet-201 and multilayer perceptron to reduce the effects of varying conditions in gait recognition. The pre-trained DenseNet-201 was used to encode the salient features of gait energy images. The relationship between the learnt features and the associated class was discovered using the multilayer perceptron.

3. VGG-16 and Multilayer Perceptron

The binary human silhouettes throughout the gait cycles are first captured in the gait energy image. Subsequently, the gait energy image is passed to the pre-trained VGG-16 model for fine-tuning. The output from the VGG-16 model is the feature map that encodes the low-level and high-level representation of the gait energy image. The multilayer perceptron then learns the relationships between the feature map and the associated class. Lastly, a classification layer returns the class probability distributions. The final class label corresponds to the class with the highest probability.

3.1. Gait Energy Image

The gait energy image (GEI) [45] consists of static information at the top of the image, while the dynamic information at the bottom of the image. Both the static and dynamic information are crucial in the gait recognition process. Hence, the GEI is widely applied in the gait biometric domain. The GEI is calculated by averaging the frames over a gait cycle as shown below:

$$\mathrm{GEI} = \frac{1}{T}\sum_{t=1}^{T} I_t(c, r) \tag{1}$$

where T is the total number of frames of a gait cycle and $I_t(c, r)$ is the gait silhouette with image pixel at column c and row r at time t. The acquired GEIs for all the datasets are normalized to the 128 × 128 input size. Examples of GEIs are displayed in Figure 1.

3.2. Network Architecture

The network consists of a pre-trained VGG-16 model and a multilayer perceptron. The pre-trained VGG-16 model is employed to extract the deep gait features. The multilayer perceptron is added to the network to further encode the relationship between the learned features and the associated class. Figure 2 shows the architecture of the proposed network.

Figure 1. Sample GEIs obtained from different datasets, CASIA-B (**first row**), OU-ISIR D (**second row**), and OU-LP (**third row**).

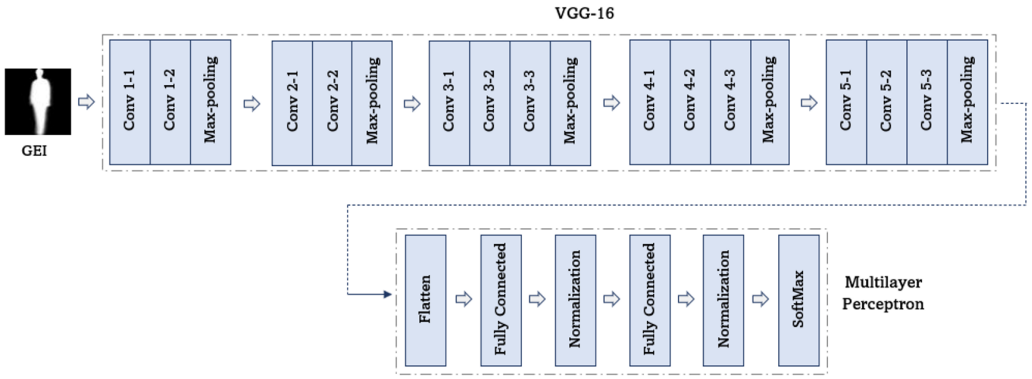

Figure 2. The architecture of the proposed VGG16-MLP model.

3.2.1. Fine-Tuning VGG-16

The adaptation of learned knowledge of a pre-trained model from one problem to another problem is known as transfer learning. In deep learning, transfer learning is a popular technique because the model was pre-trained on a large dataset, thus requiring significantly lower computational resources. Mostly, the technique is applied by transferring the learned features of pre-trained models to solve downstream tasks. To ensure that the pre-trained model effectively adapts to the downstream tasks, fine tuning is required. During the fine-tuning process, the entire model or part of the model is unfrozen. The number of dense layers and classifier layers to be added to the network depends on the complexity of the downstream tasks.

In this work, a pre-trained VGG-16 model [46] was fine-tuned. The VGG-16 model is an improved version of the AlexNet [3] model where the VGG-16 model consists of more convolutional layers and uses the smallest kernel size. The VGG-16 model was trained over the ImageNet dataset that contains 15 million images. The VGG-16 model is made up of 13 convolutional layers, 5 max-pooling layers, and 3 fully connected layers, where the convolutional and max-pooling layers were segmented into five sets. The first two sets contain two convolutional layers followed by a max-pooling layer. The subsequent three sets consist of three convolutional layers and a max-pooling layer. The kernel size of 3×3 with stride 1 and padding 1 was used throughout the network that makes the model stand out among the other pre-trained models. The usage of the smallest kernel size greatly reduces the number of parameters. Moreover, the application of the smallest kernel size also avoided the network from becoming overfit. Max pooling was performed over a 2×2-pixel window with stride 2. By doing so, the dimension of the feature maps is halved from the original size. The rectified linear unit (ReLU) activation function was applied on

all the convolutional layers, as it is computationally efficient and reduces the vanishing gradient problem.

The pre-trained VGG-16 model is chosen due to its easy implementation. Furthermore, a smaller number of parameters are involved in the model, which resulted in a faster learning network.

3.2.2. Multilayer Perceptron

The feature map obtained from the last max-pooling layer is flattened into a vector and fed to the multilayer perceptron. The multilayer perceptron is made up of two fully connected layers, two batch normalization layers, and a classifier layer. Both the fully connected layers consist of 512 neurons. The association among the obtained features and the class label is established in the fully connected layer. The leaky ReLU activation function is utilized in both the fully connected layers. Leaky ReLU is an improved version of the ReLU activation function, where the leaky ReLU function is able to represent the negative values along with the positive values. In doing so, the layer becomes more optimized and balanced, which speeds up the training process. The leaky ReLU activation function is computed as below:

$$f(x) = \left\{ \begin{array}{ll} x & x > 0 \\ \alpha x & x \leqslant 0 \end{array} \right\} \tag{2}$$

where x is to be multiplied with α when the x is a negative value. As a consequence, the neurons in the negative regions are triggered and produce an output. Other than that, a dropout technique is used in the fully connected layers to avoid overfitting problems where several neurons are dropped randomly throughout the training. By doing so, the dropped neurons' weight is not revised and the contribution of the neurons is disregarded.

Batch normalization layers are included after both the fully connected layers. The batch normalization is performed in mini batches, where subtraction and division of the mini batch's mean and standard deviation are carried out. The normalization ensures the training is executed in an efficient manner. As the gait recognition problem involves multiple classes, the Softmax function is used in the classifier layer. The Softmax function outputs the probability of each input pertaining to a particular class. The Softmax function is calculated as:

$$S(y_i) = \frac{\exp(y_i)}{\sum_{j=1}^{n} \exp(y_j)} \tag{3}$$

where y_i denotes the Softmax activation for class i and n is the number of classes.

The Adam optimizer is employed during the training to accelerate the network convergence. In order to avoid over-training the network, an early stopping mechanism is applied in the model. The early-stopping mechanism plays a significant role in stopping the training process when the performance has ceased enhancing. Subsequently, the categorical cross entropy loss function is employed in the VGG16-MLP model as gait recognition involves multi-class classification. The categorical cross entropy loss function is calculated as:

$$\text{loss} = -\sum_{i=1}^{n} \hat{y}_i \cdot \log y_i \tag{4}$$

where $\hat{y}_i$ is the true class label, y_i is the Softmax activation for class i, and n is the number of classes. The layer-wise architecture of the proposed model is presented in Table 1.

Table 1. Layer-wise architecture of the proposed VGG16-MLP model.

Model	Layers	Configurations
Pre-trained VGG-16	Conv 1-1	3 × 3 conv, stride = 1, padding = 1
	Conv 1-2	3 × 3 conv, stride = 1, padding = 1
	Max-Pooling	2 × 2, stride = 2
	Conv 2-1	3 × 3 conv, stride = 1, padding = 1
	Conv 2-2	3 × 3 conv, stride = 1, padding = 1
	Max-Pooling	2 × 2, stride = 2
	Conv 3-1	3 × 3 conv, stride = 1, padding = 1
	Conv 3-2	3 × 3 conv, stride = 1, padding = 1
	Conv 3-3	3 × 3 conv, stride = 1, padding = 1
	Max-Pooling	2 × 2, stride = 2
	Conv 4-1	3 × 3 conv, stride = 1, padding = 1
	Conv 4-2	3 × 3 conv, stride = 1, padding = 1
	Conv 4-3	3 × 3 conv, stride = 1, padding = 1
	Max-Pooling	2 × 2, stride = 2
	Conv 5-1	3 × 3 conv, stride = 1, padding = 1
	Conv 5-2	3 × 3 conv, stride = 1, padding = 1
	Conv 5-3	3 × 3 conv, stride = 1, padding = 1
	Max-Pooling	2 × 2, stride = 2
Multilayer Perceptron	Fully Connected	512
	Batch Normalization	-
	Leaky ReLU	-
	Dropout	0.3
	Fully Connected	512
	Batch Normalization	-
	Leaky ReLU	-
	Dropout	0.3
	SoftMax	-

4. Experiments and Discussions

This section discusses the datasets used in the evaluation, hyperparameter settings, and performance comparison of the proposed VGG16-MLP model with the state-of-the-art methods.

4.1. Datasets

The CASIA-B dataset [47] is a large multi-view gait database, which contains 124 subjects. The gait sequences were captured from 11 viewing angles, with different clothings and carrying conditions. Ten sequences were recorded for every subject, six of which acquired under normal walking, two of which were acquired with coats on, and the remaining two were acquired with bags.

The OU-ISIR dataset D [48] consists of 185 individuals with 370 gait sequences, recorded from a lateral view. The dataset comprises two subsets, namely DBlow and DBhigh. Each of the subsets contain 100 individuals with high fluctuation (DB_{high}) and small fluctuation (DB_{low}). Normalized autocorrelation (NAC) was used to compute the gait fluctuations in every gait sequence.

The OU-LP dataset [49] is a relatively large dataset that comprises 4016 individuals (1 to 94 years old). The dataset was categorized into two subsets where two sequences were recorded for every individual in sequence A, while one sequence was recorded for every individual in sequence B. The gait sequences were captured from four viewing angles, namely 55°, 65°, 75°, and 85°. In this work, sequence A with 3916 individuals is used for performance evaluation. A summary of the datasets is presented in Table 2.

Table 2. Summary of datasets.

Datasets	Number of Subjects	Sequences	Angle Views	Variations
CASIA-B	124	10	11	Normal walking, carrying condition, clothing
OU-ISIR DB_{low}	100	370	1	Steady walking
OU-ISIR DB_{high}	100	370	1	Fluctuated walking
OU-ISIR LP (Sequence A)	3916	2	4	4 viewing angles

4.2. Hyperparameter Tuning

Hyperparameter tuning is essential in order to optimize the performance of the deep learning models. In this work, the hyperparameters were tuned using a grid search technique, which was applied on the CASIA-B dataset. The tuning engages four hyperparameters, i.e., the batch size B, dropout value P, learning rate L, and optimizer θ. The grid search is performed by altering the value of a particular hyperparameter at a time, while the values of the other three hyperparameters remain constant. The tested and optimal values of hyperparameters for the proposed method are shown in Table 3. Based on the highest accuracy with a low computation time, the optimal value for each hyperparameter is chosen.

Table 3. Summary of hyperparameter tuning and optimal hyperparameter settings for the proposed VGG16-MLP model.

Hyperparameters	Tested Values	Optimal Value
Batch Size	32, 64, 128	32
Dropout Value	0.2, 0.3, 0.4	0.3
Learning Rate	0.0001, 0.001, 0.01	0.0001
Optimizer	SGD, Adam	Adam

Table 4 displays the accuracy of the proposed VGG16-MLP at different batch sizes B. The batch size at 32 acquired the highest accuracy, even though the time consumed is slightly longer compared to larger batch sizes. The larger batch sizes are computationally expensive and the accuracy is lower than the smaller batch sizes.

Table 4. Accuracy at different batch sizes B [P = 0.3, L = 0.0001, θ = Adam].

Batch Size	Accuracy (%)	Execution Time (s)
32	**100.00**	**695.8941**
64	99.93	590.2854
128	99.85	537.8056

The accuracy of the proposed VGG16-MLP at various dropout rates P is shown in Table 5. The highest accuracy is obtained at dropout values 0.3 and 0.4. However, the dropout value 0.3 takes less computation time than the dropout value 0.4. The dropout technique is a stochastic regularization that mitigates network overfitting.

Table 5. Accuracy at different dropout rates P [B = 32, L = 0.0001, θ = Adam].

Dropout Value	Accuracy (%)	Time (s)
0.2	99.93	653.8167
0.3	**100.00**	**695.8941**
0.4	100.00	777.492

The accuracy of the proposed method using various learning rates L is illustrated in Table 6. The highest accuracy is attained when the learning rate is set to 0.0001. A smaller learning rate is more appropriate for a large network than a larger learning rate to reduce overfitting issues.

Table 6. Accuracy at different learning rates L [B = 32, P = 0.3, θ = Adam].

Learning Rate	Accuracy (%)	Time (s)
0.0001	**100.00**	**695.8941**
0.001	97.87	1134.4350
0.01	0.96	406.1634

Table 7 presents the accuracy of the proposed method at different optimizers θ. The Adam optimizer is known to be the suitable optimizer when noisy or sparse gradients are involved. Due to this, the Adam optimizer achieved higher accuracy on the CASIA-B dataset than SGD. Furthermore, the Adam optimizer requires less memory and consumes less time to converge, which makes it computationally efficient.

Table 7. Accuracy at different optimizers θ [B = 32, P = 0.3, L = 0.0001].

Optimizer	Accuracy (%)	Time (s)
SGD	99.20	2280.8636
Adam	**100.00**	**695.8941**

4.3. Comparison Results

In this experiment, the performance of the proposed method is compared with five existing methods, namely GEINet [29], deep CNN [30], CNN with leaky ReLU [31], CNN [34], and deep CNN [37]. All the datasets are partitioned into 80% training, 10% validation, and 10% testing. The input image size of 128 × 128 is used for all the existing methods to have an objective comparison. The accuracy of the proposed VGG16-MLP and the existing methods are presented in Table 8.

Table 8. Comparison results on different datasets.

Methods	Accuracy (%)			
	CASIA-B	**OU-ISIR DB$_{high}$**	**OU-ISIR DB$_{low}$**	**OU-LP**
GEINet [29]	95.66	99.86	99.72	88.34
Deep CNN [30]	54.82	96.73	97.15	12.02
CNN [31]	98.75	99.86	99.65	88.88
CNN [34]	92.94	99.10	97.85	0.0005
Deep CNN [37]	91.17	98.26	97.98	55.40
VGG16-MLP	**100.00**	**100.00**	**100.00**	**99.10**

As the CASIA-B dataset consists of incomplete silhouettes, the performance of most of the existing methods have marginally decreased, specifically the deep CNN [30] method. Even so, the proposed VGG16-MLP method achieves the highest accuracy as 100%. This is due to the incorporation of the fine-tuned VGG-16 model and multilayer perceptron, which

has the ability to project complex patterns, such as incomplete and noisy silhouettes. Moreover, the effects of the incomplete and noise silhouettes are suppressed, which encourages the performance of the method.

All the existing methods acquired high accuracy in the OU-ISIR dataset D due to the small number of subjects and clean silhouettes. In both the DB_{high} and DB_{low} datasets, the proposed VGG16-MLP method obtained 100% accuracy. The experimental results demonstrate that both pre-trained model and multilayer perceptron are able to produce promising performance with both small and large datasets. The pre-trained VGG-16 model represents the features of the GEIs in the feature map, while the multilayer perceptron captures the associations between the feature map and classes. Thus, incorporation of the pre-trained VGG-16 model and the multilayer perceptron contributes to the outstanding performance in the OU-ISIR dataset D.

Using the OU-LP dataset, the accuracy of the [30,34,37] methods are quite low due to the large number of subjects. As the network was developed for a small number of subjects, the performance is quite low. Nonetheless, the proposed method obtained 99.10% accuracy, which shows the scalability and generalization capability of the proposed VGG16-MLP method. The good performance is also attributable to the techniques applied, namely the fine tuning of the pre-trained model, multilayer perceptron, early stopping, the dropout technique, and batch normalization.

5. Conclusions

Variations such as the viewing angle, carrying conditions, and clothing cause great challenges in the gait recognition process. This paper proposes the incorporation of a pre-trained VGG-16 model and a multilayer perceptron. The gait energy image is obtained by averaging the gait silhouettes over a gait cycle. The deep gait features of the GEIs are extracted using the pre-trained VGG-16 model via the transfer learning technique. The fully connected layers, batch normalization, and classifier in the multilayer perceptron defined the association among the features and corresponding class. The results show that the proposed model works well with large datasets, fluctuating the walking style. Other than that, integration of the pre-trained VGG-16 model and the multilayer perceptron can suppress the effect of noise and incomplete silhouettes, which promotes the performance of the proposed method.

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

Funding: The research in this work was supported by the Fundamental Research Grant Scheme of the Ministry of Higher Education under award number FRGS/1/2021/ICT02/MMU/02/4, Multimedia University Internal Research Grant with award number MMUI/220021, and Yayasan Universiti Multimedia MMU/YUM/C/2019/YPS.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
2. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
3. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012), Lake Tahoe, NV, USA, 3–6 December 2012; Volume 25.
4. Vedaldi, A.; Zisserman, A. *VGG Convolutional Neural Networks Practical*; Department of Engineering Science, University of Oxford: Oxford, UK, 2016.
5. Zhen, H.; Deng, M.; Lin, P.; Wang, C. Human gait recognition based on deterministic learning and Kinect sensor. In Proceedings of the 2018 Chinese Control And Decision Conference (CCDC), Shenyang, China, 9–11 June 2018.
6. Deng, M.; Wang, C. Human gait recognition based on deterministic learning and data stream of Microsoft Kinect. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 3636–3645. [CrossRef]
7. Choi, S.; Kim, J.; Kim, W.; Kim, C. Skeleton-based gait recognition via robust frame-level matching. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2577–2592. [CrossRef]
8. Sah, S.; Panday, S.P. Model Based Gait Recognition Using Weighted KNN. In Proceedings of the 8th IOE Graduate Conference, London, UK, 17–19 July 2020.
9. De Lima, V.C.; Melo, V.H.; Schwartz, W.R. Simple and efficient pose-based gait recognition method for challenging environments. *Pattern Anal. Appl.* **2021**, *24*, 497–507. [CrossRef]
10. Ahmed, M.; Al-Jawad, N.; Sabir, A. Gait recognition based on Kinect sensor. In Proceedings of the Real-Time Image and Video Processing 2014, Brussels, Belgium, 16–17 April 2014; Kehtarnavaz, N., Carlsohn, M.F., Eds.; SPIE: Washington, DC, USA, 2014.
11. Sattrupai, T.; Kusakunniran, W. Deep trajectory based gait recognition for human re-identification. In Proceedings of the TENCON 2018-2018 IEEE Region 10 Conference, Jeju, Korea, 28–31 October 2018.
12. Sun, J.; Wang, Y.; Li, J.; Wan, W.; Cheng, D.; Zhang, H. View-invariant gait recognition based on kinect skeleton feature. *Multimed. Tools Appl.* **2018**, *77*, 24909–24935. [CrossRef]
13. 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. (JOEUC)* **2020**, *32*, 67–92. [CrossRef]
14. Lee, C.P.; Tan, A.W.; Tan, S.C. Gait probability image: An information-theoretic model of gait representation. *J. Vis. Commun. Image Represent.* **2014**, *25*, 1489–1492. [CrossRef]
15. Lee, C.P.; Tan, A.W.; Tan, S.C. Time-sliced averaged motion history image for gait recognition. *J. Vis. Commun. Image Represent.* **2014**, *25*, 822–826. [CrossRef]
16. Mogan, J.N.; Lee, C.P.; Tan, A.W.C. Gait recognition using temporal gradient patterns. In Proceedings of the 2017 5th International Conference on Information and Communication Technology (ICoIC7), Melaka, Malaysia, 17–19 May 2017.
17. Mogan, J.N.; Lee, C.P.; Lim, K.M.; Tan, A.W.C. Gait recognition using binarized statistical image features and histograms of oriented gradients. In Proceedings of the 2017 International Conference on Robotics, Automation and Sciences (ICORAS), Melaka, Malaysia, 27–29 November 2017.
18. Rida, I. Towards human body-part learning for model-free gait recognition. *arXiv* **2019**, arXiv:1904.01620.
19. 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]
20. Okusa, K.; Kamakura, T. Fast gait parameter estimation for frontal view gait video data based on the model selection and parameter optimization approach. *IAENG Int. J. Appl. Math.* **2013**, *43*, 220–225.
21. Lee, C.P.; Tan, A.W.; Tan, S.C. Gait recognition via optimally interpolated deformable contours. *Pattern Recognit. Lett.* **2013**, *34*, 663–669. [CrossRef]
22. Lee, C.P.; Tan, A.; Lim, K. Review on vision-based gait recognition: Representations, classification schemes and datasets. *Am. J. Appl. Sci.* **2017**, *14*, 252–266. [CrossRef]
23. Lishani, A.; Boubchir, L.; Bouridane, A. Haralick features for GEI-based human gait recognition. In Proceedings of the 2014 26th International Conference on Microelectronics (ICM), Doha, Qatar, 14–17 December 2014.
24. Lee, C.P.; Tan, A.W.; Tan, S.C. Gait recognition with transient binary patterns. *J. Vis. Commun. Image Represent.* **2015**, *33*, 69–77. [CrossRef]
25. Alvarez, I.R.T.; Sahonero-Alvarez, G. Gait Recognition Based on Modified Gait Energy Image. In Proceedings of the 2018 IEEE Sciences and Humanities International Research Conference (SHIRCON), Lima, Peru, 20–22 November 2018; pp. 1–4.
26. Khan, M.H.; Farid, M.S.; Grzegorzek, M. Spatiotemporal features of human motion for gait recognition. *Signal Image Video Process.* **2019**, *13*, 369–377. [CrossRef]
27. Mogan, J.N.; Lee, C.P.; Lim, K.M. Gait recognition using histograms of temporal gradients. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2020; Volume 1502.
28. Yeoh, T.; Aguirre, H.E.; Tanaka, K. Clothing-invariant gait recognition using convolutional neural network. In Proceedings of the 2016 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Phuket, Thailand, 24–27 October 2016.

29. Shiraga, K.; Makihara, Y.; Muramatsu, D.; Echigo, T.; Yagi, Y. GEINet: View-invariant gait recognition using a convolutional neural network. In Proceedings of the 2016 International Conference on Biometrics (ICB), Halmstad, Sweden, 13–16 June 2016.
30. Alotaibi, M.; Mahmood, A. Improved gait recognition based on specialized deep convolutional neural network. *Comput. Vis. Image Underst.* **2017**, *164*, 103–110. [CrossRef]
31. Min, P.P.; Sayeed, S.; Ong, T.S. Gait recognition using deep convolutional features. In Proceedings of the 2019 7th International Conference on Information and Communication Technology (ICoICT), Kuala Lumpur, Malaysia, 24–26 July 2019; pp. 1–5.
32. Tong, S.; Ling, H.; Fu, Y.; Wang, D. Cross-view gait identification with embedded learning. In Proceedings of the Thematic Workshops of ACM Multimedia 2017, Mountain View, CA, USA, 23–27 October 2017; ACM Press: New York, NY, USA, 2017.
33. Tong, S.; Fu, Y.; Yue, X.; Ling, H. Multi-view gait recognition based on a spatial-temporal deep neural network. *IEEE Access* **2018**, *6*, 57583–57596. [CrossRef]
34. Aung, H.M.L.; Pluempitiwiriyawej, C. Gait biometric-based human recognition system using deep convolutional neural network in surveillance system. In Proceedings of the 2020 Asia Conference on Computers and Communications (ACCC), Singapore, 6 December 2020.
35. Zhu, X.; Yun, L.; Cheng, F.; Zhang, C. LFN: Based on the convolutional neural network of gait recognition method. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2020; Volume 1650, p. 032075.
36. Su, J.; Zhao, Y.; Li, X. Deep metric learning based on center-ranked loss for gait recognition. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020.
37. Balamurugan, S. Deep Features Based Multiview Gait Recognition. *Turkish J. Comput. Math. Educ. (TURCOMAT)* **2021**, *12*, 472–478.
38. Han, F.; Li, X.; Zhao, J.; Shen, F. A Unified Perspective of Classification-Based Loss and Distance-Based Loss for Cross-View Gait Recognition. *Pattern Recognit.* **2022**, *125*, 108519. [CrossRef]
39. Chao, H.; Wang, K.; He, Y.; Zhang, J.; Feng, J. GaitSet: Cross-view gait recognition through utilizing gait as a deep set. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3467–3478. [CrossRef]
40. Elharrouss, O.; Almaadeed, N.; Al-Maadeed, S.; Bouridane, A. Gait recognition for person re-identification. *J. Supercomput.* **2021**, *77*, 3653–3672. [CrossRef]
41. Arshad, H.; Khan, M.A.; Sharif, M.I.; Yasmin, M.; Tavares, J.M.R.; 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**, *39*, e12541. [CrossRef]
42. 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**, 1–21. [CrossRef]
43. Ambika, K.; Radhika, K. View Invariant Gait Authentication Using Transfer Learning. In Proceedings of the International Conference on Innovative Computing & Communication (ICICC), Delhi, India, 19–20 February 2021.
44. Mogan, J.N.; Lee, C.P.; Lim, K.M.; Anbananthen, K.S.M. Gait-DenseNet: A hybrid convolutional neural network for gait recognition. *IAENG Int. J. Comput. Sci.* **2022**, *49*, 393–400.
45. Han, J.; Bhanu, B. Individual recognition using gait energy image. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *28*, 316–322. [CrossRef]
46. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
47. Yu, S.; Tan, D.; Tan, T. A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006.
48. Makihara, Y.; Mannami, H.; Tsuji, A.; Hossain, M.A.; Sugiura, K.; Mori, A.; Yagi, Y. The OU-ISIR gait database comprising the treadmill dataset. *IPSJ Trans. Comput. Vis. Appl.* **2012**, *4*, 53–62. [CrossRef]
49. Iwama, H.; Okumura, M.; Makihara, Y.; Yagi, Y. The OU-ISIR gait database comprising the large population dataset and performance evaluation of gait recognition. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 1511–1521. [CrossRef]