Gait-DenseNet: A Hybrid Convolutional Neural Network for Gait Recognition

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

***Abstract*—Gait is the walking posture of a human, which involves movements of joints at upper limbs and lower limbs of the body. In gait recognition, the human appearance changes are taken into account, which makes it easier to differentiate every individual. However, covariates such as viewing angle, clothing and carrying condition act as the crucial factors that affect the gait recognition process. In this work, a hybrid model that integrates pre-trained DenseNet-201 and multilayer perceptron is presented. The method first extracts the gait energy image by windowing the gait binary images. Subsequently, transfer learning of the pre-trained DenseNet-201 model is leveraged to learn the representative features of the gait energy image. A multilayer perceptron is then used to further capture the relationships between these features. Finally, a classification layer assigns the features to the associated class label. The performance of the proposed method is evaluated on CASIA-B dataset, OU-ISIR D dataset and OU-ISIR Large Population dataset. The experimental results show significant improvements on all the datasets compared to the state-of-the-art methods.**

***Index Terms*—Convolutional neural network, DenseNet-201, Gait recognition, GEI, Multilayer perceptron**

I. Introduction

GAIT recognition is a biometric technology that can be used to monitor people without their cooperation. Unlike other biometric modalities such as fingerprints, iris and face, gait is obtainable at a distance, making it well-suited for security applications. Other than that, gait is hard to conceal and imitate, which results in wide utilization at airports and banks. Despite these perks, variations such as clothing, carrying condition as well as the camera viewpoints induce great changes in the human appearances. Hence, the gait recognition becomes highly challenging.

Over the years, various handcrafted techniques were proposed for gait recognition. The techniques can be classified as model-based and model-free approaches. Model-based approach [1, 2, 3, 4, 5] models the human body by using stick figures and sets of joints. The movement of the model is tracked based on the length of limbs and trajectories and angles between the joints. On the other hand, the model-free approach [6, 7, 8, 9, 10, 11, 12, 13, 14] extracts the gait features directly from the gait silhouettes. The model-free approach requires low computational cost as it was easy to implement. Although the handcrafted methods produce high accuracy, these methods only focused on extracting the manually defined features where the significant features could be neglected.

In recent times, the deep learning approach [15, 16, 17, 18, 19, 20, 21] has gained a lot of attention due to its ability to automatically learn high-level features. The deep learning approach extracts a large set of features, which contains intricate patterns. The extracted features aid in achieving higher accuracy compared to the handcrafted techniques. Currently, the researchers are focused on using a pre-trained model, which was learned from a large dataset to extract deep features. There are numerous well-known pre-trained models, namely VGG16 [22], AlexNet [23], Inception [24], and DenseNet [25]. The models are constructed based on CNN network. The usage of pre-trained models with transfer learning techniques has been achieving significant performance in the biometric domain. Nonetheless, the utilization of the pre-trained model in gait recognition is very limited.

Therefore, this paper proposes a hybrid deep learning model that integrates the pre-trained convolutional neural network and multilayer perceptron, referred to as the Gait-DenseNet model. Specifically, Gait-DenseNet model features a DenseNet-201 based transfer learning network, which is able to perform promisingly under different covariates. First, gait energy image (GEI) is acquired by averaging the silhouettes over a gait cycle. The pre-trained DenseNet-201 model is fine-tuned on the gait datasets to learn the features of the GEIs. Multilayer perceptron is then added to the network to obtain the class-specific representative features. Lastly, a classification layer is used to classify the features accordingly. The main contributions of this paper are:

- A hybrid convolutional neural network, known as the Gait-DenseNet model, that consolidates the strengths of pre-trained DenseNet-201 and multilayer perceptron for gait recognition.
- Fine tuning is performed on the pre-trained DenseNet-201 to better extract the salient gait related features.
- Integration of multilayer perceptron to encode the relationship between the extracted features and the corresponding walking subject.
- Optimization techniques, including batch normalization, dropout, leaky ReLU, Adam optimizer, and early stopping, are incorporated to optimize the performance of

Manuscript received December 13, 2021; revised March 15, 2022. 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, Internal Research Fund of Multimedia University MMUI/210112, and Yayasan Universiti Multimedia MMU/YUM/C/2019/YPS.

J. N. Mogan is a student in the Faculty of Information Science and Technology, Multimedia University, Melaka, 75450 Malaysia. (e-mail: 1121116804@student.mmu.edu.my)

C. P. Lee is a Senior Lecturer in the Faculty of Information Science and Technology, Multimedia University, Melaka, 75450 Malaysia. (e-mail: cplee@mmu.edu.my)

K. S. M. Anbananthen is an Associate Professor in the Faculty of Information Science and Technology, Multimedia University, Melaka, 75450 Malaysia. (e-mail: kalaiarasi@mmu.edu.my)

K. M. Lim, the corresponding author, is a Senior Lecturer in the Faculty of Information Science and Technology, Multimedia University, Melaka, 75450 Malaysia. (e-mail: kmlim@mmu.edu.my)

the proposed Gait-DenseNet model.
- Experiments were conducted on CASIA-B, OU-ISIR D and OU-LP datasets, to evaluate the scalability and generalization of the proposed Gait-DenseNet model.

II. Related Works

The handcrafted approach extracts manually specified features, while the deep learning approach learns the features automatically from the gait silhouettes. There are various techniques proposed under both approaches for gait recognition.

A. Handcrafted Approach

There are two broad categories of handcrafted approach, namely model-based methods and model free methods. Model-based methods mostly use a skeleton model to extract features such as length of limbs and angles between joints. In Zeng et al. [26], a 2D five-link biped model was utilized to extract lower limb joint angles. Radial Basis Function (RBF) was employed to identify the gait dynamics and the smallest error principle was used to classify the gait patterns. Similarly, Deng et al. [27] used a five-link biped model to capture the kinematic parameters. Deterministic learning was employed to generate the spatial-temporal features and kinematic features. Wang et al. [28] built a walking model where the length between joints were chosen as static features while the angles between skeletons as dynamic features. Recently, Sah and Panday [29] rotated the subjects in each frame by transforming the Kinect coordinates into Centre of Body (CoB) coordinates. By doing so, the dimension and positions of the subjects' body parts were captured in every frame. The CoB coordinates and the distance of the same joint of succeeding frames were considered as the features.

On the contrary, the model-free methods do not rely on any specific model to learn the gait features. Arora and Srivastava [30] proposed a technique termed as Gait Gaussian Image (GGI) to extract spatial and temporal features. The GGI was computed for each pixel of a frame over a gait cycle. In order to conduct the fuzzification, Gaussian function was applied on the acquired vector. Lee et al. [31] divided a gait cycle into several windows to generate an equal number of time-sliced averaged motion history image (TAMHI) composite images. Histograms of oriented gradients (HOG) of the obtained composite images were computed. Mogan et al. [9] incorporated motion history image (MHI), binarized statistical image features (BSIF) and histograms of oriented gradients (HOG) to capture the motion patterns and direction of a gait sequence. Rida [32] generated motion-based vectors by determining the horizontal motion of GEI images using Shannon entropy. Group fused lasso was employed to segment the body parts based on the shared change-point across the acquired motion vectors. Mogan et al. [33] presented a method to encode both spatial and temporal information. The gait images were convolved with Independent Component Analysis pre-learned filters where a set of feature maps were produced. The obtained feature maps were divided into several regions and the gradient of each pixel was computed. The gradient of each pixel was then concatenated into a histogram of temporal gradient.

B. Deep Learning

Deep learning approach performs both feature extraction and classification in a network. The deep learning approach learns the features automatically from the gait silhouettes without having to be manually specified. Most of the existing deep learning approach is based on convolutional neural networks (CNN) due to the ability to analyze the visual patterns in an image.

Shiraga et al. [34] presented a network for cross-view gait recognition, which comprises two sequential triplets of convolution layer, pooling layer and normalization layers. The network used GEIs as input and Softmax function to perform the classification. Alotaibi and Mahmood [35] developed a deep CNN with four convolutional layers and four pooling layers, to reduce the effects of variations and occlusions. The network performed well by using a small dataset without the need of data augmentation technique. Wu et al. [36] proposed three different architectures namely local@bottom (LB), mid-level@top (MT) and global@top (GT), which accept a pair of inputs. The difference between the three architectures is when the similarities between the pair is computed. The local features were compared at the bottom layer in the LB network, while the local features were compared at the mid-level layer in the MT network. As for the GT network, global features were compared at the top layer. Wang and Zhang [37] developed two different types of two-branch CNN networks. The difference among the networks is the position of the concatenation layer where the feature maps are fused in both the networks. Wen [38] applied a Gabor filter at the input layer to preprocess the gait silhouettes and extract the gait features. The classification was performed using a metric learning-based algorithm and Mahalanobis distance. Elharrouss et al. [39] constructed two CNN models to estimate the camera viewing angle and to classify the subject. The output of the estimated angle was fed to the second model to recognize the gait. Balamurugan [40] presented a deep CNN model, which consists of four convolution layers, four max-pooling layers, a fully connected layer and a Softmax layer.

Some of the deep learning-based work utilized a pre-trained network to extract features using transfer learning techniques. Li et al. [41] used pre-trained VGG-D model with no fine-tuning along with Joint Bayesian Model for view invariant gait recognition. Arshad et al. [42] proposed a gait recognition method consisting of two phases. The first phase is where the pre-trained VGG19 and AlexNet models without any fine-tuning extract the features. The obtained features were then fused together. During the second phase, entropy and skewness vectors were computed using the fused feature to identify the optimum sets of features. Liu and Liu [43] presented a two-stream network called mainstream network and auxiliary stream network. The mainstream network was developed based on DenseNet to extract the similarity of dynamic gait features. The auxiliary stream network was developed based on a stacked convolutional autoencoder to capture the similarity of static gait features.

III. Gait-DenseNet

In this work, a pre-trained DenseNet-201 model along with a multilayer perceptron is proposed for gait recognition. The proposed network is depicted in Fig. 1.

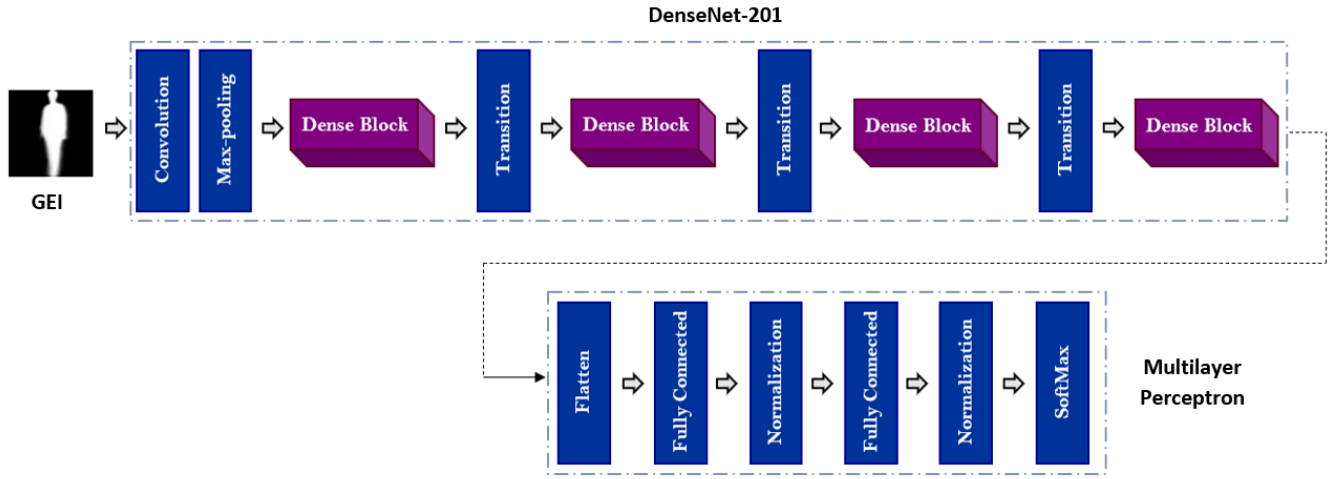

Fig. 1. The architecture of the proposed Gait-DenseNet model.

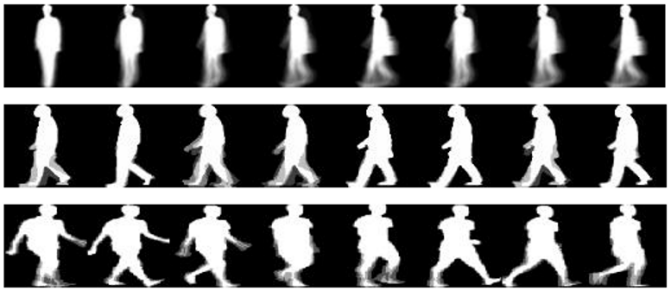

Fig. 2. Sample GEIs of CASIA-B (first row), OU-ISIR D (second row), and OU-LP (third row) Datasets.

A. Gait Energy Image

Gait Energy Image (GEI) [44] is the most widely adopted gait feature among the researchers. This is due to its ability to extract both static and dynamic information from a gait sequence. GEI is computed by simply averaging the frames over a gait cycle as stated below:

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

where $I_t(c,r)$ is the gait silhouette at time t and T is the total number of frames of a gait cycle. The obtained GEIs are then resized to 128×128 for all the datasets. Fig. 2 displays some examples of GEIs of three different datasets.

B. Network Architecture

The proposed Gait-DenseNet consists of a pre-trained DenseNet-201 model and a multilayer perceptron. The pre-trained DenseNet-201 model is employed to extract the deep gait features, while the multilayer perceptron further encodes the relationship between the learned features and the associated class.

1) Fine-tuning DenseNet-201: A basic CNN contains convolution layer, pooling layer, normalization layer, fully connected layer, and a classifier layer. The input is convolved with a number of kernels, which extract the gait patterns and produce a number of feature maps. In the pooling layer, the dimension of the obtained feature map is reduced without affecting the features. The normalization layer ensures to sustain the contribution of every feature in order to have an impartial network. The connection among the features and classes are extracted in the fully connected layer. An activation function is applied on both the convolution layer and the fully connected layer. The activation function decides whether to activate a neuron based on the weight and bias value. Lastly, classification is conducted in the classifier layer.

Transfer learning technique is using a pre-trained model trained on a large dataset and transferring the knowledge to solve a downstream task. Hence, the network is trained by fine-tuning the model. During fine-tuning, all the layers or a part of the pre-trained model are unfrozen. Several dense layers along with an output layer are added to the network according to the problem to be solved. As the pre-trained model is quite large, the whole model is trained using a low learning rate to avoid overfitting issues. In this work, the fine-tuning technique is applied on a pre-trained DenseNet-201 model. The model was trained on ImageNet dataset. The model consists of a convolution layer, a max-pooling layer, four dense blocks and three transition layers. In the case of connectivity of the layers, the prior layers in the model are directly connected to all the subsequent layers. The model concatenates all the feature maps of former layers with the latter layers, which strengthens the information flow among the layers. The feature concatenation of the layers is defined as:

$$f_l = H_l\left([f_0, f_1, \ldots, f_{l-1}]\right) \tag{2}$$

where l is the layer and $[f_0, f_1, ..., f_{l-1}]$ is the concatenation of features. $H_l(.)$ is a composite function, which consists of batch normalization, ReLU and a convolution of 3×3 operation. To ease the implementation, several inputs of $l(.)$ were concatenated into a single tensor. The dense blocks were created to enable the down-sampling of the layers. In each dense block, a bottleneck layer with 1×1 convolution was added before the 3×3 convolution to form BN-ReLU-Conv(1×1)-BN-ReLU-Conv(3×3). In doing so, the number of feature maps was reduced, thus the network requires lower computational cost. The transition layers were added in between the dense block, which performs 1×1 convolution followed by 2×2 average pooling. This is to

reduce the dimension of the feature maps into half of the original dimension. Growth rate of the network determines the contribution of every new information to the collective feature maps of every dense block. It is proven that the network performed well by using a small growth rate. This is due to the concatenation process of all the feature maps from the former layers. In other words, every layer receives concatenated feature maps from all the previous layers, which are connected to the newly produced feature maps.

The DenseNet-201 model is selected due to its ability to decrease the vanishing gradient problem. Other than that, the model involves a smaller number of parameters with the prospect of feature reuse.

2) Multilayer Perceptron: The output of the DenseNet-201 model is flattened into a vector and fed into the multilayer perceptron. The multilayer perceptron consists of two fully connected layers with 512 neurons. The fully connected layers determine the relationship between the extracted features and the classes. Batch normalization layers are added after each of the fully connected layers. The purpose of batch normalization layers is to normalize the outputs from fully connected layers in batches so that every neuron has a standard distribution across the batch. The batch normalization technique is done in batches to speed up the training process, which results in an efficient learning.

Leaky rectified linear unit (Leaky ReLU) function is employed as the activation function in both the fully connected layers. Leaky ReLU is a modified version of the ReLU function where it has the ability to map the negative values, which makes the layer more optimized. The Leaky ReLU function is stated as:

$$f(x) = \left\{ \begin{array}{ll} x & x > 0 \\ \alpha x & x \leqslant 0 \end{array} \right\} \tag{3}$$

where α is the value to be multiplied with x, which gives an output even when x is a negative value. By this adjustment, the neurons in the negative regions are activated and become functional.

Apart from that, the dropout technique is applied in both the fully connected layers to prevent overfitting issues. The dropout technique randomly drops certain neurons during the training where its contribution is not counted during forward propagation and the weights of the neurons are not updated during the backpropagation. Since gait recognition is a multiclass problem, a classifier layer with Softmax function is added to classify the subjects. The Softmax function provides probabilities of the input belonging to a specific class. Softmax function is defined as:

$$S\left(y_i\right) = \frac{\exp\left(y_i\right)}{\sum_{j=1}^{n} \exp\left(y_j\right)} \tag{4}$$

where n is the number of classes and y_i is the i-th input vector y to the Softmax function.

In the training process, the Adam optimizer is adopted to accelerate the network convergence. Adam optimizer is a combination of RMSprop and AdaGrad, thus inheriting the benefits of smoothening effect and noise reduction. Early stopping mechanism is used to avoid over-training the network. The early stopping mechanism stops training the network once the performance has stopped improving based on the validation set accuracy. As this work entails multiclass classification, categorical cross entropy loss function is used to compute the loss, which is defined as:

$$\text{loss} = -\sum_{i=1}^{n} \hat{y}_i \cdot \log y_i \tag{5}$$

where $\hat{y}_i$ is the true class label, y_i is the Softmax activation for class i, and n is the number of scalar values in the output. Table I illustrates the layer-wise structure of the proposed Gait-DenseNet model.

TABLE I
LAYER-WISE ARCHITECTURE OF THE PROPOSED GAIT-DENSENET MODEL

Model	Layers	Configurations
Pre-trained DenseNet-201	Convolution	7×7 conv, stride = 2
	Max-Pooling	3×3, stride = 2
	Dense Block	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
	Transition	1×1 conv
		2×2 average pool, stride = 2
	Dense Block	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
	Transition	1×1 conv
		2×2 average pool, stride = 2
	Dense Block	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
	Transition	1×1 conv
		2×2 average pool, stride = 2
	Dense Block	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Multilayer Perceptron	Fully Connected	512
	Batch Normalization	-
	Leaky ReLU	-
	Dropout	0.3
	Fully Connected	512
	Batch Normalization	-
	Leaky ReLU	-
	Dropout	0.3
	SoftMax	-

IV. EXPERIMENTS AND DISCUSSIONS

This section describes the datasets used in the performance evaluation, namely CASIA-B dataset, OU-ISIR dataset D, and OU-LP dataset. The hyperparameter tuning is discussed and the optimal values are determined empirically. The performance comparison and analysis with the existing gait recognition methods is also presented.

A. Datasets

The CASIA-B dataset [45] consists of 124 subjects. It is known as a large multi-view gait database. The gait sequences were recorded based on three different variations, such as view angle, clothing and carrying condition. The CASIA-B dataset allows assessing the methods under different covariates.

The OU-ISIR dataset D [46] comprises 370 gait sequences of 185 individuals captured from lateral view. The dataset was sampled based on how the gait fluctuates over time in a gait sequence. The dataset was divided into two subsets, namely DB_{low} and DB_{high}. Both the subsets consist of 100 individuals with steady walking style (DB_{high}) and fluctuated walking style (DB_{low}). The dataset permits to examine the proposed method under fluctuated walking style.

The OU-LP dataset [47] contains 4016 subjects with ages of 1 to 94 years old. The dataset was divided into two subsets: Sequence A (two sequences per subject) and Sequence B (one sequence per subject). The subsets were further broken down into five subsets based on the observation angles namely, 55°, 65°, 75°, 85° and all four angles. In this work, Sequence A with 3916 subjects is used in the experiments. By using the OU-LP dataset, the generalization of the proposed method is evaluated.

B. Hyperparameter Tuning

Six hyperparameters are involved in the proposed method, namely input size N, optimizer θ, dropout rate P, activation function A, batch size B and learning rate L. A grid search is performed on CASIA-B Dataset to tune the hyperparameters of the proposed Gait-DenseNet model. In every analysis, the value of a certain hyperparameter is changed, while the values of other hyperparameters remain the same.

TABLE II
ACCURACY AT DIFFERENT INPUT SIZES N [θ = ADAM, P = 0.3, A = LEAKY RELU, B = 32, L = 0.0001]

Input Size	Accuracy (%)
32 × 32	99.34
64 × 64	99.56
128 × 128	**100**

TABLE III
ACCURACY AT DIFFERENT OPTIMIZERS θ [$N = 128 \times 128$, P = 0.3, A = LEAKY RELU, B = 32, L = 0.0001]

Optimizer	Accuracy (%)
SGD	92.49
Adam	**100**

TABLE IV
ACCURACY AT DIFFERENT DROPOUT RATES P [$N = 128 \times 128$, θ = ADAM, A = LEAKY RELU, B = 32, L = 0.0001]

Dropout Rate	Accuracy (%)
0.2	99.71
0.3	**100**
0.4	99.85

Table II displays the accuracy of the Gait-DenseNet with different input sizes N. It is observed that the accuracy increases along with the input size. The highest accuracy is achieved when the input size is 128×128. This is due to the bigger input size containing more features than the smaller input size.

TABLE V
ACCURACY AT DIFFERENT ACTIVATION FUNCTIONS A [$N = 128 \times 128$, θ = ADAM, P = 0.3 , B = 32, L = 0.0001]

Activation Function	Accuracy (%)
ReLU	99.93
Leaky ReLU	**100**

TABLE VI
ACCURACY AT DIFFERENT BATCH SIZES B [$N = 128 \times 128$, θ = ADAM, P = 0.3 , A = LEAKY RELU, L = 0.0001]

Batch Size	Accuracy (%)
32	**100**
64	99.56
128	99.34

TABLE VII
ACCURACY AT DIFFERENT LEARNING RATES L [$N = 128 \times 128$, θ = ADAM, P = 0.3 , A = LEAKY RELU, B = 32]

Learning Rate	Accuracy (%)
0.01	91.83
0.001	99.63
0.0001	**100**

TABLE VIII
SUMMARY OF HYPERPARAMETER TUNING

Hyperparameters	Tested Values	Optimal Values
Input Sizes	32 × 32, 64 × 64, 128 × 128	128 × 128
Optimizers	SGD, Adam	Adam
Dropout Values	0.2, 0.3, 0.4	0.3
Activation Functions	ReLU, Leaky ReLU	Leaky ReLU
Batch Sizes	32, 64, 128	32
Learning Rates	0.01, 0.001, 0.0001	0.0001

The accuracy of the Gait-DenseNet model with various optimizers θ is shown in Table III. Adam optimizer is more effective on noisy silhouettes compared to SGD optimizer. As CASIA-B Dataset consists of noisy silhouettes, Adam yields higher accuracy than SGD. Other than that, Adam optimizer also converges faster than SGD, which consumes less computation time.

Table IV illustrates the accuracy of the Gait-DenseNet model with different dropout rates P. The experimental results show that the highest accuracy is attained at value 0.3. The dropout rate is to decide on how many neurons to be deactivated where the weights are neglected during the training. The larger value causes the network to overfit, while the smaller value causes the network to underfit.

Table V presents the accuracy of the Gait-DenseNet model using different activation functions A. Leaky ReLU function achieved higher accuracy than ReLU function. This is due to the slight improvement on the negative values in the Leaky ReLU function, where the negative values are activated and

provide an output, unlike the ReLU function where the negative values are deactivated.

The accuracy of the Gait-DenseNet model with various batch sizes B is displayed in Table VI. The highest accuracy is obtained when the batch size is set to 32. The larger batch size requires higher computational cost. Moreover, the accuracy of larger batch sizes is lower than the smaller batch sizes.

Table VII shows the accuracy with different learning rates L. The learning rate at 0.0001 attained the highest accuracy. As the proposed Gait-DenseNet involves a pre-trained network which is quite large, a smaller learning rate is more suitable in order to avoid overfitting problems.

The optimal value for every hyperparameter is selected based on the highest accuracy in every analysis. Table VIII shows the summary of the tested and optimal hyperparameter values for the proposed Gait-DenseNet model.

C. Comparison with the Existing Methods

Six existing methods were included in the experiments for comparison purposes, namely GEINet [34], Deep CNN [35], CNN with Leaky ReLU [48], CNN [49] and deep CNN [40]. In the experiments, all the datasets are divided into 80% training, 10% validation, and 10% testing. In order to have a fair comparison, the input size is set to 128 $\times$ 128 for all the existing methods. Table IX shows the comparison of accuracy among the proposed and existing methods on three datasets.

Due to the incomplete silhouettes in the CASIA-B dataset, the accuracy of most of the methods slightly dropped, especially the deep CNN [35] method. Nonetheless, the proposed Gait-DenseNet model outperforms the existing methods with an accuracy of 100%. As the Gait-DenseNet model is made up of fine-tuned DenseNet-201 and multilayer perceptron, the deep neural structure well maps the complicated patterns such as incomplete silhouettes and noisy silhouettes which resulted in the high accuracy.

Using the OU-ISIR dataset D with the DB_{high} and DB_{low} subsets of only 100 subjects, all the existing methods obtained promising results. The proposed Gait-DenseNet model achieved 100% accuracy in both the DB_{high} and DB_{low} datasets. The pre-trained model and multilayer perceptron are known to work well with both small and large datasets. Hence, the hybrid of the pre-trained DenseNet-201 and multilayer perceptron contributes to the high accuracy in all the datasets.

As for the OU-LP dataset with 3916 subjects, the accuracies of the CNN methods [40, 47, 35] are quite low due to the network being constructed for a small number of classes. Nevertheless, the proposed Gait-DenseNet model performed promisingly with an accuracy of 99.17%, which demonstrates the scalability and generalization ability of the proposed Gait-DenseNet. The fine-tuning of the pre-trained model, multilayer perceptron, batch normalization, Leaky ReLU activation function, early stopping, dropout layer, etc, collectively contributes to the outstanding performance in gait recognition.

V. Conclusion

Gait recognition has become a challenging task due to the covariates such as viewing angles, clothings and carrying conditions. In this work, a hybrid model that integrates the pre-trained DenseNet-201 and multilayer perceptron is proposed. The pre-trained DenseNet-201 model is fine-tuned to learn the salient gait features which produces the feature maps. A multilayer perceptron that consists of the fully connected layers, batch normalization layers and a classifier layer are appended to discover the relationship between the feature maps and the associated class for gait recognition. The experimental results demonstrate that the proposed Gait-DenseNet model is less sensitive to noise and incomplete silhouettes, fluctuations in walking patterns, and large number of subjects attributable to the deep structure of the hybrid model. Not only that, the enhancement techniques, namely batch normalization, dropout layer, and early stopping also help in reducing the overfitting and improving the generalization ability of the model.

References

[1] J. Sun, Y. Wang, J. Li, W. Wan, D. Cheng, and H. Zhang, "View-invariant gait recognition based on Kinect skeleton feature," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 24 909–24 935, 2018.

[2] H. Zhen, M. Deng, P. Lin, and C. Wang, "Human gait recognition based on deterministic learning and Kinect sensor," in *2018 Chinese Control And Decision Conference (CCDC)*. IEEE, 2018, pp. 1842–1847.

[3] T. Sattrupai and W. Kusakunniran, "Deep trajectory based gait recognition for human re-identification," in *TENCON 2018-2018 IEEE Region 10 Conference*. IEEE, 2018, pp. 1723–1726.

[4] S. Choi, J. Kim, W. Kim, and C. Kim, "Skeleton-based gait recognition via robust frame-level matching," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2577–2592, 2019.

[5] R. Liao, S. Yu, W. An, and Y. Huang, "A model-based gait recognition method with body pose and human prior knowledge," *Pattern Recognition*, vol. 98, p. 107069, 2020.

[6] C. P. Lee, A. W. Tan, and S. C. Tan, "Gait recognition via optimally interpolated deformable contours," *Pattern Recognition Letters*, vol. 34, no. 6, pp. 663–669, 2013.

[7] C. P. Lee, v. W. Tan, and S. C. Tan, "Gait probability image: An information-theoretic model of gait representation," *Journal of Visual Communication and Image Representation*, vol. 25, no. 6, pp. 1489–1492, 2014.

[8] C. P. Lee, A. W. Tan, and S. C. Tan, "Gait recognition with transient binary patterns," *Journal of Visual Communication and Image Representation*, vol. 33, pp. 69–77, 2015.

[9] Mogan, Jashila Nair and Lee, Chin Poo and Tan, Alan WC, "Gait recognition using temporal gradient patterns," in *2017 5th International Conference on Information and Communication Technology (ICoIC7)*. IEEE, 2017, pp. 1–4.

[10] M. H. Khan, M. S. Farid, and M. Grzegorzek, "Spatiotemporal features of human motion for gait recognition," *Signal, Image and Video Processing*, vol. 13, no. 2, pp. 369–377, 2019.

[11] M. Sharif, M. Attique, M. Z. Tahir, M. Yasmim, T. Saba, and U. J. Tanik, "A machine learning method

TABLE IX
PERFORMANCE COMPARISON ON GAIT DATASETS

Methods	Accuracy (%)			
	CASIA-B	OU-ISIR DB$_{high}$	OU-ISIR DB$_{low}$	OU-LP
GEINet [34]	95.66	99.86	99.72	88.34
Deep CNN [35]	54.82	96.73	97.15	12.02
CNN with Leaky ReLU [48]	98.75	99.86	99.65	88.88
CNN [49]	92.94	99.10	97.85	0.0005
Deep CNN [40]	91.17	98.26	97.98	55.40
Gait-DenseNet	**100**	**100**	**100**	**99.17**

with threshold based parallel feature fusion and feature selection for automated gait recognition," *Journal of Organizational and End User Computing (JOEUC)*, vol. 32, no. 2, pp. 67–92, 2020.

[12] K. Okusa and T. Kamakura, "Gait parameter and speed estimation from the frontal view gait video data based on the gait motion and spatial modeling." *International Journal of Applied Mathematics*, vol. 43, no. 1, pp. 37–44, 2013.

[13] K. Okusa and T. Kamakura, "Fast gait parameter estimation for frontal view gait video data based on the model selection and parameter optimization approach," *IAENG International Journal of Applied Mathematics*, vol. 43, no. 4, pp. 220–225, 2013.

[14] K. Okusa and T. Kamakura, "Human gait modeling and statistical registration for the frontal view gait data with application to the normal/abnormal gait analysis," in *IAENG Transactions on Engineering Technologies*. Springer, 2014, pp. 525–539.

[15] H. Wu, J. Weng, X. Chen, and W. Lu, "Feedback weight convolutional neural network for gait recognition," *Journal of Visual Communication and Image Representation*, vol. 55, pp. 424–432, 2018.

[16] C. Song, Y. Huang, Y. Huang, N. Jia, and L. Wang, "GaitNet: An end-to-end network for gait based human identification," *Pattern Recognition*, vol. 96, p. 106988, 2019.

[17] X. Li, Y. Makihara, C. Xu, Y. Yagi, and M. Ren, "Joint intensity transformer network for gait recognition robust against clothing and carrying status," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 12, pp. 3102–3115, 2019.

[18] C. Fan, Y. Peng, C. Cao, X. Liu, S. Hou, J. Chi, Y. Huang, Q. Li, and Z. He, "GaitPart: Temporal part-based model for gait recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 225–14 233.

[19] Y. Liu, Y. Zeng, J. Pu, H. Shan, P. He, and J. Zhang, "SelfGait: A spatiotemporal representation learning method for self-supervised gait recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 2570–2574.

[20] A. Sepas-Moghaddam and A. Etemad, "View-invariant gait recognition with attentive recurrent learning of partial representations," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 1, pp. 124–137, 2020.

[21] B. Lin, S. Zhang, and X. Yu, "Gait recognition via effective global-local feature representation and local temporal aggregation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 648–14 656.

[22] A. Vedaldi and A. Zisserman, "VGG convolutional neural networks practical," *Department of Engineering Science, University of Oxford*, vol. 66, 2016.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Cmputer Vision and Pattern Recognition*, 2015, pp. 1–9.

[25] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Cmputer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[26] W. Zeng, C. Wang, and Y. Li, "Model-based human gait recognition via deterministic learning," *Cognitive Computation*, vol. 6, no. 2, pp. 218–229, 2014.

[27] M. Deng, C. Wang, F. Cheng, and W. Zeng, "Fusion of spatial-temporal and kinematic features for gait recognition with deterministic learning," *Pattern Recognition*, vol. 67, pp. 186–200, 2017.

[28] Y. Wang, J. Sun, J. Li, and D. Zhao, "Gait recognition based on 3d skeleton joints captured by Kinect," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3151–3155.

[29] S. Sah and S. P. Panday, "Model based gait recognition using weighted KNN," in *Proceedings of 8th IOE Graduate Conference*. IOE, 2020, pp. 1019–1026.

[30] P. Arora and S. Srivastava, "Gait recognition using gait Gaussian image," in *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 2015, pp. 791–794.

[31] C. P. Lee, A. W. Tan, and S. C. Tan, "Time-sliced averaged motion history image for gait recognition," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 822–826, 2014.

[32] I. Rida, "Towards human body-part learning for model-free gait recognition," *arXiv preprint arXiv:1904.01620*, 2019.

[33] J. N. Mogan, C. P. Lee, and K. M. Lim, "Gait recognition using histograms of temporal gradients," in *Journal*

of Physics: Conference Series, vol. 1502, no. 1. IOP Publishing, 2020, p. 012051.

[34] K. Shiraga, Y. Makihara, D. Muramatsu, T. Echigo, and Y. Yagi, "GEINet: View-invariant gait recognition using a convolutional neural network," in *2016 International Conference on Biometrics (ICB)*. IEEE, 2016, pp. 1–8.

[35] M. Alotaibi and A. Mahmood, "Improved gait recognition based on specialized deep convolutional neural network," *Computer Vision and Image Understanding*, vol. 164, pp. 103–110, 2017.

[36] Z. Wu, Y. Huang, L. Wang, X. Wang, and T. Tan, "A comprehensive study on cross-view gait based human identification with deep CNNs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, pp. 209–226, 2016.

[37] X. Wang and J. Zhang, "Gait feature extraction and gait classification using two-branch CNN," *Multimedia Tools and Applications*, vol. 79, no. 3, pp. 2917–2930, 2020.

[38] J. Wen, "Gait recognition based on GF-CNN and metric learning," *Journal of Information Processing Systems*, vol. 16, no. 5, pp. 1105–1112, 2020.

[39] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, and A. Bouridane, "Gait recognition for person re-identification," *The Journal of Supercomputing*, vol. 77, no. 4, pp. 3653–3672, 2021.

[40] S. Balamurugan, "Deep features based multiview gait recognition," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 10, pp. 472–478, 2021.

[41] C. Li, X. Min, S. Sun, W. Lin, and Z. Tang, "DeepGait: a learning deep convolutional representation for view-invariant gait recognition using joint bayesian," *Applied Sciences*, vol. 7, no. 3, p. 210, 2017.

[42] H. Arshad, M. A. Khan, M. I. Sharif, M. Yasmin, J. M. R. Tavares, Y. D. Zhang, and S. C. Satapathy, "A multilevel paradigm for deep convolutional neural network features selection with an application to human gait recognition," *Expert Systems*, p. e12541, 2020.

[43] X. Liu and J. Liu, "Gait recognition method of underground coal mine personnel based on densely connected convolution network and stacked convolutional autoencoder," *Entropy*, vol. 22, no. 6, p. 695, 2020.

[44] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 316–322, 2005.

[45] S. Yu, D. Tan, and T. Tan, "A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 4. IEEE, 2006, pp. 441–444.

[46] Y. Makihara, H. Mannami, A. Tsuji, M. A. Hossain, K. Sugiura, A. Mori, and Y. Yagi, "The OU-ISIR gait database comprising the treadmill dataset," *IPSJ Transactions on Computer Vision and Applications*, vol. 4, pp. 53–62, 2012.

[47] H. Iwama, M. Okumura, Y. Makihara, and Y. Yagi, "The OU-ISIR gait database comprising the large population dataset and performance evaluation of gait recognition," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1511–1521, 2012.

[48] P. P. Min, S. Sayeed, and T. S. Ong, "Gait recognition using deep convolutional features," in *2019 7th International Conference on Information and Communication Technology (ICoICT)*. IEEE, 2019, pp. 1–5.

[49] H. M. L. Aung and C. Pluempitiwiriyawej, "Gait biometric-based human recognition system using deep convolutional neural network in surveillance system," in *2020 Asia Conference on Computers and Communications (ACCC)*. IEEE, 2020, pp. 47–51.