CHAPTER 3

Surface Covaraite database development

3.1 Introduction

Gait recognition remains a formidable challenge due to its susceptibility to various factors, including attire, footwear, and walking pace. In this chapter, a small gait dataset designed to assess the efficacy of gait recognition algorithms under varying surface conditions is introduced.

The vision-based and sensor-based approaches are widely used for gait data collection. The vision-based method involves using a camera to capture image frames from video sequences, while the sensor-based method uses sensing devices such as floor or wearable sensors to collect signals.

In terms of data collection, vision-based methods are considered more holistic and cost-effective for real-time implementation, as they allow more detailed and comprehensive capture of gait data, including body posture and movements, which is useful for identifying individuals based on their gait patterns.

A vision-based approach for human identification in varying surface conditions through gait involves using image processing techniques to extract gait features from video footage of individuals walking on different surfaces. The work in this chapter aims to analyze the accuracy of human gait pattern representation for identifying individuals based on their gait patterns under different surface conditions.

One of the challenges of the vision-based approach is dealing with the variation in gait patterns due to different surface conditions. For example, Walking on grass can present unique challenges compared to walking on a smooth, flat surface. To address this, the classification algorithms may be trained on data collected from multiple surface conditions. Also, it is observed in many of the recent works on gait recognition that most of the gait datasets have unique characteristics like gait data in multiview, gait data with static and dynamic occlusion, gait data with varying speed, and gait patterns with appearance change from clothing and carrying conditions. Therefore, considering that human walking movement is not confined to a specific area or walking style, a dataset with more walking conditions in the natural environment is needed to improve the accuracy and robustness of various existing gait recognition algorithms.

3.2 Methodology

In this chapter, we introduce the gait dataset, which has been meticulously crafted to represent real-world scenarios, taking into consideration various natural environments and their corresponding walking surface conditions. This dataset encompasses gait sequences captured from 50 individuals walking on three distinct surfaces: concrete, grass, and slopes. Each subject's movements were recorded from two distinct viewing angles: 90 degrees and 45 degrees. The dataset was acquired using two smartphone cameras boasting 48 and 64 megapixels, respectively.

After the data are recorded and collected, several steps are involved including video acquisition, pre-processing, gait feature extraction, and classification. During video acquisition, cameras are placed in strategic locations to capture the walking motion of individuals on different surfaces. The video footage is then pre-processed to remove any background noise and enhance the visibility of the motion.

Within the scope of this research work, performance evaluation of a cutting-edge CNN model for gait recognition and present baseline results using the proposed dataset. Furthermore, our work introduces an optimized convolutional neural network (CNN) model by fine-tuning the hyperparameters of an enhanced LeNet architecture designed for gait recognition. Our experimental findings indicate that the proposed CNN model performs comparably to the existing state-of-the-art models on a relevant database. The subsequent subsections delve into the step-by-step process for creating the compact gait dataset mentioned within this chapter.

3.2.1 Dataset development

The characteristics of publicly available gait datasets are unique in terms of human walking movement. It is observed that each dataset studied in table 1.2 to carry this research work mentioned challenges in gait data collection and the factors that influence the gait recognition system performance. Consequently, despite large-scale benchmark datasets, robustness for real-time implementation remains a challenge. In the current state-of-the-art surveillance system, CCTV cameras are placed in various locations without human intervention and allow data to be captured independently. As a result, in order to provide meaningful information, such data must be processed and stored. Gait is one type of data that can be used effectively in surveillance systems for human monitoring. Because each person has a unique walking pattern, the gait patterns of different people are distinct. One of the issues in gait identification performance is the inclusion of objects and the walking environment, in addition to the person's walking patterns. To address these issues, we created a new gait dataset from a complex environment. The new dataset introduced in this research has a unique characteristic in comparison with other benchmark gait datasets in two cases (i) data collection is performed in a natural walking environment, and (ii) the captured data includes overlapping objects in the background. The study also investigated a background segmentation technique for removing dynamic backgrounds from human walking movements.

3.2.2 Data Acquisition

The data collection procedure was challenging. One of the challenges in the data collection procedure required participants to understand the environment in which the subject moved. This implies that the participants need to comprehend and adapt to the specific conditions, scenarios, or contexts and to walk naturally. This includes awareness of the varying walking surface conditions such as concrete, grass, and slope surface types. Figure 3.1a shows the setup for the data acquisition. Thus 50 willing participants were identified from our university campus, comprising 31 males and 19 females between the age group of 23 to 40 years. Each individual was made to walk normally in three environmental setups with different background settings:(i) a concrete surface with a color strip background, (ii) a grass surface with swaying trees, and poles, and (iii) a slope surface with overlapping objects. Each background has its own complexities for all covariate conditions. A multicolored stripe wall ran across the background of the concrete surface. Similarly, in the case of grass, the background includes birds and swaying trees, whereas in the case of stair/slope walking, the background includes various types of vehicles, trees, and other overlapping objects. Figure 3.2 depicts the different types of walking-environment setups for data collection. In addition, two viewing angles of 90° and 45° were set around the subjects, as shown in figure 3.1b. Every individual walked three times in the setup environment. An individual's walking patterns were captured using two smartphone

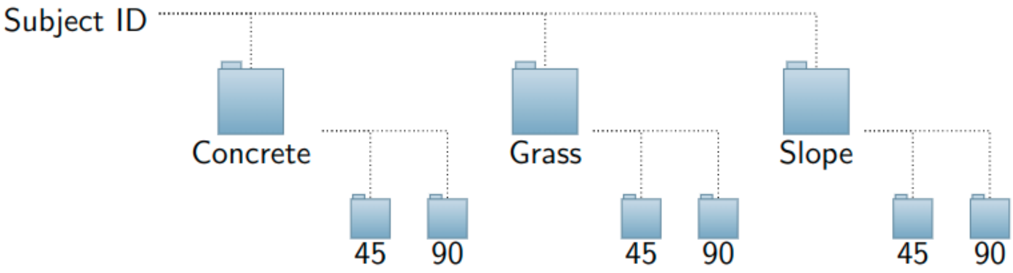
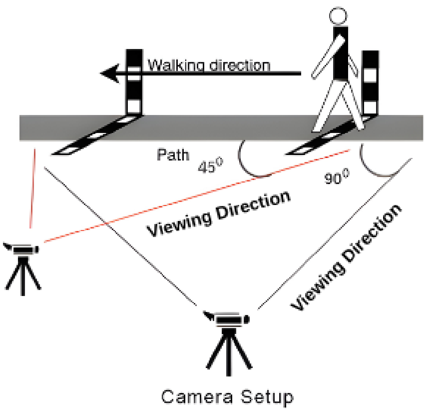


Figure 3.3: File arrangement steps

cameras with 48 and 64 megapixels. The captured data were in a video sequence and saved in mjpeg-encoded video files. The resultant dataset consisted of 50 ×3 ×2 =300 video sequences of 50 subjects in three different environments with two different view angles.



(a) Data acquisition setup

(b) Subject walking in two view angles

Figure 3.1: Setup for dataset

(a) Concrete surface with color strips background

(b) Grass surface with tree swaying and poles

(c) Slope surface with overlapping objects in the background

Figure 3.2: Walking environment setup with different backgrounds

3.2.3 Dataset creation

To start building the dataset, all video files are reorganized into folders based on subject ID, Covariate conditions, and Angles. Figure 3.3 shows the folder naming and arrangement. Subsequently, each video file was run using a pre-trained model for image segmentation and pose extraction. This study applies two pre-trained models on the concerned dataset: i) a Deep Convolutional neural network (DCNN) [16] model for efficient background subtraction in a complex background, and ii) a Blaze pose [9] model for keypoint extraction to address covariate conditions. It is observed that using a DCNN model for semantic segmentation with an image morphological operation yields better results for image segmentation from complex backdrops.

In addition, a human pose estimation model was used to address the covariate conditions in human walking. This pose estimation extracts body keypoints to create a unique gait feature that is independent of covariates. Upon processing the video files, we provide the label dataset in two formats: i) a total of 36235 silhouette images of size 640 X 480 pixel, the naming of the silhouette images is done in the given format: ID_SC_Ang_fn.png, where ID is the subject identification number(i.e S1, S2...S_n), SC indicates the walking surface covariates condition(i,e Concrete(C), Grass(G), Stair(S)), and 'Ang' is the angle varaiations(i.e 90 and 45), the "fn" is the frame number(i.e frame1, frame2...frame_n) respectively ii)a label. csv files of the extracted keypoints of each subject, irrespective of the covariate conditions.

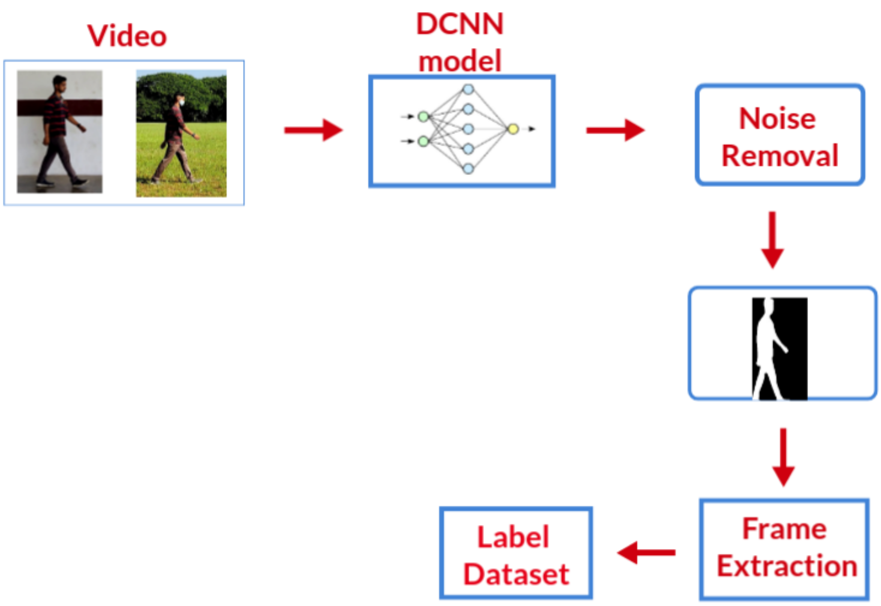


Figure 3.4: Dataset creation steps

3.2.4 Data preprocessing

The framework for the dataset creation is shown in figure3.4. The following subsections describe the theoretical concept of data preprocessing.

3.2.4.1 Background subtraction to extract silhouette

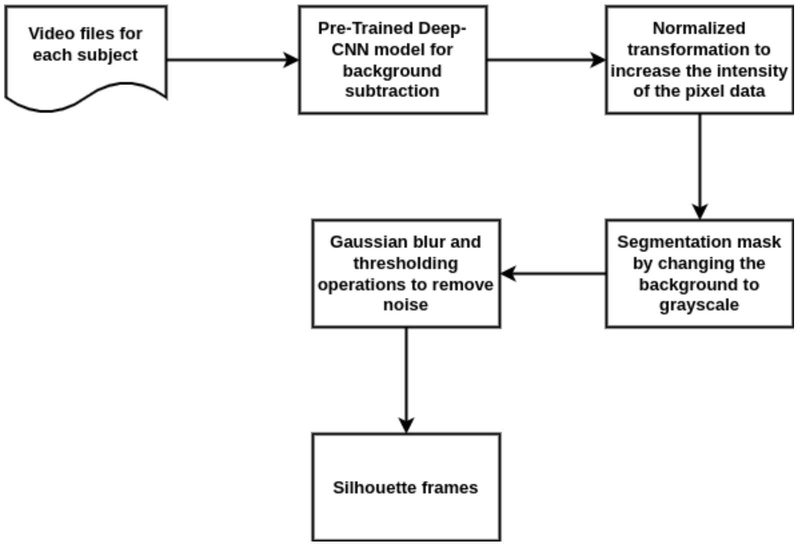


Figure 3.5: Steps for extracting silhouette frames

Background subtraction is a crucial step for eliminating unwanted information from a frame. In this case the complex background objects tend to confuse the machine learning model with making correct predictions. Background subtraction is used to detect moving objects in scene and extract the silhouette of the detected moving object[94]. To detect moving object from a video sequence, many frames must be extracted for comparison. Each frame of the video series was removed pixel-by-pixel from a planar background frame to obtain the silhouette. The background pixel is assigned a value “1” and the foreground pixel with value “0” for a given threshold value “T”. The silhouette image S is defined as:-

$$S = \begin{cases} 1, & \text{for } 0 \leq T \\ 0, & \text{for } T > 0 \end{cases} \tag{3.1}$$

The approach described above for background subtraction works well when the background models have constant and uniform patterns. Our proposed dataset includes environmental effects such as swaying trees, flying birds, and overlapping objects. As a result, we investigated existing background subtraction (BS) algorithms, such as MOG2[81],, KNN[108], and DCNN models[7]. The F1-Score and percentage of correct classification (PCC)of each BS algorithm were evaluated for the first three subjects on the proposed dataset. The F1-Score and percentage of correct classification (PCC) are the most widely used performance metrics for measuring a binary classifier's performance[24]. The evaluation of F1-score and PCC using the BS algorithms mentioned in the work are shown in Table 3.1,3.3, and 3.2. For performance analysis, a ground truth image is created manually using image editing tools against the concerned dataset. After the object is detected from the video sequence, ground truth is created by drawing bounding boxes around objects in an image and labeling them with the corresponding object class. To measure the effectiveness of the background subtraction algorithms, a performance analysis was performed to compare the output silhouette image with the ground truth image. The following metrics were computed:

Background subtraction true positive (BsTP): When the output pixel is correctly recognised as belonging to the ground truth pixel.

Background subtraction false positive (BsFP): The output pixel is classified as positive when it is not a part of the silhouette image.

Background subtraction false negative (BsFN): The ground truth pixel is classified as negative when it belongs to the silhouette, but the output pixel is not recognized as such.

Background subtraction true negative (BsTN):When a pixel does not belong to the silhouette and is recognised as such, it is classified as negative.

Accordingly F1-score and PCC are evaluated using the following formulas:

$$\text{F1\_Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.2}$$

$$\text{PCC} = \frac{\text{BsTP} + \text{BsTN}}{\text{BsTP} + \text{BsTN} + \text{BsFP} + \text{BsFN}} \tag{3.3}$$

Table 3.1: Evaluation of PCC and F1-Score using DCNN for Concrete, Grass and Stairs

DCNN MODEL									
Walking Surface	**Subjects**	**BsTP**	**BsFP**	**BsFN**	**BsTN**	**Precision**	**Recall**	**F-1 SCORE**	**PCC**
Concrete	1	14397	473	188	292142	0.9682	0.9871	0.9776	0.9978
	2	14027	579	99	292495	0.9604	0.9930	0.9764	0.9978
	3	14073	633	114	292380	0.9570	0.9920	0.9741	0.9976
Grass	1	5523	318	204	301155	0.9456	0.9644	0.9549	0.9983
	2	5323	303	167	301407	0.9461	0.9696	0.9577	0.9985
	3	5524	222	285	301169	0.9614	0.9509	0.9561	0.9983
Stairs	1	6118	692	96	300294	0.8984	0.9846	0.9395	0.9974
	2	7060	289	201	299650	0.9607	0.9723	0.9665	0.9984
	3	7213	472	152	299363	0.9386	0.9794	0.9585	0.9980

Where,

$$\text{Precision} = \frac{\text{BsTP}}{\text{BsTP} + \text{BsFP}} \tag{3.4}$$

$$\text{Recall} = \frac{\text{BsTP}}{\text{BsTP} + \text{BsFN}} \tag{3.5}$$

However, the results obtained from the aforementioned BS algorithms are still influenced by superimposed noise from the clutter backdrops. sample of the results obtained on the concerned dataset for a first subject is shown in Figure 3.6. The Mean F1- score and mean PCC evaluated for all video files of three subjects for three covariate conditions are shown in Table 3.4 and Table 3.5. The performance analysis with various BS algorithms is shown graphically in Figure 3.7 and Figure 3.8 respectively. The Analyses show that the DCNN model outperforms the others, with a mean F1-score of 95.62% and an average PCC value of 99.74% across all covariate conditions. As a result, on the proposed dataset, a semantic segmentation approach

Table 3.2: Evaluation of PCC and F1-Score using MOG2 for Concrete, Grass and Stairs

MOG2									
Walking Surface	**Subjects**	**BsTP**	**BsFP**	**BsFN**	**BsTN**	**Precision**	**Recall**	**F-1 SCORE**	**PCC**
Concrete	1	3909	2597	10676	290018	0.6008	0.2680	0.3707	0.9568
	2	3792	2067	10334	291007	0.6472	0.2684	0.3795	0.9596
	3	4416	2827	9771	290186	0.6097	0.3113	0.4121	0.9590
Grass	1	1988	1177	3739	300296	0.6281	0.3471	0.4471	0.9840
	2	2112	1145	3378	300565	0.6484	0.3847	0.4829	0.9853
	3	1797	1148	4012	300243	0.6102	0.3093	0.4106	0.9832
Stairs	1	1258	449	4956	300537	0.7370	0.2024	0.3176	0.9824
	2	1471	944	5790	298995	0.6091	0.2026	0.3041	0.9781
	3	1253	784	6112	299051	0.6151	0.1701	0.2665	0.9776

Table 3.3: Evaluation of PCC and F1-Score using KNN for Concrete, Grass and Stairs

KNN									
Walking Surface	**Subjects**	**BsTP**	**BsFP**	**BsFN**	**BsTN**	**Precision**	**Recall**	**F-1 SCORE**	**PCC**
Concrete	1	12034	11090	2551	281525	0.5204	0.8251	0.6383	0.9556
	2	12036	15355	2090	277719	0.4394	0.8520	0.5798	0.9432
	3	11491	11092	2696	281921	0.5088	0.8100	0.6250	0.9551
Grass	1	4888	3372	839	298101	0.5918	0.8535	0.6989	0.9863
	2	4855	4806	635	296904	0.5025	0.8843	0.6409	0.9823
	3	5048	4099	761	297292	0.5519	0.8690	0.6750	0.9842
Stairs	1	5926	8414	288	292572	0.4132	0.9537	0.5766	0.9717
	2	6528	6180	733	293759	0.5137	0.8990	0.6538	0.9775
	3	6530	3435	835	296400	0.6553	0.8866	0.7536	0.9861

Table 3.4: Mean F1 Score for various Backgroud substraction model

Mean F1- Score			
Walking Surface	**DCNN MODEL**	**KNN**	**MOG2**
CONCRETE	0.9760	0.61436	0.3874
GRASS	0.9562	0.6716	0.4468
STAIRS	0.9548	0.6613	0.2960

Table 3.5: Mean PCC for various Backgroud substraction model

Mean PCC			
Walking Surface	**DCNN MODEL**	**KNN**	**MOG2**
CONCRETE	0.9977	0.9513	0.9584
GRASS	0.9983	0.9615	0.9841
STAIRS	0.9979	0.9784	0.9793

pretrained DCNN model was applied.

3.2.4.2 Human Pose Estimation(HPE) model for keypoints extraction

Human pose estimation is a computer vision task that infers the pose of a person or an object in an image or a video. Human Pose Estimations (HPE), [3, 45] is a way to capture a set of coordinates for each joint (arm, head, hip, etc.,) which are known as a key point that can describe a pose of a person. Human pose estimation helps in detecting joints and body parts in a video sequence or in an image frame. The detected keypoints consist of hip, knee, and ankle joint rotations, mean hip, knee, and ankle joint angles, and thigh, trunk, and foot angles. These keypoints can be used in various applications such as health care, sports, and activity recognitions. In this research work, a blazepose human pose estimation model is employed to extract the body key points.

BlazePose is a lightweight convolutional neural network architecture designed for real-time inference on visual devices for human pose estimation [79, 9]. The network generates 33 landmark keypoints for a single individual during inference, as shown in Figure3.9.

Figure 3.6: Comparison of segmentation results of the three background subtraction models for subject one.(a) Original image,(b)Ground truth,(c) DCNN ,(d) MOG2,(e) KNN

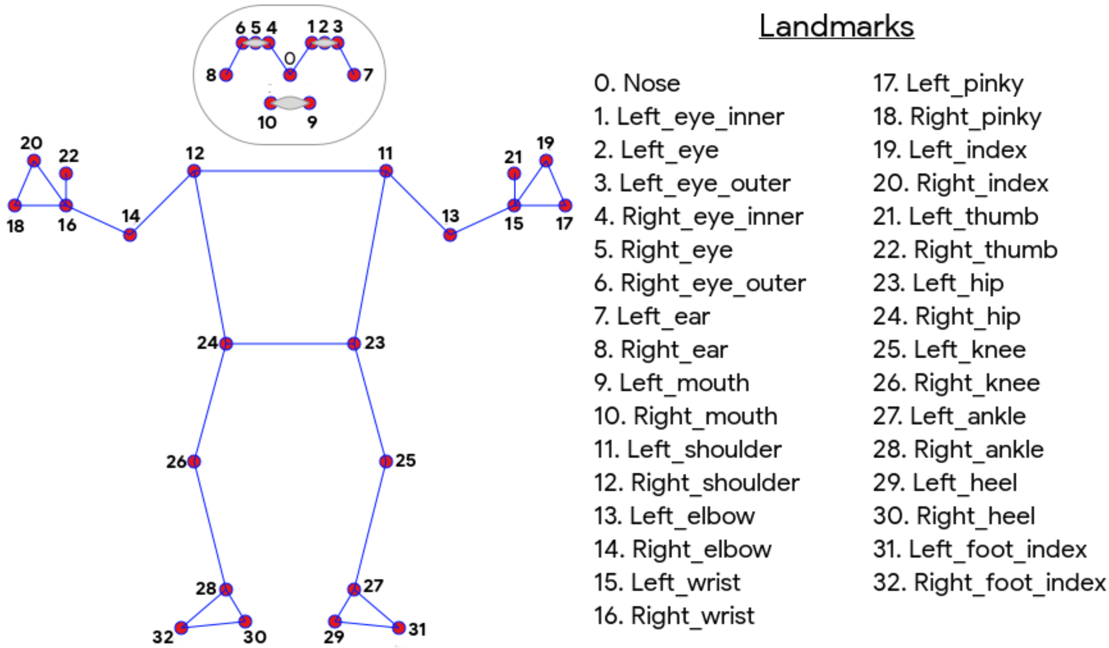


Figure 3.9: Blaze pose model and the landkmark[9]

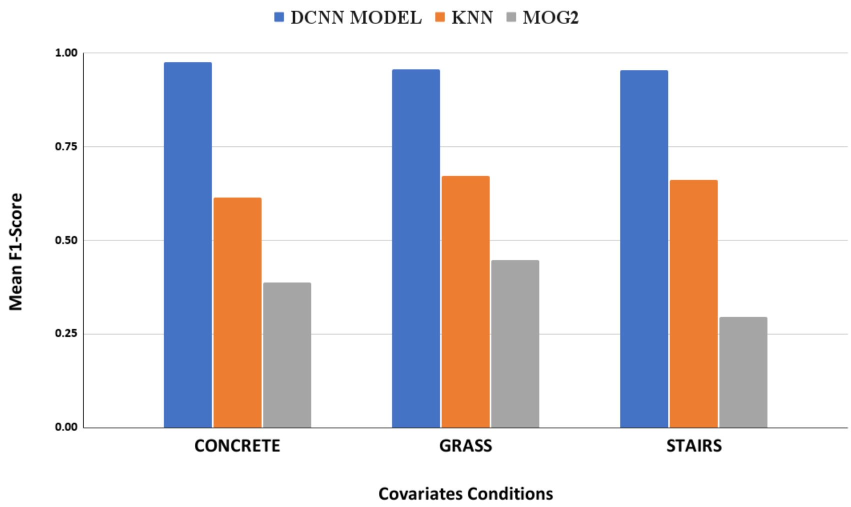


Figure 3.7: F1_Score of the three background subtraction models on three covariates conditions.

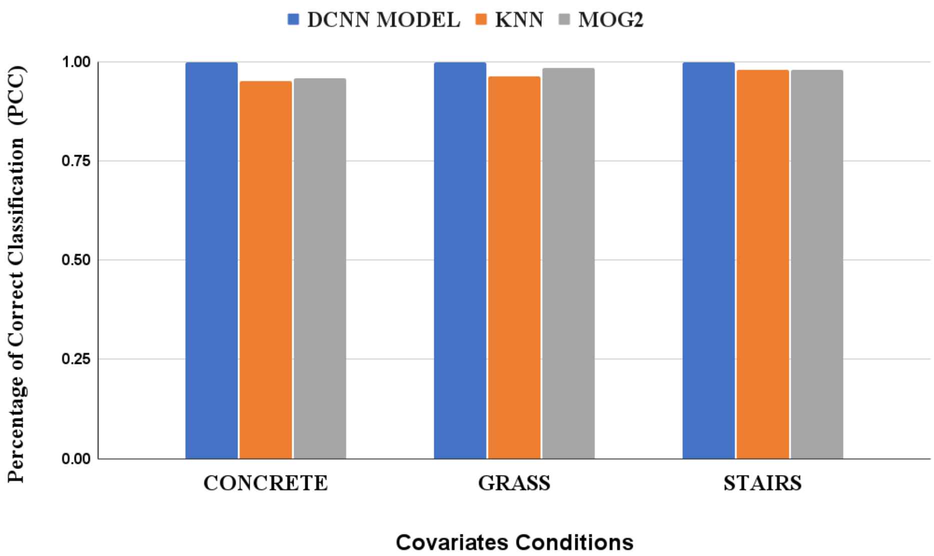


Figure 3.8: PCC of the three background subtraction models on three covariates conditions.

Blazepose is well suited for real-time applications such as fitness tracking and sign language recognition. The objective of employing a blazepose model is to detect more instances and can be a basis for detecting a human accurately from the video sequences, irrespective of covariate issues in gait analysis. Further, the extracted keypoints are utilized to determine the angle and distance of any desired points for subject identification. The steps for extracting the dynamic landmark keypoints using the Blaze pose-estimation model are shown in Figure3.10.

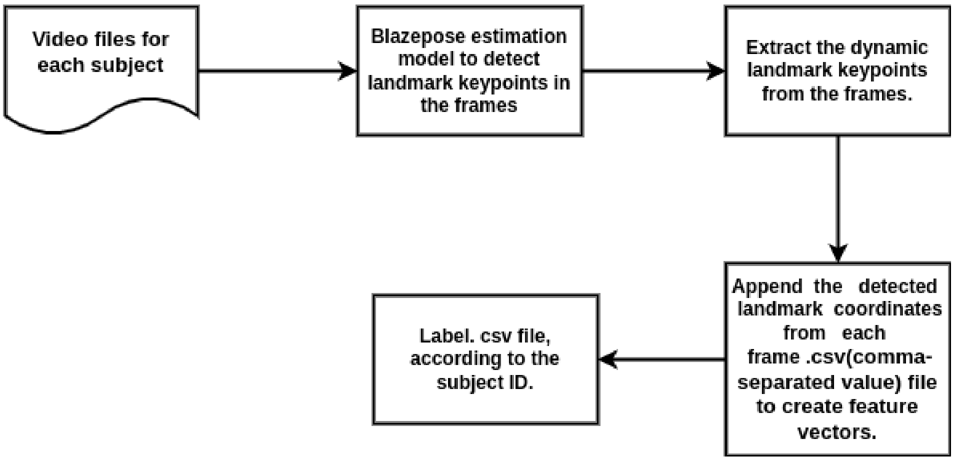


Figure 3.10: Steps for keypoints extraction to create csv file

(a) Original Image (b) Skeleton Feature Image

Figure 3.11: Skeleton image extraction from developed dataset

3.2.5 Performance analysis on developed gait dataset

Over the last few decades, researchers have studied gait analysis for human identification and classification using various techniques. The representation of gait features is based on static and dynamic image features, such as spatial features extracted from silhouettes of human walking in video sequences. One of the most popular types of gait analysis is the silhouette-based model gait analysis, with classification performed using various classifiers. Another recent and popular technique for dealing with covariate issues in gait recognition performance is gait analysis using HPE (human pose estimation). The use of Convolutional Neural network (CNN) models for dynamic feature extraction is the current state-of-the-art approach to feature extraction in gait-based human identification systems. In this work, we present a performance analysis of a silhouette based on an edge detection method using a convolutional neural network (CNN) model for subject classification. A CNN model is also proposed with a deeper model by optimally tuning the hyperparameter that sets the standard performance result against the dataset. For the performance analysis, a model is presented in Figure 3.12.

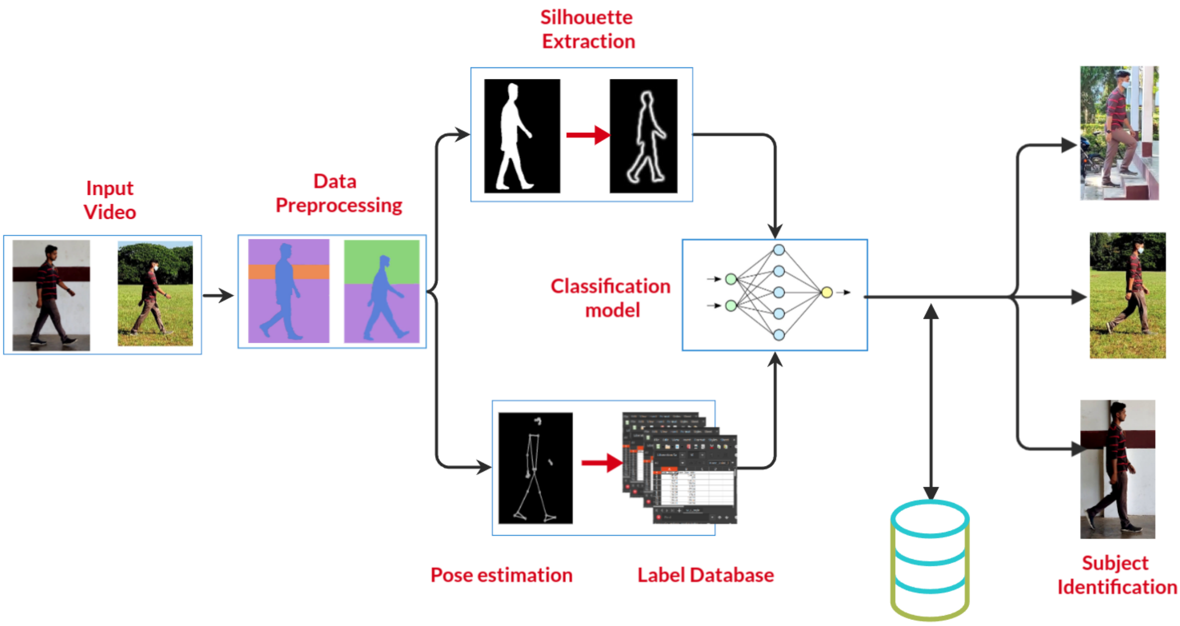
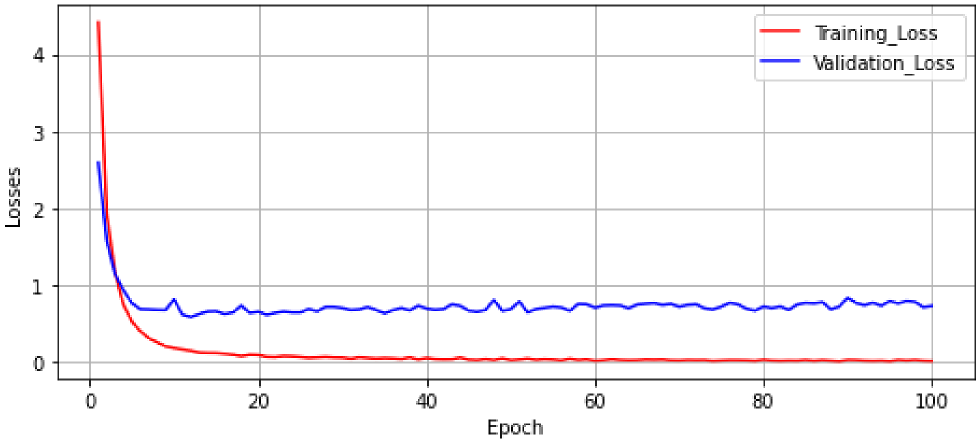


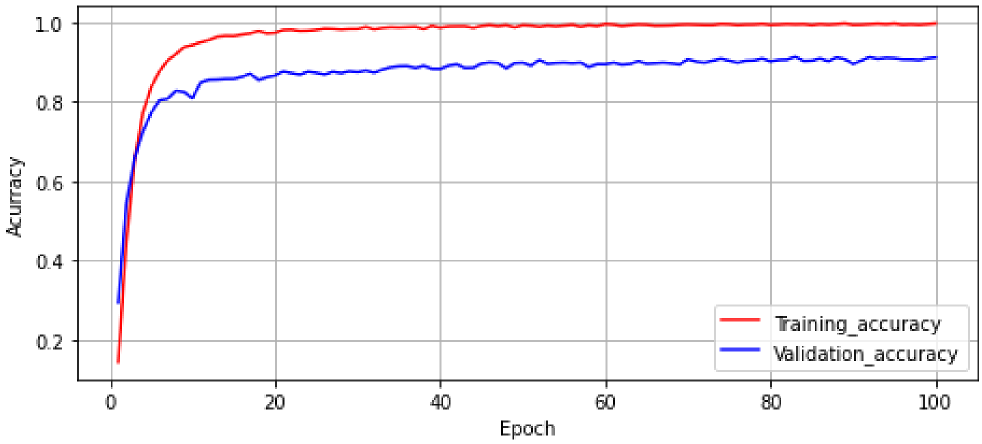
Figure 3.12: Model for performance analysis

3.2.5.1 Performance of the proposed dataset using Improved LeNet architecture

The LeNet network model is a simple and straightforward architecture that is used for image classification. The network model consists of three convolution layers with a 5×5 kernel mask with two max pooling layers of 2×2 window size and, stride equal to 2. It also has two fully connected layers(FC), with 1000 neurons in the first FC layer and 124 neurons in the second FC layer. For the proposed gait dataset, the number and step size of the kernel in the improved LeNet model remains the same. Only neurons in the fully connected layer were adjusted. The first FC layer was set to 1000 neurons, and the second FC layer was changed to 50 neurons. The CNN model was implemented using the Tensor-flow Keras API software library at the top level. A LeNet model, based on a gradient descent optimization algorithm with a mini-batch was used with a learning rate of 0.0001. A rectified linear unit (ReLu) activation function was used for all convolution layers. Rectified linear unit (ReLu) activation was employed in this study because it allows models to train faster and perform better by overcoming the problem of vanishing gradients. In larger network layers, activation functions, such as sigmoid and tangent activation functions, suffer from a vanishing gradient. A softmax activation function with a cross-entropy loss function is employed in the output layer. For the experimental analysis of the proposed dataset, we tuned the batch size and learning rate hyperparameters to train and test the dataset. Using the LeNet architecture with a mini-batch size of 32 and optimal learning of 0.0001, we achieved a training accuracy of 90.01% when the model was trained for 100 epochs. Figure3.13 shows the training and validation accuracy and losses obtained on the dataset using the improved LeNet model.



(a) Training and validation losses



(b) Training and validation accuracy

Figure 3.13: Performance analysis on Improved LeNet CNN model

3.2.5.2 Proposed CNN architecture on the proposed dataset

Convolutional neural networks (CNN), a dynamic feature extraction technique in which features are trained by the network itself, have gained popularity in gait recognition systems. Convolutional neural networks (CNN) are a type of neural network with three layers: a convolutional layer, a pooling layer, and a fully connected layer. The CNN architecture's main building block is the convolution layer. It comprises the majority of computing jobs in the network. This layer computes the dot product of two matrices, one of which is the set of trainable parameters known as a kernel and the other is the reduced section of the receptive field. In addition to the convolution layer, the pooling layer contributes to the management of the network architecture to reduce computational load and feature weights. A fully connected layer coordinates the feature map representation between the input and output layers. To improve the recognition accuracy, the CNN architecture formation is determined by the problem statement and the network architecture designer.

In this work, a CNN model is proposed with 14 layers apart from the input layers. There were five convolution layers, five max-pooling layers, and two fully connected, dropout, and output layers. A complete overview of the architecture is shown in Figure 3.14. The first convolution layer had a kernel mask of 5× 5, producing 32 feature maps. The max pooling layer had a window size of 2×2 pixels with a stride of 1. For the remaining convolution layers, 64, 128, 256, and 512 layers with a kernel mask of 3 × 3 were used for feature mapping. The model incorporates a ReLu activation function and two fully connected (dense layer) layers of 1024 and 512 nodes, respectively, with a dropout of 0.5 between the first and second fully connected layers. To prevent overfitting caused by large weights, an L2 kernel regularizer with a learning rate of 0.0001 was added to both dense layers. The output layer of the model employs a softmax function with a categorical cross-entropy loss function. For training and testing the proposed model, the Adam optimizer algorithm with a learning rate of 0.0001 was used with four different minibatch sizes of 8, 16, 32, and 64. For the performance analysis, a batch size of 32 with 100 epochs was used to train the new dataset. To train the network, the training and test sets were randomly divided into training (80%) and testing (20%) groups, respectively.

On the proposed CNN model, a silhouette-based method for edge detection in images is used. Edge detection is an important preprocessing step in many computer vision applications, including object recognition, tracking, and segmentation. Silhouette-based edge detection is a common method that involves extracting the boundary of an object from a binary image. The objective of using the edge detection method is to extract edge information from the image, which can help to identify the contours of objects and retain their characteristic features without losing temporal information. In a CNN model, the extracted edge information can be used as input to the network to improve the accuracy of the classification. The edge features are trained on a proposed CNN model for all covariate conditions. Using the proposed gait dataset, 36,300 silhouette images were extracted from 300 video sequences. These silhouette images were used for the performance analysis. On the extracted silhouette images, a Sobel edge detection operator was applied. Furthermore, the extracted edge features were hot-encoded and labeled for training in the CNN model. Using the proposed CNN model, a training accuracy of 99.56% was achieved, which sets the benchmark on the developed dataset. The training and validation accuracy and losses are shown in Figure 3.15 respectively. The experiment demonstrates that the proposed CNN model is well-suited for the specific dataset. The higher accuracy achieved by the CNN model suggests its potential for effectively handling variations in gait patterns due to covariate conditions. The training accuracy of 99.56% indicates that the model successfully learned the features of gait patterns in the training dataset.

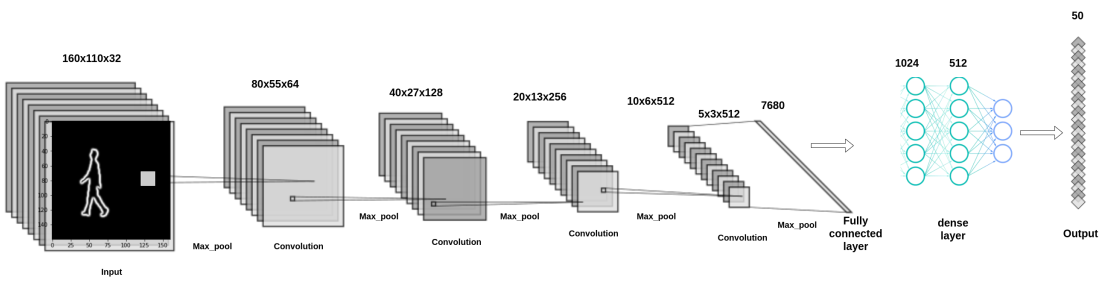
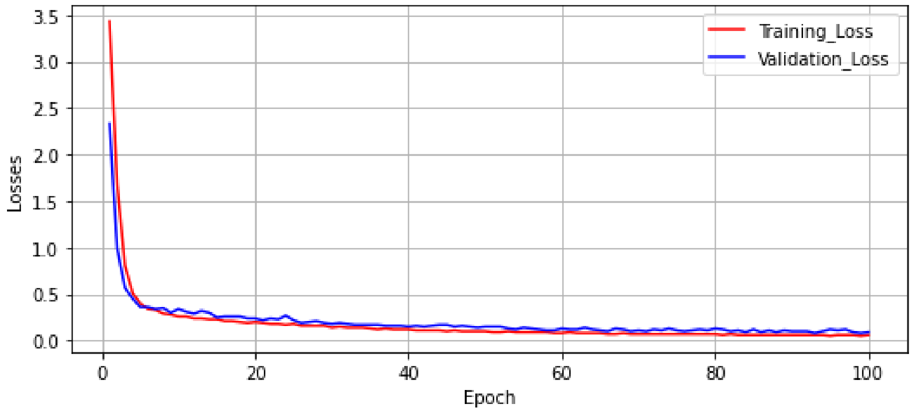


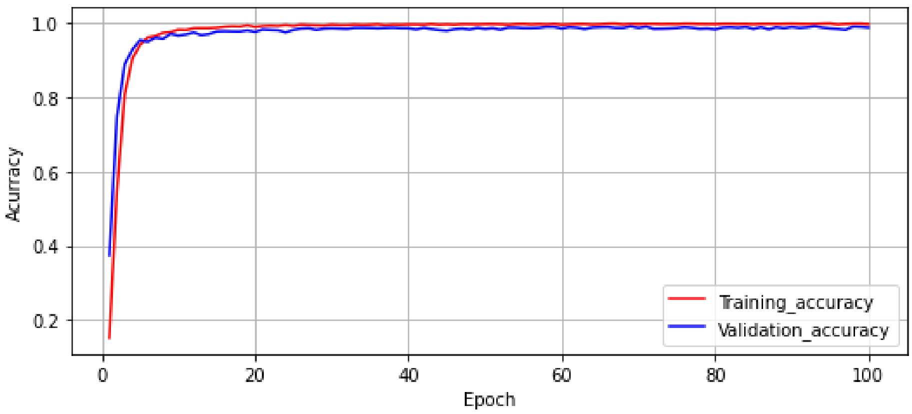
Figure 3.14: Proposed CNN architecture for known covariate classifications

3.2.6 Performance analysis using HPE on the developed dataset

In this chapter, gait analysis under covariate conditions is evaluated using two approaches: known covariate conditions and unknown covariate conditions. In the known covariate conditions approach, the models are trained and tested on the same covariate conditions. In contrast, the opposite of the trained and test processes is referred to as unknown covariate conditions [12]. Using human pose estimation (HPE) an experimental analysis was performed for unknown covariate conditions in gait recognition on the concerned dataset. A BlazePose pose estimation



(a) Training and validation losses



(b) Training and validation accuracy

Figure 3.15: Training and Validation analysis on the proposed CNN model

is implemented to extract the body keypoints for the subjects walking sequences from the input video files. Furthermore, the extracted keypoints were divided into two parts. The upper landmark keypoints are considered static keypoints, and the lower landmark keypoints are considered dynamic key points. From this division, we consider dynamic keypoints as a unique key feature to identify an individual.

First, a performance analysis of unknown covariates was performed using dynamic keypoints. These keypoints are saved as features and stored in. CSV file format for future studies. Test accuracy using these features with a few popular classifiers such as Random forest (RF), K-Nearest Neighbors (KNN), Support Vector Machine (SVM) are shown in table 3.6. For each classifier, the required parameters were selected and set. The experimental results show that the performance of the various classifiers is reasonably stable. The RF classifier with a maximum depth of 100 and entropy criterion achieved a higher test accuracy of 97.63%, whereas the KNN classifier with the minimum neighbor achieved a higher accuracy of 93.34%. For the SVM classifier, a test was performed using various kernel functions with different component values(c). It is observed that the radial basis function(RBF) kernel function with a higher component value yields a better test accuracy of 96.96%. Among the three classifiers used for the current investigation, the random forest classifier provided the highest accuracy for the unknown covariates issue on the concerned dataset.

Table 3.6: Test accuracy using dynamic pose features.

Classifier	Parameter	Test accuracy(%)
RF	criterion= "entropy", n_estimation =100 ,max_depth=50	97.49
	criterion= "entropy", n_estimation =100,max_depth=100	**97.63**
	criterion= "entropy", n_estimation =100,max_depth=200	97.43
KNN	p =1, weight= distance ,n_neighbor =1	**93.34**
	p =1, weight= distance,n_neighbor =3	90.97
	p =1,weight= distance, n_neighbor =5	88.75
	p =2,weight= distance,n_neighbor =1	87.41
	p =2,weight= distance ,n_neighbor =3	83.76
	p =2,weight= distance, n_neighbor =5	81.11
SVM	Kernel= "rbf",c= 10, gamma = 'scale'	79.20
	Kernel ="rbf",c=100, gamma = 'scale'	94.84
	Kernel = "rbf",c=1000, gamma = 'scale'	**96.96**
	Kernel= "linear",c= 10	56.10
	Kernel ="linear",c=100	61.66
	Kernel = "linear",c=1000	61.66
	Kernel= "poly",c= 10	63.14
	Kernel ="poly",c=100	81.00
	Kernel = "poly",c=1000,	89.39

3.3 Summary

We conclude that the small gait dataset proposed in this chapter is a valuable resource for evaluating the performance of gait recognition algorithms in varying surface conditions. The data collection procedure was based on a standard benchmark dataset that considered all aspects of human walking conditions. The dataset can also be used to develop new algorithms and techniques for gait recognition. We hope that this work will stimulate further research in this area and contribute to the development of robust gait recognition systems.

The next chapter of this thesis work discusses the use of CNNs for gait recognition. It emphasizes the need for robust recognition under different conditions and highlights the use of a feature fusion technique to enhance performance. The aim is to improve the effectiveness of gait recognition systems in real-world scenarios where covariate conditions may vary.