VIEW INVARIANT GAIT RECOGNITION FOR PERSON RE-IDENTIFICATION IN A MULTI SURVEILLANCE CAMERA ENVIRONMENT

By

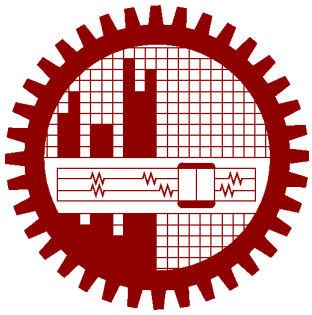
Md Mahedi Hasan

1014312019

MASTER OF SCIENCE

IN

INFORMATION AND COMMUNICATION TECHNOLOGY



Institute of Information and Communication Technology

Bangladesh University of Engineering and Technology

Dhaka, Bangladesh

February, 2020

This thesis titled, **"View Invariant Gait Recognition For Person Re-Identification in a Multi Surveillance Camera Environment"**, submitted by Md Mahedi Hasan, Roll No.:1014312019, Session: October 2014, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of MASTER OF SCIENCE in Information and Communication Technology on February, 2020.

BOARD OF EXAMINERS

Dr. Hossen Asiful Mustafa Assistant Professor IICT, BUET, Dhaka	Chairman (Supervisor)
Dr. Md. Saiful Islam Director and Professor IICT, BUET, Dhaka	Member (Ex-officio)
Dr. Md. Liakot Ali Professor IICT, BUET, Dhaka	Member
Name of the Supervisor Designation Address	Member
Name of the Supervisor Designation Address	Member (External)

CANDIDATE'S DECLARATION

This is to certify that the work presented in this thesis entitled "View Invariant Gait Recognition For Person Re-Identification in a Multi Surveillance Camera Environment", is the outcome of the research carried out by Md Mahedi Hasan under the supervision of Dr. Hossen Asiful Mustafa.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Md Mahedi Hasan
1014312019

Abstract

Person re-identification (re-id) across multiple cameras with non-overlapping fields of view is one of the most significant problems in computer vision and intelligent video surveillance system. However, most of the existing re-id algorithms are designed for closed-world scenarios that consider the same descriptors across the camera network regardless the dramatic change in view-angle due to different camera positions, eventually lead to them to perform poorly in real-world surveillance scenarios. In this thesis, we introduce an efficient gait-based person re-identification algorithm that addresses the challenges arise from real-world multi-camera surveillance environments. Again, recognizing individual people from their walking pattern or gait in an unconstrained environment is a challenging problem in computer vision research due to the presence of various covariate factors like varying view-angle, change in clothing, walking speed, and load carriage, etc. Most of the earlier works were based on human silhouettes which have proven to be efficient in recognition but are not invariant to change in illumination and clothing. In this research, to address this problem, we present a simple yet effective approach for robust gait recognition using a recurrent neural network (RNN). Our RNN network with GRU architecture is very powerful in capturing the temporal dynamics of the human body pose sequence and perform recognition. We also design a low-dimensional gait feature descriptor, which is discriminant, and at the same time robust to the different variations, by concatenating four different types of spatio-temporal feature vector each of which are extracted from the 2D coordinates of human pose information which is proven to be not only invariant to various covariate factors but also effective in representing the dynamics of various gait pattern. For multi-view gait recognition, we also propose a two-stage network in which we initially identify the walking direction by extracting the spatio-temporal features from gait video using a 3D convolution. The experimental results on challenging CASIA A and CASIA B gait datasets demonstrate that the proposed method has achieved state-of-the-art performance on both single-view and multi-view gait recognition which prove the effectiveness of our method.

Acknowledgement

First and foremost, I express my deepest gratitude to **Almighty Allah** for bestowing His blessings on me and giving me the ability to accomplish this work successfully.

I would like to express my deepest sense of thankfulness and gratitude to my thesis supervisor **Dr. Hossen Asiful Mustafa**, Assistant Professor, IICT, BUET for leading me into the research field of computer vision and deep learning. His scholarly guidance, constant and energetic supervision, and valuable advice made this work a successful one. He has been a continuous source of inspiration and a real motivating force throughout my research work. I am also extremely grateful to him for providing me a high-end GPU instance to accomplish this work.

I would like to thank my classmates **Md Abdul Aowal, Abu Noman**, and **Imran Khan** for their firm-backing and co-operation. I am truly grateful to my roommate cum brother **Abdullah Al Mahmud** for his endless support, and encouragement with my studies and works. He always tolerates my frustration. We studied together and shared a lot of discussions which were very helpful for this research.

Finally, I want to dedicate the essence of my purest respect to my parents and to my colleagues for providing me support throughout my years of study and through the process of writing this thesis. This accomplishment would not have been possible without them. Thank you.

Md Mahedi Hasan

Dhaka
February, 2020

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence. 1, 13

ANN Artificial Neural Networks. 16, 22

BGRU Bidirectional Gated Recurrent Unit. 23, 35, 36

BLSTM Bidirectional Long Short-Term Memory. 36

BN Batch Normalization. 36

BPTT Backpropagation Through Time. 17

BRNN Bidirectional Recurrent Neural Network. 23

CCR Correct Class Recognition. 44–48, 50

CNN Convolutional Neural Network. 10, 14, 15

DL Deep Learning. 13, 14

DNN Deep Neural Network. 14, 25

GRU Gated Recurrent Unit. 6, 21–23, 35, 36

LSTM Long Short-Term Memory. 18, 20–22, 36

ML Machine Learning. 13

MSE Mean Squared Error. 41

ReID Re-identification. 1, 2, 4

RNN Recurrent Neural Network. 5, 6, 14, 16–18, 22, 23, 35, 36, 38

SGD Stochastic Gradient Descent. 41

Chapter 1

Introduction

1.1 Person re-identification

In recent years, with the increasing demand for public safety and security, the network of video surveillance cameras are proliferating in both public and private areas. In order to reduce the number of crimes and terror attacks, and to provide a safer and secure environment, it is necessary to improve the current state of the surveillance system by implementing computer vision-based algorithms to automatically recognize a suspicious person. However, unfortunately, existing surveillance systems only have the capacity to capture and store the video leaving the task of detecting abnormal events in human operators. Therefore, there has been a great effort by the computer vision and artificial intelligence (AI) communities to develop an intelligent video surveillance system capable of real-time monitoring and alerting. Person re-identification (ReID), a fundamental task of intelligent video surveillance systems, refers to recognizing the same person across a network of cameras with non-overlapping fields of view from given single or multiple images. Besides security and surveillance, it is needed for a lot of applications such as authentication, human-computer interaction, cross-camera person tracking, human behavior and activity analysis.

A basic person re-identification scenario can be shown in Figure 1.1. In a surveillance system person ReID helps us to know when and where a person appears with respect to a given camera, and in a network of multiple cameras, it potentially allows us to estimate his/her trajectory over a short period of time.

However, person ReID remains an open problem to be addressed in real-time surveillance due to the large variation in camera view angle, pose and illumination variation, partially or complete occlusions, and subject intrinsic variations. However, among

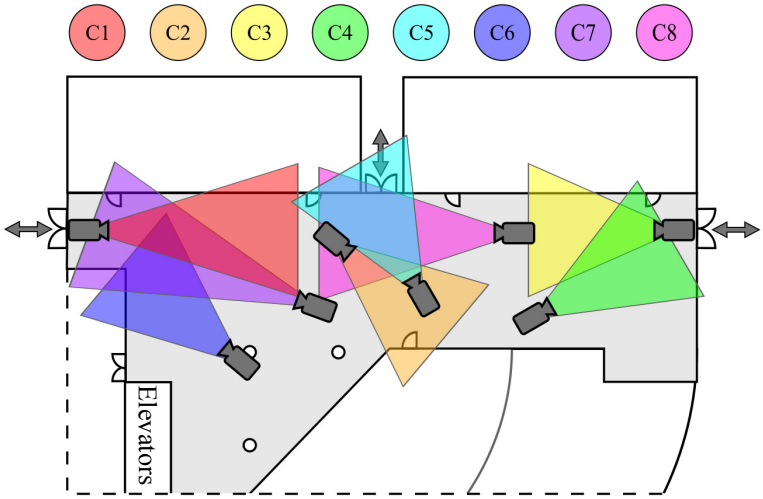


Figure 1.1: A basic topology of person re-identification in a multi-camera network environment. A person should have the same label when walking through the surveillance camera network.

these, the viewpoint variation is one of the most challenging problems which simultaneously creates the intra-class variation and the inter-class confusion.

1.2 Gait Recognition

1.2.1 Definition

Biometrics refers to the automatic identification or authentication of people by analyzing their physiological and behavioral characteristics. Physiological biometrics is related to the shape of body parts such as the face, fingerprints, shape of the hand, iris, retina, etc., which are not subject to change due to aging. It is now used as the most stable means for authenticating and identifying people in a reliable way. However, for efficient and accurate authentication, these traits require cooperation from the subject along with a comprehensive controlled environmental setup. Hence, these traits are not useful in surveillance systems. Behavioral biometrics such as signatures, gestures, gait, and voice, etc., is related to a persons behavior. But, these traits are more prone to change depending on factors such as aging, injuries, or even mood.

Gait can be defined as the coordinated, cyclic combination of movements that result in human locomotion [1]. The movements of gait are coordinated in the sense that they must occur with a specific temporal pattern and cyclic in nature since a walker cycles between steps with alternating feet. It is both the coordinated and cyclic nature of the motion that makes gait a unique phenomenon to each individual. Although these

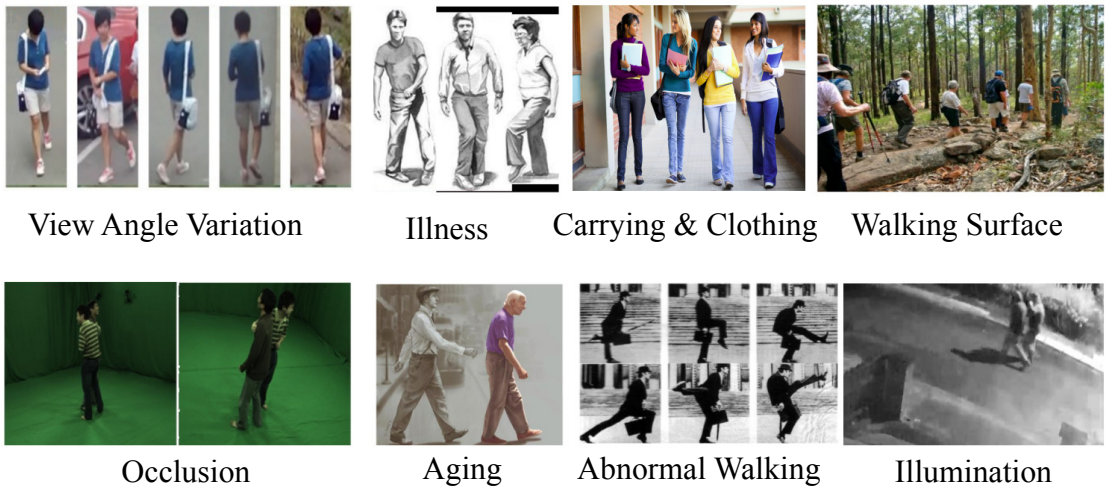


Figure 1.2: Common challenges in gait recognition

movements follow the same basic bipedal pattern for all humans, they seem to vary from one person to another considerably in their relative timing and magnitudes [2].

Gait recognition is a behavioral biometric modality that identifies a person based on his/her gait pattern. Among the behavioral biometrics cues, gait is very relevant to person ReID in surveillance networks. It is the most prevalent human movement in typical surveillance spaces.

1.2.2 Advantages

In contrast to other biometrics such as face and fingerprint, gait has several attractive properties. For example, it is a non-invasive technique for identifying an individual which is hard to copy. It doesn't require any cooperation or awareness of the subject. Another unique advantage of gait as a biometric is that it offers recognition at a greater distance with simple instrument. Additionally, recognition can also be done reliably at low-resolution images. Other biometrics may not provide the required accuracy under these conditions. Due to these potential advantages, gait has attracted significant attention in recent years. Consequently, gait biometric signature is now considered as the only likely identification technology suitable for access control, covert video surveillance, criminal investigation, and forensic analysis where the method is not vulnerable to spoofing attacks and signature forgery. In addition to the authentication applications, gait analysis can also be used in medical applications for abnormality detection, Parkinson's disease [3], and Chronic disease [4].

1.2.3 Challenges

It has been observed that some recent gait recognition algorithms have achieved 100% classification accuracy under controlled environmental setup. However, these results significantly degrade in real-world scenarios due to the presence of various challenging factors. Due to these covariate factors gait recognition in real-world multi-camera environment remains an open research problem which yet prevent it to be employed in public places for security surveillance-based applications.

Gait recognition is highly affected by the change in environment such as variations in view angle, illumination, and walking surface as well as different intraclass variations such as variation in clothing and carrying condition. Some of the challenging factors and their effects are explained in following.

- **View angle variation**: Since with the variation of camera position the distance between the subject and the camera as well as the walking direction of the subject are varying, different sizes or shapes of the subject can be observed in different viewing angles. Moreover, one view can contain partial information of subject's body pose. Even some the body parts of the subject are not visible in one view angle, but could be visible in another view angle. Furthermore, in same view angle different people's shape may look more similar compare to the shape of same subject under different view angle.
- **Partial occlusion**: Sometimes subjects are partially or completely occluded by overlapping with other people or objects.
- **Low resolution**: Surveillance cameras are usually installed in high places on walls which are far away from the subject. Therefore, the captured videos are in low resolution which often cause inaccurate body pose estimation.
- **Clothing or carrying variation**: The subject may appear with different clothing and carrying object in different camera views. For example, subject may wear a coat or take the backpack in hand from back.
- **Illumination variation**: In a network of multiple camera illumination condition can vary. Therefore, the same subject can have a color difference on the appearance under different lighting conditions.

Some of the examples images illustrating these challenges are shown in Figure 1.2. So, these challenges are needed to be addressed for robust gait recognition.

1.3 Problem Definition

Gait based person re-identification in surveillance is a problem of recognizing individuals based on their gait pattern at different times and locations from a network of interconnected cameras, without overlapping views. However, the presence of various covariate factors make the problem very challenging. Although a multitude of researches have been done in recent years, it remains an open problem and many of its aspects have yet to be addressed. This research aims to build a robust algorithm in order address these challenges. It also investigate ways to incorporate the multi-view data to solve the problem of view-dependency in multi-camera network.

There are fundamental differences between the gait-based recognition and person ReID in general scenarios. For general person ReID, the operator has the control over most of the acquisition factors, e.g., camera viewpoint, the number of persons in the image, chance of occlusion, subject pose, and illumination, etc. However, in person ReID under real-world surveillance most of these conditions are uncontrolled, e.g., changes in viewpoint and illumination over a large number of different cameras, no control on the number of people and possible occlusions, also subjects varying walking direction. Therefore, due to the unconstrained nature of the problem, gait recognition algorithms are now considered as a potential biometric tool to solve person ReID in real-world video-based surveillance applications [5].

1.4 Objectives of the Thesis

The objective of this thesis is to design a gait recognition system for person re-identification in a multi surveillance camera environment. To achieve this objective, we have identified the following specific aims.

- To design a novel low-dimensional gait feature descriptor based on the pose information of the people detected in the gait videos To design a mechanism to detect people in gait videos and determine their pose sequences.
- To develop a robust pose based gait recognition algorithm using recurrent neural network (RNN), which will be invariant to factors like viewing angle, clothing, presence of bags, etc.
- To identify people across a set of interconnected surveillance cameras.
- To compare the results with state-of-the-art methods.

1.5 Overview of the Thesis

Modern deep learning-based algorithms have recently gained increasing popularity while achieving outstanding performance in many computer vision tasks such as video classification [6], pose estimation [7], and action recognition [8, 9], etc. A recurrent neural network (RNNs), a type of artificial neural network, where connections between nodes form a directed graph along a temporal sequence. RNNs have also achieved a promising performance in many sequence labeling tasks. The reason behind their effectiveness for sequence-based tasks lies in their ability to capture long-range dependencies in a temporal context from a sequence. RNNs have been successfully employed to achieve state-of-the-art results in many vision-based tasks like image captioning [10] and action recognition [8, 9]. Furthermore, advancement on human body pose estimation can significantly assist in accurately modeling different human body parts required for gait recognition.

In this thesis, we propose a model-based gait recognition method where we consider human 2D pose information for our effective gait feature as human pose is proven not to be dependent on people's body appearance, and is invariant to change of clothing and carrying conditions. Additionally, as gait can be considered a time series of walking postures, body pose information has a powerful capacity to capture the temporal pattern of gait. Therefore, the proposed method will be less affected by the variation of covariate factors. It is also worth mentioning that, in this work, we didn't use 3D pose data as our gait feature: firstly, computing 3D poses is computationally expensive, and secondly, most of the 3D pose estimation algorithms recover 3D pose from 2D RGB images which often require multiple views, and hence multiple cameras, rendering the technique unsuitable for surveillance. Again, recovering 3D pose from a single RGB image is an ill-posed problem and often causes large pose estimation errors.

Compared to other gait covariates, view is the most important factor which severely affects gait recognition performance. To handle view variation efficiently, gait algorithms have generally been studied under three experimental setups: single-view, multi-view, and cross-view setup. In single-view gait recognition, both probe and gallery gaits are kept within same view angle, wherein cross-view gait recognition, the probe and gallery gaits are kept in different views; and in multi-view gait recognition, multiple views of gallery gaits are combined to recognize a probe gait under a specific view.

Thus, the main idea of our proposed method is to develop a pose-based recurrent neural network for robust gait recognition by modeling the temporal dynamics associated with human gait. Most of the descriptors proposed in the literature for gait recognition

often lead to a high dimensional feature space which are computationally expensive to map. In this research, we designed a lower-dimensional spatio-temporal feature descriptor from 2D pose estimation for improved performance at a reduced computational cost. Our gait descriptor is a concatenation of four different types of feature vector. For multi-view gait recognition, we also propose a two-stage network in which we first determine the walking direction, i.e., the view angle of the camera using a 3D convolutional network and later identify the subject using proposed RNN-based temporal network trained on that particular angle.

We demonstrate the effectiveness of our proposed method through extensive experiments on two public benchmark datasets: the CASIA A and CASIA B gait dataset [11]. Our method achieved state-of-the-art performance on these two challenging gait datasets in both single-view and cross-view recognition, providing better results as compared to other state-of-the-art methods. Besides that our method is far simpler and efficient in terms of time and space compared to other methods proposed in the literature. Again, in multi-view gait recognition, our method outperforms other architecture at a significant margin.

1.6 Contributions

Original contributions resulting from the research presented in this thesis are fourfold:

- We introduce a novel low-dimensional discriminative gait feature vector from 2D body pose information which is invariant to covariate factors and achieved comparable performance to the methods which require to calculate gait energy image (GEI) or expensive 3D poses for gait descriptors.
- We design a novel RNN network with GRU architecture and devise several strategies to effectively train the network for robust gait recognition.
- We also propose a two-stage network for multi-view gait recognition in which we first identify the walking direction using a 3D convolutional network and then performs subject recognition using a temporal network trained on that particular angle.
- The proposed pose-based RNN network achieves the best results on two challenging benchmark datasets CASIA A and CASIA B by outperforming other prevailing methods in single-view and multi-view gait recognition at a significant margin.

1.7 Thesis Outline

In the rest of this thesis, we present the details of our approach for robust gait recognition. Here

- **Chapter 2** provides a survey of existing gait recognition techniques and evaluates them for their strengths and limitations.
- **Chapter 3** describes the proposed framework used in this thesis with all required preprocessing, modeling and network architecture for both single-view and multi-view gait recognition. It also presents the training strategies of these models with complete implementation details.
- **Chapter 4** gives the experimental evaluation of the proposed framework on publicly available datasets, namely CASIA A and CASIA B dataset. It also compare our results with other state-of-the gait recognition algorithms in different experimental setup on these datasets and discusses them.
- **Chapter 5** concludes this work with a summary over the different contributions and presents some perspectives about possible future research directions.

Chapter 2

Literature Review

Over the last two decades, several methods have been studied to develop a robust gait recognition system [12]. In this chapter, we briefly discuss some of these techniques which can be divided into the following two main classes.

- **Sensor-based methods:** Wireless or wired sensors are attached to the joints of the subjects and the displacement are recorded with respect to a reference point. Joint angle trajectories and the distance between different parts of the body during gait is then calculated and used for recognition. Pressure plates or carpet is also used to measure the pressure profile of the feet during gait. These methods are more prevalent in medical research and rehabilitation studies. The experiments are usually done in the laboratory setup and therefore has limited scope.
- **Image-based methods:** A video is recorded as the person walks along a preset trajectory without having any sensors attached to any part of the bod. The data can be recorded both indoor and outdoor using single or multiple cameras. This type of methods have much wider scope and applications. Image-based methods can be further subdivided into two categories, i.e., marker-based and markerless methods.
  - **Marker-based methods:** Active or passive markers are placed on the body of the subject at different joints. This helps to detect and track the motion of desired joints in the video during gait motion. The subjects usually wear black and tight clothing and then reflective markers are placed on the joints.
  - **Markerless methods:** A video is recorded without using any markers and with normal clothing.

The work proposed in this thesis falls into image based markerless category. There are two main approaches to person gait recognition in markerless systems. In the first approach, known as appearance-based approaches, no a priori human geometric shape model is assumed. While in case of model based approaches, a priori geometric shape model is available.

We will present a brief review of the previous works in both categories. However, with time these traditional handcrafted appearance-based techniques, as discussed in Section 2.1, and model-based techniques, as discussed in Section 2.2, are being shifted toward automated learning methods. In recent years, deep learning-based algorithms, e.g CNN and RNN, have been applied to gait recognition. Section 2.3 discuss some these methods which achieved state-of-the-art results. And in Section 2.5, we review some of the recent pose-based gait recognition approaches which are closely related to our work.

2.1 Appearance-based Methods

The appearance-based gait recognition methods first perform motion detection to segment the regions corresponding to the moving humans. Some form of shape analysis is then applied to these human image sequences to extract the gait signatures. Static body parameters such as lengths and widths of limbs, height of the person are extracted in some techniques and used to represent gait. Some works rely on the dynamic features that are extracted by shape changes and motion flow. Most of the previous work following this approach [2, 13–16] used human silhouette masks as the main source of information and extracted features that show how these mask change.

BenAbdelkader *et. al.* [2] presented a parametric method for person identification based on the height and stride parameters of the gait from low resolution video sequences. A non-parametric background modeling approach was adopted for the segmentation of the moving objects. Foreground blobs were then tracked using spatial and temporal coherence. The height and stride parameters were determined from the extracted binary silhouettes. The experiments were performed on a database containing 45 subjects and an accuracy of 49% was achieved by using both the stride and height parameters and only 21% by using the stride parameter only. Although, they did not achieve a significant performance, yet their results show that stride and height parameters may be used as potential candidates for the gait recognition systems.

In [13], Liu and Sarkar computed the average silhouette during the whole gait sequence. Their algorithm consists of three steps. In the first step, the background pixel statistics

were calculated using the Mahalanobis distance and EM algorithm. The second step calculated the periodicity of the gait by simply counting the number of foreground pixels in the silhouette in each frame over time. The pixels belonging to the leg area were used to increase the sensitivity for determining the periodicity of the gait. In third and last step, the average silhouettes were computed. The similarity measure is defined as the negative of the median of the Euclidean distance between the averaged silhouettes from the probe and the gallery.

However, the most popular gait representation employed in appearance-based methods is gait energy image (GEI) [14], a binary mask computed through aligning and averaging the silhouettes over the complete gait cycle. Though there are many other alternatives for GEI, e.g., gait entropy image (GEnI) [15], gait flow image (GFI) [16], and Chrono-gait Image (CGI) [17], due to its in-sensitiveness of incidental silhouettes error, it has been considered as the most stable gait features. It can achieve good performance under controlled and cooperative environments, but does not show robustness when the view angle and clothing condition change.

In order to reduce drastic change of the shape of GEI, Huang *et al.* [18] fused two new gait representation: shifted energy image and the gait structural profile to increase the robustness to some classes of structural variations. But, the performance of this method is not good enough due to the loss of temporal information while calculating GEI.

To preserve temporal information, Wang *et. al.* [17] modified the gait energy image and constructed Chrono-gait image to include temporal information. After gait period detection, they used local information entropy to obtain the gait contour from the silhouette images. Synthetic chrono gait images were also constructed to avoid over fitting due to smaller number of real chrono gait images. LDA and PCA were applied for dimensionality reduction. A comprehensive experimental evaluation was reported using 3 major gait databases. An average CMS value of 48.64% and 66.81% was achieved at rank 1 and rank 5 respectively using all 12 probe sets of HumanID database [19]. These results did not show marked improvements over related gait energy image method and were only marginally higher.

Therefore, appearance-based methods are sensitive to the covariate factors as the extraction of human silhouettes is affected by the changes in lighting. Moreover, when the shape of the human body and appearance change substantially, the performance of these methods severely degrades. Therefore, these methods are not completely robust toward these covariate change.

2.2 Model-based Methods

Model-based methods [20–22] are often built with a structural and a motion model to capture both static as well as dynamic information of gait. The salient advantage of the these approach is that, in contrast to silhouette-based approaches, it can efficiently handle the covariate factors if the human bodies are correctly and high accurately modeled.

There have been a considerable amount of work on tracking human body based on the pose and body shape. However, these techniques have not caught much attention over the past years into the research community due to the fact that tracking human body is itself a challenging problem which involves very intensive computations. The geometrical model of human body is usually parameterized and tracking of the shape is achieved by establishing the correspondence between model configurations and image features. The most common methods for tracking include Kalman filter, dynamic Baysian network [23] and condensation algorithm [24].

The model-based approaches generally extract gait features from either static parameters or relative motion of joint angles. The static parameters of human body such as torso height, leg length and stride are calculated from fitting the model in each frame and then further analyzing it for feature extraction. The joint angle trajectories are calculated using some methods and gait features are extracted from them. These approaches can also be distinguished by the dimension of the shape model which could be 2D or planar or a 3D model. The following paragraphs elaborates some of the popular model-based methods for human gait recognition.

Wagg and Nixon developed a model-based method based on the biomechanical analysis of walking people and used it for recognition [22]. The image sequences were segmented to extract the moving regions and an articulated model is fitted to the edge by a hierarchical procedure. Motion estimation is performed by using a sinusoidal model for the leg and angle trajectories are extracted. The method is evaluated by using SOTON database [25] and the feature vector is 63 dimensional. A recognition rate of 84% on the indoor dataset and 64% for the outdoor dataset was achieved.

Yam *et al.* [20] developed an automated model-based approach to recognize the people using walking as well as running gait. They used a modeling technique based on the concept of coupled oscillators and the underlying principles of human locomotion. The two approaches derive a phase-weighted Fourier Descriptor (FD) gait signature by automated non-invasive means. Assuming the gait symmetry, the same model was used to describe either leg since both perform the same motion but out of phase with each other by half a period. These motions operate in space and time satisfying the rules of spatial

symmetry and temporal symmetry. This model of forced coupled oscillators is fitted to the image data extracting the lower leg motion in both walking and running gait. The gait features were derived from the magnitude and phase of FDs of thigh and lower leg rotation. A statistical analysis was also performed to find the most effective feature set.

A 3D human body model consisting of 11 body segments was developed by Gu *et. al.* [21]. The head was represented by a sphere and other segments were cylindrical. The model contains 10 joints with 24 Degrees Of Freedom (DOF). The kinematic structure of the model was estimated by employing anthropometric constraints between ratios of limb lengths. After the body segmentation, adaptive particle filter was used to track the body segments. Gait features were extracted from pose parameters and joint position sequences. Two gait models were obtained from normalized joint sequence of the whole body and the normalized joint sequence of two legs using an exemplar-based Hidden Markov Model (HMM). Maximum a Posteriori (MAP) estimation was used for pattern classification. The test database consisted of multiple video streams of 12 subjects that were simultaneously captured from multiple static calibrated cameras. Volumetric representation sequences were created using visual hull method after foreground extraction. An average recognition rate of 94.4% was reported on the test database.

So, model-based approaches are generally invariant to various intraclass variations like clothing, carrying and view angle variations, etc. However, the main drawback of this approach is the extraction process of body parameters like height, knee, and torso are computationally expensive and highly dependent on the quality of the video.

2.3 Deep Learning for Gait Recognition

Due to its powerful feature learning abilities, convolutional neural networks (CNNs) have achieved great success in object recognition task in recent years. In contrast to methods presented in the previous sections in which features are handcrafted, CNNs implement a data-driven approach to find the best feature extractors based on the training data. Several CNN-based gait recognition methods [26–31] have been proposed which can automatically learn robust gait features from the given training samples. Additionally, using CNNs, we now can execute feature extraction and perform recognition within a single framework.

Wu *et al.* [26] performed cross-view gait recognition by developing three convolutional layer network using the subject's GEI as input. Shiraga *et al.* [27] designed a eight-layered CNN network, GEINet, for cross-view gait recognition using GEI as input.

The network consist of two sequential triplets of convolution, pooling, normalization layers, and two fully connected layers. The network was evaluated under cross-view gait recognition setup using the OU-ISIR large population dataset.

In [28], Wolf *et al.* used 3D convolutions for multi-view gait recognition by capturing spatio-temporal features to find a general descriptor for human gait which is invariant to view angles, color and different walking conditions. In order to make the model color invariant they formulated special type of input having 3 channels where the first channel of the input image was the RGB-image converted to grey-scale and for second and third channel the optical flow in x and y were employed. The algorithm was evaluated on three different datasets namely the CMU Motion of Body (MoBo) [64], the USF HumanID gait dataset [19] and CASIA B [11].

A Siamese neural network-based gait recognition system has been developed in [29] where GEI was feed as input. In [30], Yu *et al.* used generative adversarial nets to design a feature extractor in order to learn the invariant features. In [31], they further improved the GAN-based method by adopting a multi-loss strategy to optimize the network to increase the inter-class distance and to reduce the intraclass distance at the same time.

2.4 Pose Estimation

In recent years, there has been a huge interest in the study of deep learning-based approaches for the task of real-time pose estimation from image and video. [7, 32].

Authors in [32] introduced Convolutional Pose Machines (CPMs) for the task of articulated pose estimation. It consists of a sequence of convolutional networks that repeatedly produce 2D belief maps for the location to make a dense predictions at each image location. CPMs are completely differentiable and their multi-stage architecture can be trained end to end.

To recognize multi-person pose, Cao *et al.* [7] developed a deep CNN-based regression method to estimate the association between anatomical parts on the image. The architecture jointly learned the part of locations and their association through the two branches of the same sequential prediction process. Furthermore, this bottom-up method achieved state-of-the-art performance in multiple benchmark datasets while achieving real-time performance. On the COCO 2016 key points challenge dataset, this architecture set the state-of-the-art and the results significantly exceeds the previous state-of-the-art methods on the MPII multi-person dataset [33]. In this work, we employed

their pretrained model on our experimental dataset to get an accurate 2D coordinate information of the body parts.

2.5 Pose-based Gait Recognition

With the advent of the pose-estimation algorithms in computer vision, the recognition of human gait based on pose information has received much more attention [34–36] due to its effective representation of gait features and robustness toward covariate condition variations. Feng *et al.* [34] used the human body joint heatmap to describe each frame. They fed the joint heatmap of consecutive frames to long short term memory (LSTM). Their gait features are the hidden activation values of the last timestep.

In [35], Liao *et al.* constructed a pose-based temporal-spatial network (PTSN) to extract the spatial-temporal features of gait from 2D human pose information.

Authors in [36], introduced a model-based gait recognition method, PoseGait, which employed 3D body joint coordinates estimated from 2D pose as input their network as a feature for gait recognition. They also fused four different kinds of features at the input level where some of the handcrafted features were also extracted based on human prior knowledge to form a spatio-temporal feature vector. Finally they trained a 7-layer CNN architecture for CASIA B dataset and a 20-layer CNN architecture for CASIA E dataset to achieve better performance compared with 2D pose estimation. In their experiment, they found that instead of RNN, CNN can achieve high recognition rate in gait recognition.

Again, some of the most successful approaches for human action recognition employ RNNs [8, 9] to effectively model the temporal sequences of human skeleton data. Song *et al.* [8] proposed an end-to-end spatial and temporal attention model with LSTM for human action recognition from skeleton data. In [9], Du *et al.* proposed an end-to-end hierarchical RNN network for skeleton-based action recognition. They divided the human skeleton into five different parts and then separately feed them into five sub-networks.

Our approach to gait recognition is similar to these approaches. In this study, we have proposed a simple RNN architecture that effectively models the discriminative gait features in a temporal domain.

Chapter 3

Methodology

In this chapter, we are going to discuss the proposed framework and its main components in detail. The proposed method is efficient and computationally inexpensive compared to other methods proposed in literature.

3.1 Deep Learning Basics

3.1.1 Deep learning

Machine Learning (ML) is a field of AI that utilizes statistical techniques to learn hidden patterns from available data and make decisions on unseen records. The core task of a ML algorithms is to first build a general model based on the probability distribution of training examples, and then generalize its experience on unseen examples. The process of learning is highly dependent on the quality of data representation.

Deep Learning (DL) is an advanced branch of the ML field that aims to discover the complex representation out of simpler representations. DL methods are typically based on artificial neural networks that consist of multiple hidden layers with nonlinear processing units. The word 'deep' refers to the multiple hidden layers that are used for transforming the data representation. Using the concept of feature learning, each hidden layer of neural networks maps its input data into a new representation. The succeeding layer tends to capture a higher level of abstraction from the less abstract concept in the preceding layer and the hierarchy of learned features in multiple levels are finally mapped to the output of the ML task (e.g., classification and regression) in an unified framework.

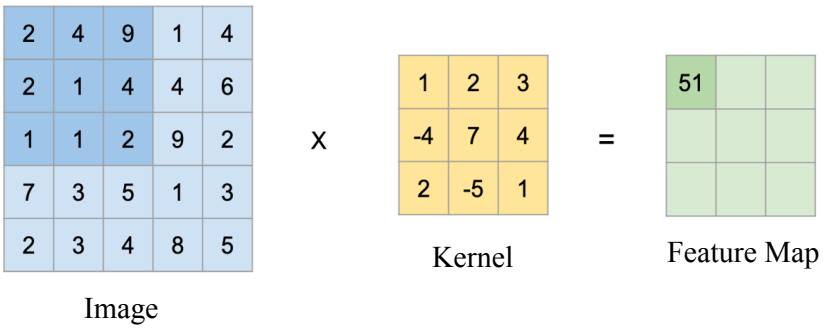


Figure 3.1: Computing output activation of a convolutional layer

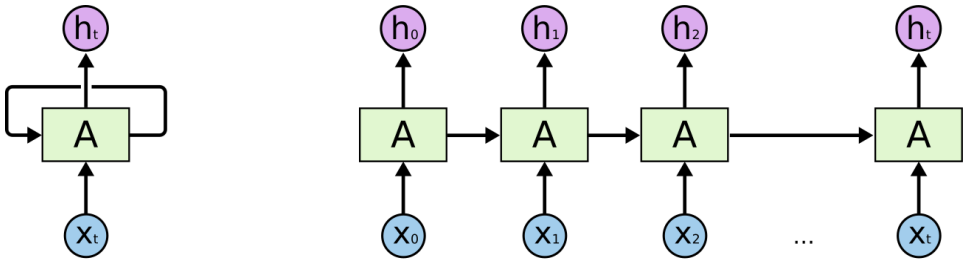
Although, deep neural networks automatically extract rich and high level features which is needed for feature engineering, one of the most time-consuming parts of machine learning practice, increasing the depth does not necessarily improve their performance. Firstly, because they easily suffer from the over-fitting problem, which means the model does not generalize well to test cases. Secondly, because they are more difficult to train and more training data is needed for convergence. To alleviate this problem special types of DNN, e.g., convolutional neural network and recurrent neural networks are introduced.

DL architectures are divided into two broad categories: (1) unsupervised learning approaches including restricted boltzmann machines (RBMs), deep autoencoders, and generative adversarial networks (GANs), (2) supervised learning approaches including deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs).

Some of the real-world applications of Deep Learning include image recognition [37], image captioning [10], machine translation [38], video classification [6] and speech recognition [39]. This thesis seeks to contribute to this growing area of research by exploring the potential power of deep learning techniques in gait recognition.

3.1.2 Convolutional Neural Networks

Convolutional neural networks (ConvNets or CNNs) are a category of neural networks which are proven to be very effective in areas such as image recognition [37], video classification [6], action recognition [40]. They were inspired by biological processes [41] in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the *receptive field*. The receptive fields of different neurons partially overlap such that they cover the entire visual field.



(a) (b)

Figure 3.2: (a) A typical rolled representation of a recurrent neural network. (b) Unrolled recurrent neural network for $t\%$ timesteps. [Image courtesy Chris Olah [42]]

When the input, e.g. images, have a local topological structure that does not depend on the specific location in the global reference system, a dense connectivity pattern might be wasteful. It is usually preferable to be able to exploit the data structure. Firstly, because adapting the connectivity pattern according to the structure of the data reduces total number of parameters and hence number of the operations performed by the network, which consequently reduces the risk of overfitting greatly. Additionally, it also reduce the memory usage and the computation time. Secondly, constraining the connectivity pattern can have the effect of forcing the network to focus on what is important, yielding faster training and better performance.

CNNs exploit this understanding of the data by applying the same pattern detector at every locations in the image. This is formally done through a *convolution*, a signal processing operation that superimposes a pattern detector usually called a *filter* or *kernel* on different locations of the image and emits an activation in each position to produce a matrix of activations, typically referred to as *feature map*.

Let, $\mathbf{X}$ is a two-dimensional image and $\mathbf{W}$ is the weight matrix, also called a kernel, then the convolution operation can be defined as

$$(W * X)(i,j) = \sum_m \sum_n X(m,n)W(i-m, j-n) \tag{3.1}$$

Intuitively, the output of the convolutional layer is formed by sliding the weight matrix over the image and computing the dot product (see Figure 3.1). In any real-world application, it would be common to apply multiple kernels at once with the same convolution hence obtaining a tensor of feature maps.

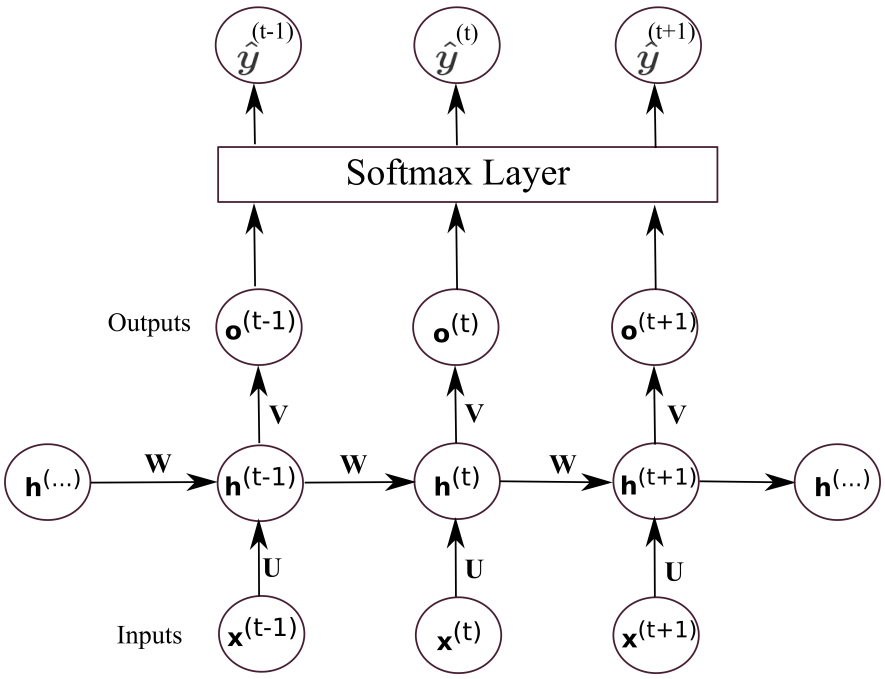


Figure 3.3: The computational graph of a unrolled recurrent network that maps an input sequence of x values to a corresponding sequence of output o values

3.1.3 Recurrent Neural Networks

Recurrent neural networks (RNNs) are a type of artificial neural networks (ANNs) where the output from the previous step is fed as input to the current step. It achieves state-of-the-art performance on various tasks in different domains that include language modeling [43], speech recognition [39], and machine translation [38].

RNNs implement feedback loops (see Figure 3.2(a)) that propagate some information from one timestep to the next. It might not be immediately obvious what it means in practice to put a loop in an ANN and how to backpropagate through it. To better comprehend how RNNs work it is useful to consider its behavior explicitly by *unrolling* the RNN, as shown in Figure 3.2(b).

The forward propagation equations for the RNN depicted in Figure 3.3. Forward propagation begins with a specification of the initial state $\mathbf{h}^{(0)}$. Then, for each timestep from t we apply the following update equations:

$$
\begin{aligned}
\mathbf{a}^{(t)} &= \mathbf{b} + \mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)} \\
\mathbf{h}^{(t)} &= \tanh(\mathbf{a}^{(t)}) \\
\mathbf{o}^{(t)} &= \mathbf{c} + \mathbf{V}\mathbf{h}^{(t)} \\
\hat{\mathbf{y}}^{(t)} &= softmax(\mathbf{o}^{(t)})
\end{aligned}
\tag{3.2}
$$

Here, the parameters are the bias vectors **b** and **c** along with the weight matrices **U**, **V** and **W**, respectively, for input-to-hidden, hidden-to-output and hidden-to-hidden connections. The activation of an RNN (see Figure 3.3) at time t depends on the input at time t as well as on the information coming from the previous step $t-1$. RNNs have a very simple internal structure, that usually amounts to applying some affine transformation to the input and to the previous output, and computing some non-linearity (typically a $tanh$) of their sum.

That sequential information is preserved in the recurrent network's hidden state, which manages to span many timesteps as it cascades forward to affect the processing of each new example. It is finding correlations between events separated by many moments, and these correlations are called *long-term dependencies*, because an event downstream in time depends upon, and is a function of, one or more events that came before. One way to think about RNNs is that they applies the same model to each timestep of the sequence or, equivalently, applies different models at each timestep which share their weights.

For training these networks, it is required to unroll the computation graph and use the backpropagation algorithm to proceed from the most recent timestep, backward in time. This algorithm is usually referred to as *Backpropagation through time* (BPTT). The problem of BPTT is that it requires the application of the chain rule all the way from the current timestep to $t = 0$ to propagate the gradients. This results in a long chain of products that can easily go to infinity or become zero if the elements of the multiplication are greater or smaller than 1 respectively [44]. These two issues, i.e., going to infinity and becoming zero, are known in the literature as *exploding gradient* and *vanishing gradient* [45] problem respectively. The first one can be partially addressed by *clipping the gradient* when it becomes too large, but the second is not easy to overcome and can make training these kind of models very hard.

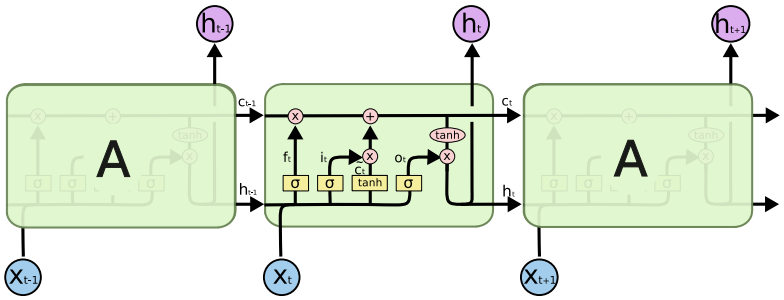


Figure 3.4: A Long Short Term Memory (LSTM). [Image courtesy Chris Olah [42]]

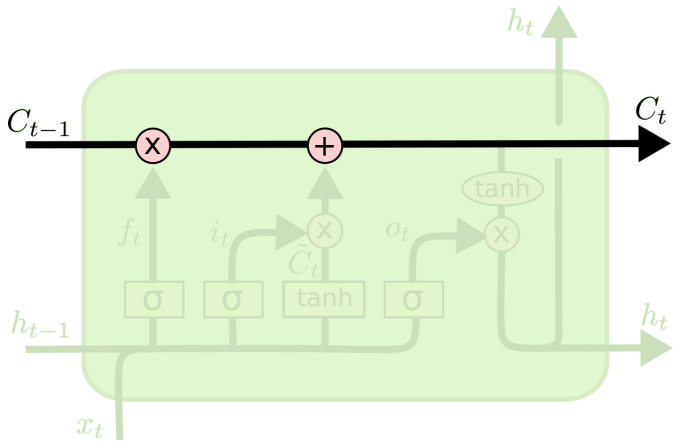


Figure 3.5: The internal state of LSTMs. [Image courtesy Chris Olah [42]]

3.1.4 Long Short-Term Memory

Long short-term memory (LSTM) networks (Figure 3.4) have been proposed to solve the problems of RNNs in modeling long-term dependencies. LSTMs have been designed to have an internal memory, or *state*, that can be updated at each timestep. As opposed to vanilla RNN, this internal memory allows LSTM to separate their output from the information they want to carry over into the future steps.

Figure 3.5 highlights the internal memory path. From the figure it can be observed that how the internal memory of the previous timestep $\mathbf{c}_{t-1}$ is carried over to the current timestep where it is updated through a multiplicative and an additive interaction and concurs to determine the current state of the memory $\mathbf{c}_t$. Thereafter, this state once again, propagated to the next timestep.

LSTMs interact with memory through *gate*, a computational node, that determines the behavior of the model. The *forget gate*, as shown in Figure 3.6, determines how much of the previous step's memory to forget or, equivalently, how much of the previous state to retain. This is modeled through a sigmoid layer (σ) that takes the current input $\mathbf{x}_t$ and the output of the previous step $\mathbf{h}_{t-1}$ and produces an activation vector between 0 and

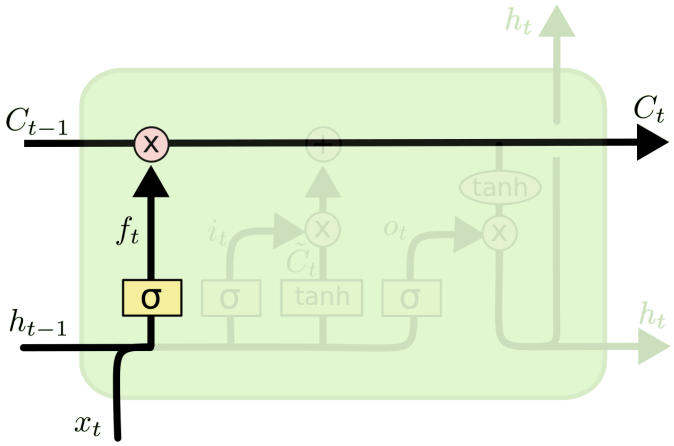


Figure 3.6: The LSTM forget gate. [Image courtesy Chris Olah [42]]

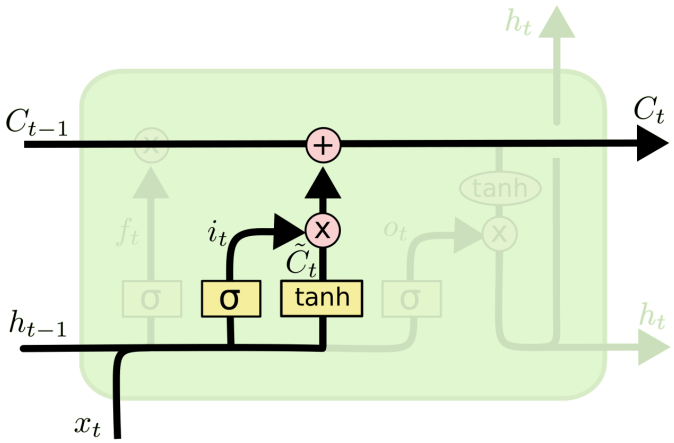


Figure 3.7: The LSTM input gate. [Image courtesy Chris Olah [42]]

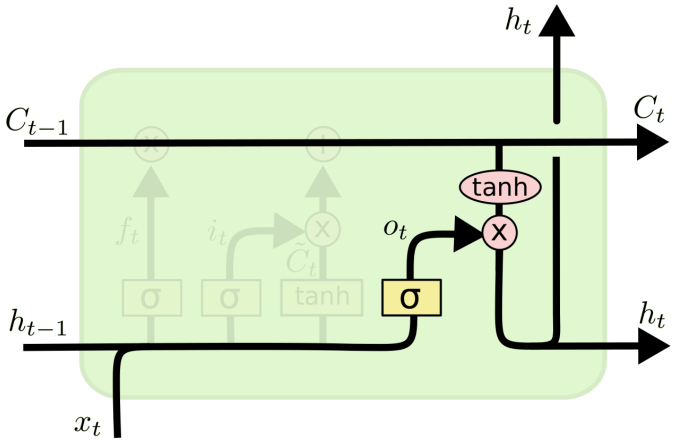


Figure 3.8: The LSTM output gate.[Image courtesy Chris Olah [42]]

1. This activation is multiplied by the previous state $\mathbf{c}_{t-1}$ and results in an intermediate memory state where some of the activations can be weaker than those in $\mathbf{c}_{t-1}$ and some others are potentially zeroed out.

The forget gate allows the LSTM to discard information that is not relevant anymore.

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f \cdot \mathbf{x}_t + \mathbf{U}_f \cdot \mathbf{h}_{t-1} + \mathbf{b}_f\right), \tag{3.3}$$

Again, LSTMs have a mechanism to add new information to the memory. This behavior is controlled by an *input gate* (Figure 3.7) that modulates the amount of the current input that is going to be stored in the memory. This operation is split over two computation paths: similarly to the forget gate, the input gate takes the current input $\mathbf{x}_t$ and the output of the previous step $\mathbf{h}_{t-1}$ and exploits a sigmoid layer to produce an activation vector between 0 and 1. Simultaneously, a $tanh$ layer generates a state update $\tilde{\mathbf{c}}_t$ between -1 and 1. This is governed by the following equations:

$$\begin{aligned} \mathbf{i}_t &= \sigma\left(\mathbf{W}_i \cdot \mathbf{x}_t + \mathbf{U}_i \cdot \mathbf{h}_{t-1} + \mathbf{b}_i\right) \\ \tilde{\mathbf{c}}_t &= tanh\left(\mathbf{W}_c \cdot \mathbf{x}_t + \mathbf{U}_c \cdot \mathbf{h}_{t-1} + \mathbf{b}_c\right) \end{aligned} \tag{3.4}$$

The input gate modulates how much of this state update will be applied to the old state to generate the current state. The forget gate $\mathbf{f}_t$ and the input gate $\mathbf{i}_t$, together with the state update $\tilde{\mathbf{c}}_t$ and the previous state $\mathbf{c}_{t-1}$ fully determine the state at time t.

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t \tag{3.5}$$

The last gate of LSTM is the *output gate* (Figure 3.8) $\mathbf{o}_t$ that, as the name reveals, manipulates the output of the LSTM at time t. The usual sigmoid layer determines the state of the output gate and the memory resulting from the transformations due to the forget and input gates goes through a $tanh$ nonlinearity and is multiplied by the output gate to finally produce the output.

$$\begin{aligned} \mathbf{o}_t &= \sigma\left(\mathbf{W}_o \cdot \mathbf{x}_t + \mathbf{U}_o \cdot \mathbf{h}_{t-1} + \mathbf{b}_o\right) \\ \mathbf{h}_t &= \mathbf{o}_t \circ tanh\left(\mathbf{c}_t\right) \end{aligned} \tag{3.6}$$

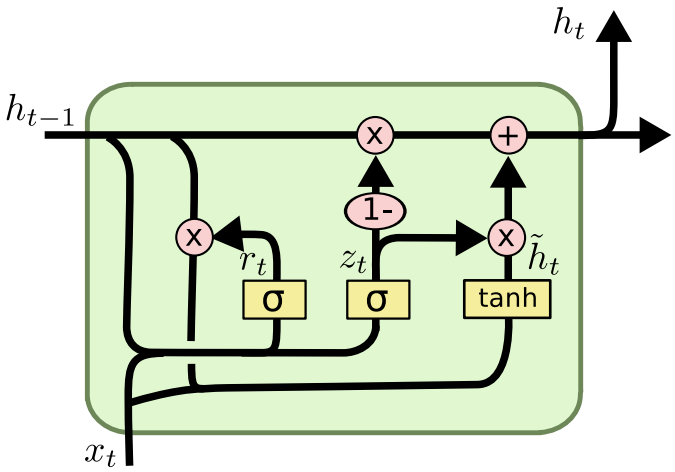


Figure 3.9: Gated Recurrent Units (GRUs). [Image courtesy Chris Olah [42]]

3.1.5 Gated Recurrent Unit

Cho *et al.* [46] proposed a new kind of RNN called gated recurrent unit (GRU), as shown in Figure 3.9, with less gates than LSTM and a different internal structure. In GRUs the forget and input gates are coupled into an *update gate* $\mathbf{z}_t$. The memory and output are also merged into a single state and the internal structure is modified to cope with these changes. Figure 3.9 shows the internal structure of a GRU unit.

The *update gate* $\mathbf{z}_t$ for timestep t helps the model to determine how much of the information from the previous timestep needs to be passed along the future. It is analogous to the output gate in an LSTM cell.

$$\mathbf{z}_t = \sigma\left(\mathbf{W}_z \cdot \mathbf{x}_t + \mathbf{U}_z \cdot \mathbf{h}_{t-1} + \mathbf{b}_z\right) \tag{3.7}$$

On the other hand, *reset gate* in GRU is used to decide how much of the past information needs to forget. It is analogous to the combination of the input Gate and the forget Gate in an LSTM cell.

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_r \cdot \mathbf{x}_t + \mathbf{U}_r \cdot \mathbf{h}_{t-1} + \mathbf{b}_r\right) \tag{3.8}$$

GRU has also a new memory content will use the reset gate to store the relevant information from the past

$$\tilde{\mathbf{h}}_t = tanh\left(\mathbf{W}_h \cdot \mathbf{x}_t + \mathbf{U}_h \cdot (\mathbf{r}_t \circ \mathbf{h}_{t-1}) + \mathbf{b}_h\right) \tag{3.9}$$

In the last step, the GRU calculate the current information $\mathbf{h}_t$ from update gate ($\mathbf{z}_t$),

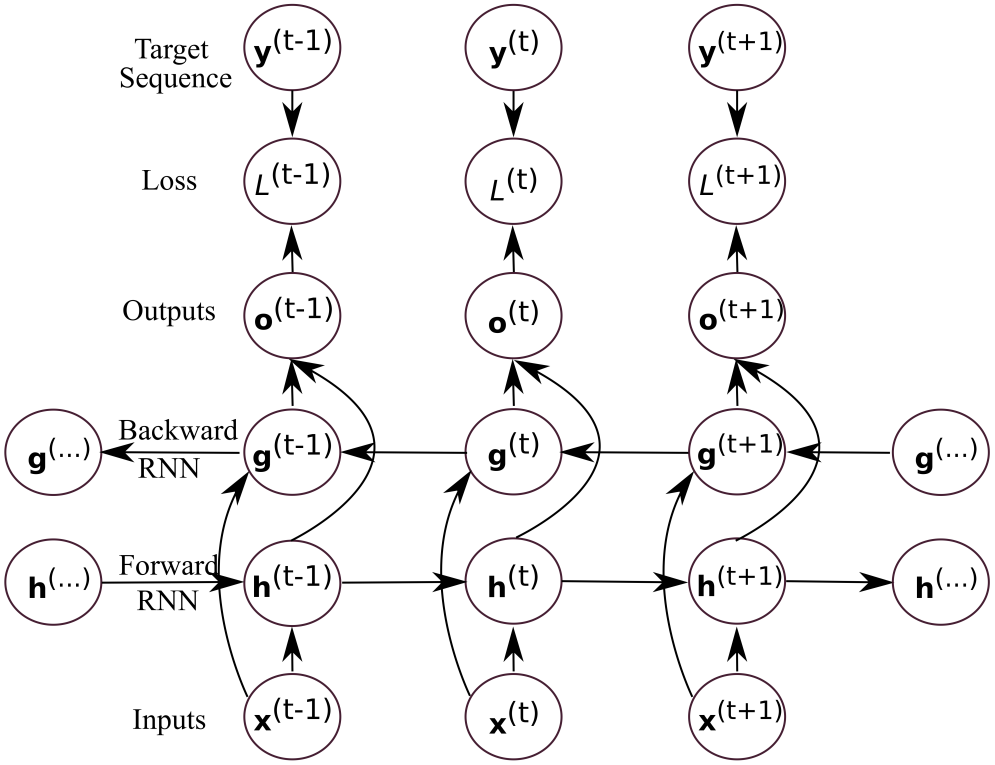


Figure 3.10: The architecture of a vanilla bidirectional recurrent neural network

previous information ($\mathbf{h}_{t-1}$) and memory content ($\tilde{\mathbf{h}}_t$) and passes it down to the network.

$$\mathbf{h}_t = \mathbf{z}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \circ \tilde{\mathbf{h}}_t. \tag{3.10}$$

Therefore, the differences between LSTM unit and GRU are:

- GRUs have 2 gates while LSTMs have 3 gates
- GRUs do not have any internal memory in contrast to LSTMs
- Nonlinearity is not applied when computing the output of GRUs

The advantage of GRUs over LSTMs is the smaller number of gates that make them less memory as well as computationally intense, which is often a critical aspect for ANNs. Therefore, GRU involves less computation compared with LSTM while keeping similar performance and improving the efficiency of the original RNNs. Moreover, GRU has shown better classification performance on smaller datasets [47].

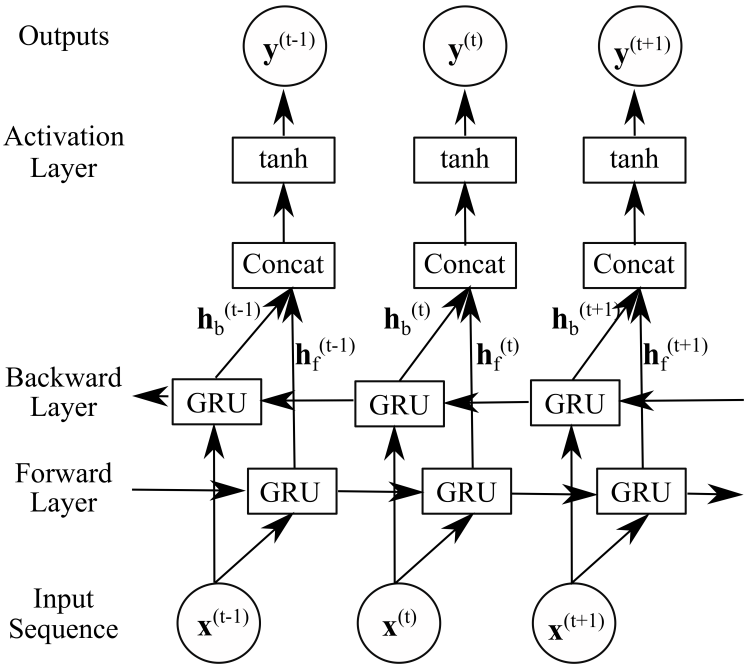


Figure 3.11: The architecture of a bidirectional gated recurrent neural network

3.1.6 Bidirectional RNNs

Bidirectional recurrent neural networks (BRNNs) [48] connect two hidden layers running in opposite directions to a single output, allowing them to receive information from both past and future states. Here, the input sequence is fed in normal time order for one network, and in reverse time order for another. The outputs of the two networks are usually concatenated at each timestep. So, this type of structure allows the networks to have both backward and forward information about the sequence at every timestep. BRNN are especially useful when the context of the input is needed. For example, in handwriting recognition [49], the performance can be enhanced by knowledge of the letters located before and after the current letter. A vanilla architecture of the BRNN is illustrated in Figure 3.10.

3.1.6.1 Bidirectional GRU

In a bidirectional GRU (BGRU) consists of 2 vanilla unidirectional GRUs stacked side by side, but the second GRU reads the input sequence from the opposite direction. Figure 3.11 illustrates the basic architecture of a Bidirectional GRU.

3.1.7 Regularization for Deep Learning

Regularization is a technique which makes slight modifications to the learning algorithm such that the model generalizes better. This in turn improves the model's performance on the test data. As we know, DNNs are highly complex models (many parameters and many non-linearities) and they are easy to overfit, hence, we need some form of regularization.

3.1.7.1 L2 Regularization

This regularization is popularly known as *weight decay*. This strategy drives the weights closer to the origin by adding the regularization term. This technique is also known as *ridge regression*.

3.1.7.2 Dropout

Dropout is a computationally inexpensive but powerful regularization method.

One advantage of dropout is that it is very computationally cheap. Using dropout during training requires only $O(n)$ computation per example per update, to generate n random binary numbers and multiply them by the state. Another significant advantage of dropout is that it does not significantly limit the type of model or training procedure that can be used. It works well with nearly any model that uses a distributed representation and can be trained with stochastic gradient descent.

3.1.7.3 Dataset Augmentation

The simplest way to reduce overfitting is to increase the size of the training data. But mostly we are provided with limited data. One way is to create fake data and add it to our training dataset, for some domains this is fairly straightforward and easy.

3.1.7.4 Early Stopping of Training

One way to think of early stopping is as a very efficient hyperparameter selection algorithm. The idea of early stopping of training is that as soon as the validation error starts to increase we freeze the parameters and stop the training process. Or we can also store the copy of model parameters every time the error on the validation set improves and return these parameters when the training terminates rather than the latest parameters.

Early stopping has an advantage over weight decay that early stopping automatically determines the correct amount of regularization while weight decay requires many training experiments with different values of its hyperparameter.

3.1.7.5 Noise Robustness

Noise is often introduced to the inputs as a dataset augmentation strategy. the addition of noise with infinitesimal variance at the input of the model is equivalent to imposing a penalty on the norm of the weights. Noise injection is much more powerful than simply shrinking the parameters, especially when the noise is added to the hidden units.

Another way that noise has been used in the service of regularizing models is by adding it to the weights. This technique has been used primarily in the context of recurrent neural networks. This can be interpreted as a stochastic implementation of Bayesian inference over the weights.

3.2 Human Pose Estimation

Human pose estimation, one of the core problems in computer vision, refers to the process of inferring poses in an image or video. Essentially, it entails predicting the body parts or body joint positions of individuals in an image. It is the key component which enables machines to have a perception of the people in images and videos. It has been successfully employed in many real-world applications such as action recognition [8], augmented reality [50], gaming [51], and gait recognition [36].

3.2.1 Types of Pose Estimation

Depending upon the output dimension requirement, a pose estimation algorithm can be categorized into 2D pose estimation and 3D pose estimation. In 2D pose estimation, the location of body joint is predicted in terms of pixel values of the image frame. On the other hand, 3D pose estimation is predicting a three-dimensional spatial arrangement of all the body joints as its final output. Again, depending on the number of people being tracked, pose estimation can be further classified into single-person and multi-person. Single-person pose estimation guarantees of only one person present in the frame, whereas, in multi-person pose estimation, each image may contain an unknown number of people who can appear at any position or scale. Therefore, it needs to handle the additional problem of inter-person occlusion.

Figure 3.12: Realtime multi-person 2D pose estimation using OpenPose algorithm that is independent of the number of people in the image. [Image courtesy Cao et al. [7]]

3.2.2 Techniques for Pose Estimation

There are two overarching approaches to pose estimation: a *bottom-up* approach, and a *top-down* approach.

With a bottom-up approach, the model detects every instance of a particular keypoint in a given image and then attempts to assemble groups of keypoints into skeletons for distinct objects. In simpler terms, the algorithm first predicts all body joints present in the image. This is typically followed by the formulation of a graph, based on the body model, which connects joints belonging to the same human. Integer linear programming (ILP) or bipartite matching are two common methods of creating this graph.

While, a top-down approach involves a segmentation step at the start. The network first uses an object detector to draw a box around each instance of an object and then estimates the keypoints within each cropped region.

The potential simplest model for pose estimation used DNN-based regressor to predict X, Y, and potentially Z coordinates for each keypoint location from an input image. In practice, however, this architecture does not produce accurate results without additional refinement.

A slightly more complicated approach employs a deep learning-based encoder-decoder architecture. In this type of approach, instead of estimating the keypoint coordinates directly, the encoder is fed into a decoder, which creates heatmaps representing the likelihood that a keypoint is found in a given region of an image. During post-processing, the exact coordinates of a keypoint are found by selecting heatmap locations with the highest keypoint likelihood. In the case of multi-pose estimation, a heatmap may contain multiple areas of high keypoint likelihood (e.g. multiple right hands in an image).

In top-down approach, an object detection module is placed between the encoder and

Figure 3.13: An example of a bottom up approach. [Image courtesy Cao et al. [7]]

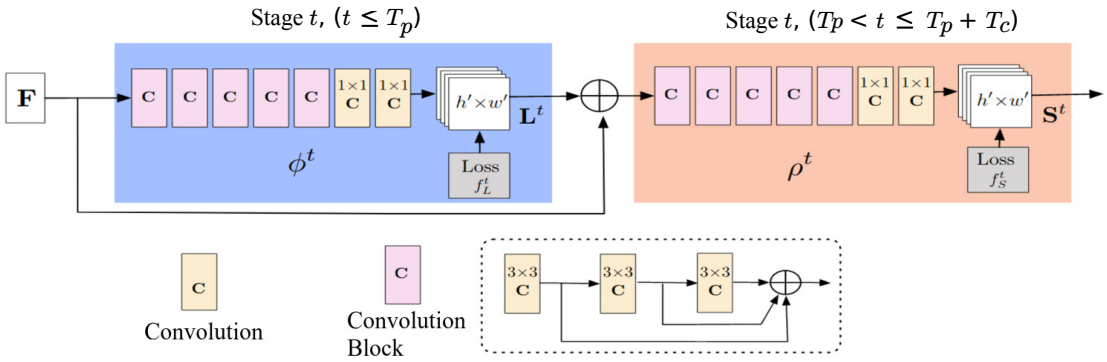


Figure 3.14: Network architecture of the multi-stage CNN. The first set of stages predicts PAFs $\mathbf{L}^t$, while the last set predicts confidence maps $\mathbf{S}^t$. [Image courtesy Cao et al. [7]]

decoder which is used to crop regions of an image likely to contain an object. Keypoint heatmaps are then predicted individually for each box. Rather than having a single heatmap containing the likely location of all of the specific body part in an image, we get a series of bounding boxes that should only contain a single keypoint of each type.

So, top-down approach makes it easy to assign the keypoints to specific instances without a lot of post-processing. However, it suffers greatly when the person detector fails due to close proximity among people. Furthermore, their runtime is proportional to the number of people in the image. Contrarily, bottom-up approaches show robustness to early commitment and have the potential to decouple runtime complexity from the number of people in the image [7].

3.2.3 Introduction to OpenPose Library

In this research, we have employed OpenPose [7], an open-source library for realtime multi-person 2D pose detection including body, foot, hand, and facial keypoints. This bottom-up approach achieves state-of-the-art accuracy in realtime performance.

The overall pipeline of the OpenPose library is illustrated in Figure 3.13. An RGB image(Figure 3.13a) is fed as input to the library and it outputs the 2D locations of anatomical keypoints for each person in the image (Figure 3.13e). Firstly, a feedforward network predicts a set of 2D confidence maps $\mathbf{S}$ of body part locations (Figure 3.13b) and a set of 2D vector fields $\mathbf{L}$ of part affinity fields (PAFs), which encode the degree of association between parts (Fig. 3.13c). The set $\mathbf{S} = (\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_J)$ has J confidence maps, one per part, where $\mathbf{S}_j \epsilon \mathbb{R}^{w \times h}$. The set $\mathbf{L} = (\mathbf{L}_1, \mathbf{L}_2, ..., \mathbf{L}_C)$ has C vector fields, one per limb, where $\mathbf{L}_c \epsilon \mathbb{R}^{w \times h \times 2}$. Each image location in $\mathbf{L}_c$ encodes a 2D vector. Finally, the confidence maps and the PAFs are parsed by greedy inference (Figure 3.13d) to output the 2D keypoints for all people in the image. The network architecture of OpenPose algorithm, shown in Figure 3.14, iteratively predicts affinity fields that encode part-to-part association.

3.3 Extracting Spatio-Temporal Feature Vector

The workflow of the proposed network is illustrated in Figure 3.15. Many strategies have been taken to designed a lower-dimensional spatio-temporal feature descriptor based on the 2D human poses estimated from the raw video frames. In this section, we elaborate the feature extraction procedure of our proposed method.

3.3.1 2D Body Joint Features

As all the joints in the human body do not play a significant role in gait pattern, they cannot improve gait recognition accuracy. Some joints perform even worse. So, among the 25 body joints estimated from the OpenPose algorithm, we searched out for those joints which have a rich and discriminative gait representation capacity. Cunado *et al.* [52] used the human leg-based model as they found that change of human leg contains the most important features for gait recognition. In our study, we found that knee along with the joints located in the feet show more robustness than any other body joints because they do not alter while people are walking in cloths or carrying bags. Some joints, e.g. hip, get wider in coat than normal condition. Again, in some gait videos, some subjects put their hands into their coat pocket, which they cannot do in normal walking. This situation significantly changes the joint coordinates. Therefore, raw body joints above hip do not have any significant impact on gait pattern. Hence, in our method, we did not consider hip or any other body joints above it.

Consequently, in our work, as shown in Figure 3.16a, we selected 6 body joints (RKnee,

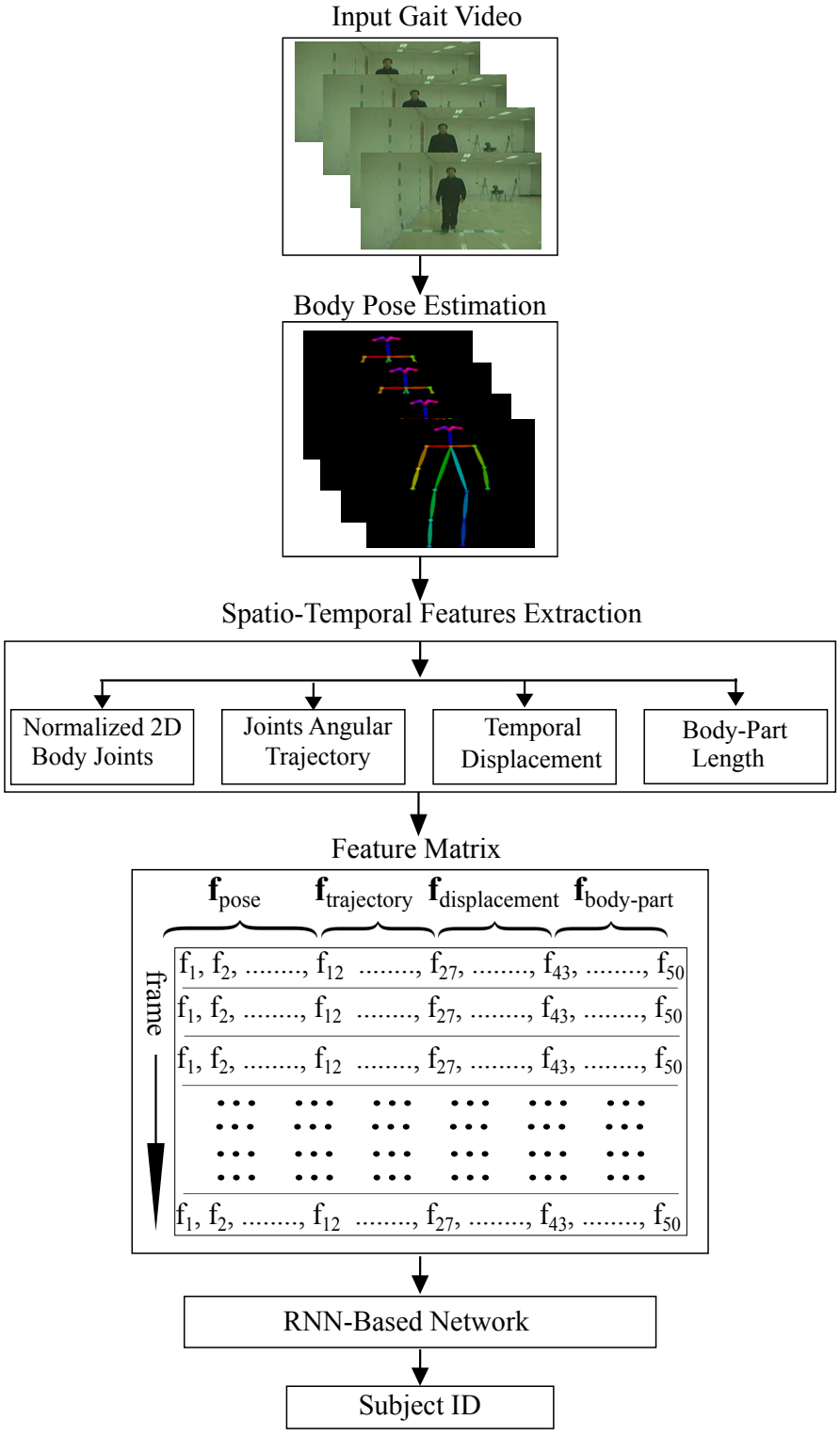
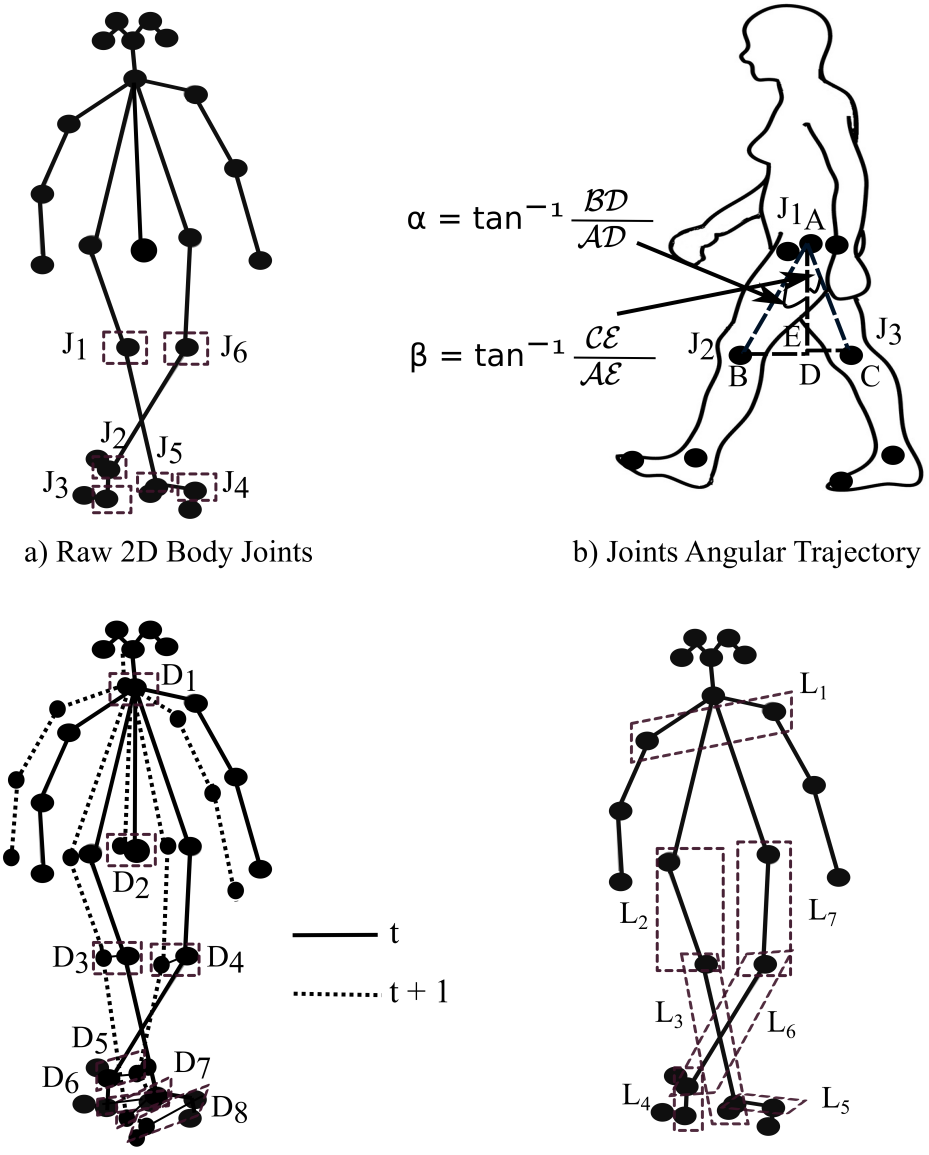


Figure 3.15: The overview of the proposed framework for gait recognition. 2D human poses were first extracted from raw video frames using improved OpenPose [7] algorithm. Four different types of spatio-temporal features were then extracted to form a 50-dimensional feature vector. Thereafter a pose sequence of timestep each having a length of 28 frame was formed to feed into a temporal network. The temporal network identified the subject by modeling the gait features.



a) Raw 2D Body Joints

b) Joints Angular Trajectory

c) Temporal Displacement

d) Body-part length

Figure 3.16: Scheme for the four different types of feature extraction process of the proposed method. a) 6 effective joints were selected out of 25 body joints. These selected joints formed a 12-dimensional pose vector. b) 5 angular trajectories from the lower limbs were considered to form a joint-angle feature vector. c) A total of 8 body joints were selected to get a temporal displacement feature vector. d) 7 body parts were taken to form a body part length feature vector.

Rankle, RBigToe, LKnee, LAnkle, LBigToe) to form our effective body pose features. Thus, we have 12-dimensional pose feature vector, $\mathbf{f}_{body-joint}$, for a single frame.

$$\mathbf{f}_{body-joint} = [x_1, y_1, x_2, y_2, \ldots\ldots, x_6, y_6]^T \tag{3.11}$$

It is necessary to normalize the pose sequence data with regard to the subject position in frame, size, or speed of walking to get improved performance. Now, in different gait datasets, as people walk through the fixed camera, the size of the subject's body alters due to variation in the distance between the subject and the camera. Therefore, in order to eliminate different sizes and location variations of the human skeleton we had to transform the 2D coordinates of all joints into a new coordinate system whose origin was selected as the middle of the hip ($\mathbf{J}_o$). To find the origin of the coordinate system ($\mathbf{J}_o$) for each subject, we considered the right, left, and middle of the hip joints and calculated the average of them.

$$\begin{aligned} \mathbf{J}_o = (x_o, y_o) &= (\mathbf{J}_{LHip} + \mathbf{J}_{RHip} + \mathbf{J}_{MHip})/3 \\ (\bar{x}_i, \bar{y}_i) &= (x_i, y_i) - (x_o, y_o) \quad \forall j \epsilon \mathbf{J} \end{aligned} \tag{3.12}$$

Here, $(\bar{x}_j, \bar{y}_j)$ is set by root-centered coordinate reference system defined by above equations. Again, we normalized the skeletons of different subjects to fixed size by considering h, the euclidean distance from hip to neck joint, as unit length. The following equation shows the normalization procedure of the raw 2D joints.

$$\begin{aligned} h &= \| \mathbf{J}_o - \mathbf{J}_{neck} \|_2 \\ \mathbf{J}_i^N &= (\bar{\bar{x}}_i, \bar{\bar{y}}_i) = (\bar{x}_i, \bar{y}_i)/h \end{aligned} \tag{3.13}$$

Here, $\mathbf{J}_i^N$ be the new coordinate of the i^{th} joint $\mathbf{J}_i$ of a particular pose. These two steps of normalization have the huge impact on robustness of the gait recognition algorithm. Firstly, they allow fair comparisons between different subject's poses reducing the effect due to variation of subject size and position in the camera. Secondly, as it discards the absolute coordinates of subject's body pose, pose size become homogeneous among different camera settings and proximity to camera. Thus, it makes the system robust to zooming, camera position, and subject location.

Table 3.1: List of selected joint-angle trajectories with corresponding body joints set in order to form a joint angular feature vector.

Angular Trajectory	Body Joints Set
Hip trajectory	10, 8, 13
Right knee trajectory	11, 10, 9
Left knee trajectory	14, 13, 12
Right ankle trajectory	22, 11, 10
Left ankle trajectory	19, 14, 13

3.3.2 Joint Angular Trajectory

The dynamics of gait motion can be expressed by the temporal information of joint angles. Hence, discriminative gait features can be found by considering the change in joint-angle trajectories of the lower limbs [53]. Therefore, in this study, we formulated another 15-dimensional feature vector $\boldsymbol{f}_{trajectory}$ by considering five lower limb joint-angle trajectories using the following equations:

$$\begin{aligned}
\alpha &= \begin{cases} \tan^{-1}\frac{|J_{2,x}-J_{1,x}|}{|J_{2,y}-J_{1,y}|} & J_{2,y} \neq J_{1,y} \\ \pi/2 & J_{2,y} = J_{1,y} \end{cases} \\
\beta &= \begin{cases} \tan^{-1}\frac{|J_{3,x}-J_{1,x}|}{|J_{3,y}-J_{1,y}|} & J_{3,y} \neq J_{1,y} \\ \pi/2 & J_{3,y} = J_{1,y} \end{cases} \\
\theta &= \alpha + \beta
\end{aligned} \tag{3.14}$$

As shown in Figure 3.16b, J_1, J_2, J_3 are the joints which form a set of angular trajectory. In this work, we considered total five sets of angular trajectories from the lower limb of human body. Table 3.1 demonstrated the selected angular trajectories with their corresponding body joints. For each trajectory, we took (θ, α, β) as gait features.

$$\mathbf{f}_{trajectory} = [\theta_1, \alpha_1, \beta_1, \theta_2, \alpha_2, \beta_2, \ldots\ldots, \theta_5, \alpha_5, \beta_5]^T \tag{3.15}$$

3.3.3 Temporal Displacement

Our third type of feature extractor is a simple descriptor that preserves temporal information of the gait pattern. It basically stores the local motion features of gait by keeping the displacement information between the two adjacent frames of the subject's

pose sequence. The displacement of each coordinate of a joint was then normalized by the total length of displacement of all joints. Let, t and $(t+1)$ are two adjacent frames of a particular pose sequence. Now, the displacement information of the coordinates of any joint of frame t would be the normalized difference between the corresponding coordinates of two adjacent frames.

$$\begin{aligned} \triangle x_1^t &= \frac{x_1^{t+1} - x_1^t}{\sum_{i=1}^{8} \parallel J_{i,x}^{t+1} - J_{i,x}^t \parallel_2} \\ \triangle y_1^t &= \frac{y_1^{t+1} - y_1^t}{\sum_{i=1}^{8} \parallel J_{i,y}^{t+1} - J_{i,y}^t \parallel_2} \\ \mathbf{f}_{displacement} &= [\triangle x_1, \triangle y_1, \triangle x_2, \triangle y_2 \ldots\ldots, \triangle x_8, \triangle y_8]^T \end{aligned} \tag{3.16}$$

Here, J_i^t is the 2D coordinates of the i_{th} body joint at t^{th} frame in the video and $(\triangle x_1^t, \triangle y_1^t)$ is the displacement of the coordinates of first joint at t^{th} frame of the video. As shown in Figure 3.16c, we selected 8 joints (Neck, MHip, RKnee, Rankle, RBigToe, LKnee, LAnkle, LBigToe) to get a 16-dimensional feature vector, $\mathbf{f}_{displacement}$.

3.3.4 Body Part Length Features

The static gait parameters, for example, the length of the body parts calculated from raw body joints position are also very important for gait recognition [53, 54]. They form a spatial gait feature vectors which make them robust against covariate factors such as carrying and clothing condition variation. In this study, we took seven body parts (Figure 3.16 (d)) namely length of the two leg, two feet, two thigh and width of the shoulder which formed a 7-dimensional spatial feature vector $\mathbf{f}_{body-part}$.

3.3.5 Fusion of Features

A lot of research works have been done to fuse multiple features to get improved performance [36, 53]. Different types of fusion methods were proposed in literature such as feature level fusion, representation level fusion, and score level fusion. In feature level fusion, multiple features of the same frame are concatenated before feeding into a final network and in representation level fusion, each feature vector is firstly fed into a network and the resulting global representations are then concatenated to train a final classifier. For score level fusion, each feature vector is separately fed into the final network which predicts a classification score. Then, the scores from multiple classifiers

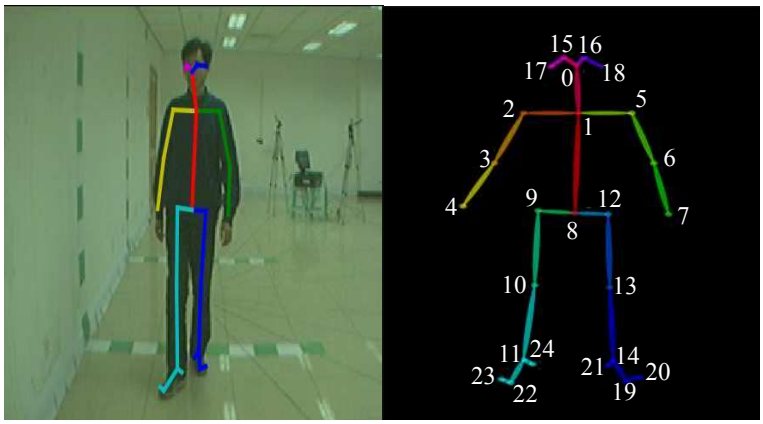


Figure 3.17: Examples of 2D human pose estimation by [7] from RGB images of CASIA dataset (left ones). Detected 25 human body joints with description are shown. (right ones)

are fused using an arithmetic mean.

In this study, we found that feature level fusion has produced better recognition results in contrast to other fusion techniques or individual feature sets.

3.4 Feature Preprocessing

From 2D pose estimation algorithm [7], we got raw 2D coordinates for each joint, as shown in Figure 3.17. In this work, several preprocessing steps have been undertaken to build a compact, robust and discriminative descriptor based on these raw coordinates. In this section, we are going to discuss these steps in detail.

3.4.1 Handling Missing Joint Information

One of the most challenging tasks for pose estimation algorithm to estimate the pose of a subject who is completely or partially occluded. This scenario often leads the algorithm to fail in estimating one or more joint coordinates. In order to make our proposed gait algorithm robust and accurate, we have to address the problem of missing joint information carefully. The main strategies we have taken in this work are:

- If the origin of the coordinate system can't be calculated due to missing hip joints, the frame should be rejected;
- If more than 1 body joint is missing in between knee and ankle joints of both leg, the frame should be rejected due to having little information;

- Persistent missing joints can be guessed by exploiting the left and right side body symmetry;
- In other cases, individual joints were not located in the frame and a position of $[0.0, 0.0]$ was given to that joint.

The above strategies are simpler which do not require any computation and proven to be effective in addressing the missing data problem. Algorithm 1 depicts the proposed techniques for handling missing joint information.

Algorithm 1 Algorithm for Handling Missing Joints Information

Input: Raw 2D poses with missing joints
Output: Refined 2D poses
Initialization: set f = 0
while pose at frame f is available **do**
  $f = f + 1$
  read pose at *f*
  if the *origin* is missing **or** missing knee and ankle joints of both legs$>$ 1 **then**
    skip current pose
  end if
end while
for 5 *to* 6 **do**
  if i-th landmark is missing for all frames **then**
    replace i-th joint information with the vertical specular joint
  else
    replace with a position [0.0, 0.0]
  end if
end for

3.4.2 Forming Feature Map

In this thesis, we designed a 50-dimensional spatio-temporal gait feature vector **p** from the raw 2D pose estimation of each frame. Firstly, we split a gait video into 28 frame segments. Each 28 frame-segment formed a timestep which can be described by the following equations.

$$\begin{aligned} \boldsymbol{p} &= [f_1, f_2, f_3, \ldots\ldots, f_{50}]^T \\ \boldsymbol{T} &= [\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \ldots\ldots, \boldsymbol{p}_{28}]^T \epsilon \quad \mathbb{R}^{28\times 50} \\ \boldsymbol{V} &= [\boldsymbol{T}_1, \boldsymbol{T}_2, \boldsymbol{T}_3, \ldots\ldots \boldsymbol{T}_N]^T \end{aligned} \tag{3.17}$$

Here, **p** is the 50-dimension pose vector for each frame; ***T*** is the feature matrix for each timestep; N is the total number of timestep sequence, and ***V*** is the sequence of features for a gait video.

3.4.3 Data Augmentation

The performance of deep neural networks is strongly correlated with the amount of available training data. Although, CASIA [11] is the largest gait dataset, the standard experimental setup of this dataset (see Table 4.2) allows us to train only the four normal walking sequence for each subject. Therefore, we need to augment our train data to obtain a stable model. One way to increase the amount of training data is to overlap video clip. So, we split the input video into an overlapping sequences of video clips. For every 28 image clip, we overlapped 24 images of the previous clip at almost **85.7%** overlapping rate. For example, a particular gait video of 100 frames would be split into the clips $(1-28), (5-32), (9-36), ...$ up to frames $(73, 100)$.

Again, in CASIA dataset, gait videos of different subject have varying timesteps. The number of timesteps in each gait video depends on the total number of frames where a person is detected. Due to the position of the camera, some angles $(0°, 18°, 36°)$ have more person detected frame than other angles $(72°, 90°, 108°)$. Therefore, the total number of timesteps in a gait video is different for different subjects and view angles. This varying timestep makes our train dataset unbalanced. Again, in CASIA B dataset, every subjects have not have all gait videos; there are some missing gait videos. To solve the problem, for improved performance we have to develop our own balance training set by making each subject pose sequence to have a fixed number timesteps. We first found the subject which had maximum timesteps for a particular gait angle and then augmented other subject's timesteps with that specific length by overlapping their sequences.

In addition to above technique, we further augment our training data by adding another gait sequence (i.e., 25% increment) by implementing Gaussian noise to a given normal walking sequence.

$$N(j_i) = (x + \tilde{x}, \quad y + \tilde{y}) \tag{3.18}$$

Here, $\tilde{x}$ and $\tilde{y}$ are two random real numbers generated by a normal distribution with zero mean and unit standard deviation. We apply noising (N) into the raw joints position of a training pose data.

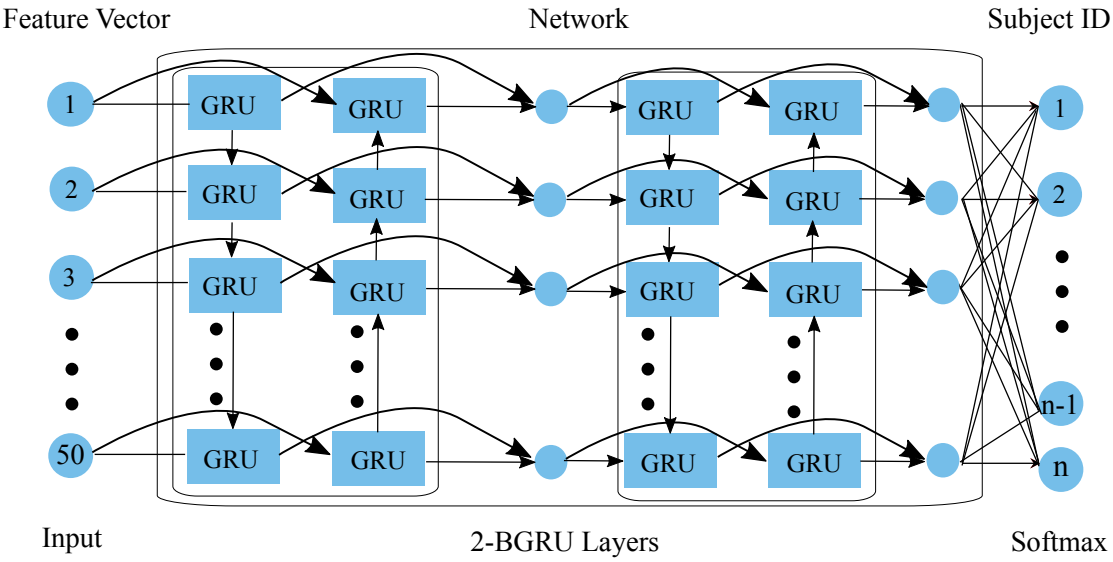


Figure 3.18: Proposed network architecture for robust gait recognition. It consists of two BGRU [48] layers each of which consists of 80 GRU cells with one batch normalization. The network was fed with a 50-dimensional spatio-temporal feature vector obtained from 2D pose estimation. Input layer was followed by a batch normalization layer [55]. The output of the recurrent layers was also batch normalized to standardize the activations and finally fed into an output softmax layer. For the output layer, the number of the output neuron equals to the number of subjects.

3.5 Single-View Gait Recognition

In this section, we will present the details of the architecture and training procedure of our proposed network for single-view gait recognition. We will also try to describe why our proposed 2-layer BiGRU network is best in modeling the gait descriptors for recognizing the subject ID.

3.5.1 Network Architecture

In this research, we experimented with different RNN architectures such as gated recurrent units (GRUs), long short-term memory units (LSTMs), bidirectional long short-term memory (BLSTM) [56] and bidirectional gated recurrent units (BGRU) [48]. Firstly, we designed the proposed network employing all these architectures with one recurrent layer and then, searched for optimum recurrent unit size between 50 to 150. Thereafter, we increased the capacity of the network by adding the second and third layers of hidden units. Finally, we found that, among different RNN architectures, 2-layer BGRU with 80 hidden units performs best. Therefore, we chose GRU in our proposed network architecture as it achieves high performance and requires a reduced number of

parameters while still retaining long-term temporal information.

After input and the second recurrent layer, we placed a batch normalization (BN) [55] layer. At last, a fully connected layer with softmax activation was used to predict the subject classes. Figure 3.18 illustrates the architecture of the proposed network.

3.5.2 Loss Function

In this work, we found that due to the influence of various covariate factors, intraclass distance related to one subject is sometime more significant than interclass distance. So, if we only use the *cross-entropy loss* as our objective function, the resulting learned features may contain large intraclass variations. Therefore, to effectively reduce the intraclass variations, we employed *center loss* as introduced by Wen *et al.* [57] for face recognition task.

Now, as the training progresses, the center loss learns a center for the features of each class. Also, the distances between the features and their corresponding class centers are minimized simultaneously. However, using only center loss may lead the learned features and the centers close to zeros due to the very small value of the center loss. Hence, with the fusion of softmax loss (L_s) and center loss (L_c), we can achieve discriminative feature learning by increasing interclass dispersion and compacting intraclass distance as much as possible.

$$\begin{aligned} L_s &= -\sum_{i=1}^{m} log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} \\ L_c &= \frac{1}{2}\sum_{i=1}^{m} \parallel \boldsymbol{x}_i - \boldsymbol{c}_{y_i} \parallel_2^2 \\ L &= L_s + \lambda L_c + \lambda_\theta \parallel \theta \parallel_2 \end{aligned} \tag{3.19}$$

Equations (3.19) describe the total loss (L) calculation of our proposed network. where $\boldsymbol{x}_i \epsilon \mathbb{R}^d$ denotes the i^{th} pose sequence which belongs to the y_i^{th} class and $\boldsymbol{c}_{y_i} \epsilon \mathbb{R}^d$ denotes to the y_i^{th} class center of the learned pose features. $W \epsilon \mathbb{R}^{d \times n}$ is the feature dimension of the last fully connected layer and $b \epsilon \mathbb{R}$ is the bias term of the network. The batch size and the class number are m and n respectively. λ, a scalar variable, is set to value 0.01 to balance between the two loss functions. $\parallel \theta \parallel_2$ refers to the kernel regularizer for all the parameters of the network with a weight decay coefficient (λ_θ) set to 0.0005 for the experiment.

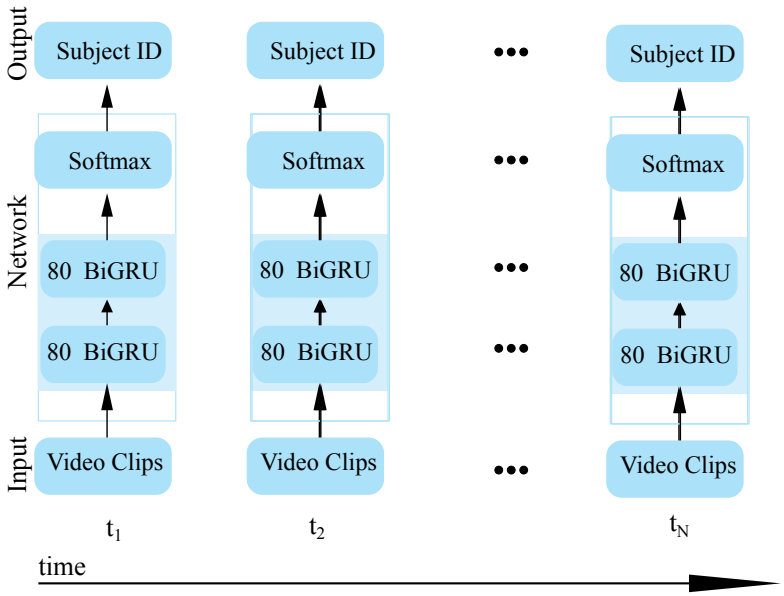


Figure 3.19: Output prediction scheme of our proposed network. Each input clip, a timestep of 28 frame, was considered as a separate video and a sequence of class probabilities was predicted at the output. Majority voting scheme was used to process the output to predict the subject ID.

3.5.3 Post-processing

While training, our proposed temporal network considered each of these video clips as a separate video (see Figure 3.19). For a given video, the prediction of our model is a sequence of class probabilities for each timestep, i.e. 28-frame clip.

But, while testing, we actually need the subject ID for the complete gait video. Therefore, we used *majority voting scheme* to process this output to predict the subject ID. In this scheme, the subject that receives the highest number of votes over all timesteps in a gait video is referred as the predicted class.

Let's consider, $\boldsymbol{s}$ is a vector IDs for n number of subjects. For a particular timestep t, a gait video has input feature map $\boldsymbol{X}^t \epsilon \mathbb{R}^{28\times 50}$ and an n-dimensional output vector $\boldsymbol{o}^t$.

$$\begin{aligned} \boldsymbol{s}^t &= [s_1, s_2, s_3, ..., s_n]^T \\ \boldsymbol{o}^t &= [o_1, o_2, o_3, ..., o_n]^T \end{aligned} \tag{3.20}$$

Here, $o_i^t = P(s_i|X^t)$ refers the probability of input feature map $\boldsymbol{X}^t$ belongs to class s_i. Now, we assign the output class $\mathbf{o}^t$ to the subject class s_i which have maximum probabilities for the timestep t. As each of our gait videos is divided into a series of timestep sequence (see equation 3.20), using majority voting scheme we can have the

Table 3.2: Training summary of our proposed temporal network.

Hyperparameter	**Value**
Optimizer	Adam [58]
Objective function	Fusion of softmax and center loss
Epochs	450
Initial learning rate	5×10^{-3}
Mini-batch size	256

subject ID. Following equations described the voting scheme:

$$\begin{aligned} s_t &= \arg\max_{s_i} \{o_i^t | 1 \leq i \leq n\} \\ s &= \arg\max_{i \in (1,2,\ldots,n)} \sum_{t=1}^{N} s_i^N \end{aligned} \tag{3.21}$$

Here, N is the total number of timesteps in which a gait is split and s is the final predicted class.

3.5.4 Training and Implementation Details

The training of RNNs allows us to learn the parameters from the sequence. We have employed Adam [58] optimization algorithm with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, which is known to work very well for training RNNs. We tried several learning rates in our experiment and found out that the best initial learning rate is $(5\text{x}10^{-3})$. We also reduced the learning rate by a factor when it hit a plateau. Reducing the learning rate will allow the optimizer to get rid of the plateaus in the loss surface. Table 3.2 summarizes all the hyperparameters setting of our network.

The proposed network was trained with a batch size of 256 for 450 epochs. Our network showed some overfitting mostly due to the high learning capacity of the network over data. We addressed the overfitting problem by adding a BN layer before and after the BGRU layer. We also tried to add dropout layer during training, but that did not help to reduce the overfitting problem. Moreover, it degraded gait recognition performance. Hence, we skip it.

For the model computations, we entirely relied on GPU programming. In particular, our implementation was based on Keras [59], a GPU-capable deep-learning library written in Python. All the experiments were performed on a server machine with 56 cores, 512

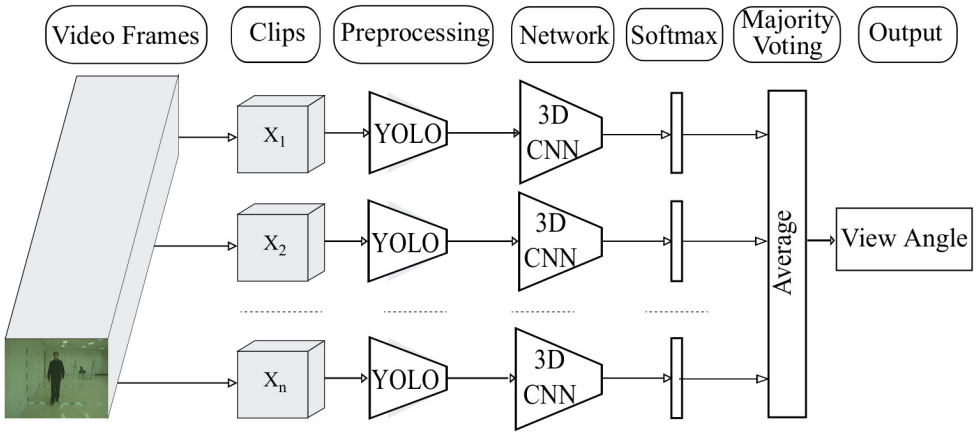


Figure 3.20: Overview of our proposed view angle identification network scheme. YOLOv3 [61] was used to detect and locate the walking people in video frames. The input of the network was a clip of 16 consecutive frame which was preprocessed and resized to 112×112 to feed into a 3D convolutional network based on C3D [40]. The network used 3D kernels to exploit spatio-temporal dynamics for view angle identification.

GB RAM and a Nvidia Tesla K40 graphic card with 12 GB memory running on Ubuntu server 18.04 LTS. Input video and image sequences were processed using Python and OpenCV library [60].

3.6 Multi-View Gait Recognition

In this section we will elaborate our propose two-stage network architecture for multi-view gait recognition. Here, in first stage, we trained a 3D convolutional network to estimate the walking direction of the subject by extracting spatio-temporal features from gait video. Thereafter, we performed subject identification using proposed temporal network which has been trained for that view angle.

3.6.1 Preprocessing

Firstly, to localize human walking in gait videos, we used YOLOv3, a state-of-the-art realtime object detection algorithm, proposed by Redmon *et al.* [61]. We then cropped each of the person detected frame using the bounding box coordinates found from YOLOv3 algorithm and resized them to 112×112 for our network input. Thereafter, we splitted each gait video into overlapping sequences of 16 consecutive frames within

Table 3.3: Training summary of our proposed 3D-CNN network.

Hyperparameter	**Value**
Optimizer	Stochastic gradient descent (SGD)
Objective function	Mean squared error (MSE)
Epochs	70
Initial learning rate	1×10^{-3}
Mini-batch size	12
Momentum	0.92

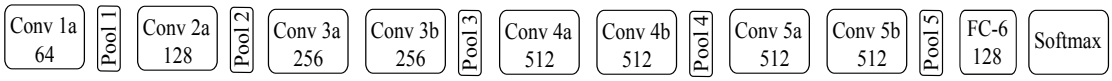


Figure 3.21: Proposed 3D-CNN for view angle identification. Last 3 layers of a pretrained C3D [40] network has been replaced by a fully connected layer of 128 neurons followed a final softmax layer of 11 neurons to classify 11 different walking directions of CASIA B dataset.

training or test set. There is an overlap of 8 frames (50%) indicating that the samples were gathered using a 16 frame sliding window with a 50% stride.

3.6.2 3D Convolution for Video Classification

Identifying walking direction from gait video is somewhat similar to action recognition problem in computer vision. Recently, in action recognition, researcher have started to exploit 3D features in video using 3D-CNN model which extracts features from both spatial and temporal dimensions by performing 3D convolutions. Tran et.al. [40] proposed a 3D convolutional neural network, also known as C3D, which has been widely used for applications like video classification, action recognition, etc. Sports-1M [6], one of the largest benchmark datasets for video classification has been employed to train the network. The dataset contains 1.1 million sports videos, where each video belongs to one of the 487 sports categories.

3.6.3 Network for View Angle Identification

Successful transfer learning within or across different domain of interest leads to significant improvement in performance due to the amount of jointly learning representations in a shared feature space. In our work, we used a pretrained C3D model and fine-tuned it for our 3D Convolutional network to determine the view angle from gait videos. Fig. 3.21 shows our proposed 3D convolutional network.

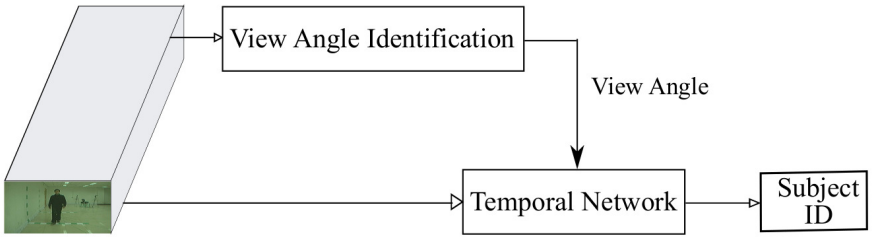


Figure 3.22: Poposed two-stage network for multi-view gait recognition.

C3D network is composed of 8 convolutional layers, 5 pooling layers, 2 fully-connected layers, followed by a softmax layer at the end. All the 3D convolution kernels are $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions. We removed the last 3 layer from the model and then added a fully connected layer of 128 neurons and a dropout layer of 0.5 to avoid overfitting. Finally, a softmax layer of 11 neuron has been added to classify any given videos into 11 different viewing angles.

The proposed method for our view angle identification is illustrated in Figure 3.20. The input of the network was a clip of 16 consecutive frame which was preprocessed and resized to 112×112 to feed into a 3D-CNN network. We used *majority voting scheme* to process the output to predict the view angle similar to section 3.5.3, i.e. the angle that receives the highest number of votes over all clips are referred as predicted angle of the video.

3.6.4 Two-Stage Network for Multi-View Gait Recognition

3.6.5 Training Details

We employed CASIA B gait dataset [11] to train our model. We trained the network using 4 normal walking sequences of 100 subjects in gallery set of CASIA B as described in Table 4.2. Our network was trained with a 12 batch size with an initial learning rate 10^{-3} for 70 epochs. Table 3.3 summarizes all of the hyperparameters setting of our proposed network.

Chapter 4

Experimental Results

In this chapter, we are going to evaluate our proposed method at different experimental setup on multiple benchmark datasets. In Section 4.1, we have explained different state-of-the-art gait recognition datasets that we used to train and evaluate our proposed method. As to estimate pose, RGB video frames are required, hence, we couldn't evaluate our method to those datasets which only consists of silhouette sequences. In Section 4.2 and Section 4.3 we have presented the experimental results of our proposed method on single-view and cross-view gait recognition respectively. The performance of our method in multi-view gait recognition is discussed in Section 4.4.

4.1 Dataset

The success of deep learning-based algorithms greatly depends on the vast amount of labeled training data. However, unfortunately, few existing gait datasets have a large number of subjects as well as a variety of covariate factors. Some of the publicly available gait datasets are CASIA A and CASIA B gait dataset [11], TUM GAID dataset [62], SOTON database [25], OU-ISIR multi-view large population dataset (OU-MVLP) [63], and USF HumanID dataset [19].

- **USF HumanID gait dataset** [19]: there are 122 subjects walking outside on two different surfaces of an elliptical path under two different time, view angle, clothing, shoes, and carrying conditions. However, every subject was not filmed under all conditions.
- **TUM GAID dataset** [62]: Another large dataset for gait recognition which consists of 305 subjects where each subject has 10 gait videos. As, all of the videos

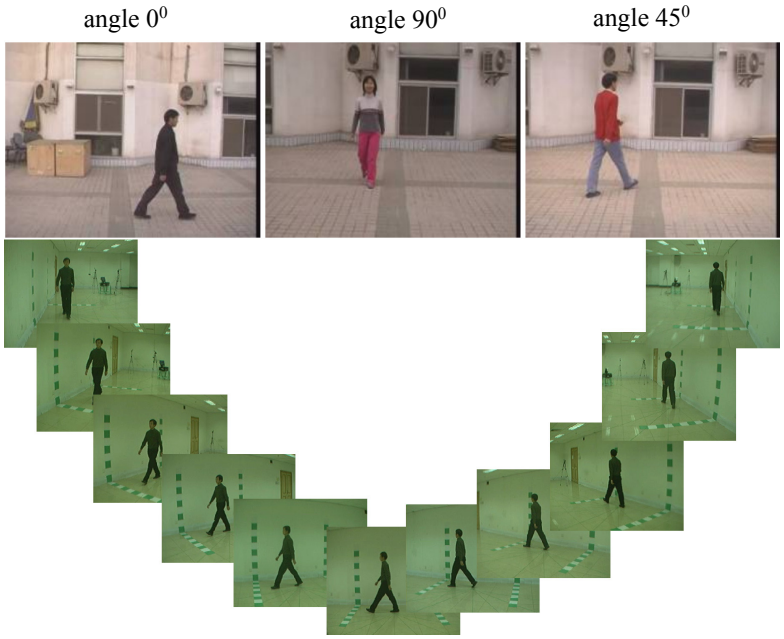


Figure 4.1: Sample video frames of CASIA A and CASIA B dataset. In top, some of the sample images from CASIA A dataset are shown where the subjects are walking along the straight line in 3 different view angles, and in bottom, CASIA B dataset is shown with its 11 view angles.

were recorded from the side view angle, this dataset is not suitable for evaluating the performance on multi-view gait recognition.

- **CMU MoBo dataset** [64]: This dataset has 25 subjects with six views and four walking styles. The main drawback of this database is that all the data is from an indoor environment (collected from a treadmill). Six cameras are positioned to cover the complete field-of-view of the walking person on the treadmill.

- **SOTON database** [25]: It contains two types of datasets a large dataset with more than 100 subjects and a small dataset with only 10 subjects. The large dataset has two viewpoints (frontal and oblique) and contains subjects in both outdoor and indoor environments and on a treadmill. The small database is used to explore gait recognition under covariates such as views, shoes, clothing, carriage, and walking speed.

- **OU-ISIR multi-view large population dataset (OU-MVLP)** [63]: The largest dataset available for gait recognition. It contains $10,307$ subjects from 14 view angles ranging from $0° - 90°$, and $180° - 270°$. Only two sequences are provided, one for the gallery and the other for the probe. But, this dataset is only formatted as a set of silhouette sequence which makes it completely different from our approach.

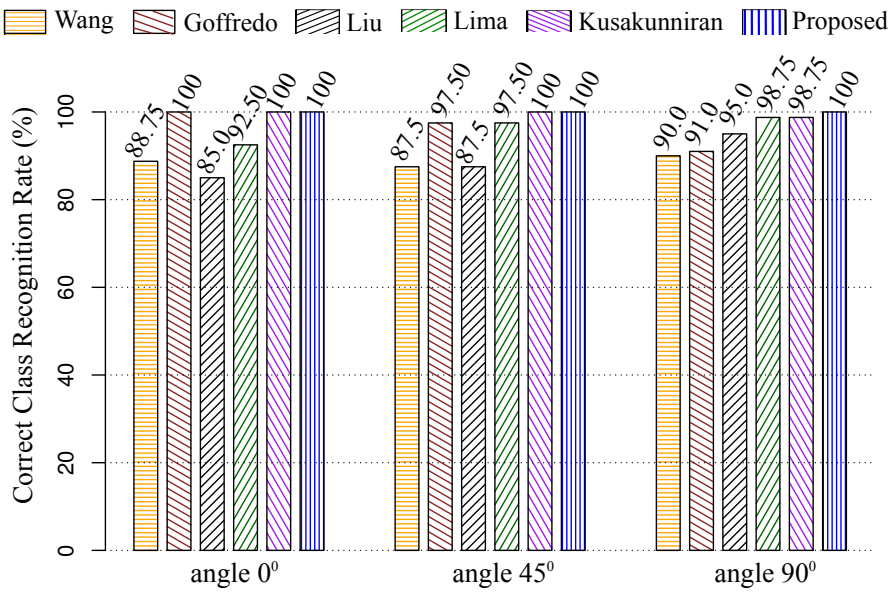


Figure 4.2: Comparison in CCR among proposed method with other prevailing gait recognition methods at different view angles on CASIA A dataset. Our method achieves **100%** CCR on all of the view angles which showed the efficacy of the proposed method.

- **CASIA (CASIA A and CASIA B) dataset** [11]: one of the largest datasets for multi-view gait recognition. CASIA A dataset contains total 20 subjects walking in an outdoor environment where CASIA B dataset contains total 124 subjects walking in an indoor environment. In CASIA A gait dataset, each subject walks along a straight line in 3 different view angles: lateral ($0°$), oblique ($45°$), and frontal ($90°$). For each view angle all subjects have total four gait sequences out of which two of them have same walking direction while the other two have opposite direction.

  In CASIA B dataset, there are 10 walking sequences for each subject: 6 sequences for normal walking ('nm'), 2 sequences for walking in coat ('cl') and 2 sequences for walking with bag ('bg') on shoulder. Therefore, this dataset separately considered three variations in people walking namely normal, clothing, and carrying condition. Also, each walking sequence were captured from 11 view angles ranging from $0°$ to $180°$. Figure 4.1 illustrates some of the sample video frames of CASIA A and CASIA B datasets.

Table 4.1: Comparison among different state-of-the-art gait recognition methods without view variation in all three view angles of CASIA A dataset. It has been observed that the proposed method achieves the highest average recognition rate, i.e., **100.0%** on all three angles and outperforms other state-of-the-art methods by a large margin.

Methods	0°	45°	90°	Mean
Wang [65]	88.75	87.50	90.00	88.75
Goffredo [66]	100.0	97.50	91.00	96.16
Liu [67]	85.00	87.50	95.00	89.17
Lima [68]	92.50	97.50	98.75	96.25
Kusakunniran [69]	100	100	98.75	99.58
Proposed	**100.0**	**100.0**	**100.0**	**100.0**

4.2 Single-View Gait Recognition

4.2.1 Experimental Evaluation on CASIA A dataset

Since, CASIA A dataset contains only 20 subjects where every subject has only four gait sequences in three different view angles, we trained a model for each of the view angle with 20 output neurons in the final softmax layer of our proposed network. To evaluate the performance of our proposed method on CASIA A dataset, we employed leave-one-out cross validation rule, i.e., one sequence was set for testing and the remainder was set for training the network. We compare our results with five other prevailing state-of-the-art gait recognition methods including Wang [65], Goffredo [66], Liu [67], Lima [68], and Kusakunniran [69] (see Figure 4.2). Table 4.1 illustrates the comparison where it is seen that the proposed method have achieved higher average correct class recognition rates (CCR) of 100.0% compared to other methods in literature.

4.2.2 Experimental Evaluation on CASIA B Dataset

4.2.2.1 Experimental Setup

We designed two experimental setups (A, B), as demonstrated in Table 4.2, for evaluating the performance in CASIA B dataset. Experimental setup A was for evaluating the performance of our proposed method in single-view gait recognition. To investigate the robustness of view variation, comparison results of the proposed method against other state-of-the-art methods in different view variations have been reported. Experiment setup B was designed for evaluating the cross-view recognition performance.

For setup A, we divided the dataset into two groups where the first group consists of

Table 4.2: Experimental setup for the CASIA B dataset. The dataset was divided into two different setups to organize two different types of experiment. The evaluation set is further divided into a gallery set and a probe set. Gallery set consists of the first 4 normal walking sequences of each subject and the probe set contains rest of the walking sequences.

Setup	Training set		Evaluation set		Sequences	
	ID	Total	ID	Total	Gallery	Probe
A	01 - 62	62	63 - 124	62	$nm01 - nm04$	$nm05 - nm06$
B	01 - 74	74	75 - 124	50		$bg01 - bg02$
						$cl01 - cl02$

62 subjects which is used to train the network. The second group contains rest of the subjects for evaluating the performance of the network. For experimental setup B, the ratio between the train and evaluation set was 24 to 100. In the evaluation set for both setup, 4 normal walking sequences of each subject are put into gallery set and rest 6 walking sequences consist three probe set (*ProbeNM*, *ProbeBG*, *ProbeCL*). *ProbeNM* consists of 2 other normal walking sequences where *ProbeBG* and *ProbeCL* consist of two sequences of carrying bag and wearing coat respectively.

4.2.2.2 Results on Single-View Gait Recognition of CASIA B Dataset without View Variation

Experimental result of single-view gait recognition on all the three probe set of CASIA B dataset without view variation is illustrated on Table 4.3. We achieved higher average recognition rate **97.80%** and **82.82%** on the probe set of (*ProbeBG*) and (*ProbeCL*) respectively. This performance proves the robustness of the proposed method toward both carrying and clothing covariate conditions. We also achieved higher average class recognition rate **99.41%** on normal walking condition.

4.2.2.3 Comparison on Single-View Gait Recognition of CASIA B Dataset with State-of-the-art Methods without View Variation

We compare our experimental results with other state-of-the-art methods such as GaitGANv2 [31], PTSN [35], PoseGait [36], and Yu *et al.* [70] as shown in Figure 4.3. The experimental setup for all these methods were set A (see Table 4.2). Table 4.4 reports that CCR of the proposed method outperforms all other methods in all three covariate conditions of CASIA B dataset. Our method achieved average CCR of **93.34%** with improvement of approximately **10%** from PTSN [35].

Table 4.3: Correct class recognition rate (CCR) of the proposed method in all three probe sets of CASIA B dataset. Here, each column represents a specific view of the gallery and probe set. It has been observed that the probe set of normal walking sequence (*ProbeNM*) achieves **99.41%** average recognition rate while the ProbeBG and ProbeCL set achieve **97.80%** and **82.82%** average recognition rates respectively.

Gallery Angle	*ProbeNM*	*ProbeBG*	*ProbeCL*
0°	100.0	100.0	81.52
18°	100.0	100.0	82.11
36°	100.0	100.0	83.58
54°	100.0	100.0	85.48
72°	100.0	98.39	84.46
90°	98.39	96.77	83.72
108°	100.0	96.77	83.28
126°	100.0	98.39	84.16
144°	100.0	98.39	83.58
162°	98.39	95.16	80.65
180°	96.77	91.93	78.45
Mean	**99.41**	**97.80**	**82.82**

4.2.2.4 Results on Single-View Gait Recognition of CASIA B Dataset with View Variation

The performance of the proposed method on single-view gait recognition with view variation is demonstrated on Table 4.5. Here, for a specific gallery (θ_g) angle the average CCR (%) of all eleven probe angles has been reported; our method achieved average CCR of 62.69%, 47.23%, and 33.46% for *ProbeNM*, *ProbeBG*, and *ProbeCL* respectively.

4.2.2.5 Comparison on Single-View Gait Recognition of CASIA B Dataset with State-of-the-art Methods with View Variation

To better illustrate the robustness of our gait recognition method to view variation, the proposed method has been compared to three other state-of-the-art methods such as GaitGANv2 [31], PoseGait [36], and Yu *et al.* [70]. It has been observed from the Figure 4.4 and Table 4.6 that the proposed method outperforms other in two covariate condition and achieves comparable performance in normal walking.

Since, to recognize gait, we consider features based on the effective body joints, our method does not get affected by the variation in covariate conditions compared to other

Table 4.4: Comparison between the proposed method and other state-of-the-art gait recognition methods in CASIA B dataset without view variation. It has been observed that the proposed method outperforms other methods at a significant margin in all three probe set of CASIA B dataset by achieving higher average CCR of **93.34%**. As our feature descriptors are discriminative and have robustness toward variation in people's appearance and shape, it shows improved performance in normal as well as covariate conditions.

Methods	*ProbeNM*	*ProbeBG*	*ProbeCL*	Average
Liao *et al.* [35]	96.92	85.78	68.11	83.60
Yu *et al.* [70]	97.58	72.14	45.45	71.72
Yu *et al.* [31]	98.24	76.25	42.89	72.46
Liao *et al.* [36]	96.63	71.26	54.18	74.02
Proposed	**99.41**	**97.80**	**82.82**	**93.34**

appearance-based method or model-based methods which consider ineffective features to build their gait descriptors. That's why our method is proven to be less sensitive to view angle variation and performs better in carrying bag and clothing condition.

4.3 Cross-View Gait Recognition

The gait recognition scheme in which gallery and probe set are getting matched from two different views is commonly known as cross-view gait recognition. In this section we are going evaluate the performance of our proposed method on cross-view gait recognition.

4.3.1 Comparison with the State-of-the-art Methods of CASIA B Dataset on Cross-View Gait Recognition

To show the effectiveness of our proposed method in cross-view gait recognition, we make the comparison between the proposed method and three other state-of-the-art methods including CNN [26], CMCC [71], and GEI-SVR [72] with the same experimental setup. The probe angles were selected $0°, 54°, 90°$, and $126°$ for comparison.

Although the proposed method contains only one model to handle any view angle variation, it achieves comparable performance with other prevailing state-of-the-art methods proposed in literature which were specially designed and trained for cross-view gait recognition. From Table 4.7, it is seen that CNN [26] achieves the highest recognition rates when the view variation is large due to the use of supervised information of all

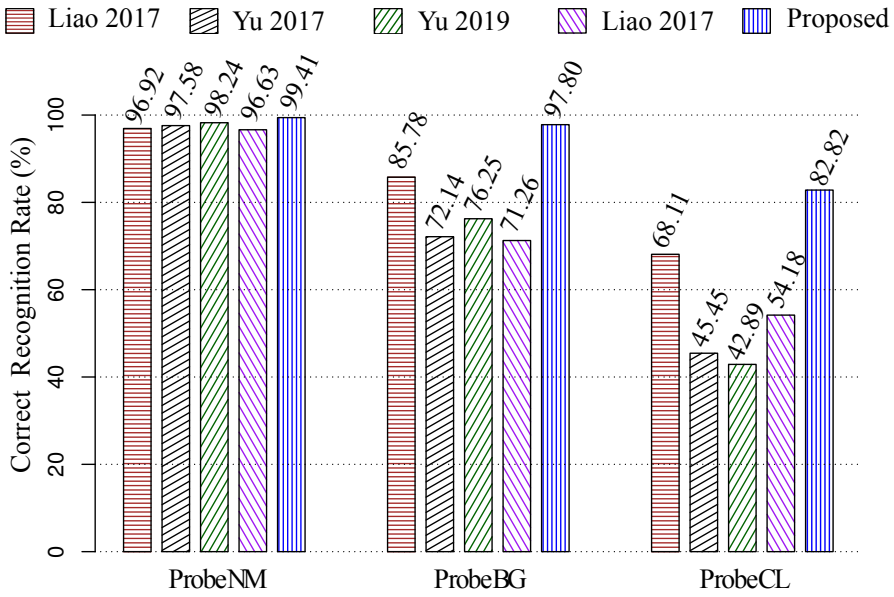


Figure 4.3: Correct class recognition rates (%) of the proposed method with other state-of-the-art methods on all three probe set of CASIA B dataset without view variation. Proposed method demonstrates better performance compared to other by achieving 89.64% and 96.45% in two covariate conditions *ProbeCL*, and *ProbeBG* of CASIA B dataset respectively. The result proves the robustness of the proposed temporal network against carrying and clothing conditions variation.

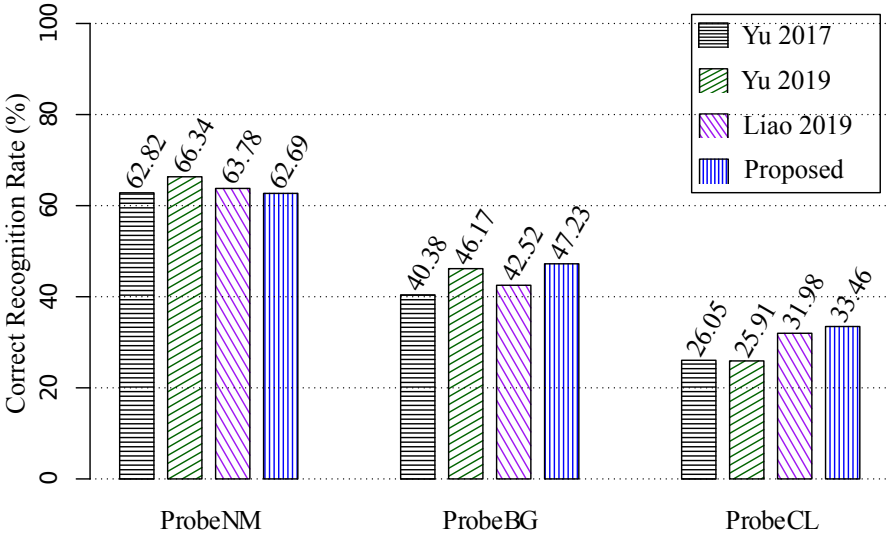


Figure 4.4: Comparison with different state-of-the-art algorithms for gait recognition with view variation in all three probe set of CASIA B dataset. Here, the value reported for each algorithm is the average of all of the gallery view's average CCR. From the comparison, it is seen that our method outperforms other state-of-the-art methods by achieving 47.23% and 33.46% averaged accuracy in two covariate conditions *ProbeBG*, and *ProbeCL* respectively.

Table 4.5: The average recognition rates for all three probe sets of CASIA B dataset. Each row represents the average value of all eleven probe angles at a specific gallery angle (θ_g) in all three probe sets.

Gallery Angle	*ProbeNM*	*ProbeBG*	*ProbeCL*
0°	61.73	45.01	32.40
18°	63.64	47.80	32.99
36°	67.30	48.97	34.46
54°	68.33	50.15	37.24
72°	68.33	50.44	39.0
90°	66.42	49.12	36.36
108°	64.22	48.39	34.75
126°	62.02	47.07	32.40
144°	58.80	47.51	31.82
162°	56.45	44.13	29.77
180°	52.35	40.91	26.83
Mean	**62.69**	**47.23**	**33.46**

Table 4.6: Comparison among different state-of-the-art methods for gait recognition with view variation in all three probe sets of CASIA B dataset. Here, each row represents the average value of all the gallery view's average recognition rate. It has been seen that, similar to first experiment, the proposed method achieves higher performance in two different probe set (*ProbeBG*, *ProbeCL*) and comparable performance in normal walking with other prevailing methods.

Methods	*ProbeNM*	*ProbeBG*	*ProbeCL*
Yu *et al.* [70]	62.82	40.38	26.05
Yu *et al.* [31]	66.34	46.17	25.91
Liao *et al.* [36]	63.78	42.52	31.98
Proposed	**62.69**	**47.23**	**33.46**

gallery angles during training.

The comparison in Table 4.7 also illustrates that the proposed method performs better when the view variation is small. The reason for not achieving better performance at large view variation is because it was trained with only one view angle of the gallery.

4.4 Multi-View Gait Recognition

In multi-view gait recognition, multiple views of gallery gaits are combined to recognize an unknown gait view. For multi-view gait recognition, we initially identify the

Table 4.7: Comparison of our proposed method with the previous best results of cross-view gait recognition at different probe angles of CASIA B dataset by CCR(%). The network was trained according to experimental setup B to have the same setup with other methods.

Probe View	Gallery View	CNN	CMCC	GEI-SVR	**Proposed**
0°	18°	95.0	85.0	84.0	**97.0**
	36°	73.5	47.0	45.0	**80.0**
54°	18°	**91.5**	65.0	64.0	83.0
	36°	98.5	97.0	95.0	**100.0**
	72°	98.5	95.0	93.0	**100.0**
	90°	**93.0**	63.0	59.0	83.0
90°	54°	–	66.0	63.0	**84.0**
	72°	**99.5**	96.0	95.0	96.0
	108°	**99.5**	95.0	95.0	95.0
	126°	–	68.0	65.0	**71.0**
126°	90°	**92.0**	78.0	78.0	76.0
	108°	**99.0**	98.0	98.0	92.0
	144°	97.0	**98.0**	**98.0**	96.0
	162°	**83.0**	75.0	74.0	77.0

Table 4.8: View angle identification rate (%) of the proposed 3D-CNN network on CASIA B dataset. The proposed view angle network successfully achieved **100%** identification accuracy in all 11 view angles of all three probe sets in CASIA B dataset.

View angle	0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°
Rate(%)	100	100	100	100	100	100	100	100	100	100	100

walking direction of a gait video using a 3D-CNN network.

In our experiment, we evaluated the proposed 3D-CNN network with all three probe sets of CASIA B dataset and have achieved **100%** identification accuracy. The experimental results proved the fact that our 3D-CNN is efficient in classifying the view angle from gait videos. Table 4.8 illustrates our test result.

4.4.1 Comparison with the State-of-the-art Methods on Multi-View Gait Recognition

To evaluate the performance of our proposed two-stage network, we compare it with the recent state-of-the-art multi-view gait recognition methods such as Dupuis *et al.* [73], Isaac *et al.* [74], and VI-MGR [75] on all three probe set of CASIA B dataset. The

Table 4.9: Comparison with other state-of-the-art methods on all three probe set of CASIA B dataset in multi-view gait recognition. From the comparison, it is been observed that proposed two-stage network achieves higher average recognition rates in 8 of 11 different probe angles.

	Methods	0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°
Normal	Dupuis	97.2	99.6	97.2	96.3	98.8	98.4	97.1	97.6	97.14	93.0	96.0
	VI-MGR	100.0	99.0	100.0	99.0	100.0	100.0	99.0	99.0	100.0	100.0	99.0
	Isaac	98.5	99.0	99.0	97.0	97.5	96.0	95.0	97.5	94.0	93.9	99.0
	Proposed	100.0	100.0	100.0	100.0	100.0	98.4	100.0	100.0	100.0	98.4	96.8
Bag	Dupuis	73.2	74.1	74.7	76.3	78.5	75.8	76.3	76.7	73.4	73.2	74.6
	VI-MGR	93.0	89.0	89.0	90.0	77.0	80.0	82.0	84.0	92.0	93.0	89.0
	Isaac	95.0	98.5	96.5	96.0	97.5	93.5	93.5	94.0	92.5	91.3	94.4
	Proposed	100	100	100	100	98.39	96.77	96.77	98.39	98.39	95.16	91.93
Coat	Dupuis	81.64	87.39	86.29	84.34	89.96	91.86	89.50	85.04	72.24	78.40	82.70
	VI-MGR	67.0	56.0	70.0	80.0	71.0	75.0	77.0	75.0	65.0	64.0	66.0
	Isaac	97.0	99.5	97.5	94.0	88.0	90.5	89.5	94.5	92.0	91.3	94.0
	Proposed	81.52	82.11	83.58	85.48	84.46	83.72	83.28	84.16	83.58	80.65	78.45

comparison, as illustrated in Table 4.9 and Figure 4.5, shows that the proposed method exceeds the previous best in result all three probe set by a significant margin. It outperforms other in total **8** of 11 view angles.

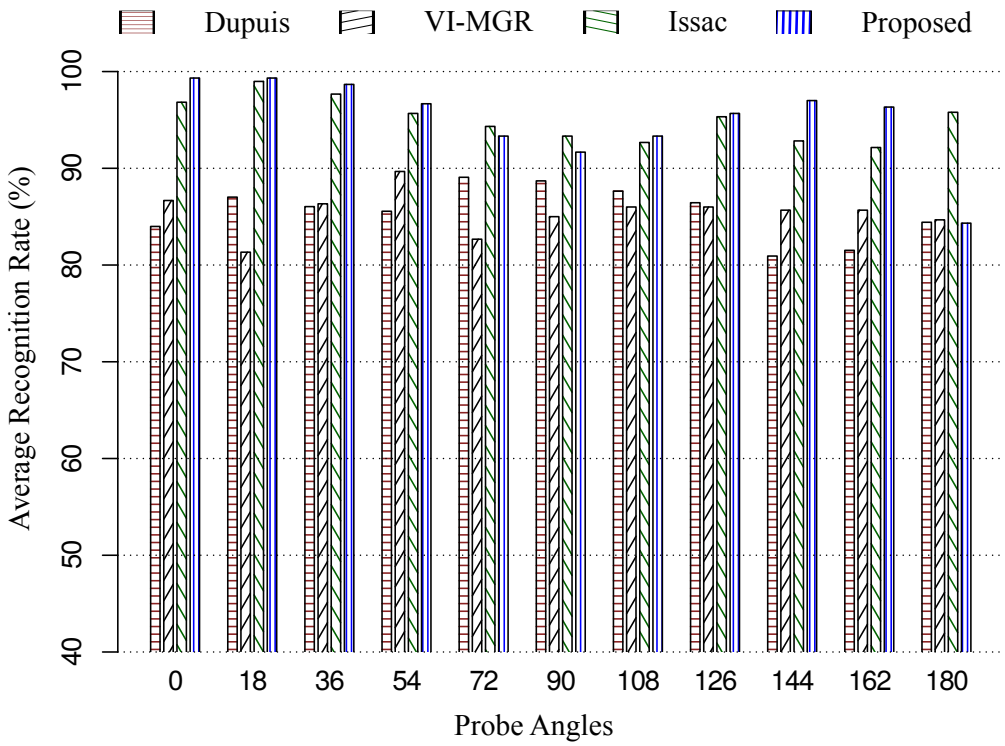


Figure 4.5: Comparison on average recognition rates (%) between the proposed method with other state-of-the-art methods in multi-view gait recognition. Our method achieves higher average recognition accuracy on **8** of total 11 view angles of CASIA B dataset.

Chapter 5

Conclusion

5.1 Summary of Our Work

In this thesis, a novel feature extraction techniques were proposed from 2D human pose estimation to find the effective and discriminative gait features for robust gait recognition. We also present a novel RNN architecture which is much more simple, efficient and computationally inexpensive compared to the existing architectures proposed in literature. We considered human pose information as gait features for our network because it not only has rich gait representation capacity but also shows robustness toward the variation of carrying and clothing condition. Experimental results on challenging CASIA A and CASIA B gait dataset clearly depicts that the method proposed in this thesis outperforms the existing state-of-the-art methods in literature.

5.2 Future Prospects of Our Work

In future, we will employ more accurate pose estimation algorithm that can improve the recognition rate greatly especially in a large view variation. Thus, it will further boost our performance and lead us to achieve state-of-the-performance in cross-view gait recognition. Using a larger dataset containing thousands of subject will help us to develop a more stable network suitable for practical applications like real-time surveillance.

Bibliography

[1] J. E. Boyd and J. J. Little, "Biometric gait recognition," in *Bometrics School 2003, LNCS 3161*. Springer-Verlag Berlin Heidelberg, 2005, pp. 19–42.

[2] C. Benabdelkader, R. Cutler, and L. S. Davis, "Person identification using automatic height and stride estimation," in *Object recognition supported by user interaction for service robots*. Quebec City, Quebec, Canada, 2002, pp. 377–380.

[3] M. Pistacchi, M. Gioulis, F. Sanson, E. D. Giovannini, G. Filippi, and et al., "Gait analysis and clinical correlations in early parkinsons disease," *Functional Neurology*, vol. 32, no. 1, pp. 28 – 34, January 2017.

[4] J. Juen, Q. Cheng, V. Prieto-Centurion, J. A. Krishnan, and B. Schatz, "Health monitors for chronic disease by gait analysis with mobile phones," *Telemed J E Health*, vol. 20, no. 11, pp. 1035– 1041, November 2014.

[5] T. K. M. Lee, M. Belkhatir, and S. Sanei, "A comprehensive review of past and present vision-based techniques for gait recognition," *Multimedia Tools and Applications*, no. 3, p. 28332869, October 2014.

[6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, and R. Sukthankar, "Large-scale video classification with convolutional neural networks," in *IEEE Conf. on Computer Vision and Pattern Recognition*. Columbus, OH, USA, 2014, pp. 1725–1732.

[7] Z. Cao, G. Hidalgo Martinez, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *IEEE Trans. on Pattern Anal. Mach. Intell.*, July 2019.

[8] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," vol. 1, no. 2, 2017, pp. 4263 – 4270.

[9] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*. Boston, MA, USA, 2015, pp. 1110 –1118.

[10] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and *et al.*, "Deep captioning with multimodal recurrent neural networks (m-rnn)," in *International Conference on Learning Representations*, 2015.

[11] 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 Int. Conf. on Pattern Recognition*. Hong Kong, China, 2006, pp. 441–444.

[12] I. Rida, N. Almaadeed, and S. Almaadeed, "Robust gait recognition: a comprehensive survey," *IET Biometrics*, vol. 8, no. 1, pp. 14 – 28, January 2019.

[13] Z. Liu and S. Sarkar, "Simplest representation yet for gait recognition: averaged silhouette," in *17th International Conference on pattern Recognition*. Cambridge, UK, 2004, pp. 211 – 214.

[14] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 28, no. 2, pp. 316–322, February 2006.

[15] K. Bashir, T. Xiang, and S. Gong, "Gait recognition using gait entropy image," in *3rd Int. Conf. on Imaging for Crime Detection and Prevention*. London, UK, 2009.

[16] T. H. W. Lam, K. H. Cheung, and J. N. K. Liu, "Gait flow image: A silhouette-based gait representation for human identification," *Pattern Recognition*, vol. 44, no. 4, pp. 973 – 987, April 2011.

[17] C. Wang, J. Zhang, L. Wang, J. Pu, and X. Yuan, "Human identification using temporal information preserving gait template," *IEEE Trans Pattern Anal Mach Intell.*, vol. 11, no. 34, pp. 2164–2176, November 2012.

[18] X. Huang and N. V. Boulgouris, "Gait recognition with shifted energy image and structural feature extraction," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2256 – 2268, April 2012.

[19] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. Bowyer, "The humanid gait challenge problem: Data sets, performance, and analysis," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 162–177, February 2005.

[20] C. Yam, M. S. Nixon, and J. N. Carter, "Automated person recognition by walking and running via model-based approaches," *Pattern Recognition*, vol. 37, no. 5, pp. 1057 – 1072, May 2004.

[21] J. Gu, X. Ding, S. Wang, and Y. Wu, "Action and gait recognition from recovered 3-d human joints," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 4, pp. 1021–1033, 2010.

[22] D. K. Wagg and M. S. Nixon, "On automated model-based extraction and analysis of gait," in *6th IEEE International Conference on Automatic Face and Gesture Recognition*. Seoul, South Korea, 2004, pp. 11 – 16.

[23] V. Pavlovic, J. M. Rehg, T.-J. Cham, and K. P. Murphy, "A dynamic bayesian network approach to figure tracking using learned dynamic models," in *Seventh IEEE International Conference on Computer Vision*. Kerkyra, Greece, 1999, pp. 94 – 101.

[24] M. Isard and B. A, "Condensation-conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5 – 28, 1998.

[25] J. D. Shutler, M. G. Grant, M. S. Nixon, and C. J. N, "On a large sequence-based human gait database," in *Applications and Science in Soft Computing*. Springer, Berlin, Heidelberg, 2004, pp. 339 – 346.

[26] 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 Trans. on Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 209 – 226, February 2017.

[27] K. Shiraga, Y. Makihara, D. Muramatsu, T. Echigo, and Y. Yagi, "Geinet: View-invariant gait recognition using a convolutional neural network," in *Int. Conf. on Biometrics (ICB)*. Halmstad, Sweden, 2016.

[28] T. Wolf, M. Babaee, and G. Rigoll, "Multi-view gait recognition using 3d convolutional neural networks," in *IEEE Int. Conf. on Image Processing*. Phoenix, AZ, USA, 2016, pp. 4165–4169.

[29] C. Zhang, W. Liu, H. Ma, and H. Fu, "Siamese neural network based gait recognition for human identification," in *IEEE Int. Conf. On Acoustics, Speech and Signal Processing*. Shanghai, China, 2016, pp. 2832 – 2836.

[30] S. Yu, H. Chen, E. B. G. Reyes, and N. Poh, "Gaitgan: invariant gait feature extraction using generative adversarial networks," in *IEEE Conf. on Computer*

Vision and Pattern Recognition Workshops. Honolulu, HI, USA, 2017, pp. 532 – 539.

[31] S. Yu, R. Liao, W. An, H. Chen, E. B. Garcia, and a. Huang, Y, "Gaitganv2: Invariant gait feature extraction using generative adversarial networks," *Pattern Recognition*, vol. 87, no. 11, pp. 179 – 189, March 2019.

[32] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *IEEE Conf. on Computer Vision and Pattern Recognition*. Las Vegas, NV, USA, 2016.

[33] Y. Cai and X. Tan, "Weakly supervised human body detection under arbitrary poses," in *2016 IEEE International Conference on Image Processing (ICIP)*. Phoenix, AZ, USA, 2016.

[34] Y. Feng, Y. Li, and J. Luo, "Learning effective gait features using lstm," in *23rd Int. Conf. on Pattern Recognition*. Cancun, Mexico, 2016, pp. 325–330.

[35] R. Liao, C. Cao, E. B. Garcia, S. Yu, and Y. Huang, "Pose-based temporal-spatial network (ptsn) for gait recognition with carrying and clothing variations," in *Chinese Conf. on Biometric Recognition*, 2017, pp. 474 – 483.

[36] R. Liao, S. Yu, W. An, H. Chen, and Y. Huang, "A model-based gait recognition method with body pose and human prior knowledge," *Pattern Recognition*, February 2020.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 2012.

[38] N. Kalchbrenner and p. Blunsom, "Recurrent continuous translation models," in *EMNLP*, 2013.

[39] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[40] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *IEEE Int. Conf. on Computer Vision*. Santiago, Chile, 2015, pp. 4489–4497.

[41] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The journal of Physiology*, vol. 195, no. 1, pp. 215–243, May 1968.

[42] C. Olah, "Understanding lstm networks," 2015.

[43] T. Mikolov, "Statistical language models based on neural networks," *Ph. D. thesis, Brno University of Technology*, 2012.

[44] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[45] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*. IEEE Press, 2001.

[46] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, and *et al.*, "Learning phrase representations using rnn encoderdecoder for statistical machine translation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar, 2014, pp. 1724–1734.

[47] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December*, December 2014.

[48] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. on Signal Proces.*, vol. 45, no. 11, pp. 2673 – 2681, 1997.

[49] A. Graves, M. Liwicki, H. Bunke, J. Schmidhuber, and S. Fernndez, "Unconstrained on-line handwriting recognition with recurrent neural networks," in *Advances in neural information processing systems*, 2008, pp. 577–584.

[50] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: A hands-on survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 12, pp. 2633 – 2651, December 2016.

[51] S.-R. Ke, L. Zhu, J.-N. Hwang, H.-I. Pai, K.-M. Lan, and C.-P. Liao, "Real-time 3d human pose estimation from monocular view with applications to event detection and video gaming," in *7th IEEE International Conference on Advanced Video and Signal Based Surveillance*. Boston, MA, USA, 2010, pp. 489–496.

[52] D. Cunado, M. S. Nixon, and J. N. Carter, "Using gait as a biometric, via phase-weighted magnitude spectra," in *Int. Conf. on Audio-and Video-Based Biometric Person Authentication*. Berlin, Heidelberg, 1997, pp. 93–102.

[53] L. Wang, H. Ning, T. Tan, and W. Hu, "Fusion of static and dynamic body biometrics for gait recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 2, pp. 149–158, March 2004.

[54] R. Araujo, G. Graa, and V. Andersson, "Towards skeleton biometric identification using the microsoft kinect sensor," in *ACM Symposium on Applied Computing*. Coimbra, Portugal, 2013, pp. 21–26.

[55] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. Conf. on Machine Learning*. Lille, France, 2015, pp. 448 – 456.

[56] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[57] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conf. on Computer Vision*, 2016, pp. 499 – 515.

[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd Int. Conf. on Learning Representations*. San, Diego, 2015.

[59] F. Chollet. (2015) Keras. [Online]. Available: https://github.com/keras-team/keras/

[60] OpenCV. (2015) Open source computer vision library. [Online]. Available: https://github.com/opencv/opencv

[61] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[62] M. Hofmann, J. Geiger, S. Bachmann, B. Schuller, and G. Rigoll, "The tum gait from audio, image and depth (gaid) database: Multimodal recognition of subjects and traits," *Journal of Visual Com. and Image Representation*, vol. 25, no. 1, pp. 195–206, January 2014.

[63] T. Noriko, Y. Makihara, D. Muramatsu, T. Echigo, and Y. Yagi, "Multi-view large population gait dataset and its performance evaluation for cross-view gait recognition," *IPSJ Trans. on Computer Vision and Applications*, vol. 10, no. 1, p. 4, February 2018.

[64] R. Gross and J. Shi, "The cmu motion of body (mobo) database," Pittsburgh, PA, June 2001.

[65] L. Wang, T. Tieniu, W. Hu, and H. Ning, "Automatic gait recognition based on statistical shape analysis," *IEEE Trans. on Image Process.*, vol. 12, no. 9, pp. 1120 – 1131, September 2003.

[66] M. Goffredo, J. N. Carter, and M. S. Nixon, "Front-view gait recognition," in *Biometrics: Theory, Applications and Systems*. Arlington, VA, USA, 2008, pp. 1 – 6.

[67] D. Liu, M. Ye, X. Li, F. Zhang, and L. Lin, "Memory-based gait recognition," in *British Machine Vision Conf.* BMVA Press, 2016, pp. 82.1 – 82.12.

[68] V. C. de Lima and W. R. Schwartz, "Gait recognition using pose estimation and signal processing," in *Iberoamerican Congress on Pattern Recognition*. BMVA Press, 2019, pp. 719 – 728.

[69] W. Kusakunniran, Q. Wu, H. Li, and J. Zhang, "Automatic gait recognition using weighted binary pattern on video," in *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*. Genova, Italy, 2009, pp. 49 – 54.

[70] S. Yu, H. Chen, Q. Wang, L. Shen, and Y. Huang, "Invariant feature extraction for gait recognition using only one uniform model," *Neurocomputing*, vol. 239, no. C, pp. 81 – 93, May 2017.

[71] W. Kusakunniran, Q. Wu, J. Zhang, H. Li, and L. Wang, "Recognizing gaits across views through correlated motion co-clustering," *IEEE Trans. on Image Process.*, vol. 23, no. 2, pp. 696 – 709, February 2014.

[72] W. Kusakunniran, Q. Wu, J. Zhang, and H. Li, "Support vector regression for multi-view gait recognition based on local motion feature selection," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*. San Francisco, CA, USA, 2010, pp. 974 – 981.

[73] Y. Dupuis, S. Xavier, and V. Pascal, "Feature subset selection applied to model-free gait recognition," *Image and Vision Computing*, vol. 31, no. 8, pp. 580 – 591, 2013.

[74] E. R. Isaac, S. Elias, S. Rajagopalan, and K. S. Easwarakumar, "View-invariant gait recognition through genetic template segmentation," *IEEE Signal Process. Letters*, vol. 24, no. 8, pp. 1188 – 1192, June 2017.

[75] S. D. Choudhury and T. Tjahjadi, "Robust view-invariant multiscale gait recognition," *Pattern Recognition*, vol. 48, no. 3, pp. 798 – 811, March 2015.