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

By

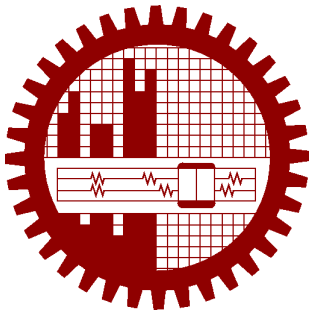
Md Mahedi Hasan

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<br>Assistant Professor<br>IICT, BUET, Dhaka | Chairman<br>(Supervisor) |
| Dr. Md. Saiful Islam<br>Director and Professor<br>IICT, BUET, Dhaka | Member<br>(Ex-officio) |
| Dr. Md. Liakot Ali<br>Professor<br>IICT, BUET, Dhaka | Member |
| Name of the Supervisor<br>Designation<br>Address | Member |
| Name of the Supervisor<br>Designation<br>Address | Member<br>(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 which consider the same descriptors across the camera network regardless the dramatic change in view angle and pose of a person 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 environment. 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 based on 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. Also, 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 with 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

**ANN** Artificial Neural Networks. 11

**BiGRU** Bidirectional Gated Recurrent Unit. 26, 27

**BLSTM** Bidirectional Long Short-Term Memory. 27

**BN** Batch Normalization. 27

**BPTT** Backpropagation Through Time. 12

**BRNN** Bidirectional Recurrent Neural Network. 17

**CCR** Correct Class Recognition. ix, 35–39, 41

**CE** Cross-Entropy. 28

**CL** Center Loss. 28

**CNN** Convolutional Neural Network. 11

**DL** Deep Learning. 10, 11

**DNN** Deep Neural Network. 11

**GRU** Gated Recurrent Unit. 4, 16, 17, 26, 27

**LSTM** Long Short-Term Memory. 14, 16, 17, 27

**ML** Machine Learning. 10

**MSE** Mean Squared Error. 32

**RNN** Recurrent Neural Network. 3–5, 11, 12, 26, 27

**SGD** Stochastic Gradient Descent. 32

# Chapter 1

# Introduction

## 1.1 Person re-identification

In recent years, there has been a great effort by the computer vision and artificial intelligence (AI) communities to develop an intelligent video surveillance systems capable of real-time monitoring and alerting. Person re-identification, a fundamental task in 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. most significant problems in computer vision and surveillance systems. Besides security and surveillance it has a lot of applications in authentication, human computer interaction, cross-camera person tracking, human behavior and activity analysis

Person re-identification (ReID) remains one of the challenging task in real-time surveillance due to large variation in camera view-angle, pose and illumination, partially or complete occlusions, and subject intrinsic variations. However, among these the viewpoint variation is one of the most challenging problems which increases at the same time the intra-class variation and the inter-class confusion.

A basic human re-identification scenario can be shown 1.1

## 1.2 Gait Recognition

### 1.2.1 Context

Biometrics refers to automatic identification or authentication of people by analyzing their physiological and behavioral characteristics. Physiological biometrics is related to

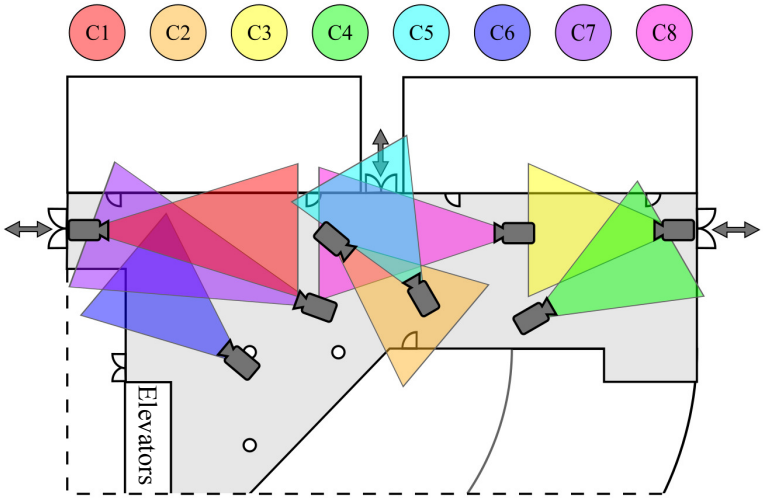



Figure 1.1: A basic person multi-camera person re-identification scenario. A person should have the same label when walking through the multiple surveillance camera network.

the shape of body parts such as 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.

### 1.2.2 Definition

Gait can be defined as *the coordinated, cyclic combination of movements that result in human locomotion* [1]. The movements in a gait repeat as 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.

Gait recognition is a behavioral biometric modality that identifies a person based on the gait pattern. In contrast to other biometrics such as face and fingerprint, it is a non-invasive technique for identifying an individual which is hard to copy. A unique advantage of gait as a biometric is that it offers recognition at a distance and at low-resolution images; 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 and the method is not vulnerable to spoof-

ing attacks and signature forgery.

### 1.2.3 Challenges

Due to the advantages of gait recognition, the past two decades have witnessed significant improvements of its algorithms. However, unfortunately, there still exist many challenges that need to be addressed for robust gait recognition. It has been observed that the performance of gait recognition is highly affected by different intraclass variations in people's appearance such as clothing and carrying variation, and in environment such as variations in illumination, walking surface, and view angle, etc. These factors can drastically reduce the performance of 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. The variation of people appearance and viewing angle at different cameras, varying lighting, and occlusion, however, 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.

## 1.4 Objectives of the Thesis

The objective of this thesis is to design a gait recognition system for human 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 like video classification [2], pose estimation [3], and action recognition [4, 5], etc. Furthermore, advancement on human body pose estimation can significantly assist in accurately modeling different body parts required for model-based gait recognition. On the other hand, recurrent neural networks 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 sequence. RNNs have been successfully employed to achieve state-of-the-art results in many vision-based tasks like human emotion detection and action recognition.

In this work, we propose a model-based gait recognition method where we consider human 2D pose data as our effective gait features. As body pose is proven not to depend on people body appearance and shape, 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 from 2D RGB images often require multiple views, and hence multiple cameras, rendering the technique unsuitable for surveillance. Again, recovering 3D pose from a single RGB images is an ill-posed problem and often causes large pose estimation errors.

Compared to other gait covariate, view is the most important factor severely affecting 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, where in 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 key to 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 of-

ten lead to a high dimensional feature space, which can be computationally expensive to map. In this work, 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 kinds of features which are robust to view variation. We demonstrate the effectiveness of our proposed method through extensive experiments on two public benchmark datasets: the CASIA A and CASIA B gait dataset [6]. 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 methods proposed in the literature.

For multi-view gait recognition, we also propose a two-stage network in which we first determine the walking direction, i.e. the viewpoint 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. Our proposed two-staged network is far simpler and efficient in terms of time and space while outperforming present state-of-the-art networks on multi-view gait recognition.

## 1.6 Contributions

In summary the contributions of this thesis are fourfold:

- We propose 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.
- We consider 2D coordinates of body pose to design a novel low-dimensional gait feature descriptor 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.

## 1.7 Thesis Outline

In the rest of this thesis, we present the details of our approach to human identification based on robust gait recognition. Here

- **Chapter 2** reviews the existing literature gait recognition and basic concepts of recurrent neural network, focused on their use in our models.
- **Chapter 3** describes our proposed network along with the steps regarding features extraction, preprocessing and network architecture for both single-view and multi-view gait recognition.. It also presents learning strategies of these models.
- **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** provides a conclusion and presents possible future directions.

# Chapter 2

# Literature Review

Over the last two decades, several methods have been studied to develop a robust gait recognition system [7]. However, robust recognition is still challenging due to the presence of large intraclass variations in a person's gait which substantially changed the performance. In this chapter, we briefly discuss the literature of the two categories of existing gait recognition techniques: appearance-based and model-based methods. Next, we review some of the recent deep learning-based gait recognition approaches which are closely related to our work.

## 2.1 Appearance-based Methods

Most of the previous work following this approach [8–10] used human silhouette masks as the main source of information and extracted features that show how these mask change. The most popular gait representation employed in such work is gait energy image (GEI) [8], 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) [9], and gait flow image (GFI) [10], 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.* [11] 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. In [12], GaitSet has been proposed where a gait is regarded as a set consisting

of independent frames rather than a template or sequence. Though it handled cross-view conditions very well, it is not good enough in handling cross-carrying and cross-clothing conditions.

These appearance-based methods in gait recognition are sensitive to the covariate factors since 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 appearance-based methods severely degrades. Therefore, these methods are not completely robust toward these covariate change.

## 2.2 Model-based Methods

In contrast, model-based [13–16] gait recognition exploits features based on the shape of human body parts and the dynamics of the motion of each of these parts. The salient advantage of the model-based approach is that, as opposed to silhouette-based approaches, it can efficiently handle many covariate changes such as view angle, body appearance and shape, so, these methods show robustness toward these variations.

These methods are based on the extraction and modeling of the human body structure as well as the local movement pattern of these parts. Therefore, this approach is often built with a structural and a motion model to capture both static as well as dynamic information of gait. For example, In [13], Yam *et al.* developed an automated model-based approach to recognize people using walking as well as running gait by analyzing the leg motion. They used the Biomechanics of human locomotion and coupled oscillators and employed a bilateral symmetric and an analytical model to successfully extract the leg motion. Ariyanto *et al.* [14] employed a structural model including articulated cylinders for fitting the 3D volumetric subject data at each joint to model the lower legs. In [15], authors presented a model-based approach where they captured the discriminatory features of gait by analyzing the leg and arm movements. For recognition, they used K-nearest neighbor classifier and Fourier components of the joint angle.

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 which is 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. Several CNN-based gait recognition methods [17–22] 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 using train samples. Wu *et al.* [17] performed cross-view gait recognition by developing three convolutional layer network using the subject's GEI as input. Shiraga *et al.* [18] designed a eight-layered CNN network, GEINet, which consist of two sequential triplets of convolution, pooling, normalization layers, and two fully connected layers for large-scale gait recognition on OU-ISIR database.

In [19], Wolf *et al.* used 3D convolutions for multi-view gait recognition by capturing spatio-temporal features from raw images and optical flow information. A Siamese neural network-based gait recognition system has been developed in [20] where GEI was feed as input. In [21], Yu *et al.* used generative adversarial nets to design a feature extractor in order to learn the invariant features. In [22], 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. The task of pose estimation mainly involves localizing the keypoints of human figure to estimate the locations of different body parts [3, 23].

Authors in [23] 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.* [3] developed a deep CNN-based regression method to estimate the association between anatomical parts in the image. Their bottom-up method achieved state-of-the-art performance on multiple benchmark datasets.

In this work, we employed their pretrained model to get an accurate 2D pose estimation

on our experimental dataset.

## 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 [16,24,25] due to its effective representation of gait features and robustness toward covariate condition variations. Feng *et al.* [16] 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 [24], Liao et al. constructed a temporal-spatial network (PTSN) to extract the spatial-temporal features of gait from 2D human pose information. Authors in [25], employed 3D pose estimation in their PoseGait network to extract the spatial-temporal gait features and achieved better performance compared with 2D pose estimation.

Again, some of the most successful approaches for human action recognition employ RNNs [4, 5] to effectively model the temporal sequences of human skeleton data. Song *et al.* [4] proposed an end-to-end spatial and temporal attention model with LSTM for human action recognition from skeleton data. In [5], 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 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. Deep learning 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 a unified framework.

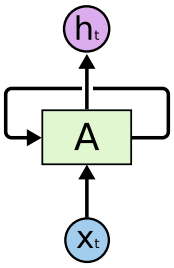



Figure 3.1: A Recurrent Neural Network (RNN). [Image courtesy Chris Olah [29]]

DL architectures are divided into two broad categories: (1) Unsupervised learning approaches including Restricted Boltzmann Machines (RBM), Deep Autoencoders, and Generative Adversarial Networks (GAN), (2) Supervised learning approaches including deep neural networks (DNN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN).

Some of the real-world applications of Deep Learning include image captioning, machine translation, text summarization, 3D-image reconstruction and text generation. 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 Recurrent Neural Network

The Recurrent Neural Network (RNN) is neural sequence model that achieves state of the art performance on important tasks that include language modeling [26], speech recognition [27], and machine translation [28].

It is well known that the areas of the brain are typically connected both in a *feedforward*, and in a *feedback* fashion. This is believed to help processing temporal data and allow for an iterative refinement of the computation. Similarly, Artificial Neural Networks (ANNs) are not constrained to process the input data in a feedforward way. Recurrent Neural Networks (RNNs) implement feedback loops (see Figure 3.1) that propagate some information from one step to the next. It is customary to refer these steps as time-steps, as RNNs are often considered in the context of a discretized time evolving domain, but nothing prevents from using RNNs with any kind of sequential data.

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

An RNN applies the same model to each time step of the sequence or, equivalently, applies different models at each time step, which share their weights. This is similar to

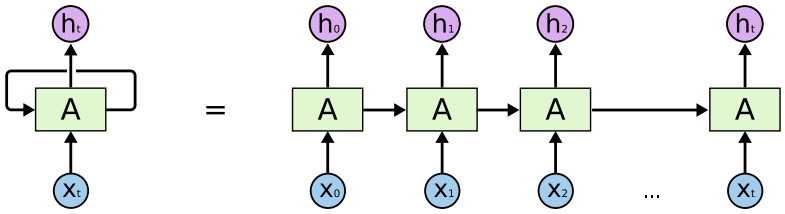



Figure 3.2: A Recurrent Neural Network unrolled for $t\%$ steps. [Image courtesy Chris Olah [29]]

what CNNs do over space with convolutions, but is rather done over time with feedback connections.

Let's derive the forward propagation equations for the RNN depicted in Figure 3.2. 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:

$$\begin{aligned} \mathbf{a}^{(t)} &= \mathbf{b} + \mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)} \\ \mathbf{h}^{(t)} &= \tanh(\mathbf{a}^{(t)}) \\ \mathbf{o}^{(t)} &= \mathbf{c} + \mathbf{V}\mathbf{h}^{(t)} \\ \hat{\mathbf{y}}^{(t)} &= softmax(\mathbf{o}^{(t)}) \end{aligned} \tag{3.1}$$

where the parameters are the bias vectors $\mathbf{b}$ and $\mathbf{c}$ along with the weight matrices $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$, respectively, for input-to-hidden, hidden-to-output and hidden-to-hidden connections.

The activation of an RNN (see Figure 3.2) 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.

To train it suffices to unroll the computation graph and use the backpropagation algorithm to proceed from the most recent time step, 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 time step to $t = 0$ to propagate the gradients through time. 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. These two issues, i.e., going to infinity and becoming zero, are known in the literature as *exploding gradient problem*

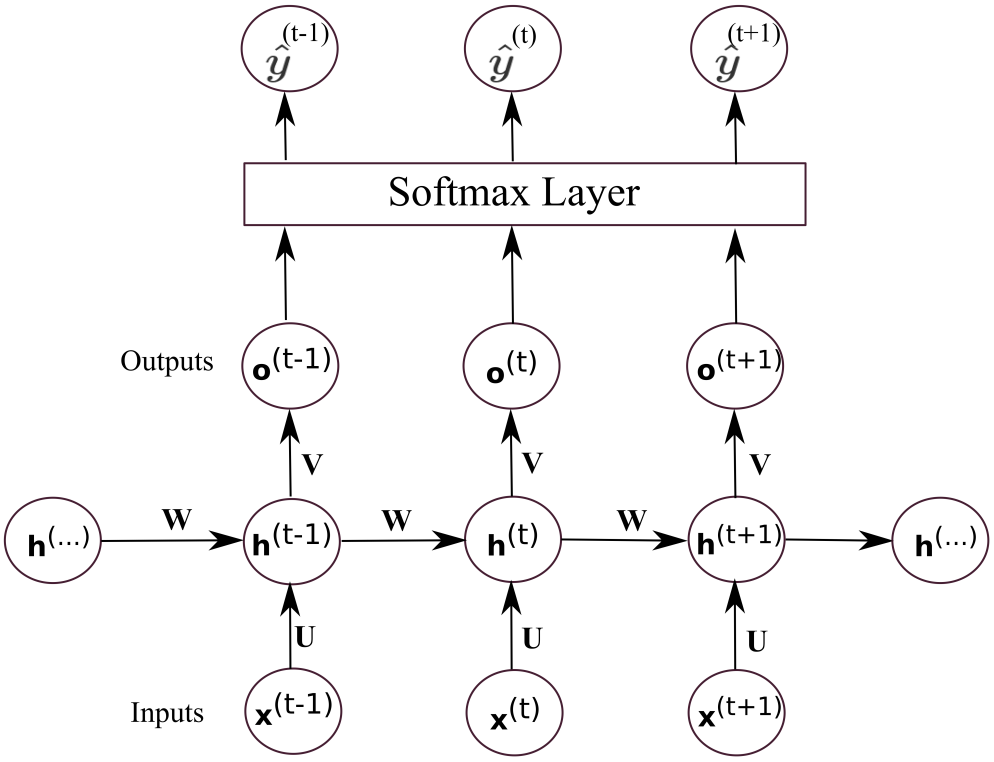



Figure 3.3: The computational graph of a unrolled recurrent network that maps an input sequence of $x$ values to a corresponding sequence of output $o$ values

and *vanishing gradient problem* [30] respectively, and have been studied extensively in the past, see e.g., [31]. 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 if not impossible.

### 3.1.3 Long Short Term Memory

Long Short Term Memory (LSTM) networks (see Figure 3.4) have been proposed to solve (or at least alleviate) the problems of RNNs in modeling long term dependencies. LSTMs have been designed to have an internal memory, or *state*, that can be updated and consulted at each time step. As opposed to vanilla RNNs, this allows LSTMs to separate their output from the information they want to carry over to future steps.

Figure 3.5 highlights the internal memory path. It can be seen how the internal memory of the previous time step $\mathbf{c}_{t-1}$ is carried over to the current time step, where it is updated through a multiplicative and an additive interaction and concurs to determine the current state of the memory $\mathbf{c}_t$. This is then, once again, propagated to the next time step.

LSTMs interact with memory through *gates*, computational nodes that determine the

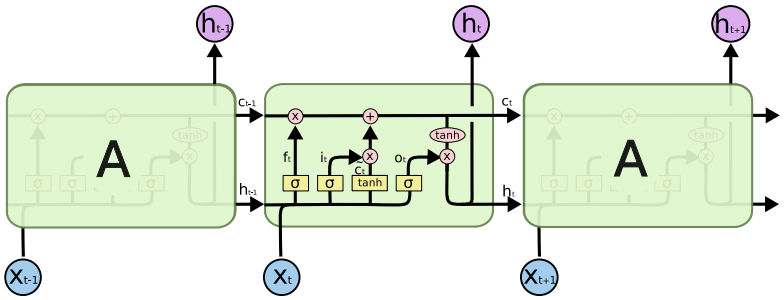



Figure 3.4: A Long Short Term Memory (LSTM). [Image courtesy Chris Olah [29]]

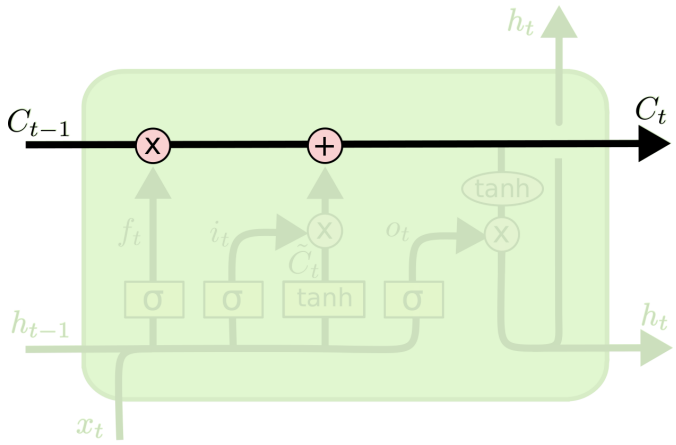



Figure 3.5: The internal state of LSTMs. [Image courtesy Chris Olah [29]]

behavior of the model. The *forget gate* (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 (depicted as $\sigma$) 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 1

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f \cdot \mathbf{h}_{t-1} + \mathbf{W}_f \cdot \mathbf{x}_t + \mathbf{b}_f\right), \tag{3.2}$$

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. Symmetrically, LSTMs have a mechanism to add new information to the memory. This behavior is controlled by an *input gate* (see 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 ac-

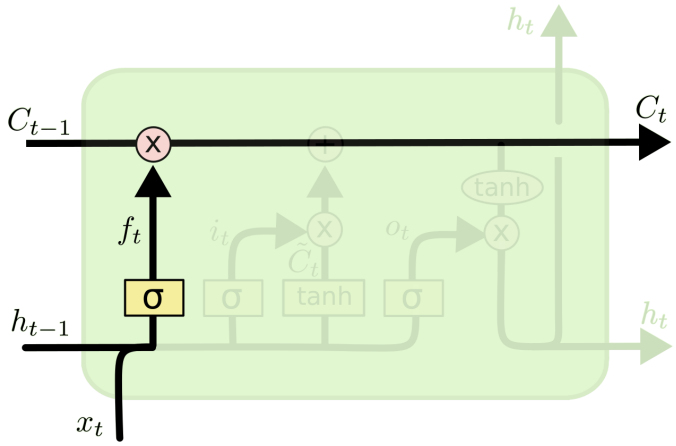



Figure 3.6: The LSTM forget gate. [Image courtesy Chris Olah [29]]

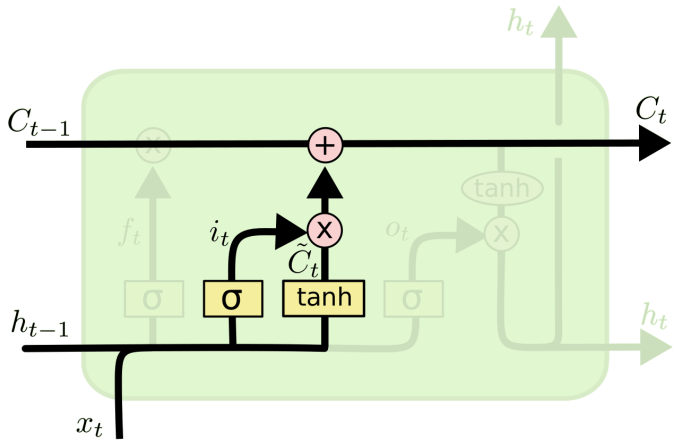



Figure 3.7: The LSTM input gate. [Image courtesy Chris Olah [29]]

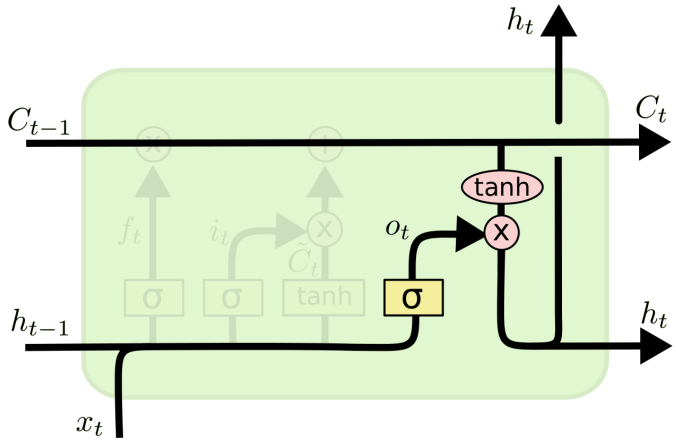



Figure 3.8: The LSTM output gate.[Image courtesy Chris Olah [29]]

tivation 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:

$$\begin{aligned} \mathbf{i}_t &= \sigma\left(\mathbf{W}_i \cdot \mathbf{h}_{t-1} + \mathbf{W}_i \cdot \mathbf{x}_t + \mathbf{b}_i\right), \\ \tilde{\mathbf{c}}_t &= tanh\left(\mathbf{W}_c \cdot \mathbf{h}_{t-1} + \mathbf{W}_c \cdot \mathbf{x}_t + \mathbf{b}_c\right). \end{aligned} \tag{3.3}$$

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 $t$ through

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t. \tag{3.4}$$

The last gate of LSTMs 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

$$\begin{aligned} \mathbf{o}_t &= \sigma\left(\mathbf{W}_o \cdot \mathbf{h}_{t-1} + \mathbf{W}_o \cdot \mathbf{x}_t + \mathbf{b}_o\right), \\ \mathbf{h}_t &= \mathbf{o}_t \circ tanh\left(\mathbf{c}_t\right), \end{aligned} \tag{3.5}$$

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. Putting it all together, the equations that govern the behavior of an LSTM are

$$\begin{aligned} \mathbf{i}_t &= \sigma\left(\mathbf{W}_i \cdot \mathbf{h}_{t-1} + \mathbf{W}_i \cdot \mathbf{x}_t + \mathbf{b}_i\right), \\ \mathbf{f}_t &= \sigma\left(\mathbf{W}_f \cdot \mathbf{h}_{t-1} + \mathbf{W}_f \cdot \mathbf{x}_t + \mathbf{b}_f\right), \\ \tilde{\mathbf{c}}_t &= tanh\left(\mathbf{W}_c \cdot \mathbf{h}_{t-1} + \mathbf{W}_c \cdot \mathbf{x}_t + \mathbf{b}_c\right), \\ \mathbf{c}_t &= \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t, \\ \mathbf{o}_t &= \sigma\left(\mathbf{W}_o \cdot \mathbf{h}_{t-1} + \mathbf{W}_o \cdot \mathbf{x}_t + \mathbf{b}_o\right), \\ \mathbf{h}_t &= \mathbf{o}_t \circ tanh\left(\mathbf{c}_t\right). \end{aligned} \tag{3.6}$$

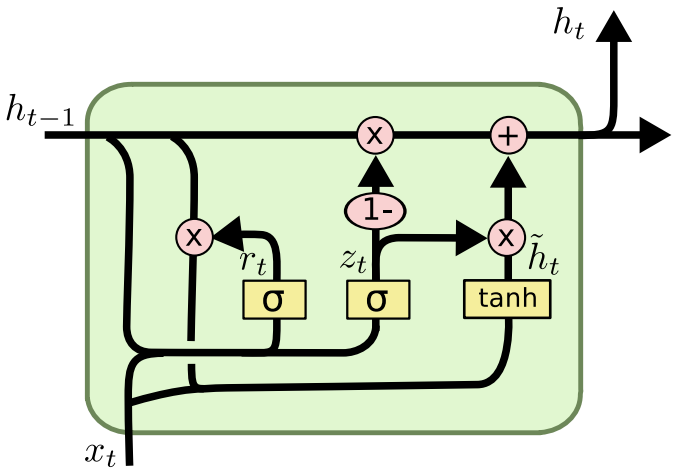



Figure 3.9: Gated Recurrent Units (GRUs). [Image courtesy Chris Olah [29]]

### 3.1.4 Gated Recurrent Unit (GRU)

Authors in [32] proposed a new kind of recurrent network called Gated Recurrent Unit (GRU), as shown in Figure 3.9, with less gates than LSTMs 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

$$\begin{aligned}
\mathbf{z}_t &= \sigma\left(\mathbf{W}_z \cdot \mathbf{h}_{t-1} + \mathbf{W}_z \cdot \mathbf{x}_t\right), \\
\mathbf{r}_t &= \sigma\left(\mathbf{W}_r \cdot \mathbf{h}_{t-1} + \mathbf{W}_r \cdot \mathbf{x}_t\right), \\
\tilde{\mathbf{h}}_t &= tanh\left(\mathbf{W}_o \cdot \mathbf{r}_t + \circ\mathbf{h}_{t-1} + \mathbf{W}_o \cdot \mathbf{x}_t\right), \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) \circ \mathbf{h}_{t-1} + (\mathbf{z}_t) \circ \tilde{\mathbf{h}}_t.
\end{aligned} \tag{3.7}$$

The advantage of GRUs over LSTMs is the smaller number of gates that makes them less memory as well as computationally intense, which is often a critical aspect for ANNs. GRUs have been shown to perform as well as LSTMs in some settings.

### 3.1.5 Bidirectional RNNs

Bidirectional recurrent neural networks (BRNN) [33] 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 time step. So, this type of structure allows the networks

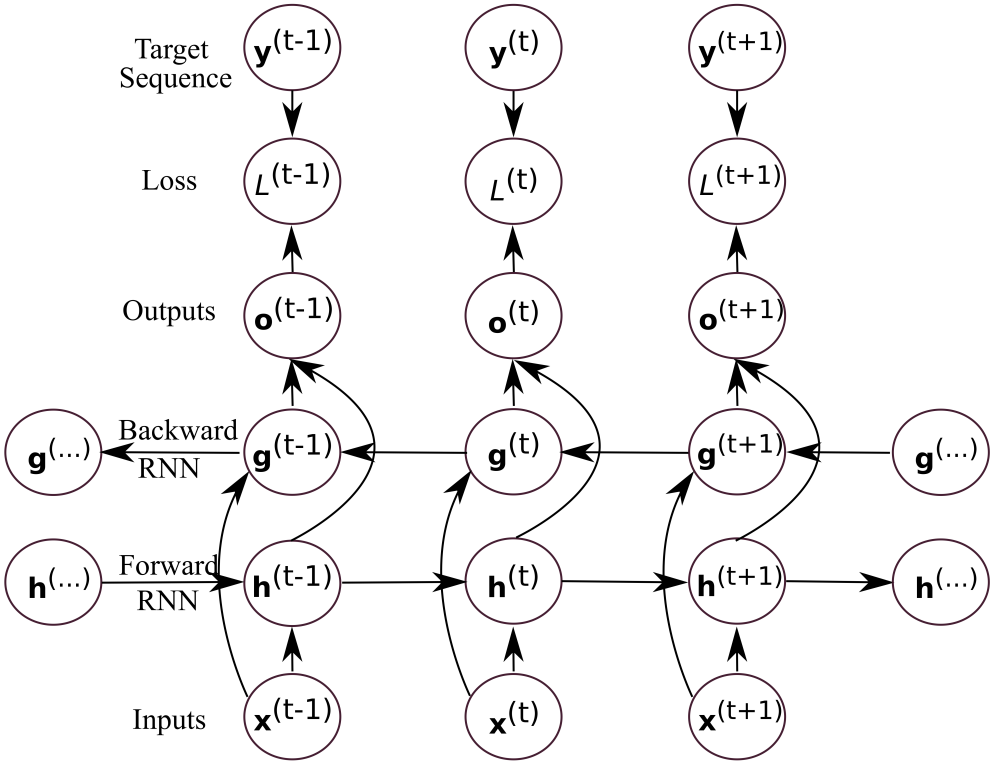



Figure 3.10: The architecture of a vanilla bidirectional recurrent neural network

to have both backward and forward information about the sequence at every time step. BRNN are especially useful when the context of the input is needed. For example, in handwriting recognition [34], the performance can be enhanced by knowledge of the letters located before and after the current letter.

#### 3.1.5.1 Bidirectional Vanilla RNN

#### 3.1.5.2 Bidirectional GRU

## 3.2 Human Pose Estimation

Human pose estimation, one of the core problems in computer vision, is the key component to enable machines to have an understand of people in images and videos. It 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 has been successfully employed in many real-world applications such as action recognition, animation, gaming, augmented reality, and robotics.

Despite various methods proposed in literature, it still remains an unsolved problem due

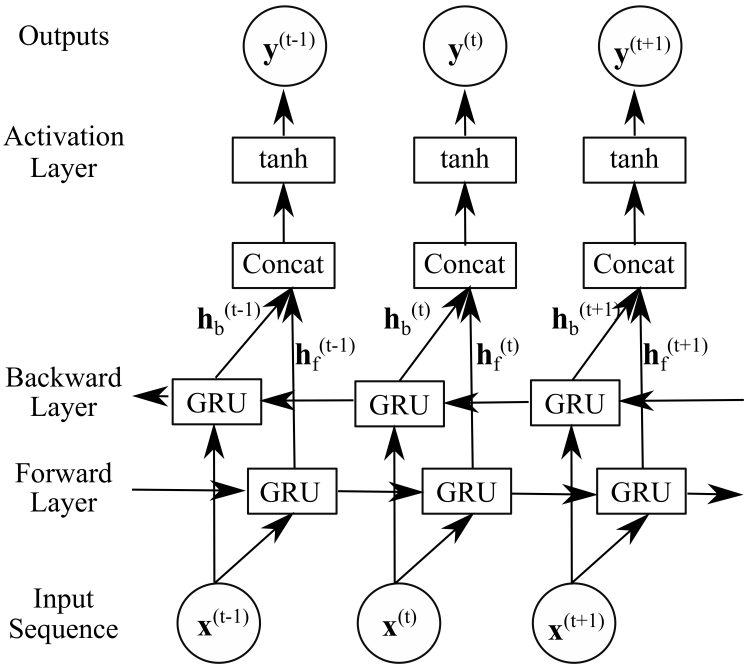



Figure 3.11: The architecture of a bidirectional gated recurrent neural network

to the presence of some difficult challenges like variability of human visual appearance and lighting conditions, partial occlusions due to self articulation and layering of objects in the scene.

### 3.2.1 Types of Pose Estimation

Depending upon the output dimension requirement, the pose estimation algorithm can be categorized into 2D pose estimation and 3D pose estimation. In 2D pose estimation, the location of body joints are predicted in terms of pixel values of 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 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 that can appear at any position or scale. Therefore, it needs to handle the additional problem of inter-person occlusion.

### 3.2.2 Techniques for Pose Estimation

There are two overarching approaches of 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.

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 simplest model possible for pose estimation use 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 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. In contrast, bottom-up approaches show robustness to early commitment and have the potential to decouple runtime complexity from the number of people in the image.

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. [3]]


Figure 3.13: An example of a bottom up approach. [Image courtesy Cao et al. [3]]

### 3.2.3 Introduction to OpenPose Library

In this work, we have employed OpenPose [3], 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 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 feed forward 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.

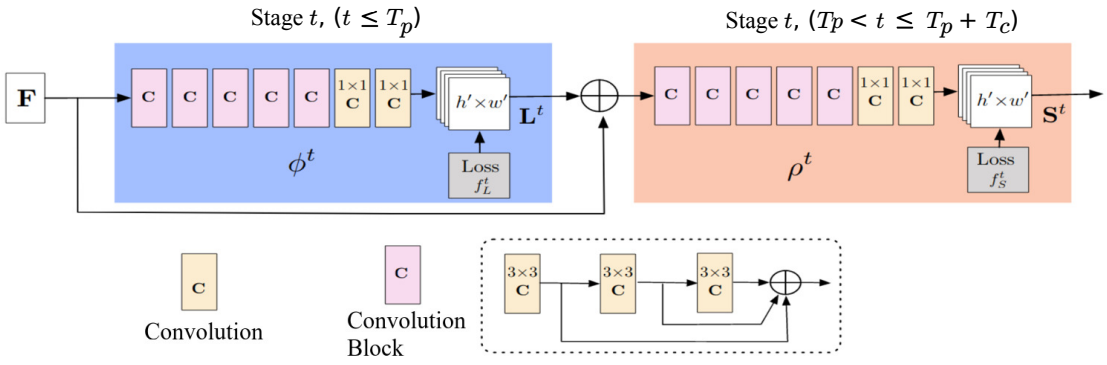



Figure 3.14: Network architecture of the multi-stage CNN. The first set of stages predicts PAFs $\mathbf{L}^t$, while the last set predicts confidence maps $\mathbf{S}^t$. [Image courtesy Cao et al. [3]]

### 3.2.4 Network Architecture

Our architecture, shown in Figure 3.14, iteratively predicts affinity fields that encode part-to-part association, shown in blue,and detection confidence maps, shown in beige.

## 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 using an improved OpenPose [3] algorithm. In this section, we elaborate the feature extraction procedure of our proposed method.

-

### 3.3.1 2D Body Joint Features

As every joint of the human body does not have 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 OpenPose algorithm we searched out for those joints which have a rich and discriminative gait representation capacity. Cunado *et al.* [35] 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. For example, hip joints get wider in coat than normal condition. Again, in some gait videos, some

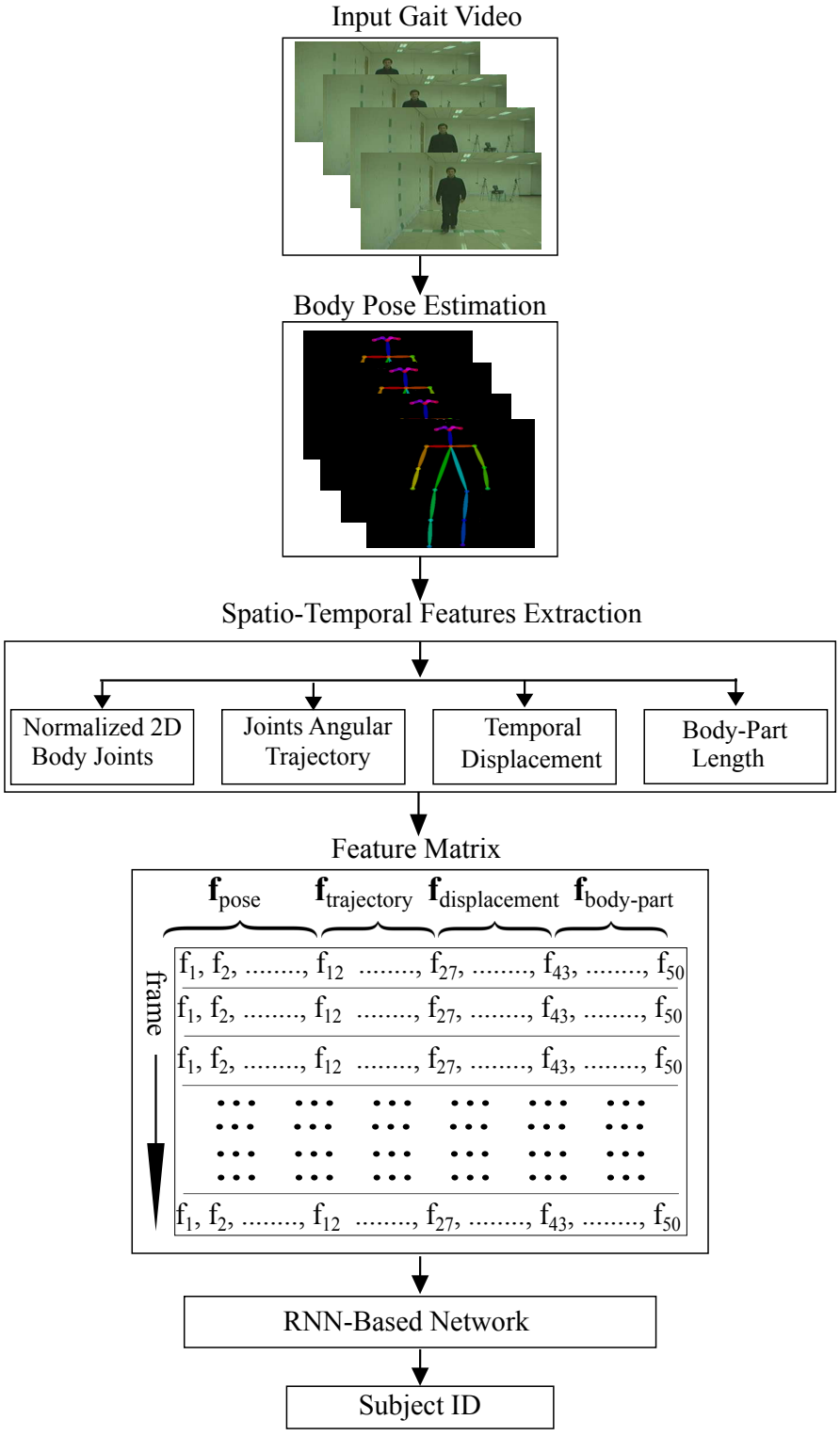



Figure 3.15: The overview of the proposed framework for gait recognition. 2D human poses were first extracted from raw video frames using improved OpenPose [3] algorithm. Four different types of spatio-temporal features were then extracted to form a 50-dimensional feature vector. Thereafter a pose sequence 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.

subjects put their hands into their coat pocket, which they cannot do in normal walking. This situation significantly changes the joint coordinates. Furthermore, joints above hip joint do not have any significant impact on gait pattern. Hence, we do not consider hip or any other body joints above it.

Consequently, in our work, as shown in Figure 3.16(a) we selected 6 body joints (RKnee, Rankle, RBigToe, LKnee, LAnkle, LBigToe) to form our effective pose features. Thus, we have 12-dimensional pose feature vector, $\mathbf{f}_{pose}$, for a single frame.

$$\mathbf{f}_{pose} = [x_1, y_1, x_2, y_2, \ldots\ldots, x_6, y_6]^T \tag{3.8}$$

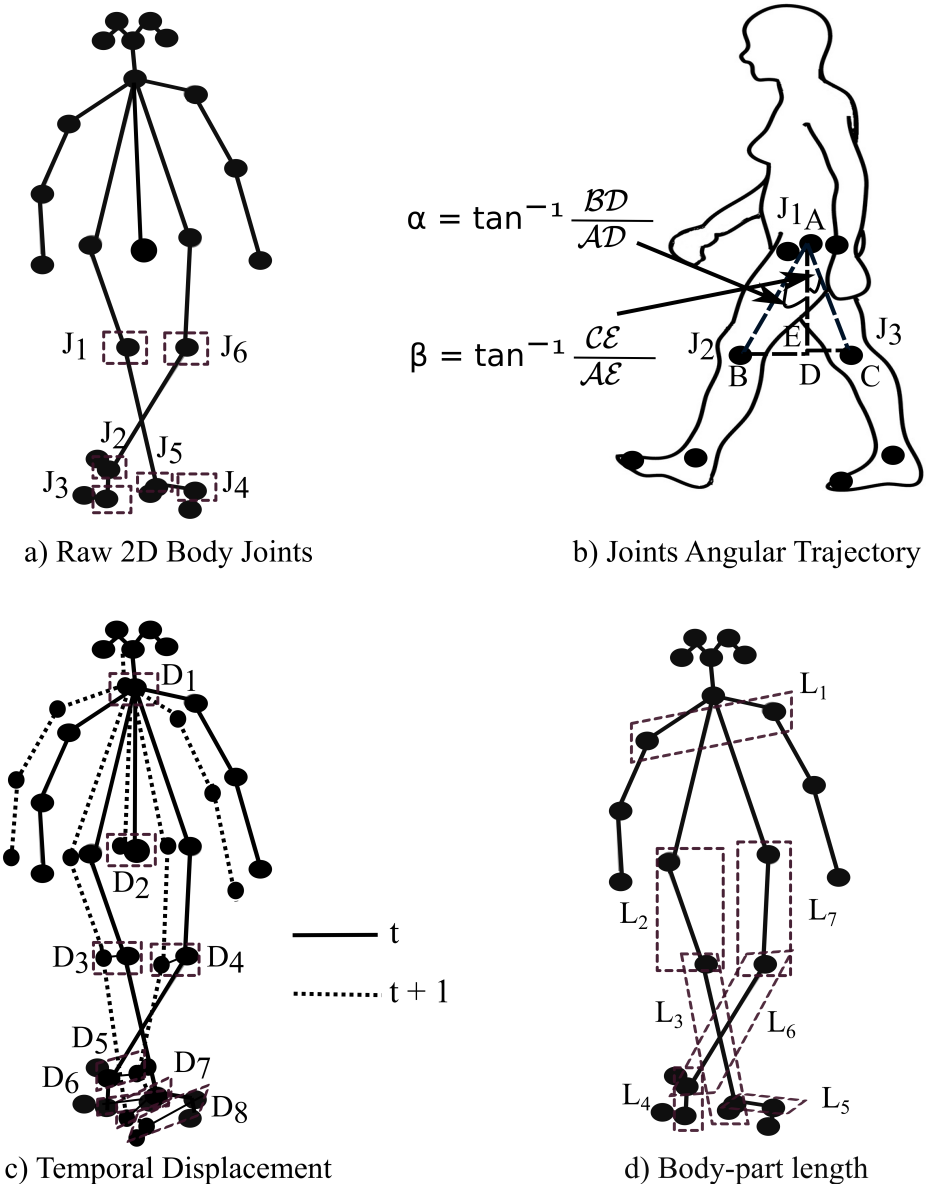
It is necessary to normalize the pose sequence data with regard to the subject position in frame, size, or speed of walking to get improved performance. In different gait datasets, as people walk through the fixed camera, the size of the subject's body alters due to change in the distance between the subject and the camera changes. In our study, to find the origin of the coordinate system ($J_c$) for each subject, we considered right, left, and middle of the hip joints and calculated the average of them. Again, to normalize the bodies of different subjects to a fixed size, we took $h$, the euclidean distance from hip to neck joint, as unit length. Equation **??** shows the normalization procedure of the raw 2D joints.

$$\begin{aligned} Jc &= (J_{LHip} + J_{RHip} + J_{MHip})/3 \\ h &= \| J_c - J_{neck} \|_2 \\ J_i^N &= (J_i - J_c)/h \end{aligned} \tag{3.9}$$

Here, ${J_i}^N$ be the new coordinate of the $i^{th}$ joint $J_i$ of a particular pose.

### 3.3.2 Joint Angular Trajectory

The dynamics of human 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 [36]. Therefore, in this study, we formulated a 15-dimensional feature vector $\boldsymbol{f}_{trajectory}$ by considering five lower limb joint-angle trajectories using following equations:




a) Raw 2D Body Joints

b) Joints Angular Trajectory

c) Temporal Displacement

d) Body-part length

Figure 3.16: Different feature extraction process of the proposed method. a) 6 effective joints were selected out of 25 body joints as estimated from pose estimation algorithm [3]. These selected joints formed a 12-dimensional pose vector. b) 5 angular trajectories from lower limbs were considered to form a joint-angle feature vector. c) A total of 8 body joints were selected to get temporal displacement feature vector. d) 7 body parts were taken to form a limb length feature vector.

Table 3.1: List of selected joint-angle trajectories with corresponding body joint set in order to form gait 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 |

$$\alpha = \begin{cases} \tan^{-1} \frac{|J_{2,x} - J_{1,x}|}{|J_{2,y} - J_{1,y}|} & J_{2,y} \neq J_{1,y} \\ \pi/2 & J_{2,y} = J_{1,y} \end{cases}$$

$$\beta = \begin{cases} \tan^{-1} \frac{|J_{3,x} - J_{1,x}|}{|J_{3,y} - J_{1,y}|} & J_{3,y} \neq J_{1,y} \\ \pi/2 & J_{3,y} = J_{1,y} \end{cases} \tag{3.10}$$

$$\theta = \alpha + \beta$$

As shown in Figure 3.16 (b), $J_1, J_2, J_3$ are the joints which form a set of angular trajectory. In this work, we considered 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 $(\theta, \alpha, \beta)$ as gait features.

$$\mathbf{f}_{trajectory} = [\theta_1, \alpha_1, \beta_1, \theta_2, \alpha_2, \beta_2, \ldots\ldots, \theta_5, \alpha_5, \beta_5]^T \tag{3.11}$$

### 3.3.3 Temporal Displacement

Our third type of extracted features was a simple descriptor that preserves temporal information. It stores the local motion features of gait by keeping the displacement information between the two adjacent frames of the 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 frame of a particular gait. The displacement information of the coordinates of any joint of frame $t$ would be the normalized difference between the corresponding coordinates.

$$\begin{aligned} \triangle x_1^t &= \frac{x_1^{t+1} - x_1^t}{\sum_{i=1}^{8} \parallel J_i^{t+1} - J_i^t \parallel_2} \\ \triangle y_1^t &= \frac{y_1^{t+1} - y_1^t}{\sum_{i=1}^{8} \parallel J_i^{t+1} - J_i^t \parallel_2} \\ \mathbf{f}_{displacement} &= [\triangle x_1, \triangle y_1, \triangle x_2, \triangle y_2 \ldots\ldots, \triangle x_8, \triangle y_8]^T \end{aligned} \tag{3.12}$$

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 3.16 (c), we selected 8 joints (Neck, MHip, RKnee, Rankle, RBig-

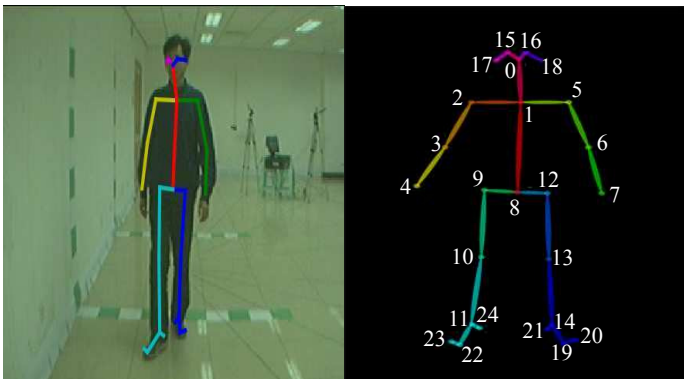



Figure 3.17: Examples of 2D human pose estimation by [3] from RGB images of CASIA dataset (left ones). Detected 25 human body joints with description are shown. (right ones)

Toe, 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 important for gait recognition [36, 37]. They form a spatial gait features which make them robust against covariate such as carrying and clothing 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 [25, 36]. 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 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 [3], we got 25 body joints from each frame (Figure 3.17).

### 3.4.1 Handling Missing Data

We took several preprocessing steps to address the problem of missing data due to occlusions. The main strategies 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.
- 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.

### 3.4.2 Forming Feature Map

In this research, 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 following equations. Here, **p** is 50-dimension pose vector for each frame; $\boldsymbol{T}$ is the feature matrix for each timestep; $N$ is the total number of timestep sequence, and $\boldsymbol{V}$ is the sequence of features for a gait video.

$$\begin{aligned} \boldsymbol{p} &= [f_1, f_2, f_3, \ldots\ldots, f_{50}]^T \\ \boldsymbol{T} &= [\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \ldots\ldots, \boldsymbol{p}_{28}]^T \epsilon \quad \mathbb{R}^{28\times 50} \\ \boldsymbol{V} &= [\boldsymbol{T}_1, \boldsymbol{T}_2, \boldsymbol{T}_3, \ldots\ldots \boldsymbol{T}_N]^T \end{aligned} \tag{3.13}$$

### 3.4.3 Data Augmentation

The performance of deep neural networks is strongly correlated with the amount of available training data. Although CASIA [6] 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, not all subjects have all gait videos; there are some missing gait videos. To solve the problem, we 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.

$$N(j_i) = (x + \tilde{x}, \quad y + \tilde{y}) \tag{3.14}$$

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.

## 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

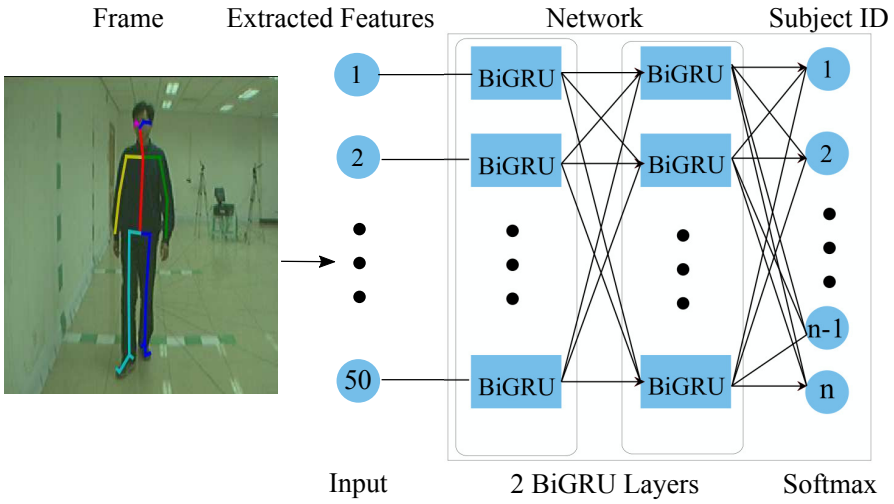



Figure 3.18: Proposed RNN architecture for robust gait recognition. It consists of two BiGRU [33] layers each of which consists of 80 GRU cells with one batch normalization and one output softmax layer. The network was fed with a 50-dimensional spatio-temporal feature vector obtained from 2D pose estimation. Input layer was followed by a batch normalization layer [38]. 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.

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) [39] and Bidirectional Gated Recurrent Units (BiGRU) [33]. 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 BiGRU with 80 hidden units performs best.

After input and the second recurrent layer, we placed a batch normalization (BN) [38] 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.

Table 3.2: Training summary of our proposed temporal network.

| **Hyperparameter** | **Value** |
|---|---|
| Optimizer | Adam [40] |
| Objective function | Fusion of softmax and center loss |
| Epochs | 450 |
| Initial learning rate | $5 \times 10^{-3}$ |
| Mini-batch size | 256 |

### 3.5.2 Training

The training of RNNs allows us to learn the parameters from the sequence. We have employed Adam [40] optimization algorithm with $\beta_1 = 0.9, \beta_2 = 0.999$, which is known to work very well for training recurrent neural networks. We tried several learning rates in our experiment and found out that the best initial learning rate is $(1\text{x}10^{-3})$. 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. This overfitting problem has been addressed by adding a batch normalization 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.

### 3.5.3 Loss Functions

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. Now, if we only use the *cross-entropy loss* (CE) as our objective function, the resulting learned features may contain large intraclass variations. Therefore, to effectively reduce the intraclass distance, we used *center loss* (CL), introduced by Wen *et al.* [41] for face recognition task. As the training progresses, the center loss learns a center for features of each class and the distances between the features and their corresponding class centers are minimized simultaneously. However, using only center loss may lead the learned features and 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 intr-

aclass distance as much as possible.

$$
\begin{aligned}
L_s &= -\sum_{i=1}^{m} log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} \\
L_c &= \frac{1}{2} \sum_{i=1}^{m} \parallel \boldsymbol{x}_i - \boldsymbol{c}_{y_i} \parallel_2^2 \\
L &= L_s + \lambda L_c + \lambda_\theta \parallel \theta \parallel_2
\end{aligned}
\tag{3.15}
$$

Equations (3.15) describe the total loss ($L$) calculation of our 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. $\lambda$, a scalar variable, is set to value $0.01$ to balance 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 $(\lambda_\theta)$ set to $0.0005$ for the experiment.

### 3.5.4 Post-processing

While training, our proposed temporal network considers each of these video clip as a separate video (see Fig. 3.19). For a given video, the prediction of our model is a sequence of class probabilities for each of the 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 of $n$ number of subjects. For a particular timestep $t$ of a gait video, input pose sequence vector $\boldsymbol{X}^t \epsilon \mathbb{R}^{28 \times 50}$ has an n-dimensional output vector $\boldsymbol{o}^t$.

$$
\begin{aligned}
\boldsymbol{s}^t &= [s_1, s_2, s_3, ....., s_n]^T \\
\boldsymbol{o}^t &= [o_1, o_2, o_3, ....., o_n]^T
\end{aligned}
\tag{3.16}
$$

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 $s^t$ to the subject class $s_i$ which have maximum

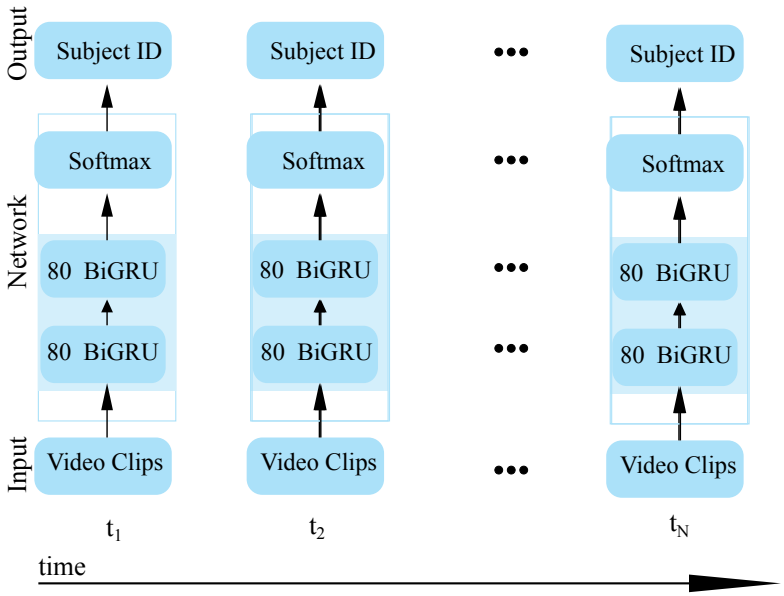



Figure 3.19: Output prediction scheme of our proposed temporal network. Each input clip was considered as a separate video and a sequence of class probabilities was predicted at output. *Majority voting scheme* was used to process the output to predict the subject ID.

probabilities for the timestep $t$. As each of our gait videos is divided into a series of timestep sequence (see equation 3.16), using majority voting scheme we can have the subject ID. Following equations described the voting scheme:

$$\begin{aligned} s_t &= \arg\max_{s_i} \{o_i^t | 1 \leq i \leq n\} \\ s &= \arg\max_{i \in (1,2,\ldots,n)} \sum_{t=1}^{N} s_i^N \end{aligned} \tag{3.17}$$

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

## 3.6 Multi-View Gait Recognition

The workflow of our proposed two-stage multi-view gait recognition network is illustrated in Fig. 3.20. 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 recognition using proposed temporal network which has been trained for that particular angle.

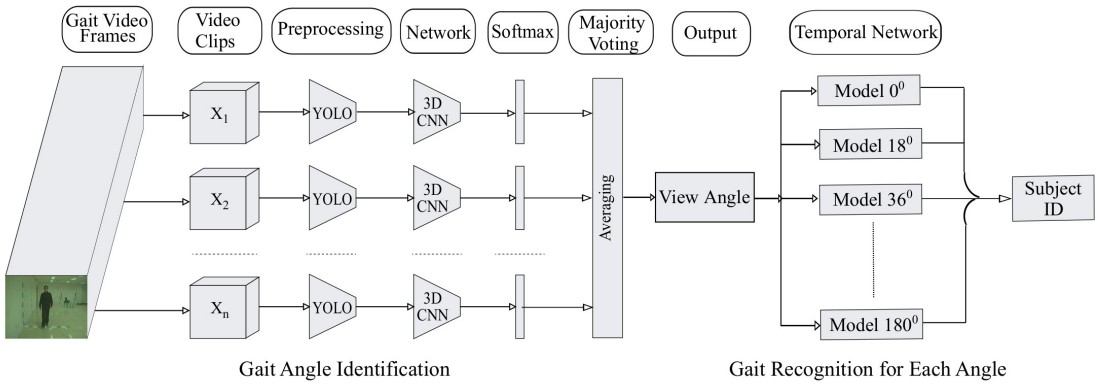



Figure 3.20: Overview of our proposed multi-view gait recognition network scheme. YOLOv3 [42] was used to detect and locate the walking people in video frames. The input of the network was a clip of 16 consecutive frame which was preprocessed and resized to $112 \times 112$ to feed into a 3D convolutional network based on C3D [43]. The network used 3D kernels to exploit spatio-temporal dynamics for viewing angle identification. Thereafter, a temporal network, trained on each viewing angle, was performed subject identification by modeling temporal dynamics from input 2D pose sequence.

### 3.6.1 Preprocessing

Firstly, to localize human walking in gait videos, we used YOLOv3, a state-of-the-art real-time object detection algorithm, proposed by Redmon *et al.* [42]. We then cropped each of the person detected frame using the bounding box coordinates found from YOLOv3 algorithm and resized them to $112 \times 112$ for our network input. Thereafter, we splitted each gait video into overlapping sequences of 16 consecutive frames within 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 Network Architecture

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. [43] proposed a 3D convolutional neural network, also known as C3D, which has been widely used for applications like video classification, action recognition, etc. This network has been trained on one of the largest video classification benchmark datasets Sports-1M [**?**]. The dataset contains 1.1 million sports videos, where each video belongs to one of the 487 sports categories.

The proposed method for our gait angle identification is illustrated in Fig. 3.20. The

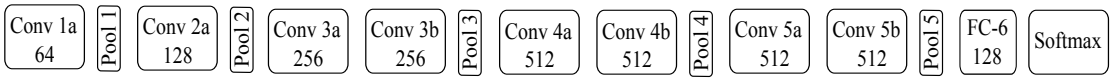



Figure 3.21: Proposed 3D-CNN for video angle identification. Last 3 layers of a pre-trained C3D [43] network has been replaced by a fully connected layer of 128 neurons followed a final softmax layer of 11 neurons to classify 11 different walking direction in CASIA-B dataset.

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 | $1 \times 10^{-3}$ |
| Mini-batch size | 12 |
| Momentum | 0.92 |

input of the network was a clip of 16 consecutive frame which was preprocessed and resized to $112 \times 112$ to feed into a a 3D-CNN network. We used *majority voting scheme* to process the output to predict the view angle similar to section 3.5.4, i.e. the angle that receives the highest number of votes over all clips are referred as predicted angle of the video.

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 viewpoint angle from gait videos. Fig. 3.21 shows our proposed 3D convolutional network.

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.

### 3.6.3 Training

We used CASIA-B gait dataset [6] to train our model. We trained the network using 4 normal walking sequences of 100 subject 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 hyper-parameters setting of our proposed network.

# Chapter 4

# Results and Discussions

This chapter briefly discusses the datasets we used to train and evaluate our model, and the results our proposed algorithm achieved at different experimental setup. As to estimate pose, RGB video frames are required. So, we couldn't evaluate our method to those dataset which only consists of silhouette sequences.

## 4.1 Dataset

The success of deep learning-based methods greatly depends on the vast amount of labeled train data. Unfortunately, few existing gait databases contain a large number of subjects as well as a variety of covariate factors. Some of the publicly available gait databases are CASIA gait dataset [6], TUM GAID dataset [44], OU-ISIR multi-view large population dataset (OU-MVLP) [45] and USF HumanID dataset [46].

In USF HumanID gait dataset, there are $122$ subjects walking outside on two different surfaces of an elliptical path under two different time, viewpoint, clothing, shoes, and carrying conditions. However, not all subjects were filmed under all conditions. TUM GAID dataset is another large dataset for gait recognition which consists of $305$ subjects where each subject has $10$ videos. But this dataset is not suitable for multi-view gait recognition as all the videos were recorded from side view angle. The largest dataset available for gait recognition is OU-ISIR multi-view large population dataset (OU-MVLP). It contains $10,307$ subjects from $14$ view angles ranging from $0^\circ - 90^\circ$, $180^\circ - 270^\circ$. Only two sequences are provided, one for the gallery and the other for the probe. But, this dataset is formatted only as a set of silhouette sequence making it different from our approach.

In this study, we used CASIA (both CASIA A and CASIA B) dataset which is one

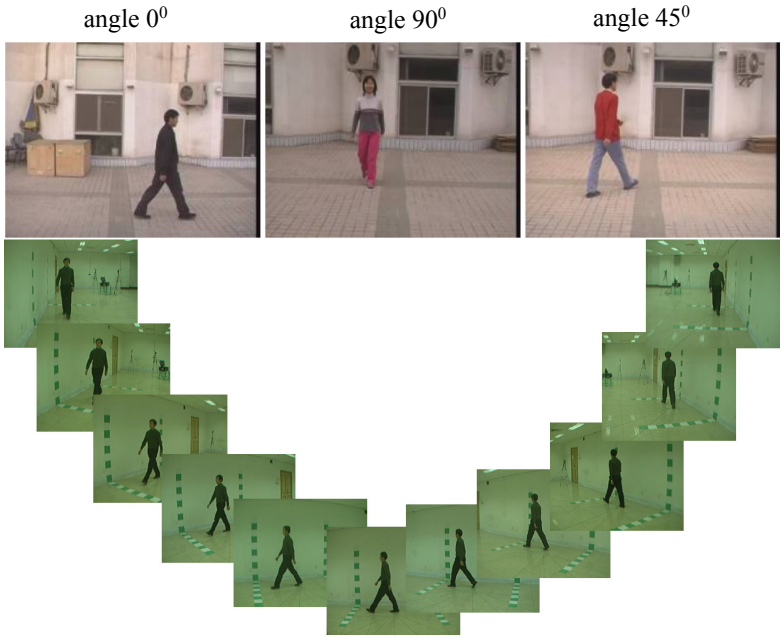



Figure 4.1: Sample video frames of CASIA A and CASIA B dataset. In top, some of the sample images from CASIA A dataset are shown where the subjects are walking along straight line in 3 different view angle, and in bottom, CASIA B dataset is shown with its 11 view angle.

of the largest multi-view gait databases. CASIA A dataset contains total 20 subjects walking in an outdoor environment where CASIA B dataset includes 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 viewing angle every subject has four gait sequences 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 of each subject captured from 11 view angles: 6 sequences for normal walking ('nm'), 2 sequences for walking in a coat ('cl') and 2 sequences for walking with bag ('bg') on shoulder. Hence, this dataset separately considered three variations in people walking namely viewing angle, clothing and carrying conditions. The view angle set of the camera is ranging from $0°$ to $180°$. Figure 4.1 illustrates some of the sample video frames of CASIA dataset.

## 4.2 Single-View Gait Recognition

### 4.2.1 Experimental Evaluation on CASIA-A dataset

Since, CASIA A dataset contains only 20 subjects each of which have only four gait sequence in three different angles, we trained three model for each of the gait angle with

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 higher average recognition rates **100.0%** and outperforms other state-of-the-art methods by a large margin.

| Methods | 0° | 45° | 90° | Mean |
|---|---|---|---|---|
| Wang [47] | 88.75 | 87.50 | 90.00 | 88.75 |
| Goffredo [48] | 100.0 | 97.50 | 91.00 | 96.16 |
| Liu [49] | 85.00 | 87.50 | 95.00 | 89.17 |
| Lima [50] | 92.50 | 97.50 | 98.75 | 96.25 |
| Kusakunniran [51] | 100 | 100 | 98.75 | 99.58 |
| Proposed | **100.0** | **100.0** | **100.0** | **100.0** |

20 output neurons in the final softmax layer of our proposed temporal network. To evaluate the performance of our proposed method on CASIA A dataset, we used leave-one-out cross validation rule, i.e., one sequence was set for testing and the remainder was set for training the network for each view angle. We compare our results with four other prevailing state-of-the-art gait recognition methods including Wang [47], Goffredo [48], Liu [49], Lima [50], Kusakunniran [51] (see Figure 4.2). Table 4.1 illustrates that the proposed method have achieved higher average correct class recognition rates (CCR) 100.0% compared to other methods.

### 4.2.2 Experimental Evaluation on CASIA-B Dataset

#### 4.2.2.1 Experimental Setup

We designed two experimental setups (A, B) in CASIA B dataset for evaluation. Experiment setup A was for evaluating the performance of the proposed method in single-view gait recognition. To investigate the robustness of view variation, comparison results of the proposed approach 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, as demonstrated in Table 4.2, we divided the dataset into two groups where the first group which consists of 62 subjects was used to train the network. The second group contains rest of the subjects for evaluating the performance of the model. For setup B, the ratio between 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*

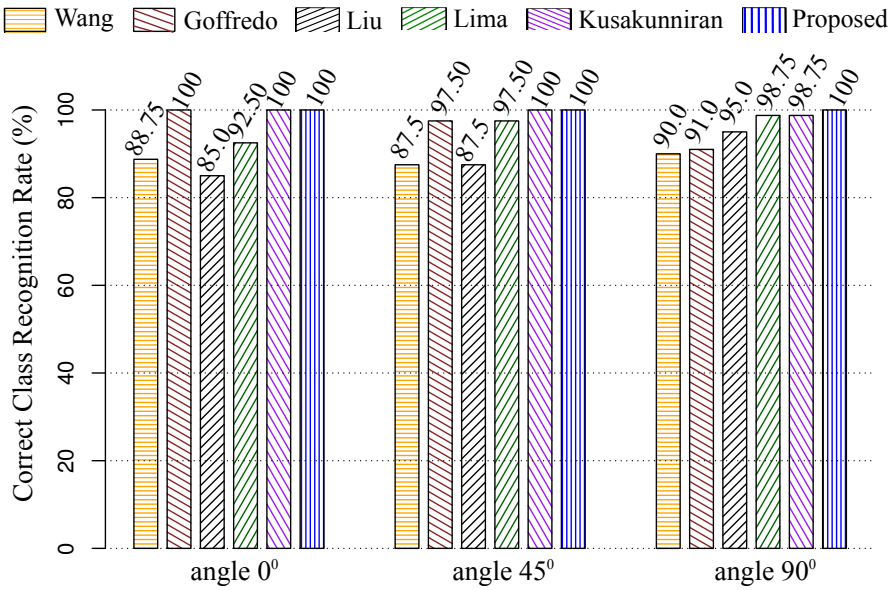



Figure 4.2: Comparison in CCR at different view angles among proposed method with other prevailing gait recognition methods proposed in literature on CASIA A dataset. Our method achieves **100%** class recognition rate on all of the view angles which proved the efficacy of the proposed method.

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 is subdivided 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 | | $bg01 - bg02$ |
| | | | | | | $cl01 - cl02$ |

consists of two subjects carrying bag and wearing coat respectively.

We divide CASIA-B dataset into two group where first group consists of 24 subjects and is used to train the network. The second group contains rest 100 subjects that are used to evaluate the performance of the model. In evaluation set, 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 2 subjects carrying bag are kept in *ProbeBG* and remaining 2 subjects wearing coat are kept in *ProbeCL*. Table 4.2 shows this experimental setup.

Table 4.3: Correct class recognition rate (CCR) of proposed method in all three probe sets of CASIA B dataset. Here, column represents a specific view of gallery and probe set. It has been observed that the probe set of normal walking (*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** |

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

Experimental results of single-view gait recognition on all the three probe set of CASIA B dataset without view variation is illustrated in 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 our proposed method towards 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 Gait-GANv2 [22], PTSN [24], PoseGait [25], Yu *et al.* [52] 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 approx. **10%** from PTSN.

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 in all three probe set of CASIA B dataset. As the proposed method doesn't depend on any body point higher than knee, it shows the robustness towards these covariate factors. It also achieves higher average CCR **93.34%** by outperforming other methods at a significant margin.

| Methods | *ProbeNM* | *ProbeBG* | *ProbeCL* | Average |
|---|---|---|---|---|
| Liao *et al.* [25] | 96.92 | 85.78 | 68.11 | 83.60 |
| Yu *et al.* [52] | 97.58 | 72.14 | 45.45 | 71.72 |
| Yu *et al.* [22] | 98.24 | 76.25 | 42.89 | 72.46 |
| Liao *et al.* [25] | 96.63 | 71.26 | 54.18 | 74.02 |
| Proposed | **99.41** | **97.80** | **82.82** | **93.34** |

#### 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 ($\theta_g$) 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 Probe NM, probe BG, and probe CL 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 [22], PoseGait [25], Yu *et al.* [52]. It is observed from Figure 4.4 and Table 4.6 comparison that proposed method outperforms other in covariate variation and achieves comparable performance in normal walking.

Since, to recognize gait, we consider features based on effective body joints, hence our method doesn't get affected by the variation in covariate conditions compared to other appearance-based method or other model-based methods which consider ineffective features to build their gait descriptor. Thats why our method is proven to be less sensitive to view angle variation and performs better in carrying-bag and clothing condition.

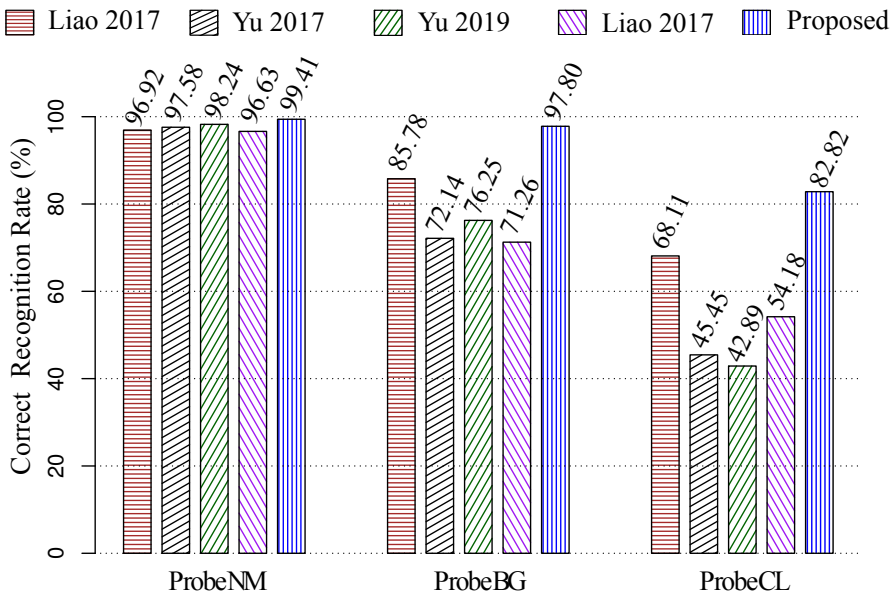



Figure 4.3: Correct class recognition rates (%) of the proposed method with other state-of-the-art methods on all three probe set of CASIA-B dataset without view variation. Proposed method demonstrates better performance compared to other by achieving $89.64\%$ and $96.45\%$ in two covariate conditions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

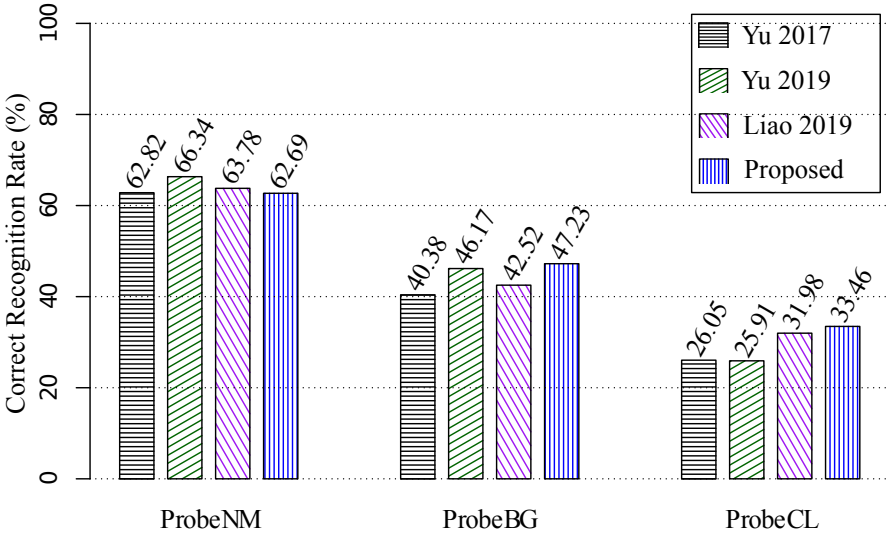



Figure 4.4: Comparison with different state-of-the-art methods for gait recognition with view variation in all three probe set of CASIA B dataset. Here, the value reported for each algorithm is the average of all the gallery view's average CCR. Proposed method outperforms other state-of-the-art methods achieving $47.23\%$ and $33.46\%$ 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 ($\theta_g$) 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. 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 to other prevailing methods.

| Methods | *ProbeNM* | *ProbeBG* | *ProbeCL* |
|---|---|---|---|
| Yu *et al.* [52] | 62.82 | 40.38 | 26.05 |
| Yu *et al.* [22] | 66.34 | 46.17 | 25.91 |
| Liao *et al.* [25] | 63.78 | 42.52 | 31.98 |
| Proposed | **62.69** | **47.23** | **33.46** |

## 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.

### 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 method in cross-view recognition, we make the comparison between the proposed method and three other state-of-the-art methods includ-

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 |

ing CNN [17], CMCC [53], and GEI-SVR [54] 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 [**?**], it is seen that CNN [17] achieves the highest recognition rates when the view variation is large due to the use of supervised information of all 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 viewing angle.

## 4.4 Multi-View Gait Recognition

In multi-view gait recognition, multiple views of gallery gaits are combined to recognize probe set for an unknown gait view. In our work, for multi-view gait recognition, we trained a two-stage network in which we initially identify the walking direction of a

Table 4.8: 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 |

Table 4.9: Correct walking direction identification rate (%) of proposed 3D-CNN network on all three probe set of CASIA-B dataset. The network achieved **100%** identification accuracy in all of the 11 view angles.

| 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 |

gait video using a 3D-CNN network.

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

We tested the proposed 3D-CNN network with all three probe set of CASIA-B dataset and have achieved **100%** identification accuracy in all viewpoint angles proving the fact that our 3D-CNN is efficient in classifying walking direction from gait videos. Table 4.9 illustrates our test result.

To evaluate the performance of the proposed two-stage network, we compare it with the recent state-of-the-art multi-view gait recognition methods such as Dupuis *et al.* [55], Isaac *et al.* [56], and VI-MGR [57] on all three probe set of CASIA-B dataset. The comparison, as illustrated in Table 4.8 and Fig. 4.5, shows that the proposed method

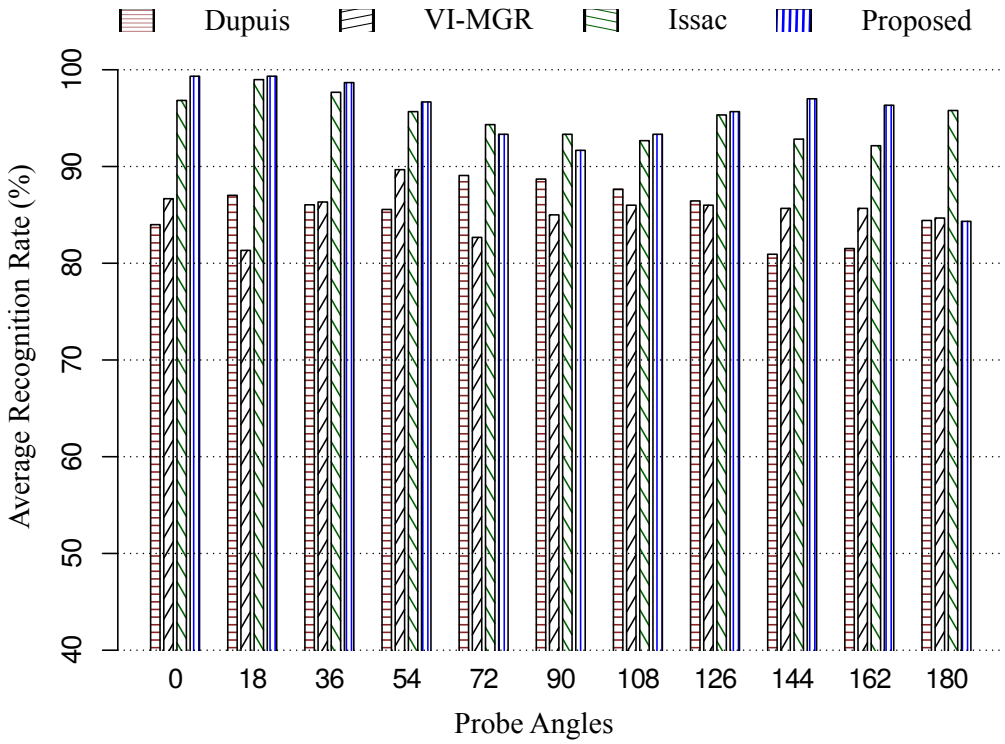



Figure 4.5: Average recognition rates(%) of the proposed method compared to the other state-of-the-art methods in multi-view gait recognition. Proposed method achieves higher average recognition accuracy on 8 of total 11 probe angles of CASIA-B dataset compared other methods in literature.

exceeds the previous best in result all three probe set by a significant margin. It outperforms other in **8** of 11 total probe angles.

# 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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] H. Chao, Y. He, J. Zhang, and J. Feng, "Gaitset: regarding gait as a set for cross-view gait recognition," in *The Thirty-Third AAAI Conference on Artificial Intelligence*. July, 2019, pp. 973 – 987.

[13] 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.

[14] G. Ariyanto and M. S. Nixon, "Model-based 3d gait biometrics," in *Int. Joint Conf. on Biometrics*. Washington DC, USA, 2011, pp. 1 – 7.

[15] F. Tafazzoli and R. Safabakhsh, "Model-based human gait recognition using leg and arm movements," *Engineering Appl. of Art. Intell.*, vol. 23, no. 8, pp. 1237 – 1246, December 2010.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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 2019.

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

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

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

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

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

[31] 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.

[32] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–Decoder approaches," in *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014.

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

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

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

[41] 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.

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

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

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