Gait recognition using a self-supervised self-attention deep learning model

Pinčić, Domagoj

UNIVERSITY OF RIJEKA
FACULTY OF ENGINEERING

Domagoj Pinčić

GAIT RECOGNITION USING A SELF-SUPERVISED SELF-ATTENTION DEEP LEARNING MODEL

DOCTORAL DISSERTATION

Rijeka 2023.

UNIVERSITY OF RIJEKA
FACULTY OF ENGINEERING

Domagoj Pinčić

GAIT RECOGNITION USING A SELF-SUPERVISED SELF-ATTENTION DEEP LEARNING MODEL

DOCTORAL DISSERTATION

Rijeka 2023.

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET

Domagoj Pinčić

IDENTIFIKACIJA OSOBA TEMELJEM HODA UPOTREBOM SAMONADZIRANOGA MODELA DUBOKOGA UČENJA PRIMJENOM MEHANIZMA SAMOPOZORNOSTI

DOKTORSKI RAD

Mentor: prof. dr. sc. Kristijan Lenac

Rijeka 2023.

Doctoral dissertation supervisor: Prof. D. Sc. Kristijan Lenac (University of Rijeka, Faculty of Engineering, Croatia)

The doctoral dissertation was defended on ________________ at the University of Rijeka, Faculty of Engineering, Croatia, in front of the following Evaluation Committee:

1. Prof. D. Sc. Ivo Ipšić, University of Rijeka, Faculty of Engineering, Croatia - Committee Chair

2. Assoc. Prof. D. Sc. Sandi Ljubić, University of Rijeka, Faculty of Engineering, Croatia

3. Doc. D. Sc. Žiga Emeršič, University of Ljubljana, Faculty of Computer and Information Science

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Prof. D. Sc. Kristijan Lenac, for all the help, guidance, advice, and mentorship during my doctoral studies and while writing this doctoral dissertation.

I would also like to extend my gratitude to the best doctoral dissertation evaluation committee, Prof. D. Sc. Ivo Ipšić, Doc. D. Sc. Sandi Ljubić and Doc. D. Sc. Žiga Emeršič, for their valuable time and invaluable suggestions.

Special thanks goes to Diego Sušanj, for all the help during the writing of this dissertation, as well as in research done in the few past years. And, of course, for all the coffees and foosball matches throughout the years.

Thanks to all the colleagues for all the advice and time spent together.

All the members of the LRV laboratory in Ljubljana, and especially Žiga, thank you for your hospitality and for boosting my motivation for science.

Furthermore, the appreciation also goes to my dear friend Siniša, for all the support over the countless years, in the form of endless table tennis matches, ordinary tennis matches, and every other possible sport, and beyond.

Also, thank you, Kristina, for your encouragement, motivation, and all the journeys over the course of many years. And for always being there for me.

And finally, I would like to thank my family for their continuous support and inspiration. With special thanks to Hogar, Helga, and Frida.

ABSTRACT

Gait biometric is unique to each individual and has numerous beneficial characteristics that make it suitable for various applications, such as crime investigation, surveillance, and access control. What makes gait recognition especially appealing is its ability to be recognized remotely without the cooperation of the individual. Furthermore, the use of low-resolution cameras is sufficient for gait recognition, thus reducing the need for specialized equipment. Additionally, obscuring or falsifying one's gait is inherently difficult, enhancing the reliability of this method.

This dissertation explores the gait recognition problem with the goal of achieving an accurate recognition rate by utilizing the self-supervised learning approach in order to train feature extraction models to learn useful gait features, without using data annotation, bypassing the need for expensive and time-consuming data annotation. Furthermore, a ViT model is proposed as a backbone model, and its performance is investigated in the context of gait recognition.

An experimental study was performed, by training the feature extraction models on the two widely used gait recognition datasets, CASIA-B and OU-MVLP. The gait features are extracted using the feature extraction model, and the features are classified using a proposed FCNN classifier, obtaining results comparable to those of the state-of-the-art approaches based on supervised learning, while being robust to various covariates and view angles. Moreover, the ablation study is performed to analyze the effect of feature extraction model pretraining on different datasets and the differences between the supervised and self-supervised learning approaches for this task.

Keywords: gait recognition, self-supervised learning, ViT, neural network

PROŠIRENI SAŽETAK

Hod je jedinstvena biometrijska značajka za svaku osobu, te kao takva ima brojne djelotvorne karakteristike koje omogućuju primjenu prepoznavanja osoba putem hoda u područjima poput kriminalističkih istraga, nadzora, te kontrole pristupa. Jedna od glavnih prednosti hoda kao biometrijske značajke je mogućnost prepoznavanja hoda osobe na daljinu, bez suradnje osobe. Nadalje, moguće je korištenje kamera niske rezolucije za prepoznavanje hoda osobe, što dovodi do smanjenja potrebe za specijaliziranom opremom. Dodatno, skrivanje ili mijenjanje hoda osobe je vrlo teško i zahtjevno, što rezultira činjenicom da je hod osobe vrlo pouzdana metoda identifikacije osobe.

Ovaj doktorski rad istražuje problem prepoznavanja osoba putem hoda s ciljem postizanja točne razine prepoznavanja osoba, koristeći samonadzirani pristup učenja za treniranje modela za izlučivanje značajki hoda, kako bi model naučio korisne značajke hoda bez korištenja oznaka podataka, time zaobilazeći potrebu za skupim i dugotrajnim označavanjem podataka. Nadalje, ViT model dubokog učenja je predložen kao bazični model, te su njegove performanse analizirane u kontekstu prepoznavanja osoba putem hoda.

Podaci korišteni u ovom radu su pripremljeni u formi skupova podataka za treniranje modela za ekstrakciju značajki hoda, iz skupova podataka CASIA-B i OU-MVLP. Podaci su pretprocesirani kako bi se uklonio suvišan šum te je generirana reprezentacija značajki hoda u obliku GEI slika za svaku osobu, te za svaki ciklus hoda. Zatim je provedena eksperimentalna studija, trenirajući modele za izlučivanje značajki hoda na dva spomenuta skupa podataka. Istrenirani su ViT modeli dubokog učenja s dvije različite veličine podjele ulazne slike na segmente, 16×16 i 8×8 piksela, kako bi se analizirao utjecaj veličine navedenog parametra na točnost prepoznavanja osoba. Značajke hoda su potom izlučene koristeći istrenirane modele za izlučivanje značajki, te su dobivene značajke hoda klasificirane koristeći predloženi FCNN model za klasifikaciju.

Rezultati ostvareni kroz navedenu eksperimentalnu studiju pokazuju kako je navedeni pristup ostvario točnost prepoznavanja osoba putem hoda usporedivu s drugim najsuvremenijim metodama koje kao osnovu koriste nadzirano učenje, te je navedeni pristup robustan na različite varijante poput normalnog hoda, hoda s torbom ili hoda u kaputu, te kuteve pod kojima je osoba snimljena. Također, provedena je provjera utjecaja komponenti sustava na učinkovitost kako bi se analizirali utjecaji pred-treniranja modela za izlučivanje značajki hoda na različitim skupovima podataka, te razlike između nadziranog i samonadziranog pristupa učenju za navedeni problem prepoznavanja osoba putem hoda.

Ključne riječi: prepoznavanje hoda, samonadzirano učenje, ViT, neuralna mreža

CONTENTS

Acknowledgements **I**

Abstract **III**

Prošireni sažetak **V**

1. Introduction **1**

1.1. Biometric Person Identification . . . 2

1.2. Gait Recognition . . . 8

1.3. Deep Learning Approaches for Gait Recognition . . . 10

1.4. Scientific Hypothesis and Contributions . . . 12

1.5. Research Methodology . . . 13

1.6. Dissertation Overview . . . 14

2. Person identification using Gait Recognition **17**

2.1. Gait Biometric . . . 17

2.1.1. Advantages of Gait Recognition . . . 19

2.1.2. Problems in Gait Recognition . . . 19

2.1.3. Applications of Gait Recognition . . . 20

2.1.4. Gait Recognition Settings . . . 21

2.2. Sensors and Data used in Gait Recognition . . . 22

2.2.1. Sensors . . . 22

2.2.2. Data representation . . . 24

2.3. Typical Gait Recognition Pipeline . . . 28

2.3.1. Data acquisition and preprocessing . . . 28

2.3.2. Feature Extraction . . . 30

2.3.3. Representation learning . . . 31

2.3.4. Classification . . . 32

3. Machine Learning Approaches for Gait Recognition 35

3.1. Types of Machine Learning Systems . . . 37

3.1.1. Supervised Learning . . . 37

3.1.2. Unsupervised Learning . . . 39

3.1.3. Self-supervised Learning . . . 41

3.2. Machine Learning Models . . . 45

3.2.1. Artifical Neural Network . . . 45

3.2.2. Convolutional Neural Network . . . 47

4. Related Work 51

5. Gait Recognition Datasets 61

5.1. CASIA-B . . . 61

5.2. OU-MVLP . . . 63

5.3. GREW . . . 64

6. Gait recognition using a self-supervised self-attention deep learning model 65

6.1. Data Preprocessing . . . 65

6.2. Training feature extraction model using the self-supervised learning model 68

6.3. Feature extraction self-attention model . . . 72

6.4. Gait features classification . . . 77

6.5. Performance metrics . . . 79

7. Experimental Setup 81

7.1. Datasets . . . 81

7.2. Experiments . . . 82

7.3. Self-supervised feature extraction model . . . 83

7.4. Training Details . . . 84

7.5. Evaluation Protocol . . . 85

8. Results 87

8.1. CASIA-B Dataset Results 87

8.2. OU-MVLP Dataset Results 91

8.3. Comparison of the ViT model patch sizes and kNN and FCNN classifier on the recognition accuracy 92

8.3.1. Effect of the patch size in ViT model 92

8.3.2. Effect of the proposed FCNN classifier 93

8.4. Ablation Experiments 94

8.4.1. Comparison of model pretraining 94

8.4.2. Comparison of supervised vs self-supervised learning 97

8.5. Feature Visualization 99

8.6. Self-Attention Visualization 100

9. Conclusions 103

Bibliography 105

List of Figures 117

List of Tables 119

Appendices 121

Curriculum Vitae 127

List of publications 129

1. Chapter

INTRODUCTION

Identification of individuals is a crucial aspect of many real-world applications, ranging from healthcare and finance to security. Each individual has certain physiological characteristics as well as behavioral traits, that are unique to that individual. The task of person identification is to identify a particular person based on these characteristics and traits. In person identification, the aforementioned characteristics are measured and analyzed to determine the identity of the person in accordance with the corresponding database of the individuals.

There are two main approaches to person identification: non-biometric and biometric. Non-biometric identification methods rely on personal information such as name, address, social security number, or passport number. Biometric identification, on the other hand, uses physical or behavioral characteristics that can be uniquely assigned to an individual, such as fingerprints, facial features, iris features, or voice. Both non-biometric and biometric methods have their advantages and disadvantages, and the choice of approach depends on the particular use case and the level of security required. For example, non-biometric methods are easier to implement and may be sufficient for lower security applications, while biometric methods are more secure and are often used in high-security situations such as border control or access to secure facilities.

Identification of individuals is important in everyday life for a variety of reasons. It helps prevent unauthorized access to sensitive information, secure facilities, or financial transactions. Biometric identification, in particular, provides an additional layer of security because physical and behavioral characteristics are unique to each individual and are difficult to replicate or forge. In addition, personal identification enables the creation of user accounts for online services, mobile devices, and other systems that provide quick and easy access without the need to remember usernames and passwords. By using person identification, organizations can also ensure they have the correct information about an individual, reducing the risk of errors and improving the accuracy of their records. In scientific studies involving human subjects, accurate person identification is critical to ensure that the correct data is collected and linked to the correct individual. In addition, person identification is used to track patient data and monitor the progress of clinical trials to ensure that patients receive the correct treatments and that results are recorded accurately. In addition, person identification helps maintain the privacy and security of sensitive data such as personal health information or genetic data.

1.1. Biometric Person Identification

A biometric is a unique, measurable characteristic of an individual [1] that can be used to verify his or her identity. Biometrics are used as an alternative to traditional identification methods such as passwords, PINs, or smart cards because they provide a more secure and convenient way to verify the identity of the individual. Some of the commonly used biometrics are shown in Figure 1.1. Biometrics are unique to each individual and are difficult to replicate or forge, making them less vulnerable to identity theft or fraud compared to traditional identification methods. Furthermore, each biometric characteristic is permanent for the individual in a given period of time, i.e., the biometric characteristic is the same throughout the time period. The gait of an individual varies between childhood and adulthood but is constant in a specific period of time, e.g., adulthood.

Biometric data can be collected using a variety of sensors, such as cameras, microphones, radars, and touch sensors. The collected data is then processed to extract unique features or patterns that can be used for identification. There are numerous biometric features that are used to identify individuals.

Fingerprints are one of the most widely used biometrics. It utilizes the unique patterns of ridges and indentations on the surface of the individual's fingertips and uses a special sensor to capture these patterns. The consistent shape of fingertips over time and various environmental factors make the fingerprint biometric a very robust and reliable method

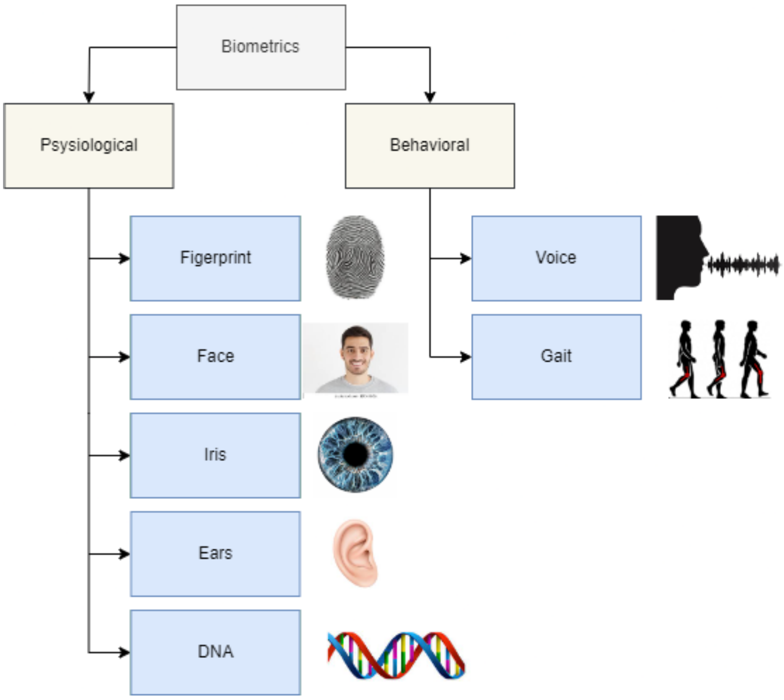


Figure 1.1: Overview of the commonly used biometrics for identifying individuals

of identifying an individual, especially considering the small size of the template and its inexpensiveness. However, cuts and deformations of the fingertip can cause problems in recognition. In addition, the fingertip can be easily replicated on the sensor using various replication techniques, which makes the fingertip biometric susceptible to impersonation attacks. Also, the individual must physically access the identification system, making the fingerprint an intrusive biometric. Biometric fingerprints are used in law enforcement, access control, and border control.

Face biometric refers to the use of facial recognition technology to identify or verify the identity of an individual. This technology captures and analyzes various features of the face, such as the distance between the eyes, nose, and mouth, as well as the shape and size of the face. Unlike fingerprints, the face biometric is non-intrusive, meaning the individual does not have to interact with the identification system. In addition, no special sensor is needed to capture the biometric feature, as normal cameras can be used for this purpose. Facial recognition has gained popularity in recent years thanks to its unobtrusiveness, easy storage of biometric templates, and fast identification process. However, the various accessories and facial occlusion can significantly limit the biometric effectiveness, as well as in cases of unstable ambient lighting or various facial expressions encountered during data capturing process. Biometric facial recognition is used in surveillance, human-computer interaction, access control, and marketing.

Another popular biometric is the iris biometric. The iris represents the colored ring of tissue that surrounds the pupil of the eye and contains unique patterns that are unique to every individual. An identification system based on iris biometric captures and analyzes the unique characteristics of the iris, such as its texture, pattern, and pigmentation, to create an appropriate representation of the individual's iris that can be used to verify their identity. Iris biometric technology has proven to be highly accurate, reliable, and non-invasive, making it suitable for a variety of applications. It is resistant to wearing glasses or contact lenses and remains constant over time. However, iris-based detection requires the individual's cooperation in data collection, must be performed at a close distance from the sensor, and is susceptible to various diseases that cause changes in the individual's iris. Iris biometric is used in access control in more secure areas such as military compounds or law enforcement facilities, border control, and healthcare.

Similar to face biometric, the individual's ears can also be used for reliable identification [2, 3]. The ear biometric uses the unique characteristics of ear geometry, such as shape, size, and contours, to create a unique representation for identification. It is a non-intrusive biometric technique, and data collection can be performed with ordinary cameras. However, ear biometric is sensitive to environmental factors such as varying illumination levels during data acquisition, occlusions of part or all of the ear, pose variations, and finally the presence of various accessories on the ear itself, such as earrings. The ear biometric is also used in access control, law enforcement, and surveillance.

Gait biometric refers to the use of walking patterns to identify or verify the identity of an individual. Gait analysis captures and analyzes various characteristics of an individual's gait pattern, such as stride length, foot placement, and stride frequency, to create a unique gait signature that can be used for identification or authentication purposes. Gait biometric has several advantages over other biometric modalities, as it is non-invasive, can be captured from a distance, and does not require physical contact with the individual. A detailed description of gait biometric will be presented in Chapter 2.

Voice has also emerged as a biometric for identifying or verifying individuals based on their unique vocal characteristics. By analyzing various parameters such as pitch, tone, and frequency, it is possible to correctly identify the individual. Voice biometrics is non-invasive, easy to use, and can be implemented in real-time, making it suitable for a wide range of applications. The biometric feature can be captured using ordinary microphones. However, voice biometric can be affected by ambient noise, the voice can be easily faked, and some diseases can affect voice consistency. Voice biometric is used in access control, healthcare, and entertainment.

DNA biometric refers to the use of genetic information to identify or verify the identity of an individual. DNA, or deoxyribonucleic acid, is the genetic material that carries the unique genetic code of an individual. DNA biometric technology analyzes specific regions of an individual's DNA to create a unique genetic profile that can be used for identification or authentication purposes. DNA biometric technology has several advantages over other biometric modalities, as it is highly accurate, reliable, and resistant to tampering or alteration. However, obtaining the DNA sample is complicated and the process is lengthy, the financial cost is very high, and real-time matching is impossible. DNA biometric is used in law enforcement investigations, medical research, and victim identification.

In biometric identification systems, a sample of an individual's biometric data is compared against a stored reference sample to verify their identity. This comparison can be performed using various algorithms and techniques, such as pattern recognition, statistical analysis, or neural networks.

Biometric person identification is performed in a series of steps. First, a person's biometric data is collected using sensors such as cameras, microphones, radars, or various touch sensors. This data is then processed to extract unique features or patterns that can be used for identification. Second, the extracted biometric data is used to create a reference template that is a representation of the individual's unique biometric characteristics. This template is stored in a database and can later be used for identity verification. Third, when an individual attempts to verify their identity, a new sample of their biometric data is collected and compared to the stored reference template. The comparison involves matching the new sample with the stored reference template to determine if the biometric data matches. Fourth, based on the result of the comparison, it is decided whether the identity of the person has been verified. Fifth, over time, the biometric templates may need to be updated to reflect changes in an individual's biometric characteristics. This can be done by collecting a new sample of the biometric data and using it to update the stored reference template.

Biometric person identification systems typically use algorithms and techniques such as pattern recognition, statistical analysis, or neural networks to perform the comparison and decision-making processes. The accuracy and efficiency of these systems are dependent on several factors, such as the quality of the collected biometric data, the complexity of the comparison algorithms, and the size and diversity of the reference database.

Biometric identification of individuals can be performed using different approaches, such as physical or behavioral biometrics. Physical biometric uses measurable physical characteristics of an individual, such as fingerprints, facial recognition, iris recognition, hand geometry, and others. Behavioral biometrics uses the behavioral traits of an individual, such as typing rhythm, gait, and voice. Furthermore, multiple types of biometrics can be combined into a multimodal approach that uses a combination of physical and behavioral biometrics to increase the accuracy and security of person identification. Each of these approaches has its own advantages and disadvantages, and the choice of approach depends on several factors, such as security requirements, target population, and the operational environment. In practice, many biometric person identification systems use a combination of approaches to increase the accuracy and security of person identification.

Given the data, person identification approaches can be divided into two categories: image-based and non-image-based approaches. Image-based biometric person identification is a method of identifying individuals that uses images of an individual's physical or behavioral characteristics to establish their identity. The biometric data is acquired using the camera sensors in the form of an image or a video sequence. The camera sensor can be an ordinary RGB sensor, an infrared sensor, a thermal sensor, or a camera with depth-sensing capability. The most common biometrics that uses the image-based procedure of data capturing are fingerprints, face, iris, and gait. The process typically involves capturing an image of the individual's biometric data and comparing it to an existing reference image to determine if there is a match. Image-based biometric identification of individuals is used in many areas due to its convenience, accuracy, and ease of interpretation, including border control, access control, mobile device security, and criminal investigations.

Non-image-based biometric person identification is a method of personal identification that uses non-visual data to establish an individual's identity. The process typically involves taking measurements or observations of the physiological or behavioral characteristics of an individual and comparing them to an existing reference to determine if a match exists. Typical sensors used for non-image data collection include embedded devices attached directly to the individual, touch sensors, pressure sensors, and gyroscopes. Some examples of non-image-based biometric person identification are voice, signature, keystroke dynamics, and DNA matching. Non-image-based biometric person identification is widely used in various applications, such as secure enrollment, mobile device security, and financial transactions, due to its accuracy and ease of use.

While biometric person identification has many advantages over traditional methods of identification, such as passwords or ID cards, it also has some limitations and challenges.

The biometric systems can be deceived by presenting a fake biometric sample. This is called biometric spoofing, using, for example, a fake fingerprint or facial image to fool the system. In the case of fingerprints, it is relatively easy to replicate one's fingertip patterns, by using readily available tools and materials to take a print of the fingertip and present it to the system. In face recognition, an image of the targeted person can be shown to the system, and if the system is lacking the ability to recognize a living person, the system can be fooled.

Accessories that are present in the targeted biometric can also drastically alter the biometric appearance and render it unusable under certain conditions. For example, by wearing earrings on the ear, the ear recognition system may not be able to correctly identify the individual. In the case of face recognition, wearing glasses or a medical mask can have the same effect.

Additionally, some of the biometric features, such as facial recognition or iris scanning, may not work effectively for an individual with certain physical characteristics, such as those with facial disfigurements or visual impairments.

Furthermore, poor sensor resolution limits the ability of the system to capture the biometric data in enough detail that is required for successful and reliable identification results, which is especially true for image-based approaches to biometric identification. For example, in the face recognition task, if there is not enough detail in the captured facial data, the system will be unreliable and will be unable to produce accurate results.

Environmental factors also greatly influence the biometric data collection process. The environmental noise present during data capture can significantly alter the data, making it difficult to process and, in extreme cases, rendering it unusable. For example, when dealing with image-based data, the illumination changes across sensors and different scenes drastically change the individual's appearance, which in turn causes the biometric system to assume they are two different individuals. In addition to lighting changes, weather conditions also play an important role.

Moreover, occlusions that obscure a part of the data also drastically affect the ability of the biometric system to recognize the individual. In some cases, the data is not visible at all, which makes the biometric identification system unusable in certain conditions. For example, in face recognition, only a part of the individual's face may be captured, and in fingerprint recognition, only a part of the fingertip may be captured.

In image-based biometric person identification, the different angles from which the biometric data is acquired also play an important role. When the image of a biometric feature such as the face, ear, or gait is captured from different viewpoints, the individual's appearance change significantly, which may result in the biometric identification system not being able to match the correct individuals, due to too big of a variation between the data samples of the same individual captured from the different viewpoints.

1.2. Gait Recognition

Gait recognition is the task of identifying individuals based on their walking patterns, depicted in Figure 1.2. It is a type of behavioral biometrics that uses the unique characteristics of an individual's gait, such as stride length, step cadence, and body movements, to identify them.

Gait recognition systems typically use video surveillance cameras to capture a person's walking motion and extract features such as the length and angle of their strides, the speed and rhythm of their gait, and the movement of their torso and arms. The features are then typically formed into a template, and the template is then compared to a database of previously stored templates to identify the individual.

Considering the image-based sensors, the gait biometric is typically captured using an RGB cameras, depth sensors, and infrared sensors. In non-image-based gait recognition, pressure plates, gyroscopes, and wearable devices, as well as smartphones and their inertial sensors are used.

These sensors capture information about the movement of the individual's legs, hips, and torso, which are then processed, by e.g. machine learning algorithms, to create a

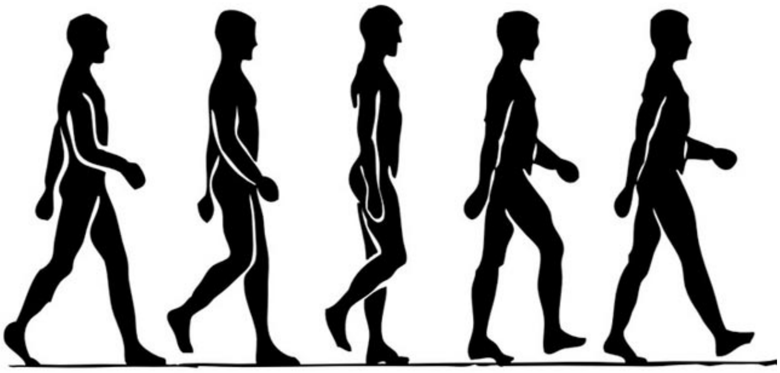


Figure 1.2: Gait Biometric

unique template.

Compared to other biometrics, gait biometric has several advantages [4]. First, it is a non-invasive biometric, meaning that the person does not need to interact with the identification system in any physical way, which can be important in situations where hygiene is a concern or when identifying individuals who may be ill or contagious. Active participation or knowledge from the individual is not required, in contrast to, for example, fingerprint or iris recognition, which requires active participation from the individual being identified. Second, gait patterns are unique to each individual and do not change significantly over time, making it difficult for imposters to replicate or spoof the biometric. Third, the gait biometric can be acquired from a distance and with a low-resolution sensors, making it suitable for applications in a typical surveillance scenario. Fourth, gait is robust to the various illumination changes in the environment, and is less affected by changes in facial appearances, such as wearing makeup or various accessories.

However, gait biometric also has limitations and challenges, such as privacy concerns, accuracy in different environments, and potential bias or discrimination based on physical characteristics or movement impairments. Furthermore, environmental factors can affect gait recognition accuracy significantly. Also, different covariates on a person present one of the main problems in gait recognition. Moreover, in image-based gait recognition, in real-world use cases, the person is recorded from multiple cameras at different angles, which results in a significantly different appearance of the person, which further complicates the process of gait recognition.

Gait recognition can be applied to various domains, such as border control, access control, and forensic investigations. However, it also raises privacy concerns, as it can be used for covert surveillance without an individual's knowledge or consent. Additionally, there are challenges in developing accurate and reliable gait recognition algorithms that can work in different environments and with different individuals, especially in terms of different covariates and multiple viewpoints at which the individuals are recorded.

1.3. Deep Learning Approaches for Gait Recognition

Based on the type of learning, the gait recognition deep learning approaches can be divided into supervised, and unsupervised learning. In supervised learning, the deep learning model is trained using annotated data, which includes both the input data (gait images or videos) and the corresponding output (the identity of the individual). Some common supervised learning techniques for gait recognition include CNNs, RNNs, and Siamese networks. The goal of supervised learning is to guide the model to learn useful gait features by providing the training data and the target labels. In unsupervised learning, the deep learning model is trained using unannotated data, without any explicit output labels. The goal of unsupervised learning is to identify patterns or structures in the data. Some common unsupervised learning techniques for gait recognition include autoencoders and GANs. The choice of the best approach depends on the availability and quality of annotated and unannotated data, as well as the specific requirements of the application. Supervised learning is typically used when annotated data is available, while unsupervised learning is useful for identifying patterns in large, unannotated datasets.

However, in recent years, a new learning approach was proposed, called self-supervised learning. Self-supervised learning combines supervised and unsupervised approaches in a way that it creates a pretext task, where a part of the input data is hidden, and the task is to predict the missing data, creating a supervised task. At the same time, it is unsupervised, as the data labels are not present, and the model trains using the training data itself. The self-supervised approach is able to learn from a large amount of data without the need of annotating the data.

In real-world use cases, the data needed for training the deep learning models is often present. In gait recognition, a vast amount of surveillance and other videos are present online, publicly available, that contain video sequences of individuals walking in various settings and environments. With suitable preprocessing steps, the said data is suitable for training the deep learning models. In view of supervised learning, the main problem is the annotation of the data, which makes the mentioned data unusable without the significant cost of time and money needed for annotation.

There are several types of deep learning models used for gait recognition. Convolutional Neural Networks (CNNs) are a type of deep neural network commonly used for image recognition. In gait recognition, CNNs are used to analyze gait images or videos and extract features that are unique to each individual's gait pattern [5, 6, 7].

Recurrent Neural Networks (RNNs) are designed to process sequential data, making them useful for analyzing gait patterns over time. RNNs can be used to analyze time series data from sensors or video footage and identify the unique features of an individual's gait [8].

Siamese networks are designed to compare two images or sequences and identify whether they are from the same individual or not. In gait recognition, Siamese networks can be used to compare the gait patterns of an individual captured at different times or locations and identify whether they match [9].

Generative Adversarial Networks (GANs) are designed to generate new data that is similar to a given dataset. In gait recognition, GANs can be used to generate synthetic gait images or videos that can be used to train deep learning models [10].

Autoencoders are a type of neural network that can learn to compress and reconstruct data. In gait recognition, autoencoders can be used to learn a compact representation of an individual's gait pattern that can be used for identification [11].

These deep learning models can be used alone or in combination to improve the accuracy and robustness of gait recognition systems. However, the choice of the best approach depends on the specific requirements of the application and the available data.

Following the success of Transformer architecture in the domain of text prediction, the Vision Transformer (ViT) deep learning model was recently proposed for the task of image classification [12]. The ViT model proved to be on par with the state-of-the-art CNN models, with a number of advantages. First, the ViT model has a larger receptive field in the lower layers, which results in learning more robust features in lower layers, leading to qualitatively different features that are being learned. Second, the model is computationally more efficient than CNNs of similar size. Considering the model's accuracy, the ViT model achieved great results on the ImageNet image classification benchmark [13], comparable to CNNs. However, the ViT model has not yet been extensively evaluated on gait recognition task in the current literature.

1.4. Scientific Hypothesis and Contributions

So far, the gait recognition research focused its attention on the supervised learning of deep learning models. Although supervised learning is easier to train than, for example, unsupervised learning, its applications in real-world use cases is limited. The limitation comes from the fact that data needs to be annotated in order to train the models. Data annotation is a very expensive process, both timewise and financially. In order to alleviate the mentioned limitation, in this dissertation, a self-supervised approach is proposed for training the gait recognition deep learning model without the need for annotating the data.

Furthermore, the recent research in the field of gait recognition is based primarily on CNNs. CNNs showed great results in the gait recognition task, and many different variants of the network were proposed. However, CNNs suffer from the problems of a narrow receptive field and have large computational complexity. In recent years, a new architecture was proposed, abbreviated ViT, that, instead of convolutions (as CNNs) bases its inner workings on the mechanism of self-attention. Compared to CNNs, ViTs have stronger modeling capability, enabling the model to incorporate more global information than a similar CNN model, since the receptive field of the ViT model is larger than the CNN model in the lower layers. Also, the computational complexity of the ViT model is lower than that of a similar CNN model, leading to a more computationally efficient model architecture.

Combining the above remarks, the hypothesis of this doctoral dissertation is proposed:

Using self-supervised learning and a self-attention deep learning model it is possible to obtain accurate gait recognition.

Along with two sub hypotheses:

- *With the self-supervised learning approach it is possible to learn discriminative gait features without using data annotations.*
- *Gait recognition using a self-attention deep learning model is robust to multiple viewpoints and covariate problems.*

Following, the scientific contributions of this dissertation are:

1. Selection procedure for gait feature extraction models based on supervised and self-supervised learning.
2. Selection procedure for gait feature classification algorithm.
3. A novel approach for gait recognition using a self-supervised self-attention deep learning model.

1.5. Research Methodology

The research in this dissertation has been conducted in several phases, each of the phases contributing to the defined scientific contributions.

In the first phase of research, the datasets for gait recognition were examined. Many datasets are available today, and each dataset has its specific use case in gait recognition, and one of the goals of this phase of research was to determine which datasets are suitable for use in this research. The criteria used in the dataset selection procedure included: the quality of the data, the amount of the data in the dataset, the number of individuals, and the modalities that the dataset has. The modalities that are a focus of this dissertation are covariates in the form of different accessories that an individual wears during the walk, and the different camera viewpoints of the individual walking. The selected datasets include different covariates, multiple viewpoints, or both. After selection, the datasets were analyzed and prepared to be used in deep learning model training.

The second phase of the research consisted of studying and implementing the self-supervised deep learning method and the ViT deep learning model. As the self-supervised learning method, the DINO method [14] was selected. Its inner workings were analyzed and studied thoroughly, to gain knowledge about how the method behaves with respect to the gait recognition task. Furthermore, the ViT model was examined in detail, as well as its underlying mechanism of self-attention. Both the DINO method and ViT model were implemented and applied to the gait recognition problem. The result of this phase is the implemented framework of the feature extraction model.

Throughout the third phase, the ablation study of both the DINO method and ViT deep learning model hyperparameters was performed. One of the goals of this phase was to find approximately the best hyperparameters of the network, for use in the gait recognition task. Moreover, the feature extraction models were trained and analyzed with respect to the accuracy they obtained, and a detailed analysis of learned representations together with other performance metrics was performed.

In the fourth phase, different classification algorithms were analyzed in the task of classification of gait features extracted from the feature extraction model. Different classification algorithms were implemented, such as the kNN algorithm that belongs to the domain of non-deep learning classification algorithms, as well as FCNN, a machine-learning algorithm. The classification algorithms were analyzed with respect to the accuracy they obtained, and a mutual comparison was performed. This phase results in a qualitative analysis of the classification algorithms used in this dissertation.

1.6. Dissertation Overview

The aim of this dissertation is to explore the efficacy of using a self-supervised self-attention deep learning model for the task of person identification using gait recognition. Through nine chapters, the dissertation delves into various aspects, techniques, and challenges of gait recognition, demonstrating the effectiveness of the proposed approach.

Chapter 1. outlines the importance of biometric person identification and application areas of gait recognition. This chapter also defines the objectives of the dissertation, hypotheses, and scientific contributions, and highlights the motivation behind the chosen topic and the challenges associated with the domain.

In Chapter 2., the concept of identifying individuals using gait recognition is described. This chapter explores the foundation of gait recognition as a biometric identifier, various sensors, and data used for performing gait recognition, the way the gait features are represented, and outlines a typical gait recognition pipeline.

Chapter 3. discusses various supervised and unsupervised learning approaches, as well as several machine learning models, that are used in this domain.

In Chapter 4., a comprehensive literature review is provided, examining past and current research in gait recognition. It analyses their methodologies, outcomes, and limitations, offering a broader perspective on the current research landscape.

The datasets used in this study are detailed in Chapter 5., with elaboration on their sources, their characteristics, and the reasons for their selection.

Chapter 6. describes the proposed approach implemented in this dissertation. It includes a description of the data preprocessing techniques utilized in this dissertation, a description of the chosen self-supervised learning method, the ViT feature extraction deep learning model, and the proposed FCNN classifier. Finally, the performance metrics used to evaluate the trained models are outlined.

The details regarding the datasets used in the experiments are described in Chapter 7. Furthermore, the details of the experiments that were conducted are outlined, together with the implementation details of the feature extraction models and a classifier, and the evaluation protocol.

Chapter 8. provide a comprehensive report on the findings of the study. It discusses the performance of the proposed approach, including detailed accuracy metrics, a comparison with previous works, and a discussion of the results. The ablation study is also analyzed, focusing on the model pretraining and the differences between supervised and self-supervised learning in the conducted experiments.

Finally, Chapter 9. summarizes the dissertation, providing a summary of the research conducted in this doctoral dissertation. It also outlines future directions, suggesting possible ways to enhance the accuracy and efficiency of person identification using gait recognition.

2. Chapter

PERSON IDENTIFICATION USING GAIT RECOGNITION

In this chapter, the details of gait biometric will be described, its advantages compared to other biometrics, as well as particular problems that arise when using gait biometric for person identification. Moreover, the sensors and data used in gait recognition task will be outlined, together with a typical gait recognition pipeline.

2.1. Gait Biometric

Gait refers to the manner in which an individual walks or runs. It is a complex series of movements that involves the coordination of the brain, bones, and muscles, with support from the heart and lungs. Gait analysis is the systematic study of human motion, that measures body movements, body mechanics, and the activity of the muscles.

Human motion involves several sequences of a gait cycle. A gait cycle is defined as the duration from one repeating locomotion event to the next identical one [15]. This complex activity engages the entire body and demands the synchronized operation of a multitude of muscles and joints within the musculoskeletal framework. Predominantly, it involves the actions of the lower and upper extremities, along with movements of the pelvis and spine.

The gait cycle, depicted in Figure 2.1, can be broken down into two primary phases, the stance phase and the swing phase, which alternate for each lower limb. The stance

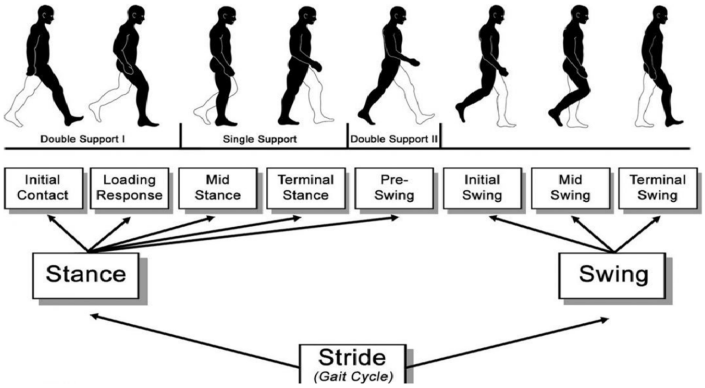


Figure 2.1: The Gait Cycle [16]

phase consists of the entire time that a foot is on the ground and bearing body weight. It can be further divided into several subphases, including initial contact, loading response, mid-stance, terminal stance, and pre-swing. The swing phase consists of the entire time that the foot is in the air. It can be further divided into initial swing, mid-swing, and terminal swing. During the stance phase, the foot is in contact with the ground, and the body's weight is transferred onto it. This phase begins with initial contact when the heel of the foot first touches the ground. The loading response follows, during which the foot rolls forward and the body's weight is transferred onto it. During mid-stance, the body's weight is directly over the foot and the leg is supporting the body. Terminal stance occurs as the body moves forward and the heel of the foot begins to lift off the ground. Pre-swing is the final subphase of the stance phase, during which the toes leave the ground and the foot is no longer bearing weight. During the swing phase, the foot is not in contact with the ground and is moving forward to prepare for the next step. This phase begins with the initial swing, during which the foot is lifted off the ground and begins to move forward. During mid-swing, the foot continues to move forward and the knee begins to straighten. Terminal swing is the final subphase of the swing phase, during which the foot is positioned for initial contact with the ground.

The characteristics of gait, more specifically the unique features of the gait cycle, can be used for individual identification. For example, by using information about various gait parameters, such as stride velocity, step length, stride length, cadence, step width, and angle, it is possible to reliably identify an individual [17]. Besides the mentioned gait parameters, other data can be used for identification, as mentioned in the Chapter 2.2.

2.1.1. Advantages of Gait Recognition

As mentioned in Chapter 1.2., gait biometric has several advantages over other biometrics. First, the gait biometric is a passive biometric modality, meaning it does not require active participation from the user. Unlike fingerprint or facial recognition, which require the user to present their fingerprint or face, gait biometric can identify individuals from a distance without the interaction of the individual with the sensor. Second, gait biometric is more difficult to spoof than other biometrics. It is challenging to mimic someone else's walking pattern, making it a robust biometric modality. Third, it can be used with low-resolution sensors, such as surveillance cameras, in situations where the image resolution is low, and the details are scarce. Fourth, gait is robust to the various illumination changes in the environment, as well as different covariates that are often present on individuals in real-world scenarios.

2.1.2. Problems in Gait Recognition

Biometric features can vary significantly depending on the conditions under which the image was captured. Poor image quality can significantly affect the accuracy of biometric identification systems. Issues such as too-low resolution, occlusions, and variations in lighting conditions can lead to incorrect recognition or failure to recognize an individual.

Furthermore, one of the main problems in image-based gait recognition is the fact that the gait information of individuals is often acquired from the sensors that have different viewing angles of the individual. The sensor perspective has a significant role in the appearance of the gait patterns of an individual, resulting in drastically different patterns for the same individual at different viewing angles. For example, the side view of an individual's gait highlights the movement of the legs, while the information about the leg's movement at the front view is significantly harder to distinguish. Consequently, comparing the two mentioned views is a very challenging problem, since its appearances are significantly different. Furthermore, the sensors that acquire the data often have different intrinsic parameters, such as sensor resolution, focal length, number of frames captured in a second, etc., that further harden the problem, since the data acquired by the two different sensors is often different.

In real-world gait recognition use cases, the covariates play an important role in the recognition accuracy. Covariates represent various factors that cause variations in individuals walking patterns. These factors introduce challenges in accurately identifying individuals based on their gait, as the extracted features might be affected by these variations rather than the unique characteristics of the individual's gait. One of the most common covariates in gait recognition is clothing. Changes in clothing can alter the appearance of an individual's silhouette, affecting the extracted gait features. For example, wearing a coat or a skirt can cause the silhouette to look different from the individual's usual appearance. Different carrying conditions also present a significant problem in typical use cases. Carrying objects, such as bags or briefcases, can change person's walking pattern, as it may affect their arms swing or body posture. Different types of shoes can influence an individual's walking pattern, particularly if they cause changes in walking posture, stride length, or foot placement.

As it will be discussed in Chapter 4., the state-of-the-art approaches to the gait recognition problem are based on deep learning. Deep learning approaches rely on the data for constructing the model for gait recognition, and the quality of the data directly influences the model performance. Although, the volume of the data plays the most important role in constructing an accurate deep learning model. As deep learning models become more complex, the demand for a large amount of data is increasing. In gait recognition, large volumes of data are actually present in various publicly available sources, in the form of surveillance footage, movies, etc. That data contains information about various individual's gait patterns, recorded at different image qualities, different angles, different sensors, different individuals, etc. The main problem with the available data is that it is not annotated, i.e. the information to which identity each image sample belongs is missing. As the prevailing approach for gait recognition in deep learning is a supervised approach, the lack of annotations presents a problem that results in the inability to use those large volumes of data.

2.1.3. Applications of Gait Recognition

Gait recognition has been gaining popularity in recent years due to its potential applications in various fields. Some of the most promising applications of gait recognition are discussed below.

Gait recognition can be used in security and surveillance systems to identify individuals at a distance without their knowledge. This can be particularly useful in areas where traditional biometric systems, such as fingerprint or facial recognition, are not suitable or feasible, such as crowded public spaces. Gait recognition can help law enforcement agencies to monitor and track suspects or persons of interest, enhance security measures at airports, borders, and other critical infrastructure, and prevent unauthorized access to restricted areas.

Gait recognition can also be used in healthcare to monitor and diagnose various medical conditions, such as Parkinson's disease, Alzheimer's disease, and multiple sclerosis. Gait patterns are known to change in individuals suffering from these diseases, and gait recognition can provide an objective and non-invasive way of detecting these changes. By analyzing the gait patterns of patients, healthcare professionals can monitor the progression of the disease and adjust treatment plans accordingly.

Gait recognition can be used in sports science to analyze the walking and running patterns of athletes to improve their performance and prevent injuries. By analyzing the gait patterns of athletes, coaches, and trainers can identify areas of weakness, such as muscle imbalances or improper form, and develop targeted training programs to address these issues. Gait recognition can also be used to monitor athletes' progress and track their recovery from injuries.

2.1.4. Gait Recognition Settings

There are two main types of gait recognition: constrained and unconstrained. Constrained gait recognition involves analyzing an individual's walking pattern in a controlled environment, such as a laboratory or a specific walking path. The walking conditions are standardized, and the individual's movements are captured using specialized cameras or sensors placed at specific locations along the walking path. Constrained gait recognition is commonly used for research purposes and has higher accuracy rates compared to unconstrained gait recognition due to the controlled environment.

Unconstrained gait recognition, on the other hand, involves analyzing an individual's walking pattern in a real-world environment, such as a busy street or a crowded shopping mall. The walking conditions are not controlled, and the individual's movements are captured using surveillance cameras or other devices without their knowledge. Unconstrained gait recognition is more challenging than constrained gait recognition due to the variations in walking patterns caused by different factors such as clothing, footwear, walking speed, and walking surface. However, it has more practical applications, such as surveillance and security, and is an active area of research in computer vision and machine learning.

In this doctoral dissertation, constrained gait recognition will be primarily explored in the experiments, using the data acquired from the controlled environment. Such data enables easy comparison with the other state-of-the-art approaches since the environmental factors are known and described in detail. Also, the data is well-annotated, especially considering the various covariates and angles at which the individuals are recorded, enabling reliable analysis of the results.

2.2. Sensors and Data used in Gait Recognition

The success of gait recognition is predicated on the application of sensors and data acquisition mechanisms that capture and process the locomotive patterns of individuals. This chapter offers a comprehensive analysis of the pivotal elements - sensors and their corresponding data - that substantiate the efficacy of gait recognition systems.

2.2.1. Sensors

Different types of sensors are used to capture the various signals that characterize the human gait. These include accelerometers, gyroscopic sensors, magnetometers, force sensors, extensometers, goniometers, active markers, electromyography, etc. Gait data can also be collected using camera sensors, such as an RGB camera, depth sensors, such as a Microsoft Kinect, or inertial sensors, such as an accelerometer. Depending on the use case and the specific application, the sensors can be located at the individual's body, or collect the data from a distance.

These sensors and data can be divided into different categories, including wearable devices, image-based sensors, and contact-based sensors [4].

Accelerometers are commonly used sensors in gait recognition [18, 19, 20, 21], which measures the acceleration of the body during walking. The data collected by accelerometers are utilized to extract various features, such as step length, step frequency, and walking speed. Accelerometer data can be collected using different wearable devices, such as smartwatches or fitness trackers, which can be worn on the wrist or leg.

Gyroscope [22, 23] is another type of sensor that measures the orientation of the body. This sensor helps to extract features such as the angle of the foot and the swing angle of the leg during walking. The data obtained from gyroscopes can be used in combination with accelerometer data to gain a more accurate representation of an individual's gait.

Pressure sensors are sensors that measure the force exerted on the ground by the feet. These sensors are useful in obtaining features such as heel strike time, toe-off time, and the contact area of the foot during walking [24, 25]. Pressure sensors can be incorporated into shoe insoles or floor mats to collect the data of an individual's gait.

Kinematic data is another type of data that can be used in gait recognition. This data is obtained through motion capture systems that use cameras to track the movement of markers placed on the body. The kinematic data obtained from motion capture systems can be used to extract features such as joint angles and angular velocities [26, 27]. This type of data provides a more comprehensive understanding of an individual's gait.

Radar sensors are capable of measuring the distance and velocity of objects using radio waves. In gait recognition, radar sensors are used to capture the movement of an individual's body parts during walking [28, 29]. This data can be used to extract features such as stride length, stride frequency, and gait speed. The advantage of radar sensors over other types of sensors is their ability to operate in adverse weather conditions and low-light environments, making them suitable for outdoor applications. They also have a longer range and wider field of view, allowing the data to be captured from a larger area.

Camera sensors are commonly used in gait recognition to capture and analyze video footage of an individual's walking patterns [30, 31, 32, 33, 34]. These sensors provide a more comprehensive understanding of gait and can be used to extract features such as stride length, stride time, and foot placement. They can also be used to track the movement of markers placed on the body for motion capture analysis. In current gait recognition literature, camera sensors are the most common method of acquiring gait data from individuals, due to their ease of acquisition and its ease of setup. In this dissertation, the ordinary RGB camera sensors are utilized for acquiring the gait data, i.e. all the data used consists of the data acquired by the said sensors.

Multimodal approaches are increasingly being used in gait recognition to improve the accuracy and robustness of the identification process [5, 35, 36, 37]. Multimodal approaches refer to the use of multiple sensors or data types to analyze an individual's gait pattern. Combining multiple sources of data, such as accelerometers, gyroscopes, pressure sensors, kinematic data, camera sensors, and radar sensors, can provide a more comprehensive understanding of an individual's gait. For example, using both camera sensors and pressure sensors can provide more accurate measurements of foot placement and timing during walking. Multimodal approaches can also help to overcome the limitations of individual sensors or data types. For instance, while accelerometer data provides accurate measurements of acceleration during walking, it may not provide sufficient information on joint angles and angular velocities. Combining accelerometer data with kinematic data obtained from motion capture systems can provide a more complete picture of an individual's gait. Furthermore, multimodal approaches can improve the robustness of gait recognition systems by reducing the impact of noise or errors from individual sensors or data types. The use of multiple sensors or data types can help to mitigate errors or inconsistencies that may occur due to sensor malfunction or variability in individual walking patterns. In conclusion, multimodal approaches are becoming increasingly important in gait recognition to improve the accuracy and robustness of identification processes. By combining multiple sensors or data types, a more comprehensive understanding of an individual's gait can be obtained, leading to more reliable identification and authentication in various applications such as security, healthcare, and sports performance analysis.

2.2.2. Data representation

Considering the image-based gait recognition problem that is addressed in this dissertation, two different data representations are typically created, model-based and model-free. Model-based gait recognition aims to identify individuals based on their unique walking patterns, by constructing a parametric model that captures the essential characteristics of an individual's body structure and movement during walking. Various approaches have been proposed in the literature, each offering a unique perspective on representing and analyzing human gait.

In the 3D human body model approach, the human body is represented as a set of interconnected rigid segments, often referred to as links or limbs [38]. These segments correspond to body parts such as the torso, upper and lower arms, and upper and lower legs. The model also includes joints, which connect the segments and allow for movement. The 3D human body model aims to capture the subject's overall motion by estimating the position, orientation, and size of each segment and the joint angles during walking.

Articulated body models are similar to 3D human body models but incorporate more detail about the skeletal structure, including the number and arrangement of joints. These models often include parameters such as bone lengths, joint locations, and joint limits, which can provide a more accurate representation of an individual's walking pattern [39]. In some cases, inverse kinematics techniques are used to estimate joint angles from the observed motion of body parts.

Geometric models focus on representing the human body using simple geometric primitives, such as cylinders, ellipsoids, or spheres [40]. These primitives are used to approximate the shape and volume of body parts, making it easier to compute certain gait features, such as the center of mass and angular momentum. Geometric models are often less computationally intensive than other methods, making them suitable for real-time applications.

Skeleton or pose-based model-based gait recognition is a subcategory of model-based gait recognition that focuses on the analysis of human body joint positions and their relationships to identify individuals based on their unique walking patterns. This approach leverages the skeletal structure of the human body, extracting the pose information from video sequences and using it to characterize the individual's gait. The first step involves extracting the human body's skeletal structure from video sequences. This is achieved using human pose estimation algorithms, which detect and localize body joints (e.g., head, shoulders, elbows, wrists, hips, knees, and ankles) in each frame. These algorithms can be either 2D or 3D, depending on the level of detail required and the availability of depth information. Recent advances in deep learning, particularly CNNs and graph convolutional networks (GCNs) [41], have significantly improved the accuracy and robustness of pose estimation. Once the pose information is obtained, gait features can be extracted from the skeleton data. These features can include joint angles, distances between joints, joint velocities, and temporal patterns of joint movements. Additional higher-level features can also be derived, such as stride length, cadence, and step frequency. The chosen features should be discriminative and invariant to factors like clothing, viewpoint, and walking speed. Skeleton-based model-based gait recognition has several advantages over alternative methods, such as its robustness to changes in clothing and viewpoint, and its relative insensitivity to occlusions. However, it can still be sensitive to factors like footwear, walking surface, and carrying objects, which may affect joint movements.

Generally, model-based methods typically offer good robustness to changes in clothing, viewpoint, and walking speed, as the parametric models capture the underlying structure and movement of the body. Model-based methods are also more interpretable. The use of human body models makes the extracted features more interpretable and allows for a better understanding of the biomechanical and physiological factors that influence gait. Furthermore, the parametric models can be adapted to different levels of detail, depending on the application requirements and the available data.

Despite several advantages, model-based gait recognition is computationally complex. Model-based methods often require more computational resources due to the complexity of the models and the algorithms used for parameter estimation. Also, these methods can be sensitive to inaccuracies in the body model or the estimation of model parameters, which can lead to errors in gait feature extraction and classification.

Model-free gait recognition is an approach to gait recognition that aims to identify individuals based on their unique walking patterns without explicitly modeling the underlying body structure and movement. Instead, this approach utilizes machine learning algorithms to automatically extract features from gait data.

The first step involves capturing video footage of individuals walking, ideally under controlled conditions (e.g., constant lighting, fixed camera position, and a straight walking path). High-resolution videos with a high frame rate are preferred, as they can provide detailed information about the subject's movements. This second step involves cleaning and preparing the video data for subsequent analysis. Common preprocessing tasks include background subtraction, noise reduction, and temporal alignment of video frames. The goal is to isolate the individual's silhouette from the background, providing a clear representation of their walking pattern. Then, gait features are extracted directly from the video data without explicitly modeling the body structure or movement. Common features used in model-free methods include Gait Energy Image (GEI), Motion History

Image (MHI), and Optical Flow.

The GEI [42] is a spatial-temporal representation of the silhouette, obtained by averaging the binary silhouette images over a complete gait cycle. The GEI captures the overall shape and motion of the individual during walking. In this dissertation, the GEI image data is used as data, and as such the GEI image generation will be thoroughly explained in the Chapter 6.1. The MHI [43] is another spatial-temporal representation, which captures the motion history of the individual by assigning higher intensity values to the most recent movements. The MHI can provide information about the dynamics of the walking pattern. Chrono-Gait Image (CGI) [44] is a temporal feature extraction method that captures the temporal changes in the silhouette. It is obtained by stacking the horizontal or vertical projections of the binary silhouettes. Optical flow represents the apparent motion of objects in the video sequence, which can be used to capture the relative motion between the individual and the background. Optical flow features can provide information about the speed and direction of walking.

Model-free gait recognition approaches have several advantages compared to model-based approaches. First, model-free methods are generally simpler and faster to implement, as they do not require the creation and estimation of complex body models. Second, these methods can be more robust to inaccuracies in pose estimation or tracking, as they typically rely on holistic features extracted directly from the video data. Third, model-free methods can better handle occlusions and self-occlusions by using holistic features, such as silhouettes or motion history images, which can still provide useful information even when parts of the body are occluded. Another advantage of the model-free approach is its ability to automatically extract features from gait data, reducing the need for manual feature selection. This approach also allows for the identification of subtle differences in gait patterns that may not be easily observed or quantified using other techniques.

Although model-free approaches generally have better results than model-based approaches, several disadvantages exist. Model-free methods can be more sensitive to changes in clothing, viewpoint, and walking speed, as they do not explicitly model the underlying structure and movement of the body, yet they rely on the visual data which is changed in the presence of various covariates or when viewed from the different angle. Furthermore, the features used in model-free methods are often less interpretable and may not provide a clear understanding of the biomechanical and physiological factors that influence gait. Moreover, the model-free approach may be limited by the quality and quantity of the gait data collected. The accuracy and robustness of the classifier depend heavily on the quality and diversity of the data used to train the algorithm. Additionally, the model-free approach may not provide a detailed biomechanical profile of an individual's gait pattern, which may limit its use in clinical applications.

2.3. Typical Gait Recognition Pipeline

The typical stages of gait recognition involve the following steps: data acquisition, preprocessing, feature extraction, representation learning, and classification. As in this dissertation the image-based approach is used, the general steps for mentioned approach will be described.

2.3.1. Data acquisition and preprocessing

The first stage involves capturing video sequences of subjects walking. Typically, a camera or multiple cameras are placed at an appropriate height and distance to ensure a clear view of the subject's gait. The data acquisition process involves capturing raw video data, which can be represented as a sequence of frames:

$$V = F_1, F_2, ..., F_n, \tag{2.1}$$

where F_i is the $i-th$ frame and n is the total number of frames in the sequence.

The preprocessing stage aims to remove noise and irrelevant information from the raw video data. This stage typically involves background subtraction and silhouette extraction. Background subtraction aims to extract the subject from the background, in order to retain only useful information from the image. Common techniques include frame differencing, running average, and Gaussian mixture models [45]. After background subtraction, the silhouette extraction is typically performed, by extracting the individual's silhouette in binary representation for each frame. This can be achieved by applying a threshold to the difference image.

Preprocessing also involves segmenting the data into individual steps or gait cycles to enable the extraction of relevant features that describe an individual's gait pattern. Gait cycle determination is an essential step in gait recognition as it provides a standardized temporal framework for the analysis and comparison of walking patterns. A gait cycle is defined as the time interval between two successive occurrences of the same event in the walking process, such as heel-strike or toe-off, of the same foot. Gait cycles allow for the temporal alignment of walking sequences, ensuring that the extracted gait features are comparable across different individuals and walking conditions. This alignment enables the classification algorithms to focus on the inherent patterns of the gait rather than being influenced by temporal misalignments.

Normalization is another common preprocessing technique used in gait recognition. Normalization is used to account for differences in the physical characteristics of individuals, such as height and weight, which can affect gait patterns. Spatial normalization focuses on scaling the gait features to account for differences in body size and camera distance. This process typically involves scaling the silhouette or body joint positions to a fixed height, width, or area.

Spatial normalization ensures that the extracted features are invariant to variations in body size and camera distance, allowing for a more accurate comparison of walking patterns across individuals. Temporal normalization aims to standardize the duration of the gait cycle, making the extracted features invariant to differences in walking speed. This process often involves resampling the gait features at a fixed number of equally spaced points within the gait cycle or normalizing the features with respect to the duration of the gait cycle (e.g., stride length per unit of time).

Temporal normalization allows for a fair comparison of gait features extracted from walking sequences with different speeds, improving the recognition performance. Intensity normalization refers to the process of adjusting the intensity values of the gait features, such as the Gait Energy Image (GEI) or Motion History Image (MHI), to a standardized scale or range. This normalization can be achieved by scaling the intensity values to a fixed range (e.g., 0 to 1) or by normalizing the values with respect to a reference value (e.g., the maximum intensity value in the image). Intensity normalization helps reduce the influence of variations in lighting conditions and background, allowing for more accurate comparisons of gait features across different environments.

2.3.2. Feature Extraction

In this stage, unique features are extracted from the preprocessed gait sequences, by generating a suitable gait data representation. There are various gait data representation methods, including spatial, temporal, and spatiotemporal approaches. Some popular methods include Gait Energy Image (GEI), Chrono-Gait Image (CGI), and Motion History Image (MHI).

Gait Energy Image (GEI) is a popular spatiotemporal feature representation for gait recognition, which was introduced by Han and Bhanu [42]. It represents the average of the silhouette images of a walking subject over a gait cycle. The gait energy image effectively captures the spatial and temporal characteristics of an individual's gait, making it a powerful tool for gait recognition. As the GEI images are used in this dissertation, the generation of the GEI data will be described in Chapter 6.1.

A multi-channel temporal encoding technique, known as Chrono-Gait Image (CGI) [44], encodes a gait sequence into a multichannel image, ensuring the preservation of the temporal information of gait patterns. Instead of silhouettes, the contours of the individuals are extracted, to preserve the spatial information. After the contour extraction, a linear interpolation function is used to encode the spatio-temporal information to k channels. Each frame is assigned different weights across channels, according to the frame's position in time. For each frame in the gait sequence the multichannel gait contour image C_t is generated [44], and the final representation is calculated as follows [44]:

$$CGI(x,y) = \frac{1}{p}\sum_{i=1}^{p} PGI_i(x,y), \tag{2.2}$$

where p is the number of 1/4 gait cycles, and PGI_i is the sum of the total multichannel contour images in the i-th 1/4 gait cycle.

The Motion History Image (MHI) [43] is constructed by accumulating the temporal information of a moving object's silhouettes over a specified duration. Each silhouette image, $I_t(x,y)$, is a binary image where the value 1 represents the foreground (moving object) and 0 represents the background. The MHI, $M(x,y)$, represents the motion history at each pixel location (x,y), where recent motion is assigned a higher value, and older motion decays over time. This results in a single image that effectively captures the temporal information of the moving object, which can then be used for motion analysis tasks such as gait recognition. Given a binary silhouette image sequence $I_t(x, y)$, where t denotes the frame index and (x, y) denotes the pixel coordinates, the Motion History Image $M(x, y)$ is calculated as follows:

$$M(x,y) = \begin{cases} T, & if\ I_t(x,y) = 1 \\ \max(0, M(x,y) - 1), & otherwise \end{cases}, \tag{2.3}$$

where T is the total number of frames in the motion history.

However, in many deep learning-based approaches, the feature extraction step from raw silhouettes into some form of template-based feature representation is not necessary. Deep learning models often perform representation learning directly on the raw silhouette data, without using any form of template-based feature extraction. By using deep networks, it is possible to capture the dynamic nature of gait from the raw data, since the networks are able to handle high dimensional data well.

2.3.3. Representation learning

After the features were extracted from the raw gait sequences, the representation learning process is employed. In representation learning, the algorithm is trained to select the most relevant features from the input data.

In the traditional methods that are not based on deep learning, this step usually receives some form of template data as input, such as a GEI image. Then, using some dimensionality reduction techniques such as Principal Component Analysis (PCA) [46] or Linear Discriminant Analysis (LDA) [47], the input template representations are reduced to the most relevant features. The most relevant features are determined through training the dimensionality reduction algorithm.

More advanced algorithms, such as various types of neural networks, are also used for learning the discriminative features. Neural networks can accept both template-based features and raw silhouettes as input data. By iterating over input silhouettes or templates of individual's gait, the deep networks are able to learn discriminative features that usually outperform the discriminatory power of features learned by employing simpler algorithms such as PCA or LDA. As output, the final features are produced, on which the classification is performed.

In this step, the feature extraction algorithm is trained on a subset of individuals from the whole dataset, and the subset is usually called the training subset. By using the training subset the goal is to learn useful representations that discriminate between different individuals most effectively.

2.3.4. Classification

The final stage of gait recognition involves classifying the extracted features using a suitable classifier. A classifier maps the feature vectors to their corresponding class labels (i.e., subject identities). Popular classifiers used in gait recognition include k-Nearest Neighbors (kNN) [48], Support Vector Machines (SVM) [49], and machine learning-based approaches, such as feed-forward neural networks, and CNNs.

The kNN classifier works by finding the k nearest training instances in the feature space to a given test instance and assigning the most common class label among these neighbors. The distance between instances can be calculated using various distance metrics, such as Euclidean distance, Manhattan distance, or Mahalanobis distance. It is one of the most used classifiers in gait recognition, especially in methods that do not rely on deep learning. Furthermore, the weighted kNN classifier is also used in literature [50, 51], and as such the same will be also used in this dissertation, and will be denoted as kNN.

The Weighted k-Nearest Neighbor (WkNN) classifier is an extended form of the standard k-Nearest Neighbor (kNN) classifier. It incorporates a weighting mechanism for each of the k neighbors based on their proximity to the query point.

In the traditional kNN classifier, for a given query point, the method finds the k nearest points (neighbors) from the training data set and allocates the query point to the class that is most frequently represented within these neighbors. However, this method does not consider the varying distances of these neighbors from the query point.

In contrast, the WkNN classifier addresses this issue by assigning weights to the neighbors. The weight of a neighbor directly impacts the class allocation, and it is inversely proportional to its distance from the query point. This concept can be mathematically represented as follows:

Let's symbolize the distance between the query point x_q and a neighbor x_i as $d(x_q, x_i)$.

We'll use w_i to represent the weight associated with the neighbor x_i, where $i = 1, 2, ..., k$.

One common method for determining the weight is to use the reciprocal of the distance, so:

$$w_i = \begin{cases} \frac{1}{d(x_q, x_i)}, & \text{for } d(x_q, x_i) \neq 0, \\ \infty, & \text{for } d(x_q, x_i) = 0. \end{cases} \tag{2.4}$$

The prediction y_q for the query point x_q is then obtained by weighted majority voting:

$$y_q = \underset{y}{\operatorname{argmax}} \sum_{i=1}^{k} (w_i \cdot I(y_i = y)), \tag{2.5}$$

where $I(y_i = y)$ is an indicator function that is equal to 1 if $y_i = y$ and 0 otherwise, y_i is the class label of neighbor x_i, and $\underset{y}{\operatorname{argmax}}$ represents the class y that maximizes the summation. If the weights are equal (i.e., all neighbors are at the same distance from the query point), the WkNN classifier reduces to the standard kNN classifier.

The WkNN classifier provides a more refined method for classifying unknown samples by taking into account the relative distance of each neighbor. This approach often leads to improved classification accuracy, particularly when the distribution of data is not uniform. Nevertheless, it also introduces an additional computational cost associated with the calculation of weights.

Besides kNN, the simple feed-forward neural network is often used for gait classification. Usually, the FCNN is comprised of three layers, the input layer which represents the input data, the hidden layer(s) where the relationships in input data are weighted, and the output layer where the final probability distribution for given individuals is given. Compared to kNN, the FCNN has the ability to model complex relationships among the input data, and as a result, usually has higher accuracy.

In the classification step, a subset of individuals different from the training set used in the feature selection process is selected from the dataset. The test subset is divided into gallery and query subsets. The gallery subset contains the known individuals and their gait representation, while the query subset contains the gait representations of the individuals for which the goal is to find the matching gait representation from the gallery subset. The goal of this split is to test the feature selection algorithm on new data, i.e. new individuals different from the ones on which the algorithm is trained, and evaluate the generalization ability of the algorithm. Furthermore, another goal is to evaluate the possible over-fitting of the algorithm on the training data.

Finally, the evaluation of the proposed approach is performed, by analyzing the classification performance. The main performance metric in gait recognition is ranked accuracy, followed by F1-score, recall, and other appropriate metrics.

3. Chapter

MACHINE LEARNING APPROACHES FOR GAIT RECOGNITION

Machine learning is a field of study that focuses on developing algorithms and statistical models that can enable computers to automatically learn from data without being explicitly programmed to do so. The process involves feeding the computer with large amounts of data and using this data to train the model to recognize patterns and relationships.

The key concept underlying machine learning is the idea of a model. A model is a mathematical representation of the underlying relationships in the data. The goal of machine learning is to build a model that can accurately predict outcomes based on new input data that it has not seen before. There are many different techniques and algorithms used in machine learning, including decision trees, neural networks, support vector machines, and random forests.

Deep learning is a specific type of machine learning that involves the use of artificial neural networks, which are computational models inspired by the structure and function of the human brain. Deep learning models use multiple layers of artificial neurons to progressively extract higher-level features from the input data. The process involves feeding the data into the input layer, which then passes the information through the layers of neurons to the output layer. The layers in between are called hidden layers, and they are responsible for processing the information and extracting relevant features from the data.

The training process in deep learning involves adjusting the weights and biases of the neurons in the network to minimize the error between the predicted output and the actual output. This is done using an optimization algorithm such as stochastic gradient descent [52]. Deep learning has been successful in a wide range of applications, including image and speech recognition, natural language processing, and autonomous vehicles.

Machine learning algorithms aim at solving a particular task. There are many tasks usually employed in machine learning, such as regression, classification, dimensionality reduction, etc. In regression, the goal is to predict a continuous target variable based on one or more input features, and for evaluation, the Mean Squared Error (MSE) metric is commonly used. In classification, the goal is to predict a categorical target variable. Classification can be binary (two classes) or multiclass (more than two classes), and for evaluation, the metrics such as accuracy, precision, recall, and F1-score are used. In dimensionality reduction, the machine learning model reduces the input data to lower dimensional feature space, while preserving the most important information in the data.

In this doctoral dissertation, gait recognition is considered a classification task. More specifically, multiclass classification is performed, where the multiple classes represent the number of individuals in the database of known individuals.

A typical deep learning system is a part of machine learning methods that models high-level abstractions in data by using multiple processing layers, with complex structures or otherwise, composed of multiple linear and non-linear transformations. It employs various types of architectures such as fully connected neural networks, CNNs, recurrent neural networks (RNNs), long short-term memory networks (LSTM) [53], and more. Each of these architectures is suitable for different kinds of data: for instance, CNNs are primarily used for image processing, while RNNs and LSTMs are designed to handle sequential data like time series or natural language.

These architectures consist of an input layer for data ingestion, one or more hidden layers for data processing and extraction of complex features, and an output layer for making the final prediction or classification. During the learning process, these systems use optimization algorithms (like gradient descent) and a loss function to adjust the parameters of the model and reduce the difference between the predicted and actual output. Activation functions introduce non-linearity into the system, helping it learn from complex patterns.

3.1. Types of Machine Learning Systems

Considering the way that machine learning algorithms work, they can be divided into two main categories: supervised and unsupervised learning approaches. Furthermore, as a special case of an unsupervised approach, self-supervised learning exists.

3.1.1. Supervised Learning

Supervised learning is a type of machine learning where an algorithm is trained on a annotated dataset to predict an output based on an input. In other words, given a set of inputs (features) and their corresponding outputs (annotations), the algorithm learns a mapping function that can predict the output for new input data. This is achieved by providing the model with a annotated dataset, which consists of pairs of input-output examples. The goal of supervised learning is to create a model that can make accurate predictions for previously unseen data based on the patterns it has learned from the annotated training data.

Mathematically, supervised learning can be described as learning a function f that maps input data X to output data Y:

$$f : X \to Y. \tag{3.1}$$

Let's assume a dataset of N samples, each with D features and a corresponding target variable. The input data can be represented as a matrix $X \in \mathbb{R}^{N \times D}$, where each row represents a sample and each column represents a feature:

$$X = [x_1, x_2, \ldots, x_D]_N \,. \tag{3.2}$$

The target variable can be represented as a vector $y \in \mathbb{R}^N$, where each element corresponds to the target for the corresponding sample.

$$y = [y_1, y_2, \ldots, y_N]\,. \tag{3.3}$$

The goal of supervised learning is to learn a function f that maps inputs to outputs. This function is typically represented as a model with a set of parameters, denoted by w:

$$f(X; w) \approx y. \tag{3.4}$$

The model is trained to minimize the error between the predicted output and the true output. This error is typically measured using a loss function, denoted by L, which quantifies the difference between the predicted output and the true output:

$$L(y, f(X; w)). \tag{3.5}$$

The optimization problem can be formulated as finding the set of parameters w that minimizes the loss function over the training data:

$$\min_{w} : L(y, f(X; w)). \tag{3.6}$$

This optimization problem can be solved using an optimization algorithm, such as gradient descent or its variants. The optimization algorithm computes the gradient of the loss function with respect to the weights and updates the weights in the direction of the negative gradient:

$$w_{ij} = w_{ij} - \eta \frac{\partial L}{\partial w_{ij}}, \tag{3.7}$$

where w_{ij} is the weight connecting neuron j to neuron i, η is the learning rate (a small positive scalar), and $\frac{\partial L}{\partial w_{ij}}$ is the gradient of the loss function with respect to the weight w_{ij}.

Once the model is trained, it can be used to predict the output for new input data. This is done by feeding the new input data into the trained model, which outputs a predicted value.

The most common methods that use supervised learning are Decision Trees, Random Forest algorithm, SVM, ANNs, and CNNs.

Supervised learning is used in a wide range of applications, such as image classification, speech recognition, and natural language processing. It is a powerful tool for making predictions based on historical data and has the potential to uncover patterns and relationships that may not be apparent to humans.

3.1.2. Unsupervised Learning

Unsupervised learning is a type of machine learning in which the model is trained on unannotated data, with the goal of discovering patterns, structures, or relationships in the data. Unlike supervised learning, which relies on annotated data to train a model to predict outputs given inputs, unsupervised learning is used to identify hidden patterns or relationships within the data without prior knowledge of the outputs.

Mathematically, unsupervised learning involves learning a function f that captures the structure or distribution of the input data X:

$$f : X \to H, \tag{3.8}$$

where H represents the hidden structure, representation, or feature of the data.

One popular method of unsupervised learning is clustering, where the model identifies groups of similar data points based on some similarity metric. A simple example of clustering would be to group together data points with similar x and y coordinates on a 2D plane. The k-means clustering algorithm [54] is one of the most common methods used for clustering, which involves iteratively updating the centroid of each cluster until convergence.

Another common method of unsupervised learning is dimensionality reduction, which involves reducing the number of features or variables in the data while preserving the most important information. This is particularly useful when dealing with high-dimensional data where there may be many irrelevant or redundant features that can be removed without affecting the overall accuracy of the model. Principal component analysis (PCA) is a popular technique for dimensionality reduction that involves identifying the principal components of the data, which are linear combinations of the original features that capture the most variance in the data. t-distributed stochastic neighbor embedding (t-SNE) [55] is another example of a dimensionality reduction technique, used primarily for visualization of high-dimensional features.

Mathematically, unsupervised learning can be formulated as an optimization problem where the goal is to minimize a certain objective function or cost function. For example, in clustering, the objective function might be to minimize the distance between data points within the same cluster and maximize the distance between data points in different clusters.

For example, in the k-means algorithm, this can be expressed as:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2, \tag{3.9}$$

where J is the objective function to be minimized, k is the number of clusters to form, C_i is the set of data points belonging to cluster i, x is a data point in the dataset, and μ_i is the centroid of cluster i.

Similarly, in dimensionality reduction, the objective function might be to minimize the reconstruction error between the original data and the lower-dimensional representation.

For example, in t-SNE, the objective function is to minimize the Kullback-Leibler (KL) divergence between the two probability distributions using gradient descent. t-SNE (t-Distributed Stochastic Neighbor Embedding) is a non-linear dimensionality reduction technique particularly suitable for visualizing high-dimensional data in a low-dimensional space (usually 2D or 3D). The main idea behind t-SNE is to preserve the local structure of the data by minimizing the divergence between two probability distributions: one in the high-dimensional space and the other in the low-dimensional space. It achieves this by measuring pairwise similarities between data points and then attempting to maintain these similarities in the lower-dimensional space.

The t-SNE algorithm can be summarized as follows:

1. Compute pairwise similarities between data points in the high-dimensional space using a Gaussian probability distribution.

2. Compute pairwise similarities in the low-dimensional space using a Student's t-distribution with one degree of freedom (also called the Cauchy distribution).

3. Minimize the Kullback-Leibler (KL) divergence between the two probability distributions using gradient descent.

The KL divergence between the high-dimensional distribution P and the low-dimensional distribution Q is given by:

$$C = \mathrm{KL}(P \parallel Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}, \tag{3.10}$$

where p_{ij} represents the pairwise similarity between data points x_i and x_j in the high-dimensional space, and q_{ij} represents the pairwise similarity between their corresponding points y_i and y_j in the low-dimensional space. The objective of the t-SNE algorithm is to minimize the KL divergence C with respect to the low-dimensional points $Y = \{y_1, y_2, ..., y_n\}$.

In the t-SNE algorithm, P and Q are probability distributions, p_{ij} and q_{ij} are the pairwise similarities, and C is the KL divergence to be minimized. The main goal is to obtain a low-dimensional representation of the data that preserves the local structure as much as possible by minimizing the divergence between the two distributions.

3.1.3. Self-supervised Learning

Self-supervised learning is a type of machine learning that involves training models on data in a semi-supervised manner, where the annotations or targets for the data are generated automatically from the data itself. The key idea behind self-supervised learning is to leverage the inherent structure and relationships in the data to create proxy annotations or targets, which can then be used to train the model.

Self-supervised learning is often used in situations where annotated data is scarce or expensive to obtain. By using the data itself to generate annotations, self-supervised learning can enable models to learn from large amounts of unannotated data, which can be more readily available.

One common approach to self-supervised learning is to use data augmentation to create pairs of similar and dissimilar examples. For example, in image classification, a pair of similar examples might be two images that are slightly different views of the same object, while a pair of dissimilar examples might be two images from different classes. The model is then trained to predict whether the two examples are similar or dissimilar, based on the structure and relationships in the data.

Another approach to self-supervised learning is to use pretext tasks, which involve training models on tasks that are related to the ultimate objective, but do not require any external annotations or targets. For example, in natural language processing, a model might be trained to predict the missing word in a sentence, or to predict the order of words in a shuffled sentence. The model can then be fine-tuned on a downstream task, such as sentiment analysis or text classification, using annotated data.

Mathematically, self-supervised learning involves learning a same function f from Equation 3.8 that captures the structure or distribution of the input data X by additionally solving a pretext task T. The task T is used to define the learning objective, and the function f is learned in the way that it is optimized to learn the pretext task T.

In the context of deep learning, this function f is typically represented by a neural network with a specific architecture and weights (parameters). The learning process involves iteratively updating the weights to minimize an objective function, which quantifies the quality of the learned structure, representation, or distribution for the pretext task.

Common self-supervised learning tasks include autoregressive models and contrastive learning. Autoregressive models predict a part of the input data based on the remaining parts. For example, in natural language processing, autoregressive models like GPT [56] aim to predict the next word in a sequence given the preceding words. The objective function is often a cross-entropy loss between the true and predicted words:

$$L(X, \hat{Y}) = -\sum_{i} x_i \cdot \log(\hat{y}_i), \tag{3.11}$$

where x_i is the true word (in one-hot encoded form), and $\hat{y}_i$ is the predicted probability distribution over words.

In contrastive learning, the goal is to learn representations by comparing similar and dissimilar data points. It aims to bring representations of similar instances closer together while pushing apart dissimilar instances in the embedding space. Contrastive learning typically employs a Siamese architecture, where two or more instances are processed through the same neural network to generate embeddings. Contrastive learning has shown significant success in computer vision tasks, such as image classification and object recognition. The objective function is often a contrastive loss like the Noise Contrastive Estimation

(NCE) loss or the InfoNCE loss [57]:

$$L(X, \hat{X}) = -\sum_i \log \left(\frac{\exp(sim(x_i, \hat{x}_i))}{\sum_j \exp(sim(x_i, \hat{x}_j))} \right), \tag{3.12}$$

where x_i and $\hat{x}_i$ are a similar pair of data points, $\hat{x}_j$ are dissimilar data points and sim is a similarity function, such as cosine similarity or dot product.

Another popular pretext task used in self-supervised learning is known as autoencoding. In this task, the model is trained to learn a compressed representation of the input data that can be used to reconstruct the original data. The model is trained by minimizing the reconstruction error between the original data and its reconstructed representation. Variants of autoencoding, such as denoising autoencoders and contractive autoencoders, have been developed to improve the quality of the learned representations.

Mathematically, the autoencoding objective function can be expressed as:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \|x_i - g(f(x_i, \theta), \theta)\|^2, \tag{3.13}$$

where θ is the set of model parameters, N is the number of data points, x_i is the input data at time i, f is the encoder function that maps the input data to a compressed representation, g is the decoder function that maps the compressed representation back to the original data, and $\|\cdot\|$ is the Euclidean distance.

Recently, many methods have been proposed that are based on self-supervised learning. SwAV (SWapping Assignments between Views) [58] is a self-supervised learning approach that optimizes the assignments of data samples to prototypes in a manner that maximizes consistency between different views of the same image. It employs online clustering with a Sinkhorn-Knopp algorithm to enforce an equal assignment of samples to clusters. SwAV learns visual representations by minimizing the divergence between the assignments of different views of the same image while maintaining a uniform distribution over the cluster assignments. This method has demonstrated competitive performance in various computer vision tasks, such as image classification and object detection.

SimCLR (Simple Contrastive Learning of Visual Representations) [59] is a method that learns representations by maximizing the agreement between different augmentations of the same image. It uses a contrastive loss function, where positive pairs are derived from different augmentations of the same image, and negative pairs are derived from augmentations of different images. SimCLR has shown significant improvements in representation learning and transfer learning tasks, outperforming several supervised and self-supervised methods.

MoCo (Momentum Contrast) [60] is another self-supervised learning approach that leverages contrastive learning with a dynamic memory bank. It maintains an online encoder and a momentum-updated encoder to generate embeddings for the current view and the memory bank, respectively. MoCo optimizes the consistency between the embeddings of different views of the same image while treating other instances in the memory bank as negative samples. The method has been effective in various computer vision tasks, including unsupervised pretraining and transfer learning.

BYOL (Bootstrap Your Own Latent) [61] is a self-supervised learning method that learns representations by predicting the latent embedding of one view of an image from the latent embedding of another view of the same image. It employs an online network with an encoder and a predictor and a target network with an encoder only. The method encourages consistency between the embeddings generated by the online and target networks using a contrastive loss. BYOL has been shown to be effective in learning powerful image representations without the need for negative samples or a memory bank.

One advantage of self-supervised learning over unsupervised learning is that it provides a way to evaluate the performance of the model. Since the pretext task is derived from the data itself, the performance of the model on the pretext task can be used as a proxy for the quality of the learned representations. Additionally, self-supervised learning can be used to pretrain models on large amounts of unannotated data, which can be especially useful in domains where annotated data is scarce.

Self-supervised learning has shown promising results in a variety of domains, including computer vision, natural language processing, and speech recognition. However, there are still challenges associated with self-supervised learning, such as the choice of pretext tasks and the generalization of the learned representations to new tasks and domains.

3.2. Machine Learning Models

There are many different model types in deep learning, each with its own unique architecture and mechanisms for processing the data. These models range from lower to higher complexity, in terms of the model size and the number of parameters in the model. Some of the most common deep learning architectures include artificial neural networks (ANNs), convolutional neural networks (CNNs), and the Vision Transformer (ViT).

3.2.1. Artifical Neural Network

An artificial neural network (ANN) is a type of machine learning model that is inspired by the structure and function of the human brain. The term neural network refers to the mimicking of the human brain neurons, that are connected in a graph-like structure. ANNs are composed of multiple layers of interconnected nodes, or artificial neurons, that can learn to perform complex computations on input data. Each neuron in an ANN receives input from other neurons and applies a mathematical function to the input to produce an output, which is then passed on to other neurons in the network. The ANN is a directed graph structure, in which all the elements perform some simple calculation.

The basic building block of an artificial neuron is the perceptron, which takes a vector of inputs x and produces a single output y. The perceptron applies a linear function to the input, followed by a non-linear activation function:

$$y = f(wx + b), \tag{3.14}$$

where w is a vector of weights that determines the strength of the connections between the input and the neuron, b is a bias term that determines the neuron's activation threshold, and f is the activation function.

The activation function is typically a non-linear function, such as the step function, sigmoid function, hyperbolic tangent function, or rectified linear unit (ReLU) [62] function. The purpose of the activation function is to introduce non-linearity into the model, which enables the ANN to learn complex relationships in the data.

Multiple perceptrons can be combined into a layer, where each perceptron receives the same input and produces a different output. The outputs of the layer are then passed on to the next layer, where the process is repeated.

Generally, the layers can be divided into input layer, hidden layer(s), and output layer. The input layer receives the raw data as input and passes it to the next layer. The number of neurons in the input layer corresponds to the dimensionality of the input data. Hidden layers are intermediate layers that perform non-linear transformations on the input data. There can be one or multiple hidden layers in an ANN, depending on the complexity of the problem and the architecture. The output layer produces the final predictions or decisions based on the learned features from the input and hidden layers. The number of neurons in the output layer depends on the number of classes or targets in the task.

Neurons in adjacent layers are connected by weighted edges, which represent the strength of the connections. The weights are learnable parameters that the ANN adjusts during the training process. The weighted sum of the input signals to a neuron can be calculated as:

$$z_i = \sum_j w_{ij} x_j + b_i, \tag{3.15}$$

where z_i is the weighted sum for neuron i, w_{ij} is the weight connecting neuron j to neuron i, x_j is the output of neuron j, and b_i is the bias term for neuron i.

The objective of training an ANN is to minimize the discrepancy between the predicted output and the ground truth. This discrepancy is measured using a loss function, such as cross-entropy loss for classification tasks, that can be represented as:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_i [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]. \tag{3.16}$$

To minimize the loss function, the ANN's weights and biases are adjusted using gradient-based optimization algorithms, such as stochastic gradient descent (SGD) or Adam [63]. The gradients are computed using the backpropagation algorithm, which calculates the gradient of the loss function with respect to each weight and bias by applying the chain rule for differentiation:

$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial y_i} \cdot \frac{\partial y_i}{\partial z_i} \cdot \frac{\partial z_i}{\partial w_{ij}}, \tag{3.17}$$

where $\frac{\partial L}{\partial w_{ij}}$ is the gradient of the loss function with respect to weight w_{ij}, and the other terms are the partial derivatives of the respective variables.

ANNs can be used for a wide range of tasks, including classification, regression, and image recognition. They have been applied to many domains, including natural language processing, speech recognition, and computer vision, among others.

3.2.2. Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of artificial neural network that is designed to process and classify data that has a grid-like structure, such as images or videos. CNNs are inspired by the organization of the visual cortex in animals, where simple cells detect edges and complex cells detect more complex features.

CNNs use a series of convolutional layers to extract features and learn spatial hierarchies of features from the input data. A convolutional layer applies a set of learnable filters, or kernels, to the input data to produce a set of output feature maps. Each filter slides over the input data, computing the dot product between its weights and a small patch of the input data, effectively capturing local spatial patterns.

Every kernel has its receptive field, which refers to the local region of the input data that a kernel “sees”. In CNNs, the kernel size determines the dimensions of the receptive field, e.g. a 3 × 3 kernel has a receptive field of 3 × 3 pixels.

Given an input matrix I and a filter matrix K, the convolution operation is defined as:

$$S(i,j) = (I \circledast K)(i,j) = \sum_m \sum_n K(i+m, j+n) \circledast I(m,n), \tag{3.18}$$

where S(i, j) represents the output feature map S at the position (i,j) in the output feature map, and $\circledast$ denotes the convolution operation.

The output of each filter is then passed through a non-linear activation function, to introduce non-linearity into the model, allowing it to learn complex patterns. Following the convolution operation, an element-wise activation function is applied to the output feature map. The Rectified Linear Unit (ReLU) is a widely used activation function in CNNs, defined as:

$$f(z) = \max(0, z). \tag{3.19}$$

The output of the convolutional layer is typically followed by a pooling layer. Pooling layers are used to reduce the spatial dimensions of the feature maps while preserving the most important features. This helps to reduce computational complexity and control overfitting. The most common pooling operation is max-pooling, which selects the maximum value from a given region. Given a feature map M and a pooling window size $p \times p$, the max-pooling operation is defined as:

$$MaxPool(M)(i, j) = \max(M(i' : i' + p - 1, j' : j' + p - 1)), \tag{3.20}$$

where (i', j') represents the top-left corner of the pooling window, and (i, j) denotes the position in the output pooled feature map.

The output of the pooling layer is then passed through one or more fully connected layers. Fully connected layers are used to perform high-level reasoning and produce the final output. These layers are responsible for combining the features learned by the convolutional and pooling layers to make predictions. The output of a fully connected layer can be calculated as Equation 3.14.

The overall architecture of a CNN typically consists of several convolutional layers, interspersed with pooling layers, followed by one or more fully connected layers.

To train a CNN, a loss function is used to measure the discrepancy between the predicted output and the ground truth. A popular choice for classification tasks is the cross-entropy loss, defined in Equation 3.16.

The CNN is trained using gradient-based optimization techniques, such as stochastic gradient descent (SGD) or Adam, by minimizing the loss function. The gradients are computed using the backpropagation algorithm, which calculates the gradient of the loss function with respect to each weight and bias by applying the chain rule for differentiation.

Beyond the core components of a CNN, there are several optional layers and techniques that can be integrated to further improve the network's performance and address specific challenges.

The batch normalization layer is a commonly used layer in CNNs. Batch normalization is a technique that helps to accelerate training and improve the overall performance of the network. By normalizing the activations within each mini-batch, it mitigates the issue of internal covariate shift, which occurs when the distribution of inputs changes during training. The normalization process is defined as:

$$BN(x) = \frac{x - mean(x)}{\sqrt{var(x) + \epsilon}}, \tag{3.21}$$

where x is the input, $mean(x)$ and $var(x)$ are the mean and variance of the input, and ϵ is a small constant added for numerical stability.

Residual connections, introduced in the ResNet [64] architecture, are a technique to address the vanishing gradient problem in deep networks. By adding skip connections that allow the input to bypass one or more layers, the network can learn residual functions and improve the flow of gradients during backpropagation. Mathematically, a residual block can be represented as:

$$F(x) = x + H(x), \tag{3.22}$$

where x is the input, $F(x)$ is the output, and $H(x)$ is the residual function learned by the intermediate layers.

These optional layers and techniques can be employed in various combinations to enhance the performance of a CNN, depending on the specific problem and the constraints of the application. By incorporating these elements, the network can learn more complex and robust features, while addressing challenges such as overfitting, vanishing gradients, and computational complexity.

CNNs have several advantages compared to the ANNs. First, CNNs are capable of learning hierarchical features from images. By stacking multiple convolution and pooling layers, the network is able to learn a hierarchical representation of the input data. Lower layers capture basic features such as edges and textures, while deeper layers learn more abstract and high-level concepts. This hierarchical learning allows the network to effectively capture complex patterns in the input data, high higher complexity than ANNs. Second, CNNs are inherently robust to translation in the input data due to the use of convolution operations and pooling layers. This property enables the network to recognize patterns and objects even when they appear in different positions within the input space. Third, CNNs are generally more robust to noise and small distortions in the input data compared to traditional ANNs, due to the use of convolution and pooling operations. These operations make the network less sensitive to small variations, allowing it to focus on the overall structure and patterns within the data.

CNNs have achieved state-of-the-art performance in a wide range of computer vision tasks, including object recognition, object detection, and image segmentation. They have also been applied to other domains, such as natural language processing and speech recognition.

4. Chapter

RELATED WORK

One of the first gait recognition systems is proposed by Niyogi and Adelson [65], in 1994. The authors propose using the specific characteristics in spatiotemporal data representation, acquired by individuals walking frontoparalled to the image plane, for identification. The data is acquired by a fixed-position camera. By using the spatiotemporal spline functions the authors model the contours of the individuals, and use that information to model a 5-segment stick model, where two sticks were used per leg and one stick for the torso. For classification, a simple weighted nearest neighbors classifier was used, with the Euclidean distance metric.

Cunado et al. [66] proposed a model-based approach for gait recognition, which uses the Hough transform to extract the lines which represent legs in the sequences of video images. To smooth the acquired data, authors use the method of least squares. The change in inclination in extracted lines is used for individual identification, and the Fourier transform analysis is used to reveal the frequency components of the change in inclination. The Fourier transform is applied to the images on which the Canny operator is applied, to produce the image that contains only the edges. After the Fourier transform, the resulting data is classified using the k-nearest neighbor classifier. In the experiments, the camera was placed at a 90° angle compared to the individual, the background was static, with controlled lightning. There were 10 individuals in the experiments, and four video sequences were collected for each individual. By using phase-weighted magnitude spectra the classification rate was 90%, compared to the 40% by using only the magnitude spectra.

BenAbdelkader et al. [31] proposed a model-based gait recognition system, where they automatically estimate the spatio-temporal gait parameters, such as stride length and cadence, from the video sequences, and use said parameters for identification. First, the background is modeled and the foreground is detected. Second, the moving object i.e. individuals are segmented, by extracting the binary silhouette and estimating their 2D position in the image. Third, the stride and cadence parameters are estimated by calculating the gait periods and distance that the individuals traveled. For classifying the former parameters, the Linear Regression model and the Bivariate Gaussian Model were used. In the experiments, 131 sequences are used, consisting of 17 people. Individuals were recorded in the outdoor setting and were walking in various paces, in a predefined trajectory. The linear regression classifier achieved the Rank 1 accuracy of around 39%, while the bivariate Gaussian classifier achieved around 30% accuracy.

Yoo and Nixon [67] proposed a new model-based markerless system for the analysis and classification of human gait. First, the authors extract the human body and the contours of the body from the video sequences, by using background subtraction technique, and simple thresholding and morphology operators for extracting the contours of the individuals. Then, the gait cycle is estimated by using gait symmetry analysis. Then, by utilizing the anatomical data, body points extraction and the tracking of moving points, the gait figures are generated, in the form of simple 2D stick figures. In a stick figure, 9 points are modeled and connected, and model the whole human body. The mean and variation of the gait angles for a single sequence for each individual are extracted, and the trajectory-based kinematic features are used for the classification. The experiments consisted of data from 100 individuals, with 7 indoor sequences for each individual. For classification, a back-propagation neural network algorithm was employed, yielding 100% accuracy on the small number of individuals (10), however, authors argue that the meaningful recognition performance was not achieved on larger population size.

Urtasun and Fua [68] were one of the first that utilized model-based 3D tracking for the gait recognition problem. They propose fitting a 3D temporal motion model to video sequences, to effectively model the gait parameters and motion while being robust to occlusions and insensitive to changes in the direction of motion. In their work, simple volumetric primitives are attached to an articulated skeleton in order to represent the human body. In the experiments, only 4 individuals are modeled using the said approach, walking at different speeds.

Han and Bhanu [42] proposed one of the first model-free gait recognition methods. In order to characterize human walking properties, the authors proposed generating a Gait Energy Image (GEI). GEI comprises both the spatial and temporal gait characteristic of an individual, in a single gait representation. First, the binary silhouettes of individuals are extracted from the RGB video sequence. Next, the gait cycles are estimated for each individual using frequency and phase estimation, and the binary silhouettes are averaged for each gait cycle. By using the template data instead of raw silhouettes, the GEI representation saves both the computation and memory requirements for performing gait recognition. For classification, authors use the combination of PCA and MDA dimensionality reduction techniques and test their approach on the USF HumanIF dataset [69], outperforming other approaches.

Several other template-based feature representation approaches exist. The MHI is another spatial-temporal representation, proposed by Ahad et al. [43], which captures the motion history of the individual by assigning higher intensity values to the most recent movements, providing information about the dynamics of the walking pattern. Chrono-Gait Image (CGI), proposed by Wang et al. [44] is a temporal feature extraction method that captures the temporal changes in the silhouette, and is obtained by stacking the horizontal or vertical projections of the binary silhouettes. Also, Liu and Zheng [70] proposed generating a template-based gait representation template, called Gait History Image (GHI).

In their approach, Lenac et al. [71] proposed tackling the problem of gait recognition by extending the standard gait features such as GEI images, with depth data. By using the fusion technique, the GEI image features are combined with the height information of the individual, which was acquired using the depth information from the input image. For the feature extraction, the PCA and LDA techniques were used, while for the classification several algorithms were utilized and compared, including kNN and SVM, obtaining promising results on the TUM-GAID dataset [72].

Yoo et al. [73] proposed one of the first shallow feed-forward neural networks for tackling the problem of gait recognition. The authors estimated fixed positions of key body points in an image and use that information for generating 2D stick figures, defined by 9 coordinate points, using the anatomical knowledge defined by Dempster and Gaughran

[74], As features, the mean and the variation of the gait angles for a single sequence for each individual as used. For training the feature extractor and for classification a simple feed-forward neural network is used, containing two layers: a hidden layer and an output layer. In the experiments, 90 individuals are comprised in a used dataset, where the individuals were split into three different categories based on the quality of the body contour acquired from the raw data, ranging from good, fair to bad. In each of the splits, 150 feature vectors were used for training and 30 vectors for testing. In mentioned scenario the recognition accuracy achieved was 83%-90%.

One of the first deep learning-based approaches for gait recognition was proposed by Yan et al. [75]. In their approach, the authors propose using a simple convolutional neural network for learning discriminative features from gait images in a supervised manner. Besides gait recognition, the goal of their approach is also to identify various attributes of an individual, such as the angle at which the individual is recorded, and the covariate condition under which the individual is recorded, such as wearing a bag or a coat. They generate GEI images from the raw data and train a simple CNN, consisting of three convolutional blocks, followed by a simple MLP layer, on the data. The goal of the approach is to identify individuals as well as identify at which angle and with what covariate was the individual recorded. The approach was evaluated on the CASIA-B dataset and achieved results that outperformed approaches that rely on hand-crafted features.

Another early work based on deep learning was presented by Feng et al. [76], where the authors proposed to use a simple CNN to estimate the individual's pose in a video frame, resulting in heatmaps. These heatmaps are then used to describe gait information in an image, and a simple recurrent neural LSTM network is used to model the gait sequence of the person based on the sequence of heatmaps. In this way, the authors argue, temporal information is better utilized than in GEI images, and they achieve good results on the CASIA-B gait recognition dataset. However, the data used in the study was limited, and the gait feature of this proposed approach is invariant only across two views.

Shiraga et al. [77] tackled the problem of view-invariant gait recognition, by using a CNN network. A simple CNN is proposed, consisting of two convolutional layers, and two FCNN layers, ending in softmax function. The authors examine gait recognition accuracy in cooperative and uncooperative settings. As data, the GEI images are used, and the

CNN is trained using only that data, and yields good results on the OU-ISIR dataset on which it was evaluated.

In another similar work, Wu et al. [6] perform a comprehensive study on cross-view gait recognition, by examining several CNN-based architectures. The authors examined how the accuracy of the proposed models changed with respect to where the features of GEI image pairs are matched, at the bottom or at the top of the network. In the study, the models were evaluated on the CASIA-B, OU-ISIR, and USF datasets, and the results outperformed the previous state-of-the-art methods.

Yu et al. [78] propose using a stacked multi-layer auto-encoder deep learning network, in order to tackle the problem of synthesizing the gait features to boost the recognition accuracy. The aim of the proposed model was to generate invariant gait features robust to the angle at which the individual is recorded, and robust to different clothing and carrying variations. By using the stacked auto-encoder network, the proposed solution was able to generate uniform GEI images, at an angle of 90°, for any input image at different angles at which the model is trained on. For classification, the PCA is employed for feature dimensionality reduction, and a kNN classifier is used to produce final results based on the reduced features, achieving good results on the CASIA-B and SZU RGB-D datasets.

Another approach, by Yu et al. [79], also tackled the problem of robustness to various variations such as view angle and various covariates such as a bag of different clothing. The authors proposed using a method called GaitGAN, based on GAN neural network, to generate invariant gait images, generated at a 90° angle, that eliminates the covariates present in the image. The method accepts any view angle as input, with any covariate, and generates uniform representation, in the form of a GEI image, at a fixed angle. However, using the said approach, useful information is potentially lost when large variations between the target and input view angles exist.

He et al. [80] also utilized the GAN architecture to learn view-specific feature representations for gait. The authors proposed using a multi-task GAN network, in addition to using the proposed multi-channel gait template, named Period Energy Image (PEI), which is a generalization of GEI image, to further boost the gait recognition accuracy involving different view angles.

Song et al. [81] proposed an end-to-end architecture to tackle the problem of gait recognition. Instead of performing several separate steps, such as silhouette segmentation, feature extraction, and feature learning, the authors proposed using a single framework that incorporates the mentioned steps, by using the CNN network, named GaitNet. GaitNet is composed of two CNN networks, where one is used for gait segmentation, and the other is used for the classification of the gait features. Combining the mentioned steps together, the training procedure is simplified and the recognition accuracy is boosted further.

Zhang et al. [82] proposed a new gait-related robust loss function, named angle center loss (ACL), for training the deep learning models, to learn discriminative gait features. Instead of learning the center for each individual, as center loss, the proposed loss learns multiple centers for each angle of the same individual, achieving better intra-subject distances. Furthermore, the authors utilize the spatial transformer network, to localize the suitable horizontal parts of the individual's body, in order to extract the gait features for that part. The horizontal parts are then concatenated and used as a final gait features representation, achieving new state-of-the-art results on the CASIA-B and OU-MVLP datasets.

In GaitSet, Chao et al. [50] proposed regarding the gait as a set that consists of independent frames. A new deep learning network is proposed, named GaitSet, that utilizes raw silhouette frames of the individual walking, to recognize the individual. The advantage of the proposed approach is that it is frame permutation invariant, enabling combining the frame of the same individuals recorded at different times and with different covariates and view angles. Furthermore, a structure called Horizontal pyramid mapping (HPM) is applied to project the set-level feature into a more discriminative space to obtain a final deep set representation. By utilizing the newly proposed approach, the results obtained were on par with the state-of-the-art on CASIA-B and OU-MVLP datasets.

In GaitPart, Fan et al. [83] analyzed the effect of different body parts in terms of recognition accuracy. Moreover, authors proposed a new temporal part-based model for gait recognition, by generating a spatiotemporal expression for each of the body parts, since the different body parts contribute to the recognition accuracy in a different amount, by using the newly proposed micro-motion capture module (MCM).

Castro et al. [5] propose using a multimodal approach, combining the gray pixels i.e. image from the video stream in grayscale, optical flow, and depth maps, to boost the recognition accuracy. A CNN network is used for training the feature extraction model, by utilizing both single modality data, and all data combined. In their experiments, authors achieved the best accuracy when combining all mentioned data modalities.

An approach using Pairwise Spatial Transformer Networks (PSTN) is introduced by Xu et al. [84], to enhance cross-view gait recognition performance by mitigating the adverse effects of feature misalignment caused by variations in viewpoints. The proposed PSTN minimizes feature misalignment prior to the recognition step, resulting in improved overall accuracy. A backbone of the proposed network is a CNN network used. When provided with a pair of corresponding gait features obtained from different source and target views, the PST estimates a nonrigid deformation field to align the features in the matching pair with an intermediate view. This approach mitigates distortion by performing registration, thereby improving the alignment compared to directly deforming features from the source view to the target view.

Recently, several new model-based approaches to gait recognition were proposed. Sokolova and Konushin [85] proposed using a pose-based CNN network for the problem of gait recognition. Instead of using silhouettes or GEI images, the authors proposed using pose-based gait descriptors for the recognition, and as input data, the optical flow information is used, by computing the optical flow maps between each pair of the consecutive frames. The deep learning network used in this approach is similar to VGG-19 [86].

Another approach leveraging the model-based approach is proposed by Liao et al. [87]. The authors proposed an approach named PoseGait, where the network extracts the pose information of the individuals from the images, which are estimated using a CNN network. Pose information is invariant to the view angles and other external factors or variations, and as such presents robust features for gait recognition. The authors performed experiments on several gait recognition datasets and proved the effectiveness of the proposed approach. However, the pose estimation process is often expensive in resources and lacking in accuracy, due to the often low resolution of the input video streams of individuals walking.

The recurrent neural networks were also employed in the gait recognition task. Sepas-Moghaddam and Etemad [8] proposed extracting the gait convolutional energy maps (GCEM) from the frame-level convolutional features, and utilizing the RNN network in order to learn useful features from the GCEM data. As input, the silhouettes of the individuals are used.

Huang et al. [88] explored the extraction of the most important frames in the gait frames. By using information weighting, the proposed network pays more attention to the high contribution frame at the input data, thus boosting the overall recognition accuracy.

In their approach, Liao et al. [89] propose a novel approach for tackling the problem of different view angles at which the individuals are recorded, by a view synthesis approach. A Dense-View GAN is proposed to model the gait attribute distribution and generate GEI images for angles that do not exist in the evaluated datasets. more specifically, the model creates the missing GEI images, from 0-180° at 1° intervals. The proposed approach enables the deep learning model to learn more discriminative features by extending the available angles in the datasets at which the individuals are recorded.

Zhao et al. [90] focused their attention on the problem of various covariates in gait recognition. The authors proposed using the pose information in their deep learning model, without complex computation for pose feature extraction, by using two independent feature extractors, extracting the body feature from silhouettes and the part features from the pose heatmaps.

Zhu et al. [91] proposed another dataset for unconstrained gait recognition, named GREW. The data was collected from the real-world scenarios, however, in this dataset the data is annotated in various forms besides the pose information, such as silhouettes and GEI images. Although the proposed dataset is aimed at the problem of unconstrained gait recognition, the authors demonstrate the effectiveness of training a gait recognition network on top of that data, to the problem of constrained gait recognition, i.e. the authors test the learned models on the datasets for controlled gair recognition, such as CASIA-B i OU-MVLP. The dataset will be described in the Chapter 5., as it is used in this doctoral dissertation for feature extraction model pretraining.

Until now, all the mentioned approaches relied exclusively on the supervised approach to learn discriminative gait features. Continuing, the approaches which do not rely exclusively on supervised learning are outlined.

In the study performed by Cosma and Radoi [92], the problem of unconstrained gait recognition is explored, by proposing a dataset with the data collected from the real-world scenarios. Furthermore, the authors proposed using a weakly supervised learning framework, WildGait, where the spatiotemporal graph CNN network is trained on the aforementioned dataset, for the gai recognition problem. The data on which the proposed network is trained included only the skeleton i.e. pose sequences obtain from the proposed dataset. The authors demonstrated the effectiveness of their approach in an unconstrained environment.

Pinčić et al. [93] proposed using the self-supervised learning method DINO for training the feature extraction and using an FCNN classifier for classifying the gait features learned by the mentioned feature extractor. By using self-supervised learning, no labels are used for training the feature extraction model, and an FCNN classifier is trained in a supervised fashion using known labels. The results achieved are on par with some of the supervised state-of-the-art approaches.

SelfGait is another self-supervised approach, proposed by Liu et al. [94], where authors propose using a self-supervised deep learning model to tackle the problem of gait ecogniti9on by using a large amount of unannotated gait data. To capture the multi-scale spatiotemporal representation of gait, the authors employ the HPM module [50], and MTB module [83], specially designed for the gait recognition information extraction task, as backbone models, to learn spatio-temporal representations. As input, the silhouettes of the individuals are used, and as a backbone basis, the CNN network is used. This approach achieved great results both on CASIA-B and the OU-MVLP datasets.

5. Chapter

GAIT RECOGNITION DATASETS

In this dissertation, the experiments were run on three popularly utilized gait recognition databases: CASIA-B [95], OU-MVLP [96] and GREW [91]. CASIA-B provides a more compact dataset that is frequently used, while OU-MVLP offers one of the most comprehensive gait datasets available presently. This enables the evaluation of how well the proposed approach performs on databases of different sizes, thereby determining whether the quantity of data is a crucial factor for effectively training a DINO feature extractor. Furthermore, to further evaluate the effect of the data quantity for training the feature extractor, the GREW dataset is also employed. GREW dataset consists of data that is acquired in the wild, thus enabling insight into how the proposed gait recognition approach behaves when presented with such data. Specifically, the GREW dataset will be used for pretraining the feature extractor before training on the earlier-mentioned target datasets CASIA-B and OU-MVLP.

5.1. CASIA-B

The CASIA-B dataset [95] is a widely recognized gait dataset in the research domain. It incorporates data from 124 individuals under three distinctive walking scenarios and offers 11 different view angles, ranging from 0 to 180 degrees at an interval of 18 degrees. The example of the images from the CASIA-B dataset is depicted in Figure 5.1. Each individual's walking patterns are captured in three conditions: normal walking (NM), carrying a bag (BG), and wearing a coat or jacket (CL). Six sequences per subject are available for normal walking, while the other two conditions each have two sequences per subject, providing 110 sequences for each subject in total. Given that the Gait Energy Images (GEI) are used in this research, this approximates nearly 13,600 images, with an average of 110 images per subject.

The data was divided into three different sets for testing and training, following a convention often used in related academic research. In the small-sample setting (ST), denoted as CASIAB-ST, the first 24 subjects' data were used for training and the remaining 100 subjects' data for testing. In the medium-sample setting (MT), denoted as CASIAB-MT, the first 62 subjects' data were used for training, and the rest (62 subjects) for testing. Lastly, in the LT setting, denoted as CASIAB-LT, data from the first 74 subjects were used for training, and the remaining 50 subjects for testing.

In each of these divisions, the initial 4 sequences of the NM condition serve as the gallery, and the remaining 6 sequences of the NM condition, along with the 2 sequences each from the BG and CL conditions, are used in the query set.

The dataset was collected in an indoor environment using a multi-camera setup, ensuring consistent lighting and background conditions. The subjects in the dataset have varying ages, heights, and weights, providing a diverse sample to facilitate robust gait recognition algorithms.

By incorporating different covariates (normal, bag, coat), the dataset enables a comprehensive evaluation of gait recognition approaches and their robustness to said covariates. Furthermore, as each subject is recorded from multiple angles, the dataset enables the study of influence how to angle at which the subject is recorded influences the recognition accuracy.

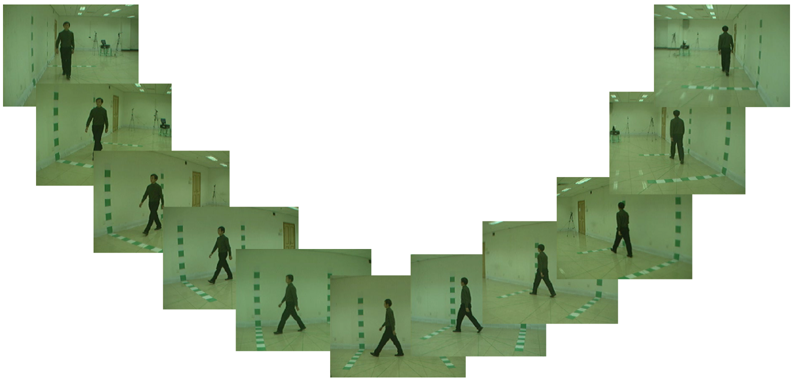


Figure 5.1: Example of the raw RGB images from the CASIA-B dataset [95]

5.2. OU-MVLP

The OU-MVLP dataset [96] stands out as one of the most comprehensive publicly accessible gait databases currently in existence. It provides data from a total of 10,307 individuals, with each subject's gait captured from 14 distinctive viewpoints ranging from 0 to 90 degrees and from 180 to 270 degrees, incremented by 15 degrees. The different view angles of one individual, in the form of GEI images, are depicted in Figure 5.2.

For each of these viewpoints, two sequences are provided (00-01). The training set consists of data from 5,153 subjects, while the remaining 5,154 subjects' data is used for the testing set. When it comes to the testing dataset, sequences marked with the index #01 are utilized as the gallery set, and sequences indexed as #00 serve as the query set.

In terms of image count, the dataset boasts more than 267,000 Gait Energy Images (GEI). This translates to an average of approximately 26 GEI images for each subject. The vastness of this dataset not only provides researchers with a wealth of data to train their models but also allows for rigorous testing to ensure the robustness and generalizability of the developed algorithms. Such a large and diverse dataset is instrumental in understanding the effectiveness of gait recognition systems in real-world scenarios.

By using large volumes of data present in this dataset, it is possible to evaluate the performance of gait recognition approaches on a large number of individuals. Also, since the dataset includes multiple angles at which the individuals are recorded, combined with the large quantity of data, the dataset enables the evaluation of deep learning models with the focus of maximizing the recognition accuracy over different angles.

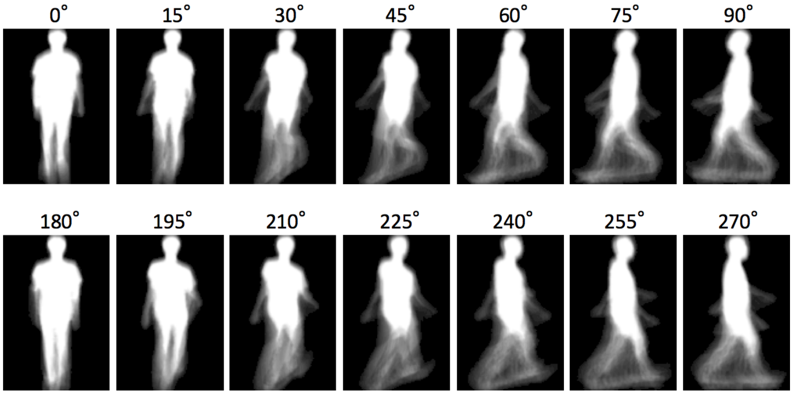


Figure 5.2: Example of the GEI images from the OU-MVLP dataset [96]

5.3. GREW

The GREW dataset [91] is one of the largest available gait recognition datasets available today. The dataset is acquired by collecting a massive amount of video streams where the different individuals walk, by using hundreds of cameras and thousands of hours in open camera systems [91]. The typical video stream in this dataset is a natural video, recorded at various angles, heights of sensors, in different day times, and geographical locations. Examples of the images acquired in this dataset are shown in Figure 5.3.

GREW dataset consists of around 26.000 individuals, with 128.000 walking sequences, that are meticulously annotated. Compared to other datasets, such as CASIA-B and OU-MVLP, GREW stands out in the aspect of annotations. The dataset provides annotations in the form of silhouettes, optical flow information, 2D and 3D poses of individuals across frames, and Gait Energy Images.

The main aim of this dataset is to provide the means for evaluation of gait recognition approaches in the task of unconstrained gait recognition. However, in this dissertation, the GREW dataset is used as a pretraining dataset, to provide the self-supervised feature extraction model enough data to learn usable gait representations. Consisting of many variations of environmental factors present are the recording, as well as the large amount of data provided, it presents itself as a valid choice for pretraining the proposed self-supervised model since the self-supervised models often require an abundance of data in order to perform adequately.

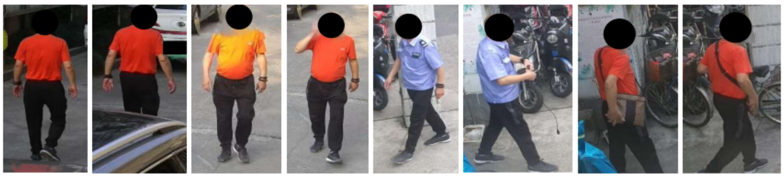


Figure 5.3: Example of the raw images from the GREW dataset [91]

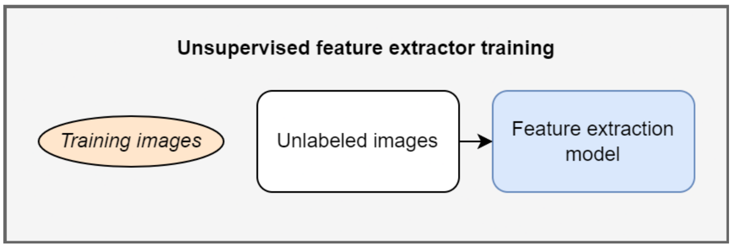
6. Chapter

GAIT RECOGNITION USING A SELF-SUPERVISED SELF-ATTENTION DEEP LEARNING MODEL

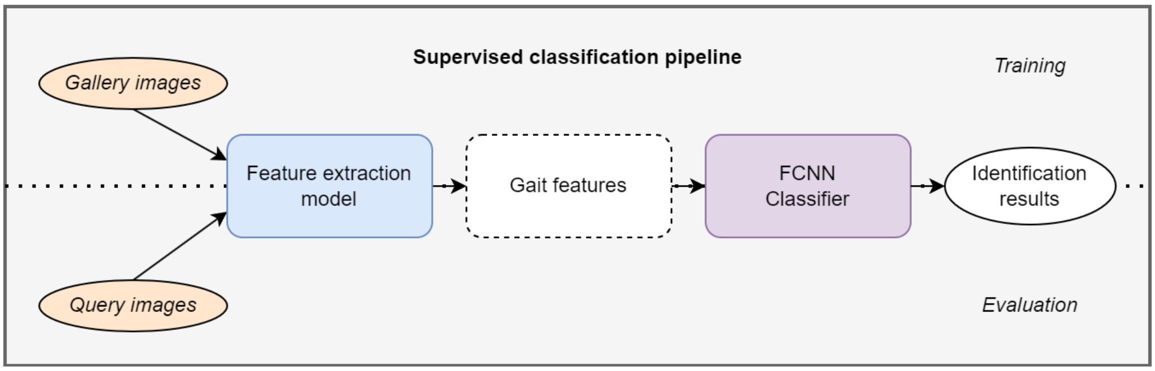
In this chapter, the proposed methodology is elaborated, providing an in-depth look at its main elements. The overall operational sequence is illustrated in Figure 6.1. The initial part of the approach employs the DINO self-supervised model [14] to derive gait characteristics from unlabelled training data, as demonstrated in Figure 6.1 a). Following this, a straightforward Fully Connected Neural Network (FCNN) acts as a classifier for the features yielded by the DINO feature extraction model. This process is trained on gallery samples and evaluated on query samples, as presented in Figure 6.1 b). Annotated samples are exclusively required for training the FCNN classifier since this classifier employs a supervised learning method.

6.1. Data Preprocessing

The initial step in the proposed approach involves the preparation of data. Ordinarily, the input data consists of raw RGB image sequences sourced from a camera and standard gait data preprocessing steps are utilized [71, 89]. The first step involves filtering out noise from the images. Secondly, the silhouettes of each subject are extracted in a binary form by employing methods such as the background subtraction method. The third step



(a) Training feature extractor



(b) Classification pipeline

Figure 6.1: Gait recognition pipeline

involves normalizing the images to ensure that all the silhouettes are of the same height and are horizontally aligned. Following this, the gait cycle is estimated to construct a final representation of the gait. In this dissertation, the image-based gait features are used in the form of Gait Energy Images (GEI) [42], depicted in Figure 6.2. GEI is effective in retaining both the static information of a gait sequence such as the subject's body shape, and dynamic information such as changes in frequency and phase during the subject's movement.

In this dissertation, gait recognition is performed using the data from the RGB camera sensors. RGB Camera sensor produces data by capturing light that reflects from the individuals that are walking in front of the camera, resulting in a video stream of an individual that is walking. During the capturing process, unwanted noise is often acquired together with meaningful data. In order to remove that noise and extract features that are relevant for gait recognition, standard gait data preprocessing techniques are applied.

First, silhouettes of individuals need to be extracted from the video stream. Since the goal of gait recognition is to identify the individual solely based on the individual's movement, only the silhouette of an individual must be preserved, while all other data is removed. Silhouette extraction can be performed in a variety of ways. Traditional techniques rely on the process of background subtraction, where the static parts of an image sequence are disregarded, while only the moving parts are isolated [71]. However, in real-world applications, the background subtraction technique is often lacking, compared to more modern approaches, such as person detection and segmentation [97]. Furthermore, other approaches such as Gaussian mixture models [98, 99], kernel density estimation [100], and others can be typically found in gait recognition approaches.

Second, after the silhouette extraction from the raw RGB video stream, the silhouettes are normalized with respect to their size. In a typical gait recognition scenario, the distance between the individual and the camera sensor that is acquiring the data varies. Depending on the individual's movement with respect to the sensor, the acquired silhouettes often vary in size across the video stream, which results in the need for normalizing the silhouettes. The silhouette normalization is performed by scaling all the silhouettes into uniform size, as in [71].

Third, after all the silhouettes are normalized, the gait cycle determination is performed. Since the gait is in nature a repetitive movement, in this step the information about each individual's gait cycle is determined from each sequence of silhouettes. The goal of this step is to remove redundant data, as well as segmenting the data in order to build a more robust feature extraction model. In a typical video sequence of an individual walking, there is often more than one gait cycle. As described in [101], the gait cycle can be measured from any subsequent event of the same foot. It is important to note that the specific starting point is not important, however, the same starting point should be used for every gait cycle. Frequently, gait cycle estimation is conducted through an analysis of the periodic time series related to the width and length of a silhouette's bounding box [102, 103, 104].

Fourth, the gait feature representation is generated for each of the gait cycles determined in the previous step. The goal of this step is to build a comprehensive feature representation of an individual's gait, comprising both the temporal and spatial characteristics of the gait. The GEI image is constructed by averaging the binary silhouette frames during one complete gait cycle:

$$G(i,j) = \frac{1}{N} \sum_{t=1}^{N} I(i,j,t), \tag{6.1}$$

where N represents the number of silhouette frames in the gait cycle, t represents the frame number in a gait cycle at a moment in time, and $I(i, j)$ is the original silhouette image with (i, j) values in the 2D image coordinate.

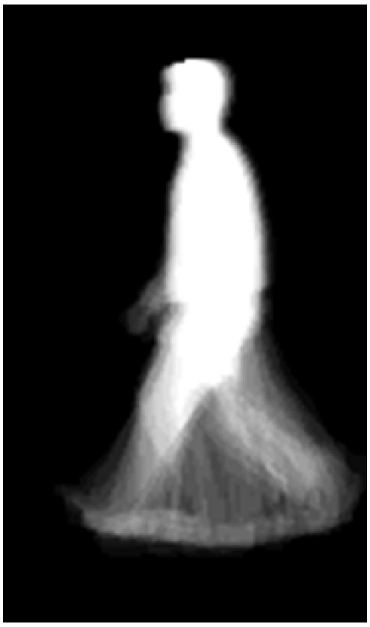


Figure 6.2: Example of the generated Gait Energy Image (GEI)

By generating the GEI images, the data preprocessing steps are complete, and the GEI images represent the final gait representation before training the feature extraction model.

6.2. Training feature extraction model using the self-supervised learning model

The second phase of the proposed gait recognition approach involves training the feature extraction component. In this dissertation, a self-supervised learning model aimed at addressing the challenge of learning distinctive gait features is introduced. The recently developed method known as DINO [14], which has demonstrated encouraging results in numerous computer vision tasks, including image classification, copy detection, and image retrieval, is employed to this end. Figure 6.3 presents a visual representation of the DINO architecture.

The DINO is a self-supervised learning method, where the self-distillation is performed, without the use of any labels. Self-distillation is a concept in machine learning where a

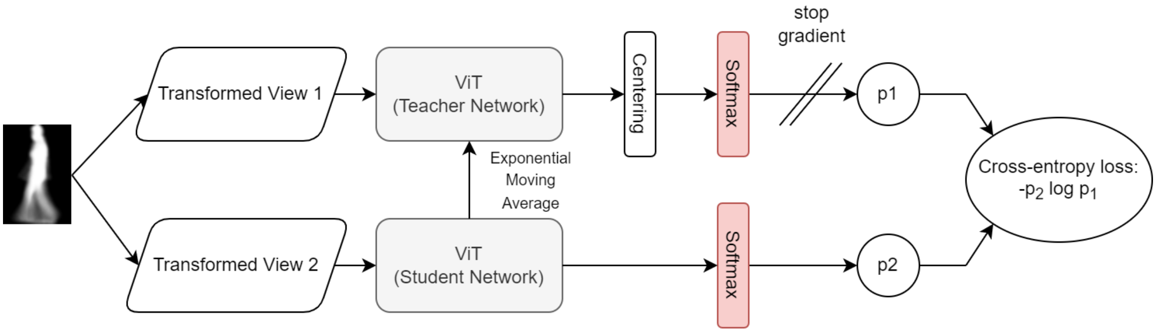


Figure 6.3: DINO self-supervised learning [14]. The goal of a student network is to match the probability distribution of a teacher network using cross-entropy loss, given different views of the same input image.

trained model, often referred to as a teacher, is used to generate soft labels or targets for a student model. It's a form of knowledge distillation, which is the process of transferring knowledge from one machine-learning model (the teacher) to another (the student). In typical knowledge distillation scenarios, the teacher is usually a larger, more complex model, and the student is smaller. The goal is to make the student model perform as well as or close to the teacher model, with the benefit of having a smaller, more efficient model that's easier to deploy in resource-constrained environments.

However, in self-distillation, the teacher and the student are the same networks, with the same architecture and size, but in different training stages. In other words, the teacher model is the model at a certain point during training, and the student model is the model at a later point in training. The student model is trained to mimic the output of the teacher model (from the earlier training stage) on the same data. This process is repeated over several training stages to gradually improve the model's performance. Self-distillation can help the model to generalize better, reduce overfitting, and potentially improve its performance on unseen data.

Instead of using the real labels from the input data, the *soft* labels are used. By generating *soft* labels, the teacher network provides the student network with rich information needed in order to enable the network to learn useful features. The crucial parameter of the *soft* labels is its *sharpness*, which determines how evenly the probability distribution is distributed over classes and is controlled by parameter τ. Higher values of the temperature parameter τ result in "softer" labels, meaning the probabilities are spread more evenly across different classes. This can offer the student model more diverse and detailed information but can also make the learning problem more complex. On the other hand, when τ is lower, it results in "sharper" labels. Here, the probabilities are more focused on a limited number of classes, simplifying the learning task for the student model. However, this also limits the breadth of information passed from the teacher model to the student, since the confidence is high for fewer classes.

In DINO, the framework consists of two networks, teacher network Θ_t and student network Θ_s. The network represents a chosen deep learning architecture, such as ResNet-50 [64], or a ViT [12]. As mentioned earlier, the two networks have the same architecture, but they have different network parameters, θ_t i θ_s, respectively. Throughout the training, the aim of the student network is to replicate the probability distribution of the teacher network. The probability output P is calculated as [14]:

$$P(x)^{(i)} = \frac{exp(g_\Theta(x)^{(i)}/\tau)}{\sum_{k=1}^{K} exp(g_\Theta(x)^{(k)}/\tau)}, \tag{6.2}$$

where x is the input data, i is the target class, g represents the network, K represents the dimension of the network output feature, and $\tau > 0$ is the temperature parameter that controls the sharpness of the output distribution [14]. In this equation, Θ represents the network parameters, and the value can be of teacher of student network, the same is for the τ value.

The training is performed by utilizing the multi-crop technique [58]. In multi-crop training, multiple crops (regions or sections) are taken from an image during training to help a model generalize better by seeing different parts and perspectives of the image. The process usually involves taking multiple random crops from the original image, each potentially with different augmentations (such as rotations, translations, shearing, scaling, flipping, and color alterations), and feeding these to the model during training. The idea is that this kind of training exposes the model to more variability, helping it learn to recognize important features regardless of their position, scale, orientation, or other visual properties. This way, the model can potentially learn more robust and generalized representations.

In DINO, for every input image i.e. GEI image, two random global views are generated, along with several local views of the same image. The global views incorporate the global information of an image, and their size is more than 50% of the whole input image. The local views represent the local information of an image, and they are of a size less than 50% of the input image. As the name suggests, the teacher network is teaching the student network to mimic its probability distribution, and as such the input to the teacher network are the global views of the input image. By accepting only global views, the teacher network gains a more global understanding of the image. The student network also takes global views of an input image as input, however, it also takes all the local views of the input image. By taking both the global and local views of an image, besides the global information, the student network also focuses its attention on the local parts of an image.

The similarity between the output vectors of the teacher and student network is measured by the cross-entropy loss [14]:

$$\min_{\theta_s} H(P_t(x), P_s(x)), \tag{6.3}$$

where $H(a, b) = -a \log b$ [14]. When calculating the loss, the teacher network weights g_{Θ_t} are fixed, and the minimization is done w.r.t the parameters of the student network g_{Θ_s} [14].

The student model parameters θ_s are optimized by reducing the cross-entropy loss using a stochastic gradient descent optimizer. On the other hand, the teacher model parameters, θ_t, are established as a weighted moving average of the student model parameters. This approach allows the model to progressively extract meaningful characteristics from input images, teaching it to understand and correlate global and local information from various augmentations of the same image.

Finally, the loss that is optimized in DINO is given by [14]:

$$\min_{\theta_s} \sum_{x \in \{x_1^g, x_2^g\}} \sum_{\substack{x' \in V \\ x' \neq x}} H(P_t(x), P_s(x')), \tag{6.4}$$

where V represents a set of different views of the input image, containing two global views, x_1^g and x_2^g, and several local views of the input image [14].

One of the main advantages of the DINO self-supervised approach is that the method does not require negative samples, compared to several other popular self-supervised methods [59, 60]. Positive samples represent the image pair in which both images are belonging to the same class and are augmented in some way. In contrast, the negative pair represents two images that do not share the same class, and do not have the same characteristics of that class. The goal of SimCLR [59] and MoCo [60] methods is to bring together similar images, and push further apart those who are dissimilar. However, when generating the negative pairs various problems occur, such as the problem of computational efficiency, inadequate negative pair representativeness, negative pairs that are in fact false negatives, etc. In DINO, such problems are alleviated, since the method does not require negative pairs, since all data augmentations are performed from the same input image.

Furthermore, the DINO technique demonstrates the capability to delineate the main objects in an image, effectively identifying object borders, through self-supervision. In natural image databases, such as ImageNet, distinguishing foreground objects can be quite challenging due to the variety of possible appearances of both the objects and the background. When it comes to gait recognition, where the images are typically presented in formats like GEI, the main subject (the foreground object) is prominently defined against the background. This could result in the model concentrating on the most crucial elements of an image, such as dynamic features represented as pixel values within the range of 0 to 255.

In this dissertation, the DINO method is proposed as a feature extractor to produce discriminative gait features of input images to be used later for classification.

6.3. Feature extraction self-attention model

In DINO, any architecture can be used as a teacher and student network, such as the common ResNet-50 convolutional neural network, and similar. However, the ViT architecture exhibited extremely good performance with DINO [14]. Compared to the similar CNN-based deep neural network, the ViT architecture generally achieved better accuracy in image classification benchmarks, such as the ImageNet challenge. As such, the ViT is used as the main backbone model in DINO, for learning the discriminative gait representations from the GEI images.

ViT is a neural network architecture that extends the Transformer model, originally designed for natural language processing tasks, to the domain of computer vision. ViT has demonstrated competitive performance compared to traditional CNNs in various vision

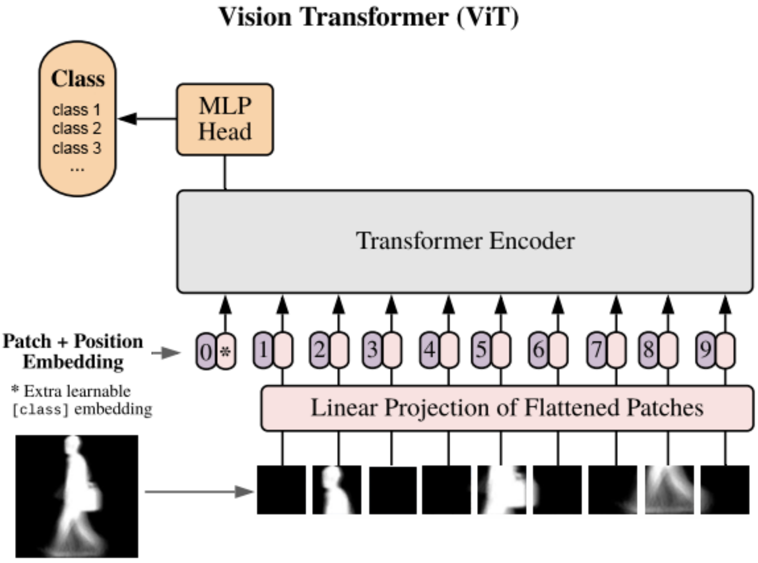

Vision Transformer (ViT)
Class
class 1
class 2
class 3
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 *
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Figure 6.4: The architecture of the ViT model [12]

tasks, such as image classification and object detection.

Compared to CNNs, the ViTs do not rely on convolutions to extract useful features from the data. Instead, the attention mechanism is employed for feature extraction. Instead of sliding multiple convolutional kernels over the input image, in ViT architecture the input image is divided into a series of patches, on which the attention mechanism performs feature extraction.

The first step in the ViT architecture is to divide the input image into non-overlapping patches of fixed size, typically 16×16 or 32×32 pixels. For an image I, with a patch size of $p \times p$ pixels [12]:

$$I \in \mathbb{R}^{H \times W \times C}, \tag{6.5}$$

where H represents the height of an image, W represents its width and C is the number of channels in an image, the resulting image patches are [12]:

$$I \in \mathbb{R}^{N \times p^2 C}, \tag{6.6}$$

where $N = \dfrac{HW}{p^2}$ is the number of patches and p is the patch resolution.

Each patch is then flattened into a 1D vector, and linearly embedded into a continuous latent space using a learnable linear projection matrix. The embedding is defined as [12]:

$$E_i = W_e P_i + b_e, \tag{6.7}$$

where E_i is the embedded patch vector, W_e is the embedding weight matrix, P_i is the flattened patch vector, and b_e is the embedding bias vector. The resulting embedded patches are treated as a sequence of tokens, analogous to word tokens in the original Transformer architecture. The dimensionality of each patch embedding is denoted as D.

To incorporate the positional information of image patches, a positional encoding is added to the embedded patch vectors. The positional encoding is a learnable parameter matrix P_{enc} with the same dimension as the patch embedding D [12]:

$$E_i' = E_i + P_{enc_i}, \tag{6.8}$$

where E_i' is the patch embedding with positional encoding, and P_{enc_i} is the i-th row of the positional encoding matrix corresponding to the i-th patch.

The core of the ViT architecture consists of multiple layers of the Transformer model, which include multi-head self-attention (MHSA) and feed-forward neural networks (FFN).

The self-attention mechanism computes the weighted sum of input vectors based on their compatibility scores. Self-attention allows the model to capture long-range dependencies between the different patches of an input image, which is crucial for accurate classification.

In self-attention, the model learns a weight matrix that determines how much attention should be paid to each patch when computing the output representation. This weight matrix is computed by taking the dot product of a query matrix Q and a key matrix K, both of which are derived from the input features, and then applying a softmax function to obtain a probability distribution over the patches. Finally, the output representation is obtained by taking a weighted sum of the value matrix V, which is also derived from the input features, using the weight matrix.

The input features are represented by a matrix X, which is split into query Q, key K, and value V matrices:

$$Q = XW_Q, \tag{6.9}$$

$$K = XW_K, \tag{6.10}$$

$$V = XW_V, \tag{6.11}$$

where W_Q, W_K, and W_V are the learnable weight matrices for the query, key, and value projections, respectively.

The weight matrix is computed by taking the dot product of Q and K, divided by the square root of the dimension of the key matrix:

$$W = softmax\left(\frac{QK^T}{\sqrt{d}}\right). \tag{6.12}$$

The output representation is obtained by taking a weighted sum of the value matrix V, using the weight matrix W:

$$Y = WV. \tag{6.13}$$

In the case of ViT, the input features are the patches of an image, and the self-attention mechanism is applied to each patch to capture long-range dependencies between them.

In Multi-Head Self-Attention (MHSA), this operation is performed in parallel using multiple attention heads. The self-attention for a single head can be computed as [105]:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V, \tag{6.14}$$

where Q, K, and V are the query, key, and value matrices, respectively, computed from the input matrix X as in Equation 6.9.

The Feed-Forward Neural Network (FFN) then applies a non-linear transformation to the output of the self-attention mechanism. The output of the Transformer block is a set of feature vectors, which capture different aspects of the input data.

The FFN in the Transformer layer is a two-layer perceptron applied independently to each input vector. The output of the FFN can be calculated as:

$$FFN(x) = ReLU(x \cdot W_1 + b_1) \cdot W_2 + b_2, \tag{6.15}$$

where x is the input vector, W_1 and W_2 are the weight matrices for the first and second layers, and b_1 and b_2 are the corresponding bias vectors.

Each Transformer layer incorporates layer normalization and residual connections to stabilize the training process and facilitate gradient flow. The layer normalization is applied as:

$$LN(x) = \frac{x - mean(x)}{\sqrt{var(x) + \epsilon}}, \tag{6.16}$$

where $mean(x)$ and $var(x)$ are the mean and variance of the input vector x, and ϵ is a small constant added for numerical stability.

After passing through the Transformer layers, the final patch embeddings are processed by an output layer to produce the predictions for the task at hand. In the case of image classification, the first patch embedding (corresponding to the "class" token) is typically used as the representation of the entire image. The classification head consists of a linear layer followed by a softmax function to produce class probabilities:

$$logits = E'_{class} \cdot W_{cls} + b_{cls}, \tag{6.17}$$

where E'_{class} is the first patch embedding after passing through the Transformer layers, W_{cls} is the weight matrix for the classification layer, and b_{cls} is the corresponding bias vector. The logits are then passed through the softmax function to obtain the class probabilities:

$$p_i = Softmax(logits)_i = \frac{\exp(logits_i)}{\sum_{j=1}^{C} \exp(logits_j)}, \tag{6.18}$$

where p_i is the predicted probability of class i, and C is the number of classes.

To train the Vision Transformer, a loss function is used to measure the discrepancy between the predicted class probabilities and the ground truth labels. For classification tasks, the cross-entropy loss is commonly used, as in Equation 3.16.

The Vision Transformer is trained using gradient-based optimization techniques, such as SGD or Adam, by minimizing the loss function. The gradients are computed using the backpropagation algorithm, which calculates the gradient of the loss function with respect to each weight and bias by applying the chain rule for differentiation.

The architecture of ViT allows it to learn long-range dependencies between the patches in the input image, without the need for convolutional layers. ViT has achieved state-of-the-art performance on several image classification benchmarks, such as the ImageNet dataset.

Different variants of the ViT model exist. The difference comes in the form of the model size. In the [12], three main model sizes were defined, ViT-Base, ViT-Large, and ViT-Huge. The ViT-Base contains 12 layers, the patch embedding dimension D is 768, and the number of parameters of the model is 86 million. ViT-Large, in contrast, has 24 layers, D is 1024, but the total number of model parameters is 307 million. The more parameters the model has, the training is more complex and expensive, especially in terms of time and hardware requirements.

To alleviate the problem of training large models for a long duration of time, the smaller ViT model is used in this dissertation. The ViT-Small model, in the rest of this doctoral dissertation denoted as ViT, is proposed in [105]. The ViT model contains 12 layers, the same as ViT-Base, however, the D is of size 384, and the total number of parameters is 22 million [105]. The ViT model, in contrast to the ViT-Base model, due to its architecture, allows triple the throughput [105], defined as a number of images processed in one second, thus enabling significantly faster training time, especially on one GPU card. Also, the ViT-Small model is size comparable to one of the most widely used deep learning architectures, ResNet-50, that is used in gait recognition, when comparing the model parameter number.

Furthermore, to study the influence of the patch size on the model performance, the ViT model is trained using two different patch sizes, 16×16, denoted as ViT-16, and 8×8, denoted as ViT-8. The aforementioned patch sizes were chosen due to the availability of the DINO feature extraction models pretrained on the large ImageNet dataset, to alleviate the need for training models directly on the ImageNet dataset on a single GPU workstation.

6.4. Gait features classification

Once the DINO feature extraction model has been trained, it can be used to obtain discriminative gait features. Gait features are extracted from the gallery data set, and from the query data set, and are subsequently used for the classification.

To classify the gait features, a straightforward feed-forward Fully Connected Neural

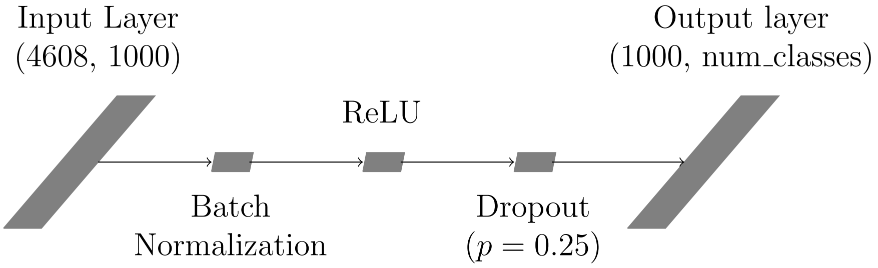


Figure 6.5: Proposed FCNN classifier

Network classifier, denoted as FCNN, is proposed. Therefore, the gait recognition task is defined as a gait classification task, with the gallery serving as the training data and the query serving as the test data for the classifier. For instance, if a gallery comprises 100 subjects, the task is treated as a 100-class classification problem.

The proposed FCNN, illustrated in Figure 6.5, is comprised of two linear layers complemented by batch normalization, ReLU activation function, and dropout layer. The hyperparameters for the FCNN are selected based on empirical observation. Furthermore, the center loss [106] method is employed to encourage the learning of a more diverse feature representation. The main loss used is the cross-entropy loss, and the combination with center loss is given by the equation:

$$L = L_{ce} + \alpha L_c, \tag{6.19}$$

where L represents the final loss value, L_{ce} and L_c are values of cross entropy loss and center loss functions respectively, and α is a scalar that balances the influence of the center loss on the overall loss value, and is set to $\alpha = 0.0001$.

In the training of the feature extractor, image normalization was carried out based on the normalization values specific to the custom dataset. Random erasing was applied as a method for data augmentation. To further enhance the learning of representations, the CLS tokens from all twelve blocks of the DINO feature extraction model were concatenated. This resulted in a final input image representation which was then input into the Fully Connected Neural Network (FCNN) classifier. Given that the dimensionality of the CLS token for ViT model is 384, the input dimensionality for the FCNN classifier subsequently becomes 4608.

6.5. Performance metrics

Deep learning model performance metrics are quantitative measures used to evaluate the effectiveness and accuracy of deep learning models in various tasks, such as classification, regression, and segmentation. These metrics help researchers and practitioners compare different models, understand their strengths and weaknesses, and guide the model selection process. In a scientific context, several common performance metrics are used, each with its underlying mathematical formulations.

In classification tasks, the performance of deep learning models is typically evaluated using a set of metrics that quantify their ability to correctly identify the class labels of instances. The commonly used metrics for classification tasks are accuracy, precision, recall, F1-score, and a confusion matrix.

Accuracy measures the proportion of correctly classified instances out of the total instances. For binary classification, it is defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{6.20}$$

where TP (True Positives) is the number of correctly classified positive instances, TN (True Negatives) is the number of correctly classified negative instances, FP (False Positives) is the number of negative instances incorrectly classified as positive, and FN (False Negatives) is the number of positive instances incorrectly classified as negative.

Precision, also known as a positive predictive value, measures the proportion of true positive instances among those predicted as positive by the model. It is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{6.21}$$

Recall, also known as sensitivity or true positive rate, measures the proportion of true positive instances among the actual positive instances. It is defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{6.22}$$

F1-score is the harmonic mean of precision and recall, providing a balance between these two metrics. It is particularly useful when dealing with imbalanced datasets. It is defined as:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{6.23}$$

The confusion matrix is a table that illustrates the distribution of predictions made by the model across the different classes. It consists of four elements for binary classification tasks: TP, TN, FP, and FN. In multi-class classification, the matrix is expanded to show the distribution of predictions for each class against the ground truth.

Rank accuracy is another performance metric used in classification tasks when the models produce a ranking of class labels instead of directly predicting a single class label for each instance. This metric is particularly useful in scenarios where the system's goal is to provide a list of top-k (k being a positive integer) most likely class labels, and the correct label's position in this list matters.

Rank accuracy is defined as the percentage of instances for which the correct class label is within the top-k predicted class labels.

Mathematically, it can be represented as:

$$RankAccuracy = \frac{N_{correct}}{N_{total}} \cdot 100\%, \tag{6.24}$$

where $N_{correct}$ represents the number of instances with correct labels in top-k predictions, and N_{total} represents the total number of instances.

To compute rank accuracy, several steps need to be performed. First, for each instance, the predicted class probabilities need to be obtained from the model. Second, the class labels need to be ranked based on their probabilities in descending order. Third, the check is performed if the true class label is within the top-k predicted labels. Fourth, the percentage of instances where the true label was found in the top-k predictions needs to be calculated.

Rank accuracy is valuable in applications where the system provides the user with a list of possible class labels instead of a single prediction, such as search engines, recommendation systems, or image retrieval systems. It helps evaluate the model's effectiveness in presenting the correct labels within the top-k ranked predictions, providing insights into its performance from a ranking perspective.

7. Chapter

EXPERIMENTAL SETUP

In order to evaluate the proposed method, a series of tests were executed to gauge the effectiveness of the suggested DINO feature extraction model and the FCNN classifier trained using the features obtained from this model. The experiments were designed in such a way that comparisons could be easily made with current state-of-the-art models utilized for gait recognition. This was achieved by adhering to identical dataset divisions and comparison metrics as used in the current standard approaches. Following this, a description of experimental setup is provided before presenting and analyzing the results.

7.1. Datasets

In this dissertation, experiments were conducted on two prevalent gait recognition datasets: CASIA-B and OU-MVLP. CASIA-B represents a smaller yet extensively utilized dataset, whereas OU-MVLP is recognized as one of the most substantial gait datasets currently available. These datasets facilitate an evaluation of the performance of the proposed approach on both smaller and larger datasets, providing insight into whether the volume of data plays a significant role in the successful training of a DINO feature extractor.

Furthermore, an additional dataset was used, GREW, for evaluating the performance of the proposed feature extraction models and proposed classifier, in the scenario where there is a large amount of unlabelled gait data present from the real-world scenarios, captured in the wild. This dataset is used only for feature extraction model pretraining, since the test data for the dataset is not publicly released, resulting in an inability to evaluate the model on the test data.

In alignment with the procedures undertaken in studies [50, 94], all images were standardized from all datasets to dimensions of 64 × 44. This measure was implemented to facilitate comparative analysis, as well as to reduce the computational demands needed to train the DINO model. Additionally, during the training phase of the DINO model, the training data is normalized utilizing the mean and standard deviation derived from the training data used.

7.2. Experiments

To assess the effectiveness of the proposed approach, the gait data is prepared as outlined in Chapter 6.1., by preprocessing the raw silhouettes and generating the GEI images for all three used datasets, CASIA-B, OU-MVLP, and GREW.

Following the data preparation, the DINO feature extraction models were subsequently trained on two of the mentioned datasets, CASIA-B and OU-MVLP. The goal of this models was to learn the discriminative gait features from the provided data.

Particular attention was given to the CASIA-B dataset, which exhibits a unique tripartite data split structure. Two separate models were trained on this dataset. The first was configured with a patch size of 16, while the second model was structured with a patch size of 8. The varied patch sizes were implemented to observe and compare the effects of different resolutions on the model's learning ability and the subsequent extraction of gait features.

Simultaneously, a parallel training procedure was carried out on the OU-MVLP dataset. Again, two models, reflecting the patch sizes of 16 and 8, were trained. The rationale behind using different patch sizes and distinct models on this dataset mirrors the approach taken with the CASIA-B dataset. This consistent methodology across both datasets allows for a balanced comparison and an understanding of the influence of dataset size and diversity on the model's learning efficacy.

Following the training of the DINO feature extraction models, the next step involved training a Fully Connected Neural Network (FCNN) classifier. This classifier was trained using a set of gallery samples, which are a collection of known and annotated gait data.

This process allowed the FCNN classifier to learn and identify different patterns and attributes in gait features.

This FCNN classifier, once trained, culminated in the creation of the final model for gait classification. The purpose of this model was to effectively classify and identify different gaits based on the complex and discriminative features learned during the training process.

In the final step of this approach, the performance of the trained FCNN classifier was thoroughly evaluated. This was carried out by using query samples, which are a set of unannotated and unknown gait data. The classifier's task was to correctly classify these gaits, with its performance being measured by the accuracy of its classifications.

Additionally, the ablation study was performed, and the experiments involved will be described in detail in Chapter 8.4.

7.3. Self-supervised feature extraction model

The official GitHub repository [107] was utilized for the DINO method implementation, incorporating minor modifications. These changes catered to the distinct data distribution of gait data, which differs from the natural images found in the ImageNet dataset. Adjustments included changes to global and local crop sizes and variations in training data augmentations.

Originally, a set of eight local views (96 × 96 crops, processed only through Φ_s) and two global views (224 × 224 crops, processed through both Φ_t and Φ_s) are constructed by DINO. To adapt to gait-specific data in this dissertation, eight local views with local crops of size 20 × 20 are used, while two global crops are of size 64 × 64. The crop sizes were adjusted to accommodate the sizes of the gait training images utilized in this dissertation, all the while preserving similar ratios of global and local crops as found in [14]. Furthermore, given that DINO was initially trained on ImageNet, the majority of image augmentations implemented during training, such as color jitter, Gaussian blur, solarization, and random horizontal flip, were omitted. Only the random erasing augmentation was retained, as the previously mentioned augmentations failed to yield a performance enhancement when employed on gait-specific data.

Since gait datasets typically lack the large amount of data needed to train the ViT model from scratch [12], the fine-tuning strategy is used in this work. Both the student and teacher networks were fine-tuned using the ImageNet pretrained DINO model checkpoint, on the target gait datasets. In this dissertation, the small ViT model is used, whose size is roughly equivalent to a typical Resnet-50 [64] architecture in terms of network parameters. The models using patch sizes of 16 and 8 are trained to investigate how the patch size influences the model's accuracy.

The remaining DINO model parameters such as the momentum teacher value, teacher temperature, and global and local crop scales were kept consistent with those specified in the original manuscript [14].

7.4. Training Details

The DINO feature extraction models were trained for 1000 epochs for all experiments on the CASIA-B, OU-MVLP, and GREW datasets. The optimizer used was AdamW [108] with a learning rate of 0.0005. The training was performed using a Nvidia 2080Ti 11 GB GPU, Nvidia 3070 8GB GPU, and Nvidia 4090 24GB GPU.

The FCNN classifier underwent a training process spanning 100 epochs, utilizing a batch size of 128. The Adam optimizer was employed for the FCNN classifier with a set learning rate of 0.0005. In a similar fashion, the Adam optimizer was also used for the center loss optimization, but with a learning rate of 0.1.

The learning rates for both the DINO models and the FCNN classifier were established through empirical investigation. The learning rates were explored within a range from 0.1 to 0.000001, using the grid search approach. The training epoch count for the DINO model was fixed at 1000, as no enhancement in accuracy was observed when the model was trained for a longer period. Similarly, the epoch count for the FCNN classifier training was set at 100. The batch size for both models was determined by seeking the optimal value within the range of 8 to 128, increasing in steps by powers of 2.

For the kNN classifier used in this experiments, the value of the nearest neighbors parameter was set to 20, as the said number of neighbors achieved the best accuracy in the preliminary experiments.

In the ablation study, Chapter 8.4.2., the self-supervised ViT models were trained as described for the main results. However, when training the ViT model in a supervised manner, the learning rate was set to 0.001, the batch size to 64, and the model was trained for 300 epochs. The ResNet-50 model was trained in a self-supervised manner by setting the learning rate to 0.03, with SGD optimizer, for 300 epochs, while keeping the rest of the parameters the same as in training the ViT model. In supervised training, the learning rate value for the ResNet-50 model was set to 0.0001, with a batch size of 64, and the Adam optimizer was used for training, for the 300 epochs. Regardless of the learning approach, both ViT and ResNet-50 models used the network weights that were pretrained on the ImageNet dataset.

7.5. Evaluation Protocol

For evaluation of the experimental results, the rank-1 accuracy is used, where the percentage of predictions where the top prediction is the correct one is of interest, i.e., where it matches the ground-truth value. To ensure comparability with other state-of-the-art methods, the identical-view cases are excluded from evaluation.

Besides using rank-1 accuracy, the results are also reported for rank-5 accuracy, in order to evaluate how the model performs across the ranks. Also, the precision, recall, and F1-score metrics are employed to gain a deeper understanding of the model's performance.

To evaluate the quality of learned features, the t-SNE method is used in order to visualize the gait features extracted from the trained feature extraction models.

To evaluate the significance of the results of the trained feature extraction models, the McNemar test [109] is used. As common in the application of the McNemar statistical test, the significance level of 0.05 is set for the tests executed in this doctoral dissertation.

8. Chapter

RESULTS

In this part, the results of the performed experiments are presented. Notably, of all the methodologies compared, only SelfGait, proposed by Liu et al. [94], applies a self-supervised learning approach, while all others are leveraging a supervised learning approach.

In the reported results, it is important to note that several view angles were omitted for brevity, and the complete results including all view angles are available in the Appendix of this dissertation.

8.1. CASIA-B Dataset Results

The results under the CASIAB-ST setting are detailed in Table 8.1. When compared with other leading techniques, this methodology demonstrates superior accuracy in both the Normal (NM) and Bag (BG) modalities. However, it's noteworthy that the accuracy in the coat (CL) mode doesn't match the other state-of-the-art approaches and stands as the least accurate.

In the CASIAB-MT setting, as shown in Tables 8.2, the proposed approach continues to outperform others in the Normal (NM) modality. However, the accuracy drops below other approaches for the Bag (BG) modality and significantly lags in the Coat (CL) modality.

Moving to the CASIAB-LT setting, detailed in Tables 8.3, the proposed approach continues to outperform others in the NM modality. The BG modality shows competitive

Table 8.1: Results for CASIA-B dataset ST setting

Model	Modality	Angle								Average
		0°	36°	72°	90°	108°	144°	180°	Mean	
ViT-16	NM	100.00	100.00	98.00	99.00	99.00	99.50	100.00	99.36	62.30
	BG	83.50	68.50	63.00	52.00	54.00	55.78	81.41	65.24	
	CL	25.50	24.00	22.50	23.00	21.00	20.10	23.50	22.29	
ViT-8	NM	100.00	98.50	97.50	99.00	98.50	99.00	100.00	99.05	67.69
	BG	87.50	74.50	72.00	73.50	72.00	78.39	85.43	78.01	
	CL	28.00	26.50	25.00	28.00	26.00	21.61	28.00	26.02	
GaitSet [50]	NM	64.60	90.40	80.20	75.50	80.30	87.10	59.60	79.54	63.03
	BG	55.80	76.90	69.70	63.40	68.00	76.20	52.50	68.64	
	CL	29.40	49.50	42.30	40.30	44.90	43.00	25.60	40.90	
mmGaitSet [90]	NM	78.50	94.00	88.10	84.40	87.40	92.40	73.90	87.63	71.84
	BG	70.40	84.70	77.40	73.00	77.90	82.00	65.40	77.95	
	CL	42.20	58.30	53.00	49.50	51.40	51.20	34.40	49.95	
Huang et al. [88]	NM	67.40	88.80	80.70	74.90	79.20	88.20	66.70	80.29	64.38
	BG	57.80	77.10	70.10	64.30	68.70	75.40	54.60	69.19	
	CL	33.40	53.10	46.10	41.20	47.40	47.10	29.30	43.65	
GaitPart [83]	NM	62.50	87.50	93.80	95.80	93.80	70.80	75.00	84.66	63.76
	BG	52.10	58.30	79.20	81.20	77.10	66.70	52.10	66.86	
	CL	22.90	35.40	39.60	62.50	52.10	33.30	33.30	39.77	

yet slightly lower accuracy compared to the other approaches, while in the CL modality accuracy is not competitive with other approaches.

On the whole, the proposed approach yields great results when applied to the NM modality across all dataset settings of CASIA-B. The BG modality performs optimally under the ST setting, maintaining an average performance in other settings. The CL modality persistently exhibits the lowest accuracy, which could be attributed to the model's primary focus on the NM modality, being the richest in training data and most distinguishable without considering other modalities.

The BG modality, which takes into account subjects carrying a bag, thereby slightly altering their appearance, demonstrates results on par with other state-of-the-art techniques. However, the CL modality, significantly transforming subjects' appearance as it involves subjects wearing a coat, is the most challenging and hence records low accuracy

Table 8.2: Results for CASIA-B dataset MT setting

Model	Modality	Angle								Average
		0°	36°	72°	90°	108°	144°	180°	Mean	
ViT-16	NM	100.00	100.00	99.19	99.19	99.19	99.19	100.00	99.41	67.05
	BG	89.52	74.19	66.94	64.52	66.94	71.77	87.10	76.36	
	CL	25.81	23.39	26.61	26.61	24.19	22.58	29.03	25.37	
ViT-8	NM	98.39	98.39	99.19	98.39	99.19	100.00	99.19	99.19	68.90
	BG	87.90	84.68	73.39	70.16	72.58	75.81	87.10	79.37	
	CL	33.87	26.61	36.29	32.26	24.19	20.16	32.26	28.15	
GaitSet [50]	NM	86.80	98.00	91.50	89.10	91.10	97.40	80.20	92.05	79.61
	BG	79.90	91.20	81.60	76.70	81.00	90.30	73.00	84.26	
	CL	52.00	72.80	63.10	61.20	63.50	67.50	45.90	62.53	
mmGaitSet [90]	NM	94.40	99.30	96.10	94.40	96.30	98.40	92.30	96.68	88.23
	BG	90.50	94.30	91.60	88.90	91.20	94.90	84.80	92.00	
	CL	73.60	82.70	76.40	73.50	74.70	77.00	65.50	76.00	
Huang et al. [88]	NM	86.70	97.80	91.60	87.00	91.40	95.90	82.50	92.25	80.43
	BG	80.10	91.30	84.00	75.80	81.10	90.70	73.70	84.41	
	CL	58.30	76.80	64.50	58.90	64.00	68.80	49.10	64.64	
GaitPart [83]	NM	63.10	84.60	77.00	72.60	77.40	84.00	63.70	76.40	58.33
	BG	47.50	64.20	61.30	56.70	63.40	61.80	47.00	58.96	
	CL	30.20	43.40	43.60	41.90	40.00	41.40	29.90	39.62	

with the proposed approach. This suggests that, practically, the proposed approach may not be the best choice for the CL modality compared to other techniques, warranting further investigation to enhance its accuracy for this specific modality.

As per the results presented, the proposed approach generally performs well across varied categories, excluding the CL modality. It also effectively distinguishes between different angles at which subjects are recorded. The highest accuracy is observed for angles close to 0° and 180°, while the area around the 90° angle tends to record the lowest accuracy.

This can be attributed to the fact that the angles close to the 0° and 180° angles contain the most discriminative information in GEI images, since the motion of an individual is the most noticeable, showing the individuals body more clearly, while the angles further than the aforementioned angles contain a lower amount of information since the motion

Table 8.3: Results for CASIA-B dataset LT setting

Model	Modality	Angle								Average
		0°	36°	72°	90°	108°	144°	180°	Mean	
ViT-16	NM	100.00	99.00	99.00	97.00	99.00	99.00	99.00	99.00	68.93
	BG	88.00	74.00	79.00	78.00	71.00	83.00	88.00	80.97	
	CL	33.00	24.00	24.00	34.00	21.00	23.00	34.00	26.82	
ViT-8	NM	100.00	100.00	98.00	97.00	98.00	100.00	99.00	99.00	68.68
	BG	88.00	78.00	76.00	81.00	75.00	78.00	87.00	79.96	
	CL	27.00	31.00	29.00	35.00	27.00	18.00	22.00	27.09	
GaitSet [50]	NM	90.80	99.40	93.60	91.70	95.00	98.90	85.80	94.96	84.18
	BG	83.80	91.80	83.30	81.00	84.10	92.20	79.00	87.24	
	CL	61.40	80.70	72.10	70.10	71.50	73.50	50.00	70.35	
mmGaitSet [90]	NM	95.60	99.90	95.90	95.40	96.20	98.90	94.40	97.45	90.09
	BG	91.40	94.10	91.40	88.60	90.00	95.70	88.10	92.54	
	CL	77.60	85.80	78.90	76.60	78.50	82.20	72.20	80.27	
Huang et al. [88]	NM	91.10	99.60	94.30	91.90	94.90	98.80	86.60	95.15	85.69
	BG	84.30	93.40	86.10	80.30	84.40	93.70	80.10	87.91	
	CL	64.70	84.10	73.70	72.30	75.00	77.90	57.00	74.02	
GaitPart [83]	NM	94.10	99.30	94.00	92.30	95.90	99.20	90.40	96.23	89.13
	BG	89.10	96.70	88.30	94.90	89.00	96.10	85.80	92.46	
	CL	70.70	86.90	77.10	72.50	76.90	83.80	66.50	78.69	

of certain body parts is occluded by the other parts of the individual's body.

The models with both 16 and 8 as patch sizes displayed analogous performance in the Normal Motion (NM) modality, without any substantial discrepancies in accuracy across all dataset settings. However, the variance in accuracy becomes apparent in the Bag (BG) and Coat (CL) modalities, where the model employing a patch size of 8 demonstrates a significant enhancement in accuracy compared to its counterpart with a patch size of 16, excluding the LT setting where the significant difference between patch size 16 and 8 for ViT model is not found. This result could potentially stem from the smaller patch size model's capacity to concentrate on more granular image segments, thereby constructing a model that is more resilient to the influence of variables such as a bag or a coat.

8.2. OU-MVLP Dataset Results

Tables 8.4 present the accuracy results for the OU-MVLP dataset, where the proposed approach presents results that are on par with other state-of-the-art approaches. The proposed approach exhibits consistent performance across all viewing angles, particularly at the angles of 30° and 210°, even though it tends to falter slightly at the 0° angle.

Table 8.4: Results for OU-MVLP dataset

Model	Angle								Average
	0°	30°	60°	90°	180°	210°	240°	270°	
ViT-16	79.12	88.19	81.36	83.42	83.80	88.68	84.01	83.89	85.02
ViT-8	77.09	86.45	78.41	82.60	79.84	87.24	79.95	83.37	83.28
GEINet [77]	11.40	41.50	39.50	38.90	14.90	43.20	39.40	36.30	35.76
Zhang et al. [110]	56.20	81.40	78.40	76.50	60.20	79.80	76.70	73.90	74.66
Zhang et al. [82]	74.00	94.60	88.00	90.00	76.70	95.00	88.00	89.80	89.02
GaitSet [50]	79.50	89.90	88.10	87.80	81.70	89.00	87.20	86.20	87.14
SelfGait [94]	85.10	92.00	89.10	90.90	87.40	89.30	90.80	87.70	89.87

Contrastingly, the SelfGait [94] method also employs a self-supervised learning strategy but enhances the model's spatiotemporal capabilities with a specialized backbone network, thereby securing state-of-the-art results on this dataset. Approach outlined in this dissertation, however, makes use of an unaltered ViT network, paired with a simple Fully Connected Neural Network (FCNN) as a classifier, yet manages to attain comparable accuracy levels.

Considering the extensive range of images in the OU-MVLP dataset, the feature extraction model successfully learned to identify distinguishing features, thereby achieving results akin to the state-of-the-art. A notable advantage of the proposed approach, when compared to SelfGait, is the utilization of a simple, general-purpose ViT architecture as opposed to the gait-specific network employed in SelfGait.

Moreover, the proposed method does not directly deduce temporal features from the data, unlike SelfGait which leverages Micro-motion Template Builder (MTB) to extract temporal features from silhouettes. Consequently, the proposed method simplifies the learning process as it primarily focuses on learning appearance features.

8.3. Comparison of the ViT model patch sizes and kNN and FCNN classifier on the recognition accuracy

In this chapter, several statistical tests were performed, in order to demonstrate the effectiveness of the proposed approach. First, the effect of the patch size in the ViT model is evaluated, and second, the comparison of two used classification algorithms is performed, in order to statistically confirm the results presented in Chapters 8.1. and 8.2.

8.3.1. Effect of the patch size in ViT model

To evaluate the effectiveness of the patch size in the ViT feature extraction model, the McNemar statistical test is performed on the results of the CASIA-B and OU-MVLP dataset, comparing the ViT model with a patch size of 16, and with the patch size of 8. Both model types were evaluated for each dataset, and for evaluation, the FCNN classifier is used.

Table 8.5: McNemar statistical significance test for different patch sizes for ViT model

Dataset	Feature extraction (ViT-16 vs. ViT-8), p-value
CASIAB-ST	$p < .001$
CASIAB-MT	$p < .001$
CASIAB-LT	$p = .700$
OU-MVLP	$p < .001$

From the obtained results from Table 8.5, it can be concluded that the differences between the ViT Small 16 and ViT Small 8 model are statistically significant, since the p-values are lower than the specified significance level (0.05), with the exception of the CASIAB-LT setting, where there was no significant difference found since the p-value is higher than the same significance level.

Both patch sizes in the ViT model performed with high accuracy across different experiments. When considering the CASIAB-ST and CASIAB-MT settings, the ViT-8 significantly outperformed the ViT-16 model. However, in the CASIAB-LT setting, the significant difference was not found. Furthermore, on the OU-MVLP dataset, the ViT model with a patch size of 16 achieved higher accuracy than the model with a patch size of 8.

In conclusion, in the performed experiments the models with the patch size of 8 performed better when the number of available individuals for feature extraction model training is low. However, when the number of individuals is higher, the ViT model with a patch size of 16 achieved better accuracy.

8.3.2. Effect of the proposed FCNN classifier

In Table 8.6, the detailed results for each setting of CASIA-B and OU-MVLP datasets are presented, outlining the difference between kNN and FCNN classifiers.

Table 8.6: Detailed results for the CASIA-B and OU-MVLP datasets

Dataset	Model	Classification	Average Metrics				
			Rank 1	Rank 5	Precision	Recall	F1-score
CASIAB-ST	ViT-16	kNN	23.89	47.05	31.17	23.93	23.74
		FCNN	62.30	77.37	78.90	62.31	66.04
	ViT-8	kNN	28.50	50.08	35.46	28.56	28.45
		FCNN	67.69	81.31	78.84	67.72	70.31
CASIAB-MT	ViT-16	kNN	31.76	56.02	39.76	31.75	31.71
		FCNN	67.05	81.57	80.91	67.04	70.21
	ViT-8	kNN	26.67	53.42	33.02	26.67	26.72
		FCNN	68.90	84.06	80.50	68.89	71.68
CASIAB-LT	ViT-16	kNN	35.14	62.46	44.37	35.13	34.83
		FCNN	68.93	85.94	81.03	68.92	71.67
	ViT-8	kNN	28.86	57.18	36.16	28.85	28.81
		FCNN	68.68	85.21	81.72	68.67	71.62
OU-MVLP	ViT-16	kNN	6.70	18.71	8.71	6.67	6.42
		FCNN	85.02	94.28	85.06	84.80	83.86
	ViT-8	kNN	8.02	18.77	10.93	7.92	7.75
		FCNN	83.28	93.66	83.55	83.02	82.15

To evaluate the effectiveness of the proposed FCNN classifier versus the ordinary kNN classifier used in the literature, the McNemar statistical test is performed on the results of the CASIA-B and OU-MVLP dataset, comparing the results achieved by the proposed FCNN classifier with the kNN classifier.

The statistical test has shown that, for all the experiments, models, and datasets, the difference between the kNN and proposed FCNN classifier is strongly statistically significant, $p < .001$. In all cases, the FCNN classifier achieved significantly higher rank-1 accuracy, demonstrating its strong classification capability.

8.4. Ablation Experiments

In order to further evaluate the proposed approach with respect to pretraining the feature extraction model and a gait feature classification, an additional ablation experiments were performed. Similarly, additional experiments were performed with the goal of comparing the influence of the type of learning used to learn gait representations.

8.4.1. Comparison of model pretraining

Model pretraining, also commonly known as pretraining, is used as a foundational step in machine learning to enhance the performance of models. It involves training a machine learning model on a large, comprehensive dataset before fine-tuning it on a smaller, task-specific dataset. Pretraining allows the model to learn general patterns from the large dataset, which it can then apply to specific tasks. This transfer of knowledge helps the model to make accurate predictions even when the task-specific dataset is small. Furthermore, training machine learning models, especially deep learning models, can be computationally expensive and time-consuming. By utilizing pretraining, computational resources and time needed to train a model can be cut down significantly, in contrast to training the model from scratch.

In this dissertation, the experiments were performed by pretraining the proposed feature extraction model on different datasets, in order to evaluate the influence of the pretraining dataset on the accuracy of the proposed gait recognition pipeline. First, the models were trained directly on the target gait dataset, without any large dataset pretraining. Features learned by that model are inferred only from the limited availability of data present in the target dataset. Second, the ImageNet dataset was used for pretraining the models. The ImageNet dataset consists of 1.2 million natural images in RGB, of various general categories such as a car, boat, animals, etc. Thanks to the large size and diversity of the ImageNet dataset, the models trained on this dataset are able to learn useful features and generalize well on other datasets, even if they do not have the same image distribution in terms of the data, i.e. images do not have to be a natural image, with characteristics similar to those of the ImageNet dataset. Third, the feature extraction models were pretrained on GREW gait dataset. GREW dataset consists of around 300.000 images, roughly double the size of the OU-MVLP dataset, and represents one of the largest gait datasets available today. Images acquired in this dataset are in the form of GEIs and follow similar data distribution and characteristics such as target gait datasets.

Table 8.7: Results for ablation study of feature extraction model pretraining

Pretraining Dataset	Classification	CASIA-B				OU-MVLP
		NM	BG	CL	Overall	Overall
No pretraining	kNN	27.91	13.93	6.00	15.95	8.44
	FCNN	98.55	60.03	16.91	58.50	79.08
ImageNet	kNN	57.55	35.51	12.36	35.14	6.70
	FCNN	99.00	80.97	26.82	68.93	85.02
GREW	kNN	47.00	20.49	6.45	24.65	19.21
	FCNN	99.09	69.86	18.91	62.62	87.36

After pretraining the feature extraction model on the pretraining dataset, the model was further fine-tuned on the target gait datasets. In this experiments, the CASIAB-LT and OU-MVLP datasets were used. After the fine-tuning process, the results were obtained by testing the learning model on the target gait dataset. The model trained was the ViT Small model with a patch size of 16. Also, the evaluation was performed using the kNN and FCNN classifiers, to evaluate the performance of different classifiers on learned features.

In both datasets, the lowest results were obtained without pretraining, as shown in Table 8.7. Often, modern deep learning architectures required a large amount of data in order to learn useful features, and the amount of data, especially in the CASIAB-LT dataset is fairly small, resulting in decreased accuracy. In contrast, the ImageNet dataset pretraining yielded great results on both datasets. Pretraining on ImageNet enabled the feature extraction model to learn more generalizable features, that are further enhanced by fine-tuning the target gait dataset. Finally, the pretraining on GREW dataset also yielded good results. In both datasets, the pretraining on GREW dataset gave better results than without the pretraining on other datasets. On the OU-MVLP dataset, the results of pretraining on GREW dataset outperformed the results achieved by ImageNet pretraining, yielding more representative features, both in kNN and FCNN evaluations.

Since the standard McNemar test is unsuitable for statistical comparison of the results of the three different feature extraction models, Cochran's Q test [111] is performed for each dataset. The Cochran's Q test determined that there was a statistically significant difference in the proportion of accuracy in three selected feature extraction models with different pretraining datasets. For CASIAB-LT dataset the obtained values of the statistical tests were $\chi^2(78.39) = 9.49 \times 10^{-18}$, $p < .05$, and for the OU-MVLP dataset were $\chi^2(1751.97) = 0.0$, $p < .05$.

Since the statistically significant difference was found, the post hoc test was carried out using multiple McNemar's tests, with manual Bonferroni correction. The McNemar test is conducted for each dataset, for each model combination, only for FCNN classifier since it significantly outperformed kNN classifier. The significance between the results of feature extraction models that were not pretrained, pretrained on ImageNet dataset, and that were pretrained on GREW dataset, was compared.

Results of the post hoc tests obtained are presented in Table 8.8. From the results, it can be inferred that the feature extraction model pretraining on different datasets has a significant impact on the performance of the proposed approach.

Table 8.8: McNemar statistical significance test for feature extraction model pretraining

Dataset	No pretraining vs. ImageNet, p-value	No pretraining vs. GREW, p-value	ImageNet vs. GREW, p-value
CASIAB-LT	$p < .001$	$p < .001$	$p < .001$
OU-MVLP	$p < .001$	$p < .001$	$p < .001$

From the obtained results from the Table 8.8, it can be concluded that the differences between the pretraining of the feature extraction model on different datasets are statistically significant since all p-values are lower than the significance level, and the null hypothesis of the McNemar's statistical test can be rejected.

From the results and the tests of statistical significance, it can be concluded that the feature extraction model training is effective both without pretraining and with pretraining on another dataset. In the case of the CASIAB-LT dataset, the pretraining on the ImageNet dataset had the highest accuracy, however, in the case of the OU-MVLP dataset, the GREW dataset performed best. The difference in performance could be due to the nature of the OU-MVLP dataset, where only the different angles are investigated at which the individuals recorded, thus making the problem easier and more comparable to the GREW dataset data distribution, compared to the CASIAB-LT dataset where the covariates such as a bag of a coat are also investigated.

8.4.2. Comparison of supervised vs self-supervised learning

Different types of learning exist, as mentioned in Chapter 3.1., among which supervised and self-supervised learning emerge as the most popular approaches for training feature extraction models. These methodologies differ fundamentally in their approach to leveraging annotated data and thereby influence the model's learning efficacy and performance. In supervised learning, models are trained with explicitly annotated data, whereas self-supervised learning exploits implicit labels inherent in the data, circumventing the need for extensive annotated datasets.

In this experiments, the effect of the type of learning is evaluated, on the training of the feature extraction models.

The experiments were performed using two different deep learning architectures, the ViT Small model with a patch size of 16, based on the concept of self-attention, and the standard widely used ResNet-50 architecture, based on convolutions. The ResNet-50 architecture is chosen since it is one of the most commonly used deep learning architectures, and since it is similar in size compared to the ViT Small model, with 23 million and 21 million model parameters, respectively. Both models were pretrained on the ImageNet dataset, in order to boost their performance. For evaluation, both the kNN and FCNN classifiers were used, and the experiments were conducted on the CASIAB-LT dataset.

In Table 8.9, the results of the experiments are presented. For the ViT Small 16 model, the self-supervised learning achieved better results than the supervised approach,

Table 8.9: Results for ablation study of supervised vs. self-supervised learning approaches

Model	Learning type	Classification	Averages			
			NM	BG	CL	Overall
ViT-16	SL	kNN	49.00	31.42	17.73	32.72
		FCNN	95.82	67.49	29.09	64.13
	SSL	kNN	57.55	35.51	12.36	35.14
		FCNN	99.00	80.97	26.82	68.93
ResNet-50	SL	kNN	49.00	26.31	6.36	27.22
		FCNN	99.09	75.31	20.09	64.83
	SSL	kNN	52.82	26.40	9.64	29.62
		FCNN	98.18	67.75	19.27	61.74

both in kNN and FCNN evaluations. This can be attributed to the fact that the ViT-16 model, pretrained on the ImageNet dataset, and trained with self-supervision, has great representation ability, as demonstrated in the [14], in some cases outperforming its supervised counterparts, thus performing better in the demonstrated experiment.

For the ResNet-50 model, in the kNN evaluation, self-supervised learning achieved better results, however, considering the FCNN evaluation, the supervised learning performed better. The aforementioned could be attributed to the fact that the DINO feature extraction model, as demonstrated in [14], achieved great feature representation ability when trained specifically on ViT models, however, when the ResNet-50 model is used as a backbone model, the results were lower, and the representation ability was decreased.

The McNemar test is performed in order to evaluate the statistical significance of the supervised versus self-supervised learning approaches. For each model type, and for each classifier, the McNemar test is conducted, comparing the significance between the results of supervised and self-supervised learning models.

From the obtained results from the Table 8.10, it can be concluded that the differences between the SL and SSL are statistically significant since all p-values are lower than the specified significance level (0.05), and the null hypothesis of the McNemar's statistical test can be rejected.

Table 8.10: McNemar statistical significance test for comparison of supervised vs. self-superivsed learning approaches

Model	Classification	Learning type (SL vs. SSL), p-value
ViT-16	kNN	$p = .033$
	FCNN	$p < .001$
ResNet-50	kNN	$p = .023$
	FCNN	$p = .001$

8.5. Feature Visualization

For a better understanding of the features extracted from the feature extraction model, the features are visualized in the feature space using the t-SNE technique, described in Chapter 3.1.2..

In the Figures 8.1 and 8.2, the visualizations were presented for the CASIAB-LT dataset, using the proposed FCNN classification algorithm, for the features extracted using the ViT-16 feature extraction model.

In Figure 8.1, the features are visualized for all 50 individuals in the query subset of the dataset, where each individual is annotated with another color. Similarly, different angles present in the dataset are annotated with various symbols. From the visualization, it can be seen that many clusters are formed, each representing an individual. In general, clusters are further away from each other, indicating that the learned features have good separability in the feature space. Furthermore, it can be seen that the angles that are close together are also fairly close one to another in the feature space, indicating that the feature extraction model is capable of understanding the relation between different angles.

In Figure 8.2, the individuals are annotated as in Figure 8.1, however, here the focus is on different modalities present in the dataset. In this case, clusters are also formed, however, they are closer together than in Figure 8.1. Considering different covariates, the clustering across the covariates is also apparent. For example, it can be seen that the CL covariates are grouped closer together in the lower part of the feature space.

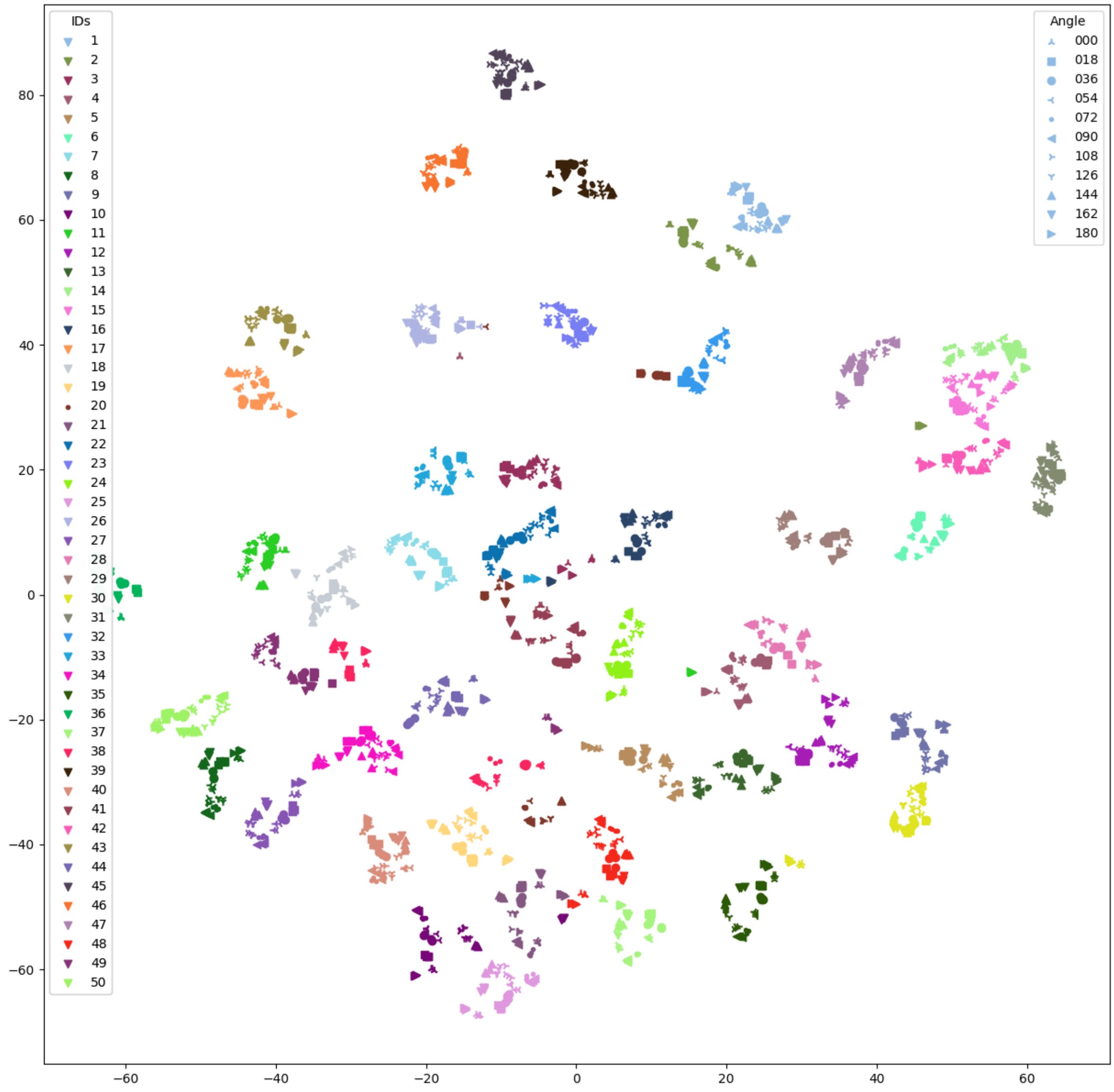


Figure 8.1: t-SNE visualization of learned gait features, with the focus on angles

8.6. Self-Attention Visualization

Our study further evaluated the feature interpretation of the DINO model by visualizing distinct attention heads present in the last multi-head self-attention block. An image from each dataset was randomly selected to showcase the attention dynamics. The model employed for this analysis was the ViT small model, characterized by $n = 6$ heads per self-attention block.

Figures 8.3 and 8.4 represent random images extracted from the CASIA-B and OU-MVLP datasets, respectively. As evident in Figures 8.3 a) and 8.4 a), each attention

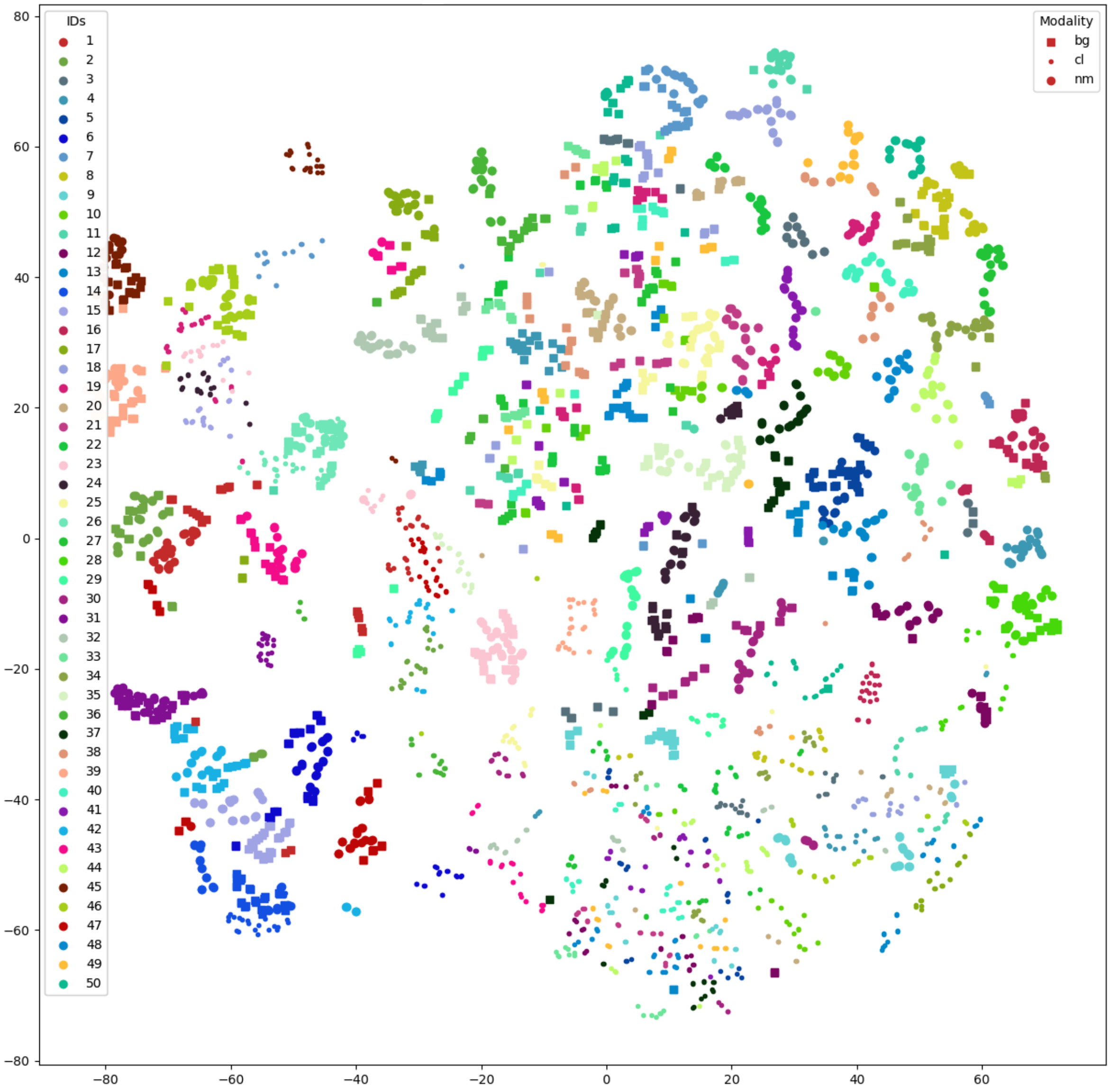
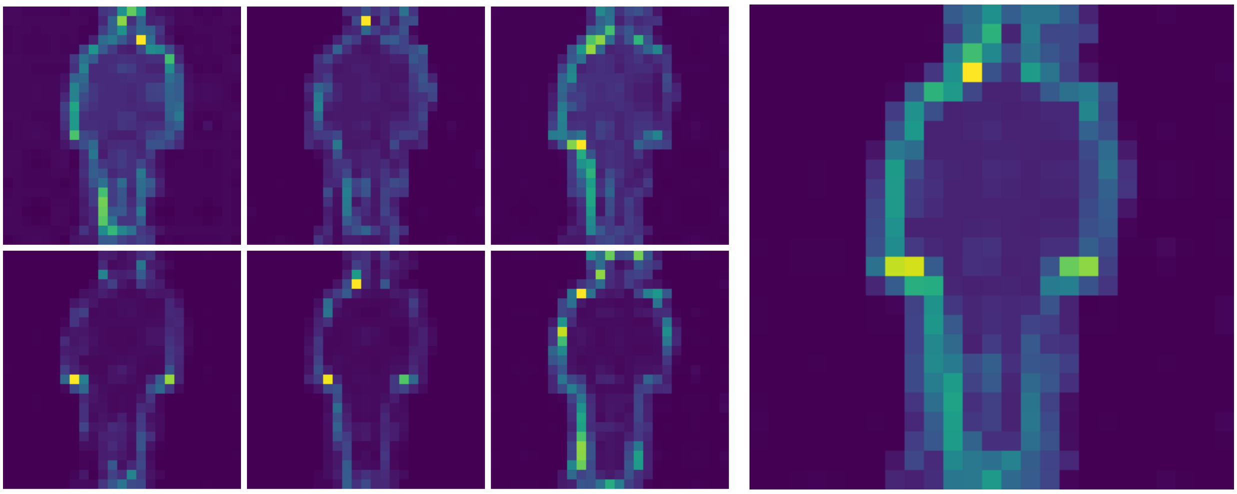


Figure 8.2: t-SNE visualization of learned gait features, with the focus on covariates

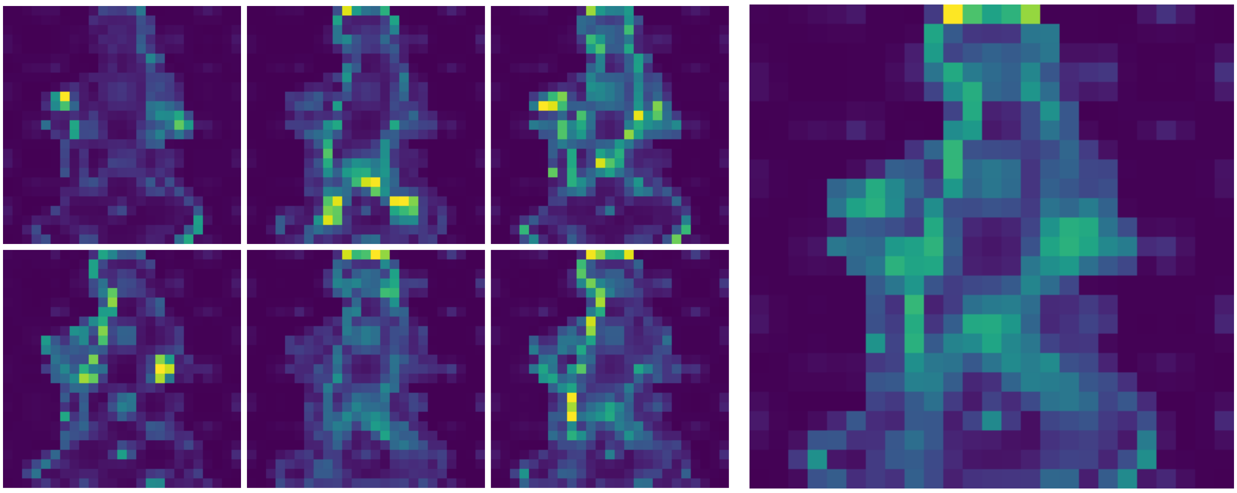
head appears to learn unique features from the data, concentrating on different regions of the image. For instance, some attention heads predominantly focus on the subject's head, while others are drawn to the legs or the left or right side of the subject. Figures 8.3 b) and 8.4 b) aggregate the attention across all heads, showcasing the average focus. These observations resonate with the original findings reported in the DINO manuscript, emphasizing that the DINO technique effectively delineates objects of interest within the image. In the context of GEI images, the subject's outline emerges as the most significant region. The proposed methodology adeptly identifies and utilizes this critical information for individual identification, leading to compelling results as delineated in Chapter 6.2..



(a) Self-attention heads

(b) Average of all self-attention heads

Figure 8.3: Self-attention of the [CLS] token on random CASIA-B sample image



(a) Self-attention heads

(b) Average of all self-attention heads

Figure 8.4: Self-attention of the [CLS] token on random OU-MVLP sample image

9. Chapter

CONCLUSIONS

In this doctoral dissertation, an approach for gait recognition using the self-supervised self-attention deep learning model is proposed. For validating the proposed approach, the series of experiments were performed. The experimental setup was detailed, the data used in the experiments was prepared, and the feature extraction model, as well as classification models, were trained on the two widely used gait recognition datasets.

First, the data was acquired in the form of the datasets, and the data was preprocessed in order to adjust the data for use for the feature extraction model. Then, the self-supervised self-attention deep learning feature extraction models were trained on two datasets, CASIA-B and OU-MVLP, where CASIA-B dataset consisted of three different settings, depending on the number of individuals in the data set for training and testing. Finally, the proposed FCNN classifier is trained on the features extracted from the feature extraction model, and evaluated using various classification metrics, as well as statistical tests to confirm the statistical significance of the obtained results.

The achieved results show great recognition accuracy of the proposed approach, on par with other state-of-the-art approaches. Despite using no labels when training the feature extraction model using self-supervision, the proposed approach in some cases outperformed supervised counterparts in global average rank-1 accuracy. Moreover, the proposed approach shows great robustness in terms of accuracy across different angles at which the individual is recorded, without significant decline in accuracy even if the angle difference is large. Also, the proposed approach is robust to the different covariates, such as a bag, demonstrating the ability to generalize well, with the potential for use in the real-world use cases, where such occurrences are common.

Additionally, an ablation study was performed outlining the importance of using feature extraction model pretraining, on the datasets such as ImageNet or GREW, in boosting the recognition accuracy compared to training without the pretraining. Furthermore, a comparison between the supervised and self-supervised learning was conducted by performing experiments on the CASIAB-LT dataset, demonstrating the efficacy of the self-supervised learning in contrast to supervised learning that is mainly used in the literature.

In the future work, different deep learning models will be explored as backbone models for self-supervised feature extraction of gait features, focusing on the learning more representative gait features. Furthermore, new approaches for self-supervised learning will be examined, with the application to the task of gait recognition. Finally, the effect of different human body parts on the recognition accuracy will be analysed, and the findings will be incorporated in future gait recognition approaches.

BIBLIOGRAPHY

[1] W. Jansen, "Authenticating users on handheld devices," in *Proceedings of the Canadian Information Technology Security Symposium*, 2003, pp. 1–12.

[2] Ž. Emeršič, D. Štepec, V. Štruc, and P. Peer, "Training convolutional neural networks with limited training data for ear recognition in the wild," *arXiv preprint arXiv:1711.09952*, 2017.

[3] Ž. Emeršič, V. Štruc, and P. Peer, "Ear recognition: More than a survey," *Neurocomputing*, vol. 255, pp. 26–39, 2017.

[4] C. Wan, L. Wang, and V. V. Phoha, "A survey on gait recognition," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1–35, 2018.

[5] F. M. Castro, M. J. Marin-Jimenez, N. Guil, and N. Pérez de la Blanca, "Multimodal feature fusion for cnn-based gait recognition: an empirical comparison," *Neural Computing and Applications*, vol. 32, pp. 14 173–14 193, 2020.

[6] Z. Wu, Y. Huang, L. Wang, X. Wang, and T. Tan, "A comprehensive study on cross-view gait based human identification with deep cnns," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 2, pp. 209–226, 2016.

[7] Y. Zhang, Y. Huang, L. Wang, and S. Yu, "A comprehensive study on gait biometrics using a joint cnn-based method," *Pattern Recognition*, vol. 93, pp. 228–236, 2019.

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

[9] C. Zhang, W. Liu, H. Ma, and H. Fu, "Siamese neural network based gait recognition for human identification," in *2016 ieee international conference on acoustics, speech and signal processing (ICASSP).* IEEE, 2016, pp. 2832–2836.

[10] Y. Wang, Y. Xia, and Y. Zhang, "Beyond view transformation: feature distribution consistent gans for cross-view gait recognition," *The Visual Computer*, vol. 38, no. 6, pp. 1915–1928, 2022.

[11] Z. Zhang, L. Tran, X. Yin, Y. Atoum, X. Liu, J. Wan, and N. Wang, "Gait recognition via disentangled representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4710–4719.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition.* Ieee, 2009, pp. 248–255.

[14] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9650–9660.

[15] J. Taborri, E. Palermo, S. Rossi, and P. Cappa, "Gait partitioning methods: A systematic review," *Sensors*, vol. 16, no. 1, p. 66, 2016.

[16] M. Kumar, N. Singh, R. Kumar, S. Goel, and K. Kumar, "Gait recognition based on vision systems: A systematic survey," *Journal of Visual Communication and Image Representation*, vol. 75, p. 103052, 2021.

[17] A. Muro-De-La-Herran, B. Garcia-Zapirain, and A. Mendez-Zorrilla, "Gait analysis methods: An overview of wearable and non-wearable systems, highlighting clinical applications," *Sensors*, vol. 14, no. 2, pp. 3362–3394, 2014.

[18] B. Huang, M. Chen, P. Huang, and Y. Xu, "Gait modeling for human identification," in *Proceedings 2007 IEEE international conference on robotics and automation.* IEEE, 2007, pp. 4833–4838.

[19] L. Rong, D. Zhiguo, Z. Jianzhong, and L. Ming, "Identification of individual walking patterns using gait acceleration," in *2007 1st international conference on bioinformatics and biomedical engineering.* IEEE, 2007, pp. 543–546.

[20] E. Vildjiounaite, S.-M. Mäkelä, M. Lindholm, R. Riihimäki, V. Kyllönen, J. Mäntyjärvi, and H. Ailisto, "Unobtrusive multimodal biometrics for ensuring privacy and information security with personal devices," in *Pervasive Computing: 4th International Conference, PERVASIVE 2006, Dublin, Ireland, May 7-10, 2006. Proceedings 4.* Springer, 2006, pp. 187–201.

[21] M. O. Derawi, P. Bours, and K. Holien, "Improved cycle detection for accelerometer based gait authentication," in *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing.* IEEE, 2010, pp. 312–317.

[22] Y. Zhong and Y. Deng, "Sensor orientation invariant mobile gait biometrics," in *IEEE international joint conference on biometrics.* IEEE, 2014, pp. 1–8.

[23] B. Sun, Y. Wang, and J. Banda, "Gait characteristic analysis and identification based on the iphone's accelerometer and gyrometer," *Sensors*, vol. 14, no. 9, pp. 17 037–17 054, 2014.

[24] J. Suutala and J. Röning, "Towards the adaptive identification of walkers: Automated feature selection of footsteps using distinction sensitive lvq," in *Int. Workshop on Processing Sensory Information for Proactive Systems (PSIPS 2004)*, 2004, pp. 14–15.

[25] J. Jenkins and C. Ellis, "Using ground reaction forces from gait analysis: Body mass as a weak biometric," in *Pervasive Computing: 5th International Conference, PERVASIVE 2007, Toronto, Canada, May 13-16, 2007. Proceedings 5.* Springer, 2007, pp. 251–267.

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

[27] A. Roy, S. Sural, and J. Mukherjee, "Gait recognition using pose kinematics and pose energy image," *Signal Processing*, vol. 92, no. 3, pp. 780–792, 2012.

[28] M. Otero, "Application of a continuous wave radar for human gait recognition," in *Signal Processing, Sensor Fusion, and Target Recognition XIV*, vol. 5809. SPIE, 2005, pp. 538–548.

[29] D. Tahmoush and J. Silvious, "Radar micro-doppler for long range front-view gait recognition," in *2009 IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems*. IEEE, 2009, pp. 1–6.

[30] R. D. Seely, S. Samangooei, M. Lee, J. N. Carter, and M. S. Nixon, "The university of southampton multi-biometric tunnel and introducing a novel 3d gait dataset," in *2008 IEEE Second International Conference on Biometrics: Theory, Applications and Systems*. IEEE, 2008, pp. 1–6.

[31] C. BenAbdelkader, R. Cutler, and L. Davis, "Stride and cadence as a biometric in automatic person identification and verification," in *Proceedings of Fifth IEEE international conference on automatic face gesture recognition*. IEEE, 2002, pp. 372–377.

[32] L. Wang, H. Ning, T. Tan, and W. Hu, "Fusion of static and dynamic body biometrics for gait recognition," *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 2, pp. 149–158, 2004.

[33] A. I. Bazin and M. S. Nixon, "Probabilistic combination of static and dynamic gait features for verification," in *Biometric Technology for Human Identification II*, vol. 5779. SPIE, 2005, pp. 23–30.

[34] Y. Wang, S. Yu, Y. Wang, and T. Tan, "Gait recognition based on fusion of multiview gait sequences," in *Advances in Biometrics: International Conference, ICB 2006, Hong Kong, China, January 5-7, 2006. Proceedings*. Springer, 2005, pp. 605–611.

[35] S.-I. Choi, J. Moon, H.-C. Park, and S. T. Choi, "User identification from gait analysis using multi-modal sensors in smart insole," *Sensors*, vol. 19, no. 17, p. 3785, 2019.

[36] P. Kumar, S. Mukherjee, R. Saini, P. Kaushik, P. P. Roy, and D. P. Dogra, "Multimodal gait recognition with inertial sensor data and video using evolutionary algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 5, pp. 956–965, 2018.

[37] J. Moon, N. A. Le, N. H. Minaya, and S.-I. Choi, "Multimodal few-shot learning for gait recognition," *Applied Sciences*, vol. 10, no. 21, p. 7619, 2020.

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

[39] G. Ariyanto and M. S. Nixon, "Marionette mass-spring model for 3d gait biometrics," in *2012 5th IAPR International Conference on Biometrics (ICB)*. IEEE, 2012, pp. 354–359.

[40] S. Yu, D. Tan, and T. Tan, "Modelling the effect of view angle variation on appearance-based gait recognition," in *Computer Vision–ACCV 2006: 7th Asian Conference on Computer Vision, Hyderabad, India, January 13-16, 2006. Proceedings, Part I 7*. Springer, 2006, pp. 807–816.

[41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[42] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 2, pp. 316–322, 2005.

[43] M. A. R. Ahad, J. K. Tan, H. Kim, and S. Ishikawa, "Motion history image: its variants and applications," *Machine Vision and Applications*, vol. 23, pp. 255–281, 2012.

[44] C. Wang, J. Zhang, L. Wang, J. Pu, and X. Yuan, "Human identification using temporal information preserving gait template," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2164–2176, 2011.

[45] D. A. Reynolds *et al.*, "Gaussian mixture models." *Encyclopedia of biometrics*, vol. 741, no. 659-663, 2009.

[46] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[47] A. J. Izenman and A. J. Izenman, "Linear discriminant analysis," *Modern multivariate statistical techniques: regression, classification, and manifold learning*, pp. 237–280, 2008.

[48] E. Fix, *Discriminatory analysis: nonparametric discrimination, consistency properties.* USAF school of Aviation Medicine, 1985, vol. 1.

[49] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[50] H. Chao, Y. He, J. Zhang, and J. Feng, "Gaitset: Regarding gait as a set for cross-view gait recognition," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 8126–8133.

[51] V. C. d. Lima, V. H. Melo, and W. R. Schwartz, "Simple and efficient pose-based gait recognition method for challenging environments," *Pattern Analysis and Applications*, vol. 24, no. 2, pp. 497–507, 2021.

[52] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[53] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[54] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[55] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[56] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[57] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[58] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9912–9924, 2020.

[59] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[60] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[61] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[62] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[65] Niyogi and Adelson, "Analyzing and recognizing walking figures in xyt," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 1994, pp. 469–474.

[66] D. Cunado, M. S. Nixon, and J. N. Carter, "Using gait as a biometric, via phase-weighted magnitude spectra," in *Audio-and Video-based Biometric Person Authentication: First International Conference, AVBPA'97 Crans-Montana, Switzerland, March 12–14, 1997 Proceedings 1.* Springer, 1997, pp. 93–102.

[67] J.-H. Yoo and M. S. Nixon, "Markerless human gait analysis via image sequences," 2003.

[68] R. Urtasun and P. Fua, "3d tracking for gait characterization and recognition," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.* IEEE, 2004, pp. 17–22.

[69] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, "The humanid gait challenge problem: Data sets, performance, and analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 2, pp. 162–177, 2005.

[70] J. Liu and N. Zheng, "Gait history image: a novel temporal template for gait recognition," in *2007 IEEE international conference on multimedia and expo.* IEEE, 2007, pp. 663–666.

[71] K. Lenac, D. Sušanj, A. Ramakić, and D. Pinčić, "Extending appearance based gait recognition with depth data," *Applied Sciences*, vol. 9, no. 24, p. 5529, 2019.

[72] 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 Communication and Image Representation*, vol. 25, no. 1, pp. 195–206, 2014.

[73] J.-H. Yoo, D. Hwang, K.-Y. Moon, and M. S. Nixon, "Automated human recognition by gait using neural network," in *2008 First Workshops on Image Processing Theory, Tools and Applications.* IEEE, 2008, pp. 1–6.

[74] W. T. Dempster and G. R. Gaughran, “Properties of body segments based on size and weight,” *American journal of anatomy*, vol. 120, no. 1, pp. 33–54, 1967.

[75] C. Yan, B. Zhang, and F. Coenen, “Multi-attributes gait identification by convolutional neural networks,” in *2015 8th international congress on image and signal processing (CISP)*. IEEE, 2015, pp. 642–647.

[76] Y. Feng, Y. Li, and J. Luo, “Learning effective gait features using lstm,” in *2016 23rd international conference on pattern recognition (ICPR)*. IEEE, 2016, pp. 325–330.

[77] K. Shiraga, Y. Makihara, D. Muramatsu, T. Echigo, and Y. Yagi, “Geinet: View-invariant gait recognition using a convolutional neural network,” in *2016 international conference on biometrics (ICB)*. IEEE, 2016, pp. 1–8.

[78] 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, pp. 81–93, 2017.

[79] S. Yu, H. Chen, E. B. Garcia Reyes, and N. Poh, “Gaitgan: Invariant gait feature extraction using generative adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 30–37.

[80] Y. He, J. Zhang, H. Shan, and L. Wang, “Multi-task gans for view-specific feature learning in gait recognition,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 102–113, 2018.

[81] C. Song, Y. Huang, Y. Huang, N. Jia, and L. Wang, “Gaitnet: An end-to-end network for gait based human identification,” *Pattern recognition*, vol. 96, p. 106988, 2019.

[82] Y. Zhang, Y. Huang, S. Yu, and L. Wang, “Cross-view gait recognition by discriminative feature learning,” *IEEE Transactions on Image Processing*, vol. 29, pp. 1001–1015, 2019.

[83] C. Fan, Y. Peng, C. Cao, X. Liu, S. Hou, J. Chi, Y. Huang, Q. Li, and Z. He, “Gaitpart: Temporal part-based model for gait recognition,” in *Proceedings of the*

IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 14 225–14 233.

[84] C. Xu, Y. Makihara, X. Li, Y. Yagi, and J. Lu, “Cross-view gait recognition using pairwise spatial transformer networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 1, pp. 260–274, 2020.

[85] A. Sokolova and A. Konushin, “Pose-based deep gait recognition,” *IET Biometrics*, vol. 8, no. 2, pp. 134–143, 2019.

[86] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

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

[88] G. Huang, Z. Lu, C.-M. Pun, and L. Cheng, “Flexible gait recognition based on flow regulation of local features between key frames,” *IEEE Access*, vol. 8, pp. 75 381–75 392, 2020.

[89] R. Liao, W. An, Z. Li, and S. S. Bhattacharyya, “A novel view synthesis approach based on view space covering for gait recognition,” *Neurocomputing*, vol. 453, pp. 13–25, 2021.

[90] L. Zhao, L. Guo, R. Zhang, X. Xie, and X. Ye, “mmgaitset: multimodal based gait recognition for countering carrying and clothing changes,” *Applied Intelligence*, vol. 52, no. 2, pp. 2023–2036, 2022.

[91] Z. Zhu, X. Guo, T. Yang, J. Huang, J. Deng, G. Huang, D. Du, J. Lu, and J. Zhou, “Gait recognition in the wild: A benchmark,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 14 789–14 799.

[92] A. Cosma and I. E. Radoi, “Wildgait: Learning gait representations from raw surveillance streams,” *Sensors*, vol. 21, no. 24, p. 8387, 2021.

[93] D. Pinčić, D. Sušanj, and K. Lenac, "Gait recognition with self-supervised learning of gait features based on vision transformers," *Sensors*, vol. 22, no. 19, p. 7140, 2022.

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

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

[96] N. Takemura, 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 Transactions on Computer Vision and Applications*, vol. 10, no. 1, pp. 1–14, 2018.

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

[98] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, vol. 2. IEEE, 1999, pp. 246–252.

[99] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," *Video-based surveillance systems: Computer vision and distributed processing*, pp. 135–144, 2002.

[100] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.

[101] J. Perry, J. R. Davids *et al.*, "Gait analysis: normal and pathological function," *Journal of Pediatric Orthopaedics*, vol. 12, no. 6, p. 815, 1992.

[102] R. T. Collins, R. Gross, and J. Shi, “Silhouette-based human identification from body shape and gait,” in *Proceedings of fifth IEEE international conference on automatic face gesture recognition.* IEEE, 2002, pp. 366–371.

[103] L. Wang, T. Tan, H. Ning, and W. Hu, “Silhouette analysis-based gait recognition for human identification,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 12, pp. 1505–1518, 2003.

[104] M. Ekinci, “Human identification using gait,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 14, no. 2, pp. 267–291, 2006.

[105] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *International conference on machine learning.* PMLR, 2021, pp. 10 347–10 357.

[106] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in *European conference on computer vision.* Springer, 2016, pp. 499–515.

[107] F. Research, “Dino,” 2021. [Online]. Available: https://github.com/facebookresearch/dino

[108] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.

[109] Q. McNemar, “Note on the sampling error of the difference between correlated proportions or percentages,” *Psychometrika*, vol. 12, no. 2, pp. 153–157, Jun. 1947. [Online]. Available: https://doi.org/10.1007/bf02295996

[110] S. Zhang, Y. Wang, and A. Li, “Cross-view gait recognition with deep universal linear embeddings,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9095–9104.

[111] W. G. Cochran, “The comparison of percentages in matched samples,” *Biometrika*, vol. 37, no. 3/4, pp. 256–266, 1950.

LIST OF FIGURES

1.1 Overview of the commonly used biometrics for identifying individuals . . . 3

1.2 Gait Biometric 9

2.1 The Gait Cycle [16] 18

5.1 Example of the raw RGB images from the CASIA-B dataset [95] 62

5.2 Example of the GEI images from the OU-MVLP dataset [96] 63

5.3 Example of the raw images from the GREW dataset [91] 64

6.1 Gait recognition pipeline 66

6.2 Example of the generated Gait Energy Image (GEI) 68

6.3 DINO self-supervised learning [14]. The goal of a student network is to match the probability distribution of a teacher network using cross-entropy loss, given different views of the same input image. 69

6.4 The architecture of the ViT model [12] 73

6.5 Proposed FCNN classifier 78

8.1 t-SNE visualization of learned gait features, with the focus on angles . . . 100

8.2 t-SNE visualization of learned gait features, with the focus on covariates . 101

8.3 Self-attention of the [CLS] token on random CASIA-B sample image 102

8.4 Self-attention of the [CLS] token on random OU-MVLP sample image . . . 102

LIST OF TABLES

8.1 Results for CASIA-B dataset ST setting 88
8.2 Results for CASIA-B dataset MT setting 89
8.3 Results for CASIA-B dataset LT setting 90
8.4 Results for OU-MVLP dataset 91
8.5 McNemar statistical significance test for different patch sizes for ViT model 92
8.6 Detailed results for the CASIA-B and OU-MVLP datasets 93
8.7 Results for ablation study of feature extraction model pretraining 95
8.8 McNemar statistical significance test for feature extraction model pretraining 96
8.9 Results for ablation study of supervised vs. self-supervised learning approaches 98
8.10 McNemar statistical significance test for comparison of supervised vs. self-superivsed learning approaches 99

A1 Complete results for CASIA-B dataset ST setting 123
A2 Complete results for CASIA-B dataset MT setting 124
A3 Complete results for CASIA-B dataset LT setting 125
A4 Complete results for OU-MVLP dataset 126

Appendices

Table A1: Complete results for CASIA-B dataset ST setting

Model	Modality	Angle												Average
		0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°	Mean	
ViT-16	NM	100.00	100.00	100.00	99.50	98.00	99.00	99.00	98.50	99.50	99.50	100.00	99.36	62.30
	BG	83.50	75.00	68.50	55.56	63.00	52.00	54.00	56.78	55.78	72.08	81.41	65.24	
	CL	25.50	25.00	24.00	20.60	22.50	23.00	21.00	15.58	20.10	24.37	23.50	22.29	
ViT-8	NM	100.00	100.00	98.50	97.50	97.50	99.00	98.50	99.50	99.00	100.00	100.00	99.05	67.69
	BG	87.50	87.00	74.50	71.72	72.00	73.50	72.00	73.37	78.39	82.74	85.43	78.01	
	CL	28.00	28.00	26.50	27.14	25.00	28.00	26.00	20.10	21.61	27.92	28.00	26.02	
GaitSet [50]	NM	64.60	83.30	90.40	86.50	80.20	75.50	80.30	86.00	87.10	81.40	59.60	79.54	63.03
	BG	55.80	70.50	76.90	75.50	69.70	63.40	68.00	75.80	76.20	70.70	52.50	68.64	
	CL	29.40	43.10	49.50	48.70	42.30	40.30	44.90	47.40	43.00	35.70	25.60	40.90	
mmGaitSet [90]	NM	78.50	90.70	94.00	92.20	88.10	84.40	87.40	91.70	92.40	90.60	73.90	87.63	71.84
	BG	70.40	81.40	84.70	82.70	77.40	73.00	77.90	83.00	82.00	79.60	65.40	77.95	
	CL	42.20	54.60	58.30	57.00	53.00	49.50	51.40	52.20	51.20	45.60	34.40	49.95	
Huang et al. [88]	NM	67.40	81.60	88.80	87.00	80.70	74.90	79.20	86.70	88.20	82.00	66.70	80.29	64.38
	BG	57.80	70.60	77.10	76.20	70.10	64.30	68.70	76.00	75.40	70.30	54.60	69.19	
	CL	33.40	47.10	53.10	48.80	46.10	41.20	47.40	47.70	47.10	39.00	29.30	43.65	
GaitPart [83]	NM	62.50	97.90	87.50	64.60	93.80	95.80	93.80	97.90	70.80	91.70	75.00	84.66	63.76
	BG	52.10	70.80	58.30	43.80	79.20	81.20	77.10	77.10	66.70	77.10	52.10	66.86	
	CL	22.90	29.20	35.40	33.30	39.60	62.50	52.10	52.10	33.30	43.80	33.30	39.77	

Table A2: Complete results for CASIA-B dataset MT setting

Model	Modality	Angle												Average
		0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°	Mean	
ViT-16	NM	100.00	100.00	100.00	99.19	99.19	99.19	99.19	98.39	99.19	99.19	100.00	99.41	67.05
	BG	89.52	87.90	74.19	77.24	66.94	64.52	66.94	73.39	71.77	80.49	87.10	76.36	
	CL	25.81	29.03	23.39	23.39	26.61	26.61	24.19	21.77	22.58	26.61	29.03	25.37	
ViT-8	NM	98.39	100.00	98.39	99.19	99.19	98.39	99.19	99.19	100.00	100.00	99.19	99.19	68.90
	BG	87.90	85.48	84.68	76.42	73.39	70.16	72.58	77.42	75.81	82.11	87.10	79.37	
	CL	33.87	29.84	26.61	28.23	36.29	32.26	24.19	20.97	20.16	25.00	32.26	28.15	
GaitSet [50]	NM	86.80	95.20	98.00	94.50	91.50	89.10	91.10	95.00	97.40	93.70	80.20	92.05	79.61
	BG	79.90	89.80	91.20	86.70	81.60	76.70	81.00	88.20	90.30	88.50	73.00	84.26	
	CL	52.00	66.00	72.80	69.30	63.10	61.20	63.50	66.50	67.50	60.00	45.90	62.53	
mmGaitSet [90]	NM	94.40	98.10	99.30	98.20	96.10	94.40	96.30	98.10	98.40	97.90	92.30	96.68	88.23
	BG	90.50	95.00	94.30	94.60	91.60	88.90	91.20	93.90	94.90	92.30	84.80	92.00	
	CL	73.60	79.50	82.70	82.20	76.40	73.50	74.70	78.30	77.00	72.60	65.50	76.00	
Huang et al. [88]	NM	86.70	95.40	97.80	96.30	91.60	87.00	91.40	96.80	95.90	93.30	82.50	92.25	80.43
	BG	80.10	89.90	91.30	87.80	84.00	75.80	81.10	88.60	90.70	85.50	73.70	84.41	
	CL	58.30	71.10	76.80	71.50	64.50	58.90	64.00	68.50	68.80	59.50	49.10	64.64	
GaitPart [83]	NM	63.10	79.40	84.60	79.80	77.00	72.60	77.40	80.30	84.00	78.50	63.70	76.40	58.33
	BG	47.50	59.60	64.20	66.30	61.30	56.70	63.40	63.30	61.80	57.50	47.00	58.96	
	CL	30.20	43.30	43.40	43.10	43.60	41.90	40.00	40.30	41.40	38.70	29.90	39.62	

Table A3: Complete results for CASIA-B dataset LT setting

Model	Modality	Angle												Average
		0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°	Mean	
ViT-16	NM	100.00	100.00	99.00	100.00	99.00	97.00	99.00	98.00	99.00	99.00	99.00	99.00	68.93
	BG	88.00	86.00	74.00	78.79	79.00	78.00	71.00	80.00	83.00	84.85	88.00	80.97	
	CL	33.00	29.00	24.00	26.00	24.00	34.00	21.00	22.00	23.00	25.00	34.00	26.82	
ViT-8	NM	100.00	100.00	100.00	98.00	98.00	97.00	98.00	100.00	100.00	99.00	99.00	99.00	68.68
	BG	88.00	83.00	78.00	73.74	76.00	81.00	75.00	79.00	78.00	80.81	87.00	79.96	
	CL	27.00	28.00	31.00	35.00	29.00	35.00	27.00	22.00	18.00	24.00	22.00	27.09	
GaitSet [50]	NM	90.80	97.90	99.40	96.90	93.60	91.70	95.00	97.80	98.90	96.80	85.80	94.96	84.18
	BG	83.80	91.20	91.80	88.80	83.30	81.00	84.10	90.00	92.20	94.40	79.00	87.24	
	CL	61.40	75.40	80.70	77.30	72.10	70.10	71.50	73.50	73.50	68.40	50.00	70.35	
mmGaitSet [90]	NM	95.60	99.50	99.90	98.80	95.90	95.40	96.20	98.90	98.90	98.40	94.40	97.45	90.09
	BG	91.40	95.60	94.10	94.30	91.40	88.60	90.00	93.00	95.70	95.70	88.10	92.54	
	CL	77.60	84.40	85.80	84.70	78.90	76.60	78.50	79.30	82.20	82.80	72.20	80.27	
Huang et al. [88]	NM	91.10	97.90	99.60	97.30	94.30	91.90	94.90	98.10	98.80	96.20	86.60	95.15	85.69
	BG	84.30	91.20	93.40	91.80	86.10	80.30	84.40	90.90	93.70	90.80	80.10	87.91	
	CL	64.70	79.40	84.10	80.40	73.70	72.30	75.00	78.50	77.90	71.20	57.00	74.02	
GaitPart [83]	NM	94.10	98.60	99.30	98.50	94.00	92.30	95.90	98.40	99.20	97.80	90.40	96.23	89.13
	BG	89.10	94.80	96.70	95.10	88.30	94.90	89.00	93.50	96.10	93.80	85.80	92.46	
	CL	70.70	85.50	86.90	83.30	77.10	72.50	76.90	82.20	83.80	80.20	66.50	78.69	

Table A4: Complete results for OU-MVLP dataset

Model	Angle														Average
	0°	15°	30°	45°	60°	75°	90°	180°	195°	210°	225°	240°	255°	270°	
ViT-16	79.12	88.62	88.19	84.89	81.36	83.20	83.42	83.80	89.66	88.68	86.58	84.01	84.92	83.89	85.02
ViT-8	77.09	87.39	86.45	84.16	78.41	82.38	82.60	79.84	87.81	87.24	85.64	79.95	83.57	83.37	83.28
GEINet [77]	11.40	29.10	41.50	45.50	39.50	41.80	38.90	14.90	33.10	43.20	45.60	39.40	40.50	36.30	35.76
Zhang et al. [110]	56.20	73.70	81.40	82.00	78.40	78.00	76.50	60.20	72.00	79.80	80.20	76.70	76.30	73.90	74.66
Zhang et al. [82]	74.00	88.30	94.60	95.40	88.00	91.30	90.00	76.70	89.50	95.00	94.90	88.00	90.80	89.80	89.02
GaitSet [50]	79.50	87.90	89.90	90.20	88.10	88.70	87.80	81.70	86.70	89.00	89.30	87.20	87.80	86.20	87.14
SelfGait [94]	85.10	89.30	92.00	94.30	89.10	90.20	90.90	87.40	91.80	89.30	88.70	90.80	91.60	87.70	89.87

CURRICULUM VITAE

Domagoj Pinčić was born on August 3, 1993, in Rijeka, Croatia. After completing his primary and secondary education, he enrolled in the University of Rijeka, Faculty of Engineering, in 2012. He completed his undergraduate and graduate university studies in computing at the same institution in 2018. In 2018, he enrolled in the Faculty of Engineering for postgraduate university doctoral study in the area of engineering sciences in the field of computer science.

During his postgraduate studies, he was employed as a professional associate / young researcher at the University of Rijeka, Faculty of Engineering, department of computer science. He has participated in several scientific projects, such as "DATACROSS" and the University of Rijeka research project "Embedded system for 3D perception", under the leadership of Prof. D. Sc. Kristijan Lenac.

LIST OF PUBLICATIONS

[1] K. Lenac, D. Sušanj, A. Ramakić, and D. Pinčić, “Extending appearance based gait recognition with depth data,” *Applied Sciences*, vol. 9, no. 24, p. 5529, 2019.

[2] D. Sušanj, D. Pinčić, and K. Lenac, “Effective area coverage of 2d and 3d environments with directional and isotropic sensors,” *IEEE Access*, vol. 8, pp. 185 595–185 608, 2020.

[3] D. Pinčić, D. Sušanj, and K. Lenac, “Gait recognition with self-supervised learning of gait features based on vision transformers,” *Sensors*, vol. 22, no. 19, p. 7140, 2022.

[4] L. Batistić, D. Sušanj, D. Pinčić, and S. Ljubic, “Motor imagery classification based on eeg sensing with visual and vibrotactile guidance,” *Sensors*, vol. 23, no. 11, p. 5064, 2023.

[5] B. Dušić, D. Beževan, D. Pinčić, and Z. Jeričević, “Automatic focusing of optical microscope using off-the-shelf hardware components and in house software,” in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).* IEEE, 2018, pp. 0201–0203.