Vision Based System for Detecting and Counting Mobility Aids in Surveillance Videos

A thesis

submitted in fulfilment

of the requirements for the Degree

of

Doctor of Philosophy in Electronics

at

The University of Waikato

by

Amir Mukhtar

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

2022

Abstract

Automatic surveillance video analysis is popular among computer vision researchers due to its wide range of applications that require automated systems. Automated systems are to replace manual analysis of videos which is tiresome, expensive, and time-consuming. Image and video processing techniques are often used in the design of automatic detection and monitoring systems. Compared with normal indoor videos, outdoor surveillance videos are often difficult to process due to the uncontrolled environment, camera angle, and varying lighting and weather conditions.

This research aims to contribute to the computer vision field by proposing an object detection and tracking algorithm that can handle multi-object and multi-class scenarios. The problem is solved by developing an application to count disabled pedestrians in surveillance videos by automatically detecting and tracking mobility aids and pedestrians. The application demonstrates that the proposed ideas achieve the desired outcomes. There are extensive studies on pedestrian detection and gait analysis in the computer vision field, but limited work is carried out on identifying disabled pedestrians or mobility aids. Detection of mobility aids in videos is challenging since the disabled person often occludes mobility aids and visibility of mobility aid depends on the direction of the walk with respect to the camera. For example, a walking stick is visible most times in front-on view while it is occluded when it happens to be on the walker's rear side. Furthermore, people use various mobility aids and their make and type changes with time as technology advances. The system should detect the majority of mobility aids to report reliable counting data. The literature review revealed that no system exists for detecting disabled pedestrians or mobility aids in surveillance videos. A lack of annotated image data containing mobility aids is also an obstacle to developing a machine-learning-based solution to detect mobility aids.

In the first part of this thesis, we explored moving pedestrians' video data to extract the gait signals using manual and automated procedures. Manual extraction involved marking the pedestrians' head and leg locations and analysing those signals in the time domain. Analysis of stride length and velocity features indicate an abnormality if a walker is physically disabled. The

automated system is built by combining the YOLO object detector, GMM based foreground modelling and star skeletonisation in a pipeline to extract the gait signal. The automated system failed to recognise a disabled person from its gait due to poor localisation by YOLO, incorrect segmentation and silhouette extraction due to moving backgrounds and shadows. The automated gait analysis approach failed due to various factors including environmental constraints, viewing angle, occlusions, shadows, imperfections in foreground modelling, object segmentation and silhouette extraction.

In the later part of this thesis, we developed a CNN based approach to detect mobility aids and pedestrians. The task of identifying and counting disabled pedestrians in surveillance videos is divided into three sub-tasks: mobility aid and person detection, tracking and data association of detected objects, and counting healthy and disabled pedestrians. A modern object detector called YOLO, an improved data association algorithm (SORT), and a new pairing approach are applied to complete the three sub-tasks. Improvement of the SORT algorithm and introducing a pairing approach are notable contributions to the computer vision field. The SORT algorithm is strictly one class and without object counting feature. SORT is enhanced to be multi-class and able to track accelerating or temporarily occluded objects. The pairing strategy associates a mobility aid with the nearest pedestrian and monitors them over time to see if the pair is reliable. A reliable pair represents a disabled pedestrian and counting reliable pairs calculates the number of disabled people in the video. The thesis also introduces an image database that was gathered as part of this study. The dataset comprises 5819 images belonging to eight different object classes, including five mobility aids, pedestrians, cars, and bicycles. The dataset was needed to train a CNN that can detect mobility aids in videos.

The proposed mobility aid counting system is evaluated on a range of surveillance videos collected from outdoors with real-world scenarios. The results prove that the proposed solution offers a satisfactory performance in picking mobility aids from outdoor surveillance videos. The counting accuracy of 94% on test videos meets the design goals set by the advocacy group that need this application. Most test videos had objects from multiple classes in them. The system detected five mobility aids (wheelchair, crutch, walking stick, walking frame and mobility scooter), pedestrians and two distractors (car and bicycle). The training system on distractors' classes was to ensure the system can distinguish objects that are similar to mobility aids from mobility aids. In some cases, the convolutional neural network reports a mobility aid with an incorrect type. For example, the shape of crutch and stick are very much alike, and therefore, the system confuses one with the other. However,

it does not affect the final counts as the aim was to get the overall counts of mobility aids (of any type) and determining the exact type of mobility aid is optional.

Acknowledgements

First, I am indebted to my thesis advisor, Dr Michael Cree, for his immense support and guidance throughout my PhD years. He has been a great mentor, and under his supervision, I learned how to define a research problem, find a solution to it, and finally publish the results. This thesis would not have been completed without his commitment and encouragement, which not only influenced the content in the thesis but also in my personal development. I am also grateful to my co-supervisors Prof. Jonathan Scott and Dr Lee Streeter, for all their efforts and unconditional support during the period of my research. I can never forget the support and helping hands lent by all my supervisors towards me in scheduling regular meetings, prompt reviews on the articles with critical suggestions, enabling me to rehearse my presentations etc., despite the amount of work at their end, reflecting their commitment and devotion.

This research has been made possible with funding from Callaghan Innovation, NZ (TRAD 1401). I am also grateful to Dr. Bridget Burdett, my project manager at Stantec, who was influential in winning the project grant from Callaghan Innovation. My special thanks to Bridget for ensuring a functional office space, flexible working hours, freedom, prompt advice and help in collecting video data from different places in Hamilton City. Also, a token of gratitude to Stroke Foundation NZ, for allowing me to visit their monthly congregations and helping me to reach participants for video data. I'd also like to thank my participants who agreed to take part in the study and took the time to walk in front of the surveillance cameras. Special thanks to Ian Honey for help in logistics and setting up the surveillance system at collection sites.

I want to thank all my colleagues at the School of Engineering for making it a fun and exciting place to work. Much respect to my office mates, especially Dr. Mohammad Hedayati, for sharing many stimulating discussions and work time fun. Thanks to Gehan for helping me to annotate training images. I also wish to thank school of Engineering Administrator Mary Dalbeth for bearing with me for all the administrative chores. My acknowledgement would be incomplete without paying thanks to Ali Abbas, my Manager at Livestock Improvement Corporation NZ, who was always encouraging, supportive and

offered flexible working hours. I am grateful for friendly chats at the end of our meetings and his support in my academic and business endeavours. Finally, I can never forget all my good friends (it is a long list) in Hamilton, for making my stay so much more pleasant. All of them were like a second family to me and were always ready to help me.

My deepest appreciation and thanks to my dad, Muhammad Mukhtar, for his love and all the sacrifices he made towards my life and education. If not for his efforts, I would not be who I am.

I want to thank Almighty, my whole family, for their support and the unconditional love. My final yet special deepest gratitude to my loving wife, Shumaila. I know how much she had to manage life during my PhD time, by compromising with many spoiled weekends.

I must say that the list is incomplete. I thank everyone for their blessings and support throughout my life.

Contents

List of Figures

List of Tables

Acronyms

BB Bounding Boxes.

BCE Binary Cross Entropy.

CMU Carnegie Mellon University.

CNN Convolutional Neural Network.

DBT Detection Based Tracking.

DFT Detection Free Tracking.

DHG Difference of Histogram of Gradients.

GEI Gait Energy Image.

GMM Gaussian Mixture Model.

GPU Graphic Processing Unit.

GRU Gated Recurrent Unit.

HOFEI Histogram of Optic Flow Energy Image.

HOG Histogram of Oriented Gradients.

ICA Independent Component Analysis.

INRIA National Institute for Research in Digital Science and Technology.

IOU Intersection over Union.

JPDA Joint Probabilistic Data Association.

KF Kalman Filter.

LBP Local binary patterns.

MEI Motion Energy Image.

MHI Motion History Image.

MHT Multiple Hypothesis Tracking.

MMSI Moving Motion Silhouette Images.

MOT Multiobject Tracking.

NMS Non-Maximum Suppression.

R-CNN Region-based Convolutional Network.

reID Re-identification.

RNN Recurrent Neural Networks.

RPN Region Proposal Network.

SORT Simple Online and Realtime Tracking.

SSD Single Shot multi-box Detector.

SVM Support Vector Machine.

YOLO You Only Look Once.

Chapter 1

Introduction

Over the past few decades, video surveillance has increased in public areas, such as railway stations, airports, workplaces, universities, shopping centres and supermarkets. Video surveillance technology is popular due to its low cost, ease of installation, freedom from interference, and its ability to capture detailed information in images/videos. Surveillance cameras capture a tremendous amount of video data, and manual analysis of the surveillance videos is very time-consuming, painstaking, expensive and prone to errors. Alternatively, automated monitoring can go around the clock and reduce the video analysis time at a reasonable price (Paul et al. 2013, Candamo et al. 2010, Chen & Huang 2013). Advancements in optical sensors and rapidly falling costs of high-resolution cameras also make them a preferred choice for an automated detection system design. Recent progress in computer vision and machine learning areas has resulted in many systems for analysing surveillance videos. These systems include abnormal event detection (Candamo et al. 2010), crowd estimation (Zhan et al. 2008), traffic monitoring (Robert 2009) and pedestrian detection (Paul et al. 2013)

This thesis is motivated by an advocacy group who wants to count people using mobility aids to advocate on behalf of disabled people. However, the government (local and central) is reluctant to act unless they can present reliable data. The data that has certain robustness to it but getting pedestrian (healthy and disabled) counts is time-consuming, expensive and prone to inaccuracies due to manual counting. Therefore, they want a cost-effective system, capable of working with surveillance cameras around the place and able to process videos in real-time or offline afterwards and spit out counts data. The solution has to be consistent, i.e., the system should always give the same count from running the same video. The end user does not need the detection of individual people but require the counts of people who have walked through the scene and the counts of disabled people with the counting accuracy of 80% and above compared with human counts.

To automate the counting process, the advocacy group needs a system that can detect people and mobility aids in the video, track them and perform data association because the system should not count them multiple times during their stay on the scene. However, people re-entering the scene can be counted twice as the application is intended to report the number of disabled pedestrians using public pathways. The system should be smart enough to track pedestrians to the point that the system does not miss/recount people who just come obscured after a certain period. Furthermore, the system should be capable of associating person and mobility aids together so the person using the mobility aid (i.e., disabled person) can be identified.

Automated recognition of disabled pedestrians in surveillance videos is a mostly unexplored application and presents new complexities as it involves pedestrian detection, gait analysis and mobility aid detection. Pedestrian detection and gait analysis are well-studied endeavours (Paul et al. 2013, Afsar et al. 2015, Lee et al. 2013) in computer vision, but the detection of disabled pedestrians or people using mobility aids (Myles et al. 2002, Huang et al. 2009) is less investigated. A few existing studies only focus on one type of mobility aid appearing in controlled indoor environments with fixed backgrounds. Disabled pedestrian detection in surveillance videos is a challenging task since outdoor scenarios with background clutter, illumination conditions, camera viewing angle and camera-resolution affect the quality of information sensed. Scenes obtained from a surveillance video are usually low resolution which adds to the complexity of the problem.

This study focuses on extracting a walking person's gait signal to investigate whether gait signal can be useful to distinguish between healthy and disabled persons. The motivation was that the human body's dynamic parts follow an oscillatory motion during the walk or run, and our brain can distinguish between a healthy and disabled person based on walker's gait. Classical gait extraction (skeletonisation) approach is experimented as it is computationally cheap and summarises a walker's gait without a prior human model. We employ an object detector and skeletonisation technique to extract the walker's head and legs' locations over time and look for an unusual pattern in disabled pedestrians' gait signal.

Manual extraction of gait signals revealed that a walker's gait contains the information indicating unusual movement patterns. However, an automated scheme based on motion detection (Stauffer & Grimson 1999) and skeletonisation (Fujiyoshi & Lipton 1998), failed to reproduce the same information due to problem of shadow and segmentation leading to inaccuracies in extracting the silhouette of the moving person. Later on, a different method is proposed after exploring a Convolutional Neural Network (CNN) called You Only Look

Once (YOLO) to detect mobility aids and pedestrians in videos. The system design is based on the assumption that a pair of person and mobility aid indicates the presence of a disabled person at the scene. There are various types of mobility aids (wheelchair, mobility scooter, walking frame, crutch, walking stick and artificial limb), and their shape/size keeps evolving with advances in technology. Identifying mobility aids requires an intelligent system to identify a variety of mobility aids in uncontrolled outdoor environments. It is also challenging to associate the detected mobility aid with a pedestrian to find a disabled pedestrian. Moreover, pedestrian's body often occludes the mobility aid due to its proximity.

A lack of annotated mobility aids image dataset makes it difficult to train a convolutional neural network for detecting mobility aids in images makes it difficult to train a system for the task. Therefore, we collected the images of mobility aids and manually labelled them to build a custom dataset for this study. A CNN is trained on this custom database to detect mobility aids and pedestrians in real-world images/videos. Mobility aids are not seen in isolation and are associated with the nearest pedestrian. This brings an advantage of re-identifying people or mobility aids that became obscured for a while or lost in tracking, during their passage. We evaluate the proposed technique on a range of surveillance videos and demonstrate the successful operation by the system.

In this research, we have integrated YOLO detections with an enhanced version of the Simple Online and Realtime Tracking (SORT) (Bewley et al. 2016), and pairing and counting modules to develop a full detection, tracking and counting system. Tracking various objects in crowded or busy intersections can be complicated and challenging to retain individual objects' identities given the cluttered and frequent occlusions. Multi-object tracking refers to tracking dynamically varying multiple objects belonging to the same class (pedestrians, cars, bicycle etc.). In contrast, multi-class tracking involves tracking objects of different appearance, shape and size in a video sequence. Due to intricate design and high computation requirements, the majority of tracking algorithms are developed to work with multiple objects of the same class (Zhao et al. 2008, Kim et al. 2015, Bewley et al. 2016, Wojke et al. 2017), but there are some studies (Khan et al. 2005, 2006, Lee et al. 2016) that have focused on multi-object tracking with unlimited classes of objects.

The main contributions of this research to the literature are:

1. Demonstration that the gait signal, which captures a walker's movement patterns, can categorise pedestrians into healthy and disabled categories (Section 4.2).

2. A new image dataset is created by gathering and labelling mobility aid images from publicly available image collections. This dataset is needed in this research to investigate a CNN-based solution for detecting mobility aids in videos. Furthermore, we demonstrate that CNNs trained on our custom dataset can detect five different mobility aids in surveillance videos.

3. SORT (Bewley et al. 2016) is a multi-object tracker but not a multi-class tracker. The mobility aid application requires both multi-object tracking and multi-class tracking. In this thesis, the SORT algorithm is enhanced to be multi-class and suitable for counting disabled pedestrians. SORT loses the accelerating objects during detection to track assignment. We solved the problem by replacing Intersection over Union (IOU) in the cost matrix with a distance metric during detection to track assignment part. The proposed algorithm is able to track accelerating or occluded objects under their existing ids, and the counts are not affected.

4. A novel pairing strategy is proposed to form pairs of mobility aid and pedestrians based on the distance between them. A pair corresponds to a mobility aid user, and results show that the pairing approach avoids over counting due to an object's reappearance after the occlusion. The proposed counting approach is successfully working to count mobility aid users in surveillance videos (Section 6.3).

On the application side, this work's main contribution is designing an end-to-end solution to identify and count people using mobility aids from surveillance videos. The proposed system reads a test video and returns the number of healthy and disabled pedestrians, their breakdown by mobility aids type, pairs of mobility aids and pedestrians, and a timeline of counts and pairing history.

1.1 Outline of Thesis

The remaining chapters of this thesis are structured as follows:

- **Chapter 2** provides a detailed literature review of pedestrian detection schemes that use hand-engineered image features and learning algorithms for classifiers. We also review modern CNN based solutions for pedestrian detection. Apart from that, a comprehensive survey of gait analysis and data association algorithms is provided. Furthermore, we detailed various image/video datasets that are commonly used in pedestrian detection and gait analysis tasks.

- **Chapter 3** describes the image and video datasets collected during this study. We have also detailed the size, composition, and set of rules to manually label the images of mobility aids.

- **Chapter 4** illustrates our gait signal extraction and analysis work to identify disabled pedestrian through abnormal gait signatures. We have documented the manual and automated ways of gait signal extraction and analysis. The experimental results on two datasets (CASIA and our custom dataset) are presented and discussed in this chapter.

- **Chapter 5** describes the framework of the overall system design. It explains the design of the mobility aid detector and describes the image dataset used to train and test the mobility aid detector. Results of the detection system are also shown in this chapter.

- **Chapter 6** focuses on counting the mobility aid users by combining the results from the mobility aid detector with a tracker, a data association step and a counting module. Details about the pairing strategy and results of the overall system are provided in this chapter.

- **Chapter 7** analyses the performance of the detection and counting modules. Remarks on how well the system performed and identifying a few limitations of the proposed system. We also listed causes for the system failures and some suggestions on improving the system's performance.

- **Chapter 8** concludes this dissertation with a summary of the research and recommendations for further work.

1.2 List of Publications

Following two papers have been published in the conferences,

- Mukhtar, A., Cree, M. J., Scott, J. B. & Streeter, L. (2018a), 'Gait analysis of pedestrians with the aim of detecting disabled people', Applied Mechanics and Materials 884, 105-112.

- Mukhtar, A., Cree, M. J., Scott, J. B. & Streeter, L. (2018b), Mobility aids detection using convolution neural network (CNN), in '2018 International Conference on Image and Vision Computing New Zealand (IVCNZ)', IEEE, pp. 1-5.

- Mukhtar, A., Cree, M. J., Scott, J. B. & Streeter, L. Vision Based System for Detecting and Counting Mobility Aids in Surveillance Videos. [1]

[1] This paper is in preparation for submission to a journal.

Chapter 2

Literature Review

2.1 Pedestrian Detection

2.1.1 Overview

A pedestrian is a person that walks rather than travelling in a vehicle and pedestrian detection is the problem of determining regions in the image or video sequence that enclose humans. Over recent years, detecting human beings in a video scene has drawn much attention due to its wide range of applications in, for example, surveillance systems (Paul et al. 2013), abnormal event detection (Candamo et al. 2010) human gait characterization (Lee et al. 2013, Ran et al. 2010), pedestrian detection in robotics (Benenson et al. n.d.), driver assistance system (Mukhtar et al. 2015), crowd behaviour detection (Chen & Huang 2011, 2013), gender classification (Hu et al. 2011) and fall detection for elderly people (Thome et al. 2008). Some examples of human detection and their applications are shown in Figures 2.1 and 2.2, respectively.

Identifying pedestrians in an image is a challenging task, but modern computer vision and machine learning researchers have mostly solved it. In general, objects in images and videos can be detected in two sequential steps: extracting candidate regions that are potentially covered by human objects and classifying/verifying the regions as human or non-human. Candidate regions help to narrow down the image area to localise the pedestrians.

There are several methods to extract candidate regions. The simplest way to segregate moving objects is using background subtraction and declaring foreground objects as candidate regions. Various background subtraction methods have been developed, of which (Benezeth et al. 2010) provide a useful comparative study. Another common approach is to use a *sliding window* protocol to extract a rectangular area of fixed width and height that "slides" across an image assuming that humans can be enclosed by a detection "window." A pedestrian can appear in various sizes as it depends on the real-world size of

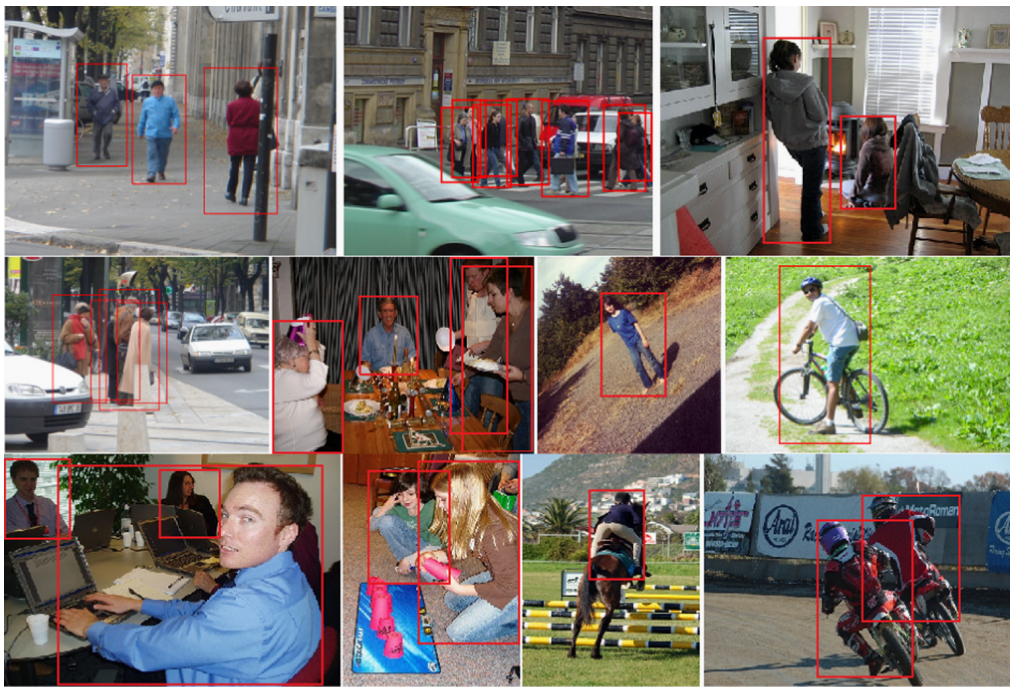


Figure 2.1: Some examples of human detection. Reprinted from Pattern Recognition, Vol 15, Duc Thanh Nguyen, Wanqing Li, Philip O. Ogunbona, Human detection from images and videos: A survey, Page No. 149, Copyright (2016), with permission from Elsevier

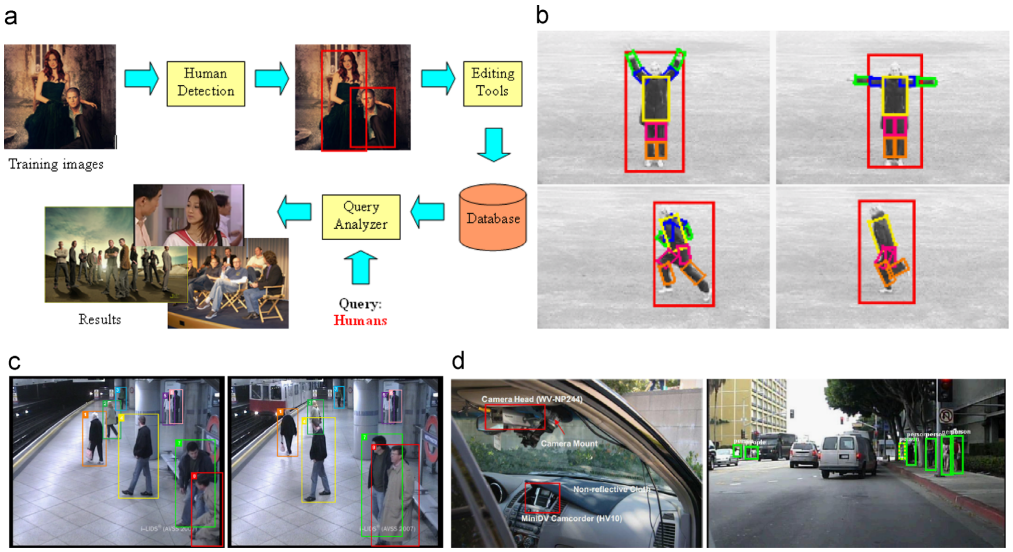


Figure 2.2: Applications of human detection (a) Image retrieval. (b) Human activity recognition. (c) Surveillance systems. (d) Driving assistance system. Reprinted from Pattern Recognition, Vol 15, Duc Thanh Nguyen, Wanqing Li, Philip O. Ogunbona, Human detection from images and videos: A survey, Page No. 149, Copyright (2016), with permission from Elsevier

the pedestrian and his or her distance from the camera. To handle variation

in size, Dollar et al. (2012) proposed a scale-space pyramid approach in which the image was rescaled with different scale ratios to form layers. Then the sliding window method is applied to the features computed for each layer of the scale-space pyramid.

The pedestrian classification stage consists of a classifier trained on the features extracted from the image database. Features can be calculated at individual pixels or at the regional level, and encode information about the shape and appearance of the human object. It can be challenging to find image features that distinguish between targets (pedestrians) and non-targets. The stronger this distinctiveness, the easier it is to correctly identify all the occurrences of the object in the image. The features should be scale and rotation invariant and able to cope with illumination changes. In most cases, a descriptor is a high dimensional feature vector formed after concatenating the locally extracted features. A big set of samples —both from humans (positives) and from the background (negatives)— undergo a feature vector transformation to be later used to train machine learning algorithms.

A machine-learning algorithm such as support vector machine, AdaBoost or a neural network, is used to learn an object model that forms the best balance between containing sufficient detail to distinguish the object from the background, and generic enough to cover inner class variation. The similarity between the pre-trained object model and the image features indicates the probability of the object being present at that location. All candidates regions undergo a classification process to determine the closeness between the image features and the model. A threshold on the classification score decides if detection is reliable or not.

Non-Maximum Suppression (NMS) (Dalal 2006) is the last step to filter out multiple detections associated with the same object. The sliding window process often leads to multiple detections of the same object in the same area, creating a need for the elimination of all detections, which are not a local maximum. NMS algorithm can knock off all unneeded detections while ensuring that each object is still enclosed by one bounding box. Crowdy situations present tricky situation for NMS since multiple objects can appear close together. NMS is generally based on the IOU metric, shown in Equation 2.1, where *B1* and *B2* represent two overlapping bounding boxes classified being positive for the same object.

$$IOU = \frac{area(B1 \cap B2)}{area(B1 \cup B2)} \tag{2.1}$$

We can refer $B1$ as the bounding box for the detected object and $B2$ is that of ground truth used to evaluate the system. A high IOU score (0.75) means a considerable overlap between prediction and ground truth bounding boxes,

and a low IOU score (<0.25) implies poor overlap. Figure 2.3 shows a few IOU scores calculating the extent of overlap of two boxes. The greater the region of overlap, the greater the IOU score. In benchmarking (Dollar et al. 2012, Enzweiler & Gavrila 2008), the IOU between the detection and the ground truth annotation has to be at least 0.5 for correct detection. NMS calculates the IOU scores among the correct detections for the same object and retains only the one with the highest classification score.

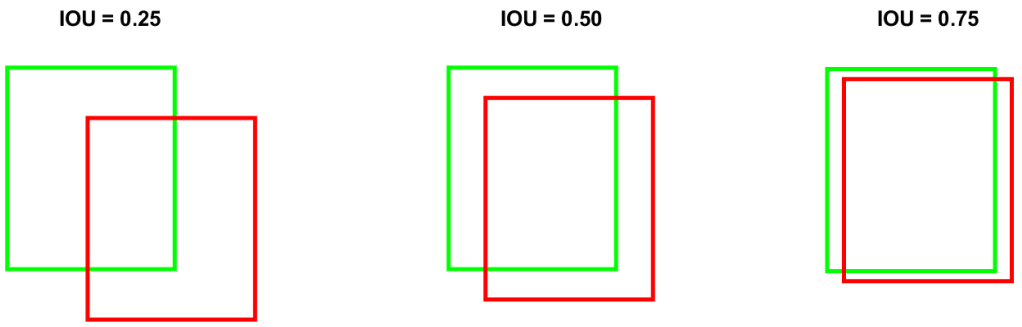


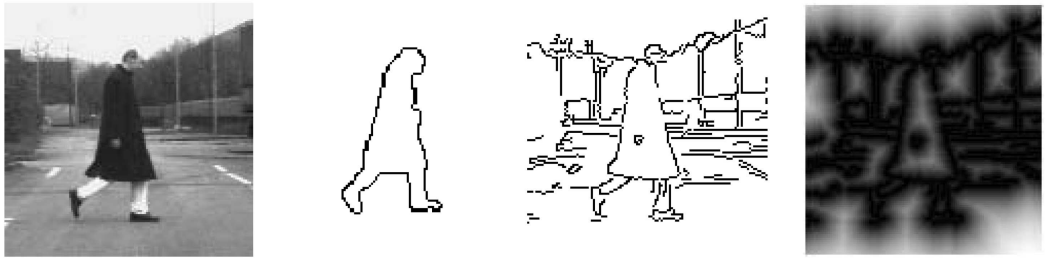
Figure 2.3: Examples of measuring bounding boxes overlaps using IOU metric

The following sections present a survey of image (shape, colour, texture) features and object classifiers that can be used to separate a human from non-human objects.

2.1.2 Image Features

In addition to computer vision, psychological studies have also verified the usefulness of image edge information to describe the shape of an object (De Winter & Wagemans 2004). Studies (De Winter & Wagemans 2004, Gavrila 2007) have shown that binary contours are capable of reflecting the human body in different viewpoints and poses. Edge detection results in gradient images which contain information about position, magnitude and orientation of edges at the pixel level. Several edge templates can be generated to capture variation in human shape, size and pose. The similarity between template images and test frames can indicate the presence of humans in test videos. Gavrila (2007) created a distance transformed image using the Chamfer distance transform (Felzenszwalb & Huttenlocher 2004) and measured the geometrical transformation for which the similarity between edge template and distance transformed image, was maximum. This approach is known as *template matching*, and it has two significant weaknesses. First, noise in images when acquired from uncontrolled environments can easily distort the pixel level edge features. Second, edge templates are pose-specific and require many templates to

incorporate several human poses resulting in increased computational burden. Figure 2.4 shows a conventional template matching process.



(a) (b) (c) (d)

Figure 2.4: (a) Original image, (b) edge template, (c) edge image, and (d) distance transformed image. (© 2007 IEEE. Reprinted, with permission, from Dariu M. Gavrila, A Bayesian, Exemplar-Based Approach to Hierarchical Shape Matching, IEEE Transactions on Pattern Analysis and Machine Intelligence, June/2007)

In contrast to pixel level edge features, regional edge histograms have shown better adaptation to the local deformation of the human shape. A regional edge histogram is calculated by quantising the edge information of pixels in a particular region and accumulating them into a histogram. A well-known example is the *Histogram of Oriented Gradients (HOG)* introduced by Dalal (2006). The HOG feature descriptor computes local edge histograms in which each edge pixel votes for a histogram bin according to edge pixel orientation (see Figure 2.5). Park et al. (2010) combined HOGs computed at multiple resolutions to form a scale-invariant human descriptor. Several extensions and variants of HOG have been suggested for more reliable performances, for example, Wang & Lien (2007) obtained rotation invariant HOG and Hou et al. (2007) extended HOG for non-rectangular regions. Satpathy et al. (2014) proposed the Difference of Histogram of Gradients (DHG) in which each histogram bin carried the absolute difference between two bins of the opposite angles in the typical HOG (see Equation 2.2). Conde et al. (2013) computed HOG descriptor from a Gabor image by filtering the human image using a Gabor filter bank.

$$DHG(bin) = \mid HOG(bin) - HOG(bin + 180) \mid \tag{2.2}$$

Appearance features capture the colour and texture information, and these can be extracted from local image regions. Since pedestrians can appear in a variety of different colours depending on their attire, colour is not an adequate discriminative feature. Some studies have used colour in combination with other features to formulate a feature vector, for example, Ott & Everingham

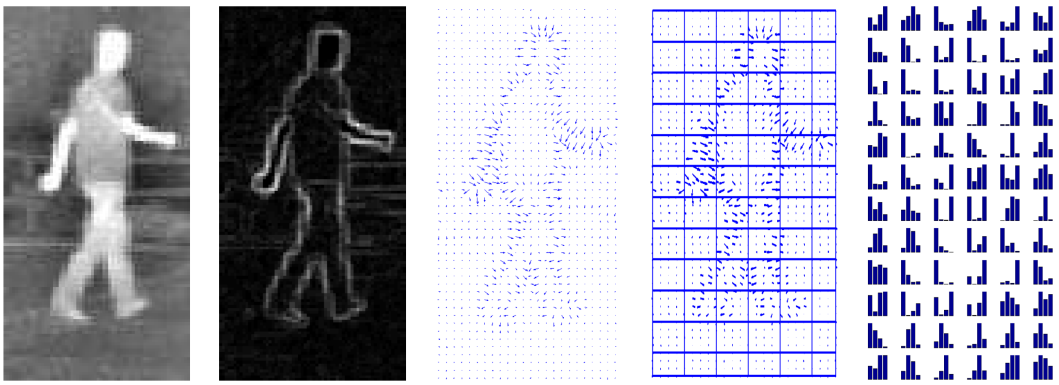


Figure 2.5: Human descriptor using HOG: (a) original image, (b) gradient image, (c) gradient orientation, (d) cell splitting and (e) histogram computation. (© 2007 IEEE. Reprinted, with permission, from M. Bertozzi, A. Broggi, M. Del Rose, M. Felisa, A. Rakotomamonjy and F. Suard, A Pedestrian Detector Using Histograms of Oriented Gradients and a Support Vector Machine Classifier, IEEE Intelligent Transportation Systems Conference, 2007)

(2009) computed the colour HOG (CHOG) vector after segmenting an input image using the distribution of background and foreground colours. In contrast, Walk et al. (2010) calculated the feature set from second-order statistics of colour, that is, self-similarity between the colour of pixels located in local image regions.

Local binary patterns (LBP), originally proposed for texture classification (Ojala et al. 1996), have been explored to describe the shape of the human body (Mu et al. 2008, Hussain & Triggs 2010). Histograms of LBPs are simple to compute and able to capture patterns in an image region. They are also famous for their discriminative power and robustness against illumination changes (Suruliandi et al. 2012, Roy & Marcel 2009). Researchers have proposed numerous extensions of LBP, which are shown in Figure 2.6.

Combining multiple cues can supplement rich information to the descriptors and improve their discriminative power. Wu & Nevatia (2008) combined three types of features: Edgelet (Wu & Nevatia 2007), HOG and covariance matrices (Tuzel et al. 2008) to designed an object detection system. In another study (Opelt et al. 2008), contours features and image intensity features were combined. Wojek et al. (2009) compared the performance of pedestrian detection system for different combinations of object features and classifiers. They concluded that the combination of multiple and complementary feature types could improve detection performance in some cases. However, in general, the feature set is quite sensitive to varying camera views and lighting conditions.

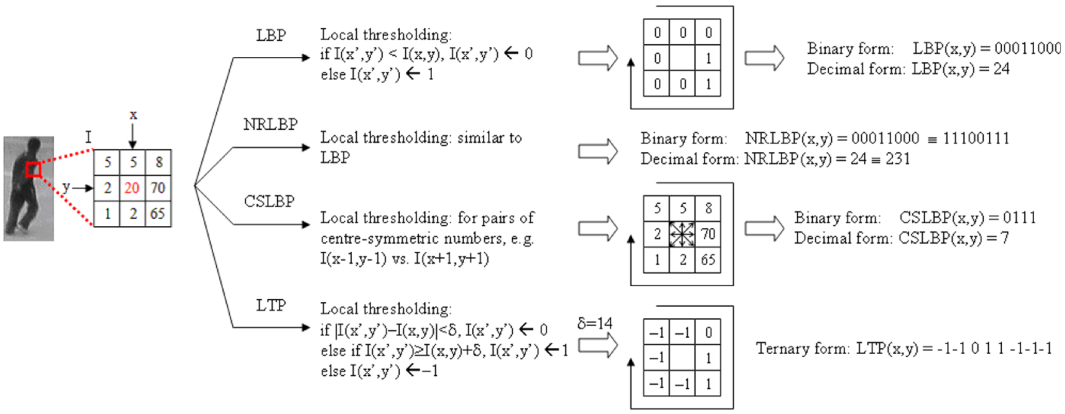
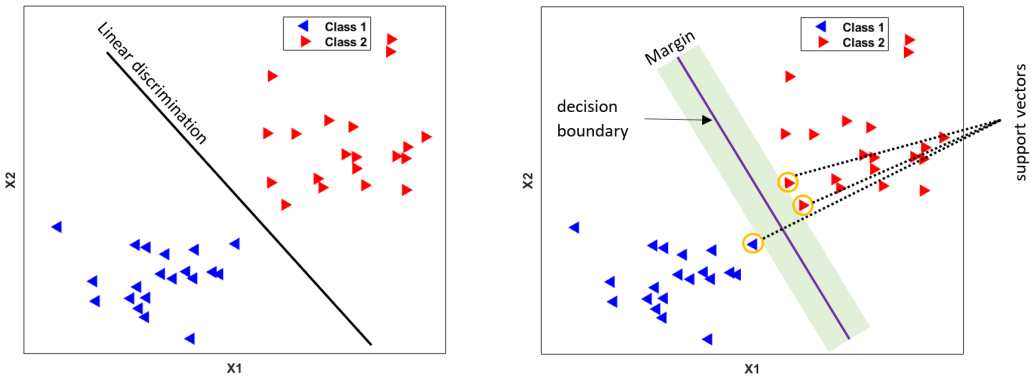


Figure 2.6: LBP and its variants. Reprinted from Pattern Recognition, Vol 15, Duc Thanh Nguyen, Wanqing Li, Philip O. Ogunbona, Human detection from images and videos: A survey, Page No. 149, Copyright (2016), with permission from Elsevier

2.1.3 Classifiers and learning algorithms

After extracting the feature vectors, the candidate regions are categorised as human or non-human classification. The *Support Vector Machine (SVM)* and boosting were the popular classifiers in the computer vision field in the era before deep learning. SVM is based on Vapnik's statistical learning theory (Vapnik 1995), and it aims to separate the training samples to maximise the margin between two classes (see Figure 2.7). The kernel technique maps the feature vectors to a higher dimensional space in which they can be classified linearly. A simple kernel can show impressive results for the pedestrian detection task if the training features are discriminative enough (Dalal & Triggs 2005, Wang et al. 2009). Deepika et al. (2019) studied the impact of the type of kernels on the performance of SVM based classification tasks. Authors observed that the choice of the kernel is dependent on the data set, and the non-linear kernel can improve classification accuracy.

Satpathy et al. (2014) reported that linear SVM classifiers only work well for binary (or distinct) classification problems. For asymmetrical classification problems like human detection versus all other objects, linear SVM were unable to discriminate between the two classes. To solve this, they extracted *Extended Histogram of Gradients* from the training samples and trained a hyper quadratic classifier with the *minimum Mahalanobis distance*. In another study (Ye et al. 2013), the combination of linear SVMs formed a *piecewise linear SVM* (PL-SVM) to discriminate pedestrians in a particular pose from other poses and non-human objects. Each linear SVM was trained on a pose-specific image dataset that contained a group of human objects in some pose.



(a) Linear Discrimination Analysis. (b) SVM

Figure 2.7: Linear discrimination vs SVM Classification

To obtain a scale-invariant human descriptor, Yan et al. (2013) measured HOG features at multiple resolutions and mapped those to a common feature to train an SVM classifier.

The cascade architecture of the Ada-boost classifier combines multiple classifiers to achieve high classification accuracy. Viola et al. (2005) introduced it for human detection, and it has been extended in later studies (Guo et al. 2012, Jang et al. 2016). In Adaboost, an input is passed to the first classifier with decides true or false (pedestrian or not pedestrian). A false outcome halts further computation and causes the detector to return false. A true result passes the input along to the next classifier in the cascade. If all classifiers vote true then the input is classified as a true example, otherwise false. The cascade design is efficient because the classifiers with the fewest features are placed at the start of the cascade, minimizing the computation. Figure 2.8 shows the architecture of a typical Adaboost classifier. Viola et al. (2005) trained Adaboost classifier on motion and appearance features to detect a walking person. The system was able to process four frames per second and recognise pedestrians as small as 20×15 pixels. The system also reported the detection rate of 80% and about one false-positive every two frames for the 360×240 pixel frames.

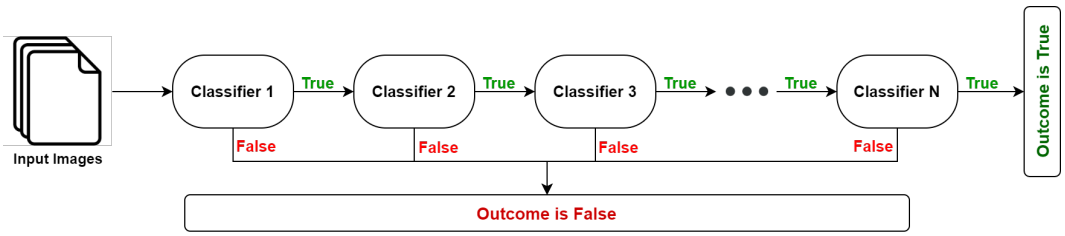


Figure 2.8: Cascade architecture.

2.2 Machine Learning

Machine Learning refers to the study of computer algorithms to build mathematical models based on sample data, known as 'training data' in order to make predictions or decisions without being explicitly programmed to do so. Recently, there has been a rise in the development of machine learning techniques for computer vision-based applications (Redmon & Farhadi 2018, Liu et al. 2016, Girshick et al. 2015, Goodfellow et al. 2016). This is because deep learning methods are outperforming previous state-of-the-art techniques in object detection, and the ImageNet competition (Russakovsky et al. 2015) is the classic example of that.

Deep learning is a family of methods encompassing neural networks, hierarchical probabilistic models, and a variety of unsupervised and supervised feature learning algorithms. CNNs have become popular among computer vision researchers due to their impressive results in object detection tasks. CNNs learn in multiple cascaded layers with each layer taking its input from the output of the previous layer. The larger the number of layers, the "deeper" the network. A key feature of deep learning is the capability of handling large amounts of data and end-to-end learning. In contrast to the techniques mentioned above, the features are not hand-designed but are learnt as part of the whole image-to-detection task and so can work on the raw image data.

The most notable factors that contributed to the prominence of deep learning algorithms are,

1. The appearance of plentiful, high-quality, publicly available and labelled datasets.

2. Parallel computing offered by Graphic Processing Units (GPU) along with the development of powerful frameworks like TensorFlow, Caffe, Theano, and Torch (Bahrampour et al. 2015) which enabled the transition from CPU-based to GPU-based training, thus a significant acceleration in training deep neural networks.

3. New regularization techniques (e.g., dropout, batch normalization, and data augmentation) which have improved the neural networks resilience to over-fitting and performance.

Deep learning has fueled great strides in a variety of computer vision problems, such as object detection (Redmon & Farhadi 2018, Liu et al. 2016, 2019), motion tracking (Doulamis 2018), action recognition (Cao & Nevatia 2016), human pose estimation (Cao et al. 2018, Toshev & Szegedy 2014, Zhao, Yuan & Chen 2019), and semantic segmentation (Long et al. 2015). In the following

subsection, we review the major developments in CNN architecture and object detection applications developed using CNNs.

2.2.1 Convolutional Neural Networks

CNN are a class of deep and feed-forward artificial neural networks (Goodfellow et al. 2016), and they have shown their pattern recognition potentials in human detection (Wojke et al. 2017, Lin et al. 2019, Zhao, Yuan & Chen 2019), lately. They dominate image detection and classification tasks in computer vision and require relatively little pre-processing compared to other image classification algorithms. CNN learns the filter parameters, and it is independent of feature selection and design that in conventional algorithms were hand-engineered.

In general, the design of the network architecture is important and requires specific knowledge and experience. Although fully connected feed-forward neural networks can be trained for learning features in image classification, their architecture is practically not suitable for images since a very high number of neurons would be required. Even a low-resolution image with a shallow architecture results in large input sizes associated as each pixel corresponds to a relevant variable. The convolution operation solves this problem by reducing the number of free parameters, allowing the network to be more in-depth with fewer parameters (Aghdam & Heravi 2017). For instance, regardless of image size, tiling regions of size 7 x 7, each with the same shared weights, requires only 49 learnable parameters. It also resolves complexities of vanishing or exploding gradients during the training phase by using back-propagation.

An image is input directly to the network, and this is followed by several stages of convolution, pooling and/or normalization layers (Hadji & Wildes 2018). Figure 2.9 shows the overall architecture of an object detection system based on CNN classification. Convolutional layers apply a convolution operation on the input data and pass the result to the next layer. The convolution function is specified by a vector of weights and a bias which are adjusted during the training stage depending on loss function and learning rate. CNNs exploit the structure in the image data to reduce the number of parameters to learn; therefore, many neurons share the same filter and require less memory for training. Pooling layers combine the outputs of neuron clusters at one layer into a single neuron in the next layer. After that, representations from these operations feed one or more fully connected layers which give class label(s). Fully connected layers connect every neuron in one layer to every neuron in another layer and are same as those in multi-layer perceptron neural network (MLP) (Gardner & Dorling 1998).

There have been much efforts (Girshick et al. 2015, Ren et al. 2015, Liu et al. 2016, Redmon & Farhadi 2018) that successfully used CNNs for a va-

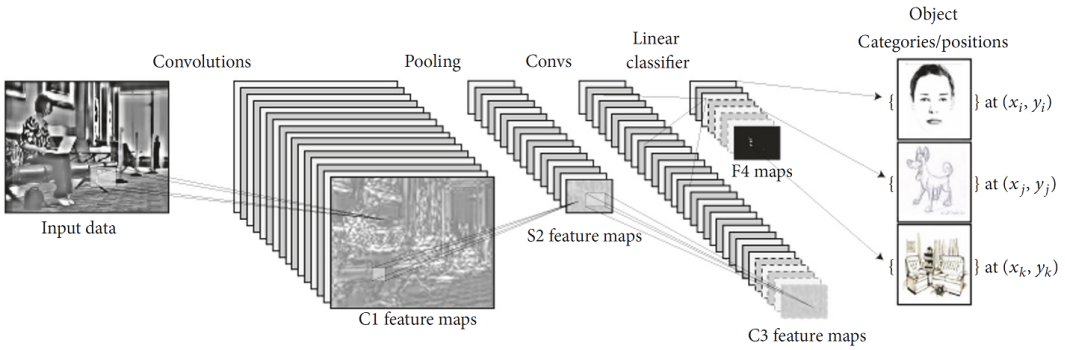
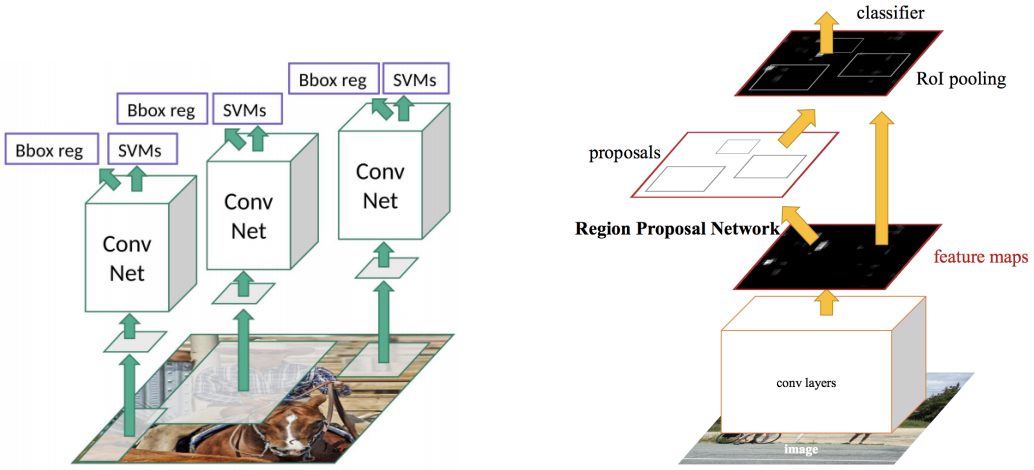


Figure 2.9: Example architecture of a CNN for object detection task. © Voulodimos et al. (2018)

riety of object detection tasks. Girshick et al. (2015) used a selective search algorithm to generate category-independent region proposals that defined the set of candidate detections for CNN. CNN extracted a fixed-length feature vector from each region, and extracted features were fed to an SVM classifier to classify the presence of an object within the region proposals. The object detection performance was measured on PASCAL VOC Challenge datasets, and the system was named Region-based Convolutional Network (R-CNN) since it combined region proposals with CNN. In Faster R-CNN (Ren et al. 2015), region proposals were identified as a computational bottleneck and run time of the RCNN was reduced. The selective search method was replaced by the Region Proposal Network (RPN) to determine a set of rectangular object proposals using a CNN. The computations by CNN were shared with the detection network that has the same set of convolution layers. Tests showed a substantial decrease in running time of region proposals from 1.5 s to 10 ms per image and in total 198 ms for both proposal and detection.

A novel framework (YOLO) was first introduced in 2015 as an advanced real-time object detection system (Redmon et al. 2016). A single neural network predicts bounding boxes and class probabilities for detected objects in images in a single evaluation. YOLO is a single stage detector and runs faster than existing CNNs (see Figure 2.11). The network can process images at 45 FPS, and a simplified version *Fast YOLO* can reach 155 FPS with results comparable with other real-time detectors. YOLOv3 (Redmon & Farhadi 2018), the improved version of YOLO series, has 53 convolutional layers and a hybrid of YOLOv2 (Redmon & Farhadi 2017), Darknet-19, and residual networks. YOLOv3 extracts feature from three different scales using the concept proposed in feature pyramid networks Lin et al. (2017). The last convolutional layer in the network, predicts a 3-d tensor encoding bounding box, objectness, and class predictions. YOLOv3 has a network structure that better utilizes the GPU, making it more efficient to evaluate and thus faster. It has average preci-



(a) R-CNN (b) Faster R-CNN

Figure 2.10: R-CNN and Faster R-CNN architectures. (© 2016 IEEE. Reprinted, with permission, from Shaoqing Ren, Kaiming He, Ross Girshick and Jian Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, June/2016)

sion comparable to the state of art CNNs and outperforms them on processing time criteria (Redmon & Farhadi 2018). The limitation of YOLO is that it struggles with small objects within the image and makes more localization errors compared to the state of art detection systems based on CNNs.

Similar to YOLO, Single Shot multi-box Detector (SSD) (Liu et al. 2016) is another CNN based model to detect objects in a single deep neural network. It predicts category scores and box offsets for a fixed set of default bounding boxes using small convolution filters applied to feature maps. To achieve high detection accuracy, SSD predicts at different scales and separates predictions by aspect ratio. The performance was evaluated on PASCAL VOC, COCO, and ILSVRC datasets and compared to a range of existing CNN based object detectors. Results showed that SSD outperforms Faster R-CNN in processing times and YOLO in location accuracy. Liu et al. (2019) improved the detection accuracy of SSD by adding a series of predictors to evolve the default anchor boxes of SSD directly. Furthermore, a bottleneck block was introduced that combined the advantages of residual learning and multi-scale context encoding to enhance the detectors' discriminative power. Experiments showed that the proposed network improved the SSD's performance by 4.36% on Caltech dataset.

In a recent study (Lin et al. 2019), a multi-scale network and a human parsing generator were jointly trained to detect pedestrians who were heavily occluded or appeared far from cameras. Multi-grained deep features added ro-

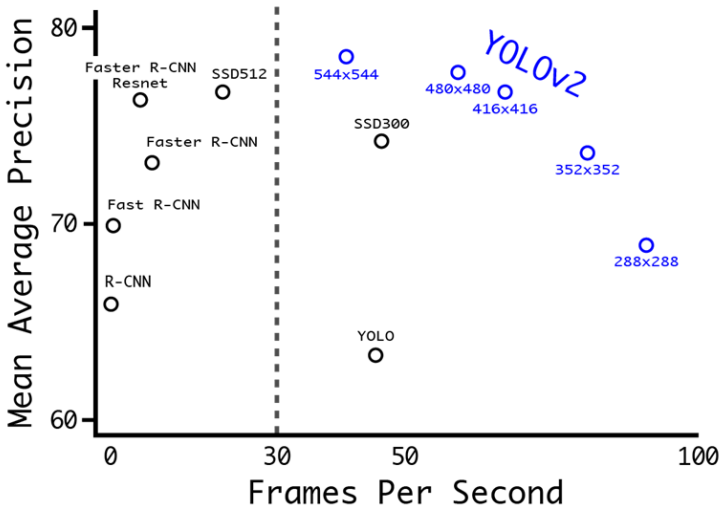


Figure 2.11: YOLOv2 performance comparison. (© 2017 IEEE. Reprinted, with permission, from Joseph Redmon and Ali Farhadi, YOLO9000: Better, Faster, Stronger, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July/2017)

bustness, while the human parsing network generated a fine-grained attention map for the detector to focus on the visible parts of occluded pedestrians and small-sized instances. Both networks are computed in parallel and formed a single-stage framework offering a trade-off between accuracy and speed. The authors also identified a lack of heavily occluded instances in training datasets and developed an adversarial hiding network to generate occlusions by hiding the parts of pedestrians artificially. This improved the robustness and performance of the system when tested on Caltech, KITTI and INRIA datasets for pedestrians class.

Further detailed description and critical analysis on existing CNN based object detection models is given by (Zhao, Zheng, Xu & Wu 2019).

2.3 Gait Analysis

A small subset of human motion analysis techniques is directed towards human recognition by gait due to the computationally intensive task of recovering the human pose from a stream of bi-dimensional images. The small size of pedestrians appearing in the surveillance videos further complicate the problem. The traditional approaches for gait recognition are classified into two major types: model-free approaches and model-based approaches (Lee et al. 2013).

Table 2.1 and Table 2.2 show a timeline of gait detection techniques and their limitations, respectively.

2.3.1 Model Free Methods

Model-free approaches acquire gait parameters by performing measurements directly on 2D images, without adopting a specific model of the human body or motion (for example, silhouettes, history of movements). The model-free approaches characterize the whole motion pattern of the human body by analysing the variations in silhouette shapes or body motion over time, regardless of the underlying structure. In model-free approaches (Kale et al. 2003, Venkat & De Wilde 2011, Fujiyoshi et al. 2004), gait is comprised of a static component based on a person's size and shape, and a dynamic component which follows the actual motion. Gross movements of a walking person can be recognised by obtaining the silhouette of that person after isolating it from the background. That has been the foundation of most model-free approaches to gait recognition to date.

2.3.1.1 Temporal Analysis

A major milestone in gait recognition was setting up the HumanID project and the creation of a test dataset by Sarkar et al. (2005). The dataset contains typical variations in the human walk, including different viewing angles, type of footwear and walking surface. Bounding boxes of the walker were manually defined, and binary silhouettes were cropped. The authors (ibid.) provided a baseline gait recognition algorithm which compared a probe walker sequence (a stride cycle), to those in gallery (database) sequences. A simple baseline algorithm achieved a decent recognition rate of up to 80% for "simple" tasks which dropped to 10% for "difficult" tasks. Sarkar et al. (ibid.) concluded that the lower 20% of the human body was responsible for 80% of the gait recognition.

Kale et al. (2002) proposed a view-based approach to recognize humans through gait. They obtained binary silhouettes after filtering the noise and removing the image background. The width of the binarised silhouette of a walking person was chosen as the image feature, and experiments disclosed that one half of a walk cycle of healthy subjects was enough to provide static and dynamic gait information. Five images (called stances) which represented different stages in a gait cycle, were selected from a video sequence using K-means clustering. L1 norms of width vectors of five stances were concatenated to form a five-dimensional feature vector. The gait of the walker was identified by generating the likelihood from the sequence of test images using a hidden Markov model. Further investigation showed that the five-state hidden Markov model performs better than three and eight state models, however, these methods were vulnerable to changes in the viewing angle. In a later

study, Kale et al. (2003) tracked gait features of the walker using optical flow and derived the angle of the walk. The gait recognition was improved by combining the height information with the leg movements.

2.3.1.2 Spatio-Temporal Motion

The video signal is a time series of bi-dimensional images, and a normal walking cycle can be summarised spatially, temporally or both. Summarised motion features reduce the difficulties in comparing images on a frame by frame basis and help to analyse the human gait for identification purposes. Temporal movement can be summarised by recording the statistics of motion detected during a walking cycle. At the same time, the spatial summary refers to a one-dimensional time-varying signal obtained after converting the two-dimensional silhouette into a single quantity. Venkat & De Wilde (2011) partitioned a silhouette into the overlapping upper, middle, lower, left and right parts and used a Bayesian network for gait identification.

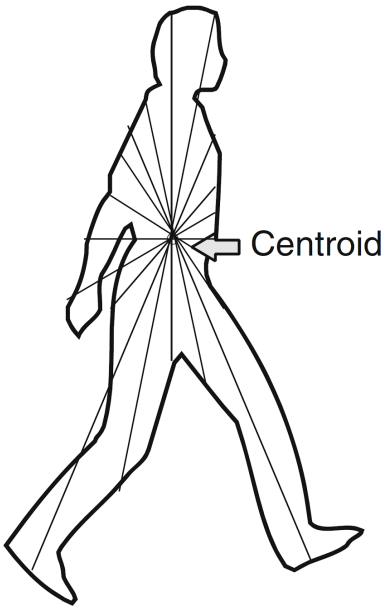
BenAbdelkader et al. (2001) tracked humans in videos and created gait templates from a series of bounding boxes enclosing the silhouette of extreme motions. The templates were scaled to a standard size and subtracted from each other pixel by pixel (spatially) for all time instances (temporally). Later a time-varying two-dimensional similarity plot (Cutler & Davis 2000) of the sum of image pixel differences was generated. The plot displayed a summary of motion and subjected to an eigenvector type of analysis for pattern recognition.

Wang, Hu & Tan (2002) used the "background primal sketch" (Yang & Levine 1992) to acquire a series of silhouette images. Based on the work of Fujiyoshi & Lipton (1998), the two-dimensional silhouette was transformed into a one-dimensional distance vector. A set of points (shown by the lines in Figure 2.12a) was chosen by sampling the silhouette boundary at a fixed angular distance. The sum of all Euclidean distances of boundary points from the origin formed a time-varying distance signal thus summarising the walking activity. A nearest neighbour comparison was used to recognise the dynamic features of the human walk.

In another experiment, Wang, Ning, Hu & Tan (2002) restricted the experimental situation to a single subject moving in the field of view without occlusion. The database had 240 video sequences of the participants walking at 0°, 45° and 90° with respect to the image plane. Silhouettes of the walking person were extracted using a background subtraction procedure, and eigenvector analysis produced the mean image describing the spatio-temporal summary of the walk. Using a k-nearest neighbour classifier, the authors reported 88% to 90% gait recognition rate.

2.3.1.3 Motion History Image and Variants

Davis & Bobick (1997) introduced an effective way to visualise gait features in the form of a motion history image (MHI) summarizing the spatio-temporal motion of pedestrians appearing in videos. In the MHI, pixel intensity is a function of the temporal history of motion at that point. In subsequent frames, values of previous moving pixels decrease while the moving pixels in latter frames add to the brightness of the MHI. The related motion energy image (MEI) results after applying a logical OR operation on the pixels in successive frames and identifies the active regions (Figure 2.12b is an example). These bi-dimensional images representing the motion can be exposed to standard pattern recognition methods to identify people. Similarly, Han & Bhanu (2006) detected pedestrians using gait energy images obtained by averaging the normalised binary silhouette.



(a) Distance from boundary points to centroid.

(b) MEI of a walker, bright pixels show movement.

Figure 2.12: Silhouette based gait features. Reprinted from Multimedia Tools and Applications, Lee, T.K.M., Belkhatir, M. & Sanei, S, A comprehensive review of past and present vision-based techniques for gait recognition, © 2013, with permission from Springer

Zhang et al. (2010) reduced the effect of translational movements on Gait Energy Image (GEI) and generated active energy images (AEI) to focus on the actively moving parts of a silhouette. They obtained the active regions by calculating the difference between two adjacent silhouette images and constructed an AEI by accumulating those active regions. The discriminative

features were extracted after reducing the dimensions of an AEI using two-dimensional locality preserving projections (2DLLP). The system was tested on CASIA gait dataset, and recognition results demonstrated the effectiveness of the proposed method. Another variant, the moving motion silhouette images (MMSI) (Nizami et al. 2010) summarises the motion information of the whole gait sequence. The dimensions of the MMSI were reduced with independent component analysis (ICA), and resulting independent components were used for training probabilistic SVMs to identify gait. The authors (ibid.) quoted a high identification rate of 100% for the CASIA A dataset and 98.67% for the Soton big dataset.

Nambiar (2017) combined the histogram of flow (Dalal et al. 2006) and GEI to compute the histogram of optic flow energy image (HOFEI) for frontal gait analysis. The HOFEI gait signature describes the relative motion of each body part with respect to the other, over a complete gait cycle. The gait period was calculated directly from the optical flow measured within the subjects' bounding box in raw images. The system was tested on HDA person dataset (Figueira et al. 2014), and it reported a correct classification rate of 74.29% under the 'normal walking' conditions.

Recently, researchers have explored deep learning-based gait recognition methods due to the CNN dominance in the pattern recognition field. The first attempt was made by Wu et al. (2016), who used a CNN-based method for human identification by their gait. Binary silhouettes of the subjects moving at eleven different viewing angles, were extracted from the CASIA-B gait dataset using the classical background subtraction techniques. Silhouettes were cropped, rescaled, aligned and stacked to form classes of GEIs for each viewing angle. The network was trained on GEIs using back-propagation and with the logistic regression loss. They evaluated the algorithm on the CASIA-B gait dataset and reported the average recognition rate of 94%. The system was further tested on the OU-ISIR gait dataset (Iwama et al. 2012) (currently the largest gait dataset, with 4007 subjects) where its average accuracy was 91%. Zhang et al. (2019) partitioned the binarised silhouette and GEI of the walker horizontally and trained a CNN for each part. The local features were concatenated to construct a final descriptor. The deep network had a gait-related loss function called angle centre loss to learn discriminative gait features. The loss function showed robustness to different local parts and window sizes.

2.3.2 Model-Based Approaches

Model-based algorithms use explicit gait models, whose parameters are estimated using the underlying kinematics of human motion (e.g., step dimensions, cadence, human skeleton, body dimensions, locations and orientations of body

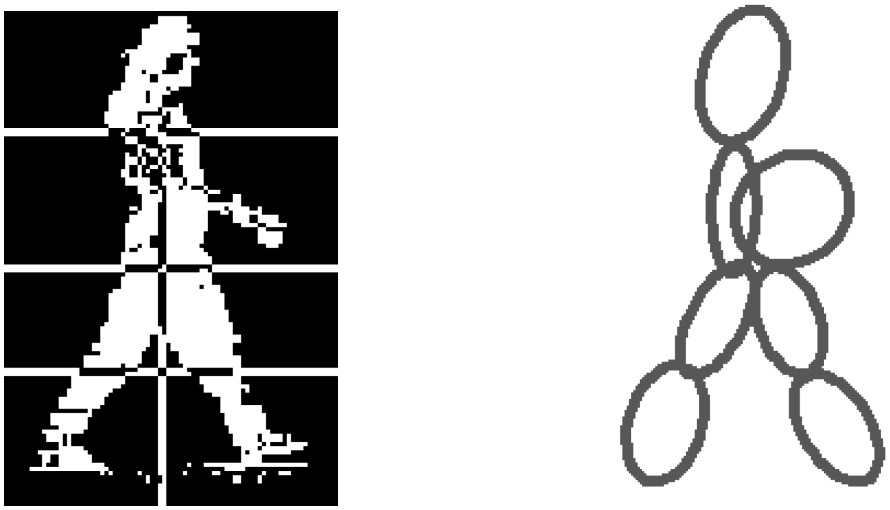
parts and joint kinematics). These methods mostly focus on gait dynamics and are resistant to problems like changes of view and scale. Model-based approaches either model the body (structural models) or walk of the person (motion models) as they appear in the video. Structural models define the human topology as functions of the body parameters, whereas the motion models determine the kinematics of the motion of each body part.

2.3.2.1 Static and Dynamic features

Gait can be modelled by a static component (size and shape of the walker) and a dynamic component that reflects the actual movement. Static features can include dimensions of silhouette, e.g., stride length and walker's height while dynamic features include frequency domain parameters of the movements. Cunado et al. (1997) derived the frequency and phase of the pedestrian walk for analysing the movement of the thigh and calf. A walker was modelled by two lines in a "lambda(λ)" shape which simplified the articulated movement. The change in inclination of these lines followed simple harmonic motion which was used as the gait biometric. Fourier transform analysis was used to calculate the frequency components of the change in inclination of the legs. By multiplying magnitude and phase, the "phase-weighted magnitude spectra" improved the classification rate (90%) than using the magnitude data alone (40%). Authors used k-means clustering to achieve 90% correct classification rate on an indoor video dataset having a static, simple and plain background, controlled lighting and no occlusion.

Lee & Grimson (2002) divided the binary level silhouette into seven elliptically shaped regions (shown in Figure 2.13) and used their geometric measurements as the recognition features. The amplitude of leg swing, arm swing and head orientation, were modelled by the mean and standard deviation of the regional features across time. In this way, the motion was summarised, and the system reported a 17.5% mean error rate when employed for the gender classification task. BenAbdelkader et al. (2002a) utilized stride length and cadence (walking frequency) to describe the gait. After extracting a bounding box around the silhouette, they derived the frequency of the walk by analysing the change in the bounding box dimensions. In a later study, authors improved the performance from 51% to 65% by incorporating height as an additional biometric (BenAbdelkader et al. 2002b).

Johnson & Bobick (2001) labelled body parts by analyzing the binary silhouette of the subject in each video frame. Silhouettes were created by background subtraction using a static background frame. A bounding box was placed around the silhouette and divided into three sections – head section, pelvis section, and foot section (see Figure 2.14a) of predefined sizes. They con-



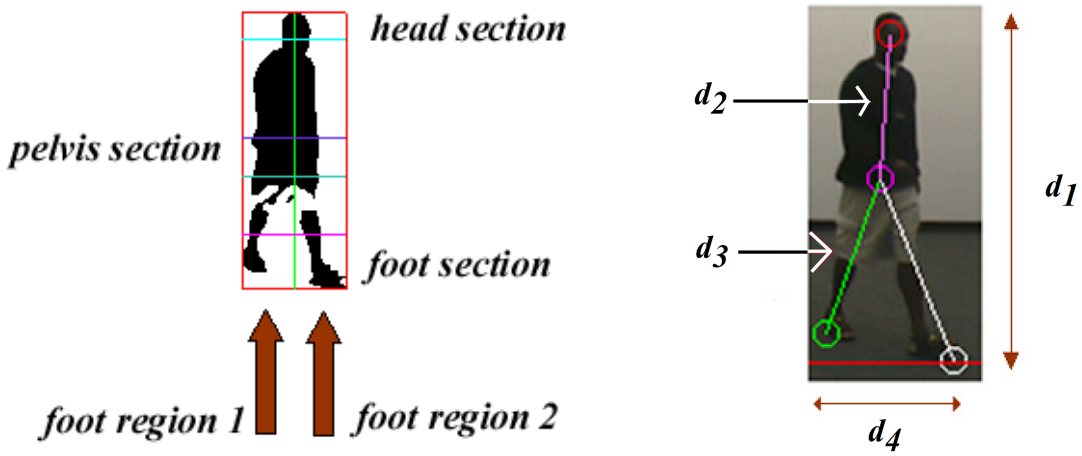
(a) Binary silhouette. (b) Ellipses fitted to each region.

Figure 2.13: The silhouette of a walker divided into 7 regions, and ellipses are fitted to each region. (© 2002 IEEE. Reprinted, with permission, from L. Lee & W.E.L. Grimson, Gait analysis for recognition and classification, Fifth IEEE International Conference on Automatic Face Gesture Recognition, May/2002)

catenated the height of bounding box around the silhouette (d_1), the distance between the head and pelvis (d_2), the maximum value of the distance between the pelvis and left foot (d_3) and the distance between the pelvis and right foot (d_4) to form a 4D-walk vector $\mathbf{w} = < d_1, d_2, d_3, d_4 >$ (see Figure 2.14b). The system reported a 94% gait recognition rate on a dataset made of 18 participants walking at the angle-view and side-on view.

2.3.2.2 Temporal Modelling

Gait sequences can be modelled in an XYT three dimensional space in which time is often considered as the third dimension complementing the XY axes in the image plane (see Figure 2.15). Niyogi et al. (1994) stacked the images from a walking sequence (XY) to form an image cube. Analysis of XT plane uncovered a unique braided pattern formed by the walker's ankle while the head showed a linear motion. The edges of the braid were found by using active contour models or "snakes" (Kass et al. 1988). These contours were calculated at various heights (in Y direction) of the XT slices, which gave a silhouette of the walker in the XY plane. After averaging the silhouette, the walker was represented by a stick figure model, and then four joint angles from that model were used to recognise gait. The L2 norm was used to compare the input image sequences with those in the dataset and 79% gait recognition rate was reported.



(a) Silhouette segmentation. (b) Four body parameters.

Figure 2.14: Segmentation of the body silhouette into regions and 4D-walk vector $\mathbf{w} = < d_1, d_2, d_3, d_4 >$. (© 2003 IEEE. Reprinted, with permission, from A.F. Bobick & A.Y. Johnson, Gait recognition using static, activity-specific parameters, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Dec/2001)

Kellokumpu et al. (2009) partitioned the XYT cube into four regions through the centroid of the silhouette, thus separating the hands and the legs of the person. LBP features were calculated for each subvolume to encode the movement information, and their local histograms were concatenated to form a global feature histogram. They improved results in comparison to multi-resolution analysis on gait recognition using the Carnegie Mellon University (CMU) database (Gross & Shi 2001). Ran et al. (2010) used an iterative local curve embedding algorithm to decompose a video sequence into XT slices to generate periodic patterns referred to as double helical signatures (see Figure 2.15). These signatures encoded the appearance and kinematics of human motion and showed geometric symmetries. The system was able to highlight the body parts in challenging environments to classify load-carrying conditions by gait recognition.

In a recent study, Deng & Wang (2018) obtained gait features by using Microsoft Kinect v1.0. They proposed a new model-based gait recognition method by combining deterministic learning theory and the data stream of the Kinect. The gait was characterised by temporal features, kinematic features and their temporal changes. Both spatio–temporal and kinematic cues were fused on the decision level to improve the gait recognition performance. Deng & Wang (ibid.) reported 97.7% gait recognition accuracy after the algorithm was evaluated on a self-made Kinect gait dataset comprising of 5760 walk sequences collected with the help of 80 participants.

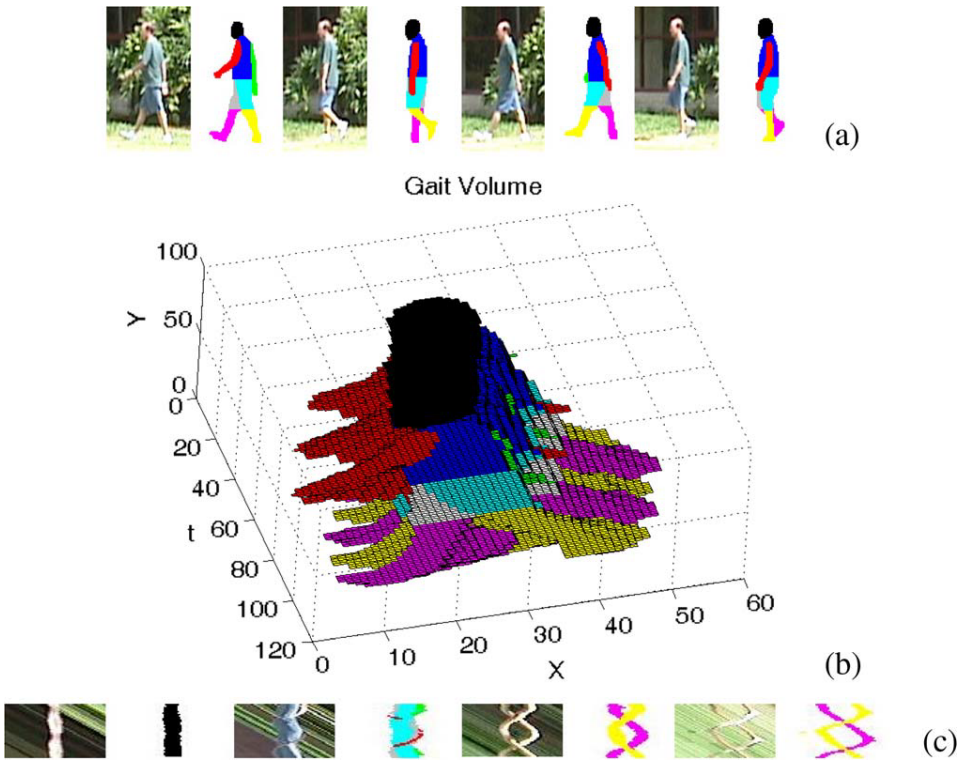


Figure 2.15: (Top) Original and silhouette images. (Middle) XYT volume (X is horizontal axis, y is vertical axis and t is into the XY-plane). (Bottom) Double Helical Signatures after slicing along XT plane. (© 2010 IEEE. Reprinted, with permission, from Yang Ran, Qinfen Zheng, Rama Chellappa and Thomas M. Strat, Applications of a Simple Characterization of Human Gait in Surveillance, IEEE Transactions on Systems, Man, and Cybernetics, Aug/2010)

Table 2.1: Timeline of gait detection and analysis techniques.

Authors	**Approach**	**Goal(s)**
Niyogi et al. (1994)	Image cube (XYT) by staking silhouettes from video frames Identify braided pattern in XT plane formed by walker's ankle	Human motion analysis
Davis & Bobick (1997)	Motion History Image to summarise the spatio-temporal motion	Action recognition
Cunado et al. (1997)	Thigh and calf motion analysis by frequency and phase of the movement	To classify gait components for the treatment of pathologically abnormal patients
Fujiyoshi & Lipton (1998)	Skeletonise the moving objects and analyse the cyclic motion of leg and torso segments	Human motion analysis
BenAbdelkader et al. (2001)	Extract silhouettes of key poses in a walking sequence and perform image self-similarity	Gait recognition
Lee & Grimson (2002)	Mean and standard deviation of amplitudes of leg, arm swing and head movement	Modelling the gait dynamics
Johnson & Bobick (2001)	Analysed four stride parameters, i) Height ii) Width iii) Distance between the head and pelvis and iv) Maximum distances between the pelvis and left/right foot	Multi-view method for gait recognition
BenAbdelkader et al. (2002*a*,*b*)	Stride length, cadence (walking frequency) and height features for gait characterization	Gait based human identification
Wang et al. (2003)	Temporal analysis of distances of silhouette boundary points from the centroid	Automatic gait recognition
Kale et al. (2002, 2003)	Train the HMM on the width vectors derived from the silhouette for several gait cycles	Gait recognition
Sarkar et al. (2005)	Observe the change in area of bottom half of the silhouette to extract gait cycle	Human identification
Han & Bhanu (2006)	Gait Energy Image (GEI) made of moving pixels between the frames	Human recognition from gait
Kellokumpu et al. (2009)	Partition the XYT cube and describe human gait by spatiotemporal LBP histograms	Gait recognition
Zhang et al. (2010)	Improved GEI with Active Energy Images (AEI). Focused on actively moving parts of a gait silhouette and reduced the effect of translational movements	Enhance dynamic gait features for gait recognition
Ran et al. (2010)	Decomposed a gait sequence into XT slices to generate periodic patterns which encoded the kinematics of human motion and showed geometric symmetries.	Detect load carrying events in surveillance videos
Nizami et al. (2010)	SVM trained on Independent Components of MSI to identify gait for moving subjects	Automatic gait recognition
Venkat & De Wilde (2011)	Partition a silhouette into overlapping upper, middle, lower, left and right parts. Bayesian network for gait identification.	Analysis of sub-gait characteristics to recognise gait
Wu et al. (2016)	Deep CNN trained on GEIs of the subjects at 11 different viewing angles. The network was tested on CASIA-B and OU-ISIR gait datasets	Gait based human identification
Nambiar (2017)	HOFEI gait signature described the relative motion of each body part with respect to the other, over a complete gait cycle.	Re-identification of pedestrians in surveillance videos

Deng & Wang (2018)	Gait characterisation by fusion of temporal features and microsoft kinematic features	Human identification
Zhang et al. (2019)	Binary silhouettes and GEIs were partitioned into horizontal parts and a CNN was trained for each part. The local features were concatenated to create a final descriptor	Gait recognition

Table 2.2: Limitations of current gait detection techniques

Authors	**Uncontrolled Environment**	**View Invariant**	**Dynamic Background**	**Multiple Class**	**Disabled Person Detection**
Niyogi et al. (1994)					
Davis & Bobick (1997)					
Cunado et al. (1997)					
Fujiyoshi & Lipton (1998)				✓	
BenAbdelkader et al. (2001)					
Lee & Grimson (2002)					
Johnson & Bobick (2001)	✓	✓			
BenAbdelkader et al. (2002*a*,*b*)	✓	✓		✓	
Wang et al. (2003)		✓			
Kale et al. (2002, 2003)		✓			
Sarkar et al. (2005)		✓			
Han & Bhanu (2006)					
Kellokumpu et al. (2009)					
Zhang et al. (2010)					
Ran et al. (2010)	✓	✓			
Nizami et al. (2010)					
Venkat & De Wilde (2011)					
Wu et al. (2016)		✓			
Nambiar (2017)		✓	✓		
Deng & Wang (2018)	✓	✓			
Zhang et al. (2019)	✓	✓			

2.4 Tracking and Data Association Algorithms

Object tracking plays a vital role in computer vision. It involves locating objects, maintaining their identities, and obtaining their trajectories given an input video or detections. Examples of tracked objects include pedestrians in surveillance videos (Paul et al. 2013), vehicles on the road (Mukhtar et al. 2015), sports players on the court (Yang & Ramanan 2012) and groups of animals (Spampinato et al. 2012). Pedestrians are typical nonrigid objects, and according to a survey (Luo et al. 2014), at least 70% of current multi-object tracking (MOT) research efforts target pedestrians for pose estimation (Andriluka et al. 2009), action recognition (Yang & Ramanan 2012), and behaviour analysis (Afsar et al. 2015). Compared with single object tracking, which uses appearance or motion models to cope with scale changes, illumination variations and out-of-plane rotations, MOT solves two additional tasks of determining the number of objects, which typically varies over time, and maintaining their identities. MOT also offers additional challenges including 1) initialization and termination of tracks, 2) similar appearance, 3) frequent occlusions and 4) interactions among multiple objects.

Generally, MOT can be linked with the data association problem which aims to associate detections across frames in a video sequence. Trackers use

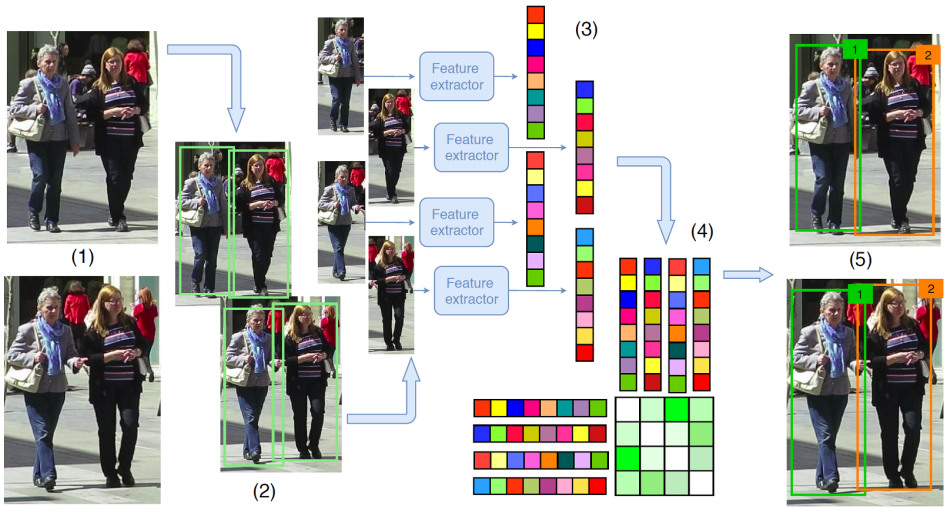


Figure 2.16: Usual workflow of a DBT algorithm: (1) Raw frames of a video, (2) An object detector outcome, (3) Features computed for every detected object, (4) An affinity computation step, (5) Outcome of data association to update object IDs. Reprinted from Neurocomputing, Vol 381, Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri and Francisco Herrera, Deep learning in video multi-object tracking: A survey, Pages No. 61-88, Copyright (2020), with permission from Elsevier

motion models (Dicle et al. 2013) and appearance (Kim et al. 2015) of target objects to aid the data association process. Depending on object initialization, most existing MOT works are classified into two groups: Detection-based tracking (DBT) and detection-free tracking (DFT). In DBT (see Figure 2.16), objects are first detected and then connected into trajectories. Given a video sequence, each frame is subjected to type-specific object detection (Mukhtar et al. 2018, Ren et al. 2015) to obtain object hypotheses, then (sequential or batch) tracking links detection hypotheses into trajectories. Therefore, the performance of DBT relies on the accuracy of the employed object detector. On the other hand, DFT requires manual initialisation of object trajectories in the first frame and then it localizes those objects in successive frames. It can not deal with the object disappearing as it has no pre-trained object detectors. MOT can also be divided into online tracking and offline tracking, depending on the use of future observations when processing the current frame. Online tracking methods (Bewley et al. 2016) only rely on the past information available up to the current frame, while offline methods employ observations both in the past and in the future.

Most online tracking methods make use of appearance features of either the individual objects themselves (Yang & Jia 2016) or at a global level (Bewley et al. 2014, Choi 2015) through online learning. In addition to appearance

models, motion is often incorporated to assist associating detections to tracks (Bewley et al. 2014, Choi 2015). Geiger et al. (2013) proposed the use of the Hungarian algorithm (Kuhn 1955) in a two-stage process. First, tracklets were created by associating detections across adjacent frames where both geometry and appearance features were combined to construct an affinity matrix. Then, tracklets were associated with each other to link broken trajectories caused by occlusion. This two-step association method restricts this approach to batch computation.

There are sophisticated data association techniques, including Multiple Hypothesis Tracking (MHT) (Reid 1979) and Joint Probabilistic Data Association (JPDA) (Hamid Rezatofighi et al. 2015) which are capable of handling MOT scenarios. The complexity of these approaches is exponential in the number of tracked objects, making them impractical for realtime applications in highly dynamic environments. Hamid Rezatofighi et al. (2015), revisited the JPDA formulation (Bar-Shalom et al. 2011) in visual MOT for time-efficient approximation by exploiting developments in solving integer programs. Similarly, Kim et al. (2015) used an appearance model for each target to prune the MHT graph to achieve state-of-the-art performance. However, these methods still delay the decision making due to the trade-off between accuracy and speed, as demonstrated in a performance comparison (Bewley et al. 2016) of several baseline trackers on a common MOT Benchmark.

Bewley et al. (2016) proposed SORT as part of a DBT framework for the MOT problem. The first module was a CNN based object detector providing bounding box information for objects detected in each frame. In the second phase, SORT performed online tracking where only detections from the previous and the current frame were presented to the tracker. The Kalman filter (Kalman 1960) was used to predict locations of the active objects and detections were assigned to those tracks (predictions) with minimum assignment cost determined by the Hungarian algorithm. A strong emphasis was on efficiency to facilitate real-time tracking and to promote greater uptake in applications. It used only the bounding box position and size were used for both motion estimation and data association and ignored appearance features in tracking.

SORT has a limited object counting mechanism, especially when objects accelerate, causing surplus counts as cost matrix ignores non-overlapping objects and generates new labels for existing objects. Furthermore, issues regarding short-term and long-term occlusion were also ignored by arguing that object re-identification adds significant overhead into the tracking framework. In subsequent work, the authors acknowledged these problems and proposed an improvement to the data association (Wojke et al. 2017). They added

a CNN pre-trained on the pedestrian class to reduce the number of identity switches by 45%. Apart from complicated data association, the improved version (deepSORT) is limited to tracking single class objects, namely pedestrians only.

In recent years, researchers have demonstrated that CNN based object detectors can aid in solving DBT problems and data association (Ciaparrone et al. 2020). Deep learning models are preferred in the feature extraction phase due to their ability to extract meaningful high-level features. Siamese CNNs (see Figure 2.17) with contrastive loss functions are also an alternative to classical CNN models, for extracting the set of features that best distinguish between subjects (Kim et al. 2016). Some studies used deep learning models to improve the association process performed by classical algorithms based on the Hungarian algorithm (Kuhn 1955) and a set of rules to manage the tracking status (e.g. to create or terminate a track). Milan et al. (2017), used recurrent neural networks (RNN) to predict the probability of the existence of a track in each frame, thus helping to identify live and decaying tracks. Ma et al. (2018) used a bidirectional gated recurrent unit (GRU) RNN to decide when to split tracklets. The algorithm was divided into three steps: tracklet generation step, tracklet cleaving step, and a tracklet reconnection step employing a Siamese bidirectional GRU. The gaps within the newly-formed tracklets were then filled with polynomial curve fitting.

Bergmann et al. (2019) noted that several CNN-based detection algorithms (Ren et al. 2015, Yang et al. 2016) involve some form of bounding box refinement through regression which can perform MOT tasks. They showed that the bounding box regressor of a trained Faster R-CNN detector could handle most tracking scenarios in the existing benchmarks. The identity of targets was automatically transferred from the previous to the regressed bounding box, effectively creating a trajectory. Two motion models and a re-identification (reID) step were combined for preserving the identities of the subjects across frames. First motion model was based on the assumption that targets exhibit a small movement between frames, while the second model was to deal with large camera movements and low video frame rates. The reID module generated appearance features by a Siamese neural network (Bromley et al. 1994, Ristani & Tomasi 2018) trained on tracking ground truth data. To reduce false reIDs, pairs of deactivated and new bounding boxes with a high IOU score were considered. The proposed system achieved state-of-art performance on several challenging tracking scenarios.

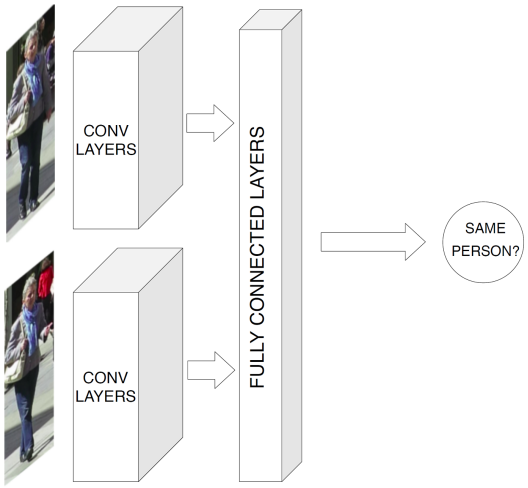


Figure 2.17: Example of a Siamese CNN architecture. Reprinted from Neurocomputing, Vol 381, Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri and Francisco Herrera, Deep learning in video multi-object tracking: A survey, Pages No. 61-88, Copyright (2020), with permission from Elsevier

2.5 Existing Datasets

Several datasets have been collected from different scenarios and made publicly available to benchmark human detection algorithms. General-purpose person detection datasets are listed in Table 2.3. These datasets complement each other by covering variations in human appearance, pose, viewpoint and occlusion. They also cover changes induced due to an uncontrolled environment such as background clutter and varying illumination conditions. Occlusions can occur in the datasets due to non-human objects, the interaction between humans, image borders and viewing angle. Imaging device, image/video resolution and compression formats can also affect the quality of a dataset. The size of pedestrians appearing in images depends on the camera resolution and its distance from pedestrians.

INRIA, Caltech, ETH and TUD-Brussels are popular among the computer vision researchers to measure the accuracy of their newly developed pedestrian detection systems. Most datasets contain annotations and provide bounding box information which represents the location of pedestrians in all training and test images. Annotations can be stored in different formats; for example, CAVIAR dataset uses the XML format which is portable and offers flexibility for the extension, while the TUD, ETH, INRIA, and Penn-Fudan datasets use the PASCAL format (Everingham et al. 2010). The detection results (bounding boxes) of several pedestrian detection methods and their performance evaluation code is available at Caltech webpage [1].

[1] `http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/`

Table 2.3: List of publicly available person detection datasets.

Name	No. of Pedestrians	Median Height of Pedestrians (in pixels)	Outdoor Samples	Background Clutter	Occlusion
MIT (Papageorgiou & Poggio 2000)	924	128			
INRIA (Dalal 2006)	1774	279	✓	✓	✓
Penn–Fudan (Wang et al. 2007)	345	-	✓	✓	
Caltech (Dollar et al. 2011)	347,000	48	✓	✓	✓
ETH (Ess et al. 2008)	15,388	90	✓	✓	✓
CAVIAR (*CAVIAR dataset 1* 2004)	5614	-			✓
GM-ATCI (Silberstein et al. 2014)	200,000	-	✓	✓	✓
TUD-Brussels (Wojek et al. 2009)	3274	66	✓	✓	✓
CVC-01 (Gerónimo et al. 2007)	1000	83	✓	✓	
Daimler (Enzweiler & Gavrila 2008)	72,052	96	✓	✓	✓
ImageNet (Russakovsky et al. 2015)	1431 (images)	-	✓	✓	✓

In addition to the datasets listed in Table 2.3, there are plenty of other publicly available datasets for pedestrian detection and among them, only a few are appropriate for gait analysis. Gait identification requires multiple-shot image sequences with complete gait cycles which can capture spatio-temporal information of the walking pattern. The nature of the background, the viewpoint and the gait direction in videos have a significant influence on the performance of gait analysis algorithms. Here we survey some commonly used datasets involving gait and enlisted them in Table 2.4.

AVAMVG: The AVA Multi-View Dataset for Gait Recognition (López-Fernández et al. 2014) contains data of 20 people walking along ten trajectories each. Six calibrated cameras captured the database images with a resolution of 640×480 pixels at 25fps from different views angles. The binary silhouettes of each video sequence are also available. Castro et al. (2014) and López-Fernández et al. (2016) used the AVAMVG dataset to recognise the human gait in video sequences with a controlled environment. López-Fernández et al. (2016) studied the variation in the pedestrian's gait who walk along a curved path.

CASIA: is comprised of four different datasets, but dataset A and B are popular because of their practical data acquisition mode (*CASIA Gait Database* 2005). Dataset A contains data of 20 participants walking at three different angles ($0°, 45°$ and $90°$) to the camera view in an outdoor environment. On the other hand, dataset B contains 13640 indoor gait samples of 124 people captured by a network of 11 cameras. Many gait recognition works (Nizami et al. 2010, Wu et al. 2016) have used dataset B since it captures the pose from 11 different viewing angles separated by $18°$.

HDA Person Dataset: HDA is a diverse multi-view video dataset dedicated for performance evaluation of video surveillance algorithms (Nambiar 2017). It contains 30 minutes long fully annotated image sequences obtained using 13 indoor cameras. Some gait based pedestrian detection works (Nambiar 2017, Wang et al. 2016) have reported test results on HDA datasets.

i-LIDS: The Imagery Library for Intelligent Detection Systems is the U.K. government's benchmarking dataset for video analytics systems. The object tracking dataset comprises of CCTV footages of 119 people at 25fps using five cameras installed inside a busy hall. The dataset is challenging due to occlusion scenarios and significant illumination changes. Wang et al. (2016) evaluated their gait based reID system on i-LIDS dataset.

KinectREID: This dataset is useful for benchmarking the pedestrian detection algorithms that use features obtained through the Kinect sensor: gait, anthropometry, skeleton features and RGB-D data (Pala et al. 2015). It has several indoor videos (of 71 people) of varying illumination conditions, viewing angles and appearance (bags and accessories).

OU-ISIR: Iwama et al. (2012) describe it as the world's largest gait dataset available for performance evaluation of vision-based gait recognition. Each participant walked 10m in front of two cameras, which were installed at a distance of 4m from the walking course. A total of 4007 participants (2135 males and 1872 females) with ages ranging from 1 to 94 years, took part in the study. The entire experiment was conducted indoors in a controlled environment with no variation in the background, walking and light conditions. Iwama et al. (2012) carried out a performance comparison of gait recognition using state-of-the-art gait features. In addition, they studied variation in gait recognition features with a change of gender and age group.

PETS2009: This is a common pedestrian tracking dataset that captures different crowd activities in an outdoor environment by a multi-camera system (Ferryman & Shahrokni 2009). Bouchrika et al. (2016) evaluated their gait based reID system on PETS2009 dataset since it offers multi-view sequences.

SOTON: The aim of collecting collect SOTON database (Shutler et al. 2004) was to support the development of new technologies for recognising pedestrians at a distance. It contains a small and large dataset. The small dataset was collected with indoor scenarios, and it contains only 12 participants with different clothing, accessories and walking speeds. The large dataset consists of 114 participants and over 5000 samples collected in the indoor and outdoor environments. Cunado et al. (1997), Nixon & Carter (2006), Nizami et al. (2010) have used SOTON dataset for gait recognition.

TUM-GAID: This dataset contains audio, video and depth information in the data recorded from 305 participants in three variations. The database is freely available for multimodal gait recognition (Hofmann et al. 2014) and

Table 2.4: List of publicly available gait recognition datasets.

Name	Participants	Scenarios	Multiview	No. of Cameras	Resolution
AVAMVG	20	Indoor	✓	6	640×480
CASIA-B	124	Indoor	✓	11	210×105
HDA	85	Indoor	✓	13	2560×1600 (max)
i-LIDS	119	Outdoor	✓	5	576×704
KinectREID	71	Indoor	✓	1	-
OU-ISIR	4007	Indoor		2	640×480
PETS2009	-	Outdoor	✓	8	768×576
SOTON (small)	6	Indoor		1	384×288
SOTON (large)	114	Outdoor		6	720×576
TUM-GAID	305	Indoor	✓	1	640×480
USF	33	Outdoor	✓	2	720×480
Vislab KS20	20	Indoor	✓	1	-

human identification (Geiger et al. 2014).

USF: The USF database contains 1870 sequences acquired from 33 subjects over four days (Sarkar et al. 2005). Some of the studies using the USF dataset are Kale et al. (2003), Han & Bhanu (2006) and Wang et al. (2011).

Vislab KS20: This dataset comprises of multi-view Kinect skeleton (KS) data sequences collected from 20 walking people using Kinect v2 (Nambiar 2017). Altogether it contains 300 skeleton image sequences of 20 people walking in five different directions $(0°, 30°, 90°, 130°, 180°)$.

There are several other datasets available for gait analysis, but only a handful was described in this section. A larger list of existing gait datasets is provided by Lee et al. (2013). The information about datasets may change over time due to publishers updating the contents from time to time. We observe that gait datasets often capture pedestrians by placing cameras at half body height, therefore, missing the surveillance factor. Alongside the raw data, some datasets offer binary silhouettes which are often needed for gait analysis and otherwise difficult to extract in real-world scenarios.

Chapter 3

Database Formation

Our custom dataset was collected due to the absence of image databases containing mobility aids (wheelchair, crutch, walking stick, walking frame and mobility scooter) required to train an automated system for recognising mobility aids. We think that a CNN based object detector trained on an image dataset of mobility aid users can identify disabled pedestrians in uncontrolled scenarios. The database has two parts; Image dataset and a Video dataset.

3.1 Image Dataset

There is no publicly available database suitable for training a CNN to recognise mobility aids. Therefore, we sourced images from ImageNet, Google Images and INRIA's pedestrian database to create a custom dataset. This image dataset has a total of eight classes inclusive of five mobility aids, pedestrians and the rest (car and bicycle) act as distractors having structures similar to those in wheelchair and mobility scooters. List of classes is provided in Table 3.1. Five mobility aids are chosen in this study, and they make up 91% of all the visible mobility aids that were publicly sighted during a study in New Zealand (Burdett 2013). The rest includes guide dogs, personal assistant and an artificial limb. From here onwards, we shall be referring to the pedestrians, mobility aids, car and bicycle as 'objects'.

The majority of images sourced were not labelled with ground truth. Thus, we manually drew bounding boxes to capture objects' size and location in the source images. The assembled database contains 5819 images, of which 4653 images (6715 bounding boxes) were labelled manually. Two PhD students at the school of engineering, University of Waikato, manually annotated the database images. Image labelling was a time consuming and tedious task as it involved drawing boxes around multiple objects in an image for 4000+ images. The image labels were later visually inspected for correctness by drawing bounding boxes on their respective images. In total, we spent more

than 320 hours labelling 6715 bounding boxes in 4653 images and manually reviewing them afterwards. The labelling for database images is encoded in Bounding Boxes (BBs) associated with a unique class identifier (ID). Such BBs are labelled objects from each class in the images and known as the Ground Truth information during this study. We used MATLAB as a software tool during the labelling process and defined a list of labelling rules:

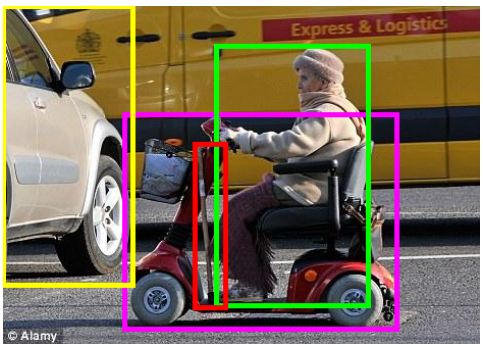
- Each BB is drawn so that it completely and tightly encloses the object.
- If an object is partially occluded, the BB is drawn estimating the whole body extent.
- When an objects extends beyond the image borders, BB's are cropped to the image boundaries.
- A unique ID is associated with each class as shown in Table 3.1.

Table 3.1: Details of custom built mobility aids dataset

Class	**ID**	**No. of BBs**	**Median Height**
Wheelchair	0	1672	155
Crutch	1	1266	240
Walking Frame	2	671	456
Walking Stick	3	1119	231
Mobility Scooter	4	699	464
Car	5	1562	140
Person	6	8209	154
Bicycle	7	983	187
	Total	16181	180

The object labels and bounding box locations were visually checked one by one by drawing bounding boxes on top of database images using a computer program. Once annotated, the bounding box information (location and size) for all labelled objects in an image, was saved in a label (.txt) file. Table 3.1 presents the statistical characterization of the database, while Figure 3.1 shows a few examples of image labelling. Our dataset also covers a vast range of objects' BB heights: from 7 to 2651 pixels.

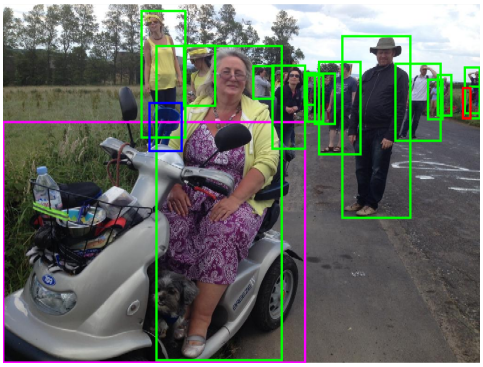
Our dataset is designed to contain images of mobility aids with and without persons using them. We observed that a disabled person (the mobility aid user) often appears alongside the mobility aid in the images. This overlap complicates the process of bounding box annotation for training images causing the same object to appear in both bounding boxes. However, this depicts the real-world situations where the disabled person or mobility aid often occlude each other's appearance.



(a) Labelling example 1

```
4 0.541667 0.640483 0.570513 0.634441
6 0.607906 0.504532 0.318376 0.767372
5 0.137821 0.415408 0.271368 0.824773
3 0.435897 0.652568 0.064103 0.483384
```

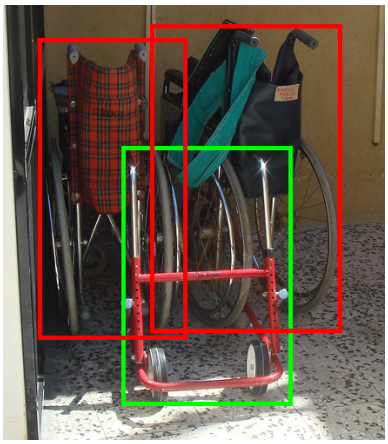
(b) Example 1 ground truth



(c) Labelling example 2

```
4 0.317383 0.664714 0.632813 0.670573
6 0.336914 0.195964 0.091797 0.352865
6 0.412598 0.199870 0.065430 0.167969
6 0.453613 0.555339 0.262695 0.876302
6 0.546387 0.218750 0.041992 0.091146
6 0.599609 0.287760 0.068359 0.231771
6 0.654785 0.263672 0.034180 0.128906
6 0.675293 0.264323 0.041992 0.140625
6 0.706543 0.290365 0.086914 0.257813
6 0.643555 0.236979 0.021484 0.088542
6 0.783203 0.343099 0.142578 0.506510
6 0.870605 0.274089 0.094727 0.214844
6 0.906738 0.266276 0.030273 0.110677
6 0.926270 0.254557 0.028320 0.113281
6 0.984375 0.240885 0.031250 0.127604
6 0.989746 0.274740 0.020508 0.088542
3 0.972656 0.276693 0.015625 0.084635
1 0.341309 0.343099 0.067383 0.134115
```

(d) Example 2 ground truth



(e) Labelling example 3

```
2 0.535308 0.624000 0.437358 0.588000
0 0.288155 0.424000 0.380410 0.684000
0 0.637813 0.400000 0.492027 0.700000
```

(f) Example 3 ground truth

Figure 3.1: BB labelling and ground truth examples

All the images are in .jpeg file format and there is label file (.txt file) for each image with a line for each ground truth object in the image that looks like: $[object\text{-}class(class\ \ ID), x, y, width, height]$. Where centre coordinates (x, y), width, and height are relative to the image's width and height.

3.2 Video Dataset

In addition to the image dataset, an outdoor surveillance video data set was developed to test the system's performance. These surveillance videos were either provided by Stantec NZ or collected with the help of two cameras installed at the congregation place of disabled and older people. Approximately 32 people participated in the data collection, most of them appearing in more than one camera. Permission to share the video data publicly is limited to parties specified in Section 2(g) of the Ethics Approval Document. It is mandatory for pedestrian faces to be blurred or blacked out in reporting results to preserve anonymity.

3.2.1 Video set 1

Stantec NZ helped us in acquiring the surveillance videos from different parts of Hamilton City. The combined length of those videos is 266 minutes, and their details are provided in Table 3.2. These videos depict real-life scenarios and captured healthy pedestrians and mobility aid users as they appear. This dataset was only used to verify the counts of healthy and disabled pedestrians, and results have been reported in chapter 6. Videos were processed by the Stantec's provided computer at their premises. These videos provided counts of people who were in the video, and the system estimated the number of counts using the proposed algorithm. We believe that location and infrastructure are vital in dictating the frequency of disabled people showing up in a camera feed. For example, a place near a hospital or retirement village is likely to host more mobility aid users than city centre or near sports arenas.

3.2.2 Video set 2

In a separate round of video collection experiments, Stroke Foundation NZ helped us to get access to the participants (mobility aid users and older people) at their monthly congregation place. Ethics approval was granted by the Human Research Ethics Sub-committee at the Faculty of Science and Engineering, University of Waikato and we followed the procedures outlined in the ethics approval. Informed written consent was obtained from participants after a briefing on the project and the need for video data. See Appendix B for a sample of the consent form for participants.

Videos were collected with the help of two SOYAL LIZM40S200 outdoor HD (1920×1080) cameras installed near the entrance of the Hamilton Marist Rugby Football Club. To obtain the surveillance angle and outdoor environment, cameras were mounted at the height of 10ft on pillars at the hall entrance (see Figure 3.2). Table 3.2 summarises the details of the video recordings.

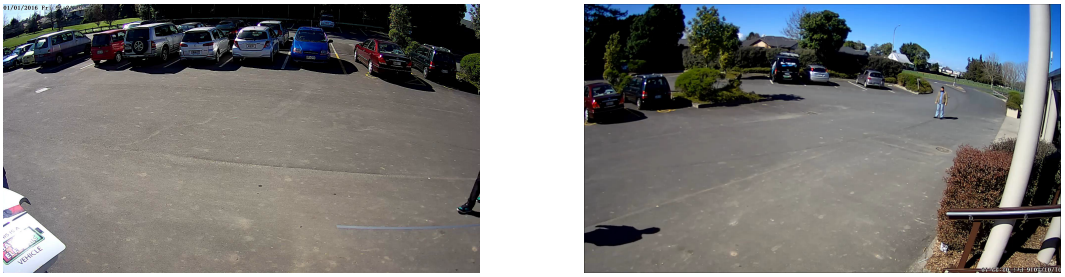
Table 3.2: Video sets 1, 2 and 3

	Set 1	**Set 2**	**Set 3**
No. of videos	24	84	13
Collection Year	2013, 2014, 2017	2016, 2017	2017
No. of participants		22	10
Resolution	various[1]	1920 × 1080	1920 × 1080
Total duration	4h 26m 17s	18m 39s	37m 22s
Total pedestrian appearances	2976	173	353
Mobility Aids appearances	11	40	69
Wheelchair	6	14	22
Crutch	1	0	18
Walking Frame	0	5	0
Walking Stick	4	21	16
Mobility Scooter	0	0	13

[1] 1920 × 1080, 768 × 576, 720 × 480, 616 × 462, 474 × 356, 387 × 288

3.2.3 Video set 3

These videos were recorded within the campus with mobility aids borrowed from the medical centre and participants mocking the movements by mobility aid users. Two SOYAL LIZM40S200 cameras captured pedestrians in front on and side on views so that the videos can be used for gait analysis and CNN based mobility aid detection system. Figure 3.3 shows the position where one of the surveillance cameras was installed and Figure 3.4 displays one frame for each camera. Table 3.2 displays the quantitative information about the set 3 videos.



(a) View 1 (b) View 2

Figure 3.2: Snapshots of the sequences (set 2)

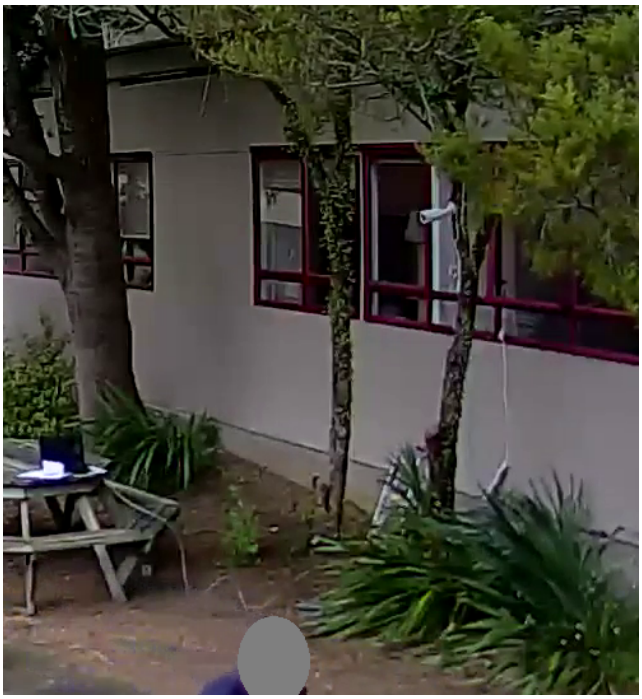
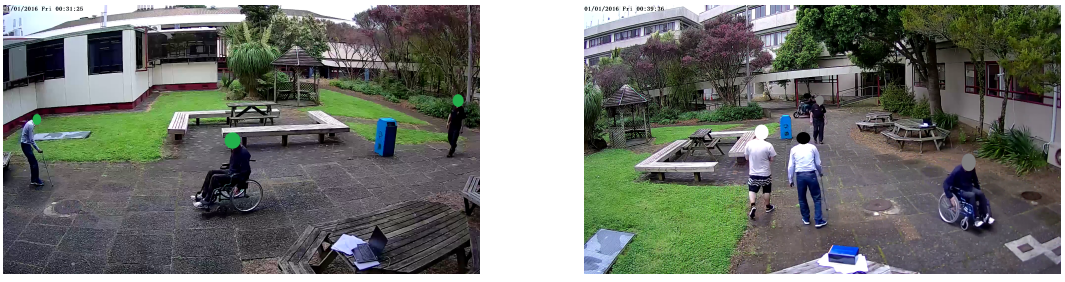


Figure 3.3: Camera setup to record movements



(a) View 1 (b) View 2

Figure 3.4: Snapshots of the sequences (set 3)

Chapter 4

Gait Analysis of Healthy and Disabled Pedestrians

Over the last two decades, many gait analysis techniques have been proposed for person identification. A number of factors that may affect the gait of a pedestrian, including gender (Hu et al. 2011), injuries to body parts, whether they are carrying a heavy load (Choudhury & Tjahjadi 2015) and potentially their mood. The majority of gait analysis approaches described in section 2.3 targeted recognition of humans in the video, action recognition and gender classification but lack application towards disabled person identification. The lateral walking view has been a preferred choice of researchers because it captures gait information better than in the frontal view (Lee et al. 2013). The lateral view captures considerable variation in the gait due to legs and hands extending to their maximum distances during the walk. Self-occlusion and perspective distortion are also minimal in lateral views. Motion from a plane perpendicular to this, the front on view, is a special case as silhouette variations are smaller. While it is useful to devise gait recognition algorithms that work in all environments, the algorithm design complications prevent us from considering the frontal view. Therefore our automated gait recognition is limited to analysing pedestrians with the side-on view.

A competent observer can pick a disabled person by appearance (if a visible mobility aid is in use) or by observing gait (in the case of a limp). Even without 3D information, an observer can also identify a disabled person by monitoring the subject's movement in a monocular video of human walk. This chapter focuses on identifying distinguishing features in human gait, which hopefully can enable automated detection of disabled persons from videos. There is a good chance that a time series analysis of body part motion will depict abnormality sensed by the human cognition. During this study, we examined the temporal behaviour of head and leg positions of pedestrians by extending the gait analysis technique proposed by Fujiyoshi et al. (2004). The authors

(ibid.) used the star skeletonisation technique to extract and analyse gait signals for leading and trailing legs only, but this obfuscates the action of a single leg.

For consistency, we name legs of pedestrian as *Leg 1* and *Leg 2* which represent the front and the rear leg during a side-on walking view, respectively. The left and right legs terminology is avoided as those depend on which way the person is walking. For the purpose of this thesis, we require the gait signal for separated legs, i.e., Leg 1 and Leg 2. Therefore, the joint gait signal produced by legs was separated into two, one each corresponding to a leg. The signal for leading (and trailing) leg is actually a combination of the leg 1 and the leg 2 signals at different times whereas we would like to split it up into the leg 1 and the leg 2 signals. So half the leading leg and half of the trailing leg makes up the leg 1 and opposite way around to make the leg 2. The new set of gait signals can better illustrate the contribution of legs in human walking style. In the later part of this chapter, we manually marked the dynamic parts of the walker in surveillance videos to extract the gait signal for identifying any discerning features to detect a disabled person. Our main contributions are as follows:

- Improvement of the gait extraction technique (Fujiyoshi et al. 2004) to generate gait signatures for individual legs. For indoor videos, our gait extraction technique does not require binary silhouettes, instead computing them using the Gaussian Mixture Model (GMM).
- Answering the question "is there a discriminating feature in the gait signal that can help to categorise a disabled person from healthy?"

4.1 Automated Approach

The topic of disability being less investigated means that there is much less training and testing video data available on the internet. Surveillance videos capture a limited number of disabled pedestrians since the majority of these people tend to stay at home, and approximately 1% (Burdett 2015) of New Zealand's population uses a mobility aid for travel at any particular time. Therefore, a dedicated video dataset was recorded that involved healthy and disable person movements with the same camera angle and background. Details of video datasets are provided in Section 3.2.

This section investigates the possibility of an automated approach to obtaining gait signals which were manually extracted in the previous section. The automated task was divided into silhouette extraction, skeletonisation and

gait analysis phases. Figure 4.1 shows the pipeline of our proposed algorithm, which is detailed in the following sections.

Figure 4.1: Proposed pipeline for gait analysis

4.1.1 Silhouette Extraction

It can be difficult to analyse the nature of human motion without extracting the silhouette motion. A human silhouette is a two-dimensional, solid shaped and monocular cutout image with its edges matching the outline of a person. A static background or pre-extracted silhouettes are often preferred as a first step towards building a gait analysis algorithms. CASIA-B dataset (*CASIA Gait Database* 2005) was used for an indoor set of experiments, and silhouettes were provided as part of the database. In outdoor videos where the camera and the background are static, the pedestrian silhouettes were extracted using a person detector (YOLO) and foreground extraction (Stauffer & Grimson 1999). YOLO is responsible for identifying pedestrians in a video frame while foreground subtraction is needed to crop the silhouette out from the YOLO's bounding box.

YOLO is a general-purpose object detector, and a pre-trained model is available at the author's website[1]. YOLO takes a video frame as input and after processing, calculates five predictions associated with an object's bounding boxes: centre of the object (x,y), width (w), height (h), and confidence value. Inside YOLO, a CNN performs the detection task with its initial convolutional layers extracting features from the image while fully connected layers predicted class probabilities and object coordinates. YOLO's code was tweaked to suppress detections for all classes except the person class and to store the bounding box information, frame number and confidence values to a local text file. Figure 4.2a shows an example detection by YOLO. The resulting bounding box serves as the region of interest (ROI) to be segmented into human silhouette and background. Human detection by YOLO filters out a significant portion of the image that contains non-human objects, thus narrowing down the search window for foreground segmentation by GMM.

GMM (Stauffer & Grimson 1999) is a multi-valued background model able to cope with multiple background objects. This model is capable of handling

[1] `https://pjreddie.com/darknet/yolov2/`

moving backgrounds in outdoor scenarios like tree leaf movement, rain, snow or sea wave movement. The probability of observing a certain pixel s value at time t is,

$$P(\mathbf{I}_{s,t}) = \sum_{i=1}^{K} w_i(s,t)\ N(\mu_i(s,t), \sigma_i(s,t))\ , \tag{4.1}$$

where $N(\mu_i(s,t), \sigma_i(s,t))$ is the i^{th} Gaussian model and $w_i(s,t)$ is its weight. K is the number of Gaussian distributions (usually between three and five) deemed to describe only one of the observable background objects. At each frame time t, the best matching distribution is updated with the new observed value $I_{s,t}$ and model parameters are updated as follows,

$$w_i(s,t) = (1-\alpha)\, w_i(s,t-1)\ +\ \alpha\ , \tag{4.2}$$

$$\mu_i(s,t) = (1-\rho)\, \mu_i(s,t-1) +\ \rho\, \mathbf{I}_{s,t}\ , \tag{4.3}$$

$$\sigma_i^2(s,t) = (1-\rho)\, \sigma_i^2(s,t-1)\ +\ \rho\, d(\mathbf{I}_{s,t}, \mu_i(s,t))\ , \tag{4.4}$$

where α is a user-defined learning rate, ρ is a second learning rate and d is the distance between the current image and background model $\mathbf{B}$ defined as,

$$d = |\mathbf{I}_{s,t}^R - \mathbf{B}_{s,t}^R| + |\mathbf{I}_{s,t}^G - \mathbf{B}_{s,t}^G| + |\mathbf{I}_{s,t}^B - \mathbf{B}_{s,t}^B|. \tag{4.5}$$

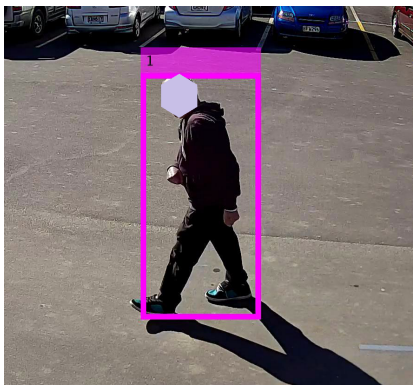
Parameters μ and σ of unmatched distributions remain the same while their weight is reduced to $(1-\alpha)\, w_i(s,t-1)$ to achieve decay. Whenever no component matches a colour $\mathbf{I}_{s,t}$, the one with the lowest weight is replaced by a Gaussian with mean $\mathbf{I}_{s,t}$, a large initial variance and a small weight. Once every Gaussian has been updated, the K distributions are ordered by weight and only the H most reliable ones are chosen as part of the background namely,

$$H = \arg\min_{h} \sum_{i=1}^{h} w_i > \tau, \tag{4.6}$$

where τ is an assigned threshold. Those pixels whose colour value is located outside of 2.5 standard deviations of every H distribution are considered as foreground pixels and are labelled "in motion". Finally, the pedestrian mask is obtained by cropping the background-subtracted image for each bounding box locations predicted by YOLO (shown in Figure 4.2b).

4.1.2 Skeletonisation and Gait Signal

The next step is to localise the head and legs position in order to observe the gait signal. The skeletonisation approach suggested by Fujiyoshi et al. (2004) can detect, segment and skeletonise the moving targets to locate head, centroid and legs position in videos with static and simplified backgrounds. The authors (ibid.) calculated two gait signals for legs based on their leading



(a) Pedestrian detection using YOLO

(b) Pedestrian mask after GMM+YOLO operation

Figure 4.2: Silhouette extraction using YOLO and GMM

or trailing status and classified moving targets as human and non-human based on their gait signatures.

We adopted the above mentioned gait extraction technique because it is independent of object size and unlike model based gait approaches, this method does not require a prior human model. Since our goal is to differentiate between healthy and disabled pedestrians among the pedestrian class therefore a YOLO implementation preceded the GMM operation for localising pedestrians. We also enhanced the original technique so it can extract a dedicated gait signal for each leg which is needed to spot a variation between two legs in case of a disabled pedestrian (see details in subsection 4.1.3).

The star skeletonisation of the pedestrian can be performed by following the steps listed below.

1. Clean the pedestrian mask by applying image morphological operations. Extract the outline of the resulting silhouette using a border following algorithm (see Figure 4.3a).

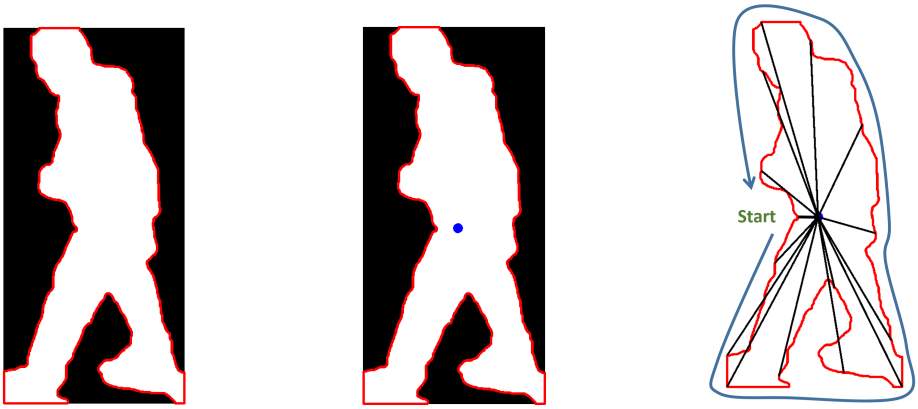
2. Estimate the centroid (x_c, y_c) of the pedestrian mask by calculating,

$$x_c = \frac{1}{N_b} \sum_{k=1}^{N_b} x_k \tag{4.7}$$

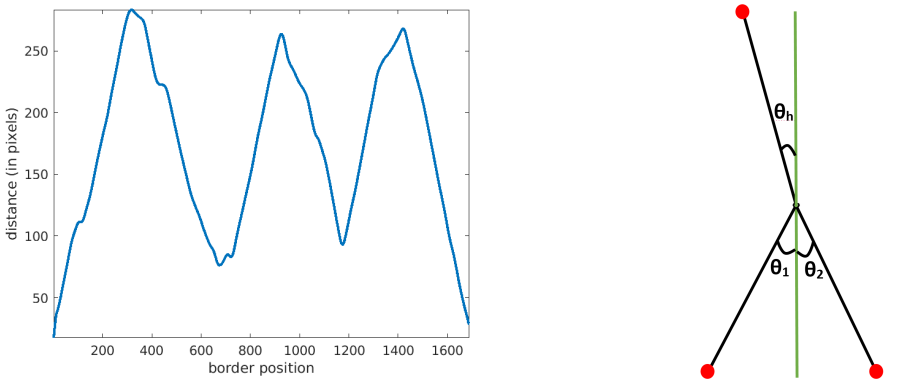
and

$$y_c = \frac{1}{N_b} \sum_{k=1}^{N_b} y_k \ , \tag{4.8}$$

where N_b is the total number of boundary pixels and (x_k, y_k) is a pixel on the boundary of the object. Figure 4.3b shows the location of the centroid in the silhouette.



(a) Border Extraction (b) Centroid location (c) Distance from centroid



(d) Distance signal plot versus border position (e) Skeleton model

Figure 4.3: Star skeletonization of a detected pedestrian.

3. Calculate the distance d_k of border points (x_k, y_k) from the centroid (x_c, y_c) by

$$d_k = \sqrt{(x_k - x_c)^2 + (y_k - y_c)^2}\ . \tag{4.9}$$

The distance signal is a one dimensional vector and it is plotted versus the border location on the horizontal axis (see Figure 4.3c and Figure 4.3d).

4. Find the local maxima in the distance signal. The location of maxima indicates the extremal points of the silhouette, which represent the location of a head and two legs (Figure 4.3e).

A skeleton with two legs and a torso was formed by connecting extremal points to the centroid of silhouette. To analyse the gait, legs and torso angles (Figure 4.3c) with the vertical axis can also be computed and analysed as time-series data. The following gait features in the temporal domain were also observed in this research:

- Leg position and velocity.
- Head position and velocity.
- Leg and head angle with respect to the vertical axis (θ_1 and θ_2).

4.1.3 Gait Signal for Legs Movement

Fujiyoshi et al. (2004) observed that both legs contributed to the leg angle signal in an alternating fashion based on leading and trailing status. In one complete walking cycle, a leg can change its status from leading to trailing and vice versa. Such a gait signal can classify between human and non-human objects but lacks a continuous stream of data corresponding to each leg to monitor its movement patterns. Therefore, the leading and trailing set of signals need to be broken down into gait signals for individual legs (referred to as leg 1 and leg 2). These splintered signals improved indicators for anomalous walking behaviour. The process of signal split (for the target moving from right to left of the screen) is summarised as:

1. Extract leg angle data θ_1 and θ_2 as shown in Figure 4.4. Here, $\theta_1(t)$ and $\theta_2(t)$ refer to leading and trailing leg angles, respectively, both measured from the vertical axis, are given by,

$$\theta_1 = \arctan\left(\frac{DE}{CE}\right) \tag{4.10}$$

$$\theta_2 = \arctan\left(\frac{EF}{CE}\right) \tag{4.11}$$

and

$$\theta_h = \arctan\left(\frac{AB}{BC}\right) \tag{4.12}$$

The plot of $\theta_1(t)$ in Figure 4.5 resembles the shape of $|\sin(t)|$ plot.

2. Identify time instants ($t = t_n$, where $n = 1, 2, 3, ..., N$) when both legs overlap and appear inline with the vertical axis. This occurs at the local minima of $|\theta_1(t)|$ or $|\theta_2(t)|$ so, define a rectangular wave $S(t)$ as,

$$S(t) = u(t) \; + 2\sum_{n=1}^{N}(-1)^n \, u(t-n) \; , \tag{4.13}$$

where $u(t)$ is the unit step function.

3. Calculate new signals α_1 and $\alpha_2$as,

$$\alpha_1(t) = \theta_1(t)S(t) \; + \; \theta_2(t)(\neg S(t)), \tag{4.14}$$

and

$$\alpha_2(t) = \theta_2(t)S(t) \; + \; \theta_1(t)(\neg S(t)). \tag{4.15}$$

The above methodology is similar to converting the absolute value of a sine to sine itself.

4. Track the extremal points in Figure 4.3e to generate the gait signals for head and leg movements. The trailing and leading leg signals are converted to individual leg signals by following step 3.

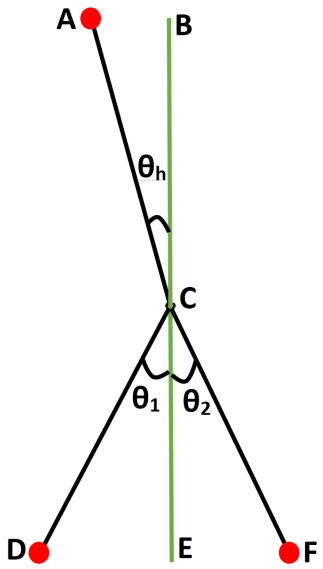


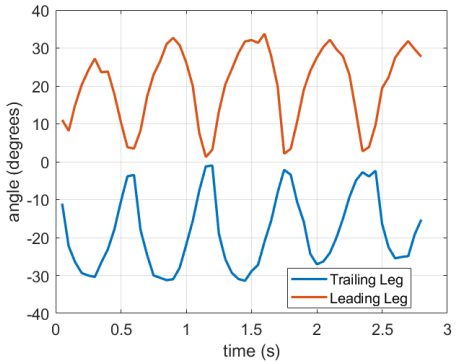
Figure 4.4: Skeleton model to compute leg and head angles

4.1.4 Testing Gait Signal Generation

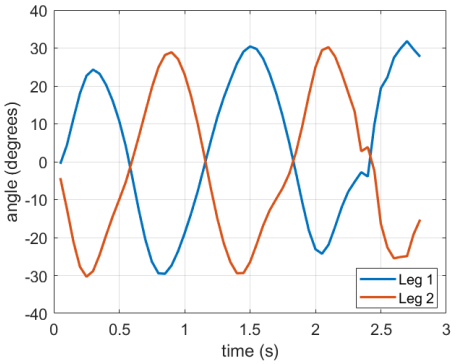
The automated generation of gait signals was tested on indoor videos acquired from The Institute of Automation, Chinese Academy of Sciences (CASIA-B) (*CASIA Gait Database* 2005) gait dataset. The dataset contains 13640 indoor gait samples of 124 people in 11 different viewing angles separated by 18°. Since the algorithm targets pedestrians with side-on camera view, test videos only had pedestrians walking with a side-on view. The dataset offers a sufficient number of test videos for testing a gait algorithm, however, it is a controlled environment that is, no variation in lightning, static and simple background, no shadows and free of occlusion.

Results (Figure 4.5) show the successful operation for retrieving leg signals (angles) of a healthy person and has a correlation of -0.96 depicting highly symmetrical but out-of-phase movement. Figure 4.5a shows the gait signals for leading and trailing legs. An approximate 50% samples in these signals are contributed by leg 1 and remaining by leg 2. Figure 4.5b shows the outcome of the proposed method to retrieve gait signal for each leg. The results are consistent when the algorithm was applied to the other pedestrian walking videos with the same viewing angle. Later on, the same experiment was repeated on outdoor videos containing healthy and disabled pedestrians to obtain leg positions over time (see Figure 4.6). Correlation between leg 1 and

leg 2 gait signals was calculated and resulted in −0.93 and −0.92 for a healthy and disabled person, respectively. The negative sign in correlation shows that gait signals are 180° out-of-phase gait signals due to the movement of legs in opposite directions.

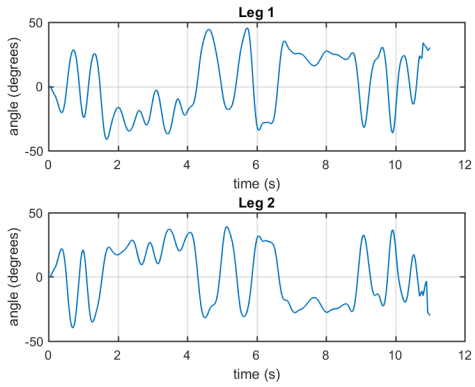


(a) Leading and trailing legs gait

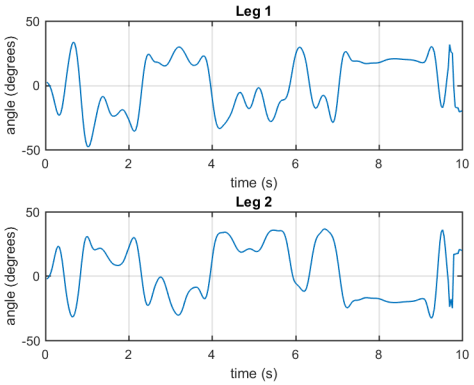


(b) Gait signals for individual legs

Figure 4.5: Leg signals of a healthy person in indoor environment



(a) Healthy person



(b) Disabled person

Figure 4.6: Gait signals (leg angles) in outdoor environment

4.1.5 Discussion

In the skeletonisation scheme, leg angles are estimated from the orientation of lines joining the lower extremal points with the centroid and vertical reference line, as shown in Figure 4.3e. Although angular information lacked distinguishing features, skeletonisation can help us analyse other motion features like position and velocity of legs. The fact that the lower 20% portion of the human body contributes 80% in human recognition by gait (Sarkar et al. 2005) implies that leg movements can contain useful gait information. Head

angle and position could also be investigated for gait signal. However, our experiments (in Section 4.2) revealed that the head exhibits translational motion, and its gait signal is too sensitive (noisy) due to fluctuations caused by rotations and minor head movements. Furthermore, the head point is a single source for its gait signal while leg data is contributed by two sources (Leg 1 and Leg 2 in our case) making it less prone to inaccurate readings compared to the head data. Therefore, we focused more on the leg signals, and the head signal was dropped and not presented here.

It is observed in Figure 4.6 that the technique for extracting the gait signal performs poorly on the outdoor videos while it performs well on indoor video dataset, but they lack disabled pedestrians. Six indoor videos (each having one person in it) were analysed, and results were consistent with the plots shown in Figure 4.5. Four videos (each having one person in it) were analysed, and none of them showed a consistent gait signal (see Figure 4.6). The gait signals for outdoor videos showed a large amount of noise and lacked a unique cyclic pattern that can serve as a template or a reference chart when analysing the motion of pedestrians for outdoor scenes. It is also hard to identify a strangeness in motion patterns for a disabled person (Figure 4.6b) from a healthy person (Figure 4.6a).

The inaccuracies in pedestrian localisation by YOLO and the presence of shadow are responsible for noise in the gait signal. In outdoor environments, shadow often appears as part of the pedestrian and moves along the target (person), thus affecting the GMM performance. According to YOLO's authors (Redmon & Farhadi 2017), localisation errors account for more of YOLO's errors than all other sources combined. Therefore, the bounding box prediction by YOLO may include unwanted space and shadows around the pedestrian that leads to poor skeletonisation and gait signal. These errors result in inaccurate centroid estimations and error propagates in head/legs angle (and position) computations resulting in gait signal less reliable for detecting abnormal walking behaviour.

The type of mobility aid, including wheelchair, mobility scooter and walking frame, also contributed to the failure of the proposed technique. These mobility aids occlude the lower portion of the pedestrians, and as a result, the pedestrian detector (YOLO) can only detect the visible (upper portion) of the pedestrian. In such cases, the star skeletonisation technique (Fujiyoshi et al. 2004) can not estimate the position of legs required for gait analysis. An example is shown in Figure 4.7. Due to the above reasons, the gait signal was not observable, and therefore the method was abandoned. The gait can also help identify wheelchair and mobility scooter users if the system can accurately locate the toe or ankle of the disabled pedestrian. Because a wheelchair

or mobility scooter user does not move his/her legs, therefore, the stride length and leg velocity feature will stay constant (with a little noise) throughout the walking cycle. In such a case, the absence of gait itself indicates a mobility aid with wheels.

Figure 4.7: Mobility aid hiding the pedestrian's leg

The question that, "is there a discriminating feature in gait signal that can help to categorise disabled person from healthy?" remains unanswered. This leads us to manually segment the location of legs and head in video frames over time, and results are reported in subsection 4.2.1. The shape of gait signals was found consistent for pedestrians appearing in multiple videos moving in fronto-parallel directions.

4.2 Manual Gait Extraction

We chose a set of nine videos from our video database to manually mark the position of legs (named as leg 1 and leg 2), centroid and head of the walker. These videos captured pedestrians walking with the side-on view. A MATLAB program was employed to load the test video that asked the user to mark the above locations on the video frames successively. A total of 1954 video frames from nine video footages were marked without skipping any frame. Figure 4.8 shows an example of manual markings and a snapshot of numerical time-series data. In order to extract the leg positions, heels were marked; for centroid, an approximate centre point was marked; and for head location, the crown region was marked. For convenience, we declare the front leg as leg 1 and rear leg as leg 2. Marked locations were later saved in a local file for further analysis. We consider that centroid location may not be suitable for a reliable outcome due to limitations in locating the exact centre of the body.

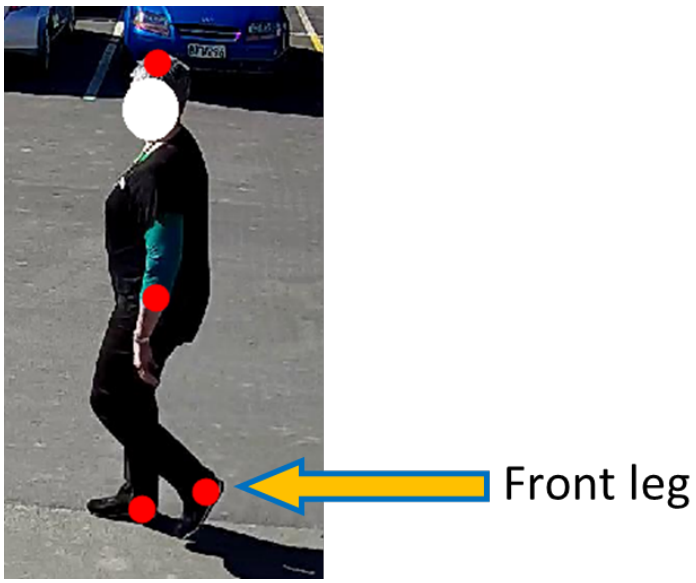


Figure 4.8: Manual marking of walker's head, centroid and legs

The spatial locations of the head and legs were plotted to spot their contribution in the gait signal. Figure 4.9 shows the motion profile of the walker, and it can be observed that the head undergoes a translation motion. At the same time, legs movements are comprised of translational and oscillatory components. The fact that a disabled pedestrian's walking style is different from that of healthy people implies that their leg movements will show a distinctive pattern which can lead to segregating the disabled pedestrians from a group of healthy and disabled people. In order to observe the leg's movement, the horizontal (stride length) and vertical (velocity) components of the motion have been studied. Stride length is defined as the horizontal distance between the two ankles while vertical velocity is the time rate of change of the vertical displacement. In addition to these features, the cross correlation was calculated to measure the similarity of leg 1 and leg 2 velocity signals as a function of time-lag.

Consider (x_1, y_1) and (x_2, y_2) represent the spatial locations of leg 1 and leg 2, respectively then the gait features can be described as,

$$Stride\ Length = S = |x_2 - x_1| \tag{4.16}$$

$$Vertical\ Velocity = V = \frac{dy}{dt} \tag{4.17}$$

$$Cross\ Correlation = R = \int_{-\infty}^{\infty} V_1(t)\, V_2(t+\tau)\, d\tau \tag{4.18}$$

where V_1 and V_2 represent the vertical velocities of leg 1 and leg 2, respectively. A plot of stride length and vertical velocities versus time is shown in Figure 4.10. If n is an even number denoting the total number of strides in a

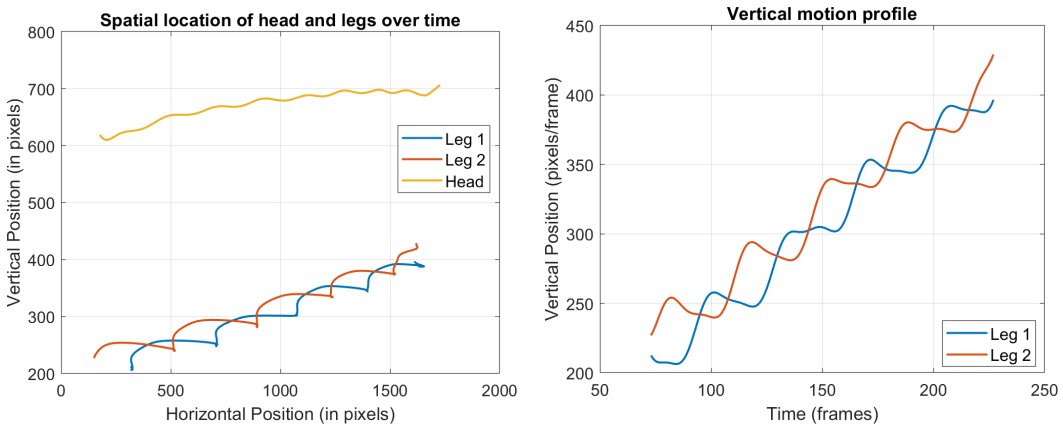


Figure 4.9: Motion profile of a walker

walk sequence, then vectors containing the stride lengths can be defined as,

$$T = [T_1, T_2, T_3, ..., T_{\frac{n}{2}}] \tag{4.19}$$

and

$$B = [B_1, B_2, B_3, ..., B_{\frac{n}{2}}] \tag{4.20}$$

such that $T_i \geq B_j \; \forall \; j = 1, 2, 3, ..., \dfrac{n}{2}$. Now the variation in stride length (dS) can be calculated as,

$$\Delta S = \left(1 - \frac{\sum_{s=1}^{\frac{n}{2}} B_s}{\sum_{s=1}^{\frac{n}{2}} T_s} \right) \times 100\%. \tag{4.21}$$

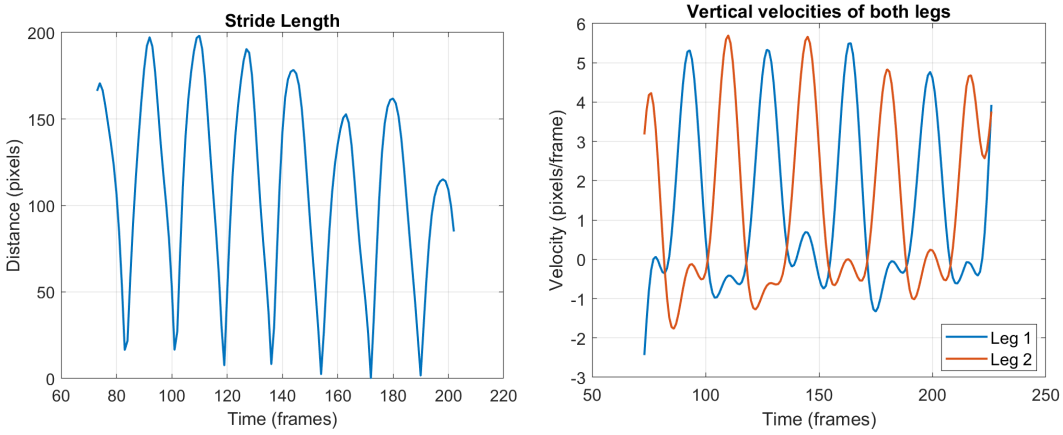


Figure 4.10: Gait features of a walker

Consider A_{1i} and A_{2i} representing the amplitude of leg 1 and leg 2 velocities during the i^{th} gait cycle. A gait cycle "starts when one foot makes contact with the ground and ends when that same foot contacts the ground again"

(Farlex 2012). For each leg, the average amplitude ($\bar{A}_1$ and $\bar{A}_2$) is calculated by averaging amplitudes of all gait cycles. The variation in vertical velocity can be estimated by comparing the mean amplitudes for relative difference, and mathematically defined as,

$$\Delta V = \frac{|\bar{A}_1 - \bar{A}_2|}{\max(\bar{A}_1, \bar{A}_2)} \times 100\%, \tag{4.22}$$

where,

$$\bar{A}_1 = \frac{1}{N} \sum_{i=1}^{N} A_{1i}, \tag{4.23}$$

$$\bar{A}_2 = \frac{1}{N} \sum_{i=1}^{N} A_{2i} \tag{4.24}$$

and

$$N = \frac{n}{2}. \tag{4.25}$$

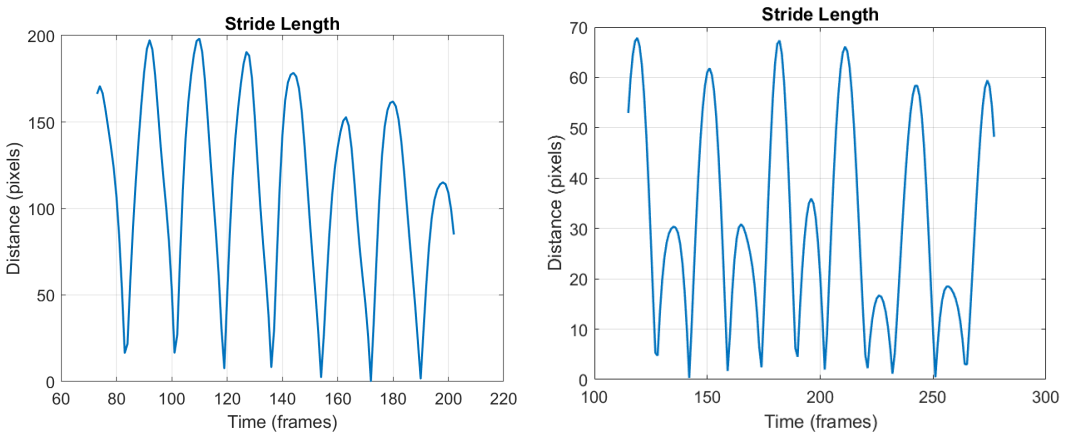
To combine variations in gait signals ($\Delta S, \Delta V$) and R, a net score metric is defined as,

$$Net\ Score = \frac{R}{\Delta S + \Delta V} \tag{4.26}$$

4.2.1 Results and Discussion

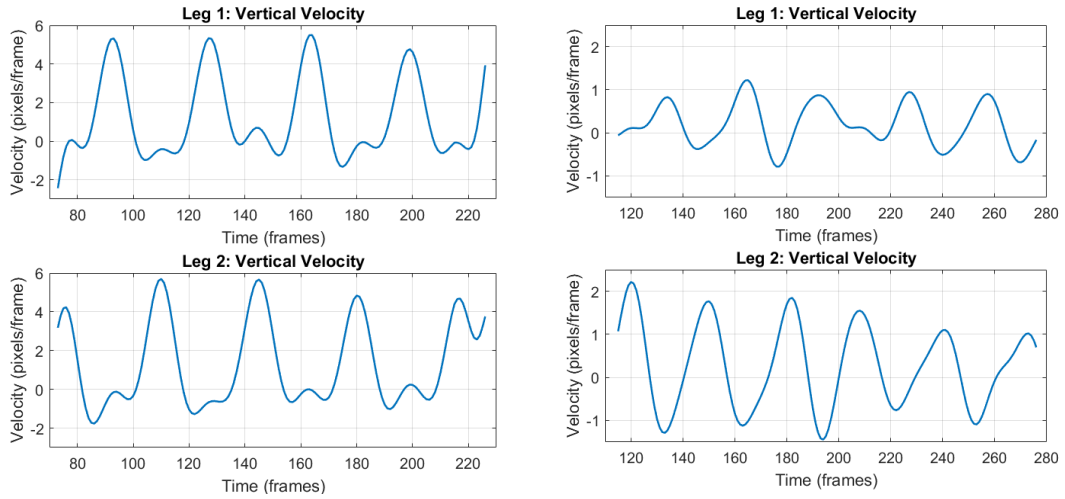
To observe the variation in gait features, stride length and vertical velocities of participants in our video dataset were plotted. A comparison between a healthy and a disabled pedestrian's gait signals is shown in Figures 4.11, 4.12 and 4.13. It can be observed in Figure 4.11 that stride length peaks are consistent for a healthy person, while, those for a disabled person have alternating shorter peaks. This trend shows that the participant is taking a regular-sized stride and a shorter stride in an alternating manner, thus indicating an abnormality with one leg (leg 1 to be precise).

Similarly, the velocity charts in Figures 4.12 and 4.13 indicate that leg 1 of the pedestrian is moving slower than leg 2, and therefore shows an abnormal movement. Table 4.1, Table 4.2 and Figure 4.14 show the variation in gait features extracted from videos having healthy and disabled pedestrians. It is observed that healthy people have smaller variation in stride length and vertical velocities when compared with those of disabled pedestrians. The small variation in a healthy person's gait features shows that both legs move identically in horizontal and vertical directions. In contrast, a disabled person legs lack an identical motion either in a horizontal direction or vertical direction or both.



(a) Healthy Person (b) Disabled Person

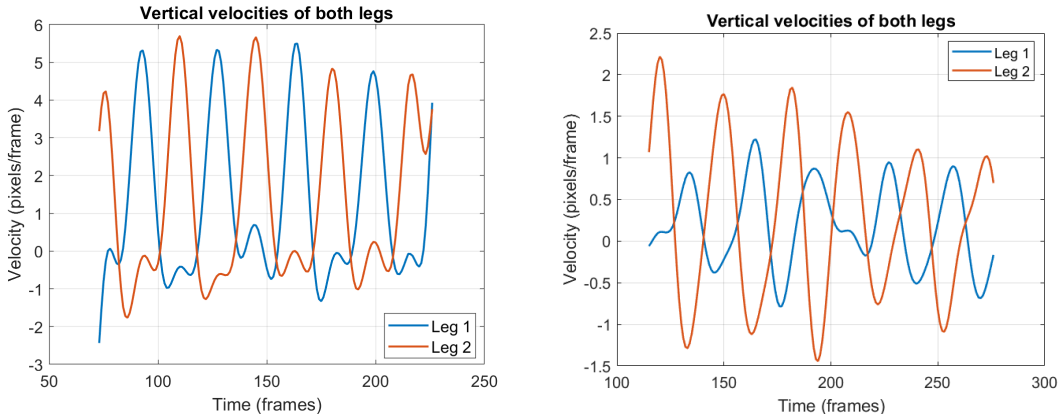
Figure 4.11: Stride Length Comparison



(a) Healthy Person (b) Disabled Person

Figure 4.12: Velocity Comparison

The manual marking experiments were performed on the nine video clips (each having one person) from the outdoor surveillance video set explained in subsection 3.2.2. Out of nine individuals, three were healthy, three people who only had a limp but did not use a mobility aid and three people using a mobility aid. Table 4.1, Table 4.2 and Figure 4.14 show the observed values of ΔS, ΔV and R for all nine individuals and we see that healthy scores are noticeably different from the rest. Figure 4.14 and Table 4.2 represent the Table 4.1 in a different manner. Figure 4.14 is just Table 4.1 shown as a figure. Table 4.2 is showing the mean and standard deviation over the subcategories in Table 4.1. We can see that there are significant differences between healthy and disabled people on ΔS and ΔV measurements but not so much on R measurement.



(a) Healthy Person (b) Disabled Person

Figure 4.13: Velocity Comparison Charts

Table 4.1: Variation of gait features for healthy and disabled pedestrians

Pedestrian Type	Variation between Leg 1 and Leg 2 Stride Lengths (ΔS)	Difference between Leg 1 and Leg 2 velocities (ΔV)	Correlation b/w Leg 1 and Leg 2 Velocity Signals (R)	Net Score
Healthy	14%	2%	0.89	5.36
Healthy	15%	6%	0.89	4.24
Healthy	16%	3%	0.82	4.45
Limp	35%	48%	0.83	1.00
Limp	47%	67%	0.75	0.66
Limp	38%	9%	0.84	1.81
Walking Frame	39%	14%	0.69	1.30
Walking stick	24%	34%	0.74	1.30
Walking stick	59%	49%	0.77	0.72

Table 4.2: Mean and standard deviation of gait signals of healthy and disabled people

	Healthy	**Disabled**
ΔS	$15\% \pm 1\%$	$40\% \pm 12\%$
ΔV	$4\% \pm 2\%$	$37\% \pm 22\%$
R	$87\% \pm 4\%$	$77\% \pm 6\%$
Net Score	4.68 ± 0.60	1.13 ± 0.43

So we calculated $p - value$ test on R measurements for healthy and disabled (limb and mobility aids) categories. At 95% confidence, it rejected the null hypothesis implying that samples are distributed with two different means, but there is some overlap in values.

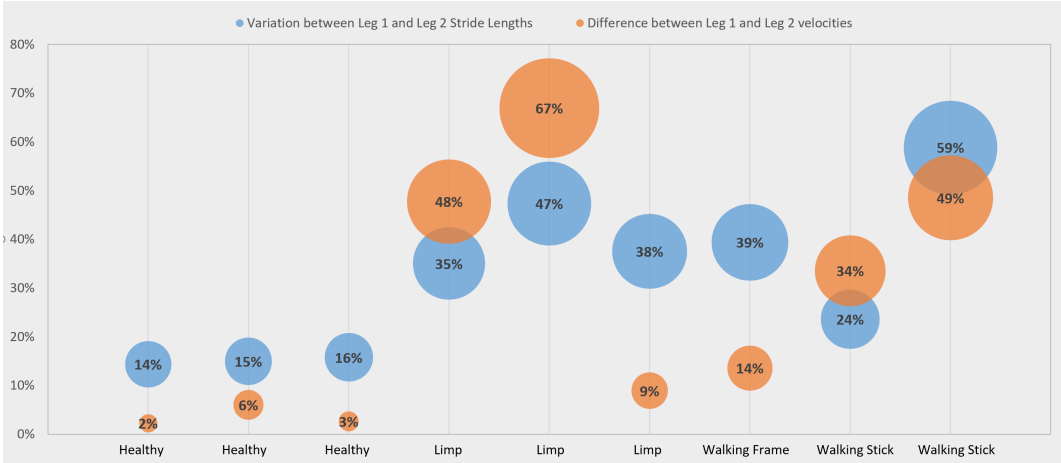


Figure 4.14: Variation in gait features

4.3 Summary

Gait recognition procedures are affected by the variation in environmental constraints, viewing angle, occlusions, shadows, an imperfection in foreground modelling, object segmentation and silhouette extraction. We experimented on video data of moving pedestrians to investigate the presence of a disabled person by automated and manual analysis of head and leg data. The automated system failed to recognise a disabled person from its gait due to imperfections in segmentation. YOLO's localisation error restricts the automated system to show impressive results. However, results from the manual localisation suggest that there is enough information in the gait signal to characterise a healthy motion given a set of gait signatures. Gait signals of a healthy pedestrian are cyclic, even functions and deterministic in the pattern. In contrast, those of disabled pedestrian differ from the standard pattern and deviation depends on the type of disability and mobility aid used.

A two-dimensional gait signal representing the head movement lacks the distinguishing features to identify an abnormality in disabled person's gait. A viewer can observe a periodic disturbance in the spatial movement constituted by the head and the type of disturbance depends on the nature of the disability. Therefore, a 3D signal is needed for gait analysis.

We believe that automated human joints detection can significantly improve gait signal, which may lead to a reliable system for disabled person detection. CNN based systems (Cao et al. 2018) have shown encouraging results in identifying joint locations and connecting them to form a skeleton model for human objects. Such a system can localise the body parts accurately and capable of producing better results than traditional gait techniques.

Chapter 5

Mobility Aid Detection

In Chapter 4, we extracted motion information crucial for differentiating a disabled person from a healthy person from videos. Manual extraction of gait signals revealed that there is useful information in the gait of a walking person for detecting unusual movement patterns. However, an automated scheme failed to reproduce the same information due to the problem of shadow and inaccurate segmentation of the pedestrian's silhouette. Now we adopt a different approach and explore the potential of a CNN (YOLO) for detecting mobility aids and pedestrians in videos. CNNs require large image datasets for training and numerous object detection applications have been developed that extensively employ pattern recognition capabilities of CNN. In the literature, we have not found any working software or an intelligent system that can identify both mobility aids and pedestrians in outdoor videos. This led us to form an image database of mobility aids and person that can be used for training a neural network. Details about the image dataset can be found in Chapter 3.

In this chapter, a single-stage CNN-based framework (YOLO) is proposed for pedestrian and mobility aid detection in RGB videos. The detection system is integrated with object tracking, data association and a counting module (as detailed in Chapter 6). The flowchart , Figure 5.1 describes the design concept of the whole system.

This chapter is organized as follows: Section 5.1 is the description of the image dataset used for the CNN training. In Section 5.2 the proposed neural network, its architecture and training phase is detailed, with the evaluation of the proposed mobility aid detection system presented in Section 5.3.

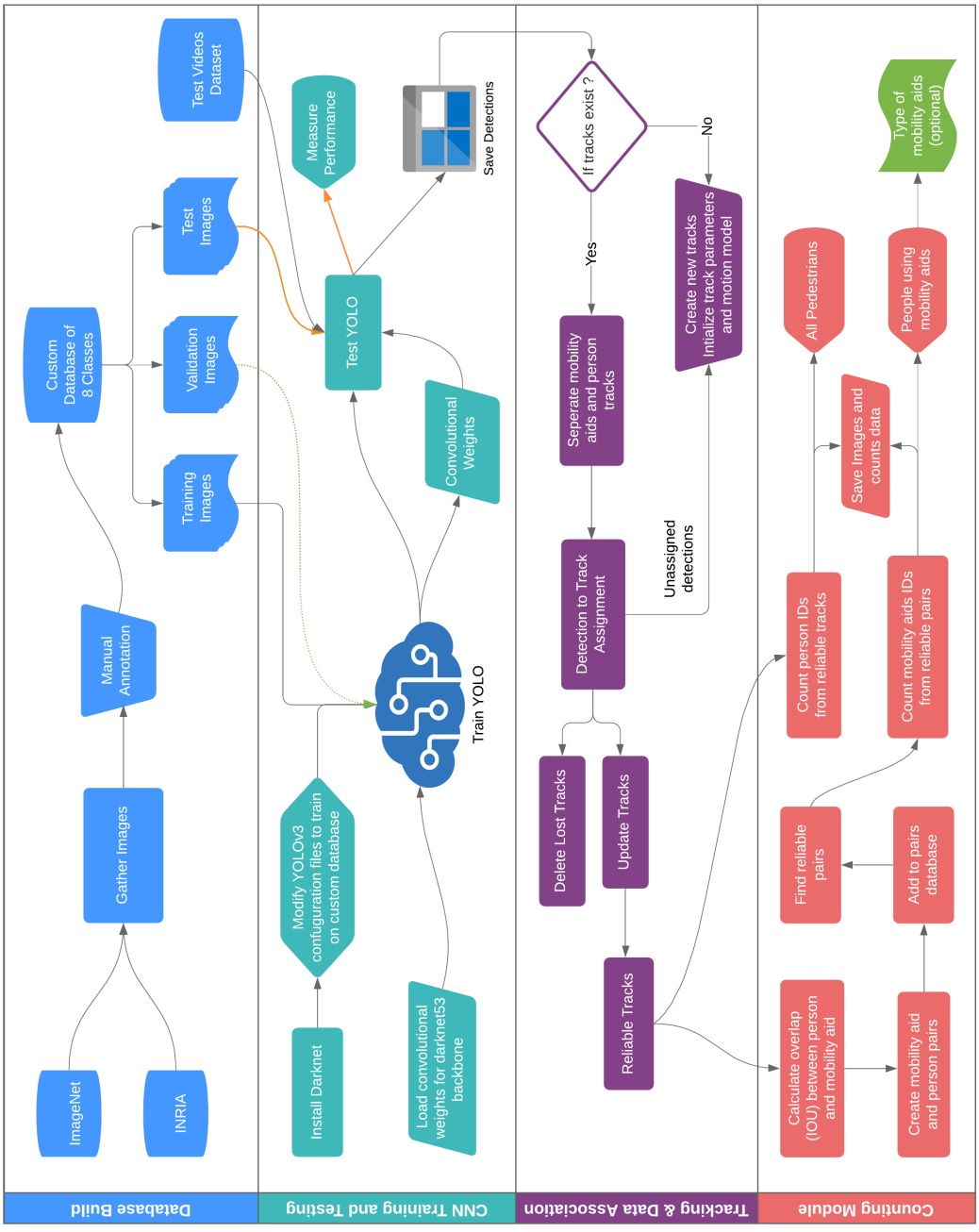


Figure 5.1: Flowchart showing the system design

5.1 Training Dataset

The current publicly available image databases lack images of mobility aids annotated with ground truth information (bounding boxes). The data acquisition and labelling add a step towards the system design for mobility aids detection. Our custom dataset of mobility aid images was collected from multiple sources and was labelled manually (see Section 3.1). The image dataset has a total of eight classes inclusive of five mobility aids, pedestrians and the rest (car and bicycle) act as distractors having structures similar to those in wheelchair and mobility scooters. The training images from different categories are not self exclusive; that is, the training images for mobility aids often contain the person using them or nearby objects in the scene, which may or may not belong to classes in our image database. See Figure 3.1 and Figure 5.2 show some examples of multiple classes appearing in one image.

The location and size of objects (of eight classes) were recorded in terms of bounding boxes by following a list of labelling rules defined in Section 3.1. For each image file, a label file (text file) was generated that contained bounding box information in the format that Darknet (YOLO) uses. Each line in the label file corresponds to a BB of an object in the image and looks like: $[object\text{-}class(class\ ID), x, y, width, height]$. Here (x, y) give the centroid of the BB, and $width$ and $height$ are dimensions of the BB. All of these numbers are scaled between 0 and 1 relative to the image's width and height. Figure 3.1 shows a few examples of images in the database and their label files.

Images are picked randomly to divide the dataset into three sets being training, validation and test images. YOLO does not require validation images at the training phase but uses a loss function based on the prediction box overlap with the ground truth. A portion of the database images (validation) was reserved for later use or for validating another CNN model if YOLO does not perform well. YOLO showed a satisfactory performance; therefore, we combined validation and test images into one category to test the mobility aid detector. Results are shown in section 5.3. Figure 5.2 shows some sample images taken from the training dataset. Test images comprise 22% of the total image dataset size used in this research. This breakdown is shown in Table 5.1.

5.2 Mobility Aid Detector

The YOLO object detector was selected for detecting mobility aids in the video frames or images. In a comparison study (Redmon & Farhadi 2018), it outperformed modern CNN based object detectors on a processing time basis and has a comparable mean average precision. YOLO is flexible in terms

Figure 5.2: A sample of mobility aid images for training CNN

Table 5.1: Number of Images containing objects of specific classes

Object Type	Training	Testing	Total
Wheelchair	931	250	1181
Crutch	514	205	719
Walking Frame	513	135	648
Walking Stick	573	155	728
Mobility Scooter	512	140	652
Person	324	95	419
Car	663	175	838
Bicycle	500	134	634
Total	4530	1289	5819

of retraining the network for custom image datasets, given the configuration and labelled image data. YOLO is capable of processing videos in offline and real-time modes.

A YOLOv3 network is a hybrid approach that combines Darknet-19 and residual neural networks (ResNet). Darknet refers to an open-source neural network framework written in C and CUDA, and the fact that YOLOv2 had

19 convolutional layers, the network was named Darknet-19. The new network is named Darknet-53 because it has 53 convolutional layers. Redmon & Farhadi (2018) trained YOLOv2, YOLOv3, ResNet-101 and ResNet-152 on the ImageNet data with identical training and test settings. Darknet-53 performed better than ResNet-101 and was 1.5 times faster, and it showed similar performance to ResNet-152 at two times faster inference time. Darknet-19 and Darknet-53 network structures are shown in Figure 5.3.

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

(a) Darknet-19 (YOLOv2)

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

(b) Darknet-53 (YOLOv3)

Figure 5.3: Network architectures of YOLO

5.2.1 Network Predictions

Modern CNN based detectors use anchor boxes to fit the location, size and dimension of objects they specialize in predicting. Setting up anchor boxes' width and height can be tricky and many modern CNN detectors (Ren et al. 2015, Liu et al. 2016) choose the anchor boxes' size and aspect ratio by hand. Instead of choosing anchor boxes by hand, YOLO runs k-means clustering on bounding boxes of the training set to automatically find optimal anchor boxes. Redmon & Farhadi (2017) found that using a standard k-means with Euclidean distance is not an adequate metric as it generates more error for larger boxes than smaller boxes. Therefore, an IOU based distance metric was formulated as it is independent of the size of the box. The distance metric is defined as,

$$d = 1 - IOU \tag{5.1}$$

The network predicts location and dimensions for each bounding box along with an objectness score for each bounding box using logistic regression. This should be 1 if the anchor box overlaps an object more than any other anchor box while a prediction is ignored if the anchor box is not the best fit but does overlap a ground truth object by more than some threshold.

During the training phase, YOLO resizes images to extract features at multiple scales. The last layer predicts a three-dimensional tensor encoding bounding box, objectness and class predictions. In our experiments, the output tensor shape is $N \times N \times [3 \times (4 + 1 + 8)]$ for predicting 4 bounding box offsets, 1 objectness prediction, and 8 class predictions at 3 scales (resolutions). Here N is the grid size which refers to all possible locations for the bounding box. Objectness is the likelihood that a given anchor box encloses an object belonging to any class in the training dataset while class prediction is the conditional probability of a particular class given that there is an object. The objectness prediction and class prediction scores are combined into one final score that tells us the probability that the bounding box contains a specific type of object.

5.2.2 Bounding Box Prediction

The loss function used for training uses the sum of squared error for bounding box regression and binary cross entropy (BCE) (Godoy 2018) for object classification to help improve detection accuracy. The loss function for the YOLOv3 is not mentioned in detail by the author in the YOLOv3 paper (Redmon & Farhadi 2018). However, the loss calculation is understood by looking at the source code[1] and referring to online blogs. There are four parts of the loss function,

1. The adjustment parameters of the BB's centre point x and y using sum of squared error loss,

2. The adjustment parameters of the BB's width and height using sum of squared error loss,

3. BCE loss term of objectness/confidence score of a bounding box. The confidence scores reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts.

4. BCE of multi-class predictions of a bounding box

Let the ground truth for some coordinate prediction be t then the gradient is the ground truth value (computed from the ground truth box) minus the

[1]YOLOv3 source code is available at https://pjreddie.com/darknet/yolo/

network's prediction $\hat{t}$, i.e., $t - \hat{t}$. The loss function uses a mask (1_i^{obj} and 1_i^{noobj}) for each cell to punish the network if it predicts an object in a cell when there was not any. 1_i^{obj} is 1 for cells having an object in them and 0 for other no-object cells. 1_i^{noobj} is the inverse of 1_i^{obj} and it is 1 when there is no object in the cell i and 0 elsewhere.

For the purpose of our study, we added a subroutine to the YOLO coding package for saving the predictions inside a text file that will be used later during tracking and counting modules. The detections file records the following set of information for each frame,

- Frame number
- Object label for example wheelchair, car and person.
- Four coordinates for each bounding box $[x, y, width, height]$. Where (x, y) indicate the upper-left corner of the rectangle, and the $width$ and $height$ specify the size.
- The likelihood that the bounding box contains a specific type of object

5.2.3 Class Prediction

In our training dataset, there are many overlapping labels (i.e., wheelchair and person) in one bounding box. Similar situations can occur in test images and videos where multiple objects can overlap and appear at the same location. To address this issue YOLO network relies on independent logistic classifiers for multilabel classification. Multiple logistic regressions are better for modelling multiclass data than a softmax which assumes that each box has objects belonging to precisely one class (Redmon & Farhadi 2017).

5.2.4 Training

For training, we adapted the "transfer learning" approach by using convolutional weights from the darknet53 model pre-trained on ImageNet. Those weights were calculated by the original authors for darknet framework to classify images for the 1000-class ImageNet challenge[2]. Pre-trained weights are useful in reducing training times when there are features common between the source (database on which YOLO was originally trained) and target (our custom) datasets (Sharif Razavian et al. 2014, Kimura et al. 2020). Car, bicycle and pedestrian objects featuring in both datasets reduce the number of epochs required to minimize the loss function.

[2]Available from https://pjreddie.com/darknet/yolo/

The training configuration file was altered so that the dimensions of the output layer (fully connected layer) can detect objects from eight classes (Five mobility aids, person, car and bicycle). Furthermore, training parameters were tuned to cater to the custom database and the number of classes in the training set.

The Darknet neural network framework is used for training and testing (Redmon & Farhadi 2018). The CNN is trained on full images with no hard negative mining, that is, without explicitly creating negative examples out of falsely detected patches. Hard negative mining involves adding negative samples to the training set to retrain the classifier, which may help reduce the number of false positives (Jin et al. 2018).

Both YOLOv2 and YOLOv3 are trained on the same dataset and evaluated for detecting mobility aids in test images. Their performance is also compared to determine a better performing network for our particular application. The average precision and recall are calculated using formulae for N multi-class problem (Sokolova & Lapalme 2009), as,

$$Average\ Precision = \frac{\sum_{i=0}^{N} tp_i}{\sum_{i=0}^{N} (tp_i + fp_i)} \tag{5.2}$$

$$Average\ Recall = \frac{\sum_{i=0}^{N} tp_i}{\sum_{i=0}^{N} (tp_i + fn_i)} \tag{5.3}$$

where tp is true positive, tn is true negative and fp stands for false positive. The network was trained for 97147 batches (at a learning rate of 0.001), and weights were saved after every 10000 iterations. The batch size was set to 64 images, and training images were processed for 100000 iterations with two NVIDIA GeForce GTX 1080Ti graphics cards. Appendix C shows a screenshot at the end of the YOLO's training phase.

5.3 Results

The trained network is tested on test images specified in Table 5.1 and on surveillance videos with people using mobility aids. These videos are part of our custom-built dataset that was collected locally. An example of the detection outcome for a video frame is shown in the Figure 5.4. Detection results from videos are shown in Figure 5.5. Processing time is also recorded

for a collection of short video clips, and experiments revealed that it depends on the number of objects detected in a given frame. Table 5.2 lists processing speeds (FPS) and times for some short videos. Later on, we processed 121 test videos from our videos dataset (section 3.2) with the CNN detector and observed that the detection system is stable over the long term and processing times are precise.

(a) Frame 7308 of a test video (b) Prediction outcome data

Figure 5.4: System testing example

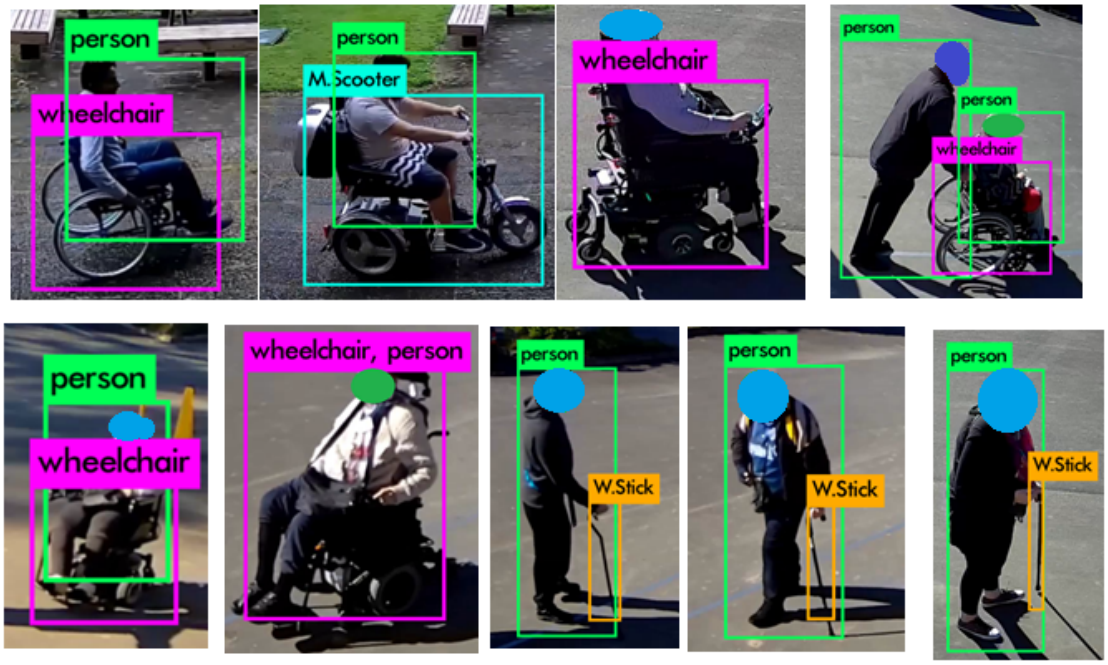


Figure 5.5: Correct mobility aids detection in test videos

The IOU metric, defined by Equation 2.1, has been used to measure the localization accuracy of the mobility aid detector on test images. Following the recommendations by Dollar et al. (2012) and Enzweiler & Gavrila (2008),

Table 5.2: CNN processing times on test videos

objects per frame	frames tested	processing speed (FPS)	processing time (ms)
1-2	570	22.80	43.86
2-3	900	22.50	44.44
4-5	390	19.50	51.28
6-7	2071	15.34	65.19

we set a cut-off value of 0.5 for the IOU overlap score. That is, an overlap of bounding boxes resulting in an IOU score of 0.5 and above is considered a *hit*, otherwise it is a *miss*. We did not have the chance to explore the IOU values further to find the best IOU cut-off score specifically for our dataset as it was beyond our scope and time.

The confusion matrix summarizing the detection result using IOU is provided in Table 5.4. Entries in 'Others' row refer to objects not belonging to any of the eight classes in our datasets but classified as one of the mobility aids, person, car and bicycle. These objects are incorrectly classified because of their appearance, similar to those in the training database.

YOLO v2 and v3 are trained and tested on the same datasets (see Table 5.1 and section 3.1 for performance comparison). Table 5.3 shows the average precision and recall values for YOLOv2 and YOLOv3.

Table 5.3: YOLOv2 versus YOLOv3 performance comparison

YOLO	Precision	Recall	F_1 Score
Version 2	0.81	0.86	0.83
Version 3	0.89	0.92	0.90

Results in Table 5.4 show a satisfactory detection performance implying that mobility aids can be detected from surveillance videos using a CNN. Table 5.3 show that YOLO version 3 performs slightly better than version 2. Therefore, the mobility aid detector is built using YOLO's version 3, and it serves as the first step towards solving the problem of recognising disabled pedestrians. The next chapter will focus on the output of the mobility aid detector, and tracking and data association to count the mobility aid users.

Table 5.4: Confusion Matrix for the eight classes (IOU=0.5)

Predicted Class

Actual Class	**Object Type**	**Wheelchair**	**Crutch**	**Walking Frame**	**Walking Stick**	**Mobility Scooter**	**Car**	**Person**	**Bicycle**	**Total**
	Wheelchair	310(81%)	1(0%)	0(0%)	0(0%)	1(0%)	5(1%)	61(16%)	5(1%)	383
	Crutch	3(1%)	250(86%)	2(1%)	11(4%)	0(0%)	8(3%)	17(6%)	1(0%)	292
	Walking Frame	2(1%)	2(1%)	138(95%)	1(1%)	0(0%)	0(0%)	2(1%)	1(1%)	146
	Walking Stick	0(0%)	15(8%)	0(0%)	162(87%)	0(0%)	1(1%)	7(4%)	1(1%)	186
	Mobility Scooter	0(0%)	0(0%)	0(0%)	0(0%)	144(95%)	0(0%)	7(5%)	0(0%)	151
	Car	0(0%)	3(1%)	0(0%)	0(0%)	0(0%)	225(93%)	14(6%)	0(0%)	242
	Person	27(2%)	15(1%)	2(0%)	5(0%)	0(0%)	6(0%)	1414(96%)	9(1%)	1478
	Bicycle	5(2%)	0(0%)	0(0%)	1(0%)	0(0%)	2(1%)	23(11%)	171(85%)	202
	Others	4(4%)	8(8%)	0(0%)	13(13%)	1(1%)	9(9%)	58(58%)	7(7%)	100
	Total	351	294	142	193	146	256	1603	195	3180

Chapter 6

Counting Mobility Aids in Surveillance Videos

In this chapter, we focus on counting mobility aid users by combining results from the mobility aid detector with a tracker, a data association step and a counting module. Because the counting module is dependent on the detection outcome, the accuracy of the counts is indirectly dependant on detection accuracy. At this stage, detections of distractors (cars and bicycle) are discarded, and only mobility aids and pedestrians are taken into account. This is because we are not interested in counting cars and bicycles in the scene, however, the system can be upgraded to include those as part of any future work.

The counting module reads the detector output and initialises a track for each object that was detected by the system. This chapter covers the implementation of Kalman filter, an enhanced SORT data association system and our proposed counting module using the idea of coupling reliable mobility aid detections with pedestrians. The results are obtained by evaluating the entire system (detection and counting) on the test videos listed in section 3.2 and are an extension of the experiments performed/presented in chapter 5. A flowchart of this module is shown in purple and red colours in Figure 5.1, and step-wise implementation is shown in Algorithm 1. The following sections explain the individual steps of tracking, data association and counting steps in detail.

6.1 Multi-Object Tracking and Data Association

SORT (Bewley et al. 2016) algorithm is capable of performing real-time multi-object tracking and data association in one step given the detection data. It uses a Kalman filter for tracking detected objects while data association manages object status and properties over time. An in-house implementation of

Algorithm 1: Implementation of the proposed system

```
Result: Mobility Aids Count, Person Count, Detection and Pairing timelines
Run 8-Class YOLOv3 detector on test video
Set minAge, minVisibleCount and invisibleForTooLong;
Initialise pedestrian and mobility aid counters;
Load Detection Data;
for All frames do
    Read detection details;
    if Tracks Exist then
        Predict new locations of the tracks;
        Separate mobility aids and person tracks;
        Detection to track assignment;
        Update assigned tracks;
        Update unassigned tracks;
        if (age<minAge and visibility<0.5) or (invisible count >
          invisibleForTooLong) then
            Track is lost;
        end
    else
        Initialise a new track for each object;
    end

    for All tracks do
        if Visible Count ≥ minVisibleCount then
            Track is reliable;
            Add IDs of reliable objects to ID bank;
            Calculate overlap score (IOU) of person and mobility aid bounding
              boxes
        end
    end
    Create pair(s) of person and mobility aid based on IOU score;
    Update records in pair database;
    Extract reliable pairs;
    Count reliable tracks and reliable pairs;
end
```

SORT was created that included improvements in the data association algorithm by replacing the IOU based cost function with the "distance between active objects" based cost matrix to catch objects undergoing acceleration. The implemented data association makes use of the object type flag and prevents an identity type switch across multiple classes during the detection to track assignment. The identity switch may occur within the class, but that does not affect the overall count. Since overall counts are required as output and we do not care about the identity switch within the class, therefore, the implemented modified version is preferred over deepSORT (Wojke et al. 2017). deepSORT is an improved version of SORT and has a complicated data association step employing a CNN pre-trained on the pedestrian class to

reduce identity switches. The enhanced version is also limited to tracking single class objects, namely pedestrians only, which is unsuitable for the present application.

6.1.1 Object Tracker and Motion Model

The first step in setting up the object tracker is to define the state of moving objects and a motion model to propagate an object's identity into the next frame. The state of each object is defined as:

$$\mathbf{x} = [x, y, s, r, \dot{x}, \dot{y}, \dot{s}] \tag{6.1}$$

where (x, y) is the centre of the bounding box enclosing an object and express its spatial position. T variables s and r are the area and the aspect ratio of the bounding box, respectively. The area (s) is essential when an object is either approaching or moving away from the camera. In both cases, the size of the moving object changes and the tracking filter adapts to the changing size. The aspect ratio represents the type of the object being tracked. For example, a pedestrian is contained in a standing bounding box whereas a square-shaped bounding box can enclose a wheelchair or a walking frame.

The Kalman filter (Kalman 1960) has been incorporated to predict the location and size of targets in the next frame, which are then associated with detections from the CNN. The Kalman filter is a recursive two-stage filter, and it performs a predicting step and an update step at each iteration. The first step predicts the current location of a moving object using equations of motion and its previous location (see Equation 6.2). The update step combines the measurement about the object's current location (if available), $\mathbf{z_t}$, with the predicted location, $\hat{\mathbf{x}}_t$, to estimate the current location (CNN detection) of the object, $\mathbf{x_t}$. The equations governing the Kalman filter are listed in Table 6.1.

In this project, a linear constant velocity model is defined that is independent of other objects and camera motion. As we are dealing with very slow-moving objects, likely a few pixels at most per frame, one might expect the simplest linear prediction model to work well. However, the motion model can approximate the inter-frame displacements of the tracked objects. In addition, the motion tracker based on the Kalman filter is adaptive to speed changes, and it can handle the accelerating objects by adjusting the Kalman gain during the correction/update step. The state transition equation is given

Table 6.1: Kalman Filter Equations

Prediction Stage	
Prediction of state:	$\hat{\mathbf{x}}_t = \mathbf{F}_t\hat{\mathbf{x}}_{t-1} + \mathbf{B}_t\mathbf{u}_t$
Covariance estimation:	$\mathbf{P}_t = \mathbf{F}_t\mathbf{P}_t\mathbf{F}_t^T + \mathbf{Q}_t$
Update Stage	
Innovation or measurement residual:	$\tilde{\mathbf{y}}_t = \mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_t$
Innovation (or residual) covariance:	$\mathbf{S}_t = \mathbf{H}_t\mathbf{P}_t\mathbf{H}_t^T + \mathbf{R}_t$
Optimal Kalman gain:	$\mathbf{K}_t = \mathbf{P}_t\mathbf{H}_t^T\mathbf{S}_t^{-1}$
Updated (a posteriori) state estimate:	$\hat{\mathbf{x}}_{t\|t} = \hat{\mathbf{x}}_t + \mathbf{K}_t\tilde{\mathbf{y}}_t$
Updated (a posteriori) estimate covariance:	$\mathbf{P}_{t\|t} = (I - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_t$

by,

$$\begin{bmatrix} x_t \\ y_t \\ s_t \\ r_t \\ \dot{x}_t \\ \dot{y}_t \\ \dot{s}_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ s_{t-1} \\ r_{t-1} \\ \dot{x}_{t-1} \\ \dot{y}_{t-1} \\ \dot{s}_{t-1} \end{bmatrix} \quad \text{or} \quad \mathbf{x_t} = \mathbf{F}\,\mathbf{x_{t-1}} \tag{6.2}$$

Δt is the duration between two successive frames in a video and set to 1 frame (i.e., $\Delta t = 1$). The tracker therefore estimates the velocity of moving objects in units of pixels/frame. However, the system can determine counting results and detection timelines using the standard SI unit of time (seconds). Here $\mathbf{F}$ is the state transition matrix that specifies how the system goes from one state to the next, and it estimates the current state vector $\mathbf{x_t}$ from the previous state vector $\mathbf{x_{t-1}}$ without using the measurement data $\mathbf{z_t}$. In Table 6.1, $\mathbf{B}$ and $\mathbf{u}$ are control-input parameters and are applicable when systems have an (external) input. These are set to zero following a common object tracker design practice.

The measurement vector $\mathbf{z}_t$ is calculated from the output of the detection system designed in Chapter 5 by,

$$\mathbf{z}_t = \begin{bmatrix} x, & y, & width \times height, & \dfrac{width}{height} \end{bmatrix}^T \tag{6.3}$$

where (x, y) is the centre of the bounding box and the *width* and *height* specify the size of the detected object. The last two elements in $\mathbf{z}_t$ are the area (s) and the aspect ratio (r) of the detected object. If a target is not detected in subsequent frames, tracker predicts the subsequent state using

the motion model without correction. $\mathbf{H}$ maps the state vector, $\mathbf{x}_t$, to the measurement vector, $\mathbf{z}_t$. The variables x, y, s and r are mapped straight from $\mathbf{z}_t$ to $\mathbf{x}_t$, whereas the derivative variables are not included in this mapping because those are not directly measured but calculated after the update step. The measurement matrix is therefore,

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{6.4}$$

The estimate covariance matrix $\mathbf{P}$ is the estimated accuracy of $\mathbf{x}_t$ at time t and the filter regulates its value over time. If exact values for state variables are known at the start-up, then $\mathbf{P}$ is initialised as a zero matrix. Because the initial position and velocities of objects are unknown, $\mathbf{P}$ is initialised as a diagonal matrix with large values due to high uncertainty in the unobservable initial velocities.

$$\mathbf{P} = 10 \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1000 \end{bmatrix} \tag{6.5}$$

The process noise matrix $\mathbf{Q}$ is the deviation of the input signal from the "ideal" transitions defined by the transition matrix $\mathbf{F}$. Large values in this matrix imply that the input signal has a high variance, and the filter needs to be extra adaptable and vice versa. In practice, $\mathbf{Q}$ can be tricky to define and may require fine-tuning. Here $\mathbf{Q}$ is taken as,

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0001 \end{bmatrix} \tag{6.6}$$

The measurement noise matrix $\mathbf{R}$ depends on the accuracy of the mobility aid detector, which can be estimated experimentally. Small values in $\mathbf{R}$ show optimism about the detector's performance with detections considered accurate. In such cases, the predicted signal mimics the observed signal. On the

other hand, large values in $\mathbf{R}$ show lack of confidence in the accuracy of the detector (or measurements), so additional smoothing is required. $\mathbf{R}$ is defined by,

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \tag{6.7}$$

In the update stage, the difference between the predicted and measured states, $\tilde{\mathbf{y}}_t$, is calculated along with the Kalman gain matrix, $\mathbf{K}$, which is the relative weight allocated to predicted and measured states. Lastly, the state vector $\mathbf{x}_t$ and its error covariance $\mathbf{P}_t$ are updated with the measured state using the update equations given in Table 6.1.

The KalmanFilter class in Python's FilterPy library can initialise the system variables with default values along with the freedom to define those variables explicitly. The FilterPy module also contains subroutines to perform most of the prediction and update steps to update system variables. During the implementation of the Kalman filter, it has been assumed all errors/noise are statistically independent and Gaussian distributed.

6.1.2 Object Attributes

The term 'Object' is used throughout the remaining section and following chapters to refer to the objects listed in Table 5.4. For each object detected by the mobility aid detector, a class with name 'tracks' is initialised and whose member variables are explained in the following:

1. **id**: The id is a unique numeric identifier assigned to each detected object in the order of their detection. Its value can range from 1 to the number of objects appearing in the test video.

2. **type**: Name of the detected object which can include a wheelchair, crutch, walking frame, walking stick, mobility scooter, car, person and bicycle. The name label helps in the data association part where an object in the current frame can only be associated with that of same type in the next frame. It is also handy when creating pairs of pedestrians and mobility aids as detailed in Subsection 6.2.1 below. Furthermore, the name field helps in breaking down the total count into counts for each sub-category.

3. **bbox**: This variable contains the location (centre) and size (width and height) of the detected object. Their values may change depending on the velocity of the moving object.

4. **centre**: The centroid of the detected object is calculated from values in *bbox* field and required by the Kalman filter to perform tracking.

5. **KF**: The KF is a programming class that contains the configuration and values required to implement a Kalman filter based tracker associated with each object. The state of these variables is updated with time as the prediction and update steps occur.

6. **age**: The age of each detected object refers to the number of consecutive frames an object was detected or predicted. The age counter starts when an object enters the scene (first detection) and keeps incrementing until it leaves the scene. The age indicates the reliability of detections which is important to identify the reliable tracks and reliable pairs of mobility aids and person as explained in Subsection 6.2.1 below.

7. **totalVisibleCount** denotes the number of frames the mobility aid detector picked up an object during the object's lifetime. This parameter is useful for filtering out noise (unwanted detections), and setting up a visible count criteria for a reliable track to have at least ten detections ie.,

$$\text{totalVisibleCount} \geq 10. \tag{6.8}$$

   We also define,

$$\text{visibilty} = \frac{\text{totalVisibleCount}}{\text{age}}\,, \tag{6.9}$$

   to compute the fraction of the tracks' age for which the object was visible.

$$\text{visibilty} \geq 0.5 \tag{6.10}$$

   If the visibility (Equation 6.10) falls below a certain threshold (0.5), the track is deleted.

8. **consecutiveInvisibleCount** This variable refers to the duration (in frames) for which the mobility aid detector fails to detect the object. During that time, the object is considered 'invisible'. This parameter is useful to know if the object has left the scene by calculating,

$$\text{consecutiveInvisibleCount} > \text{Invisible for too long } (= 3\text{s}). \tag{6.11}$$

   This parameter, along with totalVisibleCount helps to identify and filter out short-lived (thus unwanted) detections.

6.1.3 Creation and Deletion of Tracks

When objects enter and leave the scene, unique identities need to be generated or removed subsequently. A newly detected object is regarded as an untracked

target and assigned with a unique ID, a bounding box and its velocity is set to zero. Since the velocity is unknown for new detections, large values are assigned to the covariance of the velocity component to reflect ambiguity. This new tracker is then added to the system for a probationary period before being associated with detections that require enough evidence (Eqs. 6.8 and 6.10) to restrict tracking of false positives.

The detector fails to detect objects when they leave or disappear from the scene, therefore, the corresponding track is terminated if no measurement is received from the mobility aid detector after $T \times FPS$ frames. The deletion limits the number of trackers and localisation errors caused by predictions over a long duration without any update from the detector. The value of T decides when a track is to be deleted from the system's memory after the object has been invisible for too long or gone undetected. This usually occurs when an object has left the scene or occluded for long periods. A small value of T results in tracks being deleted and a fresh one created if the object is re-detected, thus inflating the counting stats. The value of T is application dependent and relies on the end user's choice to count mobility aids multiple times or not. We did several trials of automated counting with values of T ranging from 1 to 5 and found that 3 is an adequate choice. In our experiments, T is set to 3 seconds allowing the detector an adequate time to re-identify the object if it undergoes occlusion. A re-identified object is later associated with the prediction of the same tracker as explained in Subsection 6.1.4. Should an object reappear after T seconds, tracking is resumed under a new identity.

6.1.4 Detection to Track Assignment

An object's gate is defined as the circular region of radius R centred at the object's centroid (x, y). Here R is the maximum distance (in pixels) an object can move from one frame to the next frame. Should an object be re-detected in the subsequent frame, it must appear inside its own gate region. The Euclidean distance is computed between an object's predicted location and all the detected objects inside the object's gate region in question. The cost (C) and weight (W) matrices are needed to associate detections with the existing tracks, and are calculated as,

$$C(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\,, \tag{6.12}$$

$$W(i,j) = 1 - \frac{1}{\max(\underline{C}) + h} \times C(i,j)\,, \tag{6.13}$$

where $i = 1, 2, 3, \ldots, M$ with M the number of tracks and $j = 1, 2, 3, \ldots, N$ with N the number of CNN detections. Here $h = 0.1$ is added to the denominator term to avoid division by zero.

In each frame, linking the set of detections to existing tracks leads to an assignment problem that is solved using the Hungarian algorithm (Kuhn 1955) so that the total cost of the assignment is minimized. The Hungarian algorithm's cost matrix is based on the distance between centroids of detections and tracks, and the type of object (pedestrian and mobility aid). When another object in the scene occludes a target, detection can only identify the occluder and updates its dynamic motion model. In contrast, the occluded object remains unaffected, unassigned and predicted according to its motion history.

6.2 Estimation of Disabled People

6.2.1 Coupling Mobility Aids and Pedestrians

The interest in linking mobility aids and pedestrians is to locate disabled pedestrians with the assumption that coupled objects move with the same speed and direction, and that, their object IDs (tags) do not change over time. Each mobility aid is paired with its nearest pedestrian within a gated area, and information is recorded in the database of pairs until the track for mobility aid is deleted. Multiple pedestrians can appear close to a mobility aid and vice versa, causing a cost minimisation and linear assignment problem. The optimal solution of this problem is obtained using the Hungarian algorithm on a cost matrix calculated from the measure of overlap (IOU) between pedestrians and mobility aids. During the video sequence, it is probable that a mobility aid is assigned to more than one pedestrian and vice versa. We define "hits" as the number of times (frames) the system couples a certain pair of mobility aid with a pedestrian. All paired instances are totalled at the end of the track, and the pair with the most hits qualifies as a reliable pair.

Table 6.2 shows a list of all pairs created during an example video sequence. In cases where the mobility aid is paired with multiple pedestrians, a reliable pair is determined by the most hits (highlighted in pink colour). The number of reliable pairs gives mobility aid count while pedestrians' total is estimated by adding up the number of reliable tracks with 'person' label. In an example shown in Table 6.2, there were 11 people detected of which six use mobility aids.

6.3 Results

The performance of the proposed system was evaluated on 126 video sequences (combined length of 328 minutes) of varying lengths and resolutions as sum-

Table 6.2: Reliable pairs example

Aid Type	Aid ID	Person ID	Hits
wheelchair	7	2	64
wheelchair	7	3	53
wheelchair	7	8	16
wheelchair	37	33	65
wheelchair	55	34	29
wheelchair	55	44	94
wheelchair	55	51	3
wheelchair	66	62	198
wheelchair	66	60	34
wheelchair	66	64	31
crutch	77	64	95
wheelchair	80	70	198

marised in Table 6.3. Test videos were recorded and captured from various locations (university, public car-park, city centre, shopping mall). Some were staged to increase counts of mobility aid usage, whereas some are not. Each video was visually inspected for pedestrians, and mobility aids count, and stats were recorded. This manual count is the total number of objects that appeared in the entire video while the system count is defined as the number of objects counted by the algorithm. The aim was to measure the accuracy and counting abilities of the tracking module compared against the human count on the same set of videos.

To measure the effectiveness of the software on individual videos, we define the following parameters

- C_m: Manual Count (human count)
- C_s: System Count (Using proposed system)
- Hits/Correctly Identified by the system,

$$H = min(C_s, C_m) \tag{6.14}$$

- False Alarm/Incorrectly Identified by the system,

$$F = max(0, C_s - C_m) \tag{6.15}$$

- Miss/Incorrectly Rejected,

$$M = max(0, C_m - C_s) \tag{6.16}$$

Table 6.3: Test video information (number of videos collected)

Resolution	Location 1	Location 2	Location 3	Location 4	Location 5	Total
1920×1080	13	84	15		1	113
1280×720					3	3
768×576			1			1
720×480			2			2
618×464			1			1
616×462			2			2
474×356			1			1
387×288				2		2
352×288					1	1
Total	**13**	**84**	**22**	**2**	**5**	**126**

Values of the above indicators for all test videos were stacked and combined to estimate the overall performance and quality of the system in terms of,

$$\text{Counting Accuracy } (\%) = \frac{\min(C_m, C_s)}{\max(C_m, C_s)} \times 100\% \tag{6.17}$$

$$\text{Hit Fraction } (\%) = \frac{H}{C_m} \times 100\% \tag{6.18}$$

$$\text{Precision } (\%) = \frac{H}{H + F} \times 100\% \tag{6.19}$$

The overall results for pedestrians and mobility aids counting given in Fig 6.1 and Table 6.4 show that automated counts are 94% accurate, score 85% hit rate and 91% precision when compared against manual counts. Mobility aids count was further broken down by the type of mobility aid (see Table 6.4) and video resolution (see Figure 6.2). Automated counting results are encouraging and demonstrate the effectiveness of the pairing strategy employed in the counting stage.

In a separate set of experiments, we recorded improvements in the counting accuracy by comparing the automated counts reported by systems with the original SORT and our proposed counting method. Table 6.5 shows the comparison of automated counts.

The counting module is developed in Python using opencv, numpy, and filterpy libraries. In order to integrate detection, counting and reporting modules, a Python script with a user interface (using PyQt) was set up. The system prompts the user to select an input video file and destination folder where the results will be saved. In the background, the system invokes a shell command to run the mobility aid detector and passes the control over to later modules (tracking, data association and counting) followed by a reporting step. The

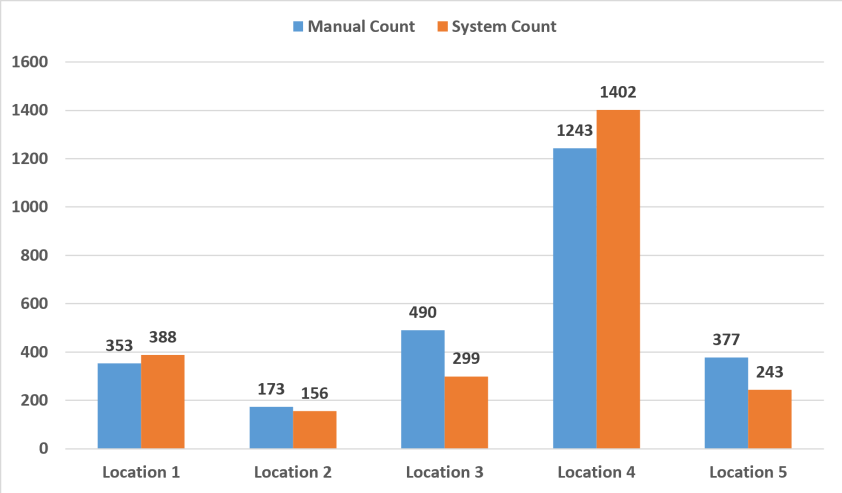


Figure 6.1: System vs Manual Counting

Table 6.4: Counting stats breakdown by mobility aid type

Type of Object	Manual Count	System Count	Accuracy	Hit Fraction	Precision
Pedestrians	2636	2488	94%	86%	91%
Mobility Aids	149	127	85%	72%	85%
Wheelchair	64	73	88%	73%	64%
Crutch	19	3	16%	11%	67%
W.Frame	5	1	20%	20%	100%
W.Stick	48	35	73%	50%	69%
M.Scooter	13	15	87%	85%	73%
Total	**2785**	**2615**	**94%**	**85%**	**91%**

system displays counting results live on top of the video frame and dynamically updates after an object enters or leaves the scene. A snapshot from the output video is shown in Figure 6.3. The pedestrians count at the top of Figure 6.3 includes healthy pedestrians and people using mobility aids.

LOCATION	Location 1	Location 2	Location 3	Location 5
RESOLUTION	1920 x 1080	1920 x 1080	474 x 356 616 x 462 618 x 464 720 x 480 768 x 576	1920 x 1080 1280 x 720 352 x 288
Person Count	91%	90%	61%	64%
Mobility Aids Count	87%	88%	47%	48%
wheelchair	79%	93%	29%	64%
crutch	11%	100%	0%	100%
W.Frame	100%	20%	100%	100%
W.Stick	84%	76%	0%	0%
M.Scooter	85%	0%	0%	100%

Figure 6.2: Counting Accuracy at different video resolutions

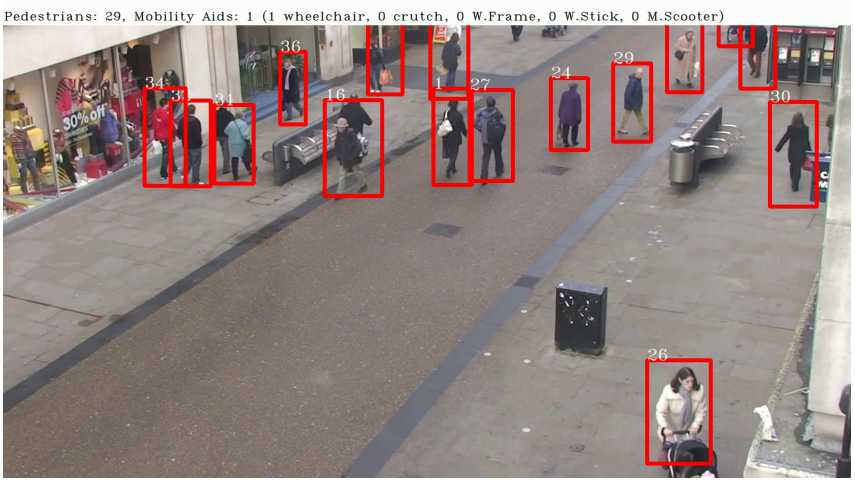
Table 6.5: Comparison of automated counts with original and enhanced SORT tracker

Location	Manual Count	System Count (with SORT)	Proposed System Count	Accuracy with SORT	Accuracy) (Proposed System)
Location 1	353	779	388	56%	82%
Location 2	173	212	156	76%	86%
Location 3	275	435	195	66%	71%
Location 5	377	698	243	56%	64%
Total	**1178**	**2126**	**982**	**61%**	**74%**

Figure 6.3: Detector screen with dynamic counters at the top

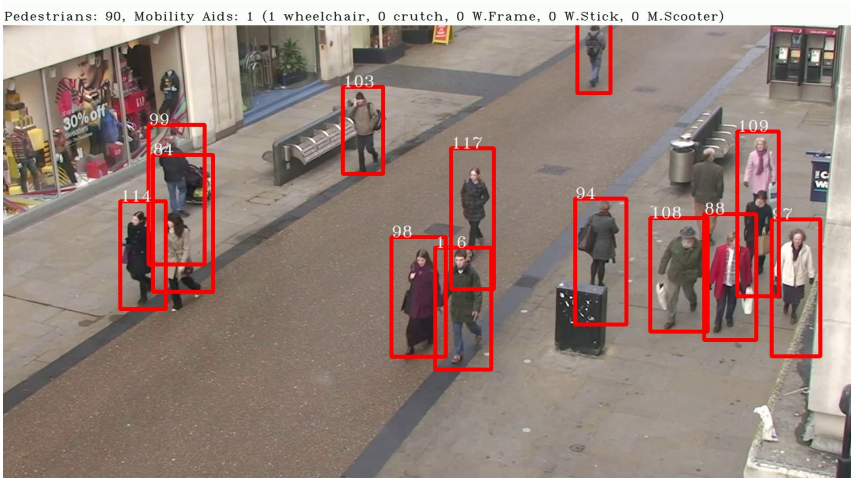
6.3.1 Correct Detections

The test videos covered a range of scenarios, including crowded scenes, occlusion, indoor, outdoor, sunny, cloudy, front-on view, side-on view, varying video resolution and distance from the camera. Detection and counting outcomes from various test videos are shown in Figures 6.4,6.5,6.6,6.7,6.8,6.9 and 6.10.



(a) Frame 1139

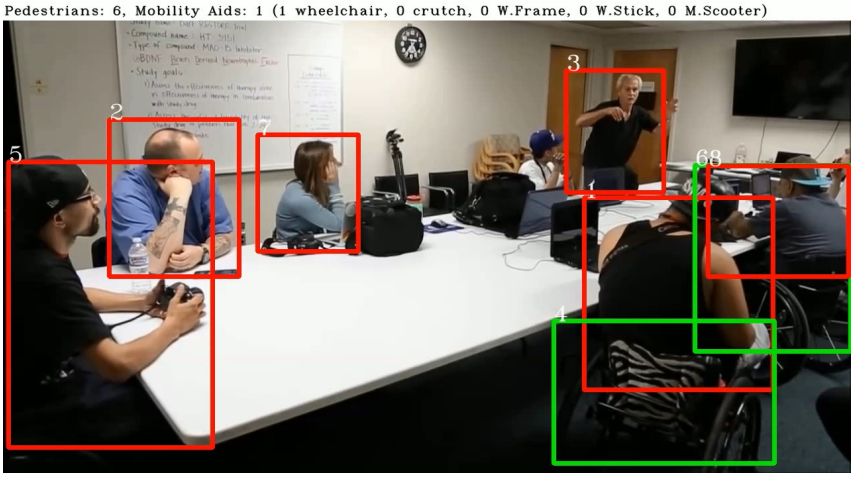
(b) Frame 5360



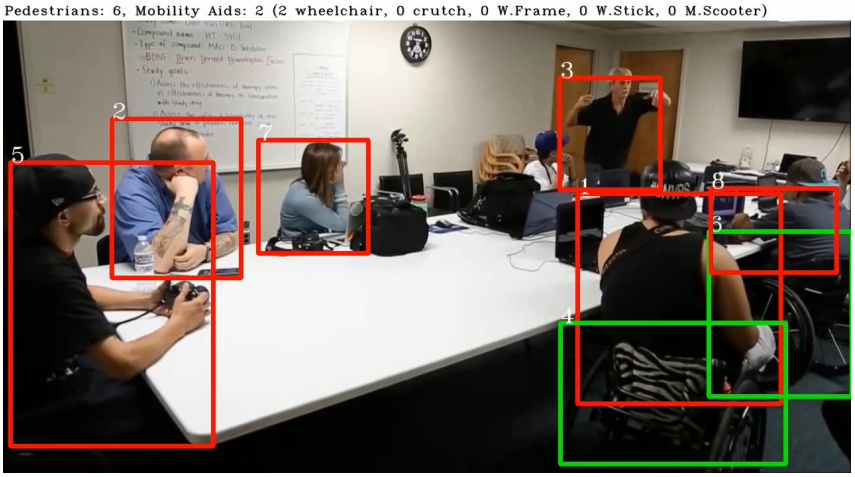
(c) Frame 6043

(d) Frame 6962

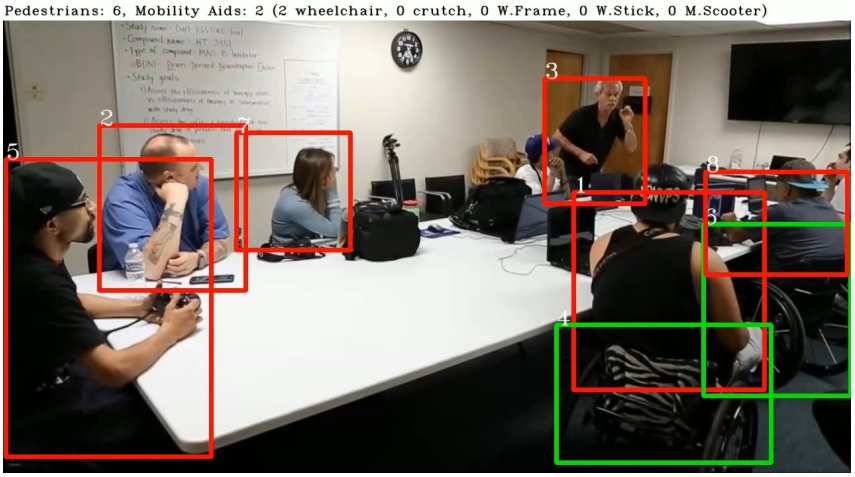
Figure 6.4: A busy outdoor public place on a cloudy day. Multiple occlusions. Video resolution is 1080p.



(a) Frame 37

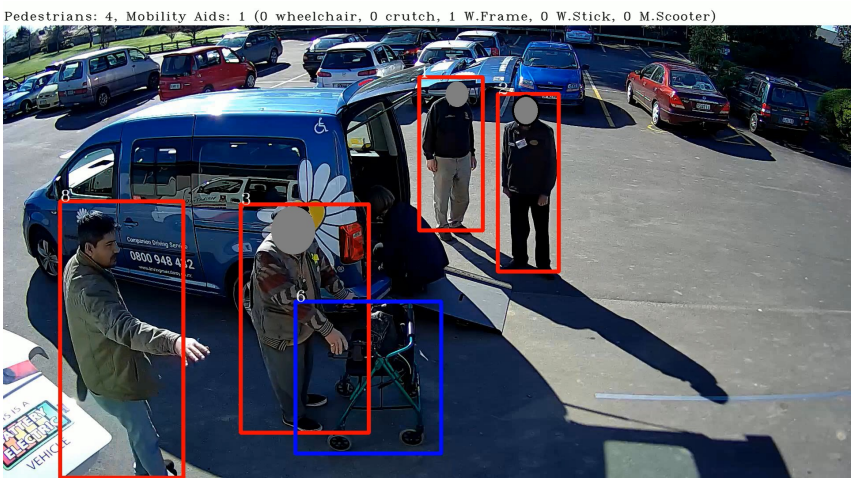


(b) Frame 80

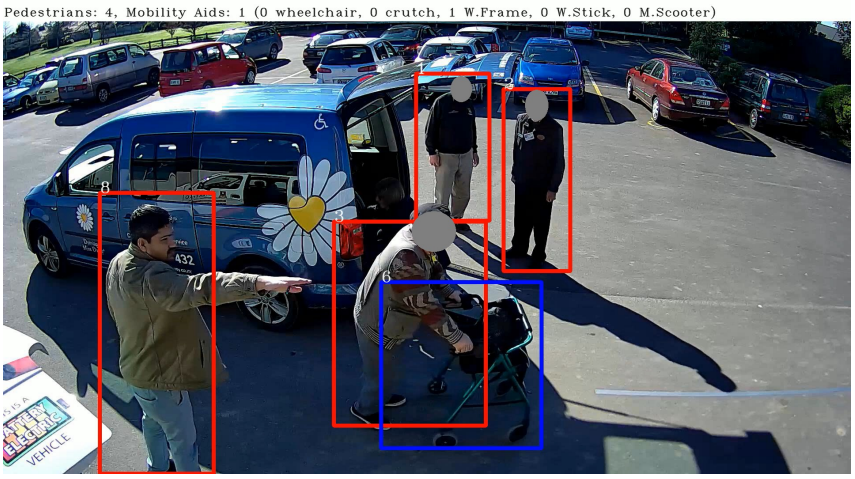


(c) Frame 187

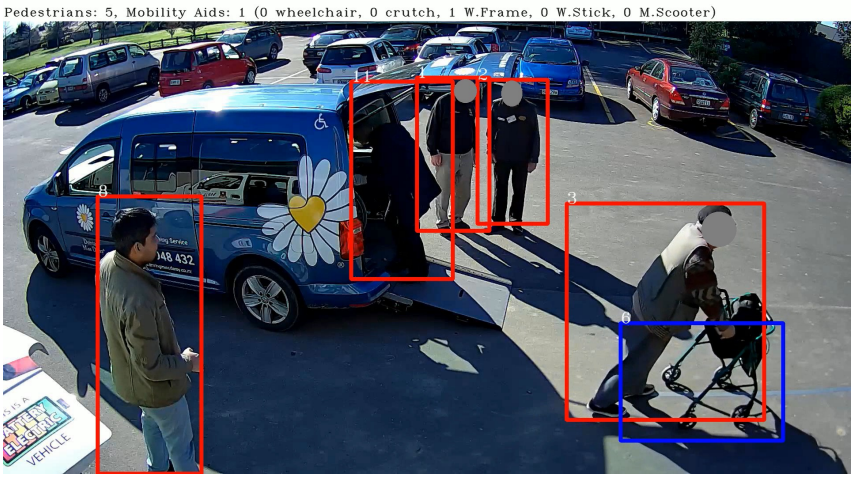
Figure 6.5: An indoor meeting place (Partial Occlusion). Video resolution is 720p.



(a) Frame 255



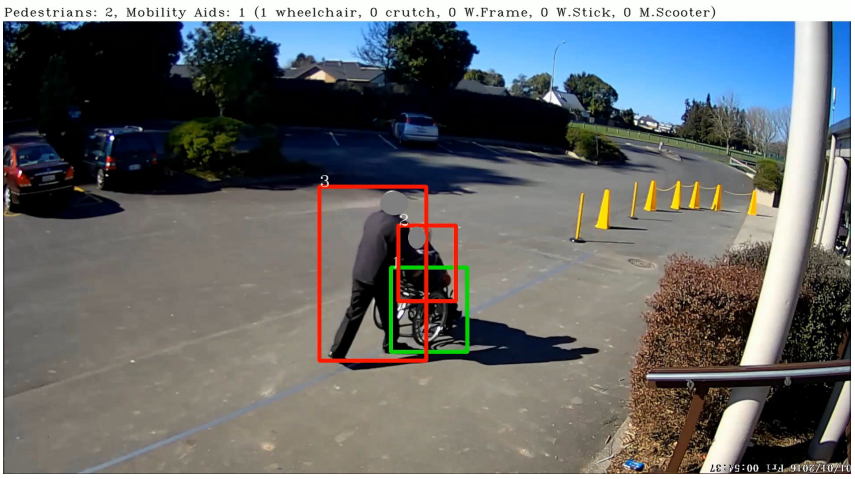
(b) Frame 324



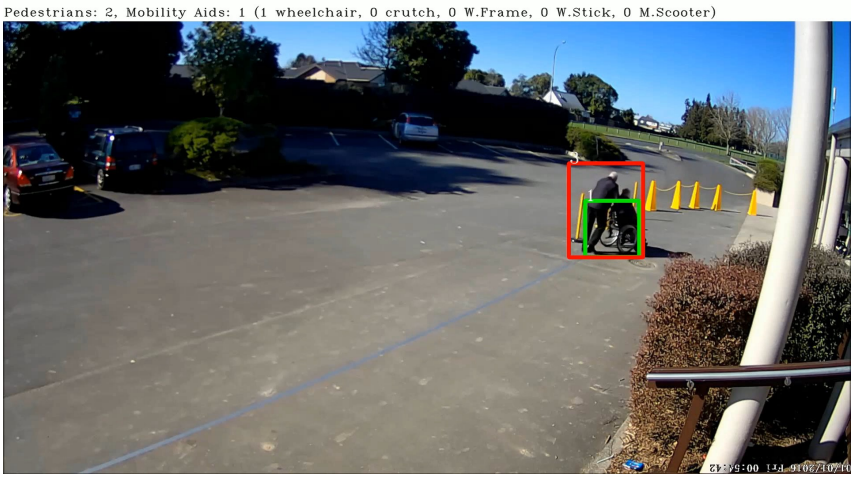
(c) Frame 451

Figure 6.6: An outdoor surveillance view on a sunny day. Side-on view. Video resolution is 1080p.

(a) Frame 17



(b) Frame 93



(c) Frame 248

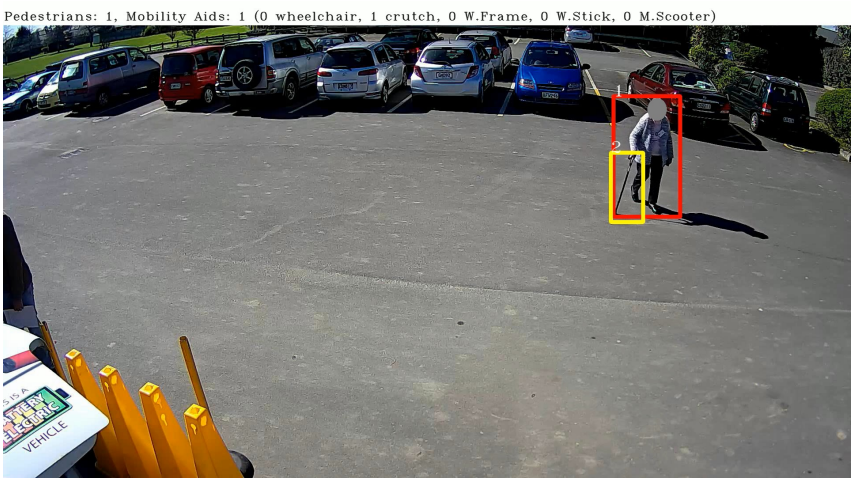
Figure 6.7: An outdoor surveillance view on a sunny day. Wheelchair user with partial occlusions is moving away from camera. Video resolution is 1080p.

(a) Frame 271

(b) Frame 436

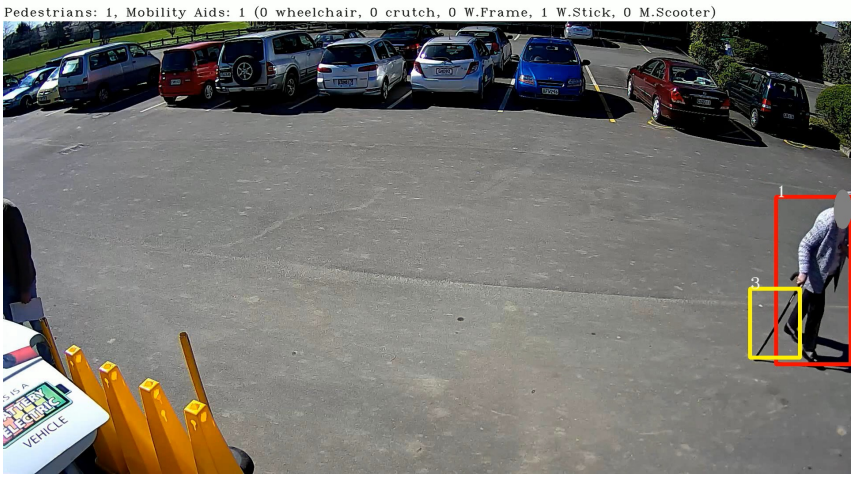
(c) Frame 537

Figure 6.8: A walking stick user in an outdoor surveillance view. Video resolution is 1080p.



(a) Frame 90

(b) Frame 144



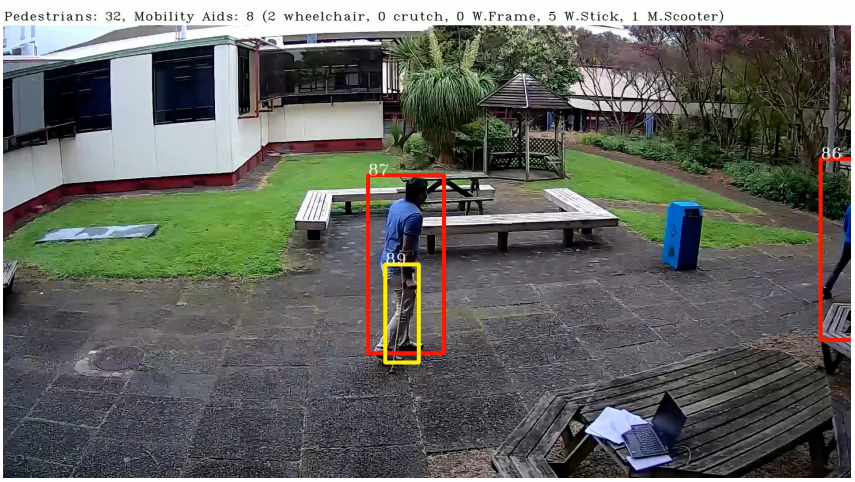
(c) Frame 225

Figure 6.9: A walking stick user in an outdoor surveillance view. Front-on view. Video resolution is 1080p.

(a) Frame 200

(b) Frame 785

(c) Frame 1582



(d) Frame 4335

(e) Frame 4443

Figure 6.10: An outdoor surveillance view on a cloudy day. Video resolution is 1080p.

6.3.2 Failures

Our system is not perfect, and there were cases when the system failed to identify a mobility aid or mixed it up with another object having an appearance similar to mobility aids. Some examples are shown in Figures 6.11,6.12,6.13,6.14 and 6.15. In Figure 6.11, the system detects a Mannequin as a real person standing on the roadside and classifies a bicycle as a wheelchair. The system also missed a walking stick linked with the pedestrian in bounding box 222. Figures 6.12 and 6.13 contain yellow prediction boxes for walking sticks, but those are false positives. A possible explanation is that thin vertical shapes such as an umbrella or pole confuse the system because they have features common with a walking stick. Similarly, the system incorrectly classified a baby stroller as a wheelchair since these objects have design similarities. In

Figure 6.15, a walking stick in the same colour as the pedestrian's clothes appears on the rear side of the walker. The mobility aid is mostly occluded, and it is only visible for a short duration during the walk cycle.

Figure 6.11: A false positive (green box) and failure in detecting mobility aids (boxes 222 and 249).

Figure 6.12: An umbrella identified as a walking stick.

Figure 6.13: A vertical shape (Tape) detected as a walking stick.

Figure 6.14: A stroller detected as a mobility scooter.

Figure 6.15: A partially occluded walking stick is not detected.

Chapter 7

Discussion

In this chapter, analysis and discussion of the results of Chapters 5 and 6 is presented. Some technical problems faced during this study are also listed with identification of their cause and possible solutions.

7.1 Detection System

YOLO was trained on a specially constructed custom database, and it shows good performance in picking mobility aids from outdoor surveillance videos (see subsection 6.3.1). An accuracy of 92% on test images is encouraging given that no similar system exists (as per the literature review, see Chapter 2) to calibrate the performance. Most test images/video frames had objects from multiple classes in it, and YOLOv3 was able to detect all sorts of mobility aids along with the pedestrians using them.

Quantity of the training image data is important to yield statistically meaningful results. Therefore, we gathered hundreds of training images for each category to build a reliable detection system. The 'total' column in Table 5.4 shows that the 'person' class has got the most number of samples (1478) among all classes in the mobility aid detector. During the dataset formation stage, 7% of the image dataset's total images comprised person images (see Table 5.1). However, the image labelling exercise revealed that pedestrian bounding boxes make up 51% of the total samples in the image dataset (see Tables 3.1 and 7.1). The high percentage of pedestrian appearances in the image dataset is due to the following two reasons:

- The images collected for person class often had multiple samples (people) in them, as opposed to most images for mobility aid classes, which had only one sample.
- Apart from the INRIA pedestrian dataset, pedestrians often appear in mobility aid images, thus increasing the sample size for the pedestrian class.

The class imbalance reflects the real-world scenarios where pedestrians are frequently sighted on public places while mobility aid users seldom appear. During the training phase, the class imbalance was avoided by limiting pedestrian images to the INRIA person dataset and ignoring images from ImageNet.

Table 7.1: Portion of images and samples of person class in the main dataset

	No. of Images	**No. of Samples**
Person Class	419	8209
Image Dataset	5819	16181
Percentage	7%	51%

False detections of the person class (see Table 5.4) are higher than false detections for other classes because the mobility aid user often appears overlapped with the mobility aid and vice versa. For example, an image with a wheelchair person has two objects; wheelchair and a person. Their annotation is two overlapped ground truth boxes with human legs and torso appearing in both bounding boxes. In case of significant overlap (IOU>0.5), the CNN can report false detections at the classification stage. Such false detections may indicate disabled pedestrians since they appear too close to the mobility aid. Training on mobility aid images without a person may not work since classifier is not trained on a realistic depiction of real-world scenarios. Therefore, our dataset is designed to contain images of mobility aids with and without a person.

There are plenty of wheelchair images available in the ImageNet database, and they make 30% of the total mobility aid images, thus outnumbering images of other individual classes. A pilot study (Burdett 2015) to manually count visible mobility aids at different parts of the Hamilton City revealed that wheelchairs (powered, manual and assisted) make up 34% of total mobility aid counts. The portion of wheelchair images in the image dataset agrees with that observed in Hamilton. Despite the focus on the wheelchair class, we can not rule out that there is a class imbalance for wheelchair versus other classes.

Few false detections caused by the objects having structures similar to those in our training dataset are shown in Figure 7.1. The yellow box with crutch prediction (in the lower right image) contains a vertical shaped structure with features that appear common in crutches. These false positives expose the imperfections in CNN based detection, but this is not a problem in our case as the proposed system is smart enough to discard falsely detected mobility aids later at the counting stage. During the counting stage, the falsely detected

mobility aids are not associated with a person in the scene, thus do not qualify as a reliable pair. They are therefore correctly not counted.

We believe that the actual frames per second (FPS) could be higher as YOLO run-time parameters were not fully optimised. Our original work was based on retraining YOLO's version 2 for detecting mobility aids, but during experiments, YOLOv3 was released, so we upgraded the system's current state to version 3. Results in Table 5.3 show that version 3 performs slightly better than version 2.

The current object detectors RCNN (Girshick et al. 2015), Faster RCNN (Ren et al. 2015), YOLO (Redmon & Farhadi 2018), and SSD (Liu et al. 2016) have demonstrated notable performances, and it is hard to choose one without hit and trial given the custom dataset. YOLO was picked as it was the fastest object detector when this study was carried out and had accuracy comparable with modern object detectors. We believe that a different object detector may give superior performance when trained on the same dataset. The number of false detections can be reduced, and counting accuracy may improve.

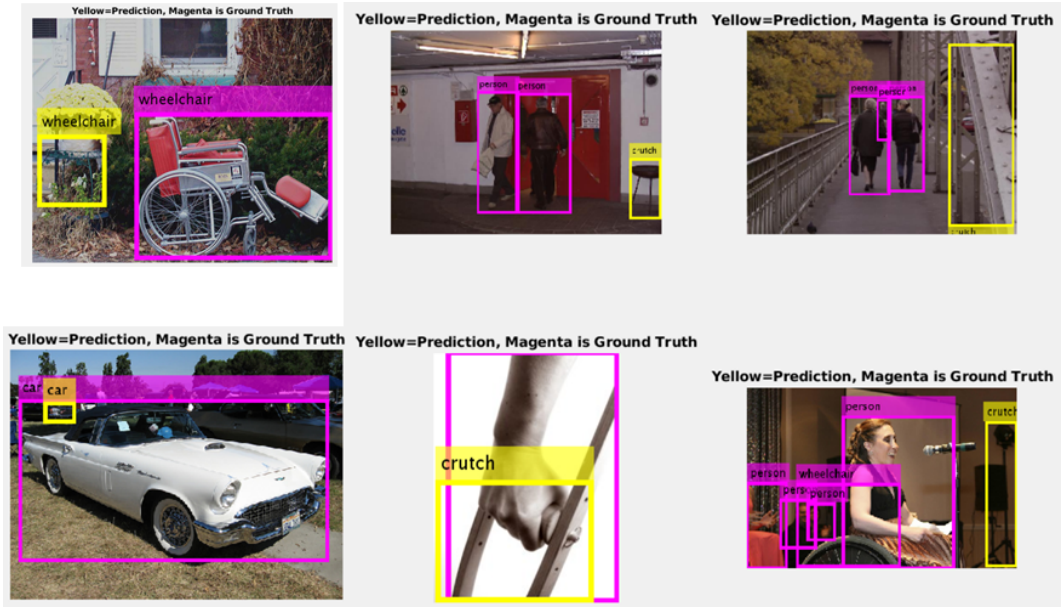


Figure 7.1: Incorrect Detections

7.2 Data Association and Counting System

The system is able to identify disabled pedestrians by detecting mobility aids from a range of different surveillance videos as shown in Subsection 6.3.1. The solution works for indoor and outdoor surveillance videos independent of camera viewing angle, weather conditions (sunny and cloudy) and time of the day. In the data association part, the detected objects' movement patterns

are analysed over time to form pairs with a mobility aid and a person. The algorithm can handle multi-class, multi-object tracking and data association problem. The default implementation of SORT algorithm uses the IOU metric to calculate the cost and weight matrices (Section 6.1.4). The accelerating objects often appear distant from their bounding boxes in the previous frame, resulting in a low IOU score between their previous and current bounding boxes. Therefore, accelerating objects are often not assigned to their tracks at the data association stage. Any unassigned detection is perceived as a new object entering the scene assigned with a fresh track, and thus the object count is incremented, resulting in surplus counts. The surplus count problem was solved by incorporating the distance feature to calculate the cost and weight matrices (Equations 6.12 and 6.13). This improvement in detection to track assignment empowers the system to handle accelerating objects and avoids assigning multiple identities to one object.

The counting module, especially with pairing approach, is robust and avoids over-counts due to identity switches and partial occlusions. Results in Table 6.4 demonstrate that the system has satisfactory performance when evaluated on outdoor surveillance videos, and an important step towards developing a similar system. Below we discuss the performance of the tracking, data association and counting modules, and factors affecting their performance.

Tracking and data association is responsible for retaining the object's identity until it leaves the scene. The assignment cost matrix has a crucial role in assigning the detections to actual tracks. IOU based cost matrix by Bewley et al. (2016) was initially tested, but results were significantly different from the ground truth. This failure is attributed to occlusions and rapid acceleration of objects in the scene, creating new IDs for pre-existing targets and inflating the counting figures. One fix is to replace IOU with the distance between centroids feature and applying a cut-off distance value to reject candidates far away from the object of interest. This may result in identity switch in some cases, but the overall count is not affected as displayed in Table 6.4.

False Positives occurred due to the uncontrolled cluttered outdoor environment (see Figures 6.11 and 6.14). The tracking algorithm can partially correct the detection failures and momentary loss of detected objects. Detection and counting accuracy is highly dependent on the detection algorithm, and these algorithms are not perfect. Data association and counting modules do not see mobility aids in isolation, and a corresponding reliable pair must exist to validate the detection. A reliable pair comprises a pedestrian and a mobility aid sharing a common velocity based on the fact that disabled pedestrians and their mobility aids lie nearby and have the same motion vector. In most cases, an object incorrectly identified as a mobility aid, will not lie close to the same

pedestrian throughout the scene and will have a different motion signature. Therefore, it will not be falsely counted. A solid counting accuracy of 94% and 91% precision show great confidence in system performance. We believe that the following reasons are responsible for the 6% of counting error,

- Occlusion in general and occlusion of sticks/crutches in particular. In some cases, the system fails to detect a stick/crutch from videos with the side-on view when mobility aid happens to be occluded on the rear side of the person (see Figure 6.15). However, the same mobility aid is detected when exposed to the camera from the same pedestrian's front side.

- The shape of crutch and stick are very much alike, and therefore, the system confuses one with the other. It does not affect the final counts as the aim was to correctly detect a mobility aid of any type and determining the exact kind of mobility aid was not a design goal.

- Crutch/Stick appears like the human leg's shape, especially like a slim person in the standing pose. On some occasions, the detection system incorrectly identifies vertical lines or shapes as a crutch/stick (for example Figures 6.12, 6.13 and 7.1).

- Similar structures in a wheelchair, walking frame and mobility scooter, cause inaccuracies in detection types.

- Imperfections in CNN based object detection algorithms can introduce under-counts and over-counts. Under-counts are the result of total failure in detecting a particular object. On the other hand, the over-counts are due to the system's inability to identify the same object in intermediate frames. Overcounts often involve an object being assigned with multiple IDs and tracked several times before leaving the scene.

- The camera view can impact the performance of a detector. Our experiments show that side on view provides better results than the front on view.

- Low-resolution videos (below 720p) can cause detection errors due to the loss of visual information required for characterising an object. This is because low-resolution images tend to capture objects with fewer pixels resulting in loss of information at the pixel level. The same is true for objects appearing in small sizes in high-resolution videos. Therefore, running the detection algorithm on too small objects may not give desirable results. We observed that the detection accuracy is affected by the

low resolution (see Figure 6.2). In our experiments, Location 3 and 5 had low-resolution videos, and we noticed a drop in the detection accuracy.

- The manual count is not truly correct as different people report different counts for densely populated videos. During the manual counting experiments, a test video was counted by two people, and both reported different counts. Three rounds of manual counting were carried out, and the final count was determined by averaging the values obtained in three rounds. Table 7.2 shows a 15% difference between the number of pedestrians counted by two different people analysing a five minutes long video.

Table 7.2: Manual counting of pedestrians in a crowded video by two different observers

	Person 1	**Person 2**
Round 1	276	315
Round 2	287	309
Round 3	275	335
Average	**279**	**320**

- Absence of appearance features in the tracker's motion model causes identity switches within the class. An improved tracker may enhance performance, but dealing with tracking issues is considered outside the scope of this thesis.

The performance comparison of the proposed and SORT based counting systems showed that the proposed method has a better counting accuracy than SORT (Table 6.5). SORT often inflates the total object count by issuing multiple new IDs for the same accelerating object. The SORT users have reported the inflated counting issue on the GitHub webpage[1] [2]. Our tracking strategy based on a combination of object gate, Euclidean distance metric and object class avoids creating unwanted object IDs. The advantage of using object class information in the cost matrix is that it prevents the assignment of objects from across the different classes at the data association stage and empowers the system to perform multi-class tracking. Performance comparison results from Location 4 are not reported in Table 6.5 as test videos are of low resolution (see Table 6.3), and the SORT tracker compensates for false

[1]https://github.com/abewley/sort/issues/109
[2]https://github.com/abewley/sort/issues/137

negatives by YOLO at the detection stage. In this case, the inflated counts by SORT bring the system counts close to the manual result, which should have been lower otherwise. Therefore, reporting such a result would be an unfair comparison of SORT with the proposed system.

Chapter 8

Conclusion

This research aimed to detect and count mobility aids in surveillance videos. Identifying disabled pedestrians in videos is difficult due to limited research on this topic. Analysis of a person's gait signal showed ample information in the gait signal to differentiate between healthy and disabled pedestrians. Investigating the stride length and velocity features revealed that the walker's leg's gait signals show abnormality when extracted for a disabled person. The automated gait recognition procedures are affected by the variation in environmental constraints, viewing angle, occlusions, shadows, an imperfection in foreground modelling, object segmentation and silhouette extraction. The automated procedure failed to recognise a disabled person from its gait due to poor localisation by YOLO, incorrect segmentation and silhouette extraction due to moving backgrounds and shadows.

We gathered 5819 images of five mobility aids (wheelchair, crutch, walking stick, walking frame and mobility scooter) to establish a database used to develop a CNN-based solution for identifying mobility aids in images and videos. Our self collected and labelled image dataset for mobility aid detection is unique and has a sufficient number of training examples to cover variations in viewing angles, shape and size of mobility aids. The dataset has proved adequate for training a mobility aid detector with reliable accuracy. Counting results indicate that the mobility aid detector was able to pick the majority of mobility aids that appeared in the test videos.

We conclude that a CNN trained on images of mobility aids can detect mobility aids in real-world videos with satisfactory performance. The successful operation of the mobility aid detector on surveillance videos solves the detection part of the research problem. To design a mobility aid detector, YOLOv3 can be customised and trained on five mobility aid classes. The system successfully detects all five mobility aids with an occasional mix-up of crutch/stick and wheelchair/walking frame classes because of their similar build. The proposed system reports 0.89 precision and 0.92 recall on test images, and it is capable of detecting multiple objects in one image/frame.

In this study, the data association algorithm (SORT) is enhanced to deal with objects from multiple classes and tackle accelerating objects. The originalSORT algorithm only deals with single class objects. Using the distance metric instead of IOU score to construct the cost matrix can improve theSORT algorithm for tackling the accelerating objects.

A pairing strategy is also proposed to count disabled pedestrians by associating a mobility aid with the nearest pedestrian. A pair of mobility aid and a person who remained in each other's proximity for most of their track's age is considered reliable. Only that can contribute towards the total counts for disabled pedestrians. Results indicate a counting accuracy of 94%, considering that it is the first-ever system built for counting mobility aids. In addition to counting mobility aids, counting success rate for pedestrians is 94% and comparable to any existing pedestrian detection system.

8.1 Future Work

The current system is designed first to identify mobility aid and then associates it with a person who uses it in most cases. It is a two-step process and involves considerable work at the data association and counting stage. Training a CNN (Ren et al. 2015, Redmon & Farhadi 2018) on images of mobility aid users labelled as 'disabled person' might also solve the research problem of detecting disabled pedestrians without a complicated counting module. Since the mobility aids image dataset will be available after this study, the proposed method can be explored. In this study, the final counting error depends on the detection and counting errors. The new approach may reduce the error down to detection errors only and can improve the counting accuracy. Figures 8.1 and 8.2 show a couple of database images used in this study and for proposed future work respectively.

Manual extraction of human gait signal produced interesting gait signals capturing legs movement. Although our automated approach could not extract the gait signal, human joints detection can significantly improve gait signal, leading to autonomous detection of disabled persons. CNN based human pose estimation (Cao et al. 2018) has shown encouraging results in identifying joint locations and connecting them to other body parts of the same person. Such algorithms can predict confidence maps for body part detection and associating those together forming a skeleton. CNN based solutions are worth exploring to solve problems that classical computer vision methods cannot due to segmentation and shadow issues.

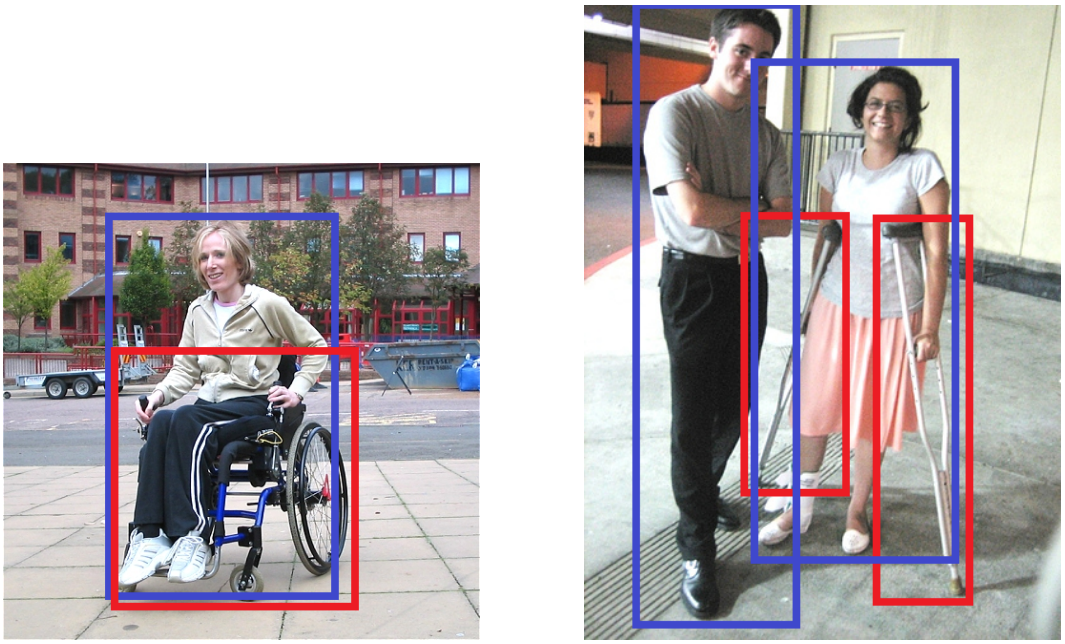


Figure 8.1: Images labelled with mobility aid and person classes

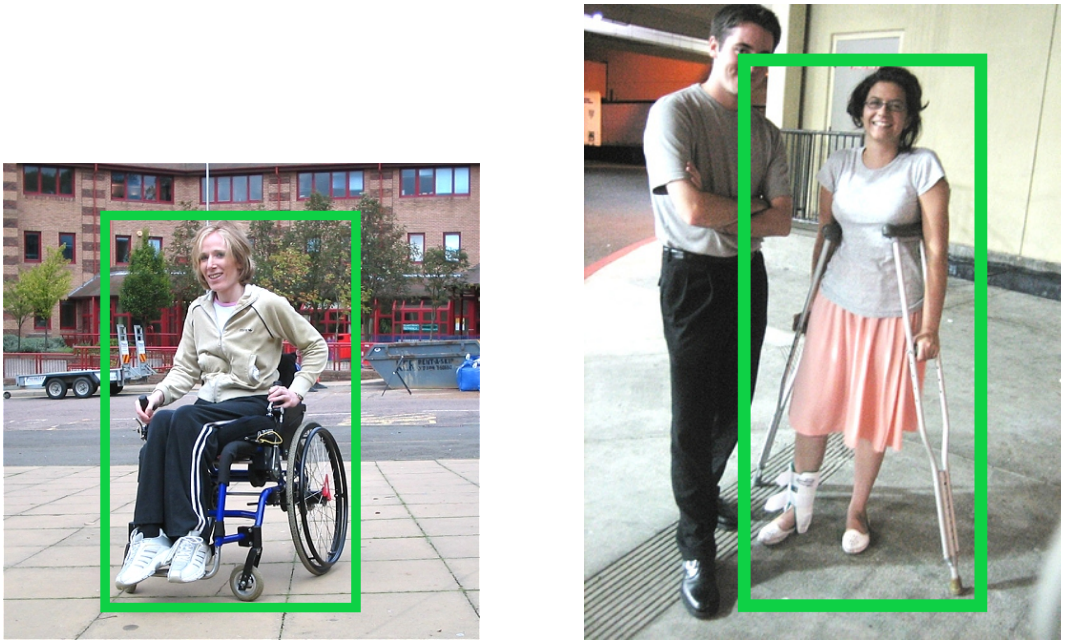


Figure 8.2: Images labelled with 'disabled pedestrian' class

References

Afsar, P., Cortez, P. & Santos, H. (2015), 'Automatic visual detection of human behavior: A review from 2000 to 2014', *Expert Systems with Applications* **42**(20), 6935–6956.

Aghdam, H. H. & Heravi, E. J. (2017), *Guide to convolutional neural networks: a practical application to traffic-sign detection and classification*, Springer.

Andriluka, M., Roth, S. & Schiele, B. (2009), Pictorial structures revisited: People detection and articulated pose estimation, *in* '2009 IEEE conference on computer vision and pattern recognition', IEEE, pp. 1014–1021.

Bahrampour, S., Ramakrishnan, N., Schott, L. & Shah, M. (2015), 'Comparative study of deep learning software frameworks', *arXiv preprint arXiv:1511.06435* .

Bar-Shalom, Y., Willett, P. K. & Tian, X. (2011), *Tracking and data fusion*, YBS publishing Storrs, CT, USA:.

BenAbdelkader, C., Cutler, R. & Davis, L. (2002*a*), 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, pp. 372–377.

BenAbdelkader, C., Cutler, R. & Davis, L. (2002*b*), *View-invariant Estimation of Height and Stride for Gait Recognition*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 155–167.

BenAbdelkader, C., Cutler, R., Nanda, H. & Davis, L. (2001), Eigengait: Motion-based recognition of people using image self-similarity, *in* 'International conference on audio-and video-based biometric person authentication', Springer, pp. 284–294.

Benenson, R., Mathias, M., Timofte, R. & Gool, L. V. (n.d.), Pedestrian detection at 100 frames per second, *in* 'Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on', pp. 2903–2910.

Benezeth, Y., Jodoin, P.-M., Emile, B., Laurent, H. & Rosenberger, C. (2010), 'Comparative study of background subtraction algorithms', *Journal of Electronic Imaging* **19**(3), 033003–033003–12.

Bergmann, P., Meinhardt, T. & Leal-Taixe, L. (2019), Tracking without bells and whistles, *in* 'Proceedings of the IEEE International Conference on Computer Vision', pp. 941–951.

Bewley, A., Ge, Z., Ott, L., Ramos, F. & Upcroft, B. (2016), Simple online and realtime tracking, *in* '2016 IEEE International Conference on Image Processing (ICIP)', IEEE, pp. 3464–3468.

Bewley, A., Guizilini, V., Ramos, F. & Upcroft, B. (2014), Online self-supervised multi-instance segmentation of dynamic objects, *in* '2014 IEEE International Conference on Robotics and Automation (ICRA)', IEEE, pp. 1296–1303.

Bouchrika, I., Carter, J. N. & Nixon, M. S. (2016), 'Towards automated visual surveillance using gait for identity recognition and tracking across multiple non-intersecting cameras', *Multimedia Tools and Applications* **75**(2), 1201–1221.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. & Shah, R. (1994), Signature verification using a" siamese" time delay neural network, *in* 'Advances in neural information processing systems', pp. 737–744.

Burdett, B. (2013), Measuring accessible journeys, Report, Traffic Design Group.

Burdett, B. (2015), Measuring accessible journeys: a tool to enable participation, *in* 'Proceedings of the Institution of Civil Engineers-Municipal Engineer', Vol. 168, Thomas Telford Ltd, pp. 125–132.

Candamo, J., Shreve, M., Goldgof, D. B., Sapper, D. B. & Kasturi, R. (2010), 'Understanding transit scenes: A survey on human behavior-recognition algorithms', *IEEE Transactions on Intelligent Transportation Systems* **11**(1), 206–224.

Cao, S. & Nevatia, R. (2016), Exploring deep learning based solutions in fine grained activity recognition in the wild, *in* '2016 23rd International Conference on Pattern Recognition (ICPR)', IEEE, pp. 384–389.

Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E. & Sheikh, Y. (2018), 'Openpose: realtime multi-person 2d pose estimation using part affinity fields', *arXiv preprint arXiv:1812.08008* .

CASIA Gait Database (2005).
URL: *http://www.cbsr.ia.ac.cn/english/Gait%20Databases.asp*

Castro, F. M., Marín-Jimenez, M. J. & Medina-Carnicer, R. (2014), Pyramidal fisher motion for multiview gait recognition, *in* '2014 22nd International Conference on Pattern Recognition', IEEE, pp. 1692–1697.

CAVIAR dataset 1 (2004).

Chen, D.-Y. & Huang, P.-C. (2011), 'Motion-based unusual event detection in human crowds', *Journal of Visual Communication and Image Representation* **22**(2), 178–186.
URL: *http://www.sciencedirect.com/science/article/pii/S1047320310001549*

Chen, D. Y. & Huang, P. C. (2013), 'Visual-based human crowds behavior analysis based on graph modeling and matching', *IEEE Sensors Journal* **13**(6), 2129–2138.

Choi, W. (2015), Near-online multi-target tracking with aggregated local flow descriptor, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 3029–3037.

Choudhury, S. D. & Tjahjadi, T. (2015), 'Robust view-invariant multiscale gait recognition', *Pattern Recognition* **48**(3), 798–811.

Ciaparrone, G., Sánchez, F. L., Tabik, S., Troiano, L., Tagliaferri, R. & Herrera, F. (2020), 'Deep learning in video multi-object tracking: A survey', *Neurocomputing* **381**, 61–88.

Conde, C., Moctezuma, D., De Diego, I. M. & Cabello, E. (2013), 'Hogg: Gabor and hog-based human detection for surveillance in non-controlled environments', *Neurocomputing* **100**, 19–30.

Cunado, D., Nixon, M. S. & Carter, J. N. (1997), Using gait as a biometric, via phase-weighted magnitude spectra, *in* 'International Conference on Audio- and Video-Based Biometric Person Authentication', Springer, pp. 93–102.

Cutler, R. & Davis, L. S. (2000), 'Robust real-time periodic motion detection, analysis, and applications', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8), 781–796.

Dalal, N. (2006), Finding People in Images and Videos, Thesis.

Dalal, N. & Triggs, B. (2005), Histograms of oriented gradients for human detection, *in* '2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)', Vol. 1, IEEE, pp. 886–893.

Dalal, N., Triggs, B. & Schmid, C. (2006), Human detection using oriented histograms of flow and appearance, *in* 'European conference on computer vision', Springer, pp. 428–441.

Davis, J. W. & Bobick, A. F. (1997), The representation and recognition of human movement using temporal templates, *in* 'Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition', IEEE, pp. 928–934.

De Winter, J. & Wagemans, J. (2004), 'Contour-based object identification and segmentation: Stimuli, norms and data, and software tools', *Behavior Research Methods, Instruments, & Computers* **36**(4), 604–624.

Deepika, K., Jyostna, D. B. & N, V. (2019), 'Effect of different kernels on the performance of an svm based classification', *International Journal of Recent Technology and Engineering IJRTE* **7**(5), 1–6.

Deng, M. & Wang, C. (2018), 'Human gait recognition based on deterministic learning and data stream of microsoft kinect', *IEEE Transactions on Circuits and Systems for Video Technology* **29**(12), 3636–3645.

Dicle, C., Camps, O. I. & Sznaier, M. (2013), The way they move: Tracking multiple targets with similar appearance, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 2304–2311.

Dollar, P., Wojek, C., Schiele, B. & Perona, P. (2011), 'Pedestrian detection: An evaluation of the state of the art', *IEEE transactions on pattern analysis and machine intelligence* **34**(4), 743–761.

Dollar, P., Wojek, C., Schiele, B. & Perona, P. (2012), 'Pedestrian detection: An evaluation of the state of the art', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(4), 743–761.

Doulamis, N. (2018), 'Adaptable deep learning structures for object labeling/tracking under dynamic visual environments', *Multimedia Tools and Applications* **77**(8), 9651–9689.

Enzweiler, M. & Gavrila, D. M. (2008), 'Monocular pedestrian detection: Survey and experiments', *IEEE transactions on pattern analysis and machine intelligence* **31**(12), 2179–2195.

Ess, A., Leibe, B., Schindler, K., & van Gool, L. (2008), A mobile vision system for robust multi-person tracking, *in* 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)', IEEE Press.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. (2010), 'The pascal visual object classes (voc) challenge', *International journal of computer vision* **88**(2), 303–338.

Farlex (2012), 'Gait cycle'. Medical Dictionary for the Health Professions and Nursing, Accessed: December 3 2020.
URL: *https://medical-dictionary.thefreedictionary.com/gait cycle*

Felzenszwalb, P. & Huttenlocher, D. (2004), Distance transforms of sampled functions, Report, Cornell University.

Ferryman, J. & Shahrokni, A. (2009), Pets2009: Dataset and challenge, *in* '2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance', pp. 1–6.

Figueira, D., Taiana, M., Nambiar, A., Nascimento, J. & Bernardino, A. (2014), The hda+ data set for research on fully automated re-identification systems, *in* 'European Conference on Computer Vision', Springer, pp. 241–255.

Fujiyoshi, H. & Lipton, A. J. (1998), Real-time human motion analysis by image skeletonization, *in* 'Proceedings of the 4th IEEE Workshop on Applications of Computer Vision (WACV'98)', p. 15.

Fujiyoshi, H., Lipton, A. J. & Kanade, T. (2004), 'Real-time human motion analysis by image skeletonization', *IEICE TRANSACTIONS on Information and Systems* **87**(1), 113–120.

Gardner, M. W. & Dorling, S. (1998), 'Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences', *Atmospheric environment* **32**(14-15), 2627–2636.

Gavrila, D. M. (2007), 'A bayesian, exemplar-based approach to hierarchical shape matching', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(8), 1408–1421.

Geiger, A., Lauer, M., Wojek, C., Stiller, C. & Urtasun, R. (2013), '3D traffic scene understanding from movable platforms', *IEEE transactions on pattern analysis and machine intelligence* **36**(5), 1012–1025.

Geiger, J. T., Kneißl, M., Schuller, B. W. & Rigoll, G. (2014), Acoustic gait-based person identification using hidden markov models, *in* 'Proceedings of the 2014 Workshop on Mapping Personality Traits Challenge and Workshop', pp. 25–30.

Gerónimo, D., Sappa, A. D., López, A. & Ponsa, D. (2007), Adaptive image sampling and windows classification for on-board pedestrian detection, *in* 'International Conference on Computer Vision Systems: Proceedings (2007)'.

Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2015), 'Region-based convolutional networks for accurate object detection and segmentation', *IEEE transactions on pattern analysis and machine intelligence* **38**(1), 142–158.

Godoy, D. (2018), 'Understanding binary cross-entropy/log loss: a visual explanation', *Towards datascience* .

Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. (2016), *Deep learning*, Vol. 1, MIT press Cambridge.

Gross, R. & Shi, J. (2001), The cmu motion of body (mobo) database, Technical Report CMU-RI-TR-01-18, Carnegie Mellon University, Pittsburgh, PA.

Guo, L., Ge, P.-S., Zhang, M.-H., Li, L.-H. & Zhao, Y.-B. (2012), 'Pedestrian detection for intelligent transportation systems combining adaboost algorithm and support vector machine', *Expert Systems with Applications* **39**(4), 4274–4286.

Hadji, I. & Wildes, R. P. (2018), 'What do we understand about convolutional networks?', *arXiv preprint arXiv:1803.08834* .

Hamid Rezatofighi, S., Milan, A., Zhang, Z., Shi, Q., Dick, A. & Reid, I. (2015), Joint probabilistic data association revisited, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 3047–3055.

Han, J. & Bhanu, B. (2006), 'Individual recognition using gait energy image', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **28**(2), 316–322.

Hofmann, M., Geiger, J., Bachmann, S., Schuller, B. & Rigoll, G. (2014), 'The tum gait from audio, image and depth (gaid) database: Multimodal recognition of subjects and traits', *Journal of Visual Communication and Image Representation* **25**(1), 195–206.

Hou, C., Ai, H. & Lao, S. (2007), Multiview pedestrian detection based on vector boosting, *in* 'Asian Conference on Computer Vision', Springer, pp. 210–219.

Hu, M., Wang, Y., Zhang, Z. & Zhang, D. (2011), 'Gait-based gender classification using mixed conditional random field', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **41**(5), 1429–1439.

Huang, C.-R., Chung, P.-C., Lin, K.-W. & Tseng, S.-C. (2009), 'Wheelchair detection using cascaded decision tree', *IEEE Transactions on Information Technology in Biomedicine* **14**(2), 292–300.

Hussain, S. U. & Triggs, W. (2010), Feature sets and dimensionality reduction for visual object detection.

Iwama, H., Okumura, M., Makihara, Y. & Yagi, Y. (2012), 'The ou-isir gait database comprising the large population dataset and performance evaluation of gait recognition', *IEEE Transactions on Information Forensics and Security* **7**(5), 1511–1521.

Jang, G., Park, J. & Kim, M. (2016), Cascade-adaboost for pedestrian detection using hog and combined features, *in* 'Advances in Computer Science and Ubiquitous Computing', Springer, pp. 430–435.

Jin, S., RoyChowdhury, A., Jiang, H., Singh, A., Prasad, A., Chakraborty, D. & Learned-Miller, E. (2018), Unsupervised hard example mining from videos for improved object detection, *in* 'Proceedings of the European Conference on Computer Vision (ECCV)', pp. 307–324.

Johnson, A. Y. & Bobick, A. F. (2001), A multi-view method for gait recognition using static body parameters, *in* 'International Conference on Audio- and Video-Based Biometric Person Authentication', Springer, pp. 301–311.

Kale, A., Chowdhury, A. R. & Chellappa, R. (2003), Towards a view invariant gait recognition algorithm, *in* 'Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.', IEEE, pp. 143–150.

Kale, A., Rajagopalan, A., Cuntoor, N. & Kruger, V. (2002), Gait-based recognition of humans using continuous hmms, *in* 'Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition', IEEE, pp. 336–341.

Kalman, R. E. (1960), 'A New Approach to Linear Filtering and Prediction Problems', *Journal of Basic Engineering* **82**(1), 35–45.
URL: *https://doi.org/10.1115/1.3662552*

Kass, M., Witkin, A. & Terzopoulos, D. (1988), 'Snakes: Active contour models', *International journal of computer vision* **1**(4), 321–331.

Kellokumpu, V., Zhao, G., Li, S. Z. & Pietikäinen, M. (2009), *Dynamic texture based gait recognition*, Springer, pp. 1000–1009.

Khan, Z., Balch, T. & Dellaert, F. (2005), 'Mcmc-based particle filtering for tracking a variable number of interacting targets', *IEEE transactions on pattern analysis and machine intelligence* **27**(11), 1805–1819.

Khan, Z., Balch, T. & Dellaert, F. (2006), 'Mcmc data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements', *IEEE transactions on pattern analysis and machine intelligence* **28**(12), 1960–1972.

Kim, C., Li, F., Ciptadi, A. & Rehg, J. M. (2015), Multiple hypothesis tracking revisited, *in* 'Proceedings of the IEEE International Conference on Computer Vision', pp. 4696–4704.

Kim, M., Alletto, S. & Rigazio, L. (2016), 'Similarity mapping with enhanced siamese network for multi-object tracking', *arXiv preprint arXiv:1609.09156* .

Kimura, N., Yoshinaga, I., Sekijima, K., Azechi, I. & Baba, D. (2020), 'Convolutional neural network coupled with a transfer-learning approach for time-series flood predictions', *Water* **12**(1), 96.

Kuhn, H. W. (1955), 'The Hungarian method for the assignment problem', *Naval research logistics quarterly* **2**(1-2), 83–97.

Lee, B., Erdenee, E., Jin, S., Nam, M. Y., Jung, Y. G. & Rhee, P. K. (2016), Multi-class multi-object tracking using changing point detection, *in* 'European Conference on Computer Vision', Springer, pp. 68–83.

Lee, L. & Grimson, W. E. L. (2002), Gait analysis for recognition and classification, *in* 'Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition', IEEE, pp. 155–162.

Lee, T. K. M., Belkhatir, M. & Sanei, S. (2013), 'A comprehensive review of past and present vision-based techniques for gait recognition', *Multimedia Tools and Applications* **72**(3), 2833–2869.

Lin, C., Lu, J. & Zhou, J. (2019), 'Multi-grained deep feature learning for robust pedestrian detection', *IEEE Transactions on Circuits and Systems for Video Technology* **29**(12), 3608–3621.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B. & Belongie, S. (2017), Feature pyramid networks for object detection, *in* 'Proceedings of

the IEEE conference on computer vision and pattern recognition', pp. 2117–2125.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. & Berg, A. C. (2016), SSD: Single shot multibox detector, *in* 'European conference on computer vision', Springer, pp. 21–37.

Liu, W., Liao, S. & Hu, W. (2019), 'Efficient single-stage pedestrian detector by asymptotic localization fitting and multi-scale context encoding', *IEEE transactions on image processing* **29**, 1413–1425.

Long, J., Shelhamer, E. & Darrell, T. (2015), Fully convolutional networks for semantic segmentation, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 3431–3440.

López-Fernández, D., Madrid-Cuevas, F. J., Carmona-Poyato, Á., Marín-Jiménez, M. J. & Muñoz-Salinas, R. (2014), The ava multi-view dataset for gait recognition, *in* 'International workshop on activity monitoring by multiple distributed sensing', Springer, pp. 26–39.

López-Fernández, D., Madrid-Cuevas, F. J., Carmona-Poyato, A., Muñoz-Salinas, R. & Medina-Carnicer, R. (2016), 'A new approach for multi-view gait recognition on unconstrained paths', *Journal of Visual Communication and Image Representation* **38**, 396–406.

Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Zhao, X. & Kim, T.-K. (2014), 'Multiple object tracking: A literature review', *arXiv preprint arXiv:1409.7618* .

Ma, C., Yang, C., Yang, F., Zhuang, Y., Zhang, Z., Jia, H. & Xie, X. (2018), Trajectory factory: Tracklet cleaving and re-connection by deep siamese bi-gru for multiple object tracking, *in* '2018 IEEE International Conference on Multimedia and Expo (ICME)', IEEE, pp. 1–6.

Milan, A., Rezatofighi, S. H., Dick, A., Reid, I. & Schindler, K. (2017), Online multi-target tracking using recurrent neural networks, *in* 'Thirty-First AAAI Conference on Artificial Intelligence'.

Mu, Y., Yan, S., Liu, Y., Huang, T. & Zhou, B. (2008), Discriminative local binary patterns for human detection in personal album, *in* '2008 IEEE Conference on Computer Vision and Pattern Recognition', IEEE, pp. 1–8.

Mukhtar, A., Cree, M. J., Scott, J. B. & Streeter, L. (2018), Mobility aids detection using convolution neural network (CNN), *in* '2018 International

Conference on Image and Vision Computing New Zealand (IVCNZ)', IEEE, pp. 1–5.

Mukhtar, A., Xia, L. & Tang, T. B. (2015), 'Vehicle detection techniques for collision avoidance systems: A review', *IEEE Transactions on Intelligent Transportation Systems* **16**(5), 2318–2338.

Myles, A., Lobo, N. D. V. & Shah, M. (2002), Wheelchair detection in a calibrated environment, *in* 'Proceedings of the 5th Asian Conference on Computer Vision', pp. 706–712.

Nambiar, A. M. (2017), Towards automatic long term Person Re-identification System in video surveillance, PhD thesis, INSTITUTO SUPERIOR TECNICO.

Nixon, M. S. & Carter, J. N. (2006), 'Automatic recognition by gait', *Proceedings of the IEEE* **94**(11), 2013–2024.

Niyogi, S. A., Adelson, E. H. et al. (1994), Analyzing and recognizing walking figures in xyt, *in* 'CVPR', Vol. 94, pp. 469–474.

Nizami, I. F., Hong, S., Lee, H., Lee, B. & Kim, E. (2010), 'Automatic gait recognition based on probabilistic approach', *International Journal of Imaging Systems and Technology* **20**(4), 400–408.
URL: *http://dx.doi.org/10.1002/ima.20256*

Ojala, T., Pietikäinen, M. & Harwood, D. (1996), 'A comparative study of texture measures with classification based on featured distributions', *Pattern recognition* **29**(1), 51–59.

Opelt, A., Pinz, A. & Zisserman, A. (2008), 'Learning an alphabet of shape and appearance for multi-class object detection', *International Journal of Computer Vision* **80**(1), 16–44.
URL: *http://dx.doi.org/10.1007/s11263-008-0139-3*

Ott, P. & Everingham, M. (2009), Implicit color segmentation features for pedestrian and object detection, *in* '2009 IEEE 12th International Conference on Computer Vision', IEEE, pp. 723–730.

Pala, F., Satta, R., Fumera, G. & Roli, F. (2015), 'Multimodal person reidentification using rgb-d cameras', *IEEE Transactions on Circuits and Systems for Video Technology* **26**(4), 788–799.

Papageorgiou, C. & Poggio, T. (2000), 'A trainable system for object detection', *International journal of computer vision* **38**(1), 15–33.

Park, D., Ramanan, D. & Fowlkes, C. (2010), *Multiresolution models for object detection*, Springer, pp. 241–254.

Paul, M., Haque, S. M. & Chakraborty, S. (2013), 'Human detection in surveillance videos and its applications-a review', *EURASIP Journal on Advances in Signal Processing* **2013**(1), 1–16.

Ran, Y., Zheng, Q., Chellappa, R. & Strat, T. M. (2010), 'Applications of a simple characterization of human gait in surveillance', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **40**(4), 1009–1020.

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016), You only look once: Unified, real-time object detection, *in* 'The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'.

Redmon, J. & Farhadi, A. (2017), Yolo9000: better, faster, stronger, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 7263–7271.

Redmon, J. & Farhadi, A. (2018), 'Yolov3: An incremental improvement', *arXiv preprint arXiv:1804.02767* .

Reid, D. (1979), 'An algorithm for tracking multiple targets', *IEEE transactions on Automatic Control* **24**(6), 843–854.

Ren, S., He, K., Girshick, R. & Sun, J. (2015), Faster r-cnn: Towards real-time object detection with region proposal networks, *in* 'Advances in neural information processing systems', pp. 91–99.

Ristani, E. & Tomasi, C. (2018), Features for multi-target multi-camera tracking and re-identification, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 6036–6046.

Robert, K. (2009), Video-based traffic monitoring at day and night vehicle features detection tracking, *in* '2009 12th International IEEE Conference on Intelligent Transportation Systems', IEEE, pp. 1–6.

Roy, A. & Marcel, S. (2009), Haar local binary pattern feature for fast illumination invariant face detection, *in* 'British Machine Vision Conference 2009', number CONF.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. & Fei-Fei, L. (2015), 'ImageNet Large Scale Visual Recognition Challenge', *International Journal of Computer Vision (IJCV)* **115**(3), 211–252.

Sarkar, S., Phillips, P. J., Liu, Z., Vega, I. R., Grother, P. & Bowyer, K. W. (2005), 'The humanid gait challenge problem: data sets, performance, and analysis', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(2), 162–177.

Satpathy, A., Jiang, X. & Eng, H.-L. (2014), 'Human detection by quadratic classification on subspace of extended histogram of gradients', *Image Processing, IEEE Transactions on* **23**(1), 287–297.

Sharif Razavian, A., Azizpour, H., Sullivan, J. & Carlsson, S. (2014), Cnn features off-the-shelf: an astounding baseline for recognition, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition workshops', pp. 806–813.

Shutler, J. D., Grant, M. G., Nixon, M. S. & Carter, J. N. (2004), On a large sequence-based human gait database, *in* 'Applications and Science in Soft Computing', Springer, pp. 339–346.

Silberstein, S., Levi, D., Kogan, V. & Gazit, R. (2014), Vision-based pedestrian detection for rear-view cameras, *in* '2014 IEEE Intelligent Vehicles Symposium Proceedings', IEEE, pp. 853–860.

Sokolova, M. & Lapalme, G. (2009), 'A systematic analysis of performance measures for classification tasks', *Inf. Process. Manage.* **45**(4), 427–437.

Spampinato, C., Palazzo, S., Giordano, D., Kavasidis, I., Lin, F.-P. & Lin, Y.-T. (2012), Covariance based fish tracking in real-life underwater environment., *in* 'VISAPP (2)', pp. 409–414.

Stauffer, C. & Grimson, W. E. L. (1999), Adaptive background mixture models for real-time tracking, *in* 'IEEE Computer Society Conference on Computer Vision and Pattern Recognition', Vol. 2, IEEE.

Suruliandi, A., Meena, K. & Rose, R. R. (2012), 'Local binary pattern and its derivatives for face recognition', *IET computer vision* **6**(5), 480–488.

Thome, N., Miguet, S. & Ambellouis, S. (2008), 'A real-time, multiview fall detection system: A lhmm-based approach', *IEEE Transactions on Circuits and Systems for Video Technology* **18**(11), 1522–1532.

Toshev, A. & Szegedy, C. (2014), Deeppose: Human pose estimation via deep neural networks, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 1653–1660.

Tuzel, O., Porikli, F. & Meer, P. (2008), 'Pedestrian detection via classification on riemannian manifolds', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **30**(10), 1713–1727.

Vapnik, V. N. (1995), *The nature of statistical learning theory*, Springer-Verlag New York, Inc.

Venkat, I. & De Wilde, P. (2011), 'Robust gait recognition by learning and exploiting sub-gait characteristics', *International Journal of Computer Vision* **91**(1), 7–23.

Viola, P., Jones, M. J. & Snow, D. (2005), 'Detecting pedestrians using patterns of motion and appearance', *International Journal of Computer Vision* **63**(2), 153–161.

Voulodimos, A., Doulamis, N., Doulamis, A. & Protopapadakis, E. (2018), 'Deep learning for computer vision: A brief review', *Computational intelligence and neuroscience* **2018**.

Walk, S., Majer, N., Schindler, K. & Schiele, B. (2010), New features and insights for pedestrian detection, *in* '2010 IEEE Computer society conference on computer vision and pattern recognition', IEEE, pp. 1030–1037.

Wang, C.-C. R. & Lien, J.-J. J. (2007), *AdaBoost learning for human detection based on histograms of oriented gradients*, Springer, pp. 885–895.

Wang, C., Zhang, J., Wang, L., Pu, J. & Yuan, X. (2011), 'Human identification using temporal information preserving gait template', *IEEE transactions on pattern analysis and machine intelligence* **34**(11), 2164–2176.

Wang, L., Hu, W. & Tan, T. (2002), A new attempt to gait-based human identification, *in* 'Object recognition supported by user interaction for service robots', Vol. 1, IEEE, pp. 115–118.

Wang, L., Ning, H., Hu, W. & Tan, T. (2002), Gait recognition based on procrustes shape analysis, *in* 'Proceedings. International Conference on Image Processing', Vol. 3, IEEE, pp. III–III.

Wang, L., Shi, J., Song, G. & Shen, I.-F. (2007), Object detection combining recognition and segmentation, *in* 'Asian conference on computer vision', Springer, pp. 189–199.

Wang, L., Tan, T., Ning, H. & Hu, W. (2003), 'Silhouette analysis-based gait recognition for human identification', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(12), 1505–1518.

Wang, T., Gong, S., Zhu, X. & Wang, S. (2016), 'Person re-identification by discriminative selection in video ranking', *IEEE transactions on pattern analysis and machine intelligence* **38**(12), 2501–2514.

Wang, X., Han, T. X. & Yan, S. (2009), An hog-lbp human detector with partial occlusion handling, *in* '2009 IEEE 12th international conference on computer vision', IEEE, pp. 32–39.

Wojek, C., Walk, S. & Schiele, B. (2009), Multi-cue onboard pedestrian detection, *in* '2009 IEEE Conference on Computer Vision and Pattern Recognition', IEEE, pp. 794–801.

Wojke, N., Bewley, A. & Paulus, D. (2017), Simple online and realtime tracking with a deep association metric, *in* '2017 IEEE international conference on image processing (ICIP)', IEEE, pp. 3645–3649.

Wu, B. & Nevatia, R. (2007), 'Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors', *International Journal of Computer Vision* **75**(2), 247–266.
URL: *http://dx.doi.org/10.1007/s11263-006-0027-7*

Wu, B. & Nevatia, R. (2008), Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection, *in* '2008 IEEE Conference on Computer Vision and Pattern Recognition', IEEE, pp. 1–8.

Wu, Z., Huang, Y., Wang, L., Wang, X. & Tan, T. (2016), 'A comprehensive study on cross-view gait based human identification with deep cnns', *IEEE transactions on pattern analysis and machine intelligence* **39**(2), 209–226.

Yan, J., Zhang, X., Lei, Z., Liao, S. & Li, S. Z. (2013), Robust multi-resolution pedestrian detection in traffic scenes, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 3033–3040.

Yang, F., Choi, W. & Lin, Y. (2016), Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 2129–2137.

Yang, M. & Jia, Y. (2016), 'Temporal dynamic appearance modeling for online multi-person tracking', *Computer Vision and Image Understanding* **153**, 16–28.

Yang, Y.-H. & Levine, M. D. (1992), 'The background primal sketch: An approach for tracking moving objects', *Machine Vision and applications* **5**(1), 17–34.

Yang, Y. & Ramanan, D. (2012), 'Articulated human detection with flexible mixtures of parts', *IEEE transactions on pattern analysis and machine intelligence* **35**(12), 2878–2890.

Ye, Q., Han, Z., Jiao, J. & Liu, J. (2013), 'Human detection in images via piecewise linear support vector machines', *Image Processing, IEEE Transactions on* **22**(2), 778–789.

Zhan, B., Monekosso, D. N., Remagnino, P., Velastin, S. A. & Xu, L.-Q. (2008), 'Crowd analysis: a survey', *Machine Vision and Applications* **19**(5-6), 345–357.

Zhang, E., Zhao, Y. & Xiong, W. (2010), 'Active energy image plus 2dlpp for gait recognition', *Signal Processing* **90**(7), 2295–2302.

Zhang, Y., Huang, Y., Yu, S. & Wang, L. (2019), 'Cross-view gait recognition by discriminative feature learning', *IEEE Transactions on Image Processing* **29**, 1001–1015.

Zhao, T., Nevatia, R. & Wu, B. (2008), 'Segmentation and tracking of multiple humans in crowded environments', *IEEE transactions on pattern analysis and machine intelligence* **30**(7), 1198–1211.

Zhao, Y., Yuan, Z. & Chen, B. (2019), 'Accurate pedestrian detection by human pose regression', *IEEE transactions on image processing* **29**, 1591–1605.

Zhao, Z.-Q., Zheng, P., Xu, S.-t. & Wu, X. (2019), 'Object detection with deep learning: A review', *IEEE transactions on neural networks and learning systems* .

Appendix A

Manual Markings Example Data

Frame #	Leg1_X	Leg1_Y	Leg2_X	Leg2_Y	centre_X	centre_Y	Head_X	Head_Y
20	1879	684	1770	691	1845	525	1875	309
21	1878	676	1762	696	1827	535	1864	309
22	1870	664	1770	697	1822	534	1864	306
23	1861	661	1768	696	1816	523	1848	303
24	1842	663	1761	696	1804	520	1843	312
25	1819	669	1765	693	1807	513	1837	294
26	1804	675	1762	691	1786	516	1828	298
27	1768	691	1768	693	1794	507	1827	298
28	1735	696	1767	696	1795	507	1819	297
29	1704	705	1774	694	1783	510	1809	300
30	1684	699	1770	688	1771	520	1797	297
31	1657	697	1773	697	1768	511	1783	301
32	1645	697	1762	693	1764	516	1782	295
33	1639	700	1765	691	1755	505	1771	304
34	1636	700	1767	685	1747	507	1770	307
35	1633	697	1767	679	1738	507	1770	304
36	1636	699	1765	673	1728	501	1756	303
37	1636	696	1761	661	1702	504	1740	294
38	1632	703	1749	652	1708	493	1732	285
39	1639	703	1738	642	1690	504	1728	292
40	1636	699	1716	645	1684	498	1728	283
41	1633	700	1698	651	1683	495	1707	288
42	1630	705	1674	652	1672	501	1708	285
43	1632	705	1650	669	1666	499	1693	282
44	1630	696	1609	676	1663	501	1684	283
45	1639	700	1585	682	1656	495	1683	282

Figure A.1: Manual Marking Data

Appendix B

Consent Form

RESEARCH ETHICS: CONSENT FORM

Vision Based Disability Detection System

Researcher: Amir Mukhtar
Email: am301@students.waikato.ac.nz

Supervisor: Dr. Michael Cree
Email: cree@waikato.ac.nz

Purpose of Work

This research study is about the development of a computer system to detect and count pedestrians using mobility aids, from traffic surveillance videos. This research is in partnership with Traffic Design Group (TDG) and Callaghan Innovation. It will enable the cost effective and efficient gathering of statistics on pedestrian access by Crippled Children Society (CCS) and TDG to advocate on behalf of the disabled community. The factual lobbying may lead the government to invest and develop more efficiently for those in our community who need to use mobility aids.

Data Collection and its Usage

To develop the computer system, we require image and video data for computer training purposes. We are building a video and photo database of people using mobility aids at different places in Hamilton, New Zealand. The collected data will be analysed and transformed into mathematical form to develop the computer system. All the material will strictly be used for research purposes, and participants will not be identifiable in any visual data published as a part of research papers, presentations and PhD thesis.

	Participant Consent(s)	
1.	I agree to take part in the above study.	☐
2.	I understand that the data of my movements will be collected and used in publications but I will be not identifiable in anyway	☐
3.	I consent to an image of me being published with my face blurred out or Please do not publish any image of me	◯ ◯
4.	I confirm that I have read and understand the information for the above study and had the opportunity to ask questions.	☐
5.	I understand that I can ask to see my data and request to remove them from this study up to 30 days after recording.	☐
6.	I can request for more information at any time.	☐

Figure B.1: page 1

If you have any concern please contact my supervisor, Dr. Michael Cree (m.cree@waikato.ac.nz). If your concern cannot be resolved, please contact the chair of the approving ethics committee, Dr. Karsten Zegwaard via k.zegwaard@waikato.ac.nz.

Name of Participant/Caregiver:

Relation with participant:

Signature:

Phone/Email address:

Date: dd/mm/2016

Researcher Details

Amir Mukhtar, PhD Student (1238202)

am301@students.waikato.ac.nz

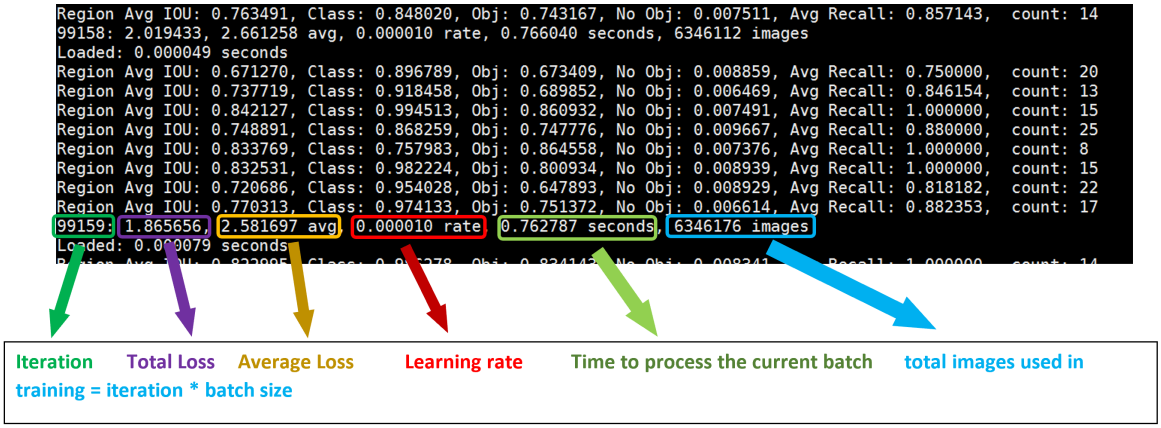
Faculty of Science and Engineering,

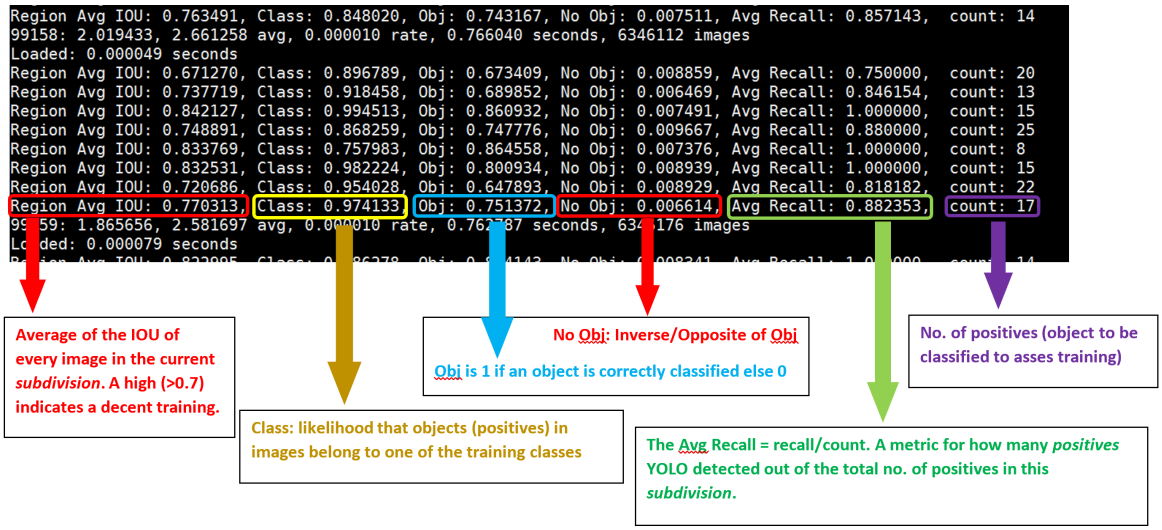
University of Waikato

Figure B.2: page 2

Appendix C

Simulation Screenshots





Appendix D

Access to the image data

We may plan to allow public access to the image dataset for academic purposes, and a web link will be generated after the final thesis submission. Dataset will only be accessible after approval from the school of Engineering, University of Waikato. The availability of a labelled image dataset will allow researchers to perform different experiments related to mobility aids, on a single training and test set.