INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Extracting Relevant Information Regarding Customer Behaviour from Surveillance Videos

Simão de São José Gregório de Oliveira Frazão Correia

Master's in Computer Engineering,

Supervisor:
PhD Luís Miguel Martins Nunes, Associate Professor,
ISCTE - Instituto Universitário de Lisboa

Co-Supervisor:
PhD Tomás Gomes Silva Serpa Brandão, Assistant Professor,
ISCTE - Instituto Universitário de Lisboa

October, 2023

TECNOLOGIAS
E ARQUITETURA

Department of Information Sciences and Technologies

Extracting Relevant Information Regarding Customer Behaviour from Surveillance Videos

Simão de São José Gregório de Oliveira Frazão Correia

Master's in Computer Engineering,

Supervisor:
PhD Luís Miguel Martins Nunes, Associate Professor,
ISCTE - Instituto Universitário de Lisboa

Co-Supervisor:
PhD Tomás Gomes Silva Serpa Brandão, Assistant Professor,
ISCTE - Instituto Universitário de Lisboa

October, 2023

Acknowledgements

I would like to express my gratitude to everyone who has accompanied me throughout this journey. Your support and encouragement have been invaluable in helping me to achieve my goals. I especially wish to acknowledge those who have contributed the most to making this possible.

I would like to give special thanks to my dedicated supervisors, Prof. Luís Nunes and Prof. Tomás Brandão, for their guidance and support during the development and writing of this dissertation, as well as to the rest of the members of the ECI 4.0 research project team: Prof.ª Patrícia Arriaga, Diogo Mendes, and Pedro Jorge.

I also want to express my warmest thanks to my family. In particular, to my parents and siblings, for their unconditional love and encouragement, who have played a crucial role in my achievements, and for that I am very grateful.

A special mention also goes to my close friends and colleagues at ISCTE-IUL, to my fellow scouts, and to the incredible people I have had the privilege of meeting and bonding with along the way. Your comfort and motivation have been a constant source of strength and inspiration.

Simão Correia

Resumo

No ambiente retalhista atual, o uso de tecnologias como videovigilância e inteligência artificial para o estudo do comportamento dos clientes é um fator essencial para melhorar aspectos como marketing, apoio ao cliente, e segurança. Neste contexto, a presente dissertação foca-se no desenvolvimento de um sistema para a extração de informações sobre o comportamento dos clientes a partir de vídeos de vigilância, o que envolve detetar e seguir pessoas, extrair pontos de trajetória, estimar velocidades de caminhada, detetar grupos, e reconhecer ações usando dados de pose (esqueleto).

Em complemento ao sistema, propomos duas contribuições para melhorar o seu desempenho: o mecanismo de compensação de oclusões e um método de suavização de trajetórias. O mecanismo de compensação de oclusões foi criado para mitigar o impacto das oclusões nos dados de localização. Os resultados indicaram melhorias estatisticamente significativas derivadas da sua utilização. Além disso, o método de suavização foi introduzido para atenuar as oscilações nos pontos de trajetória, tendo em conta informação anterior e posterior. Aliado ao mecanismo de compensação de oclusões, provou ser uma ferramenta valiosa para melhorar o mapeamento de trajectórias.

Relativamente ao reconhecimento de acções, comparámos três modelos baseados em esqueleto (ST-GCN, AGCN e PoseC3D) em subconjuntos do conjunto de dados *People in Public* com 12 classes de ação. Os resultados obtidos com o treino e teste dos modelos revelaram a sua eficácia em reconhecer ações típicas de clientes (atingindo taxas de acerto na ordem dos 90%), e permitiram-nos inferir casos de utilização adequados para cada um deles.

Palavras-chave: comportamento do cliente; deteção de objectos; seguimento de objectos; extração de trajectórias; estimativa de pose; reconhecimento de ações.

Abstract

In the modern retail environment, leveraging technologies such as high-resolution video surveillance and artificial intelligence to study in-store customer behaviour is a crucial factor in improving valuable business aspects including marketing, customer service, and security. In this context, this dissertation focuses on the development of a framework for extracting information regarding customer behaviour from high-resolution surveillance videos. This framework incorporates a series of steps, which include detecting and tracking each person, extracting trajectory points, estimating walking speeds, detecting groups, and recognising actions using pose (skeleton) data.

Along with the framework, we propose two contributions designed to enhance its performance: occlusion-aware mechanism and trajectory smoothing method. The occlusion-aware mechanism was created as a means of mitigating the impact of partial occlusions on location data. The experimental results indicated statistically significant improvements resulting from its application. Furthermore, the smoothing method was introduced to attenuate oscillations in the trajectory points, considering both past and future path information. Combined with the occlusion-aware mechanism, it proved to be a valuable tool for improving trajectory mapping.

Regarding action recognition, we compared three skeleton-based models (i.e. ST-GCN, AGCN, and PoseC3D) on subsets of the People in Public dataset featuring 12 shopping-related action classes. The results obtained from training and testing the models demonstrated their effectiveness in recognising customer behaviour (reaching *accuracy* values of around 90%) whilst ensuring privacy, and allowed us to deduce appropriate use cases for each of them.

Keywords: customer behaviour; object detection; multi-object tracking; trajectory extraction; pose estimation; action recognition.

Funding and Publications

This research task was partially supported by P2020 project ‘ECI 4.0 - Espaços Comerciais Inteligentes’ ref. LISBOA01-0247-FEDER-047155, and FCT - Fundação para a Ciência e Tecnologia, I.P. under project, UIDB/04466/2020 and UIDP/04466/2020 (ISTAR).

Part of this dissertation's results were presented at the *IWSSIP 2023* conference and published in the proceedings of the event:

- S. Correia, D. Mendes, P. Jorge, T. Brandão, P. Arriaga and L. Nunes, “Occlusion-Aware Pedestrian Detection and Tracking,” in *2023 30th International Conference on Systems, Signals and Image Processing (IWSSIP)*, Ohrid, North Macedonia, 2023, pp. 1–5, doi: 10.1109/IWSSIP58668.2023.10180296.

Furthermore, within the scope of the ECI 4.0 project, a trajectory-based person re-identification algorithm was developed for multi-camera environments:

- D. Mendes, S. Correia, P. Jorge, T. Brandão, P. Arriaga, and L. Nunes, “Multi-Camera Person Re-Identification Based on Trajectory Data”, *Applied Sciences*, vol. 13, no. 20, 2023, doi: 10.3390/app132011578.

Contents

List of Figures

List of Tables

Glossary

AGCN – Adaptive Graph Convolutional Network

CNN – Convolutional Neural Network

COCO – Common Objects in Context

CRISP-DM – Cross-Industry Standard Process for Data Mining

DBN – Dynamic Bayesian Network

ECI – Espaços Comerciais Inteligentes (Smart Commercial Spaces)

GCN – Graph Convolutional Network

GDPR – General Data Protection Regulation

GNN – Graph Neural Network

GRU – Gated Recurrent Unit

HAR – Human Action Recognition

HOG – Histogram of Oriented Gradients

IoU – Intersection over Union

LSTM – Long Short-Term Memory

mAP – mean Average Precision

MEVA – Multiview Extended Video with Activities

PIP – People In Public

PRISMA – Preferred Reporting Items for Systematic Reviews and Meta-Analyses

R-CNN – Region-based Convolutional Neural Network

RNN – Recurrent Neural Network

SORT – Simple Online and Realtime Tracking

ST-GCN – Spatial Temporal Graph Convolutional Network

SVM – Support Vector Machine

SW – Sliding Window

VIRAT – Video Image Retrieval and Analysis Tool

YOLO – You Only Look Once

CHAPTER 1

Introduction

In this chapter, we begin by delving into the motivation regarding the theme and problem addressed in this dissertation, as a means of introducing the subject. Then, the goals are presented, followed by the research questions that are intended to be answered in this dissertation. Finally, the methodology that was adopted for the development of this work is described.

1.1. Motivation

In recent years, advances in observational technology and retail analytics have facilitated the study of in-store customer behaviour for the purpose of behavioural marketing, in contrast to traditional retail marketing [1]. Using diverse tools such as high-resolution video surveillance cameras, depth sensors, traffic counters, and radio frequency identification (RFID) tags, enables the possibility to continuously evaluate how customers behave while shopping and how they respond to stimuli [1] (e.g. discounts, in-store advertisements and product placement). To this end, rather than solely drawing conclusions based on purchase records, one can analyse the decisions made by customers from the moment they enter the store until they reach the checkout line [2]. By studying these decisions it is possible to infer customer profiles [3], and to adapt the layout of the store based on their preferences, which has been shown to have a positive impact on sales [4].

In parallel with the marketing aspect, there are various practical issues that can benefit from the assessment of in-store customer behaviour, for instance detecting if a customer needs the support of a sales assistant [3]. Furthermore, store security can be improved, for example by recognising violent behaviour [5] and theft [6], both of which would traditionally require a person actively monitoring the public areas.

Several techniques can be applied to evaluate customer behaviour in order to subsequently infer profiles. These techniques include recognising customer actions (e.g. interactions with the products on the shelves [2]), extracting trajectory points relative to the paths taken by customers in the store, and generating heatmaps that illustrate the most consulted areas [4]. This dissertation focuses on the action recognition and trajectory point extraction techniques, based on image data (videos) acquired from high-resolution surveillance cameras.

With the improvements in computing power over the last decades and the availability of public datasets with huge amounts of labelled data, the interdisciplinary field of Computer Vision has seen great progress. Specifically, in the Human Action Recognition (HAR) application domain, research has

evolved from shallow approaches, which require intensive engineering work and specialised domain knowledge to develop effective feature extraction methods, to deep learning approaches, which are able to autonomously learn more generalised and powerful features [7]. Nevertheless, there are still multiple problematic challenges for this domain, namely cluttered background, viewpoint variations, lighting conditions, and occlusions [7].

Some examples of deep learning approaches for HAR include the Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and Graph Neural Networks (GNNs). There are multiple real-world scenarios in which they can be applied such as surveillance, autonomous driving systems, human-robot interaction and entertainment [8]. In addition, different data modalities can be used as input for HAR, namely: RGB, skeleton, depth, infrared, point cloud, event stream, audio, acceleration, radar, and Wi-Fi [8]. Given the scope of this dissertation, both RGB data generated by the surveillance cameras and skeleton data extracted from the generated video frames, can be used.

However, when dealing with customer information, there is an important topic that needs to be addressed first: Privacy. As stated in Recital 26 of the General Data Protection Regulation (GDPR) [9], which applies to all individuals in the European Union (EU) and European Economic Area (EEA): “The principles of data protection should apply to any information concerning an identified or identifiable natural person. (...)”. Therefore, apart from the need to have the customers' faces blurred, in order to maintain total anonymity, the action recognition process must be performed based on customer pose information (skeleton data) [10].

1.2. Goals

The work documented in this dissertation was developed as part of the “ECI 4.0 - Espaços Comerciais Inteligentes” project, which was conducted as a partnership between AXIANSEU Digital Solutions SA (Axians), SONAE MC Serviços Partilhados SA (Sonae) and ISCTE-IUL. The aim of this research project is to develop the prototype of a multimodal platform for the intelligent analysis of customer behaviour in large commercial spaces, using Computer Vision, Sensor Fusion and Machine Learning techniques [11]. This dissertation is primarily focused on the task “development of pose classification models for the detection of client specific behaviours”.

With this in mind, the main objective of this work is to create a framework capable of effectively extracting information regarding customer behaviour, based on videos acquired from high-resolution surveillance cameras. This includes trajectory-related data (location and speed) and pose-related data (actions), both of which require stable and accurate tracking data for the continuous identification of each customer. As for the latter, the framework should be able to recognise shopping-related actions performed by customers in retail stores, using only their pose information (skeleton data).

1.3. Research Questions

The assessment of customer behaviour is a demanding process owing to all the challenges inherent in real-world retail settings. These mainly include occlusion cases (caused by shelves, banners, products on display, other customers, etc.) and the requirement to ensure customer privacy. With this in mind, it is important to answer the following questions:

- How can the accuracy of location data generated by object detection and tracking algorithms be improved in real-world scenarios involving occlusions? What are the practical implications of these improvements on trajectory mapping?
- What methods can be used to effectively recognise customer behaviour while ensuring their privacy? What are the most suitable use cases for each of these methods, taking into account performance and computational efficiency?

1.4. Methodology

Considering the context in which this dissertation is conducted, the Cross-Industry Standard Process for Data Mining (CRISP-DM) [12] model was adopted for the purpose of establishing the development phases for this work. These phases and their interconnections are illustrated in Figure 1.1.

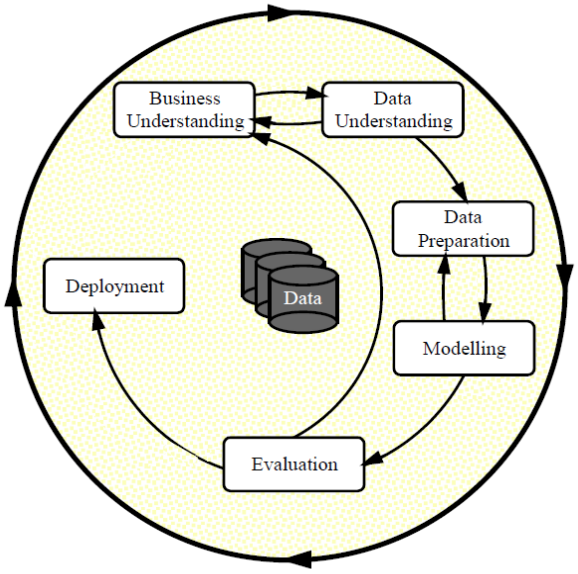


Figure 1.1. CRISP-DM methodology [12].

Initially, in the Business Understanding phase, the problem to be addressed is identified, and the goals to try to solve it are established. These were presented in Sections 1.1 and 1.2.

Subsequently, in the Data Understanding phase, the dataset provided to us as part of the ECI 4.0 project (consisting of surveillance videos recorded inside an electronics store) is analysed in order to identify the characteristics of the data, as well as the challenges that may arise from them. However, given that these may render the dataset unusable, the identification, collection and analysis of other

alternative datasets may be necessary to accomplish the established goals.

Based on what was decided in the previous phase, several techniques can be applied to the data before going to the Modelling phase. As stated in Section 1.2, the main goal of this work is to develop a framework capable of extracting valuable information regarding customer behaviour from images (video frames). This includes trajectory-related data depicting the location of customers over time, as well as pose-related data representing the actions performed by them. Thus, in the Data Preparation phase, several procedures may be required. These encompass pre-processing video frames, including metadata, as well as annotating the dataset with bounding boxes around the individuals of interest in each video frame, numerical identifiers (IDs) for these individuals, and action class labels. From there, it is possible to extract skeleton sequences as a means of building a skeleton-based HAR dataset.

After that, in the Modelling phase, the selected skeleton-based HAR deep learning methods are implemented and trained using the data that was prepared in the previous phase. These models are then tested using metrics that have been chosen for that.

Next, in the Evaluation phase, in addition to the technical results obtained in the previous phase, the models are evaluated in terms of which one best meets the business requirements, based on the identified benefits. Therefore, with this in mind, a decision is made about whether to move on to the Deployment phase or to iterate further in light of important aspects.

Finally, in the Deployment phase, the solution is migrated to the multimodal platform of the ECI 4.0 project, for the purpose of assessing its impact together with all the other components. It should be noted that this assessment is not performed by us.

1.5. Document Structure

The remainder of this dissertation is organised as follows:

- In Chapter 2, a systematic review is provided of existing literature on techniques and methods related to Human Action Recognition (HAR).
- In Chapter 3, the components that make up the proposed framework are described, including implementation decisions and algorithms used. Moreover, the datasets that were considered for use throughout the course of this work are discussed.
- Chapter 4 introduces the occlusion-aware mechanism, a contribution created to detect cases of partial body occlusion and mitigate their effect on location data. Furthermore, a trajectory smoothing method is proposed to attenuate oscillations in the extracted trajectory points.
- In Chapter 5, a rigorous statistical evaluation of the occlusion-aware mechanism is conducted. Additionally, a detailed description of the procedures performed for training and testing the selected skeleton-based HAR models is presented, along with an analysis of the results.
- Finally, Chapter 6 draws the main conclusions and suggestions for future work.

CHAPTER 2

Literature Review

In this chapter, the reader can find a review of the existing literature around techniques and methods related to Human Action Recognition (HAR). First, the methodology chosen to carry out the search for related work is described. Then, important background concepts linked to this subject are introduced. Finally, related work involving different approaches to perform HAR is presented.

2.1. Systematic Review

For the purpose of finding the existing literature related to the topic of this dissertation, a search was conducted in a database and a web search engine, i.e. Scopus and Google Scholar, respectively. The main source of information was Scopus, where a query was created in an attempt to find the most relevant related work, whereas Google Scholar was used to access other papers that were referenced in the related work found through the query. Other documents were provided by the supervisors and members of the "ECI 4.0 - Espaços Comerciais Inteligentes" project.

The query created to conduct the search in Scopus was the following: *('action recognition' OR 'action detection' OR 'pose estimation' OR 'pose orientation' OR 'customer behav*' OR 'shopping behav*') AND ('convolutional neural network' OR 'skeleton based') AND ('surveillance')*. The keywords were chosen with the aim of finding the various deep learning methods that are frequently employed for HAR (whether based on the RGB or the skeleton modality), as well as papers related to the analysis of in-store customer behaviour, both in the surveillance setting. This query was iteratively improved during the research.

In order to filter the query results so that only relevant papers were returned, some constraints were imposed, namely the requisites of being: published (publication stage set as "final"), conference papers and articles, in the field of Computer Science or Engineering, written in english, and published from 2014 to 2022.

After performing the query with all the previously described constraints, the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) [13] workflow was adopted. Starting with 201 papers extracted from Scopus, a screening was done (title and abstract analysis), where only 80 were selected with the criterion of being related to the topic of this dissertation. Next, a skimming of these documents was carried out, in which only the papers that complied with the title and abstract, were accessible under the provided licence, and have been published in recognised conferences and journals, were selected. This last criterion of the skimming phase was applied through the Conference

Ranks and ScimagoJR websites. Therefore, apart from the documents accessed from other sources, 22 papers were selected. In the end, 29 papers were fully analysed and included in Section 2.3. Table 2.1 provides a concise overview of this process, indicating the number of papers at each stage.

Table 2.1. Summary of each phase of the Systematic Review.

Phase	**# Papers**
Query performed on Scopus	201
Screening (title and abstract analysis)	80
Skimming (selective reading of the papers)	22
Selected to be fully analysed (including additional papers not retrieved by the query)	29

2.2. Background Concepts

Most literature on Human Action Recognition (HAR) attempts to classify entire trimmed videos, which means a single person performs a particular action for the whole duration of the video. However, in contrast, when it comes to surveillance footage, each video consists of continuous, untrimmed data streams, where different people perform various actions over time.

To build a classification framework, it is necessary to detect each person, track them throughout the course of the video, and only then classify the multiple actions they are performing. With this in mind, some techniques that enable the transition to the video surveillance scenario are presented in this section, including object detection, multi-object tracking and human pose estimation algorithms.

2.2.1. Object Detection Algorithms

In order to detect each person in the video frames, object detection algorithms can be employed. The concept of these algorithms consists in identifying the objects present in a given image, returning for each object: the bounding boxes with the coordinates of their location, the class to which they belong (e.g. person, dog, bicycle, or car) and the confidence level of the detection (from 0.0 to 1.0).

There are two main categories into which these algorithms can be divided, i.e. single-shot and two-stage detectors [14]. The former refers to algorithms that only analyse each image once to infer the objects and their location, thus making them computationally efficient (an important requirement when it comes to real-time applications such as surveillance systems). The latter refers to algorithms that perform two passes over each image. They first generate a set of propositions on possible object locations and then refine these propositions to provide final predictions.

Examples of single-shot detectors include YOLO (You Only Look Once) [15] and SSD (Single Shot MultiBox Detector) [16], whereas two-stage detectors include R-CNN [17], denoted Regions with CNN features, and its enhancement variations, namely Fast R-CNN [18] and Faster R-CNN [19]. Figure 2.1 shows a high-level illustration of how a single-shot detector (in this case, YOLO) works.

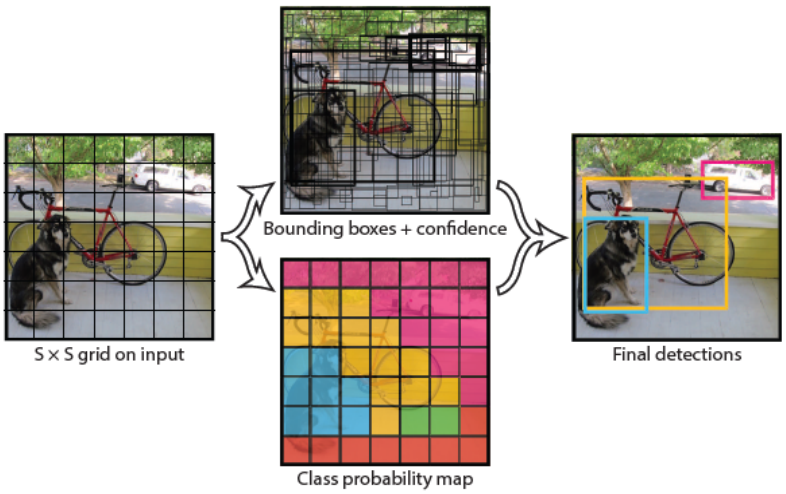


Figure 2.1. Model proposed for YOLO [15].

2.2.2. Multi-Object Tracking Algorithms

Based on the detections generated by the object detection algorithms, it is necessary to perform data association (i.e., to link bounding boxes across video frames), in order to obtain the location of each detected person over time. Hence, assigning a numerical identifier (ID) to each tracked person. This is referred to as the tracking-by-detection paradigm.

To this end, objects are represented as tracks. This representation typically includes information regarding position, velocity, and other relevant attributes, which collectively define the state of each tracked object. With this in mind, motion models (e.g. Kalman or Particle filters) are used to predict where each object should be in the current frame based on its previous states. After that, assignment algorithms (e.g. Hungarian method) are employed to associate the predicted tracks to the detections within the current frame, relying on similarity or cost scores (e.g. Euclidean distance or Intersection over Union (IoU)). Once the assignments are made, the states of the existing tracks are updated using the motion models, based on the assigned detections. Additionally, appearance information is often used to enhance tracking performance, which involves extracting features (e.g. colour histograms or deep learning embeddings) that help distinguish between objects with similar motion patterns.

There are several challenges that render this a demanding task, mainly due to the complexity and variability of real-world scenarios. These challenges include crowded environments (where multiple people can be close together, leading to frequent ID switches), occlusions (when people temporarily hide behind others, partially or completely), complex motion (abrupt stops and accelerations, as well as non-linear trajectories), appearance variations (clothing, illumination, and viewpoint changes), and scale variations (spatial location in the scene, and posture changes).

Examples of multi-object tracking algorithms, which employ the tracking-by-detection approach, include SORT (Simple Online and Realtime Tracking) [20], DeepSORT [21] (extended version of SORT that integrates appearance information based on a deep appearance descriptor), and ByteTrack [22].

A depiction of the output of these algorithms, in particular DeepSORT, is shown in Figure 2.2.

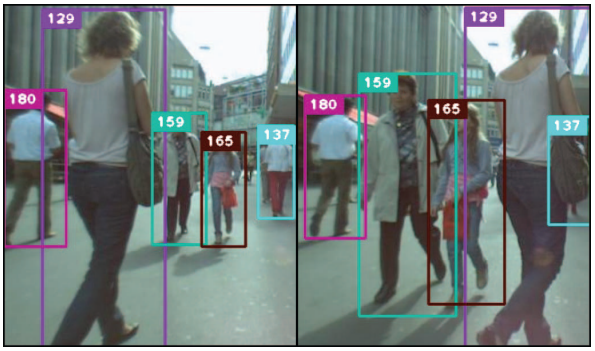


Figure 2.2. DeepSORT output example [21].

2.2.3. Human Pose Estimation Algorithms

In the case of skeleton-based HAR, human pose can either be inferred using RGB-D sensors (3D joint coordinates) or using pose estimation algorithms (2D joint coordinates). Since RGB-D sensors are not available for this work, pose estimation algorithms need to be employed. These algorithms consist of predicting the location of individuals in images by identifying and classifying each of their body joints (keypoints), which have an associated ID and prediction confidence score (ranging from 0.0 to 1.0). This process generates a structure that resembles the human skeleton, by establishing connections (also referred to as edges or limbs) that define the spatial relationships between these keypoints. The specific number of keypoints, and consequently connections, varies depending on the architecture of the model (which can be designed for either a fixed or adaptable number of keypoints), as well as the dataset used for training the model.

There are two main approaches for pose estimation algorithms: top-down and bottom-up. While top-down approaches use information from object detection algorithms about detected individuals in a given image by estimating body joints within the generated bounding boxes, bottom-up approaches detect all the body joints in the whole image and then group the joints that belong to each individual. Compared to its bottom-up counterparts, top-down methods tend to obtain superior performance on standard benchmarks [23]. Examples of bottom-up and top-down approaches include OpenPose [24] (whose model is represented in Figure 2.3) and HRNet [25], respectively.

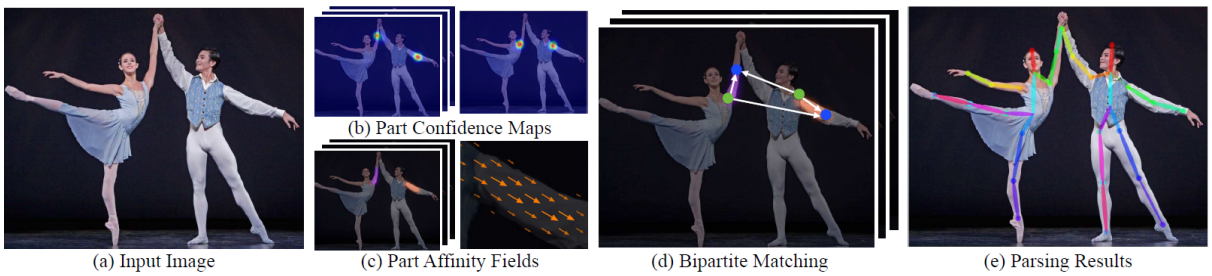


Figure 2.3. Model proposed for OpenPose [24].

2.3. Related Work

Based on the techniques that were presented in Section 2.2, both RGB data sequences and human skeleton sequences can be generated for each tracked person in a video. These two cases correspond to the RGB and skeleton modalities, respectively. There are different methods that can be employed when performing Human Action Recognition (HAR) under each of these modalities, or a combination of both. In this section, we present and discuss some of these methods, which have been employed in other studies related to the theme of this dissertation.

2.3.1. CNN-based methods for HAR

In recent years, Convolutional Neural Networks (CNNs) have garnered attention in several areas in the field of Computer Vision due to their ability to learn spatial features and patterns from still images. These types of CNNs are referred to as 2D CNNs, and can be used in tasks such as image classification, segmentation, and object detection. Plus, it is possible to exploit the availability of large amounts of annotated image data through transfer learning. In [26], a framework for HAR based on the fusion of hand-crafted and deep features was proposed. Specifically, Histogram of Oriented Gradients (HOG) features extracted from processed silhouette images and 2D CNN (pre-trained AlexNet) deep features extracted from the original video frames, were fused into a single feature vector. Then, an entropy-based feature selection technique was used to reduce the dimensionality and a multi-class Support Vector Machine (SVM) was employed to perform action classification.

However, to perform HAR based on RGB data sequences, as opposed to still images, both spatial and temporal information need to be taken into account, as videos contain the temporal dynamics of human body motion. In order to address this issue, various research projects have proposed different solutions, including the gathering of motion-related information in the form of an image, and the use of 3D CNNs, which can learn both spatial and temporal features (see Figure 2.4). Some pioneering works that have explored these approaches include [27] and [28], respectively. In [27], the authors first proposed a two-stream 2D CNN architecture for HAR (illustrated in Figure 2.5). This architecture consisted of two separate 2D CNNs, one for extracting spatial features (appearance) from still frames, and the other for extracting temporal features (motion) from dense multi-frame optical flow. The results from both networks were combined using late fusion, based on softmax scores. In [28], the authors proposed the use of 3D CNNs for HAR and, after that, several other works presented their own architectures, including C3D [29] and I3D (Inflated 3D CNNs) [30], which aimed to maintain the feasibility of using transfer learning, by inflating 2D CNNs into 3D. Also, [31] proposed a method for HAR based on 3D CNNs using 3D motion cuboids as input, which consist of a sequence of absolute temporal difference images, obtained by subtracting each frame with the previous, on a pixel-by-pixel basis, as a way to capture motion information.

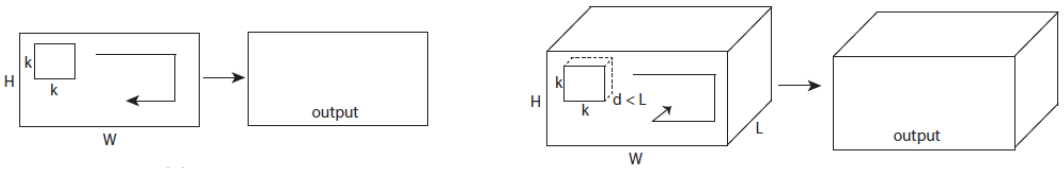


Figure 2.4. Difference between 2D (left) and 3D (right) convolutions [29].

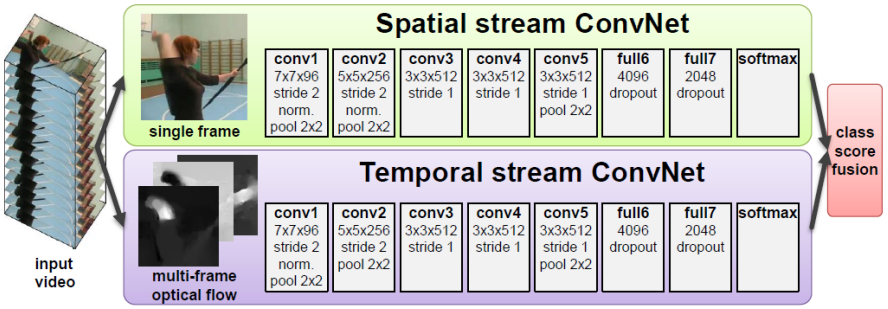


Figure 2.5. Architecture of the two-stream 2D CNNs for HAR in [27].

Given that human actions are normally performed at different temporal scales, the authors of [32] proposed a method for multi-temporal scale HAR. For each temporal scale, a two-stream 3D CNN was employed, with one stream to learn the spatio-temporal information from RGB and another from optical flow. In order to fuse the results, two types of feature-level fusion were required: RO (RGB and optical flow) fusion and MTS (multi-temporal-scale) fusion. For the former, summation was employed and, for the latter, a parallel approach to the Discriminant Correlation Analysis (DCA) method, which reduces dimensionality. Results of the tests performed on the UCF101 and HMDB51 datasets showed that when compared to the single-temporal-scale methods, the proposed method achieved higher recognition accuracy and spent less time on classification due to the smaller representation.

In [33], the authors proposed the pairwise attentive adversarial spatio-temporal network (PASTN) to perform cross-domain few-shot action recognition. PASTN comprised a pairwise TR3D network, an attentive adversarial network and a pairwise discrimination margin loss. The pairwise TR3D network consisted of a source domain TR3D network and a target domain TR3D network, each consisting of a 2D CNN (ResNet-50) to obtain the spatial information with a 3D convolution block on top to convert the spatial features into spatio-temporal features. The attentive adversarial network was employed to align actions with large domain discrepancies, which consisted of a network to generate transferable domain adaptation weights and another network to fill the gap between source and target domains. Finally, a pairwise margin discrimination loss function was used to improve the discrimination of the learned domain-invariant features.

Furthermore, with the goal of recognizing interactions between two people, in addition to single person behaviour, the authors of [34] proposed a two-stream 2D CNN model. In one branch, Motion History Images (MHI) were extracted from sequences of 10 frames and passed to a 2D CNN (VGG-16)

that would output softmax scores. In the other branch, the last frames of the sequences were fed to a Faster R-CNN model that would output both softmax scores and the bounding boxes representing the location where the actions occurred. The softmax scores from both branches were fused in order to obtain the final action classes.

The previously reported studies referred to HAR based on RGB sequences, however, due to some issues that are inherent to the RGB modality, such as illumination changes, background clutter and appearance variations, some works explored the idea of using human skeleton sequences. The main concept of using 2D CNNs to perform skeleton-based HAR is to encode the spatio-temporal dynamics of the skeleton sequence into the form of an image to be processed by 2D CNNs. In [35], an enhanced 3D skeleton visualisation method for view invariant human action recognition was proposed (shown in Figure 2.6). To achieve this, a sequence-based view invariant transform method was applied, which synchronously transformed all skeletons, thus retaining the relative motion between skeletons. Then, the transformed skeletons were depicted as colour images, which encode both spatial and temporal information of the skeleton joints. Furthermore, in order to emphasise the most discriminating cues, visual (morphology) and motion (joint weighting) enhancement methods were employed. Finally, all ten generated image types were passed to a multi-stream 2D CNN, with each stream responsible for processing one type. The probabilities generated by each stream were fused using a weighted fusion method, to generate a final class score.

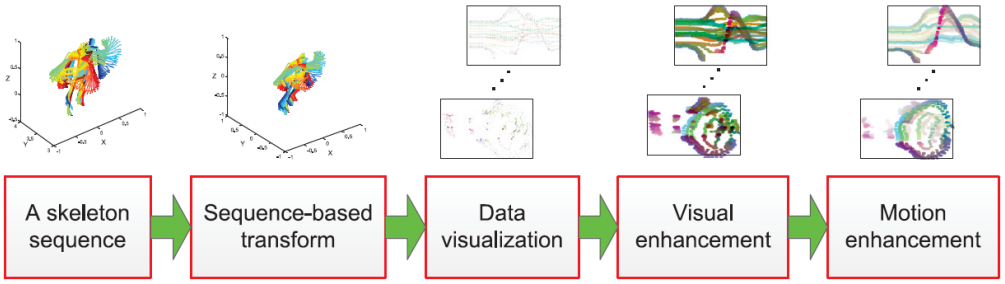


Figure 2.6. Enhanced 3D skeleton visualisation method in [35].

Similarly, the authors of [36] proposed a unified framework for learning comprehensive shape and motion representations from 3D skeleton sequences using Geometric Algebra (GA). To this end, a pipeline was developed, in which the space of a 3D skeleton sequence was first built as a subset of GA. Then, a rotor-based view transformation method was applied to eliminate the effect of viewpoint variation. After that, a space-time view invariant model (STVIM) was built and, therein, the shape and motion representations of the skeleton sequence were mutually learned: Joint Shape Representation, Joint Motion Representation, Bone Shape Representation, and Bone Motion Representation. Finally, a multi-stream 2D CNN was used to extract and fuse deep features from the representation mapping images for action classification.

2.3.2. RNN-based methods for HAR

Some of the downsides of using CNN-based methods for HAR on RGB sequences include the fact that they cannot capture long-range temporal dependencies. To address this issue, some studies combine these methods with RNN variations, i.e. Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU).

In [37], the authors employed multiple hybrid deep neural networks for HAR in videos, including Convolutional Long Short-Term Memory (ConvLSTM) and Long-Term Recurrent Convolutional (LRCN) networks. The ConvLSTM network consisted of four ConvLSTM2D layers and a final dense layer, with a ConvLSTM2D layer being similar to a typical LSTM layer, but having the ability to extract both spatial and temporal features. On the other hand, the LRCN consisted of a 2D CNN to extract spatial features followed by a RNN variant to cope with temporal dynamics and a final dense layer. Multiple 2D CNN encoders including VGG16, ResNet50, DenseNet121, and MobileNet, as well as RNN variant decoders including LSTM, bidirectional LSTM (BiLSTM), and GRU, were tested on the UCF50 dataset. The best accuracy of 87%, using 20-frame sequences, was achieved using LRCN with MobileNet as the encoder and BiLSTM as the decoder, with MobileNet being the lightest and the best choice as a visual feature extractor.

In [38], an evolving ensemble deep learning model consisting of 3 CNN-BiLSTM networks (whose architecture is shown in Figure 2.7) was proposed for HAR. Each of these networks consisted of a 2D CNN encoder (pre-trained GoogLeNet), which would extract spatial features from the video frames, and a BiLSTM decoder, which would infer both their forward and backward temporal dependencies. In addition, a Swarm Intelligence (SI) algorithm was used to optimise the relevant hyper-parameters of each BiLSTM network, such as the learning and dropout rates, and the number of hidden neurons. The results of each network were fused by averaging the softmax scores. The model was evaluated on publicly available datasets (i.e. KTH, UCF50 and UCF101), and demonstrated superior performance compared to those with default and optimal configurations identified by other classical and advanced search methods.

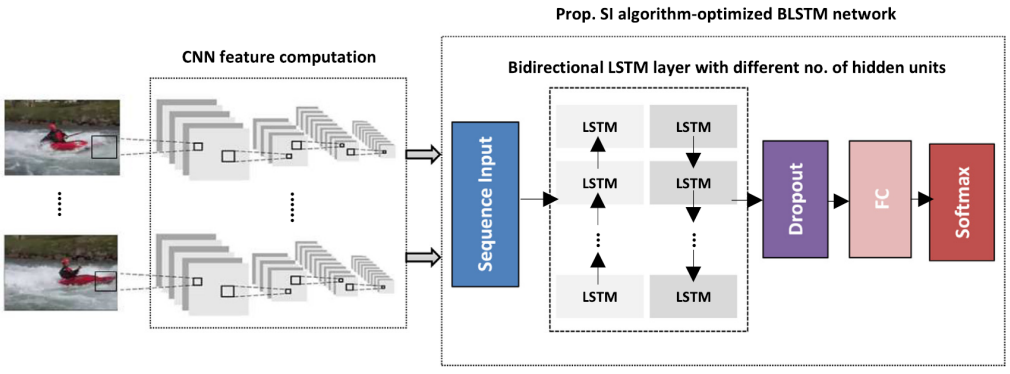


Figure 2.7. Architecture of each CNN-BiLSTM network for HAR in [38].

In addition to the RGB modality, the strengths of the RNN-based methods can also be applied in the skeleton modality. The existing literature exploits two main approaches, consisting of RNN-based methods for extracting the long-range dependencies between consecutive skeletons and, inspired by the two-stream method [27] presented in Section 2.3.1, the combination of such motion information with appearance features extracted from RGB data using CNN-based methods. In [39], the authors proposed a two-stream RNN/3D-CNN method for HAR from 3D skeleton data and RGB video frames, both captured by a Kinect sensor. Specifically, the RNN (BiGRU) stream was fed with 3D coordinates of human skeletons as input, and the 3D CNN (pre-trained C3D) stream with RGB sequences (16 frames each). Furthermore, in order to fuse the results from each stream, features from the fully connected layers were concatenated into a single feature vector, which was normalised using L2 normalisation, and fed to a linear SVM to classify the actions. Results from the experiments performed on the NTU RGB+D dataset showed that using both streams resulted in a 13% accuracy improvement over simply using the RNN stream. This study was extended in [40], in which the main contribution was the replacement of the 3D CNN with a 2D CNN (pre-trained Xception), being that only the middle frame of the RGB frame sequence was used (cropped at the location of the human subject to remove background interference). Moreover, two attention modules (self-attention and skeleton attention) were included in the 2D CNN stream, to extract the most relevant features for action representation. Also, contrary to [39], the concatenated and normalised feature vector was fed to a fully connected layer followed by a softmax layer to predict the actions. These aspects are illustrated in Figure 2.8. Results indicated that the proposed method attained competitive performance close to that of [39], but with much less parameters in the RGB stream (about one-third (76.6 million to 21.9 million parameters)).

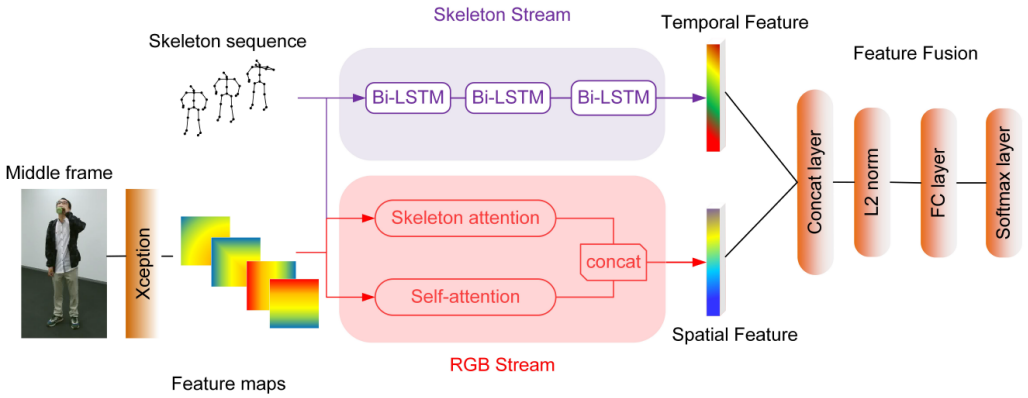


Figure 2.8. Framework of the method for HAR using multimodal data in [40].

In [41], a model for HAR based on 2D skeleton data and a two-branch stacked LSTM network was presented (shown in Figure 2.9). In order to enable structural parallelization and manage occlusions, the structure of the network was designed to process the upper and lower body parts separately. The input for each branch consisted of lightweight features derived from the skeleton joints, comprising

angle and acceleration data. In addition, supplementary information from the scene surrounding the analysed individual was extracted, using a support 3D CNN network (3D-DenseNet). The OpenPose algorithm was employed to estimate the 2D poses of all the individuals in each video frame, and the extension of an existing multi-object tracking algorithm was also included to track the detected 2D poses in the entire frame sequence. Results from experiments carried out on different HAR datasets (i.e. KTH, Weizmann, UCF Sports, IXMAS, HMDB51, UCF101, Kinetics400, UT-Kinect and NTU-RGB+D), showed the ability of the model to manage missing skeleton information and partial body occlusions, as well as its comparable performance to that of 3D skeleton-based works.

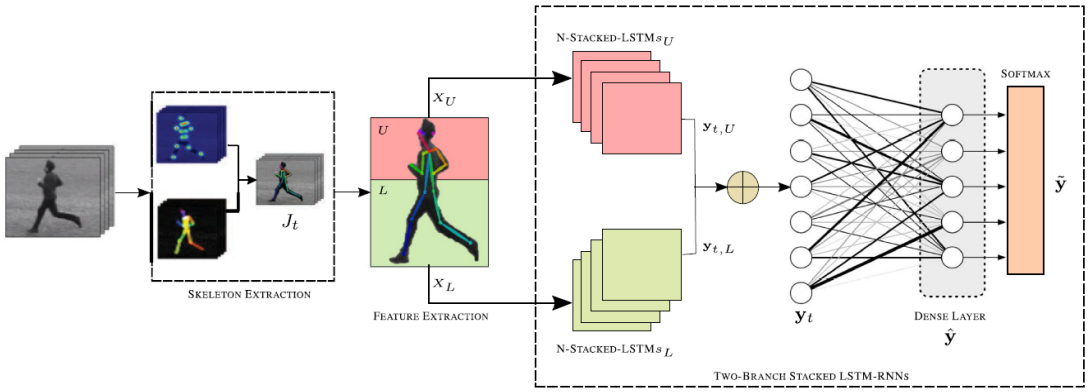


Figure 2.9. Architecture of the two-branched stacked LSTM network for HAR in [41].

2.3.3. GCN-based methods for HAR

Graph Convolutional Networks (GCNs) are a generalisation of CNNs for graphs of arbitrary structures. Due to the graph nature of skeleton data, in which joints can be represented as nodes and bones can be represented as edges, many works adopted GCN-based methods for HAR. Specifically, this subject began to receive significant attention after the Spatial Temporal GCN (ST-GCN) was proposed in [42], as the first adaptation of GCNs for HAR. To this end, based on sequences of skeleton data, the authors introduced two types of edges for generating the input graph (noting that each body joint is a node): the spatial edges (connections between different joints of the same skeleton) and the temporal edges (connections between the same joint across consecutive time steps) – these are shown in Figure 2.10 as the blue and green connections, respectively. With this representation, both spatial patterns and temporal dynamics of the skeleton sequences can be learned by the ST-GCN, where the convolution filters are applied directly on the graph nodes and their neighbours. Furthermore, both 3D skeletons generated using RGB-D sensors and 2D skeletons generated using pose estimation algorithms can be employed in this architecture. Both approaches were explored on the NTU-RGB+D (3D coordinates) and Kinetics (2D coordinates) datasets, using OpenPose as the pose estimation algorithm.

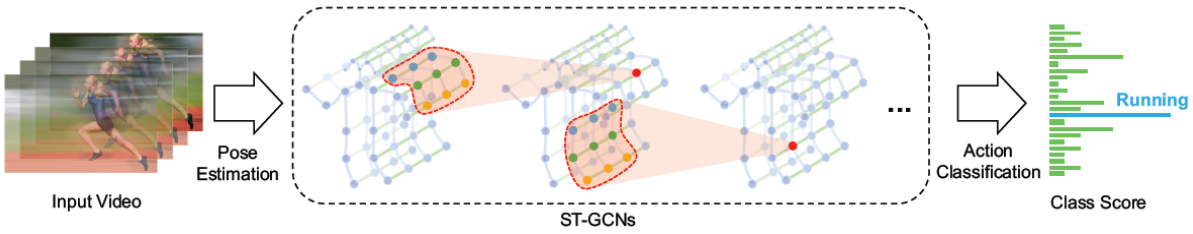


Figure 2.10. Framework of the ST-GCN for HAR using skeleton data in [42].

The ST-GCN was extended in [43], in which the authors proposed the Two-Stream Adaptive GCN (2s-AGCN). This extension consisted of explicitly using first-order (joint coordinates) and second-order (bone lengths and directions) information, as opposed to the ST-GCN, which only considers first-order information. To this end, a two-stream architecture was designed, in which softmax scores from both streams were merged through addition to classify the actions. Moreover, instead of being manually set, the topology of the graphs was parameterized and incorporated into the network to be jointly learned and updated with the model, therefore being better suited for the HAR task.

In another approach, the authors of [44] proposed a multi-task framework based on ST-GCNs to improve classification accuracy. For that, they included an attentional branch to give greater attention to more discriminative features and a co-occurrence feature learning branch to aggregate all joints globally. In the attentional branch, attention weights were learned from fully attentional blocks (FAB) in a data-driven manner. In the co-occurrence feature learning branch, after the fourth convolutional layer, the feature map tensors were transposed to arrange joints to the channel dimension, the result was input to several convolutional operations in order to aggregate the global features from all joints effectively.

2.3.4. Online Multi-Person Action Recognition

In the previous sections (2.3.1, 2.3.2 and 2.3.3), different deep learning methods that can be used for HAR, either based on the RGB or skeleton modality, were presented. Most of these studies performed HAR on trimmed videos, in which one or two people would perform a certain action for the entire duration of the video. However, as mentioned in Section 2.2, in the case of surveillance footage, each video consists of untrimmed data streams, where different people perform various actions over time. Thus, to perform HAR in these cases, it is required that we detect all individuals, track them over time (multi-person), and only then recognise the actions they are performing, while being able to detect the beginning and end of each action (online). Thereby, in this section, we present some studies that aimed to recognise human actions based on information captured by video surveillance systems.

In [45], the authors proposed a framework for HAR based on online surveillance data streams of non-stationary environments. Specifically, a 2D CNN model (pre-trained VGG-16) was used to extract spatial features relative to individual frames. These features were then passed to an optimised Deep

AutoEncoder (DAE), which would convert the high-dimensional features into low-dimensional, while learning the temporal changes between adjacent frames. Finally, a Quadratic Support Vector Machine (QSVM) classifier was employed to process the output of the DAE as a means to classify the actions being performed. Furthermore, predictions with high confidence scores were stored for iterative fine-tuning with new data, allowing the model to adapt to variations in non-stationary environments. Results from experiments performed on the UCF50, UCF101, HMDB51 and YouTube Action datasets verified that the framework was capable of performing classification at 25 frames per second, while maintaining efficient and effective performance.

In [46], the authors proposed a modular visual pipeline for surveillance systems with the goal of early detection and prevention of suicide. For that purpose, they implemented a top-down approach consisting of pedestrian detection, pedestrian tracking, pose estimation and action recognition (both normal actions and high risk behavioural cues), where each step uses the output from the previous module. In the first module, the authors used pre-trained YOLOv5 weights (YOLOv5x) and fine-tuned them according to their own private dataset. Then, in order to track the detected pedestrians over time, they applied the DeepSORT algorithm, which was also used to generate trajectory information. After that, since the top-down pose estimation approach was chosen for the pipeline, HRNet was the selected algorithm, rather than other bottom-up algorithms such as OpenPose. Finally, based on the pedestrian pose information, various features of non-geometric and geometric types were extracted from a sequence of 30 consecutive frames, and fed into a two-branch network with a stacked LSTM structure (one branch for each feature type). The resulting softmax scores from both branches were fused using an element-wise multiplication, and the fused value was then used to compute the final action prediction for the given sequence. Additionally, a rule-based logical layer was created to assess the occurrence of actions over a pre-defined duration, to infer risk behaviour. Figure 2.11 shows some examples of recognition results obtained from running the pipeline on test samples.

Figure 2.11. Examples of recognition results on surveillance testing clips in [46].

Similarly, but using a bottom-up pose estimation approach, a HAR framework was proposed in [10] for real-time on-the-edge surveillance systems. First, a multi-person human pose estimator was used to extract the pose information of each person in the video frames and generate the respective

bounding boxes. Then, each frame was cropped in the bounding boxes areas, and these crops were used to compute feature representations and temporal locations, to assign numerical ID labels to each person. Finally, an extension to the ST-GCNs named Real-World Graph Convolutional Networks (RW-GCNs) was used to classify the actions from skeleton sequences. This extension was motivated by three constraints that the authors identified in real-world scenarios, namely: flawed body joint information provided by pose estimation algorithms, the need to have a short response time, and the limited computational power of the low-power edge devices that were used in their work in order to ensure privacy. To solve these issues, they used a simple static graph with a predefined structure, adopted a Sliding Window (SW) method with no overlapping frames, and used attentive feedback augmentation that consisted in propagating information from previous SWs into the network, while processing the current SW. The results demonstrated the ability of the model to address real-world constraints, while achieving an accuracy of 94.1% on the NTU-RGB+D-120 dataset, with 32 times less latency than the baseline ST-GCN, and an accuracy of 90.4% on the Northwestern UCLA dataset in the presence of noisy input.

Besides [10], other studies have explored the idea of using the SW approach for online HAR. In [47], a framework for processing continuous, untrimmed data streams was proposed. To this end, the authors used ST-GCNs to recognise human actions from skeletal sequences that were extracted based on a SW method (depicted in Figure 2.12). This method consisted of dividing the signal into a series of fixed-size windows, with the interval between each window (SW step size) set to 1 frame, resulting in the overlapping of SWs. Due to this overlap, multiple predictions are made for each frame, so in order to generate a final classification, the Majority Voting (MV) technique was used, in which the action that received the most votes is chosen. This approach smoothes the final output labels, but implies a latency corresponding to the length of the SWs.

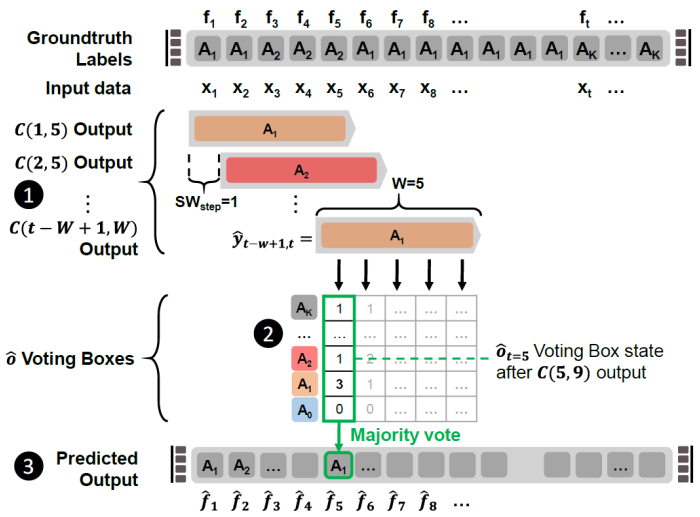


Figure 2.12. Illustration of the SW approach with the MV principle [47].

Likewise, a real-time multi-person HAR framework for surveillance applications was proposed in [14]. First, throughout the video stream, the people in the scene were detected and tracked using the YOLOv3 and DeepSORT algorithms, respectively. Also, if subjects were too far away from the camera, a zoom-in function would be applied for better recognition performance. Then, for each person, the area around the bounding box was blurred and, within an interval of 5 frames, a 16-frame SW would be fed to an Inflated 3D CNN (I3D) to perform the classifications. Finally, Non-Maximum Suppression (NMS) was applied in order to obtain a more robust decision for every group of five consecutive SWs, which resulted in a delay of about 2.5 seconds between the recognition results and the occurrence of the observed action.

Most skeleton-based approaches focus only on human skeletons, ignoring contextual information such as objects and scenes. To address this issue, the authors of [48] proposed a framework for the recognition of Object-related Human Actions using GCNs (OHA-GCN), which is shown in Figure 2.13. First, the OpenPose algorithm was applied to estimate the pose of each person in a video frame. If a person was holding an object, that object would be located by subtracting the human area (estimated joint heatmap) from the moving area (background subtraction). Subsequently, a tracking-by-detection algorithm was employed to extract the regions where each person was located over time, using the pose information. In order to only select the poses with the most complete structure, an informative frame selection strategy was used, that consisted of dividing the video in segments of equal length and choosing the frame with the highest confidence score for each segment. Finally, a human pose graph and an object-related human pose graph (object node connected to the joints of both hands) were generated in the spatial and temporal domains from the sequence of informative frames, and fed to a two-stream ST-GCN. The outputs of both streams were fused to obtain a final prediction. This method was tested in the proposed IRD dataset and in the ICVL dataset (introduced in [49]), attaining an accuracy of 80.1% and 91.9%, respectively, while still achieving real-time performance.

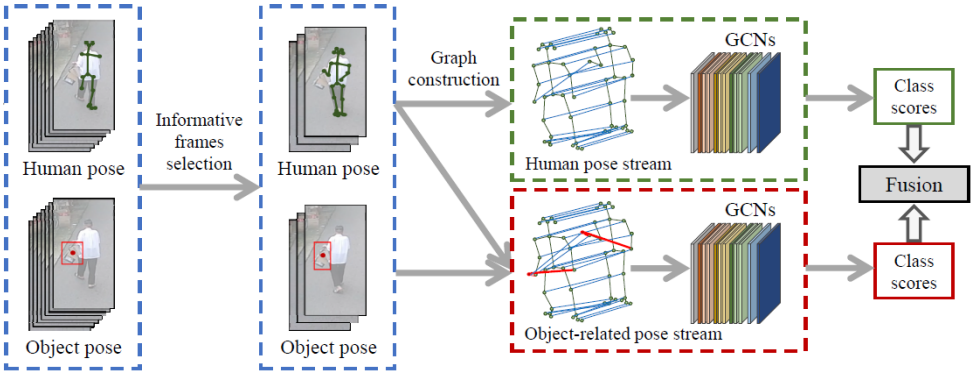


Figure 2.13. Framework of the OHA-GCN for HAR using skeleton data in [48].

2.3.5. Customer Behaviour Recognition in Commercial Spaces

There is not much literature specifically about recognising customer actions from video surveillance systems, and most of it relies on shallow, rule-based approaches. In [3], a system for the multimodal assessment of customers' appreciation for products was proposed. First, frame sequences and audio files were extracted from the respective surveillance videos. Based on the frame sequences, several modules were created to extract different features, including motion energy classification (direction and amount of movement), human detection and tracking (trajectories), face detection and tracking (facial expression analysis), and hand detection and tracking (gesture recognition). As for the audio files, a module was created to extract voice-related features, such as pitch, energy, MFCC and jitter, to detect interest and classify it as positive or negative. Finally, to take into account the relationships and the importance of each module, a Dynamic Bayesian Network (DBN) was employed. This system was tested on recordings made in a store laboratory. However, given that the nature of the data involved is quite sensitive, the ethical issues regarding privacy make it impractical in real-world scenarios.

In [2], the authors developed a system capable of recognising different types of customer actions performed in front of the shelves, such as no interest, viewing, turning to shelf, touching, picking and returning to shelf and picking and putting into basket. For that, both head and body orientation cues, as well as arm gesture information were integrated into a framework. The head and body orientation cues were discretized into 8 directions, and were inferred using a multi-class Support Vector Machine (SVM) based Semi-Supervised Learning method in order to estimate whether the customer is looking or turning to a shelf. The arm gesture information, defined as Combined Hand Feature (CHF), included features such as hand trajectory, tracking status and relative position between hand and basket, and were used by a DBN in order to classify different arm actions. Despite correctly recognizing 89.5% of the images in the private dataset of the authors, since shallow and rule-based approaches were used, there are some weaknesses that can be identified in this work, such as not being able to generalise well and not being scalable to other more demanding datasets.

In the future work section of [2], the authors discussed that they wanted to try other postures such as bending over or squatting down, rather than just standing. That aspect was addressed in [49], in which each action was deconstructed into three abstraction levels, that is: posture, locomotion and gesture. These abstraction levels provided three sub-action descriptors for each action, which were classified using three 2D CNNs that aimed to extract different appearance-based temporal features. The goal consisted of developing a real-time action recognition model for surveillance systems, where each person in a video frame would be identified through motion detection, human detection (HOG with SVM) and tracking (Kalman Filter) algorithms. Then, the respective regions of interest (bounding boxes) would be fed to the 2D CNNs, in the forms of binary difference images (shape), motion history images, and their combined cues (weighted average images), for each abstraction level respectively.

Their framework achieved a mean Average Precision (mAP) of 76.6% and 83.5% in frame-based and video-based experiments, respectively, while running at 25 frames per second on the proposed ICVL video surveillance dataset, showing its ability to perform in real time. Figure 2.14 illustrates examples of recognition results obtained from running the framework on test videos from the ICVL dataset.

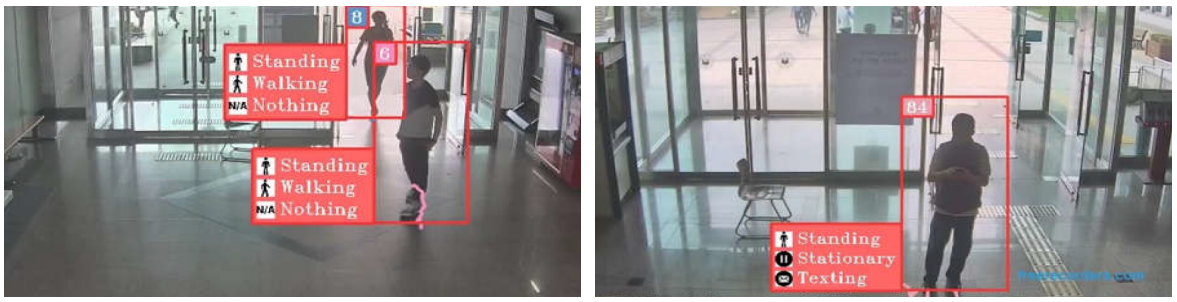


Figure 2.14. Examples of recognition results on surveillance testing clips in [49].

2.3.6. Summary and Contributions

Considering the related work that was just described, it can be concluded that there are various types of deep learning methods that can be used to perform Human Action Recognition (HAR), either using RGB data, human skeleton data, or a combination of both.

When using RGB data (images or videos), CNN-based methods have the strength of being able to learn appearance information about the scene and the objects (including people) that are involved in it. However, because 2D CNNs can only be employed to extract features from static images, and the temporal dynamics of the human body is, in most cases, an essential aspect to describe actions, many solutions have been proposed. These include the gathering of motion-related information in the form of an image (e.g. optical flow) to be learned by 2D CNNs, and the use of 3D CNNs, which can extract both spatial and temporal features from video frame sequences. Additionally, many works have used RNN-based methods as a complement to 2D CNNs, where 2D CNNs extract features from static video frames and RNN-based methods operate on them to learn temporal dependencies. Nevertheless, due to some challenges that are inherent to the RGB modality, such as illumination changes, background clutter and appearance variations, various works have explored the use of skeleton data.

There are two main forms of acquiring human skeleton data, either using RGB-D sensors, which extract the 3D coordinates of body joints from the depth information of the scene, or pose estimation algorithms, which directly extract the 2D coordinates of body joints from RGB data. When using the skeleton modality, either CNN, RNN, or GCN-based methods can be used for HAR. In the case of the CNN-based methods, most works explore different ways of encoding the spatio-temporal dynamics of skeleton sequences as pseudo-images to be processed using 2D CNNs. On the other hand, regarding the RNN-based methods, some works pass the body joint coordinates directly to the networks, while others pass lightweight features derived from the skeleton data, such as angles and acceleration data.

Finally, studies that use GCN-based methods, leverage the natural graph structure that is inherent in skeleton data, where body joints are represented as nodes and bones as edges.

Furthermore, taking into account the advantages of these two modalities, some works aimed at developing multimodal methods, in which two or more networks are used to process different types of data. With regard to this topic, most of the studies that were covered used CNN-based methods to process RGB data and RNN-based methods to process skeleton data, although GCN-based methods have also been employed for the latter case.

Regarding the video surveillance scenarios, where each subject needs to be detected and tracked throughout the data stream (multi-person), both RGB and skeleton modalities have been used. In the case of the former, the region covered by the bounding box of each tracked person is extracted in a series of consecutive frames, and the resulting RGB sequences are passed to a CNN-based method for recognizing the actions. However, since skeleton-based methods do not require much information to perform HAR, most works have chosen to use them. In such cases, RNN and GCN-based methods are employed to perform HAR on 2D skeleton sequences extracted through pose estimation algorithms, using either bottom-up or top-down approaches. However, the use of supplementary cues to further enrich the pose information was not explored. In addition, to segment the recognised actions in real time, many works applied the Sliding Window (SW) technique, each with a different approach.

In the specific case of customer action recognition in commercial spaces (i.e. stores), there are only a few studies, and most are based on shallow, rule-based approaches, which is not ideal with all the recent advances in deep learning methods. Moreover, none of them covered the implications of real-world cases, where customer faces need to be blurred due to ethical and privacy issues.

Thus, this dissertation seeks to explore this gap in the literature through leveraging existing deep learning methods to the case of customer action recognition in large commercial spaces, noting that when compared to other surveillance scenarios, these tend to be more challenging due to occlusions by shelves and banners. With this in mind, the focus of this dissertation is to employ skeleton-based methods that have been previously used for HAR in video surveillance systems (specifically ST-GCNs), while addressing the practical issues of occlusions and trajectory mapping. Table 2.2 shows the most relevant studies for the scope of this dissertation.

Table 2.2. Overview of the most relevant studies for the scope of this dissertation.

Reference	Description	Highlights
[40]	A two-stream RNN/2D-CNN method was proposed for HAR based on the multimodal fusion of 3D skeleton data and RGB data.	- Supplemented the pose information with RGB data to recognise human-object interaction actions. - Similar to [39] but with one-third of the parameters in the RGB stream due to the use of 2D CNNs instead of 3D CNNs.
[41]	A two-branch stacked LSTM network was proposed for HAR based on 2D skeleton data, in which the upper and lower body parts were processed separately.	- Robust approach to deal with partial occlusions.
[42]	The authors introduced the first adaptation of the GCNs for skeleton-based HAR (ST-GCN), which is able to automatically learn spatial and temporal patterns from skeleton sequences.	- Supports 2D skeleton data. - Provides complementary information to the RGB modality.
[46]	A modular visual pipeline was proposed for surveillance systems with the intent of early detection and prevention of suicide. The pipeline consisted of modules for pedestrian detection, pedestrian tracking, pose estimation, action recognition (RNN-based method), and logical layer to infer risk behaviour.	- Used the YOLOv5 object detector, the DeepSORT tracker, and the HRNet pose estimator (top-down approach).
[10]	A framework was proposed for HAR in real-time on-the-edge surveillance systems. The authors extended the ST-GCNs (RW-GCNs), to cope with flawed body joint information, limited computational power and short response time.	- Robust approach to deal with noisy input from pose estimators. - More efficient than the base ST-GCN. - Used a Sliding Window (SW) technique with no overlapping frames.
[47]	A framework was proposed for processing continuous, untrimmed data streams. The authors used ST-GCNs to recognise human actions from skeleton sequences that were extracted based on a SW technique.	- Used a SW technique with overlapping frames (step size of 1 frame) and Major Voting (MV) to infer the final action for each frame.
[48]	A framework was proposed for the recognition of object-related human actions using GCNs (OHA-GCN). For that, the authors introduced an object-related human pose graph, in which the object node is connected to the joints of both hands.	- Viable approach to recognise human-object interaction actions without using appearance information.
[2]	A system was proposed for recognizing different types of customer actions performed in front of the shelves.	- Similar goal to that of this dissertation.

CHAPTER 3

Design and Development

In this chapter, details regarding the design and development of the proposed solution are presented. The aim is to create a framework capable of effectively extracting information on customer behaviour from high-resolution surveillance footage. With this in mind, an overview of the various components that constitute the framework is first provided. Then, different datasets that were or could have been used to train and test the selected deep learning algorithms, as well as to simply perform inference, are described.

3.1. Overview

The process of extracting information from video regarding the behaviour of a person is composed of a series of steps, illustrated in Figure 3.1. Initially, a video is loaded and decoded into a sequence of frames, which are stored in a list. Then, for each video frame, an object detection algorithm is used to identify and locate people. Next, a multi-object tracking algorithm is employed to assign a numerical identifier (ID) to each detected person over time. Based on this result, trajectory points are extracted, the respective projections on the 2D floor plan (top view) are obtained, and an estimate of the speed at which each person moves is calculated. Besides the trajectory data, skeleton sequences regarding the pose of each tracked customer are also extracted. These sequences are then utilised to infer the actions being performed. Furthermore, groups are identified by analysing factors such as the distance between people and the scale of their bounding boxes.

The implementation of the framework is publicly available and can be accessed via the following GitHub repository: https://github.com/simaoc00/eci4.0-customer-behaviour.

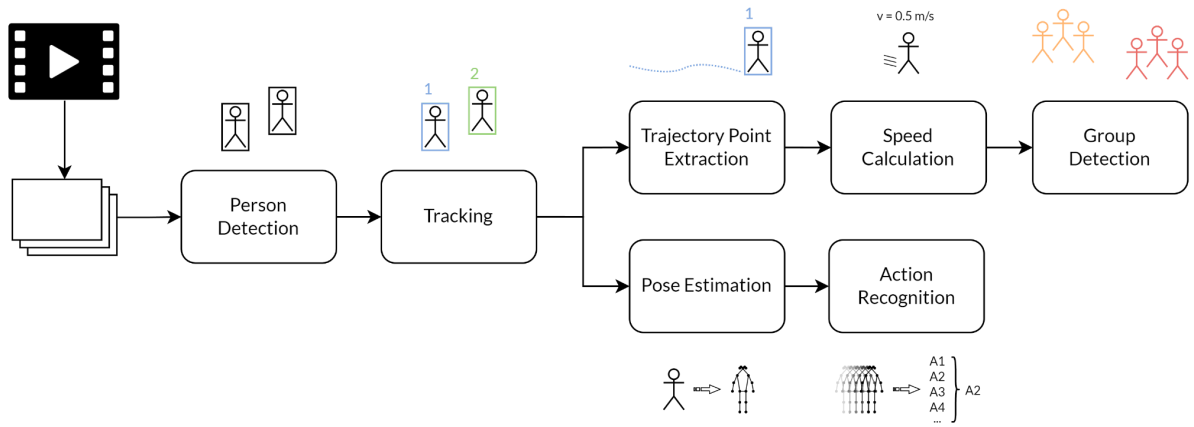


Figure 3.1. Diagram of the proposed framework.

3.1.1. Person Detection

As mentioned in Section 2.2.1, object detection algorithms can be applied to detect people in video frames, their purpose being to locate the objects present in a given image and identify them based on their class (e.g. person, bicycle, car).

You Only Look Once (YOLO), whose first version was proposed in [15], is one of the most widely used methods for this task. Its architecture consists of a single Convolutional Neural Network (CNN) capable of simultaneously generating multiple bounding boxes, as well as class probabilities for each of them. The main benefits over other object detectors, such as R-CNN [17] (including Fast [18] and Faster [19] R-CNN), are that it is extremely fast, can take into account the context in which the objects are in the image, and is very generalizable in how it learns the representations of each object.

Throughout the years, several versions of the YOLO algorithm have been proposed, with version 8 currently available. This work uses version 5 (YOLOv5 [50]), given that it is currently in a stable state, performs well, and is easily accessible through the PyTorch Hub [51] repository of pre-trained models. In particular, we chose to employ the YOLOv5x6 model because of its ability to detect distant people, as it operates on high-resolution images (1280x1280 pixels).

The weights that are provided in the Pytorch Hub repository, and were consequently used for this dissertation, result from a training process executed by the authors of YOLOv5 on the Microsoft COCO [52] dataset. Given that this dataset includes a large number of labelled images (over 200,000), many of which feature people, the pre-trained YOLOv5x6 model delivered reliable results, without the need to fine-tune it to our data.

A comparison between the various pre-trained YOLOv5 models can be seen in Table 3.1. The first column contains the names of the models, each denoted by a letter (n, s, m, l, x) representing its size, along with an additional '6' to indicate the variants designed for higher-resolution images. The second column displays the input sizes for each model in pixels. The third and fourth columns show the mean Average Precision (mAP) scores, indicating performance at Intersection over Union (IoU) thresholds of 0.5 to 0.95 and 0.5, respectively. The fifth and sixth columns display the inference speeds for CPU and GPU (NVIDIA Tesla V100) for a batch size of 1, in milliseconds. The last two columns show information on the number of model parameters (in millions), as well as the number of floating-point operations (FLOPs) required to process an image with a resolution of 640x640 pixels (in billions).

Upon analysing Table 3.1, it is clear that, beyond operating on high-resolution images, YOLOv5x6 is the best performing model, as it achieved the highest mAP scores. This was the main consideration in our choice over the other '6' variants, as our focus was on obtaining the most reliable location data possible. However, it should be noted that this superior performance comes with considerably higher computational complexity. Specifically, the YOLOv5x6 model has more parameters and requires more time to perform inference, as well as a higher number of floating-point operations, rendering it more

demanding on computational resources. Nevertheless, the decision to use it was made based on the specific requirements of being able to detect distant people and providing accurate results.

Table 3.1. Properties and validation results (on the Microsoft COCO [52] val2017 split) of the available pre-trained YOLOv5 models [50].

Model	**Size (pixels)**	**mAP*val* @0.5:0.95**	**mAP*val* @0.5**	**Speed CPU b1 (ms)**	**Speed V100 b1 (ms)**	**Parameters (M)**	**FLOPs @640 (B)**
YOLOv5n	640	28.0	45.7	45	6.3	1.9	4.5
YOLOv5s	640	37.4	56.8	98	6.4	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	86.7	205.7
YOLOv5n6	1280	36.0	54.4	153	8.1	3.2	4.6
YOLOv5s6	1280	44.8	63.7	385	8.2	12.6	16.8
YOLOv5m6	1280	51.3	69.3	887	11.1	35.7	50.0
YOLOv5l6	1280	53.7	71.3	1784	15.8	76.8	111.4
YOLOv5x6	1280	55.0	72.7	3136	26.2	140.7	209.8

3.1.2. Tracking

To obtain the location of each detected individual over time, it is necessary to associate the bounding boxes that correspond to the same person, throughout the video frames. A numerical identifier (ID) is thus assigned to each tracked individual. This is referred to as the tracking-by-detection paradigm. For this dissertation, we decided to use the ByteTrack [22] algorithm, which is the current state of the art in multi-object tracking.

In order to associate the bounding boxes, ByteTrack leverages every detection, contrary to most methods, which discard those with low confidence scores. The reason for this method to include low confidence detections is that many result from occlusions, which does not invalidate their usefulness. Hence, detections are selected based on the confidence score, using a threshold set to 0.6. With this in mind, for each video frame, the detected bounding boxes whose confidence score is greater than the threshold, are associated with the bounding boxes predicted using a Kalman filter [53] (based on information obtained from previous frames). This association consists of computing either motion or appearance similarity, i.e. Intersection over Union (IoU) or Re-Identification (Re-ID), respectively, and applying the Hungarian method [54] to assign the IDs based on the similarity. If the associations are unsuccessful, then the process is repeated for the detections whose confidence scores are lower than the threshold, using IoU only, in order to resolve occlusion and background detection cases.

The components used in the Kalman filter to predict the tracks, based on measured locations and prior knowledge, are $(x,\ y,\ a,\ h,\ vx,\ vy,\ va,\ vh)$, where $(x,\ y)$ is the centre point of the bounding box, a is its aspect ratio, h is its height, and $(vx,\ vy,\ va,\ vh)$ are the corresponding velocities.

Regarding the parameters that were chosen, relative to the ByteTrack implementation available on the GitHub repository [55], qualitative experiments were conducted with different configurations, and the values that yielded the best results are presented in Table 3.2. It should also be noted that we tried adjusting the position and velocity weights of the Kalman Filter, but since these modifications did not consistently improve the results, we decided to maintain the original values.

Table 3.2. ByteTrack parameters.

Parameter	**Value**
track_buffer	frames per second (fps)
match_thresh	0.8
track_thresh	0.5
aspect_ratio_thresh	1.6
min_box_area	10

3.1.3. Trajectory Point Extraction

Having the IDs assigned to each tracked person, we were able to extract additional information about the paths travelled by them over time. Specifically, we store every point corresponding to the centre of the bottom side of their bounding boxes (feet position). That is, given that bounding boxes, in this case of rectangular shape, are defined by the coordinates $(x\_min,\ y\_min,\ x\_max,\ y\_max)$, which refer to both the top-left $(x\_min,\ y\_min)$ and bottom-right $(x\_max,\ y\_max)$ corner points, the corresponding trajectory points $(x,\ y)$ are extracted as follows: $((x\_min\ +\ x\_max)/2\ ,\ y\_max)$.

However, due to oscillations in the dimensions of the bounding boxes that occur because of the presence of certain obstacles (occlusions) or the way each person moves, the extracted points end up showing some irregularities. To solve this issue, a smoothing method is applied, which is explained in detail in Section 4.2.

In addition, assuming that the dataset to be used includes projection matrices relative to the homography [56] of the scenes, for each extracted trajectory point, its equivalent projection in the 2D floor plan (top view) is also stored. This process consists of multiplying the projection matrix of the associated scene (with a 3x3 dimension) by the extracted trajectory point $(x,\ y,\ 1)$ (with an additional z coordinate of value 1). The result of this operation is a 3-tuple $(u,\ v,\ w)$, and the subsequent floor plan projection consists of dividing each of its components by w: $(x',\ y',\ 1)\ =\ (u/w,\ v/w,\ w/w)$.

3.1.4. Speed Calculation

In order to estimate the walking speed of each person, in a first approach, we tried to use the velocity information of the central position of the bounding box $(vx,\ vy)$ provided by the Kalman filter, one of the components used in the ByteTrack algorithm, as reported in Section 3.1.2. However, since these values are inferred from the pixel coordinates of the video frames, the walking speed of individuals

far from the camera turned out to be much lower than those close to it. To solve this issue, we chose to use the coordinates projected into the 2D floor plan, as a way to obtain a better approximation of the actual speed. It was calculated based on the distance travelled by each person in one second.

More specifically, assuming there is a list with the floor plan trajectory points of a certain person, corresponding to the locations travelled by that person so far, we fetch the last points in the list based on a sliding window whose size is equal to the number of frames per second (fps). If the total number of points available is less than the fps, the size is set to that value instead. Next, we calculate the total distance travelled in that range, by summing the Euclidean distances between every two consecutive points. The Euclidean distance between two points, e.g. $(x1, y1)$ and $(x2, y2)$, is calculated using the formula: $\sqrt{(x2 - x1)^2 + (y2 - y1)^2}$. Then, we divide the total distance calculated in the previous step by the size of the sliding window, in order to obtain the distance travelled in each fraction of a second (one frame). Finally, we multiply that value by the fps.

3.1.5. Group Detection

Another functionality that we found useful was to check if a detected person is alone or belongs to a group of people. For that purpose, based on the two bounding boxes of a pair of people, we analyse two criteria that need to be mutually confirmed. The first criterion is whether the distance between the centres of the two bounding boxes is less than 1.35 times the width of the wider bounding box. The second criterion is to measure the difference between the scales of the two bounding boxes, that is, to confirm that the result of dividing the area of the larger bounding box by the area of the smaller bounding box is less than 1.5. After all pairs are associated, a check is performed in order to join all pairs with common elements into a single group. For example, taking into account that each person is identified by its ID, the pairs $(1, 2)$, $(2, 6)$, and $(6, 5)$, would be merged into the group $(1, 2, 6, 5)$. Finally, considering the ends of the bounding boxes of all members, we mark the group with a general bounding box.

3.1.6. Pose Estimation

Considering the sensitive nature of surveillance footage, where the privacy of each individual must be ensured, the process of recognizing the actions being performed by them has to be performed under the skeleton modality. Hence, apart from the trajectory data, skeleton sequences (time sequences of keypoints) regarding the pose of each tracked customer are also extracted.

With this in mind, we decided to use the MMPose toolkit [57] (version 0.29.0), considering the variety of human pose estimation algorithms it provides, both bottom-up and top-down approaches, as well as its flexibility and ease of use. This pose estimation toolkit is part of the OpenMMLab Project [58], which is an open-source Pytorch-based algorithm system that covers a wide range of research

topics of computer vision such as image classification, object detection, semantic segmentation, and video understanding.

Amongst the available human pose estimation algorithms, we chose HRNet [25], specifically the HRNet-W32 model, which operates with an input size of 384x288 pixels. This algorithm adopts the top-down approach, which means that it uses information regarding the location of individuals in a given image (derived from object detection algorithms) to generate the 2D poses. In order to perform inference, the MMPose toolkit provides a configuration file (.py extension) that defines the model, as well as a checkpoint file with pre-existing parameters. The latter consists of pre-trained weights that resulted from a training process previously performed on the Microsoft COCO [52] dataset, which includes keypoint annotations for over 250,000 people, comprising (x, y) coordinates of 17 keypoints. Similar to the case of YOLOv5 (object detection algorithm), the pre-trained HRNet-W32 model also presented good results without the need to fine-tune it according to our data, having achieved a mAP score of 76.1 on the Microsoft COCO val2017 split [57].

3.1.7. Action Recognition

With the skeleton sequences extracted for each person over time, we can then perform human action recognition using skeleton-based action recognition models. For that, we used the MMAction2 toolkit [59], which supports many video understanding models, including action recognition, skeleton-based action recognition, spatio-temporal action detection and temporal action localization. Along with the MMPose toolkit, MMAction2 is also part of the OpenMMLab project, which means that, due to the unified interfaces among the various projects of the platform, it enables a simple integration between the pose estimation and action recognition processes.

MMAction2 version 0.24.1 provides three skeleton-based action recognition models, i.e. ST-GCN [42], AGCN [43], and PoseC3D [23]. There are numerous configuration and checkpoint files available, relative to training processes that were performed on the NTU RGB+D (60 and 120 classes), HMDB51, UCF101, and FineGYM datasets. However, since the majority of the classes included in these datasets are not of great interest for the purpose of this dissertation, we need to perform the training process either from scratch or by fine-tuning the model according to the available pre-trained weights, using another more suitable dataset.

To train any of the skeleton-based action recognition models with a custom skeleton dataset (i.e. beyond those already supported), it is necessary to store each skeleton sequence, which represents a person performing an action, into a pickle file (a binary file that comprises Python object structures as byte streams). These pose sequences are usually related to a RGB video snippet with the action being performed, and can either be extracted from it or provided by the authors of the dataset. With this in mind, each skeleton sequence is stored in the pickle file, as a dictionary with the following fields:

- frame_dir (str): The identifier of the corresponding video.
- label (int): The action label.
- img_shape (tuple[int]): The shape of a video frame, a tuple with two elements, in the format of (height, width).
- original_shape (tuple[int]): Same as img_shape.
- total_frames (int): The temporal length of the skeleton sequence.
- keypoint (np.ndarray, with shape [M x T x V x C]): The keypoint annotation.
    - M: number of persons;
    - T: number of frames (same as total_frames);
    - V: number of keypoints (17 for COCO, 18 for OpenPose, etc.);
    - C: number of dimensions for keypoint coordinates (C=2 for 2D keypoints, C=3 for 3D keypoints).
- keypoint_score (np.ndarray, with shape [M x T x V]): The confidence score of keypoints.

Then, we gather all the pickle files into separate lists for training and validation purposes. After that, we save each of these lists as individual pickle files, such as “custom_dataset_train.pkl” for the training set and “custom_dataset_val.pkl” for the validation set. Once this is done, we simply need to edit some fields in the configuration file of the model, regarding the number of classes, paths to the training and validation pickle files, and the path to the pre-trained weights (if fine-tuning is intended). More information on these procedures can be found in [59].

Having the checkpoint file resulting from the training process, in order to perform inference, we simply provide the action recognition model with the 2D skeleton sequences extracted using the pose estimation algorithm (HRNet), and it returns the resulting action predictions.

3.2. Datasets

Considering what was explained in Section 3.1, regarding the overview of the proposed framework, it is necessary to explore the multiple options of datasets that can be used to support it. In this section, we discuss some datasets that were considered during this work, taking into account that the ideal dataset should include footage obtained from video surveillance cameras in a retail setting, showing customers engaging in shopping-related activities.

3.2.1. Worten Surveillance dataset

To achieve the established objectives, we were provided with a dataset that was prepared for the "ECI 4.0 - Espaços Comerciais Inteligentes" project, with the purpose of understanding the decisions made by customers during their visit to a Worten store located at Mar Shopping (in the city of Matosinhos, Portugal).

The dataset comprises videos acquired from high-resolution surveillance cameras over a period of one week (from Monday to Sunday). It includes 91 hours of footage that was filmed at a resolution of 1080p using 5 different cameras (each covering a different area) with a frame rate of 20 frames per second (fps). The videos are accelerated, with each second corresponding to approximately 3 seconds in real-time (3.3 seconds). Furthermore, a human face-blurring algorithm was applied beforehand to anonymise the data relating to each customer. Figure 3.2 provides examples of footage.

Each video was annotated with tracking data (i.e. bounding boxes and customer IDs), along with the respective trajectory points projected into the 2D floor plan. In addition, homography information [56] was provided for each camera, in the form of projection matrices, which allow the data acquired from the videos to be projected into the 2D floor plan of the store (as if it was seen from above).

Due to restrictions imposed by the project agreement, we are not permitted to show examples of footage that features people. It should also be noted that unexpected problems (described in Section 3.2.8) related to the accessibility of the dataset, the provided tracking data and the application of the human face-blurring algorithm, made it necessary to find alternative datasets.

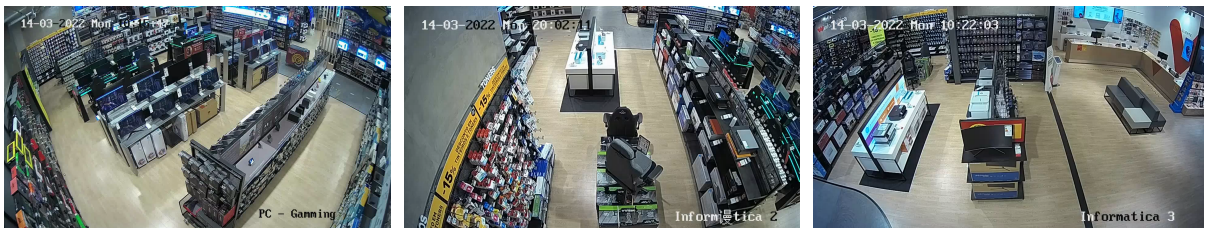


Figure 3.2. Examples of footage from the Worten Surveillance dataset.

3.2.2. Video Image Retrieval and Analysis Tool (VIRAT) dataset

The VIRAT [60] dataset consists of a large-scale surveillance video dataset designed to evaluate the performance of event recognition algorithms. The dataset is partitioned into two parts: VIRAT Ground Video Dataset and VIRAT Aerial Video Dataset. The former consists of stationary footage from ground based cameras, and contains approximately 25 hours of video distributed across 16 different outdoor scenes, which include parking lots, construction sites, open outdoor spaces, and streets. The videos were recorded in 720p and 1080p, with a frame rate of 25 to 30 fps, varying according to the camera used. The latter, however, consists of videos collected from aerial vehicles, which, by their nature, are not relevant for this dissertation.

Every video frame is annotated with the bounding boxes and numerical identifiers (IDs) of each moving object (e.g. people and vehicles), along with the events that occur over time. These events include Single Person Events (e.g. walking, running, standing, gesturing, loitering), Person and Vehicle Events (e.g. getting into or out of vehicle, opening or closing trunk, loading, unloading), and Person and Facility Events (e.g. entering or exiting facility). In addition, the homography information of each

scene is provided, in the form of projection matrices, which allow the tracking data acquired from the videos to be projected into the 2D floor plan (top view). Figure 3.3 shows examples of 3 scenes from the VIRAT dataset, along with their corresponding homography projections.

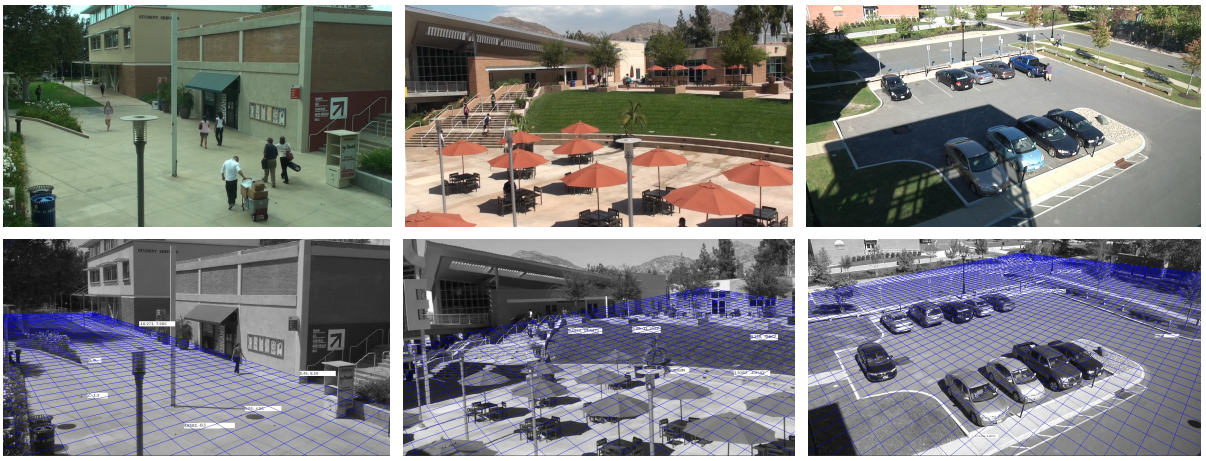


Figure 3.3. Examples of scenes from the VIRAT [60] dataset and their homography projections.

3.2.3. Multiview Extended Video with Activities (MEVA) dataset

The MEVA [61] dataset consists of a very-large-scale dataset for human action recognition (HAR). The complete dataset contains about 9300 hours of footage from ground-based cameras, collected over the course of 3 weeks, of which 144 hours are annotated for 37 action categories, including bounding boxes of actors and props. Regarding the annotations, 122 hours of the 144 were collected in support of the NIST Activities in Extended Video (ActEV) challenge [62], whereas the remaining 22 hours were released through the MEVA website, along with 328 hours of video (recorded in 1080p with 25 fps).

The recordings consist of continuous, untrimmed surveillance videos featuring around 100 actors performing scripted scenarios, as well as spontaneous background activity, with both overlapping and non-overlapping viewpoints across indoor and outdoor spaces. The action categories included in the MEVA dataset are listed below (some visual examples are provided in Figure 3.4):

- person sits down
- person stands up
- person picks up object
- person puts down object
- person carries heavy object
- person transfers object
- person abandons package
- person steals object
- person reads document
- person interacts with laptop
- person texts on phone
- person talks on phone
- person talks to person
- person embraces person
- hand interacts with person
- person purchases
- person opens facility door
- person closes facility door
- person enters scene through structure
- person exits scene through structure
- person rides bicycle
- person enters vehicle
- person exits vehicle
- person opens trunk
- person closes trunk
- person opens vehicle door
- person closes vehicle door
- person loads vehicle
- person unloads vehicle
- vehicle picks up person
- vehicle drops off person
- vehicle starts
- vehicle stops
- vehicle reverses
- vehicle turns left
- vehicle turns right
- vehicle makes U-turn

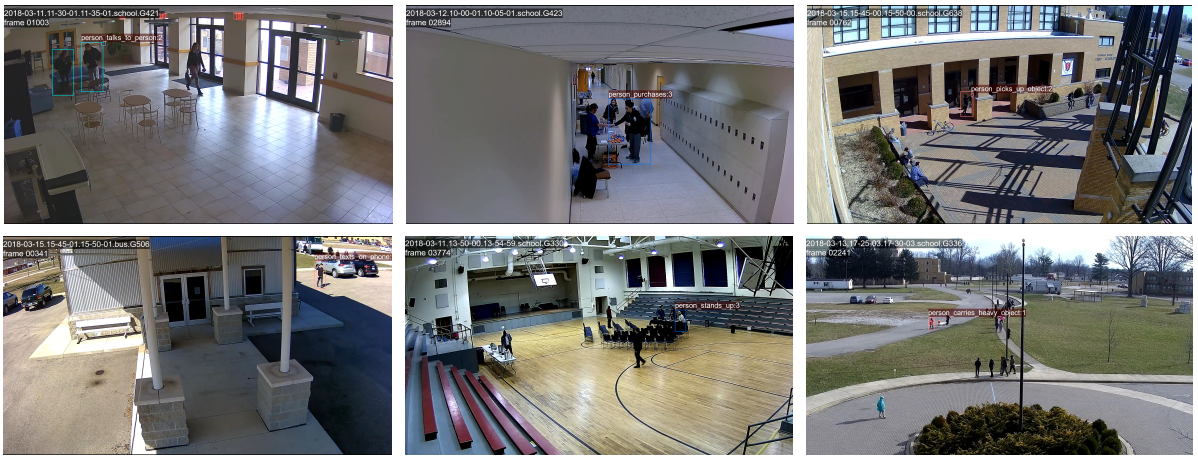


Figure 3.4. Examples of action categories from the MEVA [61] dataset.

3.2.4. People in Public (PIP) dataset

The PIP [63] dataset is a consented large-scale video dataset of people performing actions in different locations. It contains a total of 405,781 background stabilised videos (1518x1262 pixels with 30 fps) of 67 action types (examples are shown in Figure 3.5). These were collected by more than 150 subjects across 44 countries using the Visym Collector [64] mobile application. The action types in this dataset are subsets of the 37 categories in the MEVA dataset and are consistent with the ActEV challenge.

The Visym Collector mobile app was developed by the authors of the dataset, with the objective of turning the collection of annotated video datasets into an easier process, since it integrates video collection, activity labelling, and bounding box annotation into a single step. Thus, the annotations for each video, which can be intuitively accessed, modified, and visualised using the VIPY python library [65], include the bounding boxes surrounding the subject performing the action over time, as well as the action label, and the frame interval at which it occurs.

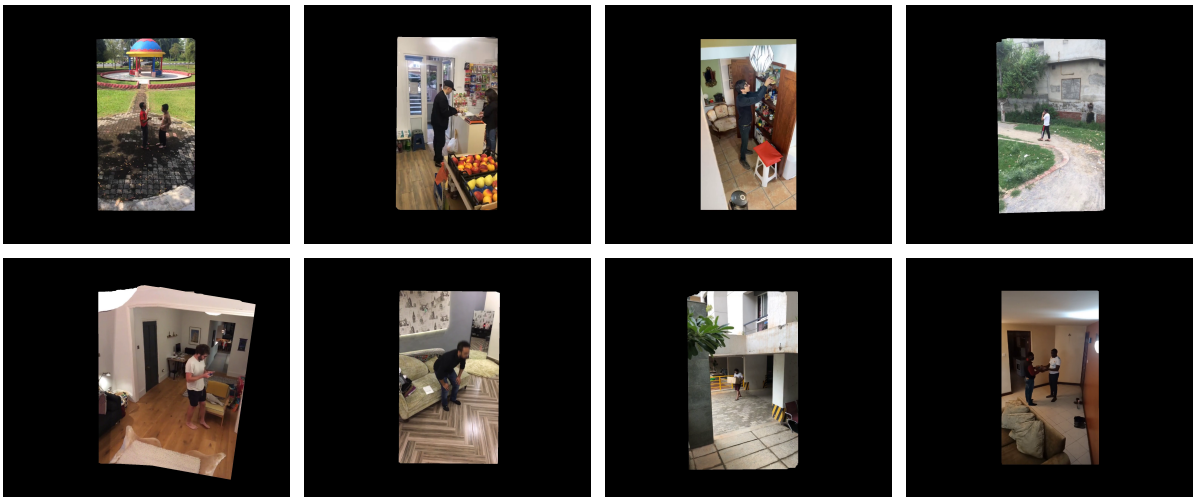


Figure 3.5. Examples of videos from the PIP [63] dataset, each representing an activity type.

3.2.5. Multi-camera Multiple People Tracking (MMPTRACK) dataset

The MMPTRACK [66] dataset is a large-scale dataset with about 9.6 hours of footage, recorded inside Microsoft indoor laboratories where 5 simulated settings were constructed, i.e. retail, lobby, industry, cafe and office. The videos were acquired from surveillance cameras positioned at different locations around each scene, with all fields of view connected (one camera has overlapped field of view with at least one of the other cameras). The filming was done at a resolution of 640x320 with a frame rate of 15 fps. Figure 3.6 shows examples of footage from three of the cameras in the retail setting.

The MMPTRACK dataset, as the name suggests, was primarily designed for training multi-camera multi-object tracking systems, so the annotated data only comprises bounding boxes and person IDs (across all camera views), as well as the respective footpoint projections on the 2D floor plan (world coordinates system). However, given that the retail setting of this particular dataset is quite similar to that of the Worten store, it provides a good alternative for training skeleton-based action recognition models in retail-related activities. In that case, it would be necessary to further annotate the dataset with the labels of the actions performed by each individual over time.

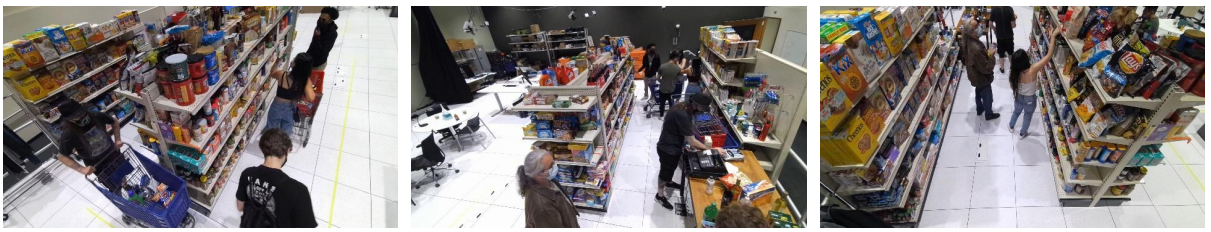


Figure 3.6. Examples of footage from the retail setting of the MMPTRACK [66] dataset.

3.2.6. MERL Shopping dataset

The MERL Shopping [67] dataset consists of 106 two-minute videos, which were captured by a static overhead HD camera (920x680 pixels with 30 fps) gazing down at people shopping from grocery-store shelves, arranged in a laboratory setting. The videos are annotated with the start and end timestamps of actions from the following 5 different classes: reach to shelf (stretch arm towards the shelf), retract from shelf (retract arm from the shelf), hand in shelf (extended period with arm on the shelf), inspect product (inspect a product while holding it in the hand), and inspect shelf (look at the shelf without touching or reaching for the shelf). It is important to note that each video contains multiple instances of these actions. Figure 3.7 presents some sample frames from the videos.

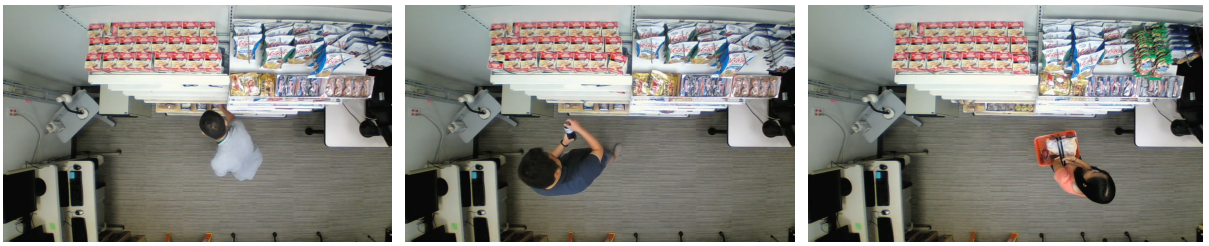


Figure 3.7. Examples of footage from the MERL Shopping [67] dataset.

3.2.7. UCF-Crime dataset

The UCF-Crime [68] dataset is a large-scale dataset with 128 hours of footage. It consists of 1900 long and untrimmed real-world surveillance videos (recorded at a resolution of 320x240 pixels), containing 13 realistic anomalous activities, i.e. shoplifting, vandalism, robbery, burglary, etc., as well as normal activities. Figure 3.8 displays visual examples for 3 of these activity types. The dataset can be used for general anomaly detection, considering all anomalies in one group and all normal activities in another group, as well as for recognizing the individual anomalous activities. Regarding the annotations, these consist of video-level labels (meaning that each video is tagged with the activity it depicts), with the exception of the testing videos, which include the start and ending frames of the anomalous event.

Figure 3.8. Examples of activity types (shoplifting, vandalism, and normal event, respectively) from the UCF-Crime [68] dataset.

3.2.8. Discussion

During the first three months of the development process of this dissertation, we did not have access to the Worten Surveillance dataset, so most of the components that the methodology includes were tested on the VIRAT dataset, except for the action recognition process (given that the events depicted on the VIRAT dataset were of no interest to the purpose of this dissertation).

When we obtained the Worten Surveillance dataset, we noticed that the provided tracking data was imprecise. Furthermore, the area that resulted from the application of the human face-blurring algorithm was rectangular and often too large, so it would frequently cover more body space beyond the face area (sometimes reaching as far as the waist). These issues made the process of extracting quality skeleton sequences from customers a challenging task. Making this dataset inadequate for the

goals intended for this dissertation. Therefore, it was necessary to find an alternative dataset for the process of action recognition.

Among the multiple datasets we found, the ones that interested us the most (due to the nature of the actions involved) were the MEVA, PIP, MMPTRACK, MERL Shopping, and UCF-Crime datasets, with the MMPTRACK dataset being the closest to what we previously described as the ideal dataset. However, of all the options of interest, both MMPTRACK and UCF-Crime did not include information regarding the frame interval at which each action is performed, and so they would need to be further annotated. Furthermore, besides not including tracking data, the viewpoints of the cameras used in the MERL Shopping dataset (gazing down at people) are substantially different from the ones used in the Worten Surveillance dataset, so we decided to discard it as well.

Regarding the PIP and MEVA datasets, the main advantage of PIP over MEVA, for the purposes of this dissertation, is that it provides straightforward, feature-rich video data for the process of training human action recognition models. This is because the videos are trimmed, meaning that each video (in its entirety) contains a person performing a single action, contrary to the continuous, untrimmed videos of the MEVA dataset, where each video contains multiple people performing different actions over time, with challenges such as camera distance. Therefore, we decided to use the PIP dataset for action recognition.

Taking into account the 67 types of activities included in the dataset, we selected the classes that could be useful for the task at hand. We were left with a total of 139,236 video files corresponding to the following classes: person walks, person sits down, person stands up, person picks up object (from floor, shelf or table), person puts down object (from floor, shelf or table), person carries heavy object, person transfers object to person, person interacts with laptop, person texts on phone, person talks on phone, person talks to person, and person purchases from cashier.

In summary, for experiments involving the people detection, tracking, trajectory point extraction, speed calculation, group detection and pose estimation processes, we utilised the VIRAT [60] dataset (see Sections 4.1, 4.2, 4.3 and 5.1). Alternatively, for the action recognition process, we decided to use the PIP [63] dataset (see Section 5.2).

CHAPTER 4

Contributions

In this chapter, some issues that were discovered throughout the design and development stages of the framework are explained, as well as the proposed solutions to mitigate them. First, the impacts that occlusions have on the trajectory point extraction and pose estimation processes are presented, and a mechanism for detecting and rectifying occlusion cases is proposed. Then, some of the factors that may cause irregularities in the extracted trajectory points are explained, along with a smoothing method that was introduced to correct them.

4.1. Occlusions

When certain obstacles are present in the scene, for instance cars, lampposts, trees, and bushes (in a street scene) or items, shelves, and banners (in a retail store scene), they are likely to occlude people standing close to them. This causes the detections made by the YOLOv5 object detection algorithm to simply surround the visible area of the occluded individuals. To address this problem, we developed a mechanism to detect whether or not an individual is occluded and, if so, to automatically adjust the dimensions of its bounding box to include the occluded area. Figure 4.1 depicts how this mechanism (referred to as the occlusion-aware mechanism) interacts with the other components of the proposed framework.

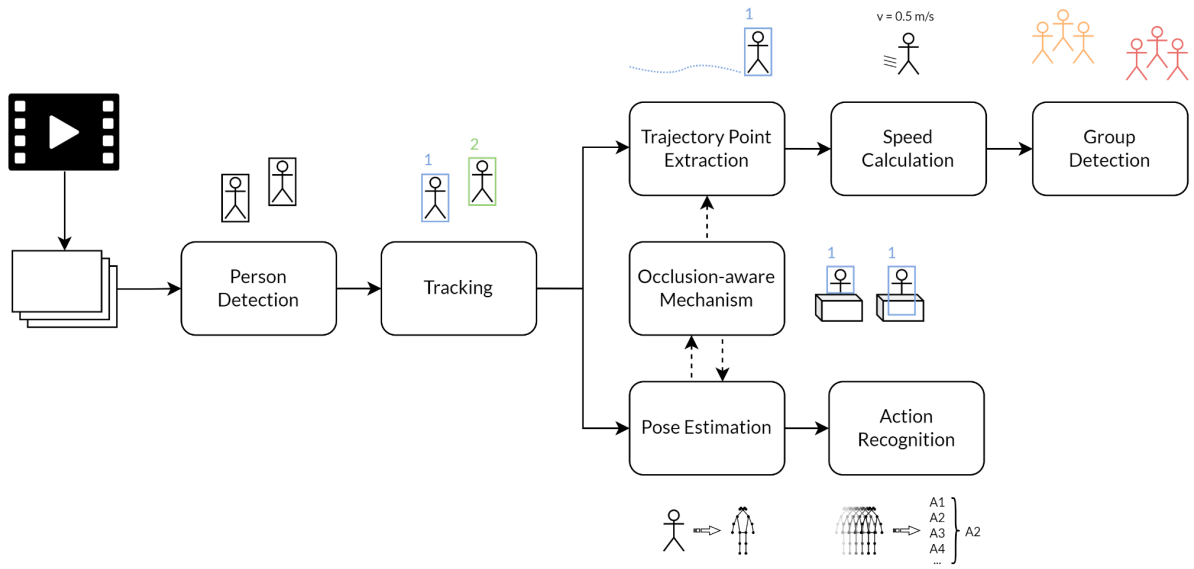


Figure 4.1. Diagram of the integration of the occlusion-aware mechanism into the proposed framework.

4.1.1. Impacts

There are several aspects that are negatively affected by the occurrence of occlusions. As previously mentioned, when a subject is occluded, the only area that is detected is the one that is visible. This leads to two problems: irregularities in the extracted trajectory points and, as a result, inconsistencies in the calculation of the speed at which the person walks (both when the feet are occluded). In Figure 4.2, a few occurrences of these cases are presented.

Regarding the inconsistencies in the calculation of the walking speed, given what is described in Section 3.1.4, these occur because as the bounding box shrinks and expands, the extracted trajectory points show a significant displacement in consecutive frames. Thus, when the frame-level trajectory points are projected into their world-level projections (2D floor plan), it creates the illusion that the person has travelled a greater distance than they actually have, and often on an irregular path.

In addition to these trajectory-related issues, occlusion cases also complicate the pose estimation process, as the position of occluded body parts cannot be estimated, which in turn makes the process of skeleton-based action recognition considerably more complex. Nevertheless, it is through this pose information that we were able to create the occlusion-aware mechanism, which is explained in detail in the following sections.

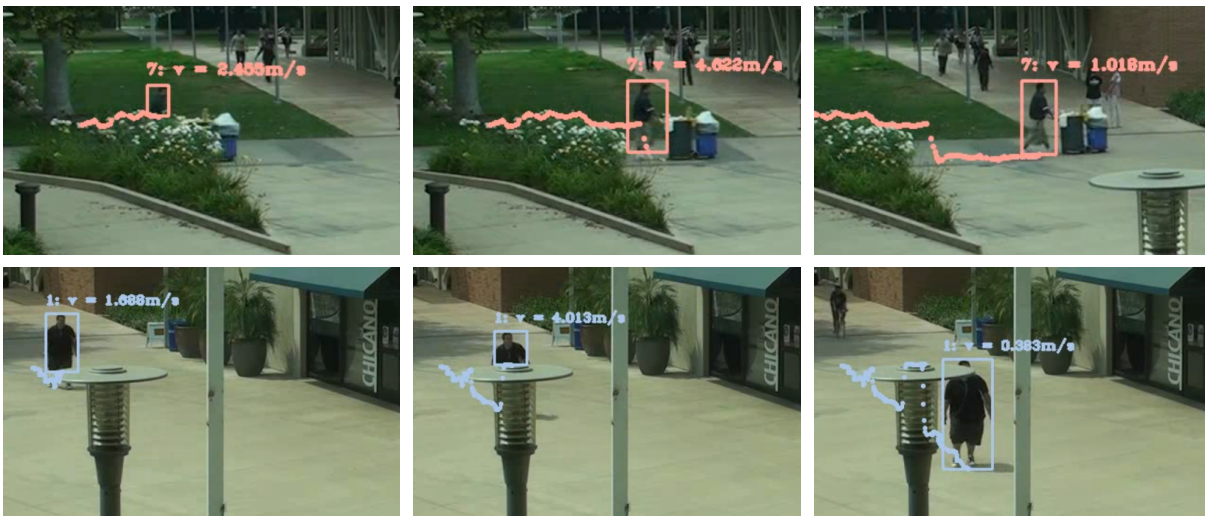
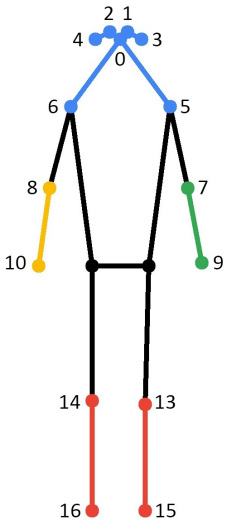


Figure 4.2. Examples of the impact of occlusions on the extracted trajectory points (footage from the VIRAT [60] dataset).

4.1.2. Detection

To mitigate these occlusion cases, it is first necessary to identify them. To do so, we decided to utilise the pose information generated by the HRNet human pose estimation algorithm. Thus, the skeleton of each individual is divided into four parts: upper part, lower part, left arm and right arm. Having the body parts established, and access to the confidence score of each joint, we determined that if more than 50% of the joints that constitute a body part have a low confidence score (a threshold of 0.3 was

used), then that body part is considered occluded. An illustration of the body part division process is depicted in Figure 4.3.



upper part	0 - nose 1 - left eye 2 - right eye 3 - left ear 4 - right ear 5 - left shoulder 6 - right shoulder
left arm	7 - left elbow 9 - left wrist
right arm	8 - right elbow 10 - right wrist
lower part	13 - left knee 14 - right knee 15 - left ankle 16 - right ankle

Figure 4.3. Correspondence between each joint and its respective body part. The hip body joints (11 and 12) were not taken into account for occlusion detection.

4.1.3. Mitigation

Depending on which body part is occluded, we apply different adjustments to the dimensions of the bounding box of the occluded person, in order to approximate them to the actual body proportions. Hence, considering that a bounding box is defined as $(x,\ y,\ w,\ h)$, where $(x,\ y)$ are the coordinates of the top-left corner point, and $(w,\ h)$ correspond to the width and height values of the bounding box, respectively, the following cases are considered:

A. *Arms* (the left and right arms are occluded) - The bounding box must be widened to include the location of the arms. To this end, two operations are performed, where one increases the width of the bounding box and the other translates its position to the left, ensuring it remains centred in relation to the body:

$$w' = w + (w \times factor) \quad (1)$$

$$x' = x - (w \times factor \times \tfrac{1}{2}) \quad (2)$$

   - $factor$ - defines the amount by which the width of the bounding box is increased.

B. *Lower Part* (the lower part of the body is occluded) - The bounding box must be heightened to include the location of the feet. For that, a modified sigmoid function $sig(\alpha)$ is calculated to define the amount by which the height of the bounding box is increased, given its aspect ratio ($\alpha = w \div h$). Hence, when the aspect ratio is lower (meaning that the bounding box is thinner and therefore the occluded area is smaller), the height does not increase as much as

when the aspect ratio is larger. Moreover, extreme aspect ratio cases are attenuated. A visual representation of $sig(\alpha)$ is shown in Figure 4.4. This process can be described as follows:

$$sig(\alpha) = \frac{amplitude}{1 + e^{-steepness(\alpha - centre)}} \quad (3)$$

$$h' = h + (h \times sig(\alpha)) \quad (4)$$

- $amplitude$ - represents the maximum scaling factor applied to the original height;
- $steepness$ - determines how steep the sigmoid function is;
- $centre$ - represents the midpoint of the sigmoid function.

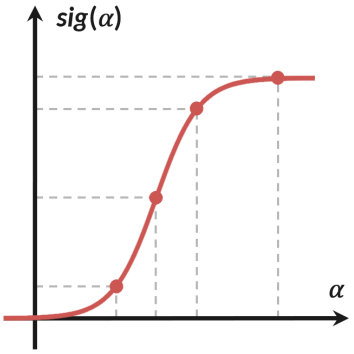


Figure 4.4. Illustrative example of the modified sigmoid function.

C. *Upper Part* (the upper part of the body is occluded) - The bounding box must be heightened to include the location of the head. This is similar to what is performed when the lower body part is occluded, but with the additional step of decrementing the y coordinate of the top-left point with the same value by which the height was increased:

$$y' = y - |h - h'| \quad (5)$$

For visual reference, an illustration of these processes is provided in Figure 4.5.

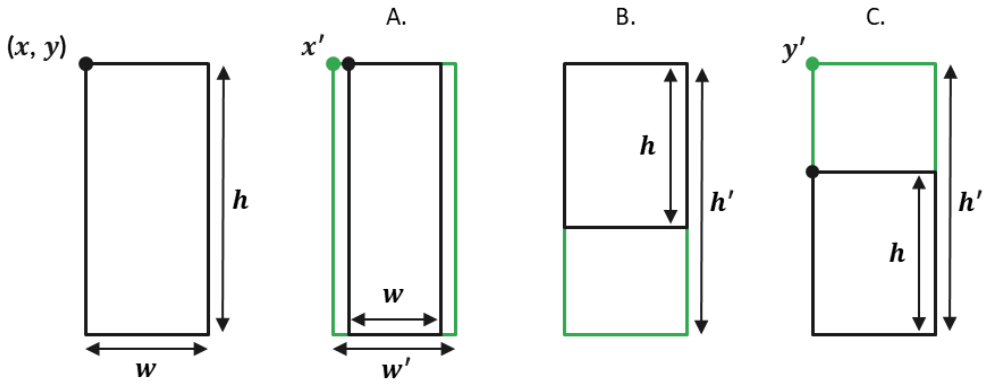


Figure 4.5. Illustrative examples of the dimensions of the bounding boxes, as well as the different adjustments applied to them in the considered cases (A, B and C).

4.2. Trajectory Irregularities

There are a few circumstances that can lead to oscillations in both the location and the dimensions of the bounding boxes throughout a sequence of consecutive video frames. These oscillations cause the extracted points to present irregularities in the trajectory. In order to mitigate these irregularities and, consecutively, generate more reliable path information, a smoothing method was applied.

4.2.1. Factors

Two major factors that may cause these unnatural changes in the values of the bounding boxes, and therefore create irregularities in the trajectory information of each person associated with them, are occlusions and gait movement. Illustrative examples of these factors are presented in Figure 4.6.

The former refers to when an individual is partially occluded by an object (addressed in depth in Section 4.1). This is an issue for the trajectory point extraction process because, as the person moves along the area covered by the occluding object, the extracted points accompany the contours of the obstacle rather than the feet of the individual. In part, this is largely resolved by the occlusion-aware mechanism we proposed. However, given that the mechanism does not take into account information concerning previous corrections, as a means of smoothing the adjustments performed, an additional post-processing step is required.

The latter, on the other hand, refers to the movements that the limbs perform while a person is walking. The object detectors often capture these movements by aligning the edges of the bounding boxes with the positions of the extremities of the body (in particular the hands and feet). The result is an undulatory effect in the trajectory of each moving person, which accompanies their gait.

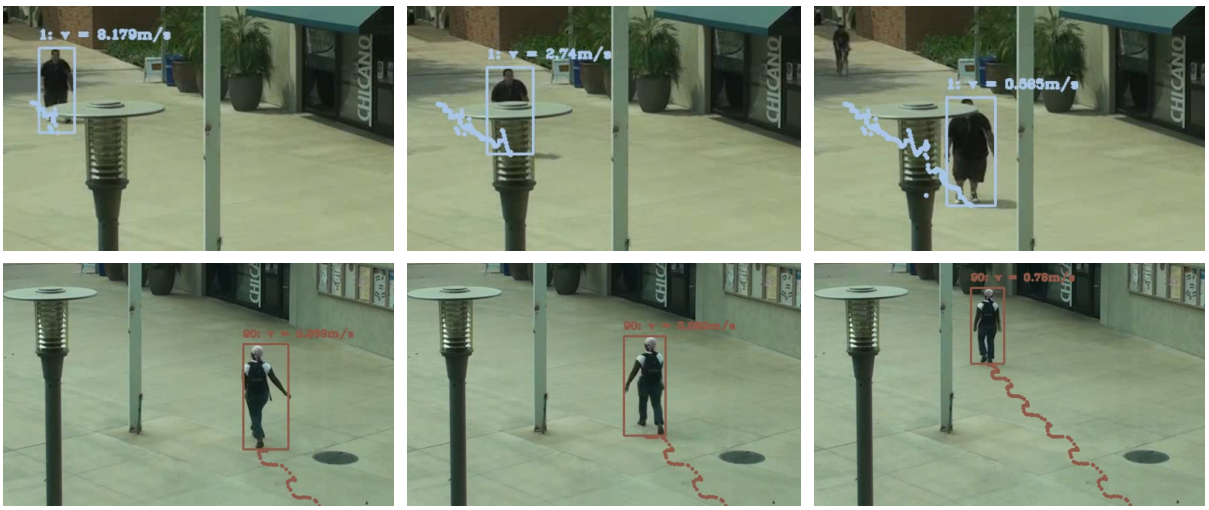


Figure 4.6. Examples of the two major factors that create irregularities in the extracted trajectory points: occlusions (top) and gait movement (bottom).

4.2.2. Smoothing

In order to attenuate the irregularities in the extracted trajectory points, we introduced a smoothing method. This method consists of calculating the moving average of the N most recent points, using a sliding window of fixed size (if fewer than N, only the available points are considered). Furthermore, a recurrent approach was adopted, meaning that the smoothing process executed for the current point considers the smoothing corrections applied to the previous points (within the N most recent).

To evaluate the effectiveness of this first approach, we performed a visual examination according to the factors that caused the irregularities. Regarding the oscillations caused by the manner in which each person walks (gait movement), using a sliding window of 5 points, the smoothing method was able to obtain satisfactory results. However, when addressing the oscillations caused by the presence of obstacles (occlusions), a larger window of 10 points was necessary. While this helped to attenuate the irregularities in the extracted trajectories, it introduced a significant delay in the smoothed points as only past points were considered. Therefore, the points calculated for a given frame, representing the closest location for each individual at that time, no longer aligned with their current position, but instead corresponded to their position from several frames prior. This problem is illustrated in Figure 4.7.

Figure 4.7. Example of applying the trajectory smoothing method to irregularities caused by obstacles: 5-point window (left) and 10-point window (right).

In this sense, to improve the trajectory smoothing process it was necessary to consider trajectory information from past and future video frames. Hence, we decided to apply the trajectory smoothing method twice. More specifically, given a video file, we store raw information regarding each extracted trajectory point (without smoothing), along with the corresponding person and video frame IDs. After that, we apply the trajectory smoothing method twice: a first time for the original trajectory points (as was done before) and a second time for the reversed trajectory points (as if the method was being applied to the video in reverse). In this way, we are able to generate two sets of trajectory points: one smoothed based on past trajectory information and another based on future trajectory information. Having these two sets, in order to obtain the final trajectories, it is necessary to match them based on the person and video frame IDs, combining the corresponding points.

Intuitively, this combination would be done by assigning a weight of 0.5 to each of the points, so that equal importance is given to both the past and future trajectory information. However, we need to consider the delays caused by the smoothing method, which, due to its recursive nature, become longer as the size of the smoothing window increases. Thus, at the beginning of the trajectory of each individual, these delays are more noticeable on the smoothed reversed points, whereas at the end of those trajectories, they are more noticeable on the smoothed original points. With this in mind, the assignment of weights is accomplished as follows (relative to a single trajectory point):

1. First, the interval that marks the beginning and end of the trajectory is defined (corresponds to the aforementioned delay factor).

$$interval = \frac{(N \times (N+1))}{2} \quad (6)$$

   This expression captures the range of influence of the smoothing process on each trajectory point, considering the sliding window (of size N) and the recursive nature of the smoothing. It provides a meaningful measure of how many points, including the current one, are involved in the smoothing calculation, taking into account the previous smoothing corrections applied within the sliding window.

2. Then, the weight for the smoothed original point is computed.

   - If the point is located at the beginning of the trajectory ($point\ index < interval$):

$$o\_weight = 1 - (\frac{1}{(2 \times interval)} \times point\ index) \quad (7)$$

   - If the point is located at the end of the trajectory ($point\ index > total\ points - interval$):

$$o\_weight = 0.5 - (\frac{1}{(2 \times interval)} \times (point\ index - (total\ points - interval))) \quad (8)$$

   - Else (general case): $o\_weight = 0.5$.

3. Finally, the weight for the smoothed reversed point is computed.

$$r\_weight = 1 - o\_weight \quad (9)$$

4.3. Results

Prior to conducting quantitative experiments, we first performed an initial qualitative analysis of the two contributions described above. These were specifically designed to mitigate the issues identified over the course of the design and development phases of the framework. One of the contributions, the occlusion-aware mechanism, was developed to detect occluded persons and dynamically adjust their bounding boxes to include the occluded areas. The other contribution, the trajectory smoothing method, aimed to minimise oscillations in trajectory points. The outcomes of this analysis are specific

to the application of the proposed framework to videos from the VIRAT [60] dataset.

4.3.1. Occlusion Awareness

Regarding the occlusion-aware mechanism, for the particular video samples of the VIRAT dataset, we empirically assigned the variables described in Section 4.1.3 with the following values:

- Width adjustments:
  - $factor = 0,25$
- Height adjustments ($sig(\alpha)$ parameters):
  - $amplitude = 1,5$
  - $steepness = 7,5$
  - $centre = 0,75$

Based on the obtained visual outputs, we were able to confirm the effectiveness of the proposed mechanism in detecting occluded body parts and estimating the actual body boundaries of occluded individuals. Some examples can be seen in Figure 4.8.

Figure 4.8. Examples of applying the proposed occlusion-aware mechanism: without occlusion awareness (top) and with occlusion awareness (bottom).

In cases where the lower body part was occluded, our method was able to predict the location of the feet of the occluded individual, so as to improve the accuracy of the resulting trajectory. However, given that the occlusion-aware mechanism takes the aspect ratio into account, and moving the arms influences the width of the bounding boxes, this creates some noise in the trajectory. Moreover, its performance is impacted by the fact that each person has its unique aspect ratio, owing to their body proportions. In certain instances, the posture also affects the aspect ratio. For example, in the case of cyclist occlusions, where they tend to lean forward, the width of the bounding boxes is larger, leading

to suboptimal results. The aforementioned cases are shown in Figure 4.9. Nevertheless, as evidenced in Section 5.1, the occlusion-aware mechanism proved to be an asset in handling occlusion cases.

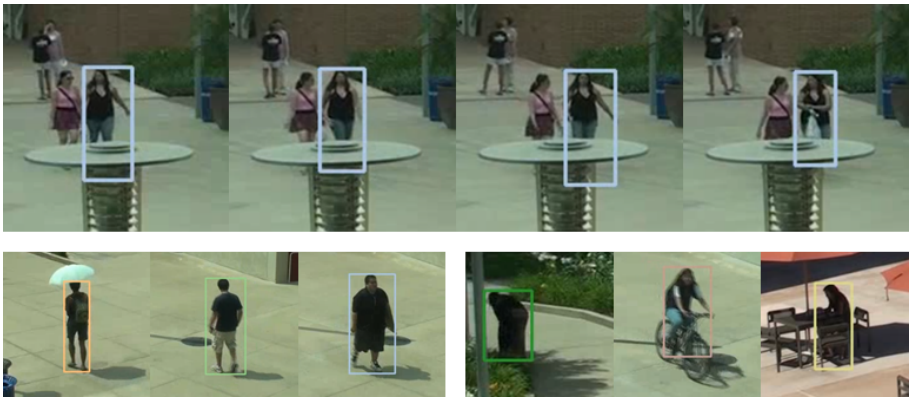


Figure 4.9. Shortcomings: width oscillations (top) and variable aspect ratios (bottom).

It is also worth mentioning that as the confidence scores of body joints are used to infer whether a body part is occluded, this may lead to some false positives, although their impact is not significant. On the other hand, when reapplying the HRNet algorithm to the corrected bounding boxes, the pose estimation results were generally improved, as depicted in Figure 4.10, which is expected to enhance the skeleton-based action recognition process, as the noise in the pose data is reduced.

Figure 4.10. Example of reapplying the HRNet algorithm to the corrected bounding boxes (only joints with a confidence score over the threshold of 0.3 are drawn): before occlusion awareness (left) and after occlusion awareness (right).

4.3.2. Trajectory Smoothing

In terms of the trajectory smoothing method, as discussed in Section 4.4.2, using only past trajectory information to perform the smoothing process introduced a delay in the resulting smoothed points, which became more pronounced as the size of the sliding window increased. Thus, to try to improve the smoothing process, we chose to incorporate both past and future trajectory information.

Based on the obtained visual outputs, we were able to confirm the added value of utilising future trajectory information, in addition to past trajectory information. Besides effectively solving the delay problem, the resulting trajectories became more precise and context-aware, meaning that the overall

performance of the trajectory smoothing method was significantly improved. Figure 4.11 represents an example of solving the delay problem.

Figure 4.11. Examples of applying the smoothing process (10-point window) using: only past trajectory information (left), and both past and future trajectory information (right).

Furthermore, we compared various sizes for the sliding window. Naturally, as the size increased, the resulting trajectories became smoother. However, in certain circumstances such as sharp turns, a larger size resulted in trajectories deviating from the actual path taken by the subject. For this specific use case, in order to attain an appropriate balance between smoothness and precision, we decided to adopt a sliding window with a size of 8 trajectory points. An example of this trade-off is illustrated in Figure 4.12.

Figure 4.12. Examples of applying the trajectory smoothing process on a sharp turn: 5-point window (left), 8-point window (middle), and 10-point window (right).

With this in mind, using an 8-point sliding window, the trajectory smoothing method proved to be effective in mitigating the impact of oscillations on the trajectory points. It successfully addressed both types of oscillations: those caused by gait movement and those resulting from occlusions. Some examples can be observed in Figures 4.13 and 4.14. Figure 4.13 illustrates the impact of the trajectory smoothing method on points derived from the initial bounding boxes, whereas Figure 4.14 illustrates the effect of applying the trajectory smoothing method to points obtained from the bounding boxes adjusted by the occlusion-aware mechanism (in scenarios involving occlusions).

In addition, Figure 4.15 shows a comparison of the floor plan projections (top view) using the first example depicted in Figures 4.13 and 4.14. In this comparison, we present the projections before and after applying the occlusion-aware mechanism and the trajectory smoothing method. This allows for a clearer understanding of the influence of these two contributions on trajectory mapping.

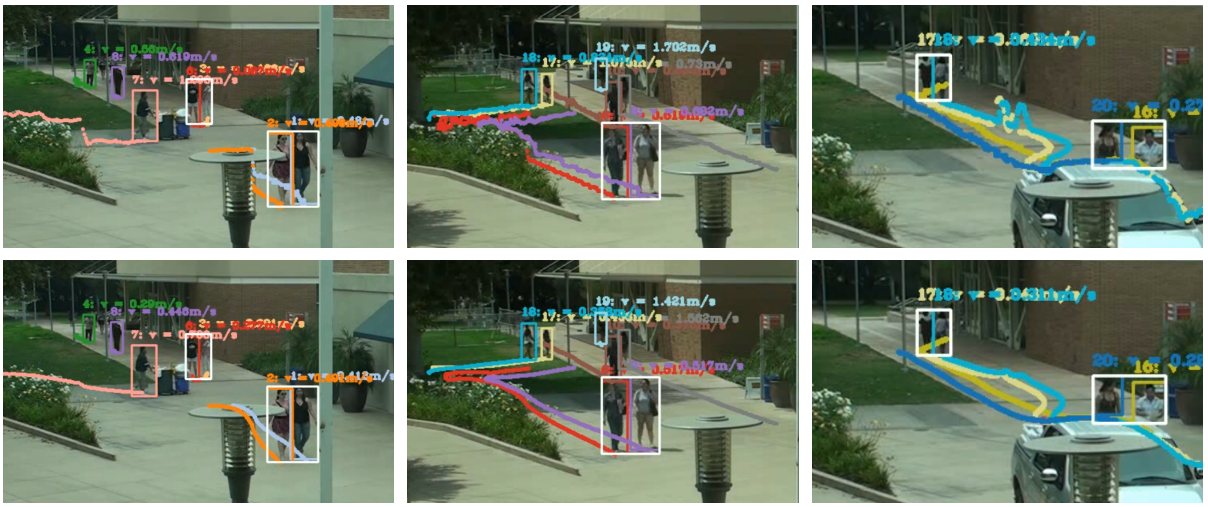


Figure 4.13. Examples of applying the proposed trajectory smoothing method (without occlusion awareness): without trajectory smoothing (top) and with trajectory smoothing (bottom).

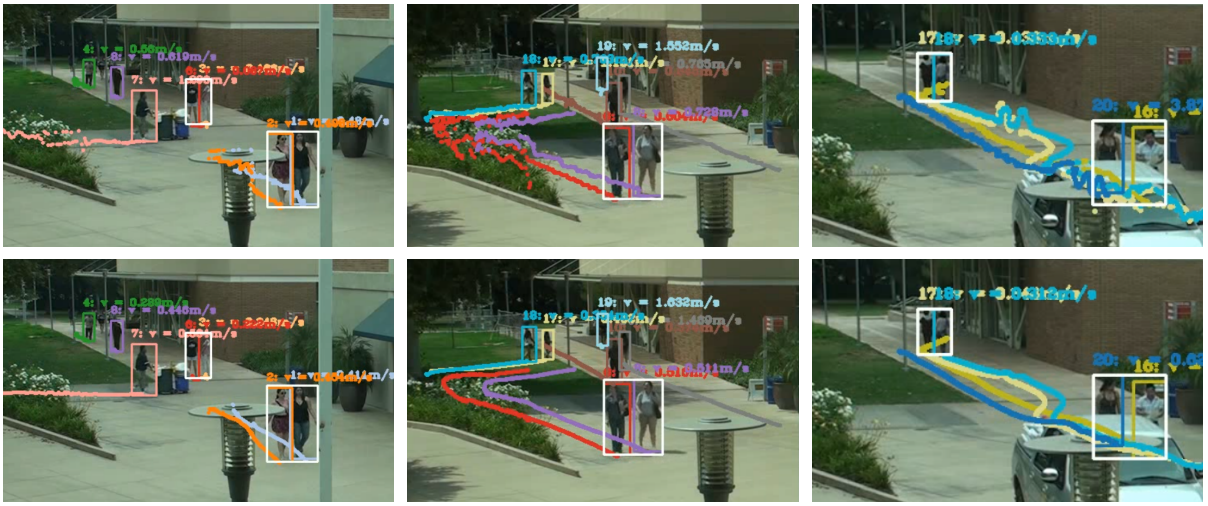


Figure 4.14. Examples of applying the proposed trajectory smoothing method (with occlusion awareness): without trajectory smoothing (top) and with trajectory smoothing (bottom).

Figure 4.15. Example of floor plan projections before (left) and after (right) applying the occlusion-aware mechanism and trajectory smoothing.

CHAPTER 5

Experiments

In this chapter, we explore two important aspects of our framework. Firstly, we report on the rigorous quantitative assessment performed on the proposed occlusion-aware mechanism. Secondly, we delve into the extensive process of training the skeleton-based action recognition models.

5.1. Occlusion-Aware Mechanism

In addition to the qualitative analysis presented in Section 4.3.1, we also carried out a comprehensive quantitative and statistical analysis of the proposed occlusion-aware mechanism in order to provide a rigorous, data-driven assessment of its performance.

To this end, we started by manually selecting occlusion cases representative of the typical variety of occlusion scenarios. As a result, we acquired 35 instances from 21 videos in the VIRAT [60] dataset. These instances had an average duration of approximately 139 frames (with a median of 100 frames), corresponding to the period from when the individual was occluded until the moment they were no longer covered by the obstacle. After the selection process, we used the ground truth bounding boxes provided in the dataset (which depict the precise location and proportions of occluded individuals) to compute two separate lists of Intersection over Union (IoU) values. Specifically, for each instance, we calculated IoU scores for the original predicted bounding boxes and for the adjusted ones, separately comparing them to the ground truth (an example is illustrated in Figure 5.1). These calculations were then averaged to generate a single IoU value for the original detections and another for the adjusted ones, representing their overall quality. Ultimately, we ended up with two sets (paired samples), each consisting of 35 averaged IoU values, intended for evaluating the extent to which the occlusion-aware mechanism enhanced the accuracy of detections in scenarios involving occlusion.

IoU is a metric commonly applied to quantify the degree of overlap between predicted bounding boxes and ground truth bounding boxes. In particular, for a pair of these bounding boxes, it produces a value between 0 and 1, where 0 indicates no overlap (complete mismatch), and 1 denotes a perfect match, meaning that the predicted bounding box is perfectly aligned with the respective ground truth bounding box. It is calculated using the following formula, where $Intersection\ area$ corresponds to the area of overlap between the two bounding boxes (common area), and $Union\ area$ is the area covered by both bounding boxes (total area):

$$IoU = \frac{Intersection\ area}{Union\ area} \tag{10}$$

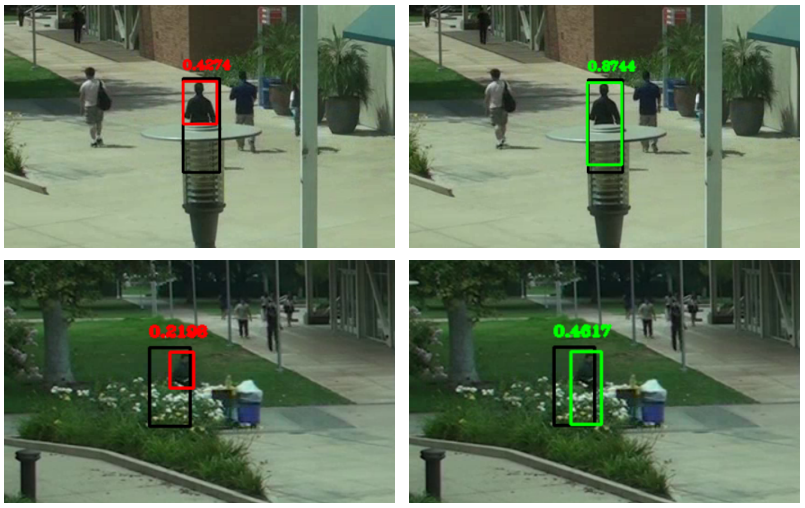


Figure 5.1. Example of IoU scores for the original predicted bounding boxes (left) and the adjusted predicted bounding boxes (right), in one video frame. The ground truth is depicted in black.

With the two sets of IoU values prepared, in order to assess the improvements resulting from the application of the occlusion-aware mechanism, we computed descriptive statistics for each set. These are shown in Table 5.1, where sample A represents the IoU values derived from the original predicted bounding boxes, and sample B denotes those derived from the adjusted predicted bounding boxes. It should be noted that the ground truth bounding boxes provided in the dataset were not perfect, so in some cases this resulted in lower IoU scores for both the original and adjusted detections (as seen in Figure 5.1). Nevertheless, since it affected each set in a similar way, there was no significant effect on the evaluation outcomes.

Table 5.1. Paired samples descriptive statistics.

Sample	Size (N)	Mean (M)	Median (Mdn)	Standard Deviation (SD)	Coefficient of Variation (CV)	Skewness	Kurtosis
A	35	0.439	0.471	0.116	0.263	-0.681	-0.188
B	35	0.579	0.575	0.112	0.193	-0.214	-0.345

Based on the descriptives presented in Table 5.1, it was evident that sample B had a higher mean and median (M = 0.579, Mdn = 0.575) compared to sample A (M = 0.439, Mdn = 0.471). This indicates that the adjusted detections in B, resulting from the occlusion-aware mechanism, had indeed shown improvements when compared to the original detections in A. Moreover, both the standard deviation and the coefficient of variation showed that the IoU values in B (SD = 0.112, CV = 0.193) exhibited less variability around their mean compared to those in A (SD = 0.116, CV = 0.263). This implies that, from a general perspective, the adjustments produced by the occlusion-aware mechanism were consistent, while also suggesting that certain adjusted instances had more significant improvements than others. Finally, regarding the shape of the distributions, sample B exhibited less negative

skewness compared to sample A, meaning that the adjustments in B resulted in a less left-skewed distribution compared to the original detections in A. In other words, a larger proportion of detections in B achieved higher IoU scores, reducing the left tail of the distribution. Furthermore, both samples had negative kurtosis values, indicating platykurtic distributions. Specifically, sample B had even lower kurtosis compared to sample A, meaning that the distribution in B was flatter and had thinner tails (IoU scores were more evenly spread across a wider range). Figure 5.2 shows a visual depiction of these distributions, as well as the characteristics of the data.

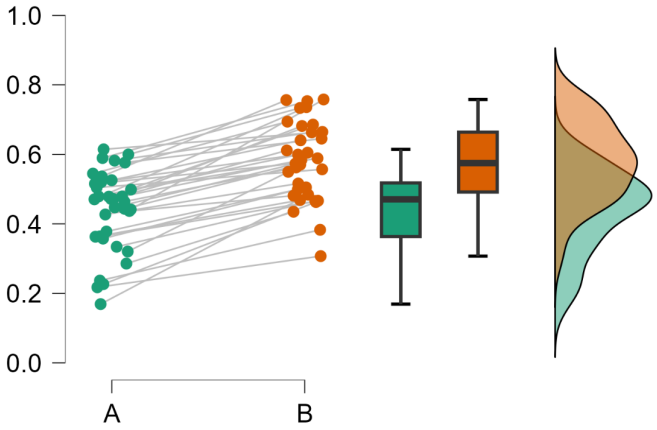


Figure 5.2. Graphical display of the characteristics of samples A and B.

Having examined the descriptive statistics and compared the differences between samples A and B, the next phase of our analysis involved rigorously evaluating the significance of these differences. For this purpose, we employed hypothesis testing as a way of scientifically determining whether the improvements observed in the IoU scores of sample B, attributed to the occlusion-aware mechanism, were statistically significant. Typical options include the paired Student's t-test [69] (parametric) and the Wilcoxon signed-rank test [70] (nonparametric alternative). Parametric tests assume that the data follows a particular distribution, typically the normal distribution, and rely on parameters such as the mean and variance. Nonparametric tests, on the other hand, make less or no assumptions regarding the underlying data distribution and are therefore more suitable when handling data that does not meet the assumptions of parametric tests. However, if the data meets the assumptions, parametric tests are often preferred due to their greater statistical power.

With this in mind, we began by examining the assumptions underlying the paired Student's t-test to evaluate its appropriateness for our study. This examination focused primarily on verifying whether the differences between paired data points (sample B − sample A) were roughly normally distributed, which is a fundamental assumption of the paired Student's t-test (normality). It should be noted that other assumptions of the t-test had already been inherently satisfied due to the characteristics of the data. These assumptions include continuous data (IoU scores are continuous numeric values), paired data (each data point in sample A corresponds to a specific data point in sample B), random sampling

(the instances were obtained through a random selection process), independence (the instances are not related to each other), and absence of outliers (no extreme values are present).

To check the normality assumption of the paired Student's t-test, we used the Shapiro-Wilk test [71], a widely used statistical test that evaluates whether a given set follows a normal distribution. In this examination, we applied the Shapiro-Wilk test to the differences between paired data points (as expected for the t-test) to assess the distribution of the improvements in IoU scores between sample A and sample B. In addition, we applied the Shapiro-Wilk test separately to each sample. The results are presented in Table 5.2.

Table 5.2. Shapiro-Wilk test results.

	Statistic (W)	**Significance (*p*-value)**
A	0.943	0.071
B	0.973	0.521
B − *A*	0.908	0.007

The null hypothesis in the Shapiro-Wilk test assumes that the data follows a normal distribution. Therefore, if the *p*-value is greater than a significance level of .05, we fail to reject the null hypothesis, which suggests that there is not enough evidence to conclude a deviation from normality in the data. On the other hand, if the *p*-value is less than .05, we reject the null hypothesis, implying that the data does not follow a normal distribution. Thus, based on the obtained results, both sample A (p = 0.071) and sample B (p = 0.521) were assumed to follow a normal distribution, given that both their *p*-values were greater than .05. However, when analysing the differences between these samples, the *p*-value of 0.007 suggested that they deviated from a normal distribution. To assess the degree of deviation, we decided to generate a normal probability plot on these differences (shown in Figure 5.3).

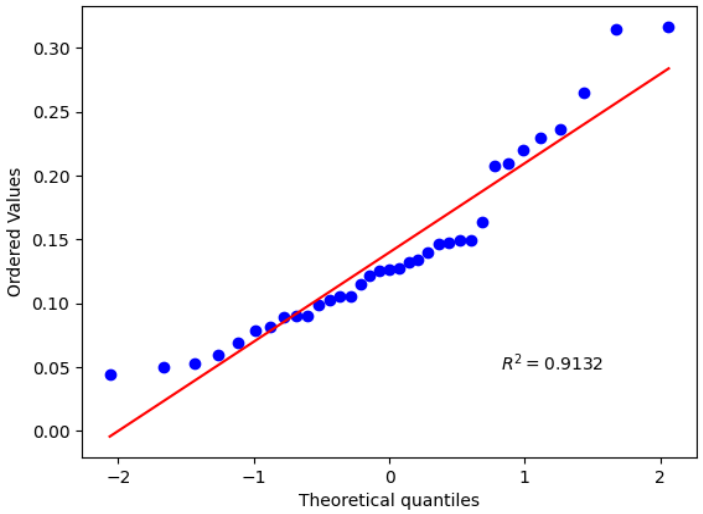


Figure 5.3. Normal probability plot on the differences between samples A and B.

A normal probability plot is a graphical tool used to assess the degree to which a set conforms to a normal distribution. It compares the quantiles of the observed data (i.e. the differences arranged in ascending order) against the expected quantiles of a normal distribution. If the data follows a normal distribution, the points on the plot should align along a straight line. Deviations from linearity indicate departures from normality. For a more quantitative interpretation, these plots also provide numerical measures to evaluate how closely the data adheres to a normal distribution. One such measure is the coefficient of determination (R^2), which quantifies the goodness of fit between the observed data and the expected normal distribution, on a scale of 0 to 1. In this particular analysis, the obtained R^2 value of 0.9132 suggests that there is a reasonably good fit, as it is close to 1. While the observed data does not perfectly follow a normal distribution, the extent of deviation appears relatively modest.

Considering the robustness of the paired Student's t-test to departures from normality, especially when dealing with reasonably large sample sizes (equal to or larger than 30) due to the Central Limit Theorem [72], we decided to use both the paired Student's t-test and the Wilcoxon test. The purpose was to ascertain whether both tests yielded similar outcomes, ensuring the reliability and robustness of the findings.

Table 5.3. Statistical hypothesis tests results for the differences between samples A and B.

	Mean	**Std. Deviation**	**Test**	**Statistic**	**Z**	**Degrees of Freedom (df)**	**Significance (*p*-value)**	**Effect Size***
B − *A*	0.140	0.071	*Student*	11.698	–	34	< .001	1.977
			Wilcoxon	630.000	5.159	–	< .001	1.000

* Student's t-test: Cohen's d; Wilcoxon test: matched rank biserial correlation.

The null hypothesis for both tests state that there is no significant difference between the paired observations. More specifically, the paired Student's t-test focuses on the mean difference, while the Wilcoxon test evaluates the distribution of the signed rank differences, which is related to the median difference. As shown in Table 5.3 both tests produced extremely low *p*-values (paired Student's t-test: t(34) = 11.70, $p < .001$; Wilcoxon test: z = 5.16, $p < .001$), rejecting the null hypothesis at a significance level of .05 (*p*-value is less than .05). This provides strong statistical evidence that the occlusion-aware mechanism significantly enhanced detections for occlusion cases. In other words, it is highly unlikely that the differences in IoU scores between sample A and sample B occurred by chance.

Furthermore, to analyse the practical implications of these differences and given the similarity of results from both tests, we decided to assess the effect size of the paired Student's t-test (Cohen's d), leveraging its advantages in providing a standardised and easily interpretable measure. The retrieved Cohen's d value was 1.98, indicating a substantial effect size, surpassing the conventional threshold of 0.8 for a *large* effect size [73]. To put it into perspective, a value of 1.98 indicates that the differences

in IoU scores between samples A and B were nearly two standard deviations (SD = 0.071) apart. This suggests that the adjustments performed by the occlusion-aware mechanism had a significant impact on the IoU scores, emphasising the real-world significance of these improvements.

5.2. Skeleton-Based Action Recognition

In this section, a detailed overview of the different stages required for preparing the skeleton-based action recognition models is provided, including the decisions made during each phase. These stages comprise Data Preparation, Modelling, and Evaluation, which correspond to the development phases of the CRISP-DM process model described in Section 1.4.

5.2.1. Data Preparation

As mentioned in Section 3.2.8, for experiments involving the action recognition process, we chose to use the PIP [63] dataset, in particular, the PIP 370k stabilised dataset (described in Section 3.2.4). This dataset comprises 405,781 videos of 67 action classes (subsets of the 37 action classes in the MEVA [61] dataset). The distribution of videos per action class is illustrated in Figure 5.4.

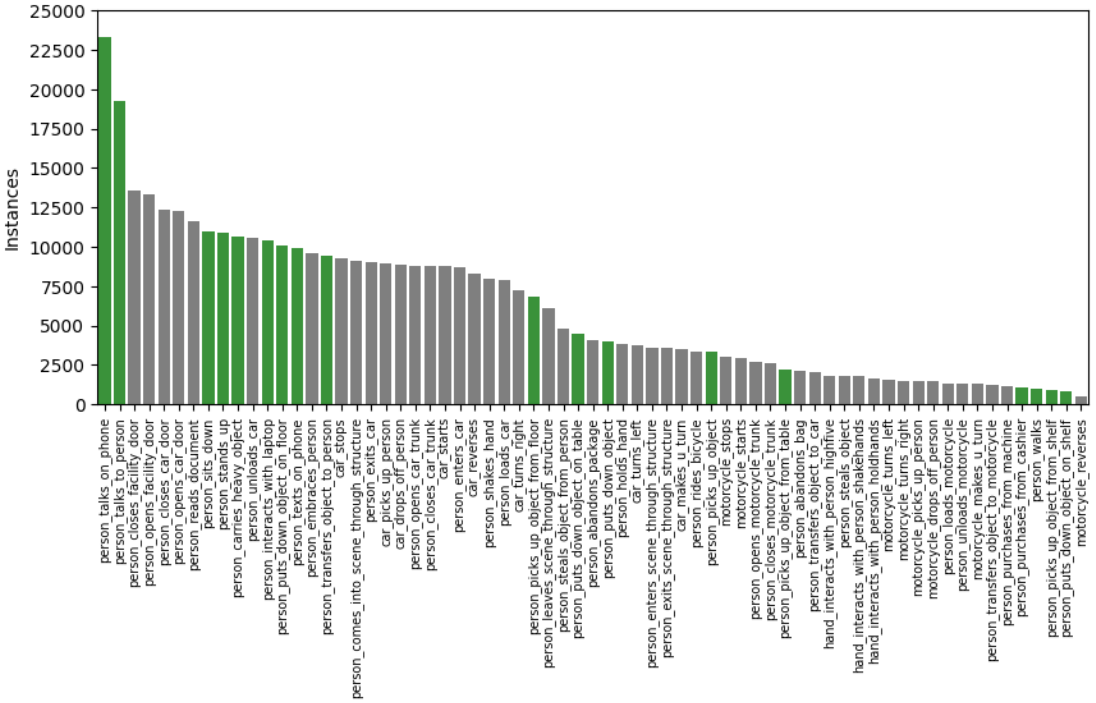


Figure 5.4. Distribution of videos per action class in the PIP [63] dataset.

However, a large part of the action classes included in the dataset are not related to the topic of this dissertation (in-store customer behaviour), such as vehicle activity and hand interaction classes. With this in mind, we only selected those relevant to our task, represented as the green bars in Figure 5.4. Furthermore, we merged and simplified some of the selected classes, as these were redundant (shown in Table 5.4). This resulted in a subset of PIP that we refer to as the PIP Retail dataset, which

encompasses 12 distinct action classes spanning a total of 139,236 video files. Figure 5.5 displays the label assigned to each action class, along with the corresponding count of associated videos.

Table 5.4. Labelling correspondence for some of the selected action classes.

PIP	**PIP Retail**
- person_picks_up_object - person_picks_up_object_from_floor - person_picks_up_object_from_shelf - person_picks_up_object_from_table	- person_picks_object
- person_puts_down_object - person_puts_down_object_on_floor - person_puts_down_object_on_shelf - person_puts_down_object_on_table	- person_places_object
- person_carries_heavy_object	- person_carries_object
- person_texts_on_phone	- person_interacts_with_phone

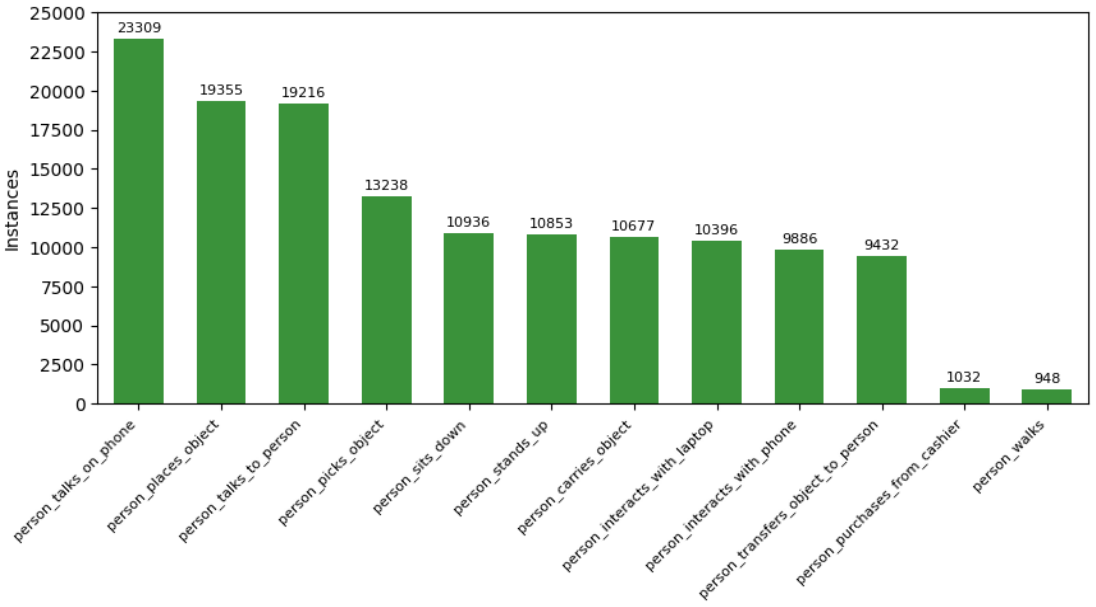


Figure 5.5. Distribution of videos per action class in the PIP Retail dataset.

Annotations for each video include the bounding boxes that surround the subject performing the action over time (actor), the label of the action class, and the frame interval at which it occurs, along with additional metadata (e.g. video ID, dimensions, and frame rate). These annotations are grouped into a single JSON representing the entire dataset (VIPY annotation format), which can be intuitively accessed using the open source VIPY [65] python package. VIPY was developed by the authors of the dataset for representing, transforming and visualising annotated videos and images. It provides tools for applying transformations such as downsampling, padding, scaling, cropping and rotating, so that the annotations are transformed along with the video pixels.

Moreover, it should be noted that the authors of the dataset performed post-processing tasks to the collected videos. For instance, given that the hand-held cameras were used to record the videos, the background is not stabilised. Therefore, background stabilisation was carried out using an affine coarse-to-fine optical-flow method. This procedure was designed to minimise distortion for motion in the regions around the centre of the actor bounding box, as if the cameras were rigidly mounted on a tripod. In addition, bounding boxes were refined and transformed to be aligned with the background stabilised video.

In this context, given that the PIP Retail dataset does not include information about the pose of the actors, we executed the procedures outlined in Section 3.1.7 regarding the creation of a custom skeleton dataset. Initially, we gathered the necessary annotations to generate the pickle files for each of the 139,236 videos, which include the video ID (frame_dir), the action label (label), the dimensions of the video (img_shape and original_shape), the duration of the action sequence (total_frames), the skeleton sequence (keypoint), and the keypoint confidence scores (keypoint_score).

The 'frame_dir', 'label', 'img_shape', and 'original_shape' fields were directly retrieved from the JSON annotation file of the PIP Retail dataset. To ensure compatibility with the 'label' field, which only accepts integer values, we assigned each action class with a number, ranging from 0 to 11. As for the 'total_frames', 'keypoint', and 'keypoint_score' fields, based on the annotated bounding boxes, which define the location of the actor over time, and the frame interval at which the action was performed, it was possible to extract skeleton sequences corresponding to the poses that were estimated within that interval. For this purpose, the HRNet [25] human pose estimation algorithm was employed (see Section 3.1.6). This procedure resulted in 139,236 pickle files with the following structure (example of one of the pickle files):

```
{
    'frame_dir': '20200517_1233581741317354_2',
    'label': 10,  # person_stands_up
    'img_shape': (1276, 1536),
    'original_shape': (1276, 1536),
    'total_frames': 65,
    'keypoint': array([[[[769.27, 567.75], [772.20, 563.35], …]]])
    'keypoint_score': array([[[0.9366, 0.9120, ...]]])
}
```

After generating the pickle files, we gathered them into separate lists for training and validation purposes. In particular, for each action class, we split the pickle files into 80% for training and 20% for validation. Subsequently, we saved each of these lists as individual pickle files: “pip_retail_train.pkl”

for the training set and “pip_retail_val.pkl” for the validation set.

It is important to emphasise that the extraction of skeleton sequences from 139,236 video files is a slow and resource-intensive operation, which was estimated to take several weeks to a few months to complete. Hence, in the midst of this time-consuming process, a decision was made to generate an additional, smaller version of the PIP Retail dataset. This version was assembled using the pickle files that had been generated so far. To do so, we selected a fixed number of shuffled pickle files from each class, making a total of 13,000 instances. The distribution of instances per action class is illustrated in Figure 5.6. The process of gathering the selected files into training and validation sets was performed using the aforementioned percentages, which resulted in the pickle files: “pip_retail_small_train.pkl” and “pip_retail_small_val.pkl”.

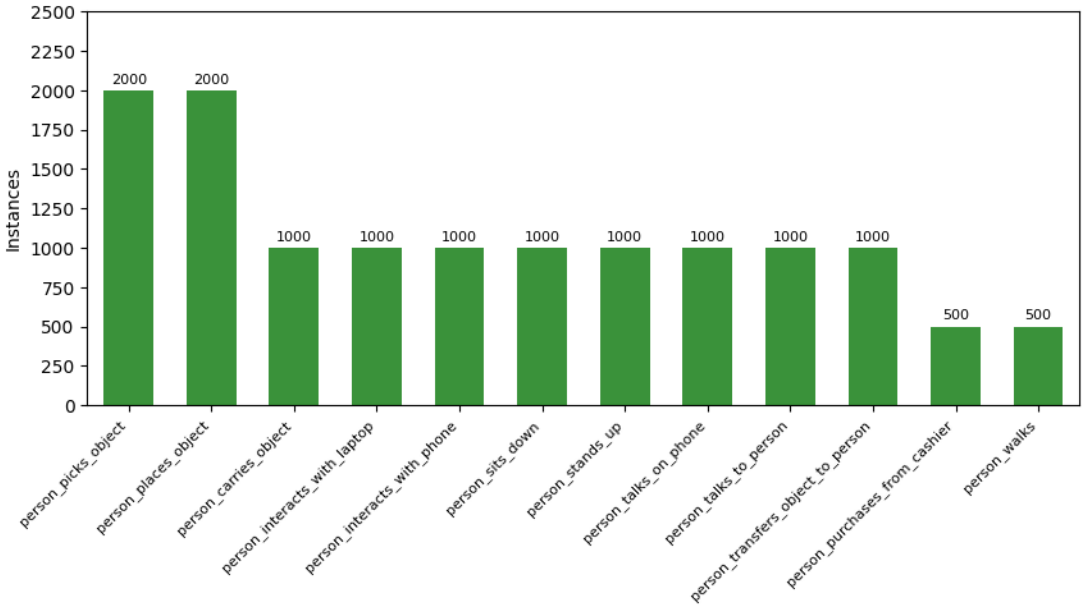


Figure 5.6. Distribution of videos per action class in the PIP Retail Small dataset.

As for testing, we opted to use the validation set instead of creating an additional split specifically for that purpose. The rationale for not including a test set was that we considered the PIP Retail Small dataset to be too small for an additional split to be created. We maintained this same strategy for the PIP Retail dataset to ensure that the results of both the small and complete versions could be directly compared.

5.2.2. Modelling

Once the datasets were prepared, we could then proceed to train skeleton-based action recognition models, and subsequently test them. As mentioned in Section 3.1.7, we used the MMAction2 toolkit [59] version 0.24.1, which provides three skeleton-based action recognition models, i.e. ST-GCN [42], AGCN [43] and PoseC3D [23].

Both ST-GCN and AGCN leverage Graph Convolutional Networks (GCNs) to model the spatial and temporal dependencies in skeleton data. Thus, human skeleton sequences are represented as graphs, where each joint is a node, and the edges that connect the nodes represent the spatial and temporal relationships between the joints. However, as opposed to ST-GCN where the topology of the graph is defined manually and remains fixed over all layers and input samples, AGCN parameterises the graph structure and incorporates it into the network, allowing it to be learned and updated with the model. This enables the network to learn specific relationships between joints that are relevant to the action being performed. Moreover, AGCN incorporates a two-stream design to simultaneously model both first-order (joint coordinates) and second-order (bone lengths and directions) information, in contrast to ST-GCN, which only considers first-order information.

PoseC3D was proposed as an alternative to the GCN-based approaches. It also takes 2D skeleton sequences as input, which are obtained through human pose estimation algorithms. However, rather than processing coordinates on a human skeleton graph, it represents the 2D poses as stacks of joint heatmaps. In this approach, the heatmaps from different time steps are combined along the temporal dimension, creating a 3D heatmap volume. PoseConv3D then applies a 3D CNN over this 3D heatmap volume to recognise actions. Compared to GCN-based methods, PoseC3D is more effective at learning spatio-temporal features, more robust against pose estimation noise, and can handle multiple-person scenarios without incurring additional computational costs.

Each of these models has an associated configuration file (.py extension) to conduct experiments. These configuration files follow a modular, inheritance-based design pattern, defining aspects such as the model architecture, dataset specifications (including data loading and augmentation pipelines for the training, validation, and testing processes), optimiser settings, learning rate schemes, and runtime configurations (e.g. number of epochs). In addition, the MMAction2 toolkit provides tools for training and testing the models based on these configuration files, i.e. the “train.py” and “test.py” scripts. The “train.py” script is used to train the models. During this process, it generates logs containing relevant statistics, and checkpoint files (.pth extension) representing the state of the model at various training stages. The “test.py” script is then used to test the models. It loads one of the saved checkpoints and evaluates the performance of the model according to the selected metrics. It also generates a results file (.json extension) for offline evaluation.

With this in mind, we started by running the “train.py” script for the aforementioned models on the PIP Retail Small dataset. Some important details regarding the training process of each model, as well as results obtained from the logs (including execution time per epoch and *top-K accuracy* scores), are shown in Tables 5.5 and 5.6, respectively.

The *top-K accuracy* is the metric we chose to evaluate the checkpoints generated for each model during training. In the context of these models, predictions for a given instance are made by assigning a probability (confidence score) to each possible class. With this in mind, the *top-K accuracy* considers the *K* model predictions with higher probability. If one of them matches the true class, the prediction is classified as correct. The *top-1 accuracy*, referred to simply as *accuracy*, is a particular case in which only the prediction with the highest probability is taken into account. Besides the *top-1 accuracy*, we also used the *top-2 accuracy* (*K* = 2) in order to select the best-performing checkpoint for each model, whose validation scores are presented in Table 5.6.

Table 5.5. Training process details.

Model	**Optimiser**	**Learning Rate**	**Momentum**	**Weight Decay**	**Batch Size**	**Max Epochs**
ST-GCN	Stochastic Gradient Descent (SGD)	0.1 (Step Decay)	0.9	0.0001	16	80
AGCN		0.1 (Step Decay)	0.9	0.0001	16	80
PoseC3D		0.2 (Cosine Annealing)	0.9	0.0003	16	240

Table 5.6. Training results for the PIP Retail Small dataset.

Model	**Execution Time per Epoch (mm:ss.SSS)**	**Best Checkpoint**	**Validation Top-1 Accuracy**	**Validation Top-2 Accuracy**
ST-GCN	01:45.988	70	0.7588	0.8754
AGCN	02:58.013	80	0.8038	0.9069
PoseC3D	05:15.279	240	0.8423	0.9408

Based on the training results, we noticed that ST-GCN is the lightest but worst performing model, while PoseC3D is the heaviest but best performing model. On the other hand, AGCN presents a good trade-off between accuracy and computational efficiency. Furthermore, we also noticed a significant increase in the *top-2 accuracy* scores compared to the *top-1*, implying that approximately 10% of the total number of instances from the validation set were incorrectly predicted as the first choice of the models, but correctly predicted as their second choice.

Subsequently, we executed the “test.py” script on the best checkpoint generated for each model during training. The metrics chosen to further evaluate these checkpoints were the *accuracy*, as well as the macro-averaged *recall*, *precision*, and *F1-score*. The macro-averaged measures were calculated in order to obtain an indication of how well a model performs for each individual class, as opposed to simply considering the overall accuracy, which can be misleading when classes are unbalanced. These metrics are detailed below.

- The *accuracy* refers to the overall performance of a classification model, and is defined as the ratio of correctly predicted instances to the total number of instances.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \quad (11)$$

- The *recall*, in multi-class classification, indicates the proportion of instances of a class that the model correctly predicted among all the instances that belong to that class. It is expressed as the ratio of true positive predictions to the total number of actual positive instances.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (12)$$

- The *precision*, in multi-class classification, reflects the fraction of correctly predicted instances for a certain class among all the instances that the model predicted as being of that class. It is the ratio of true positive predictions to the total number of positive predictions made.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (13)$$

- The *F1-score* corresponds to the harmonic mean of *precision* and *recall*. It provides a balance between the two metrics, considering both false negatives and false positives.

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (14)$$

To get the macro-averaged *precision*, *recall*, and *F1-score*, these metrics are calculated separately for each action class, and then the values are averaged across all classes. This ensures that each class contributes equally to the final metric, regardless of class imbalance.

The results obtained from the tests are illustrated in Figures 5.7, 5.8, and 5.9, which present the normalised confusion matrices resulting from the ST-GCN, AGCN, and PoseC3D models, respectively. The performance scores relative to the metrics described above are shown in Table 5.10.

Upon analysing the generated confusion matrices, we noticed that, in general, the highest degree of confusion occurred between the “person_picks_object” and “person_places_object” classes. This can be attributed to the similarity of the arm movements in these two action classes, bearing in mind that the models solely rely on pose information and do not consider appearance cues to determine whether the person is holding an object. Similarly, the “person_sits_down” and “person_stands_up” classes were occasionally confused with each other due to noise in the respective videos, given that in some instances the actor would perform both of these actions consecutively. Moreover, the class “person_purchases_from_cashier” was often mistaken for others that can be perceived as subclasses (i.e. atomic interactions between the person and the cashier), which include “person_picks_object”, “person_places_object”, “person_transfers_object_to_person”, and “person_talks_to_person”.

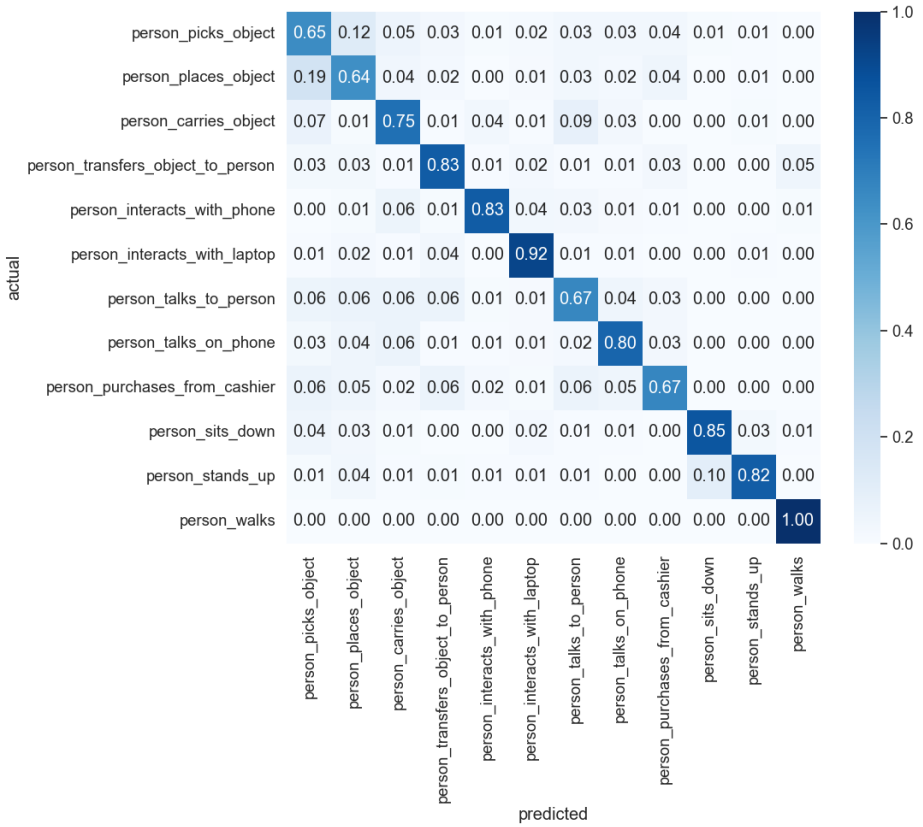


Figure 5.7. Confusion matrix of the ST-GCN model in the PIP Retail Small dataset.

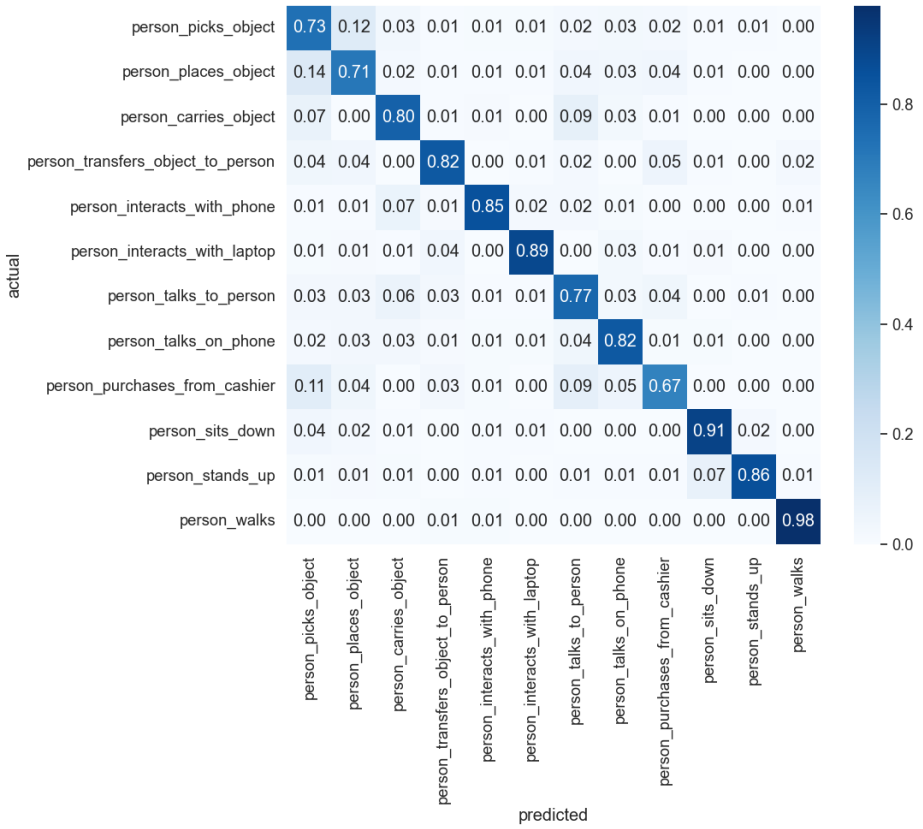


Figure 5.8. Confusion matrix of the AGCN model in the PIP Retail Small dataset.

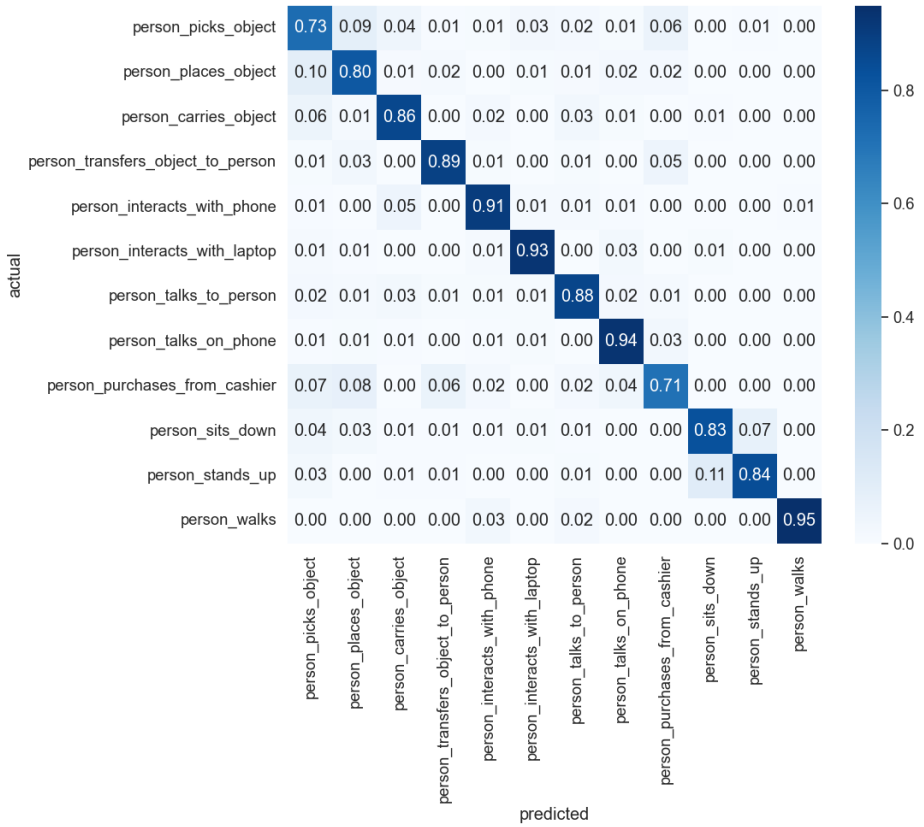


Figure 5.9. Confusion matrix of the PoseC3D model in the PIP Retail Small dataset.

To enhance the results on the PIP Retail Small dataset, we opted to fine-tune model checkpoints provided in the MMAction2 toolkit, which were originally trained on large-scale benchmark datasets. Starting with ST-GCN, the only available checkpoint that leveraged 2D skeleton sequences was trained on the NTU60-XSub [74] dataset. Therefore, we selected that particular checkpoint for fine-tuning. As for AGCN, all the available checkpoints resulted from training processes performed using 3D skeleton sequences, which made it unviable to experiment with fine-tuning using this model. Finally, regarding PoseC3D, the provided checkpoints were relative to the NTU60-XSub [74], NTU120-XSub [75], UCF101 [76], HMDB51 [77], and FineGYM [78] datasets (all trained using 2D skeleton sequences). Considering the actions featured in these datasets, HMDB51 is the one most closely related to PIP Retail. Thus, we selected that checkpoint for the purpose of fine-tuning PoseC3D.

Some details regarding the training process of each model (noting that fine-tuning requires fewer epochs and a lower initial learning rate), as well as results taken from the logs, can be found in Tables 5.7 and 5.8, respectively. Furthermore, Figures 5.10 and 5.11 depict the confusion matrices obtained from testing the fine-tuned ST-GCN and PoseC3D models, while Table 5.10 shows the *accuracy*, along with the macro-averaged values for *recall*, *precision*, and *F1-score*.

Table 5.7. Training process details (fine-tuning).

Model	Optimiser	Learning Rate	Momentum	Weight Decay	Batch Size	Max Epochs
ST-GCN	SGD	0.05 (Step Decay)	0.9	0.0001	16	40
PoseC3D		0.1 (Cosine Annealing)	0.9	0.0003	16	120

Table 5.8. Training results for the PIP Retail Small dataset (fine-tuning).

Model	Execution Time per Epoch (mm:ss.SSS)	Best Checkpoint	Validation Top-1 Accuracy	Validation Top-2 Accuracy
ST-GCN	01:41.225	40	0.7973	0.9119
PoseC3D	21:40.283	120	0.8727	0.9508

Compared to training from scratch, fine-tuning the ST-GCN and PoseC3D models proved to be an asset in improving results, considering the limited scale of the PIP Retail Small dataset. In both cases, the *accuracy* scores increased by around 3%. Upon closer examination of the confusion matrices, we noticed that ST-GCN improved its performance for most of the action classes, especially for those that showed higher confusion rates when trained from scratch, such as “person_purchases_from_cashier” “person_picks_object”, “person_places_object”, and “person_talks_to_person”. Another action class that noticeably benefited from fine-tuning was “person_interacts_with_phone”. These enhancements can be attributed to the presence of other classes with similar motions in the NTU60-XSub dataset. An identical outcome was observed for PoseC3D, with most action classes exhibiting lower confusion rates when compared to training from scratch. In particular, classes such as “person_sits_down” and “person_stands_up”, as well as “person_picks_object”, were considerably improved, since these are provided in the HMDB51 dataset. However, it should be noted that the HMDB51 dataset covers four body visibility levels (i.e. full body, upper body, lower body, and head), and the “talk” action class in the HMDB51 dataset was only recorded for the upper body and head visibility levels, rather than the target full body. This limitation explains why the performance of the “person_talks_to_person” action class did not improve.

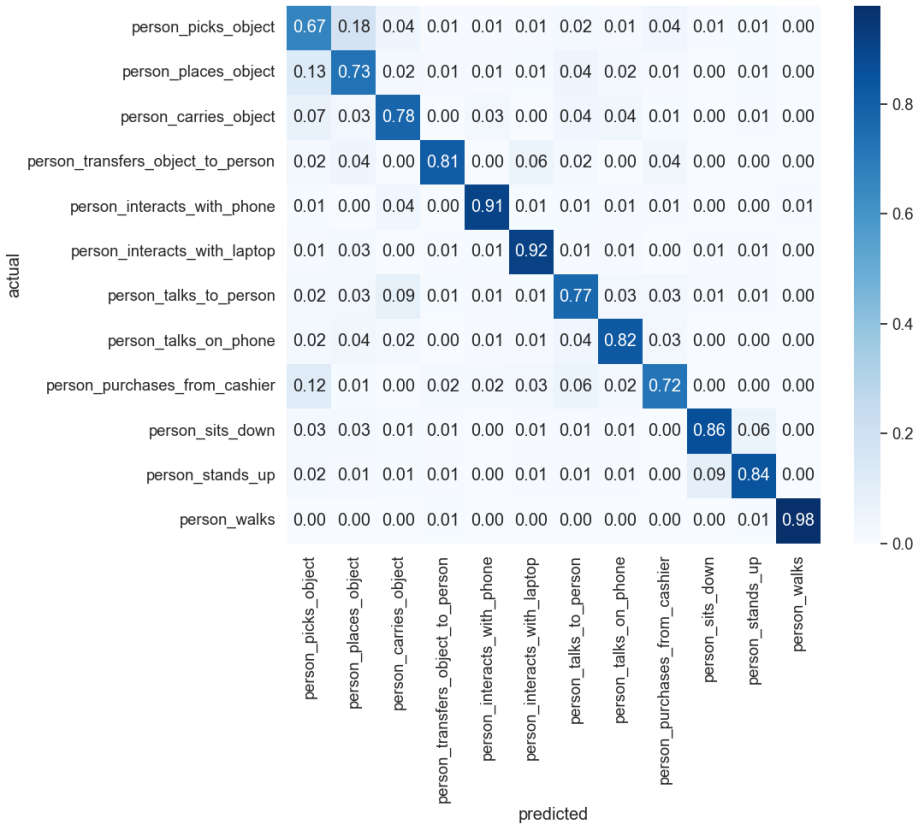


Figure 5.10. Confusion matrix of the fine-tuned ST-GCN model in the PIP Retail Small dataset.

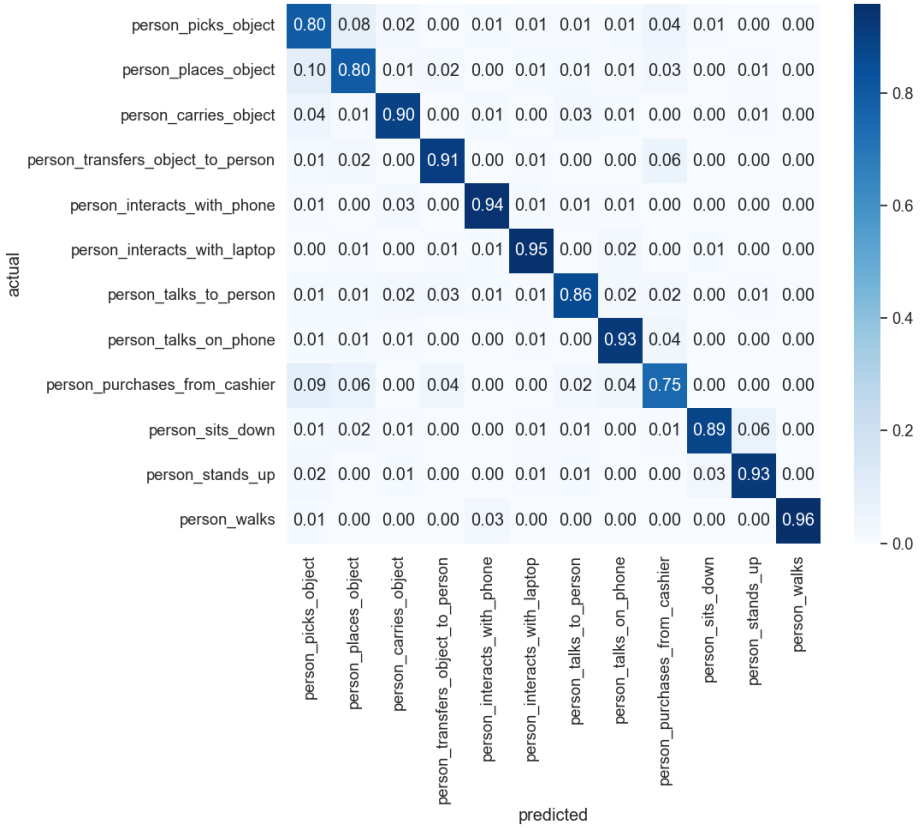


Figure 5.11. Confusion matrix of the fine-tuned PoseC3D model in the PIP Retail Small dataset.

Once the entire PIP Retail dataset had been assembled, we were able to train the models with it. The configurations adopted remained consistent with those used for training the models from scratch with the PIP Retail Small dataset, which are presented in Table 5.5. The statistics drawn from the logs during the training process are provided in Table 5.9. The confusion matrices generated in the testing process are displayed in Figures 5.12, 5.13, and 5.14. Furthermore, the performance scores calculated based on these matrices are shown in Table 5.10. However, since training with the PIP Retail dataset requires substantially more time and computational resources than using the PIP Retail Small dataset, we were not able to obtain results for PoseC3D as it is significantly more resource-intensive than both ST-GCN and AGCN.

Table 5.9. Training results for the PIP Retail dataset.

Model	**Execution Time per Epoch (mm:ss.SSS)**	**Best Checkpoint**	**Validation Top-1 Accuracy**	**Validation Top-2 Accuracy**
ST-GCN	16:35.075	55	0.8784	0.9553
AGCN	30:35.463	65	0.9127	0.9679

Upon analysing the results, we observed that training the ST-GCN and AGCN models with the PIP Retail dataset led to an increase in *accuracy* of approximately 11% to 12% compared to training them with the PIP Retail Small dataset. This indicates a substantial improvement in performance. However, it is important to note that the macro-averaged *F1-score* values increased by approximately 8% to 9%, which, while also a notable improvement, was less substantial compared to the *accuracy* gains. These outcomes emphasise the inherent imbalance of the complete dataset, given that the majority classes outperformed the overall class performance. Nevertheless, they indicate significant enhancements in the performance of the models when trained with the complete PIP Retail dataset, due to the greater number and variety of instances they were able to learn from.

Regarding the confusion matrices, we noticed that the results for practically all classes improved, with most of them exhibiting lower counts of false positives and false negatives. However, the action “person_purchases_from_cashier”, previously mentioned for being the combination of several other atomic actions of interactions between a customer and the cashier, was the only one that maintained its poor true positive rate (*recall*) or even worsened it. In particular, although the predictions for this specific class were mostly correct (few false positives), a significant number of instances were falsely predicted as the “person_transfers_object_to_person” atomic action (false negatives).

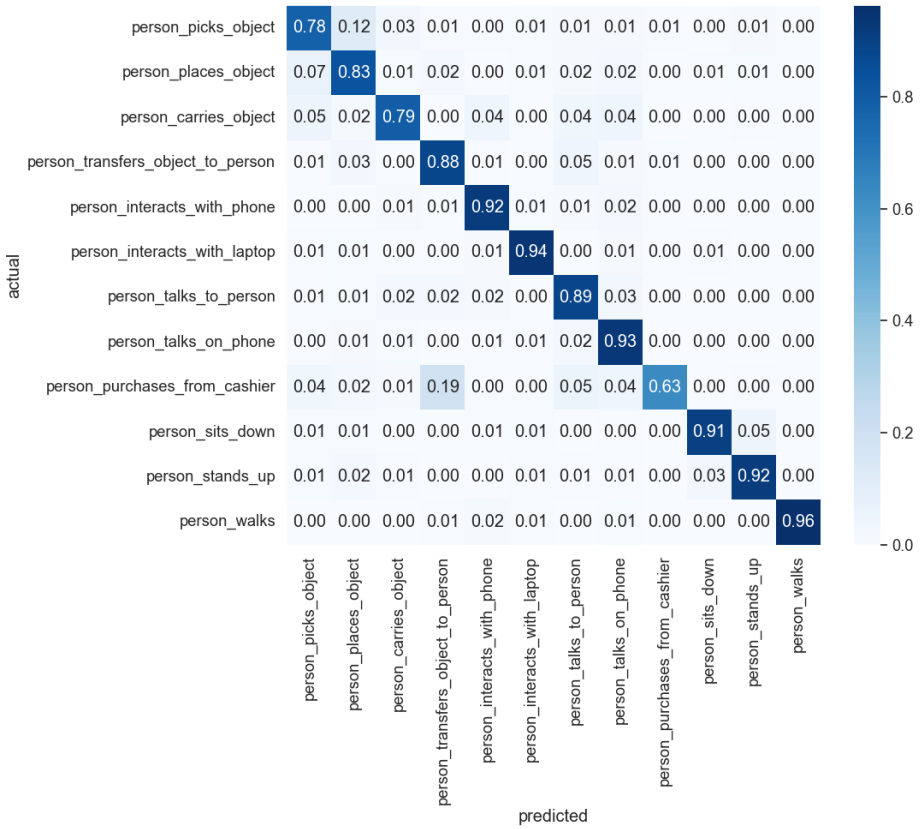


Figure 5.12. Confusion matrix of the ST-GCN model in the PIP Retail dataset.

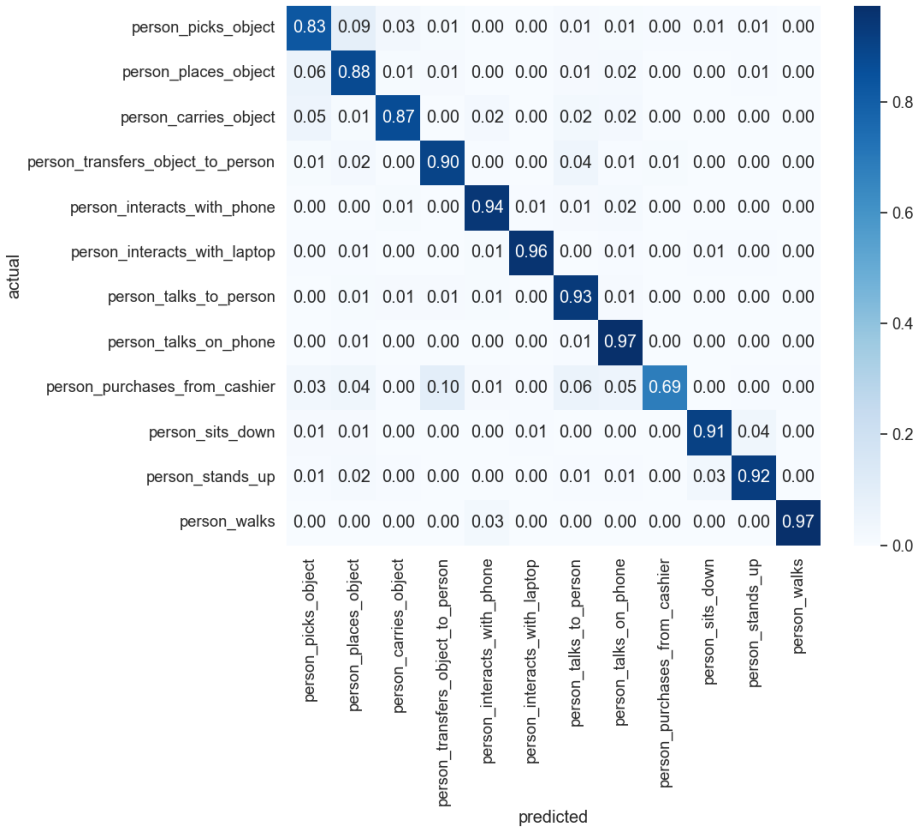


Figure 5.13. Confusion matrix of the AGCN model in the PIP Retail dataset.

Table 5.10. Overall test results.

Model	Dataset	Fine-tuning	Inference Time (ms)	Macro Recall	Macro Precision	Macro F1-Score	Accuracy
ST-GCN	PIP Retail Small	No	4	0.7850	0.7694	0.7754	0.7588
ST-GCN	PIP Retail Small	Yes (NTU60-XSub)	4	0.8183	0.8135	0.8150	0.7973
AGCN	PIP Retail Small	No	17	0.8192	0.8136	0.8154	0.8038
PoseC3D	PIP Retail Small	No	267	0.8556	0.8468	0.8503	0.8423
PoseC3D	PIP Retail Small	Yes (HMDB51)	267	0.8835	0.8758	0.8785	0.8727
ST-GCN	PIP Retail	No	4	0.8656	0.8655	0.8653	0.8784
AGCN	PIP Retail	No	17	0.8975	0.8992	0.8983	0.9127

5.2.3. Evaluation

Based on the results of our experiments in the Modelling phase, we can now discuss the performance of each model on a more practical level. Among the 3 models, ST-GCN showed the shortest inference times, highlighting its computational efficiency. However, it achieved lower scores for all performance metrics compared to the other models, even though it still managed to achieve a notable *accuracy* of 87.84% on the PIP Retail dataset. Based on these findings, we would recommend its use in scenarios where resource constraints are a factor, or in situations where the fastest inference time is imperative (e.g. real-time applications). On the other hand, AGCN showed a good balance between performance and computational efficiency, obtaining a 91.27% *accuracy* on the PIP Retail dataset, with moderately longer inference times than those observed for ST-GCN. With this in mind, it also offers a favourable choice for real-time applications, particularly when there are no considerable limitations in terms of computing power. Lastly, despite not being able to evaluate its performance on the PIP Retail dataset, PoseC3D achieved the best results in all metrics for the PIP Retail Small dataset, reaching an *accuracy* of 87.27% through fine-tuning. However, this improved performance came at the cost of substantially longer inference times. Therefore, while PoseC3D offers impressive results, it may be better suited for scenarios where computational resources are abundant and the priority is obtaining the best possible performance. For applications where real-time processing is essential, it may not be the best option.

In terms of dataset size, in general, all the models exhibited superior performance when trained with the PIP Retail dataset rather than with the PIP Retail Small dataset, owing to the greater number and variety of instances. Nonetheless, considering the limited scale of the PIP Retail Small dataset, it was still possible to achieve respectable results. Furthermore, fine-tuning the models (pre-trained on large-scale datasets) proved to be a valuable tool for improving results on small-scale datasets. These improvements may not be extremely significant, but they do offer practical advantages, as fine-tuned models leverage prior knowledge to produce more refined predictions.

As for class-specific aspects, among the 12 action classes included in the PIP Retail dataset, some consistently showed a higher level of confusion than the others. Notably, “person_picks_object” and “person_places_object” were often mistaken for each other due to the similarity in arm movements. To address this issue and enhance results, one potential approach would involve extracting additional appearance cues from the video frames so as to gain insights into the scene surrounding the subjects and any objects they may be holding, all while taking privacy into account. Noting that several other classes would benefit from this solution. Furthermore, the “person_purchases_from_cashier” action class was frequently confused with others that can be regarded as subclasses (i.e. atomic interactions between the subject and the cashier), which include “person_picks_object”, “person_places_object”, “person_transfers_object_to_person”, and “person_talks_to_person”. This issue suggests that it may be more beneficial to train the models exclusively for predicting individual atomic actions, which can then be further processed using a rule-based system to infer the actual complex actions.

In summary, the results obtained in our experiments provided valuable insights into the practical applications of the three skeleton-based action recognition models (i.e. ST-GCN, AGCN, and PoseC3D) for analysing customer behaviour in retail settings. Overall, the performance demonstrated by each of these models confirms their suitability for this use case, laying a strong foundation for future research in this area. Beyond their individual performance characteristics, it is essential to emphasise how the outputs of these models, together with trajectory-related data, can aid decision-making processes in real-world scenarios, contributing to an enhanced retail experience.

CHAPTER 6

Conclusions

In this study, a framework was developed to extract information regarding customer behaviour from high-resolution surveillance videos. This includes trajectory-related data that describes the location of customers over time, as well as pose-related data that specifies the actions being performed by them, both of which require reliable tracking data for the continuous identification of each customer.

For this purpose, we were provided with a dataset exclusively prepared for the "ECI 4.0 - Espaços Comerciais Inteligentes" research project, of which this dissertation is a part. However, due to certain problems inherent in this dataset, we found it necessary to search for alternative datasets. Hence, for experiments involving trajectory-related data, we chose the VIRAT dataset, and for action recognition, we opted for the PIP dataset.

During the development of the framework, we identified some issues and proposed solutions to mitigate them. One of them was the impact of occlusions on the location data provided by the object detection and tracking algorithms. To address this issue, we created the occlusion-aware mechanism, which aims to detect occlusion cases and rectify their impact. To evaluate its effectiveness, in addition to qualitative experiments, we conducted an in-depth statistical assessment of its performance using parametric and nonparametric tests: the paired Student's t-test and the Wilcoxon test, respectively. These tests were applied to IoU values calculated by comparing the ground truth bounding boxes of the VIRAT dataset with the outputs of the object detection algorithm, before and after being adjusted using the occlusion-aware mechanism. The results from both tests indicated that the improvements made by the mechanism were statistically significant (paired Student's t-test: $t(34) = 11.70$, $p < .001$; Wilcoxon test: $z = 5.16$, $p < .001$). Moreover, the paired Student's t-test yielded a Cohen's d value of 1.98, indicating a substantial effect size.

These results directly satisfy the first part of our first research question, which aimed to provide a viable solution for improving the accuracy of location data generated by object detection and tracking algorithms in occlusion-prone scenarios. With this in mind, we can conclude that the occlusion-aware mechanism is a valuable tool for enhancing location data in such scenarios.

Furthermore, we observed that the trajectory points, extracted from the location data, exhibited irregularities due to different factors. Therefore, we introduced a smoothing method to correct them, considering both past and future trajectory information. Based on the visual outputs produced using an 8-point sliding window, we were able to verify the effectiveness of the method in attenuating the irregularities. In particular, at an individual level, the method proved to be capable of handling natural

types of irregularities, such as those caused by gait (walking) movement. Besides, in conjunction with the occlusion-aware mechanism, it provided significant practical contributions in mitigating the effect of occlusions on trajectory mapping, resulting in more reliable path information.

Specifically, our research confirmed that applying the occlusion-aware mechanism to the location data of occluded subjects, and subsequently applying the smoothing method to the trajectory points extracted from the corrected data, notably enhances the precision of the resulting trajectories. This is because the smoothed trajectory points end up tracing the actual positions of the subjects over time, rather than being influenced by the obstacles that occluded them. These findings address the second aspect of our first research question, which sought to examine the practical implications of improving the location data on trajectory mapping, particularly in scenarios involving occlusions.

In terms of the action recognition process, we compared 3 skeleton-based HAR models: ST-GCN, AGCN, and PoseC3D. These were trained and tested on both the PIP Retail dataset, which comprises a subset of 12 classes selected from the original PIP dataset, and its scaled-down version, the PIP Retail Small dataset. The results obtained from assessing the models on these datasets suggest that ST-GCN is the most efficient, but has lower performance, making it ideal for resource-constrained or real-time applications, and that AGCN strikes a balance between efficiency and performance, thus rendering it a compelling choice for real-time applications with reasonable computational resources. Conversely, despite showing superior performance, PoseC3D proved to be considerably more resource-intensive than the others, meaning it is suitable for scenarios where the priority is to achieve the best possible performance, rather than real-time processing. Nevertheless, aside from their unique characteristics, the scores achieved by each of these models (reaching *accuracy* values in the order of 90% on the PIP Retail dataset) confirm their effectiveness in recognising customer behaviour, whilst ensuring privacy, given that only anonymous skeleton data is processed.

The aforementioned findings effectively address the second research question, which concerned the selection of suitable models to recognise customer behaviour while safeguarding their privacy, as well as the identification of the most adequate use cases for each model, based on their performance and computational efficiency.

6.1. Future Work

Regarding future research opportunities, an important aspect to consider is the acquisition of a more specialised dataset, consisting of high-resolution surveillance videos captured in retail environments, as was initially intended for this work with the Worten Surveillance dataset. Besides enabling a more adequate assessment of the framework as a whole, a more specialised dataset would be particularly useful for refining the action recognition process. This is because it would allow models to be trained,

or possibly even fine-tuned (e.g. those trained on the PIP Retail dataset in this study), to respond to a more specific use case, in order to rigorously evaluate their effectiveness in practical settings.

Still on the subject of action recognition, given that recognising actions with similar body motions proved to be challenging using skeleton data only, it would be interesting to explore the integration of supplementary visual features extracted from the video frames. This approach could provide valuable insights into the context surrounding the subjects, all while maintaining the requirements associated with their privacy. Furthermore, the contributions of the occlusion-aware mechanism can be further explored, specifically by analysing the extent to which its improvements to the location data influence the action recognition process in scenarios involving occlusions.

Another aspect that can also be explored is applying the occlusion-aware mechanism before the tracking process. This approach would make it possible to provide the multi-object tracking algorithm with the bounding boxes adjusted using the mechanism, instead of only those predicted by the object detection algorithm. This adaptation is expected to reduce the number of lost tracks occurring due to partial occlusions, improving the tracking process.

References

[1] N. M. Larsen, V. Sigurdsson, and J. Breivik, "The Use of Observational Technology to Study In-Store Behavior: Consumer Choice, Video Surveillance, and Retail Analytics", *Behavior Analyst*, vol. 40, no. 2, pp. 343–371, Nov. 2017, doi: 10.1007/s40614-017-0121-x.

[2] J. Liu, Y. Gu and S. Kamijo, "Customer Behavior Recognition in Retail Store from Surveillance Camera," in *2015 IEEE International Symposium on Multimedia (ISM)*, Miami, FL, USA, 2015, pp. 154–159, doi: 10.1109/ISM.2015.52.

[3] M. Popa *et al.*, "Analysis of shopping behavior based on surveillance system," in *2010 IEEE International Conference on Systems, Man and Cybernetics*, Istanbul, Turkey, 2010, pp. 2512–2519, doi: 10.1109/ICSMC.2010.5641928.

[4] K. Nguyen, M. Le, B. Martin, I. Cil, and C. Fookes, "When AI meets store layout design: a review", *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5707–5729, Oct. 2022, doi: 10.1007/s10462-022-10142-3.

[5] M. Ahmed *et al.*, "Real-Time Violent Action Recognition Using Key Frames Extraction and Deep Learning", *Computers, Materials & Continua*, vol. 69, no. 2, pp. 2217–2230, 2021, doi: 10.32604/cmc.2021.018103

[6] L. Kirichenko, T. Radivilova, B. Sydorenko, and S. Yakovlev, "Detection of Shoplifting on Video Using a Hybrid Network", *Computation*, vol. 10, no. 11, 2022, doi: 10.3390/computation10110199.

[7] Y. Kong and Y. Fu, "Human Action Recognition and Prediction: A Survey", *International Journal of Computer Vision*, vol. 130, no. 5, pp. 1366–1401, May 2022, doi: 10.1007/s11263-022-01594-9.

[8] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang and J. Liu, "Human Action Recognition From Various Data Modalities: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3200–3225, March 2023, doi: 10.1109/TPAMI.2022.3183112.

[9] "Recital 26," GDPR Text, 2023. [Online]. Available: https://gdpr-text.com/read/recital-26/

[10] J. Sanchez, C. Neff and H. Tabkhi, "Real-World Graph Convolution Networks (RW-GCNs) for Action Recognition in Smart Video Surveillance," in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, San Jose, CA, USA, 2021, pp. 121–134, doi: 10.1145/3453142.3491293.

[11] "ECI4.0 - Espaços Comerciais Inteligentes," Ciência-IUL, 2023. [Online]. Available: https://ciencia.iscte-iul.pt/projects/espacos-comerciais-inteligentes/1736

[12] R. Wirth and J. Hipp, "CRISP-DM: Towards a Standard Process Model for Data Mining", in *Proceedings of the 4th International Conference on the Practical Application of Knowledge Discovery and Data Mining*, 2000, vol. 1, pp. 29–39.

[13] M. J. Page *et al.*, "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews", *BMJ*, vol. 372, p. n71, Mar. 2021, doi: 10.1136/bmj.n71.

[14] J.-K. Tsai, C.-C. Hsu, W.-Y. Wang, and S.-K. Huang, "Deep Learning-Based Real-Time Multiple-Person Action Recognition System", *Sensors*, vol. 20, no. 17, 2020, doi: 10.3390/s20174758.

[15] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[16] W. Liu *et al.*, “SSD: Single Shot MultiBox Detector,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[17] R. Girshick, J. Donahue, T. Darrell and J. Malik, “Region-Based Convolutional Networks for Accurate Object Detection and Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, Jan. 2016, doi: 10.1109/TPAMI.2015.2437384.

[18] R. Girshick, “Fast R-CNN,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[19] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2017, doi: 10.1109/TPAMI.2016.2577031.

[20] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, “Simple online and realtime tracking,” in *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, 2016, pp. 3464–3468, doi: 10.1109/ICIP.2016.7533003.

[21] N. Wojke, A. Bewley and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *2017 IEEE International Conference on Image Processing (ICIP)*, Beijing, China, 2017, pp. 3645–3649, doi: 10.1109/ICIP.2017.8296962.

[22] Y. Zhang *et al.*, “ByteTrack: Multi-object Tracking by Associating Every Detection Box”, in *Lecture Notes in Computer Science*, Cham: Springer Nature Switzerland, 2022, pp. 1–21, doi: 10.1007/978-3-031-20047-2_1.

[23] H. Duan, Y. Zhao, K. Chen, D. Lin and B. Dai, “Revisiting Skeleton-based Action Recognition,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 2022, pp. 2959–2968, doi: 10.1109/CVPR52688.2022.00298.

[24] Z. Cao, T. Simon, S. -E. Wei and Y. Sheikh, “Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 1302–1310, doi: 10.1109/CVPR.2017.143.

[25] K. Sun, B. Xiao, D. Liu and J. Wang, “Deep High-Resolution Representation Learning for Human Pose Estimation,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 5686–5696, doi: 10.1109/CVPR.2019.00584.

[26] M. A. Khan, M. Sharif, T. Akram, M. Raza, T. Saba, and A. Rehman, “Hand-crafted and deep convolutional neural network features fusion and selection strategy: An application to intelligent human action recognition”, *Applied Soft Computing*, vol. 87, p. 105986, 2020, doi: 10.1016/j.asoc.2019.105986.

[27] K. Simonyan and A. Zisserman, “Two-Stream Convolutional Networks for Action Recognition in Videos”, *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[28] S. Ji, W. Xu, M. Yang and K. Yu, “3D Convolutional Neural Networks for Human Action Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, Jan. 2013, doi: 10.1109/TPAMI.2012.59.

[29] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, “Learning Spatiotemporal Features with 3D Convolutional Networks,” *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 4489–4497, doi: 10.1109/ICCV.2015.510.

[30] J. Carreira and A. Zisserman, “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 4724–4733, doi: 10.1109/CVPR.2017.502.

[31] J. Arunnehru, G. Chamundeeswari, and S. P. Bharathi, “Human Action Recognition using 3D Convolutional Neural Networks with 3D Motion Cuboids in Surveillance Videos”, *Procedia Computer Science*, vol. 133, pp. 471–477, 2018, doi: 10.1016/j.procs.2018.07.059.

[32] G. Yao, T. Lei, J. Zhong, and P. Jiang, “Learning multi-temporal-scale deep information for action recognition”, *Appl. Intell.*, vol. 49, no. 6, pp. 2017–2029, Jun. 2019, doi: 10.1007/s10489-018-1347-3.

[33] Z. Gao, L. Guo, W. Guan, A. -A. Liu, T. Ren and S. Chen, “A Pairwise Attentive Adversarial Spatiotemporal Network for Cross-Domain Few-Shot Action Recognition-R2,” *IEEE Transactions on Image Processing*, vol. 30, pp. 767–782, 2021, doi: 10.1109/TIP.2020.3038372.

[34] C. Liu, J. Ying, H. Yang, X. Hu, and J. Liu, “Improved human action recognition approach based on two-stream convolutional neural network model”, *The Visual Computer*, vol. 37, no. 6, pp. 1327–1341, Jun. 2021, doi: 10.1007/s00371-020-01868-8.

[35] M. Liu, H. Liu, and C. Chen, “Enhanced skeleton visualization for view invariant human action recognition”, *Pattern Recognition*, vol. 68, pp. 346–362, Aug. 2017, doi: 10.1016/j.patcog.2017.02.030.

[36] Y. Li, R. Xia, and X. Liu, “Learning shape and motion representations for view invariant skeleton-based action recognition”, *Pattern Recognition*, vol. 103, p. 107293, Jul. 2020, doi: 10.1016/j.patcog.2020.107293.

[37] P. Dasari, L. Zhang, Y. Yu, H. Huang and R. Gao, “Human Action Recognition Using Hybrid Deep Evolving Neural Networks,” in *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1–8, doi: 10.1109/IJCNN55064.2022.9892025.

[38] L. Zhang, C. P. Lim, and Y. Yu, “Intelligent human action recognition using an ensemble model of evolving deep networks with swarm-based optimization”, *Knowledge-Based Systems*, vol. 220, p. 106918, May 2021, doi: 10.1016/j.knosys.2021.106918.

[39] R. Zhao, H. Ali and P. van der Smagt, “Two-stream RNN/CNN for action recognition in 3D videos,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017, pp. 4260–4267, doi: 10.1109/IROS.2017.8206288.

[40] G. Liu, J. Qian, F. Wen, X. Zhu, R. Ying and P. Liu, “Action Recognition Based on 3D Skeleton and RGB Frame Fusion,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, 2019, pp. 258–264, doi: 10.1109/IROS40897.2019.8967570.

[41] D. Avola, M. Cascio, L. Cinque, G. L. Foresti, C. Massaroni and E. Rodolà, “2-D Skeleton-Based Action Recognition via Two-Branch Stacked LSTM-RNNs,” *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2481–2496, Oct. 2020, doi: 10.1109/TMM.2019.2960588.

[42] S. Yan, Y. Xiong, and D. Lin, “Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition”, *AAAI*, vol. 32, no. 1, Apr. 2018, doi: 10.1609/aaai.v32i1.12328.

[43] L. Shi, Y. Zhang, J. Cheng and H. Lu, “Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 12018-12027, doi: 10.1109/CVPR.2019.01230.

[44] D. Tian, Z.-M. Lu, X. Chen, and L.-H. Ma, “An attentional spatial temporal graph convolutional network with co-occurrence feature learning for action recognition”, *Multimed. Tools Appl.*, vol. 79, no. 17–18, pp. 12679–12697, May 2020, doi: 10.1007/s11042-020-08611-4.

[45] A. Ullah, K. Muhammad, I. U. Haq, and S. W. Baik, “Action recognition using optimized deep autoencoder and CNN for surveillance data streams of non-stationary environments”, *Future Gener. Comput. Syst.*, vol. 96, pp. 386–397, Jul. 2019, doi: 10.1016/j.future.2019.01.029.

[46] X. Li, S. Onie, M. Liang, M. Larsen, and A. Sowmya, “Towards Building a Visual Behaviour Analysis Pipeline for Suicide Detection and Prevention”, *Sensors*, vol. 22, no. 12, 2022, doi: 10.3390/s22124488.

[47] M. Dallel, V. Havard, Y. Dupuis, and D. Baudry, “A Sliding Window Based Approach With Majority Voting for Online Human Action Recognition Using Spatial Temporal Graph Convolutional Neural Networks”, in *2022 7th International Conference on Machine Learning Technologies (ICMLT)*, Rome, Italy, 2022, pp. 155–163, doi: 10.1145/3529399.3529425.

[48] S. Kim, K. Yun, J. Park and J. Y. Choi, “Skeleton-Based Action Recognition of People Handling Objects,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, 2019, pp. 61–70, doi: 10.1109/WACV.2019.00014.

[49] C.-B. Jin, T. D. Do, M. Liu, and H. Kim, “Real-Time Action Detection in Video Surveillance using a Sub-Action Descriptor with Multi-Convolutional Neural Networks”, *Journal of Institute of Control, Robotics and Systems*, vol. 24, no. 3, pp. 298–308, Mar. 2018, doi: 10.5302/J.ICROS.2018.17.0243.

[50] G. Jocher *et al.*, “ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference,” Zenodo, 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6222936

[51] “Pytorch hub,” PyTorch Foundation, 2023. [Online]. Available: https://pytorch.org/hub/

[52] T.-Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014, pp. 740-755, doi: 10.1007/978-3-319-10602-1_48.

[53] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME-Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960, doi: 10.1115/1.3662552.

[54] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955, doi: 10.1002/nav.3800020109.

[55] Y. Zhang *et al.*, “ByteTrack: Multi-object tracking by associating every detection box,” GitHub, 2023. [Online]. Available: https://github.com/ifzhang/ByteTrack

[56] E. Dubrofsky, “Homography estimation,” *Diplomová práce. Vancouver: Univerzita Britské Kolumbie*, vol. 5, 2009.

[57] MMPose Contributors, “OpenMMLab Pose Estimation Toolbox and Benchmark v0.29.0,” GitHub, 2022. [Online]. Available: https://github.com/open-mmlab/mmpose/tree/v0.29.0

[58] “OpenMMLab,” OpenMMLab, 2020. [Online]. Available: https://openmmlab.com/

[59] MMAction2 Contributors, “OpenMMLab Video Understanding Toolbox and Benchmark v0.24.1,” GitHub, 2022. [Online]. Available: https://github.com/open-mmlab/mmaction2/tree/v0.24.1

[60] S. Oh *et al.*, “A large-scale benchmark dataset for event recognition in surveillance video,” in *CVPR 2011*, Colorado Springs, CO, USA, 2011, pp. 3153–3160, doi: 10.1109/CVPR.2011.5995586.

[61] K. Corona, K. Osterdahl, R. Collins and A. Hoogs, “MEVA: A Large-Scale Multiview, Multimodal Video Dataset for Activity Detection,” in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, 2021, pp. 1059–1067, doi: 10.1109/WACV48630.2021.00110.

[62] “ActEV: Activities in Extended Video,” NIST, 2023. [Online]. Available: https://actev.nist.gov/

[63] “People in Public 370k Stabilized,” Visym Labs, 2023. [Online]. Available: https://visym.github.io/collector/pip_370k_stabilized/

[64] Visym Labs, “Visym Collector: On-demand Datasets for Visual AI,” GitHub, 2023. [Online]. Available: https://github.com/visym/collector

[65] Visym Labs, “VIPY: Python Tools for Visual Dataset Transformation,” GitHub, 2023. [Online]. Available: https://github.com/visym/vipy

[66] X. Han *et al.*, “MMPTRACK: Large-scale Densely Annotated Multi-camera Multiple People Tracking Benchmark,” in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, 2023, pp. 4849-4858, doi: 10.1109/WACV56688.2023.00484.

[67] B. Singh, T. K. Marks, M. Jones, O. Tuzel and M. Shao, “A Multi-stream Bi-directional Recurrent Neural Network for Fine-Grained Action Detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 1961–1970, doi: 10.1109/CVPR.2016.216.

[68] W. Sultani, C. Chen and M. Shah, “Real-World Anomaly Detection in Surveillance Videos,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 6479–6488, doi: 10.1109/CVPR.2018.00678.

[69] Student, “The Probable Error of a Mean”, *Biometrika*, vol. 6, no. 1, pp. 1–25, 1908, doi: 10.2307/2331554.

[70] F. Wilcoxon, “Individual Comparisons by Ranking Methods”, *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945, doi: 10.2307/3001968.

[71] S. S. Shapiro and M. B. Wilk, “An Analysis of Variance Test for Normality (Complete Samples)”, *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965, doi: 10.2307/2333709.

[72] S. G. Kwak and J. H. Kim, “Central limit theorem: the cornerstone of modern statistics”, *Korean J. Anesthesiol.*, vol. 70, no. 2, pp. 144–156, 2017, doi: 10.4097/kjae.2017.70.2.144.

[73] J. Cohen, *Statistical power analysis for the behavioral sciences*, 2nd ed. New York, NY: Routledge, 1988, doi: 10.4324/9780203771587.

[74] A. Shahroudy, J. Liu, T. -T. Ng and G. Wang, “NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 1010–1019, doi: 10.1109/CVPR.2016.115.

[75] J. Liu, A. Shahroudy, M. Perez, G. Wang, L. -Y. Duan and A. C. Kot, “NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2684–2701, 2020, doi: 10.1109/TPAMI.2019.2916873.

[76] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild”, *arXiv:1212.0402*, 2012.

[77] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio and T. Serre, “HMDB: A large video database for human motion recognition,” in *2011 International Conference on Computer Vision*, Barcelona, Spain, 2011, pp. 2556–2563, doi: 10.1109/ICCV.2011.6126543.

[78] D. Shao, Y. Zhao, B. Dai and D. Lin, “FineGym: A Hierarchical Video Dataset for Fine-Grained Action Understanding,” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, pp. 2613-2622, doi: 10.1109/CVPR42600.2020.00269.