A COMPUTER VISION SYSTEM FOR DETECTING AND ANALYSING CRITICAL EVENTS IN CITIES

Doctoral dissertation

Mohamed R. Ibrahim

SpaceTimeLab

Department of Civil, Environmental and Geomatic Engineering

University College London (UCL)

JULY 2021

Supervisors
Dr James Haworth
Prof Tao Cheng

STATEMENT OF AUTHENTICITY

This dissertation contains no material which has been accepted for an award of any other degree or diploma in any institution and to the best of my knowledge, the research contains no material previously published or written by another person, except where due references have been made in the text of the thesis.

Mohamed R. Ibrahim
London, 1st JUL 2021

ABSTRACT

Whether for commuting or leisure, cycling is a growing transport mode in many cities worldwide. However, it is still perceived as a dangerous activity. Although serious incidents related to cycling leading to major injuries are rare, the fear of getting hit or falling hinders the expansion of cycling as a major transport mode. Indeed, it has been shown that focusing on serious injuries only touches the tip of the iceberg. Near miss data can provide much more information about potential problems and how to avoid risky situations that may lead to serious incidents. Unfortunately, there is a gap in the knowledge in identifying and analysing near misses. This hinders drawing statistically significant conclusions to provide measures for the built-environment that ensure a safer environment for people on bikes. In this research, we develop a method to detect and analyse near misses and their risk factors using artificial intelligence. This is accomplished by analysing video streams linked to near miss incidents within a novel framework relying on deep learning and computer vision. This framework automatically detects near misses and extracts their risk factors from video streams before analysing their statistical significance. It also provides practical solutions implemented in a camera with embedded AI (URBAN-i Box) and a cloud-based service (URBAN-i Cloud) to tackle the stated issue in the real-world settings for use by researchers, policy-makers, or citizens. The research aims to provide human-centred evidence that may enable policy-makers and planners to provide a safer built environment for cycling in London, or elsewhere. More broadly, this research aims to contribute to the scientific literature with the theoretical and empirical foundations of a computer vision system that can be utilised for detecting and analysing other critical events in a complex environment. Such a system can be applied to a wide range of events, such as traffic incidents, crime or overcrowding.

KEYWORDS: Cities, critical events, artificial intelligence, deep learning, computer vision, cycling near misses

STATEMENT OF IMPACT

In this research, we introduce a computer vision system that is able to assist city planners and policy-makers to detect and analyse cycling near misses and their risk factors. Based on artificial intelligence, the tool automatically analyses cycling near misses from video streams by understanding the interactions between people, the built and natural environment and different transport modes. The research will have several benefits in terms of improving road safety. This knowledge of risk factors will enable:

- Individuals (cyclists and other road users) to change their behaviour to minimise risk.
- Transport authorities to plan safer infrastructure and run informed awareness campaigns.
- The production of more accurate risk maps, showing which routes are safest for cycling, and what types of incidents to be wary of.

The application of the research output can be operationalised in the form of a cloud-based service (URBAN-i Cloud) or as a camera with embedded AI (URBAN-i Box) that can be used as an edge computing device to analyse cities from images/videos for better understanding cycling near misses.

ACKNOWLEDGMENT

I would like to thank Dr James Haworth, my first supervisor, who always offered his time and effort to assist me. This research would not be possible without his advice. I would like also to thank Prof Tao Cheng for her time and effort to supervise me when I needed help and advice. A special thanks goes to Prof Nicola Christie for her continuous support and help that also made this research possible.

A special thanks goes to my family (mother, father, and my two sisters) who unconditionally support and believe in me. I dedicate my work to them and to my late brother and grandmother who influenced me in many ways and their absence leaves a gap in my life. Also, a special thanks goes to my fiancé for her continuous support during my long hours of work.

I would also like to thank my friends and colleagues who supported me during my research time through cooperative work, advice, or even chatting.

Last, this research would not also be possible without the funds I received. Thanks are due to UCL Overseas Research Scholarship, Road Safety Trust Studentship, and Nvidia company for the GPU grant.

ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Networks
CNN	Convolutional Neural Networks
DRL	Deep Reinforcement Learning
GANs	Generative Adversarial Neural Networks
GQN	Generative Query Network
R-CNN	Region based Convolutional Neural Network
RNN	Recurrent Neural Network
SI	Framework Stringency Index
SSD	Single Shot Mulitbox detector- - A model for object detection
SORT	Simple Online and Realtime Tracking Method
YOLO	You Only Look Once- A model for object detection
URBAN-i	It refers to the introduced overall methodology of this research.
URBAN-i Box	It refers to the introduced camera implementing research methodology.
URBAN-i Cloud	It refers to the cloud-based service implementing research methodology.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1
INTRODUCTION

1.1 Overview

Understanding the dynamics of cities is a difficult task due to their complex and rapidly evolving nature (Batty, 2008; Bettencourt, 2013; Bettencourt and West, 2010). In particular, understanding how the various components of cities interact to cause critical events such as traffic incidents, fires, or overcrowding is challenging. The heterogeneous nature of these issues, which often occur at the intersection of different city systems, makes the development of integrated and transferrable urban modelling methods difficult. Therefore, developing reliable frameworks for urban modelling tasks that make use of consistent, widely available input data is a priority in order to create a more holistic representation of the city. In recent years, new types of urban data sources have emerged that have the potential to address this task, including satellite imagery, volunteered geographic information and street-level images (Arribas-Bel, 2014). There is great potential in using such data to analyse urban scenes to facilitate decision making.

Globally, all governments have a responsibility to protect their citizens and provide a safe environment. However, it remains a challenge to ensure safety in cities due to the many and varied risks that people face daily, from exposure to traffic incidents to crimes and terrorism. In general, recognising and modelling unsafe behaviour is complex because it requires understanding how humans interact with one another in the urban environment to produce risk. Linked to this, designing a safe built environment is a challenge because it requires understanding how people are likely to interact with infrastructure before it is built, balancing safety and performance. Behaviour depends on a multitude of factors, both tangible (e.g. crowdedness, weather, group dynamics) and intangible (e.g. personality, mood, cultural factors). If the individual and collective behaviour is not fully understood when policy decisions are made then there can be serious unintended consequences.

Understanding the dynamics of cities is essential for detecting and analysing critical events in complex scenes. While our knowledge of the dynamics of cities is still limited, substantial urban models have been achieved by perceiving cities as complex systems. Various urban scholars have attempted to model cities through cellular automata, fractals, or multi-agent models based on complexity science and network theory (Batty, 1976, 1997, 2009; Batty, Couclelis, & Eichen, 1997; Michael Batty, Xie, & Sun, 1999b; Batty & Torrens, 2005; Bretagnolle, Daudé, & Pumain, 2006; W. Zhou

& Li, 2013). However, these models, in many cases, either tend to over-simplify the initial settings of urban systems or explore cities from a mono-dimensional perspective (Batty and Torrens, 2001). Until recently, the main limiting factor for these models has been a lack of available data and computing power to feed large scale simulations. However, this is now changing with the emergence of new forms of data and urban analytical techniques drawn from the fields of artificial intelligence and machine learning.

Building our knowledge about cities through images is not a new concept. In fact, Lynch (1960) introduced how to perceive and understand cities from features– landmarks, focal points, skyline, pedestrian flow etc.– that anyone can recognize and understand, which was a very effective approach in perceiving the nuances of the urban world. Lynch's ideas concerning a city's features, in their original forms, may not cope with today's rapid urban challenges. However, with the growth of the fields of deep learning and computer vision, understanding cities through the eyes of a computer opens the door for analysing the missing attributes of city dynamics. Large-scale analysis of digital images and patterns of captured features that may not be recognized of significance by human eyes can potentially enable various urban issues to be tackled. Such techniques can track information and extract elements from images in a similar way to how urban scholars used to perceive cities.

One domain that stands to benefit from the use of large-scale image datasets is transport and urban mobility. Transport networks are a crucial element of cities, and traffic congestion and incidents place a significant burden on society. For this reason, urban transport networks have been monitored for many years through various means such as CCTV for automatic number plate recognition, loop detectors for flow measurement and signal timing adjustment, and GNSS for route analysis (Tarigan et al., 2017). These data can facilitate monitoring and forecasting network parameters. However, they fall short of being able to reveal deep insights into how the behaviour of agents (pedestrians, vehicles, bicycles) within the environment lead to undesirable consequences such as traffic congestion or incidents. Such insights are particularly important for vulnerable transport modes, such as cycling, for which large scale quantitative data is typically not collected.

Cycling has increased in popularity in Europe and elsewhere (Dozza et al., 2017). Whether for leisure or commuting, its benefits in terms of public health and the reduction of environmental pollution have influenced planners and policy-makers to invest in cycling infrastructure (de Hartog et al., 2010; Juhra et al., 2012; Pucher et al., 2010; Steinbach et al., 2011). Globally, various policies, programmes and physical and non-physical interventions have been implemented to promote cycling (Pucher et al., 2010; Savan et al., 2017). In the UK, for instance, Transport for London (TfL) has invested in many cycling infrastructure projects such as Cycle Superhighways, Quietways, Mini-Hollands and cycle hire schemes aimed at promoting a safer environment for people on bikes (TfL, 2018). However, generating evidence on the effectiveness of such schemes remains a challenge due to the sparsity of data on usage and incidents involving bicycles as a result of its low mode share. This masks the fact that even though the health benefit of cycling exceeds its risk (de Hartog et al., 2010), the risk remains high: *"Compared with car occupants and with regard to time spent traveling, cyclists were 8 times more likely to be injured, 12 to be hospitalized, 16 to be seriously injured, and 3 to be killed"* (Blaizot et al., 2013, p. 43). Furthermore, the experience of near misses, where a person on a bike was destabilised or had to take action to avoid a crash, can also add to the perception that cycling is dangerous. In the UK, people on bikes are likely to face at least one near miss for every six miles of a commute, according to Aldred and Crosweller (2015). This fear of getting hit or falling whilst cycling

hinders the wider adoption of cycling as a transport mode (Aldred, 2016; De Rome et al., 2014; Winters and Branion-Calles, 2017). However, due to their reported frequency, if data of near misses can be recorded, then they potentially provide a rich source of information for studying cyclists' accident risk and to identify the factors that are associated with them. A promising source of such data is naturalistic data, such as that collected from action cameras often used by commuting cyclists. Video data, in particular, provide an opportunity to study the scene of near-misses to extract the range of factors related to them that may or may not be transport-related (Aldred, 2016; Beck et al., 2016; De Rome et al., 2014; Imprialou and Quddus, 2017; Teschke et al., 2014). These factors can be related to aspects such as visibility, physical conditions of the built environment, interaction among different agents, or behavioural and psychological factors related to the cyclist. Put together, cycling near misses can be seen as an urban system that occurs in cities based on different factors and events that may or may not be directly related to transportation. Understanding the different urban systems of cities and their dynamics is a crucial step for understanding and analysing cycling near misses and critical events in general.

In general, it is challenging to quantitatively analyse the risk of cycling due to the low number of recorded incidents (Aldred, 2018; De Rome et al., 2014), or the impact of the reporting bias in road crash data, in which the less severe the crash, the higher the probability of under-reporting it (Abay, 2015). On the other hand, although many incidents may not result in a hospital visit or being reported to the police, people on bikes still report frequent situations where they need to take direct action to avoid a collision or feel destabilised. Cycling crashes are initiated by near miss situations that are not avoided and therefore result in a crash. By using this analogy, if data on these near misses can be recorded, then they can provide a rich source of information with which to study cyclists' crash risk and identify the factors that are most associated with them.

Cycling near miss is a transport-related subject. However, the factors related to near misses may or may not be transport-related (Aldred, 2016; Beck et al., 2016; De Rome et al., 2014; Imprialou and Quddus, 2017; Teschke et al., 2014), which requires understanding the whole picture. These factors can be related to aspects such as visibility, physical conditions of the built-up areas, interaction among different agents, or behavioural and psychological factors related to the cyclist. Put together, cycling near misses can be seen as an urban system that occurs in cities based on different factors and events that may or may not be directly related to transportation. Understanding the different urban systems of cities and their dynamics is a crucial step for understanding and analysing cycling near misses.

1.2 Research questions

The research aims to answer one crucial question: *How can we tackle different scenarios of interaction among multi-agents to predict critical events in a complex environment, bearing in mind the conditions and the dynamics of the built and natural environments?*

Due to the interdisciplinary nature of the risk factors related to near miss events- that include visibility, physical conditions of the built environment, the interaction of the different transport modes and cycling/driving behaviour- only looking from the perspective of transportation may not be sufficient to address and the research question. While the risk factors related to near misses are diverse, the complexity is not only in tackling the causality between each factor and the near miss incidents, but rather the combination of the different factors that are more likely to cause near misses. Therefore, different methods are required to tackle the different nature of the factors.

To address such a wide scope question, we divide the problem into sub-questions, which are:

1. How can we identify and predict urban systems that may influence near misses in cities?
2. To what extent can computer vision be used to understand the nuances of urban components from images/ videos?
3. To what extent can computer vision detect environmental conditions and visibility related factors from urban scenes?
4. To what extent can computer vision recognise a safe or a near miss scene from the overall interactions of road users in complex scenes?
5. When and where do cycling near-misses take place in cities?
6. Which factors are more likely to cause cycling near misses?

It is crucial to find methods that respond to each question, whereas can be pipelined to contribute to the wider perspective of the PhD research for understanding which scene belongs to a critical event besides extracting risk factors.

1.3 Research scope and objectives

This thesis focuses on the case of cycling near misses, as an example of tackling critical events, in unprecedented detail to develop an indicator for safety. Importantly, such an indicator will be based on events that are frequent rather than rare and will, therefore, offer the policymaker an opportunity to evaluate interventions. Also, the goal of this research is to 1) develop a method to record near miss experiences using artificial intelligence by analysing video data streams linked to near miss incidents and 2) identify infrastructural elements which offer the greatest level of safety for the greatest number of cyclists and identify risk factors associated with the interaction of a specific type of cyclist and the type of environment in which they are travelling.

On the other hand, it introduces a novel multi-purpose method that offers a realistic framework environment for understanding the dynamics of cities that contribute to understanding instant actions and critical events in cities, which can be applied to different domains. The method aims to map some of the agents and events in cities at a given time and space with respect to their behavioural complexity and without any simplification. The goal of this method is to extract and geo-reference information from unlabelled urban scene images or video streams that can act as an urban sensor. This will offer urban modellers a realistic platform for urban simulation for tackling the dynamics of various urban issues.

The overall research objectives are:

1. Detecting agents and their actions: Object detection methods will be used to detect humans and the transportation modes that they use,
2. Sensing the environment: All aspects of the physical environment related to the various layers of the city will be sensed using computer vision methods,
3. Recognising unsafe interactions of agents in a complex environment,
4. Highlighting the causes and effects of the different risk factors on the detected critical events.

The research will have several benefits in terms of improving road safety. This knowledge of risk factors will enable:

- Individuals (cyclists and other road users) to change their behaviour to minimise risk.
- Transport authorities to plan safer infrastructure and run informed awareness campaigns.
- The production of more accurate risk maps, showing which routes are safest for cycling and what types of incidents to be wary of.

The proposed methods will have long term impacts by building a database of near-miss incidents from which continued analysis of risk factors can take place. On the other hand, this research will exemplify the application of computer vision and deep learning in understanding risky situations in urban areas beyond the concept of smart cities that are discussed and perceived widely in the mainstream.

1.4 Research overall methodology

The research aims to develop a computer vision framework for analysing video data streams linked to near miss incidents to identify the risk factors associated with cycling near misses. The research introduces a framework for sensing, quantifying, analysing and understanding the environmental, behavioural and interactional factors associated with risky events in cities in the case of cycling near misses. The developed AI technologies, relying on various computer vision methods, can process and analyse a large-scale of heterogeneous images and video streams that are widely generated in cities through different sub-methods, which are explained in detail in the overall methodology chapter (Chapter IV) to fulfil the aforementioned research objectives. Typically, deep learning models, most specifically Convolutional Neural Networks (CNN), have shown substantial progress in classifying images of a wide spectrum of classes (LeCun et al., 2015). Various deep CNN models with different architectures and hyper-parameters have been computed to recognize objects in large image repositories, such as the ImageNET dataset that contains 14,197,122 images that belong to 22,000 different classes (Russakovsky et al., 2015). In order to address the multi-faced nature of the aforementioned questions, different types of quantitative methods are conducted. In general, the methods are selected based on two main reasons, first, in addressing the individual sub-question while contributing to the wider perspective of the research topic. Therefore, finding methods that can be implemented as pipelines where they can interact with each other in a framework is essential to carry out this research.

1.5 Research ethical approval

Ethical approval for data collection from individuals has been sought to collect data for analysis in this research.

1.6 Research structure

In **Fig. 1.1**, we show the research structure by introducing the stated topic, its motivation, research questions, and general methods in chapter 1. In chapter 2, we review computer vision in understanding the different aspects and complexity of cities. It explores the different algorithms related to computer vision and their application to date in understanding cities and the related urban systems that are applicable for understanding critical events such as near misses. In chapter 3, we review research related to near misses and the different methods used, highlighting the knowledge gap in the current methods for tackling near misses. We link chapter 2 and chapter 3 by highlighting

how computer vision can be utilised and an embedded AI system can be built to understand near misses. Chapter 4 addresses the overall research methodology, introducing the main framework and the motivations for the sub-methods in this research. In chapter 5, we introduce methods for extracting risk factors from images related to weather and visual conditions, named WeatherNet, We also introduce a method for detecting and mapping objects and transport modes from images, in addition to understanding the degradation of the built environment and road surface, named URBAN-i. In chapter 6, we introduce CyclingNet, an action recognition method for detecting cycling near misses from video streams. Chapter 7 shows the causal inference method for understanding the effect of the detected factors, introduced in chapter 5, on the detected near miss events. Chapter 8 addresses the finding and the discussion of the sub-methods, followed by chapter 9 which summarises and concludes the achieved work, in addition to drawing recommendations for planners and policy-makers.

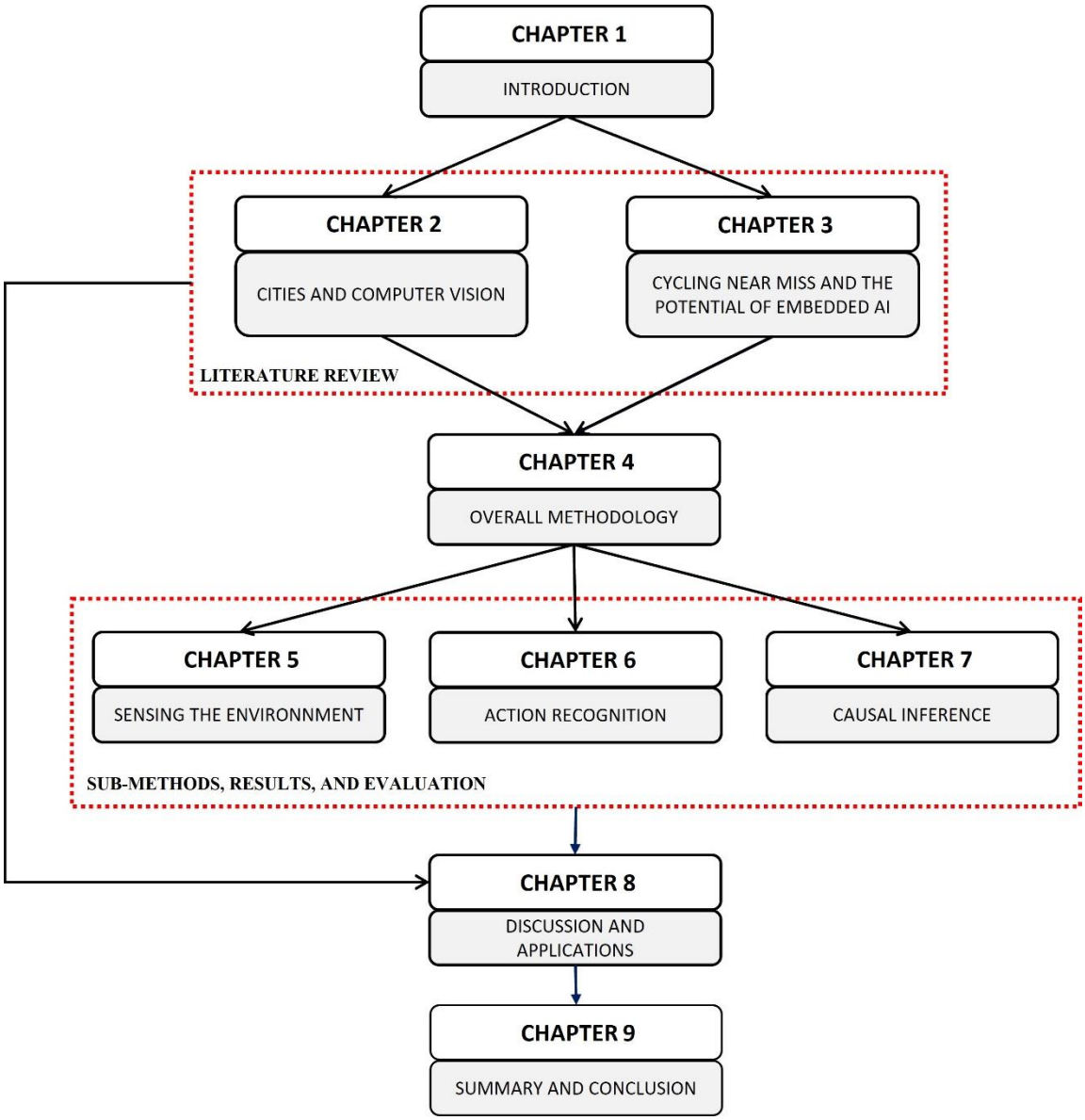

Figure 1.1 Research structure

2

CITIES AND COMPUTER VISION

2.1 Overview

Modelling urban systems has interested planners and modellers for decades. Different models have been achieved relying on mathematics, cellular automation, complexity, and scaling. While most of these models tend to be a simplification of reality, today within the paradigm shifts of artificial intelligence across the different fields of science, the applications of computer vision show promising potential in understanding the realistic dynamics of cities. While cities are complex by nature, computer vision shows progress in tackling a variety of complex physical and non-physical visual tasks. In this chapter, we review the tasks and algorithms of computer vision and their applications in understanding cities. We attempt to subdivide computer vision algorithms into tasks and cities into layers to show evidence of where computer vision is intensively applied and where further research is needed. We focus on highlighting the potential role of computer vision in understanding urban systems related to the built environment, natural environment, human interaction, transportation, and infrastructure. After showing the diversity of computer vision algorithms and applications, the challenges that remain in understanding the integration between these different layers of cities and their interactions with one another relying on deep learning and computer vision. This review aims to provide a resource for urban planners and practitioners by 1) reviewing the main methodologies of computer vision and their applicability to various tasks of urban analytics, 2) illustrating the variation and nuances of deep learning and computer vision algorithms and their limitations in understanding cities, 3) giving a descriptive understanding of the algorithms of computer vision for policy-makers and planners, and how they are used in cities, 4) paving the way for developing AI-generated urban policies by highlighting the key enabling technologies and research directions.

The materials and outcomes of this chapter are published as a journal article in *Cities* journal, entitled: *"Understanding cities with machine eyes: A review of deep computer vision in urban analytics"* (Ibrahim et al., 2020a).

2.2 Review Methodology

The methodology of this review is divided into two parts: 1) manuscripts are collected that summarise the progress in deep learning methods and algorithms that are applicable to computer vision tasks, 2) manuscripts are collected that reflect the application of deep learning and computer vision in understanding cities in the last decade (since 2010). For the first part, we present only the major methodological approaches. Papers that vary or improve on these main approaches are excluded. Most of these studies are presented in premier computer science conferences, including but not limited to CVPR, ICCV, ECCV and NeurIPS. For the second part, we extend the search to peer-reviewed journals and conference proceedings listed in Scopus, Web of Science, Google Scholar and Science Direct, that can be accessed via a combination of keywords such as deep learning, cities, computer vision, land-use modelling, urban perception, prediction, detection, street-level images, aerial or satellite images. This is because the applied computer vision literature is often found in domain-specific journals rather than computer science conferences.

In total, 641 manuscripts were collected to cover the two parts of the methodology. For the second part, the collected manuscripts were filtered to include only those related to computer vision of street-level or aerial images, which use deep learning or hybrid models that include a convolutional structure. Studies that involve deep learning of other data types, such as 2D/3D LIDAR data are excluded. Studies that use classical machine learning or computer vision algorithms without involving deep learning are also excluded, except where they are required to draw a baseline to emphasise advancement or contrast. The algorithms are presented at a descriptive level, and readers are referred to the relevant literature for further details.

2.3 The basics of computer vision

Before exploring the domains where computer vision is applied in cities, it is worth identifying first what computer vision is and what its algorithms are capable of achieving from a generic perspective. Computer vision can be narrowed to the task of learning the qualitative representation of visual elements in their raw form in order to quantify them (LeCun et al., 2015). Similar to human eyes, the computer sees visual objects and creates a cognitive understanding of a scene based on a sequential sample of the presented images or frames of images in a task-specific manner. While computer vision is not new (i.e. Viola & Jones, 2001), deep learning, most specifically Convolutional Neural Networks (CNN), has made it possible for computer vision to tackle various issues and process images more precisely and efficiently (He et al., 2016; LeCun et al., 2015). These deep models, computation capabilities, and the availability of large datasets have made it possible for computer vision to permeate a wide range of applications in realistic settings (Cordts et al., 2016; Lin et al., 2014; Russakovsky et al., 2015). Generally, the logic of computer vision, relying on these deep models, can be summarized as the construction of multiple hidden layers that are capable of accomplishing a range of vision tasks by extracting digital features that may or may not be recognisable to human eyes (Guo et al., 2016; Kuo, 2016; LeCun et al., 2015). The most commonly used are convolutional, pooling, flatten, and fully-connected layers. The general functions of these layers can be summarised as follows:

2.3.1 Convolution layers

Convolution layers refer to the convolution operation or the dot product of a multi-dimensional input array (I) with its kernel (K) to output a feature map. It is defined as:

$$S(i,j) = (k * I)(i,j) = \sum_m \sum_n I(i-m, j-n)\, K(m,n) \quad (2.1)$$

where i,j refer to the size of the input, and m, n refer to the size of the kernel. These layers are responsible for extracting features and they are often coupled with activation functions, such as Rectified linear units (ReLU), to add nonlinearity to the model. More hyperparameters can be used to control the outputs of the convolution operation, such as stride values (the distance where the convolutional kernel is applied). **Fig. 2.1** shows an example of a convolution operation.

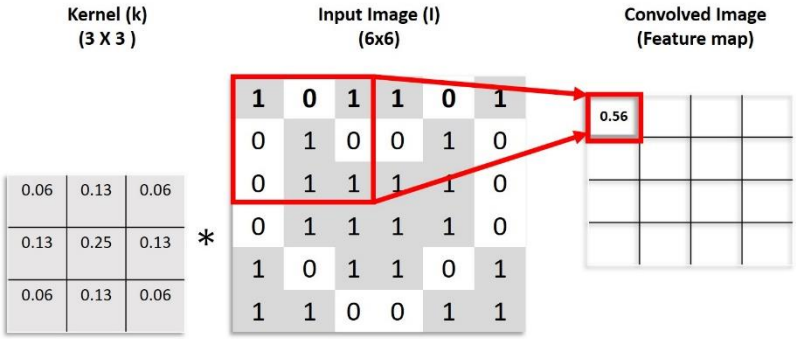

Figure 2.1 Convolution operations with kernel size (3 x 3) and stride (1 x 1).

2.3.2 Pooling layers

Pooling layers replace the values of their inputs, at a given pixel location, with a statistical summary of the nearby values. This helps in making the representations of the feature maps invariant to small changes or translations of the input. Accordingly, this leads to a reduction in the dimensionality of the data. It is worth mentioning that there are different types of pooling functions (i.e. max pooling, average pooling, etc.) of which Max Pooling is the most common. In Max Pooling, the maximum values in a predefined filter size are kept, whereas the remaining ones are discarded. An example of Max Pooling is shown in **Fig. 2.2**. The 4 X 4 input array is divided into four 2 X 2 arrays. The 2 X 2 output returns the maximum value of each 2 X 2 array, shown in grey on the figure.

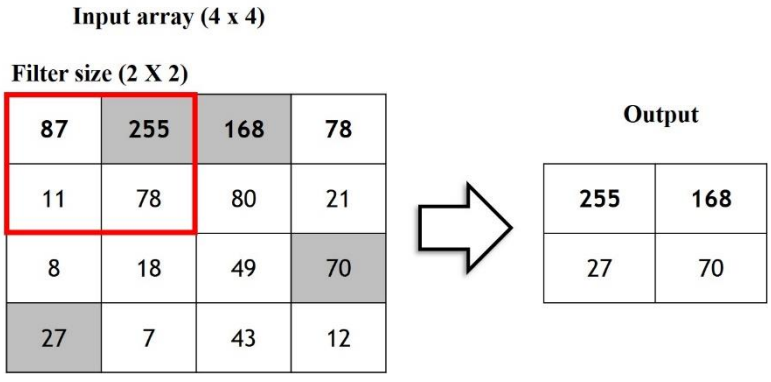

Figure 2.2 Max pooling operations with kernel size (3 x 3) and stride (2 x 2).

2.3.3 Flatten layers

A Flatten layer refers to the transformation of a multi-dimensional matrix to a vector to be fed forward to a fully-connected layer. For example, if an input of a shape (Batch size, 4, 4) is given to a Flatten layer, the output would be (Batch size, 16).

2.3.4 Fully-connected layers

A fully-connected layer, or a dense layer, is the primary component of a Multi-Layer Perceptron (MLP) model. In fully-connected layers, all neurons are fully connected to the neurons in the next layer and all previous activations in the case of a previous layer. Their activations can be computed based on the summation of their input multiplied by their weights followed by a bias offset as:

$$\widehat{Y} = \mathrm{s}(\sum_i^n w_i x_i + b) \tag{2.2}$$

given that n is the total number of neurons, x_i are the input neurons and w_i represents the weight of each neuron, b represents the bias, s is the activation function.

2.3.5 Residual blocks

The residual blocks refer to the ability of the model to learn the underlying mapping by fitting stacked layers to an identity mapping denoted as $\mathrm{f}(\mathrm{x}) + x$). This formation can be achieved by feed-forward of neural networks with a shortcut connection that can skip one or more layers. These skip connections aim to perform identity mapping by adding their outputs to the outputs of the stacked layers (See **Fig. 2.3**). This has proven to be a successful approach for maximising the ability of the model to learn from the representation of its input while addressing the vanishing problem of gradients in deep networks. There is a variation of residual networks or so-called ResNet, for instance, ResNet 18, 34,50, 101, or 152. These values refer to the number of layers built in a given network, including their residual links. For a further explanation for architecture and the hyperparameters of the model, see (He et al., 2016).

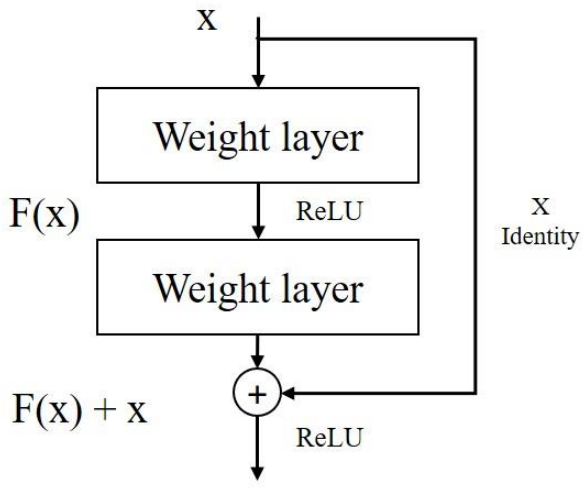

Figure 2.3 Example of Residual block adapted from (He et al., 2016)

2.3.6 Advanced layers and techniques

There are more advanced layers such as Long Short-Term Memory (LSTM) and self-attention, which will be explained in the upcoming chapters when they are utilised (i.e. Chapter V, and VI). The types, numbers, and orders of these layers are responsible for determining the functionality and the optimisation of both accuracy and time needed for the training and the inference of the model. The structure of the model and the fine-tuning of the various hyperparameters represents the innovation and the advancements of the state-of-the-art for pattern recognition for a given task (LeCun et al., 2015).

2.4 The tasks of computer vision

Depending on the type of visual task, deep models can be trained differently with different layers and different sets of algorithms (Guo et al., 2016). As shown in **Fig. 2.4**, these algorithms of computer vision can be subdivided based on eight fundamental tasks, upon which other tasks can be framed and built. These are; image classification, segmentation and localisation, tracking, action recognition, perception, generative models, clustering, and decision-making. **Table 2.1** shows the literature related to different computer vision tasks. It expands on the methods related to each task and its subcategories.

Figure 2.4 Computer vision algorithms types

Table 2.1: Methods related to different computer vision tasks

VISION TASK	SUB-CATEGORY	METHOD	
CLASSIFICATION		ALEXNET	(Krizhevsky et al., 2012)
		VGGNET	(Simonyan and Zisserman, 2015)
		GOOGLENET	(Szegedy et al., 2015)
		RESNETS	(He et al., 2016)
		DENSNET	(Huang et al., 2017)
SEGMENTATION AND LOCALISATION	OBJECT DETECTION	R-CNN	(Girshick et al., 2014a)
		FAST R-CNN	(Ren et al., 2015)
		YOLO	(Redmon et al., 2016)
		SSD	(Liu et al., 2016)
		YOLOV2	(Redmon and Farhadi, 2017)
		YOLOV3	(Redmon and Farhadi, 2018a)
		RETINANET	(Lin et al., 2020)
	SEMANTIC SEGMENTATION	DEEPLAB	(L.-C. Chen et al., 2016b)
		U-NET	(Ronneberger et al., 2015)
		SEGNET	(Badrinarayanan et al., 2017)
		-	(Long et al., 2015)
		-	(Peng et al., 2017)
		-	(L.-C. Chen et al., 2016a)
		-	(H. Zhao et al., 2017)
		-	(Yu and Koltun, 2016)
		REFINENET	(Lin, Milan, Shen, & Reid, 2017)
		-	(Chen et al., 2017)
		-	(Jegou et al., 2017)
		FOVEANET	(Li et al., 2017a)
		LINKNET	(Chaurasia and Culurciello, 2017)
		-	(Yang et al., 2018)
TRACKING OBJECTS		-	(Kang, Ouyang, Li, & Wang, 2016)
		-	(Girdhar et al., 2017)
		PATHTRACK	(Manen et al., 2017)

ACTION RECOGNITION	HUMAN POSE ESTIMATION	DENSEPOSE	(Guler et al., 2018)
		MULTIPOSENET	(Kocabas et al., 2018)
		RMPE	(Fang et al., 2017)
		-	(Z. Cao et al., 2017)
	ACTION CLASSIFICATION	-	(Girdhar and Ramanan, 2017)
			(Bilen et al., 2016)
			(Zhu et al., 2019)
			(Guo et al., 2018)
			(B. Zhang et al., 2016)
	TEMPORAL ACTION DETECTION		(Diba et al., 2017)
			(Gemert et al., 2015)
			(Shou et al., 2017)
			(Escorcia et al., 2016)
			(Li et al., 2016)
			(Xu et al., 2017)
			(Chao et al., 2018)
			(Buch et al., 2017)
			(Y. Zhao et al., 2017)
	SPATIO-TEMPORAL ACTION DETECTION		(Chen & Corso, 2015)
			(Becattini et al., 2021)
			(Saha et al., 2017)
			(Gemert et al., 2015)
			(Zhu et al., 2017)
			(El-Nouby and Taylor, 2018)
			(Saha et al., 2016)
			(Singh et al., 2017)
			(Mettes et al., 2016)
			(Weinzaepfel et al., 2015)
TRAINED TO PERCEIVE	UNDERSTANDING SCENES		(Eslami et al., 2018)

	ESTIMATING DEPTH	-	(Cao et al., 2018)
		-	(He, Wang, & Hu, 2018)
TRAINED TO CREATE	GANS	-	(Goodfellow et al., 2014a)
		-	(Radford et al., 2015)
		-	(S. Reed et al., 2016)
		STACKGAN	(H. Zhang et al., 2017)
		-	(Isola et al., 2017)
CLUSTERING		-	(Caron et al., 2018)
		-	(Xie et al., 2016)
		DEEPCLUSTER	(Tian et al., 2017)
MAKING DECISIONS	DEEP Q-LEARNING		(Mnih et al., 2013)
			(Hester et al., 2017)
	DOUBLE DEEP Q-LEARNING		(van Hasselt et al., 2015)
	DUELING DEEP Q-LEARNING		(Wang et al., 2016)
	A3C		(Mnih et al., 2016)

2.4.1 classification

Deep learning models, most specifically Convolutional Neural Networks (CNN), have shown substantial progress in classifying images of a wide spectrum of classes (LeCun et al., 2015). Various deep CNN models with different architectures and hyper-parameters have been designed to recognise visual objects in large repositories of images (Russakovsky et al., 2015). Starting with AlexNet (Krizhevsky et al., 2012), VGGNet (Simonyan and Zisserman, 2015), GoogLeNet (Szegedy et al., 2015), ResNet (He et al., 2016) and most recently, DenseNet (Huang et al., 2017), these CNN models are able to accurately recognize and classify a wide range of images. For instance, ResNet-152 achieved a 4.49% top-5 error score on the validation set of ImageNET (He et al., 2015a).

2.4.2 Segmentation and localisation

Segmentation and localisation are the processes of identifying multiple objects in a single image. These models use a single deep model in an end-to-end fashion, in which the first part of the model is an image classifier followed by different types of layers to localise different objects with a given confidence. Notable examples include the Region-based CNN model (R-CNN) (Girshick et al., 2014b), Fast R-CNN (Ren et al., 2015), You Only Look Once (YOLO) (Redmon & Farhadi, 2017, 2018) and the MultiBox Detectors for fast image segmentation, or so-called Single Shot Multi-Box Detector (SSD) technique (W. Liu et al., 2016). CNN models have shown significant progress in recognising and detecting objects in images with a minimal inference time and high overall validation accuracy. YOLOv3 achieves a 93.8% top-5 score on the COCO dataset (Redmon and Farhadi, 2018b).

On the other hand, understanding the different components of urban scenes from street view images relying on computer vision is another crucial application for scene awareness (Li et al., 2017a). Scene parsing relying on semantic segmentation is a continuous success of CNN models for understanding and classifying the different components of an urban scene (Badrinarayanan et al., 2017; Chen et al., 2017; L.-C. Chen et al., 2016a; Lin et al., 2017; Long et al., 2015; Peng et al., 2017; Yu and Koltun, 2016; H. Zhao et al., 2017). This pixel-level classification made it possible to recognize and understand the deep subtleties of the different components of an urban scene (i.e. road area, building, people, cars, vegetation). While such a complex approach is still exclusive to the applications of autonomous vehicles, it can be used to understand and extract information for urban studies and urban modelling. For further explanation related to localisation and object detection, see (Xiao et al., 2020).

2.4.3 Tracking objects

After building a system of object detection, computer vision can be used for tracking multiple objects in a complex scene by adding features that correlate a pair of consecutive frames. This tracker system is capable of identifying a candidate box at each frame-level jointly with their time deformations (Girdhar et al., 2017). While different tracker systems can be built based on correlation filtering and online learning techniques between consecutive frames (X. Zhang et al., 2018), the state-of-the-art research in object tracking uses an end-to-end CNN model to tackle both detection and tracking. This can add more advanced features (i.e. dealing with occlusion issues) for tracking various elements (Girdhar et al., 2017; Hou et al., 2017; Kang et al., 2016). For further explanation related to deep visual tracking, see P. Li, Wang, Wang, & Lu (2018).

2.4.4 Action recognition

Computer vision coupled with deep CNN models is not only capable of tracking the motion of an object in a complex scene, but also classifying its multiple actions while tracking (Bilen et al., 2016; Wang et al., 2015; B. Zhang et al., 2016). Various computer vision algorithms have been developed to tackle humans poses and their interaction with an external object in a complex scene (El-Nouby and Taylor, 2018; Saha et al., 2016; Soomro and Shah, 2017; Weinzaepfel et al., 2016). 2D or 3D convolution layers (with or without the spatiotemporal dimensions) can identify the action of the object from its pose in relation to another target object. For instance, from the pose of a person sitting on a bike, the algorithms of computer vision can identify cycling as an action. This concept of the triplet inputs (object, verb, target) has been seminal for tackling real-world events and behaviours, from a simple still image to multi-frame images (Girdhar et al., 2017).

2.4.5 Perception

Perception tasks can be seen as classification or regression tasks that predict information that is not necessarily embedded directly in the image but can be inferred from the overall structure of the image. Perceiving a neighbourhood as safe or unsafe, for example, can be seen as a perception task, in which the machine extracts features from the structure of an image to classify the safety of the image. Even though understanding the overall gist of a scene is seminal for understanding more than an object in an image (Oliva and Torralba, 2006), few works have been done in this domain. The complexity of tackling this subject lies in sensing the class of an image by sensing the overall profound features of the image rather than identifying an object in the image.

Moreover, seeing what is far and what is close just by looking at a still image is another advantage of computer vision relying on deep CNN models. Cao, Wu, & Shen (2017) trained deep CNN models to estimate the depth in a single image by labelling the different depths on the image and training the model as a classification task. In contrast, He, Wang, & Hu (2018) trained a deep CNN model to estimate the depth of a monocular image relying on the information of focal length that has proven to outperform the other state-of-the-art depth estimation algorithms based on deep learning models.

2.4.6 Generative models

Generative models refer to the ones that tend to output synthesized data by learning the representation of their input data in an unsupervised fashion, conditionally or unconditionally.

There is a range of algorithms that are classified as generative models, such as Restricted Boltzmann Machine (RBM), deep belief networks, Autoencoders, and Generative adversarial Networks (GANs) (Goodfellow et al., 2017). This section refers only to GANs, which generate synthetic graphical data in an unsupervised training fashion relying on images as input. Unlike other tasks related to computer vision, the deep models of GANs, introduced in 2014, enable machines to generate new information that is similar to what the model has been trained to identify (Goodfellow et al., 2014). In other words, if the model is trained on images of trees, by using GANs, the model can generate a new image of a tree that preserves the fundamental features of a tree but with a new visual identity. This progress of deep learning enables the creation of unique objects or scenes by understanding the underlying features of the trained images or videos.

GANs are trained differently from the abovementioned deep models, not only in terms of layers but rather, instead of the single end-to-end model, two deep parallel models are trained that compete with one another (Goodfellow, 2016; Goodfellow et al., 2014; Radford et al., 2015). The first one, the Generator model, generates new images to deceive the second model that holds the ground truth data, while the second model, the Discriminator model, blocks this new image until the generator model becomes advanced enough to generate new images that are similar enough to the ground truth that the discriminator model can no longer refuse them. This computationally intensive training, in an unsupervised manner, opens the door for computer-based creativity without the prior supervision of humans.

GANs have been utilised in various applications. Isola et al. (2017) used conditional GANs to translate from one form of an image to another. For instance, by giving the model a satellite image of a location, the model can give the semantic segmentation of the location or vice versa. Zhang et al. (2016) created the stackGAN model to transform a text description of an image into a photo-realistic synthesis. Moreover, Reed et al. (2016) have pushed the algorithms of GANs further. The machines can learn to draw not only from text distributions but also by telling the machine what and where to draw on the canvas. Apart from the daily-life applications, GANs have been used in the simulation of 3D energy particle showers and physics-related applications (Paganini et al., 2018).

2.4.7 Clustering

Commonly, clustering is a form of unsupervised learning, in which the machines are able to cluster different still images or multi-frame images based on their content or embedded objects without prior human supervision (Caron et al., 2018; Tian et al., 2017; Xie et al., 2016). So far, different computer

vision algorithms have been developed to tackle this task and eliminate the need for a long process of manual labelling from still images. Recently, Eslami et al. (2018) introduced the Generative Query Network (GQN) for scene representation without human supervision. The GQN takes images from a different perspective as an input and generates a visual representation of the scene from an unobserved perspective. This process of coupling generative models with clustering introduces a new form of artificial intelligence to understand scene representation without human supervision.

2.4.8 Making decisions

By looking at the edge of computer vision and coupling its deep models with reinforcement learning, or so-called Deep Reinforcement Learning (DRL), machines can be trained to explore and compute the outcomes of different scenarios in order to make real-time decisions based on visual aspects of the environment (Hester et al., 2017; Mnih et al., 2016). This level of cognitive ability of machines by applying one or more of the abovementioned tasks can enable an agent to grasp information and interact with an environment to optimize target resources without human supervision.

Due to the complexity of the algorithms related to this subject, most examples are in virtual or gaming environments (Mnih et al., 2013). However, most significantly, Mirowski et al. (2018) utilised DRL to enable a machine to navigate through the unstructured environment of the street network relying on street-level images. In this work, the machine learns to navigate by understanding landmarks from images and to determine its location and its target destination.

In summary, Table 1 shows the literature related to the different computer vision tasks. It expands on the methods related to each task and their sub-categories.

2.5 Recognising the urban world

Understanding the dynamics of cities remains a complex issue. Data collection, for instance, is one of the crucial domains where automation is highly desirable, in which computer vision has been successfully applied in capturing and analysing various objects depicted in urban scenes. Specifically, scene parsing and semantic segmentation represent crucial tasks of computer vision for a better understanding of the elements of an urban scene. From images, computer vision can localize multiple objects in cities or simply segment the entire scene based on a group of themes, such as sky, ground, road, building, vegetation, etc. (Chaurasia and Culurciello, 2017; Zhou et al., 2017). By putting all the above-mentioned tasks together, computer vision shows good potential in urban analytics for analysing the multi-layers of cities. For the purposes of this review, we define these layers as; the built environment, the natural environment, humans and their physical interactions, transport modes and traffic-related issues, and infrastructure. The main reason for breaking down cities in these layers is to be able to tackle the applications of computer vision in each field of science related to urban analytics, in which the methods, scope, language used, and the nature of work may vary depending on the discipline. For instance, research that has been done in understanding the built environment may vary in nature from that done to understand transportation, even though the methods of deep learning and computer vision may be similar.

Fig. 2.5 shows examples of computer vision applications in cities to detect multidisciplinary tasks that belong to the five layers of cities, whereas **table 2.2** shows the applications of computer vision to these layers. Each layer is broken down into further subcategories as appropriate.

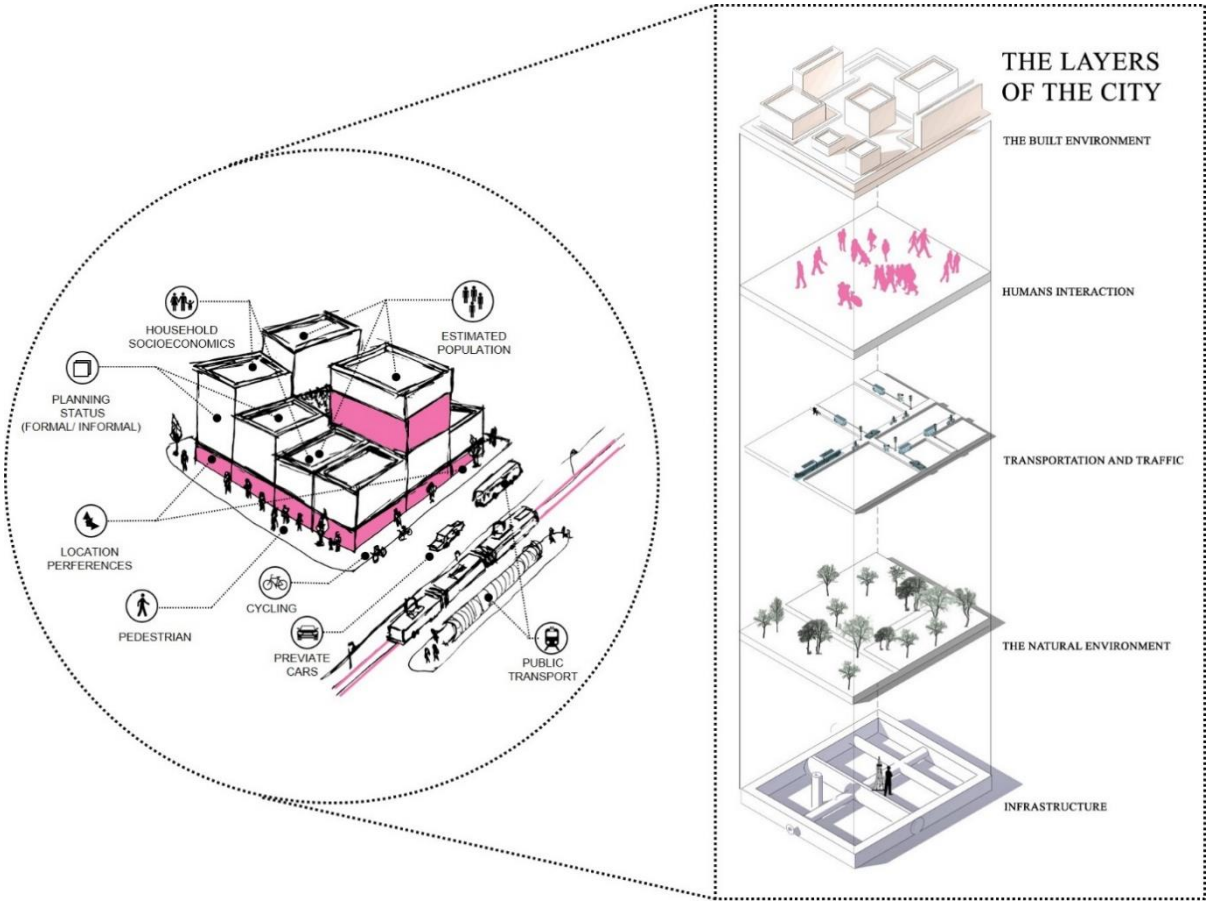

Figure 2.5 The layers of the city where computer vision is applied

Table 2.2 Computer vision algorithms that tackle urban-related issues

CITY LAYER	CATEGORY	METHOD	
THE BUILT ENVIRONMENT	URBAN COMPONENTS	SEMANTIC SEGMENTATION	(Zhou et al., 2017) (Chaurasia and Culurciello, 2017) (Chen et al., 2016) (He et al., 2019) (Helbich et al., 2019) (Amirkolaee and Arefi, 2019) (Wurm et al., 2019) (Cordts et al., 2016)
		OBJECTDETECTION	(Yang et al., 2019) (R. Chew et al., 2018)
	LAND USE CLASSIFICATION	CLASSIFICATION AND SEMANTIC SEGMENTATION	(Demir et al., 2018) (Sharma et al., 2017) (Audebert et al., 2018)
		CLASSIFICATION	(Wang, Xu, Dong, Gui, & Pu, 2018) (Srivastava et al., 2019) (R. F. Chew et al., 2018)
	URBAN PERCEPTION	CLASSIFICATION AND PERCEPTION	(J. Zhao et al., 2018)
			(Law et al., 2018) (Zhang et al., 2019) (Seresinhe et al., 2017) (Oliva and Torralba, 2006) (W. Wang et al., 2018) (Salesses et al., 2013) (Dubey et al., 2016) (Naik et al., 2016) (Quercia et al., 2014)
	URBAN SAFETY		(De Nadai et al., 2016) (Naik et al., 2014)
HUMAN INTERACTION		OBJECT DETECTION	(Priya et al., 2015)
TRANSPORTATION AND TRAFFIC	TRAFFIC SURVEILLANCE	CLASSIFICATION AND OBJECT DETECTION	(Bottino et al., 2016)
		ACTION RECOGNITION	(Yu et al., 2017)
		OBJECT DETECTION	(Yang and Pun-Cheng, 2018)
	SAFETY/ ACCIDENTS	CLASSIFICATION AND OBJECT DETECTION	(Sayed et al., 2013) (Zaki et al., 2013)
THE NATURAL ENVIRONMENT	FLORA AND FAUNA	OBJECT DETECTION	(Cai et al., 2018a)
			(Hong et al., 2019)
		SEMANTIC SEGMENTATION	(Krause et al., 2018) (Williams et al., 2017)
		CLASSIFICATION	(Mohanty et al., 2016) (Sun et al., 2017)
	ENVIRONMENTAL AND WEATHER CONDITIONS	CLASSIFICATION AND PERCEPTION	(C. Liu et al., 2016) (W. Liu et al., 2017) (Villarreal Guerra et al., 2018) (Elhoseiny et al., 2015)

			(Sirirattanapol et al., 2019)
INFRASTRUCTURE	CONCRETE CONDITION	OBJECT DETECTION	(Cha, Choi, & Büyüköztürk, 2017) (B. Wang et al., 2018)
	PAVEMENT/ ROAD CONDITION	OBJECT DETECTION	(Maeda et al., 2018)
	BRIDGE COMPONENT RECOGNITION	SEMANTIC SEGMENTATION	(Narazaki et al., 2017)

2.5.1 The built environment

This section addresses cities from an architectural and urban design perspective, for example, understanding cities from a land-use perspective, the level of the physical appearance of the street-level that may indicate or measure housing prices, or even the level of safety with a certain neighbourhood.

When it comes to understanding the built environment, different challenges face urban planners and policy-makers. For example, modelling the physical appearance of complex urban areas is a multi-faceted issue that is vital for planners and policy-makers for making decisions for improving living conditions in cities. The collection of data that reflects the current status of the built environment is a critical issue for urban analytics. So far, the applications of computer vision have merged not only to detect various urban components but also to understand the appearance and the safety factors of an urban scene. While there is a wide range of applications of computer vision in cities, these applications can be divided into two approaches that either analyse cities from street-level images or remote sensing data such as satellite images.

2.5.1.1 Seeing cities from above

Analysing cities from above relying on remote sensing and geographical information systems (GIS), perhaps, is the most common approach for planners (J. Chen et al., 2016). Applications of computer vision jointly with these systems are capable of automating urban tasks such as mapping and zoning. Most recently, the notion of DeepGlobe (Demir et al., 2018) aimed to describe the earth from satellite images. DeepGlobe can extract streets, buildings and the different types of land-cover. Similarly, (Wang, Xu, Dong, Gui, & Pu, 2018) used a CNN model to segment satellite images into multi-classes at the pixel level. Marcos, Volpi, Kellenberger, & Tuia (2018) used the CNN model for land cover mapping, solving the issue of rotation of objects. Vanhoey et al. (2017) introduced VarCity as an approach of automating the construction of a city-scale 3D model based on semantic segmentation and machine processing of urban components (buildings, built environment, vegetation, roads, etc.).

Furthermore, relying on deep learning, Amirkolaee & Arefi (2019) estimated heights from single aerial images, Wang et al. (2018) used deep CNN models for remote sensing image registration. Wurm, Stark, Zhu, Weigand, & Taubenböck (2019) relied on semantic segmentation to classify slum areas from aerial images.

These presented methods may differ from one another in terms of accuracies or purposes. However, the main limitation remains in how these models can be generalised to fit for multiple locations beyond the context where the models are trained and tested.

2.4.1.2 Seeing cities from a street-level

While it is vital to understand the overall urban systems of cities from an aerial view, seeing cities from the street-level adds more layers of information. These images can capture rapid urban changes in day-to-day life and offer more opportunities to model urban dynamics. However, capturing these rapid urban changes is a more complex task. Street-level images, taken by individuals or represented in Google's Street View API, have been used to identify a wide range of urban components, from buildings to small objects such as street signs. For instance, Nguyen et al. (2018) used a CNN model to detect building types, crosswalks, and street greenness as a way to automatically quantify neighbourhood qualities.

Similarly, a range of applications based on classifying, segmenting and localising pixels from street-level images was a common approach for understanding the components of an urban scene (Chaurasia & Culurciello, 2017; Li, Jie, et al., 2017; Yang, Yu, Zhang, Li, & Yang, 2018; Zhou et al., 2017). Scene parsing relying on semantic segmentation is a continual success of CNN models for understanding and classifying the different components of the built environment at a pixel-level (Badrinarayanan et al., 2017; Chen et al., 2017; L.-C. Chen et al., 2016a; Lin et al., 2017; Long et al., 2015; Peng et al., 2017; Yu and Koltun, 2016). Relying on both street-level images and satellite images, Kang, Körner, Wang, Taubenböck, & Zhu (2018) used a deep CNN model to classify land use in satellite images by learning from building blocks of similar functions.

Quantifying the physical and non-physical appearance of cities is another area that has been intensively researched. Naik et al. (2016) quantified the physical appearance of neighbourhoods based on individuals' ranking perceptions of the urban spaces using a framework of two CNN models that are concatenated and fused to predict a score for paired street-level images, known as Streetscore-CNN. Similarly, Zhang et al. (2018) quantified urban spaces of street-level images labelled into six categories (Depressing, Boring, Beautiful, Safe, Lively, Wealthy) based on a crowdsourced dataset (MIT places pulse). By applying a supervised deep CNN model, they can predict the class for a given street view image. Liu, Silva, Wu, & Wang (2017) evaluated the urban visual appearance based on two indicators of the quality of street façade and the continuity of the street walls relying on the expert ranking that is evaluated with a public survey. Moreover, Naik, Kominers, Raskar, Glaeser, & Hidalgo (2017) have used computer vision to measure the dynamics of neighbourhood characteristics from time series street view images adjoined with socioeconomic data in five US cities. As a different approach, Law et al. (2019) used street view images to identify housing prices from urban perception relying on computer vision.

While seeing cities at a street-level adds more information and gives an opportunity to understand the rapid changes that occur in an everyday urban scene in cities, the images used from Google street-view images only represent urban areas at a single weather condition, commonly clear weather, neglecting other visual and weather conditions that impact the appearance of cities. Furthermore, more research is needed on how to make the best use of street-level images coming from various sources, such as CCTV, dashcams or crowdsources, within and across domains.

2.5.2 Humans interactions

Deep learning and computer vision have shown substantial progress in understanding a wide range of applications not only related to human detection but also understanding their activities and interaction with other objects (Kale & Patil, 2016; Mohamed & Ali, 2013; Zhang et al., 2017). Such approaches can assist planners and policy-makers to better understand tasks related to wellbeing and human behaviour in cities. For instance, Priya, Paul, & Singh (2015) used deep learning and computer vision to classify human actions, such as walking, running, sitting or dancing, for multi-frame images. Guler, Neverova, & Kokkinos (2018) used a region-based CNN model (RCCN) to estimate the various human poses from a single image to better understand human interactions. Gkioxari, Girshick, Dollar, & He (2017) used computer vision to predict human actions over a specific target object from every day still images. This novel approach provides substantial progress in understanding human interaction with different objects. Furthermore, adjoining human pose detection with tracking (Girdhar et al., 2017) used computer vision to detect and track key human body points from videos. This could enable, for example, tackling various issues related to human safety and wellbeing in cities, such as detecting when a person falls or detecting abnormal behaviour such as crime-related actions. Indeed, a knowledge gap appears in scaling up deep computer vision algorithms for monitoring and detecting irregular behaviours at a city level in real-time.

2.5.3 Transportation and traffic

Transportation and traffic is a crucial and complex layer that merges and interacts with other layers of the city. There is a wide range of computer vision applications that aim to tackle transport modes and their common issues, such as road safety and optimisation of traffic (Buch et al., 2011; Priya et al., 2015). Subjectively, traffic surveillance and intelligent transportation systems hold the largest share of computer vision related applications in cities. Typical tasks include vehicle detection, counting, overtake detection, and traffic incident detection (Mahmud, Ferreira, Hoque, & Tavassoli, 2017; Yang & Pun-Cheng, 2018). A full review of the literature on vehicle detection is beyond the scope of this chapter. For a comprehensive review, consult Yang and Pun-Cheng (2018).

Understanding the different traffic scenarios and interactions of the different transport modes by computer vision is crucial. Bottino, Garbo, Loiacono, & Quer (2016) introduced ‘Street Viewer’ as a system to tackle and analyse the different scenarios of traffic behaviour from street view images. Sayed, Zaki, & Autey (2013) used computer vision to evaluate the safety measures of vehicle-bicycle conflicts. Zaki, Sayed, Tageldin, & Hussein (2013) used computer vision to analyse the conflicts among pedestrians and vehicles at a signalized intersection. Zaki & Sayed (2013) introduced a framework relying on computer vision to classify the different types of road users.

Building on the aforementioned artificial intelligence approaches for traffic-related issues, computer vision is a core element when it comes to smart mobility and autonomous vehicles. Different applications relying on computer vision are being used to make transport modes aware of the surrounding environments either for safety indications or moving towards a self-navigation system. However, the technology of autonomous vehicles is not the focus of this research but rather the interactions of transport modes with the aforementioned layers in cities (Faisal et al., 2019).

2.5.4 The natural environment

The natural environment (i.e. green space, landscape, climate conditions, etc.) is a crucial layer when it comes to understanding cities. It influences our perception of the visual appearance of the built environment and also affects mobility and human interaction in cities. Different aspects related to this natural layer of cities have been tackled by computer vision. These applications vary from mapping vegetation and greenery in cities, or so-called 'Treepedia' (Cai et al., 2018b), estimating vegetation area (Stubbings et al., 2019), identifying plant types (Krause et al., 2018; Sun et al., 2017), to a deeper understanding of the natural environment and wildlife such as detecting plant-related diseases (Mohanty et al., 2016) and understanding the patterns of social interaction among animals (Robie, Seagraves, Egnor, & Branson, 2017).

Deep learning and computer vision have also been used to infer the weather, climatic and air conditions in cities. Liu et al. (2016) used the CNN model to identify extreme weather conditions from aerial images of climate simulations and reanalysis products. Liu, Tsow, Zou, & Tao, (2016) used images to analyse particle pollution for Beijing, Shanghai and Phoenix relying on region of interest selection, feature extraction and regression models. Z. Li et al. (2019) developed a model to detect clouds from high-resolution aerial view images relying on CNNs, named multi-scale convolutional feature fusion.

While there is noticeable progress in terms of methods development and accuracy enhancement among the presented papers, the common limitation remains in the lack of a single model or a framework that fuses various models to infer the different weather and environmental conditions.

2.5.5 Infrastructure

Cities comprise a range of infrastructure systems that represent a large portion of their economy. Inspecting these systems and detecting their deficiencies is a crucial aspect for engineers and planners in cities. The focus of this section differs from the built environment section by analysing materials and the civil engineering related issues that are not covered in the aforementioned sections.

So far, the applications of computer vision have been seen in a wide range of domains related to infrastructure and civil engineering (Gopalakrishnan, 2018; Griffiths and Boehm, 2018), most importantly in analysing defects (Feng et al., 2017). For instance, B. Wang, Zhao, Gao, Zhang, & Wang (2018) used computer vision to detect concrete crack damage. Similarly, Cha, Choi, & Büyüköztürk (2017) applied computer vision relying on a deep CNN model to detect crack damage of concrete. On the other hand, Maeda et al. (2018) used computer vision to detect road damage from images that are taken from mobile devices.

2.6 What remains missing?

Section 3 of this chapter presented the different types of computer vision algorithms that are available to researchers, and the sectors in which they have been applied were presented in section 4. Typically, these models have been applied in a sectoral fashion to a specific problem. Comparatively, little attention has been placed on how to understand the interconnections between the different layers of the city. These interconnections will eventually lead to increased capabilities of computer vision and AI to aid decision making and policy. In this section, we outline two under-researched areas in which computer vision has enormous potential.

2.6.1 Integrated models of the layers of the city

A significant challenge remains in modelling the interconnectedness and dependencies of the different layers of the city that were introduced in **Fig. 2.5**. The first step in this regard is the integration of models that have been developed for each layer in isolation. For example, there is still a knowledge gap in how to use computer vision coupled with deep learning to understand the interaction between people in cities and transport modes, or the influence of one mode on the others in terms of accessibility and safety. While the technology is there, the challenge remains in combining different models in a framework that enables them to tackle complex, multi-layered issues using the same data source, rather than just combining or fusing outputs from different data sources. On the other hand, even if the knowledge of the models is transferable among the different layers of cities, the challenges remain in finding comprehensive image data sources that cover a wide scope of tasks and functions in cities.

2.6.2 The scale of applying computer vision in cities

Understanding cities requires both local and global perspectives, in which scale plays a crucial role in tackling urban issues. Different algorithms have been used to understand, for instance, individuals' actions and activities. Challenges remain in applying and scaling up such algorithms to the city level. Although there are different models, as discussed in the literature, that extract information at the city scale, the nature of the developed algorithms is still limited to the analysis of a certain location or a city. The reason for this is either because of a lack of computational resources or the inability of trained models to generalise to a larger dataset at a city level. Models often require further training and optimisation to be deployed in real-life applications. It is well known that computer vision algorithms require large sets of labelled data, which must often be manually labelled. Labels can be crowdsourced, but there is often a cost involved and accuracy is difficult to guarantee. Semi- or weakly supervised learning methods are promising approaches in this regard (S. Guo et al., 2018).

2.7 Summary

Understanding cities has been a profound interest for many scholars across a wide range of disciplines. Modelling the different urban systems of cities is a long term purpose for many urban and transport planners. While cities are complex by nature and classical urban modelling may not capture the actual complexities of urban systems, computer vision shows progress in tackling a variety of complex physical and non-physical visual tasks. In this chapter, we provide a review of deep learning and computer vision and its application so far in understanding cities. The chapter highlights the different types of algorithms of computer vision and their application to cities and their multifaced issues. It aimed to show the nuances of the variations of these algorithms within the same task. It also aimed to show what has been done so far to understand cities by machine vision and what remains missing for future research work within this domain.

We attempt to highlight the potential role of computer vision in understanding the interactions between the built environment, people and transportation to tackle the complexity and nonlinearity of many urban and transport issues for better policy-making and planning safer cities. We also highlight the current limitations that require further work to reach an integrated computer vision-based urban models that are capable of making automatic decisions. While there are substantial works achieved in response to the different layers of cities as shown in this chapter, the challenges remain

in understanding the intersection between these different layers and their interaction and causality to one another by computer vision. For instance, there are still knowledge gaps for utilising the application of computer vision and deep learning to understand the interaction between people in cities and transport modes, or the influence of one mode on the others in terms of accessibility and safety. While the technology there, for instance, for tackling and predicting incidences, the issue remains in deploying a framework that agglomerate various models in a pipeline fashion.

Inference and prediction are at no doubts a great success for assisting our understanding and estimating different events that occur in cities, however, making an automated and optimised decision based on models' predictions remains a crucial aspect. There are still a limited number of researches that have been done in the domain of deep reinforcement that tackles real-life issues in urban areas. As previously mentioned, the main reason for this is the complexity of applying such algorithms in real-life events. However, DRL remains a promising sector that takes computer vision from a model to detect a model that can be learned to know where to look and how to decide without humans' supervision.

3

CYCLING NEAR MISSES AND THE POTENTIAL FOR AN EMBEDDED AI SYSTEM

3.1 Overview

Whether for commuting or leisure, cycling is a growing transport mode in many countries. However, cycling is still perceived by many as a dangerous activity. Because the modal share of cycling tends to be low, serious incidents related to cycling are rare. Nevertheless, the fear of getting hit or falling while cycling hinders its expansion as a transport mode, and it has been shown that focusing on fatal and seriously injured casualties alone only touches the tip of the iceberg. Compared with reported incidents, there are many more incidents in which the person on the bike was destabilised or needed to take action to avoid a crash, named near misses. Because of their frequency, data related to near misses can provide much more information about the risk factors associated with cycling. The quality and coverage of this information depends on the method of data collection, from survey data to video data and processing, from manual to automated. There remains a gap in our understanding of how best to identify and predict near misses and draw statistically significant conclusions, which may lead to better intervention measures and the creation of a safer environment for people on bikes. In this chapter, we review the literature on cycling near misses, focusing on the data collection methods adopted, the scope and the risk factors identified. In doing so, we demonstrate that, while many near misses are a result of a combination of different factors that may or may not be transport-related, the current approach of tackling these factors may not be adequate for understanding the interconnections between all risk factors. To address this limitation, we highlight the potential of extracting data using a unified input (images/videos) relying on computer vision methods to automatically extract the wide spectrum of near miss risk factors, in addition to detecting the types of events associated with near misses. This review aims to provide a resource for planners, policy-makers and researchers by 1) defining near misses, their types and their risk factors, 2) reviewing the main methodologies of recording and analysing near misses and their applicability and limitations, 3) showing a summary of the variation of near misses and their limitations in understanding the stated issue, 4) introducing a new potential framework understanding near misses through machine vision, in which models can be utilised to extract risk factors and infer near misses, 5) paving the way for developing AI-related automated systems that could be used to tackle crash risk by focusing on near misses, in which we highlight the key enabling technologies and research directions.

The materials and outcomes of this chapter are published as a journal article in *transport reviews* journal, entitled: *"Cycling near misses*: A review of the current methods, challenges and the potential of an AI-embedded system*"* (Ibrahim et al., 2020b).

3.2 Review methodology

We adopted a systematic review approach using PRISMA guidelines (PRISMA, 2015). Figure one shows the flow of the information through the different stages of the systematic review.

First, all manuscripts related to cycling near misses were gathered to date. These manuscripts included peer-reviewed journal articles, governmental and non-governmental reports, and conference proceedings. They covered the different aspects of cycling near misses, methods used, and the risk factors identified. These manuscripts can be accessed from four search engines (Scopus, Google Scholar, and Web of Science) via a combination of 'cycling' or 'road 'with keywords in a Boolean expression such as Cycling AND near AND miss*, cycling AND perceived AND risk*, perceived AND traffic and risk*, cycling AND near AND collision*, road AND conflict*, cycling AND risk and risk*. The total results for the specified combined terms are 556, 435, and 389 for google scholar, Web of Science, and Scopus, respectively. After removing duplicates, the sample size was reduced to 531. Second, records were first screened by title and abstracts, which reduced the records to 325. The second phase of screening included reviewing each manuscript, which reduced the number to 189 manuscripts that focus on the various types of conflicts between different road users, including near misses. At this phase, manuscripts were filtered to exclude studies that involved collisions without addressing near misses were excluded, except where they were required to draw a baseline or lesson learned that could be beneficial for near miss studies. Studies that involved safety policies that address cycling without addressing near misses are excluded. Last, we reduced studies to 19 manuscripts that focus only on cycling near misses which we focused on analysing in detail (See **Fig. 3.1**).

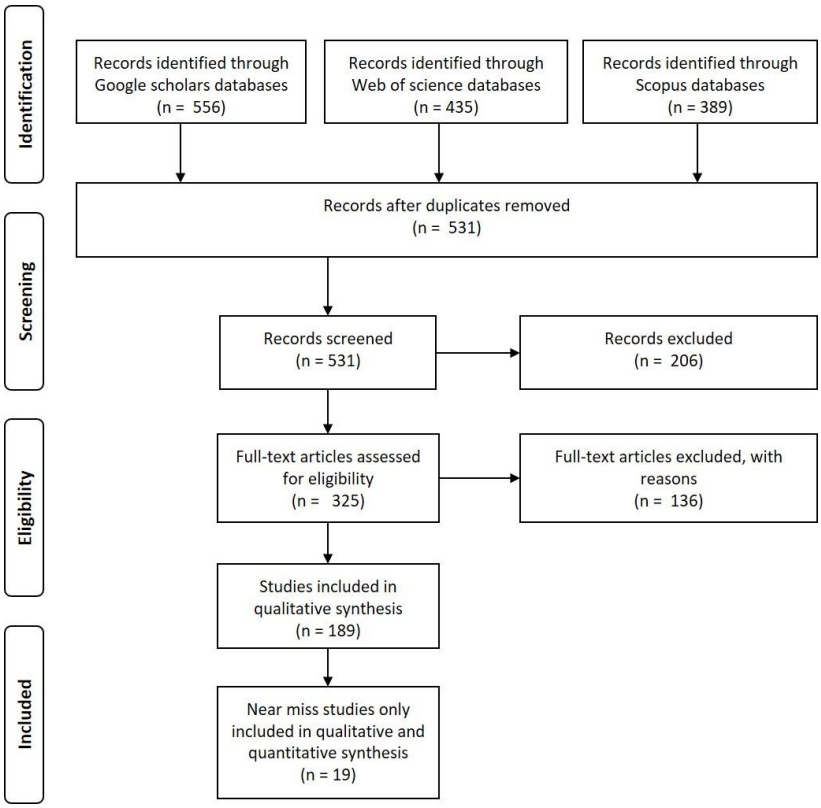

Figure 3.1: Flow chart for the screened records used for the systematic review

3.3 What is cycling near miss?

3.3.1 Definition

There are different conceptions and terms that pinpoint the subject of a 'narrowly avoided collision'. It can be defined as: 'perceived crash risk' (Chaurand and Delhomme, 2013; Strauss et al., 2013), 'perceived traffic risk' (Sanders, 2015), 'near collision' (Johnson et al., 2010), or simply 'near miss' (Aldred, 2016; Poulos et al., 2012). While a 'cycling near miss' is a subjective term that may differ based on individuals' experiences and their perceptions of risk, in most cases it is defined as a situation in which a person on a bike was required to act to avoid a crash, such as braking, speeding, swerving or stopping. In some cases, the definition may be extended to include those events that caused the person on the bike to feel unstable or unsafe, such as a close pass or tailgating.

3.3.2 Types of cycling near miss

Different studies have categorised near misses in different ways. For instance, some studies focus on the conflicts between people on bikes and drivers (Beck et al., 2019), or people on bikes and pedestrians (Paschalidis et al., 2016), in which different types of risky situations are categorised. Another approach is to categorise near misses depending on the type of conflict; either a moving object or stationary (Nelson et al., 2015). The most comprehensive categorisation to date was introduced by Aldred & Goodman (2018), based on an empirical study that included 2586 diaries from a sample of 396 participants over two years (2014-2015). This study summarised the types of cycling near misses were summarised in eight groups. These groups are: 1) a close pass, 2) a near left or right hook, 3) someone pulling in or out, 4) a near-dooring, 5) swerve around an obstruction, 6) pedestrian steps out, 7) someone approaching head-on, or 8) tailgating. Their study found that close pass near misses were the most frequent incident type.

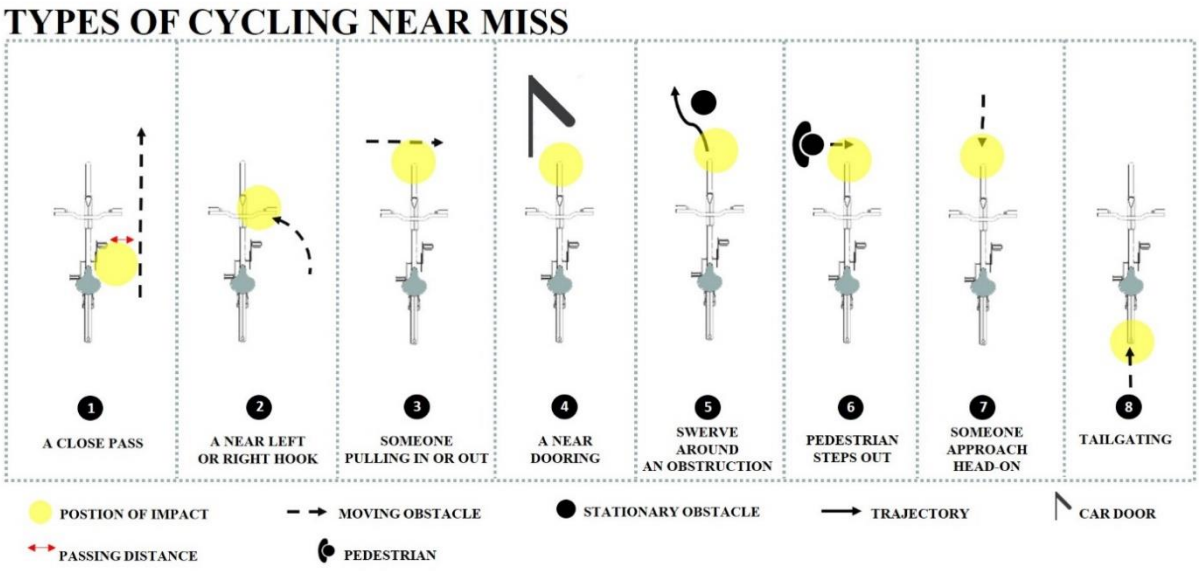

Figure 3.2: Types of cycling near misses and their potential impact

In **Fig. 3.2**, we aim to visualise the different types of cycling near misses addressed in the literature to better understand the road users or objects involved and their potential impact on people on bikes. In doing so, we aim to highlight the areas that need to be covered for any future method to be comprehensive when addressing cycling near misses.

3.4 Methods and materials used for understanding near misses

Studies have focused on different factors related to cycling near misses and have also used different data collection techniques and methods. These methods can be divided into three types of observational studies: 1) Observational studies relying on self-reported questionnaires, 2) Analysis of video of cyclist behaviour at specific sites, i.e. at an intersection, and 3) analysis of video from cameras used on their bike, the so-called naturalistic study.

3.4.1 Self-report studies

Self-report studies are the dominant design for most cycling near miss studies. In an observational study, data are gathered and statistically analysed to show associations and draw conclusions. There are three main ways in which data are collected for a self-reporting study: 1) using a self-reporting mechanism based on a questionnaire survey for a group of participants (Aldred and Crosweller, 2015; Chaurand and Delhomme, 2013; Fuller et al., 2013; Lawson et al., 2013; Paschalidis et al., 2016), or 2) using a self-reporting mechanism based on crowdsourcing platforms where data can be uploaded (Nelson et al., 2015; Poulos et al., 2012).

Data gathered based on crowdsourcing have led to significant progress in mapping cycling ridership and safety measures (Jestico et al., 2016; Nelson et al., 2015). However, while observational studies can offer insights about the behaviour of people on bikes over a longer period, the data gathered is limited by potential biases such as over or under-representation of certain cyclist groups or the types of risk factors, in addition to limitation and biases due to manual labelling and processing based on the collectors' interpretations (Dozza & Werneke, 2014).

3.4.2 Video analysis at specific sites

In a site observational study, video streams for a given context are used that highlight certain safety issues. A focus group of cyclists or non-cyclists participate and observe these video streams to evaluate behaviours. Rather than focussing on near miss events, these studies ask participants to evaluate the level of risk or presence of hazards in the video stream. Vansteenkiste et al. (2016) used site observation to develop a hazard perception test for children, finding that children's reactions to, and interpretations of, hazards are less developed than adults. Lehtonen et al., (2016) asked frequent and infrequent riders to watch video clips and rate risk through a caution estimate, finding that more frequent riders identified a higher number of caution estimate rises. This indicates that awareness of risk increases with rider experience. Such studies are important because they enable understanding of the differences in exposure to risk between certain groups of riders.

While site observations using cameras may overcome the limitation of interpretation found in self-report studies, the amount of data processing, specifically image processing, limits the scalability of this type of method. Additionally, multiple cameras are required at different positions to capture the entire environment and observe the dynamics of cycling behaviour and the interactions among the different agents (Dozza & Werneke, 2014). However, this approach can potentially enable the analysis

of near misses at scale by leveraging the supply of CCTV cameras installed on road networks, particularly in cities. Zangenehpour et al. (2016) used cameras to collect overview footage of intersections and analyse variations in vehicle-bicycle interactions in the presence/absence of cycle tracks. This work was facilitated by automated tracking of road users using a computer vision approach, which could be applied to CCTV feeds (Zangenehpour et al., 2015).

3.4.3 Naturalistic study

The naturalistic approach is often perceived as one of the most reliable methods for understanding road-user behaviours and analysing risk factors (Dozza et al., 2016a; Dozza and Werneke, 2014; Schleinitz et al., 2017). In this method, a group of participants ride instrumented bikes while carrying out their routine activities. Cameras and sensors are fitted to the bike or rider that collect data related to riding behaviour, the surrounding environment and the interactions with other road-users. The types of data gathered vary depending on the purpose of the research and the installed equipment and sensors. The first example of a naturalistic study was in Melbourne, Australia, where riders were given helmet-mounted cameras (Johnson et al., 2010). This camera position was chosen because it gave the rider's perspective and enabled their behaviour to be analysed (e.g. shoulder checks). While this approach enables risk factors to be analysed through manual analysis of the video data, quantitative features such as rider speed, geolocation and acceleration/deceleration cannot be extracted. More recent studies have used instrumented bikes containing at least a video camera and a GPS device (Gustafsson and Archer, 2013). More sophisticated setups can contain units for inertial measurement, a signal button to record critical incidents and near misses, brake force sensors (Dozza & Werneke, 2014), or even LiDAR sensors for measuring the range of nearby objects such as passing vehicles (Beck et al., 2019). Naturalistic approaches have also been used to study differences in behaviour between pedal bike and e-bike users, which is an emerging area of concern for policymakers (Schleinitz et al., 2017).

The main advantage of the naturalistic approach is that detailed information is collected about the behaviours and actions of agents involved in an event, as well as the instantaneous features of the environment. By definition, the naturalistic approach also collects data in which no event occurred, which can be used as counterfactual events in a case-control framework. However, processing data collected in naturalistic studies is labour intensive and automated methods are required if such data are to be collected at scale.

3.5 Factors related to near misses and their impacts

Different methods have different advantages and limitations when it comes to including and analysing the wide range of risk factors. In general, cycling near misses are transport-related, however, their risk factors may or may not be transport-related (Aldred, 2016; Beck et al., 2016; De Rome et al., 2014; Vanparijs et al., 2015). This creates challenges for methods currently used in the literature – either in extracting these factors or analysing them - to assess in a single study. Based on the literature, we categorise these factors into aspects related to visibility, physical conditions of the built environment, interaction among different agents (e.g. people or animals), and behavioural and psychological factors related to the cyclist.

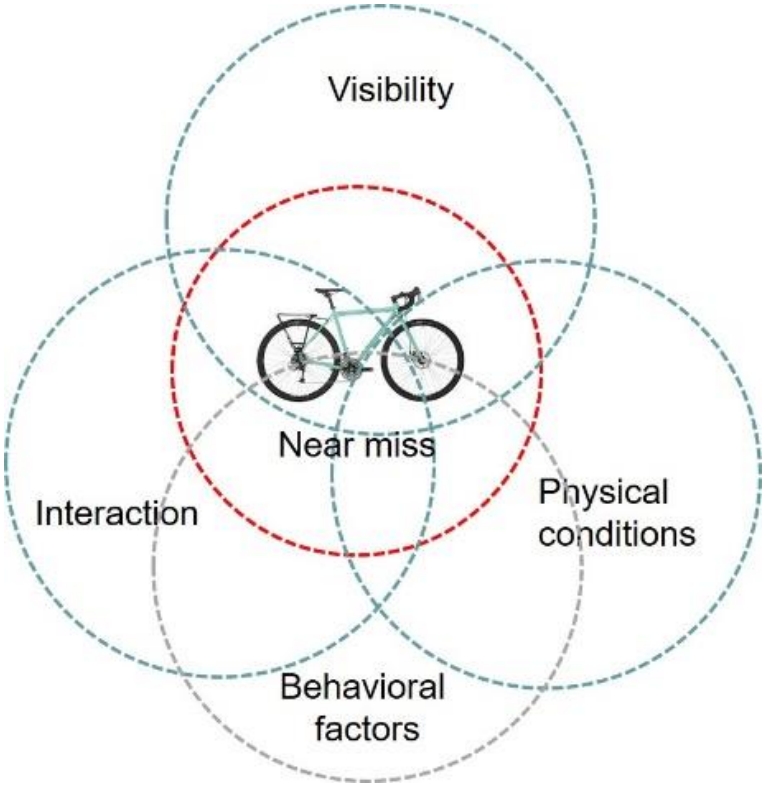

Figure 3.3: Types of factors related to cycling near misses

Fig. 3.3 shows the interaction of these four categories that could lead to a near miss. Most of the factors belong to behavioural, physical, or visibility related factors. In many cases, near misses occur as a result of a combination of several independent factors represented by the 'interaction' section of the diagram.

Many studies focus on the impacts of one of the risk factors for a near miss while excluding other factors. Poulos et al. (2012) analysed near misses based on the type of cycling infrastructure, including pedestrian footpath, shared path, road with no bicycle lane, bicycle path, and road with a bicycle lane. Johnson et al. (2013) studied collision and near-collision characteristics based on people on bikes and open vehicle doors, highlighting how an open vehicle door could lead to frequent and serious injuries that could sometimes be fatal. Few studies, however, explore the holistic nature of the factors related to near misses.

Table 3.1 summarises the 19 studies identified in the literature in terms of methods used, near miss type and risk factors covered, broken down by the categories shown in figure 3. The categories are further subdivided according to specific risk factors. The interaction risk factors are combined with the near miss type in column three. The extent to which each risk factor has been analysed in the literature by each method is discussed in sections 5.1 to 5.4, below.

Table 3.1: Current near miss literature and covered scope and risk factors

NEAR MISS STUDY	METHOD USED	TYPES OF NEAR MISS OR SCOPE OF THE STUDY	RISK FACTORS COVERED																												
			Visibility											Physical conditions					Behavioural factors													
			Weather conditions					Light conditions and time of the day						Built environment conditions					Cyclist characteristics					Safety equipment							
			Clear	Cloud	Rain	Snow	Fog	Dusk/	Day-time	Night-lighting conditions				Road	Road	Existence		Parked cars	Intersection	Age	Gender	Experience	Trip	Helmet	Hi-vis	Bike light						
(Paschalidis et al., 2016)	Self-report studies	Conflicts with cars and pedestrians – 8 types						-		-					-	x		-	x	x	x	x	x	-	-	-					
(Vansteenkiste et al., 2016)	site observational analysis	Visual awareness, environmental awareness, risk perception, and reaction time						-		-				x	-	-		-	x	x	-	-	-	-	-	-					
(Gustafsson and Archer, 2013)	Naturalistic study	All types of conflicts are divided into 17 types						x		-					-	x		-	x	x	x	-	x	-	-	-					
(Nelson et al., 2015)	Self-report studies	Near miss with a stationary or moving object or vehicle						x		x					x	-		x	-	x	x	-	-	x	-	x					
(Branion-Calles et al., 2017)	Self-report studies / comparative study	Near miss with a stationary or moving object or vehicle						x		x					x	x		x	x	x	x	-	x	x	-	x					
(Schleinitz et al., 2015)	Naturalistic study	Conflict with vehicles, cyclists and pedestrians						x		-					x	x		-	-	x	-	-	-	-	-	-					
(Aldred, 2016)	Self-report studies/ discussion	Eight types of near misses						-		-					x	x		x	x	x	x	-	x	-	-	-					
(Chaurand and Delhomme, 2013)	Meta-analysis/ Comparative study	Bike-car interaction at road intersections						-		-					-	-		-	x	x	x	x	x	x	-	-					
(Johnson et al., 2013)	Naturalistic study	One type of near misses- door opening						-		-					-	x		x	-	x	x	-	x	-	-	-					

(Lehtonen et al., 2016)	Site observational study	Risk perception of bicyclists in a city environment		-	-		-	x	-	-	x	x	x	x	-	-	-				
(Poulos et al., 2012)	Self-report studies	Near misses- undefined types		-	-		-	x	-	-	x	x	x	x	-	-	-				
(Dozza et al., 2012)	Naturalistic study	A pilot study showing six unique risky events.		-	-		-	-	-	-	-	-	-	-	-	-					
(Dozza and Werneke, 2014)	Naturalistic study	Critical events - 44		-	-		x	x	-	x	x	x		x	-	-	-				
(Aldred and Crosweller, 2015)	Self-report studies	Near miss types – 8 types		x	-		-	-	-		x	x	-	x	-	-	-				
(Sanders, 2015)	Self-report studies	exploration of perceptions of traffic risk between cyclists and drivers/ 4types of near misses		-	-			x	x	-	x	x	x	x	-	-	-				
(Johnson et al., 2010)	Naturalistic study	Collision/ near collision for on-road cyclists		-	x		-	x	-	x	x	x		x	x	-	-				
(Lawson et al., 2013)	Self-report studies	Perception of safety for cyclists		-	-		x	x	x	x	x	x	x	x	x	x					-
(Aldred and Goodman, 2018)	Self-report studies	Near miss types- 8		-	-		-	-	-	-	x	x	x	-	-	-	-				
(Fuller et al., 2013)	Self-report studies	the impact of implementing a public bicycle share program on events of near misses		-	-		-	-	-	-	x	x	x	-	x	-	-				

3.5.1 Behavioural aspects

Behavioural aspects play an important role in defining the various cycling styles that may influence the types and the frequency of near miss events. They can be subcategorised into three groups: 1) Individual characteristics, 2) trip characteristics, and 3) safety measures.

All of the 19 reviewed studies cover individual characteristics to a greater or lesser extent. First, most of the cycling near miss studies focus on covering multiple factors related to socio-demographic aspects of people on bikes. It is noticeable from **table 3.1** that self-report studies generally collect more demographic data than site observational and naturalistic studies, which is due to their design.

Safety measures related to individual behaviours are crucial for avoiding near misses. Johnson et al. (2014) studied the behaviours and perceptions of both drivers and people on bikes towards cycling safety. They found that drivers who also cycle are more likely to have a positive attitude towards cycling. They also highlighted the importance of rethinking driver education when overtaking people on bikes; considering head checks, buffer space, and adequate indications. Walker et al. (2014) found that the appearance of the person and the type of outfit they wore has an insignificant effect on the clearance distance drivers gave when overtaking people on bikes, contrary, police/video-recording jacket is the only outfit that has a significant correlation with the passing proximities. Nelson et al. (2015) found that the frequency of near misses was higher when cycling without bike lights than when using front and backlights.

3.5.2 Physical conditions

Fifteen of the 19 studies referred to physical conditions. While there are different factors related to the built environment that influence the choice of cycling routes from a behavioural perspective (Broach et al., 2012), these are also factors that cause potential risk for encountering collisions or a near miss (Cho et al., 2009). As Aldred notes: *"The vast majority of near misses were judged potentially preventable by changes to road user behaviour and/or the cycling environment*'' (Aldred, 2016, p. 78). Therefore, physical conditions play an important role in either experiencing a near miss or avoiding one. Based on the literature review, we have subcategorised these into four groups: 1) Infrastructure, 2) surface conditions, 3) location and 4) surrounding context.

In terms of infrastructure, Parkin and Meyers (2010) found that in the presence of a cycle lane, drivers may drive within their marked lane with less consideration of ensuring a comfortable passing distance for people on bikes in the adjacent cycle lane. This has been confirmed in a recent study by Beck et al. (2019) on close pass events.

Several studies include factors related to surface conditions, such as wet, dry, well-maintained, or deteriorated surfaces (Aldred, 2016; Branion-Calles et al., 2017; Gustafsson and Archer, 2013; Nelson et al., 2015; Schleinitz et al., 2015). Dozza et al. (2012) found that some near misses occurred when the condition was icy. Nelson et al. (2015) found that the near misses mostly took place on dry surfaces (64.6% of the total near misses) with no parking on the road (60.1% of the total near misses). However, these frequencies are based on the count of responses rather than the significance of the result, which may be due to the self-selection of either trip routes or time. The challenge remains in understanding how the individual factors combine with physical conditions to produce risk.

Different studies have focused on the location of cycling, highlighting a higher exposure to near misses at intersections (Branion-Calles et al., 2017). Strauss et al. (2013) analysed 650 intersections in Montreal in which cyclists were exposed to injury incidents. They found that more cyclists tended to suffer injuries at junctions but with a lower injury rate due to the non-linear correlation between injury occurrence and bicycle volume. They also highlighted that the frequency of cycling crashes is

associated with the changes in the flows of both vehicles and bicycles based on injury data of people on bikes between 2003 and 2008 in Montreal, Canada. Additionally, crashes were more likely to happen at intersections that include a bus stop. The important role of the built environment was underscored, in which a change in conditions (i.e. presence of cycle lane, land use mix, presence of a school, etc.) is more likely to cause a direct impact on cyclist activity and safety.

There is still an absence of near miss studies that directly investigate the impact of the surrounding context that may densify the flow of traffic for cyclists, pedestrians, and vehicles in a certain location in cities, which may cause potential risk exposure for the people on bikes. This has been outlined by Vanparijs et al. (2015) in their review of studies related to exposure measurement. They reviewed the different methods that measure cycling exposure to incidents, including time, distance, and trips as exposure units. They showed that the lack of exposure data hinders the ability to draw significant conclusions. They also highlighted that this data often neglects minor incidents, subsequently near miss events are more likely to be missing as well, which makes it difficult to understand safety levels between different types of infrastructure, and the age categories of people on bikes.

3.5.3 Visibility-related conditions

Visibility related conditions play a role in crashes and near misses (Lacherez et al., 2013), particularly, factors related to 1) the time of the day, 2) weather conditions, and 3) the level of illumination, including the existence of glare. Of the reviewed studies, only two cover weather conditions, while nine cover lighting conditions or time of day. (Branion-Calles et al., 2017) use secondary data sources to infer weather and lighting conditions, using the time of sunrise/sunset and meteorological data at a single location in the two cities studied. This may fail to capture local variations in weather, street lighting and glare caused by the direction of travel. (Dozza et al., 2012) highlights some factors related to six unique events but does not analyse risk factors.

There are a limited number of studies that included factors related to weather conditions to address their impact on the occurrence of near miss events. Branion-Calles et al. (2017) addressed weather conditions such as clear, cloudy, rain, and fog in analysing near misses. Based on descriptive analysis, they found that a higher frequency of near misses and crashes are reported when the weather is rainy, snowy or foggy compared with cloudy or clear conditions.

While the time of the day could be a significant risk factor in cycling crashes (Johnson et al., 2013), many studies neglect the issue of time completely (Aldred, 2016; Aldred and Goodman, 2018; Chaurand and Delhomme, 2013; Fuller et al., 2013; Lawson et al., 2013; Lehtonen et al., 2016; Paschalidis et al., 2016; Poulos et al., 2012; Sanders, 2015; Vansteenkiste et al., 2016). Other studies use a binary classification of day and night, without considering more nuanced effects on the lighting conditions such as those caused by direct sunlight at dawn and dusk. For instance, Branion-Calles et al. (2017) studied near misses according to the time of either peak hours or off-peak hours. Gustafsson and Archer (2013) categorised time to ‘morning (06:30-9:30)’ and ‘afternoon (15:30-18:30)’, highlighting that more incidents and safety issues occur in the morning time. These nuances are important and have been recognised by recent studies analysing the impact of the built environment and its influence on near misses. For instance, Aldred and Crosweller (2015) studied incidents and

exposure by the time of the day including am and pm peaks, in which they found that the numbers of trips and subsequently the incidents are considered to be low after 12 am till 6 am.

The impact of the lighting conditions, including glare, on near miss events, remains under-investigated. Branion-Calles et al. (2017) mentioned lighting conditions as risk factors represented in two classes of day or night-time only. Dozza et al. (2012) mentioned the effect of glare on the quality of the video streams, without studying its impact on the frequencies of near misses. Even though Nelson et al. (2015) had looked at glare based on self-reporting data, this data only provides a subjective account of the incident and glare may have been present but not mentioned. Therefore the data may not represent a reliable source of information to study the impact of glare.

3.5.4 Interaction between road-users

While near misses can involve a single individual riding a bike, they are often the result of interactions between the rider and other road-users, such as people in cars, people driving or other people on bikes. During a journey, a bike rider will have many safe interactions with other road-users, which makes it difficult to define those situations that lead to elevated risk. Therefore, quantitative studies in this area have tended to focus on a single type of interaction. A notable example is (Beck et al., 2019), who focussed on passing distance. Their study used a distance sensor attached to a bicycle to measure the range of passing vehicles. Using a passing distance of 1 metre at <= 60km/h (1.5m at >= 60km/h) informed by Australian legislation, they identified that 1 in 17 passing events was a close pass. As mentioned in section 5.2, they found that the presence of a bike lane was associated with a closer passing distance. This type of study is important in identifying a particular type of risky behaviour, but the use of a 1 metre passing distance is somewhat arbitrary. For example, in the UK, Operation Close Pass uses 1.5 metres as a safe passing distance; if this figure was used in (Beck et al., 2019) it would change the interpretation of results. In general, there is no good evidence on what is safe under what circumstances, and as the authors, note: *"It is important to understand how cyclists' subjective experiences align with quantified passing distances"* (Beck et al., 2019, p. 259).

Some studies focused on a specific type of interaction that may result in a near miss if the safety measures are not considered properly. For instance, Dozza et al. (2016) provided an in-depth analysis of how drivers overtake bicycles during passing events. They found that manoeuvres, especially on rural roads, are often more critical since they happen at higher speeds (approx. 70km/h) with less time to avoid collisions (less than 2s) if critical or unforeseen events took place.

3.5.5 Combined factors

The various permutations of factors described above indicate the complexity of understanding how risk factors interact to cause near misses. **Fig. 3.4** shows a dendrogram of the different factors that may be involved. Factors highlighted in black are those identified from the reviewed studies, while those in blue are additional factors that could be considered. The number of potential factors illustrates that, theoretically, even if one of the individual factors is not statistically significant from a linear perspective, it may influence the occurrence of a near miss from a nonlinear perspective. Consequently, due to the potential of the existence of nonlinearity in near-miss research, the types of methods used to conduct near miss research may vary depending on which factors are addressed and how the data are collected.

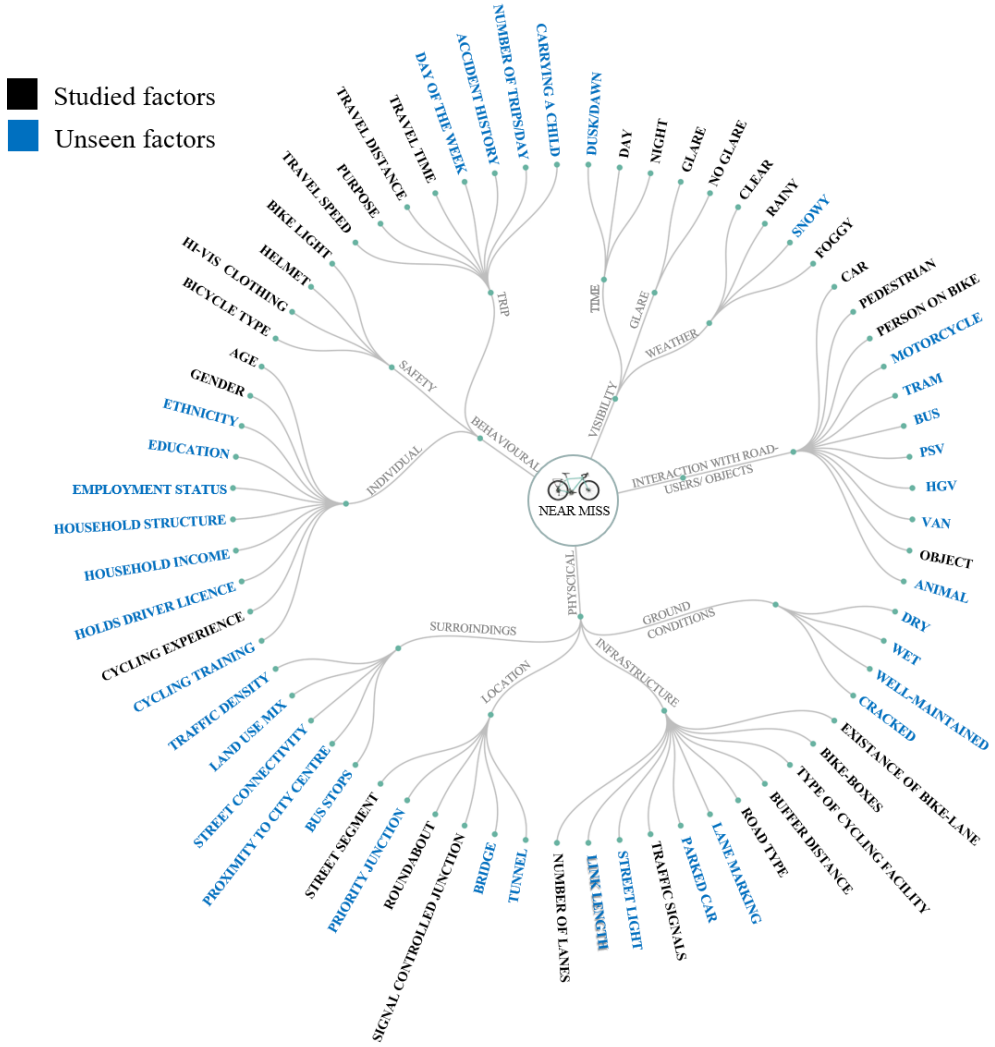

Figure 3.4: Risk factors related to cycling near misses

Table 3.2 summarises the risk factors covered in the 19 studies, broken down by method, where the number in each cell is the count of studies of that type that refer to that risk factor. In general, the built environment and cyclist characteristics are covered by more studies because they are static or slowly changing variables. It should be noted that, while the same factors are covered by many studies, the form and quality of the data can be very different. For example, a self-report study may include a question on the presence/absence of a cycle lane, while a naturalistic study using video will allow interpretation of the type of cycle lane, the road condition and the surrounding context, albeit usually with expert interpretation.

Dynamic variables such as lighting and weather conditions are more difficult to capture using each of the methods, but for different reasons. In self-reporting studies, these variables require recall and may be subjective, which means that they are usually incorporated in simple terms such as day/night time or rain/no rain. Naturalistic studies and site video analysis can capture more nuanced factors but they require labelling of video data, which is usually manual and time-consuming.

Table 3.2: summary of near miss studies

Method (Count)	Weather conditions					Light conditions and time of the day				Built environment conditions					Cyclist characteristics				Safety equipment		
	Clear	Cloud	Rain	Snow	Fog	Dusk/ Dawn	Day-time	Night-time	Lighting conditions	Road types	Road conditions	Existence of bicycle	Parked cars	Intersection	Age	Gender	Experience	Trip purpose	Helmet	Hi-vis clothing	Bike light
Self-reporting (10)	1	1	1	0	1	3	3	3	2	6	4	6	5	4	10	10	6	7	4	1	2
Site Video Analysis (2)	0	0	0	0	0	0	0	0	0	1	0	1	0	1	2	1	1	1	-	-	-
Naturalistic (5)	1	0	1	0	0	2	5	4	2	2	3	5	1	3	6	4	0	4	1	0	0
Total	**2**	**1**	**2**	**0**	**1**	**5**	**8**	**7**	**4**	**9**	**7**	**12**	**6**	**8**	**18**	**15**	**7**	**12**	**5**	**1**	**2**

A combination of naturalistic studies with self-reporting has the potential to capture the broadest range of information, while site video analysis is severely limited in its ability to capture demographic factors. It is also important to note that, while self-reporting captures the characteristics of the rider, it cannot easily capture the characteristics of the other road users involved. In the next section, a detailed discussion of the limitations of current methods and gaps in the literature is presented.

3.6 Gaps in the literature

There are methodological challenges for collecting and analysing road safety data and its risk factors (Schlögl and Stütz, 2017). There are several limitations in the existing methods used to understand near misses. These limitations are: 1) How to eliminate factors due to manual labelling of data, 2) The current lack of functionality of sensors, 3) small data samples, 4) limited scope of studies, 5) The absence of a unified method for understanding near miss, which leads to 6) lack of understanding of the impact of the risk factors.

3.6.1 Elimination of factors due to manual labelling

One of the crucial drawbacks in current studies is that the analysis of sensor data has been dependent on manual labelling, which is highly time-consuming and may also introduce some bias in how data sets are labelled, limiting the transferability of findings to a different context.

3.6.2 Limitations of sensor data

The current sensors used have not been developed to measure the range of possible factors that might influence near misses. In particular, current approaches mean that it is difficult to compare sensor data from one location to another, especially where the environmental context can be very different (e.g. urban vs rural, hot vs cold climate, etc.).

3.6.3 Limitation of data sample size

Current studies have involved small data sets making it difficult to draw statistically significant conclusions, which again limits the ability to draw conclusions relevant across a range of contexts. Even though the naturalistic approach shows progress in collecting rich data on the context and factors related to near misses, the current approach of labelling data manually reduces its potential for large scale implementation without automated data processing.

3.6.4 Limitation of the study scope

Most studies focus either on certain types of near misses, i.e. passing events, with a wide range of risk factors (Beck et al., 2019), or on a range of near-miss types with a limited number of analysed risk factors (Aldred, 2016; Aldred and Crosweller, 2015).

3.6.5 The absence of a unified framework for understanding near misses

Previous studies of cycling near misses have lacked a method for understanding near misses regardless of the context, the types of near misses, or the related risk factors associated with these incidents. Overcoming this limitation may lead to a deeper understanding of near misses and for drawing transferable guidelines.

3.6.6 Limitation in understanding the impact of the risk factors

Understanding the causality and effect of the given risk factors on near misses in a Bayesian approach requires the variables to be random and controlled via unbiased variables that show a direct effect on both risk factors and near misses. Given the limitation in extracting a wide range of risk factors with a large sample size as aforementioned, we cannot currently quantify and assess the impact of risk factors in an unbiased and systematic way.

3.7 The potential for computer vision to recognise near misses and their risk factors

Cycling near misses can be viewed in the wider context of the place and time in which they occur. Our knowledge about near misses and their related risk factors could be built through video streams that may provide a very effective approach to identifying patterns related to their occurrence. In general, the success of artificial intelligence and computer vision in pattern recognition in the last decade, (LeCun et al., 2015) has added a new dimension towards understanding cities generally. Computer vision has the potential to understand cycling near misses by extracting safety-related features from still or multi-frame images in complex daily life scenes. However, we argue that near miss studies should not focus solely on extracting known risk factors. All the different layers of cities (the built environment, human interaction, transportation and traffic, the natural environment and infrastructure) can have potential impacts on the experience of a rider and should not be discounted (as explained in chapter II). Using the different algorithms of computer vision, coupled with deep

learning, we can extract and analyse these features to develop automated systems that can be applied to multiple tasks, functioning at different scales. This will help draw significant conclusions and assist the process of policy-making.

To develop an autonomous and multi-tasking system to detect near miss scenes, their types, and the associated risk factors, seven steps need to be considered: 1) Sensing and classifying the physical environment, 2) detecting objects and obstacles, 3) inferring distance and detecting safety measures, 4) recognising motion, 5) recognising actions and inferring behaviours, 6) inferring individual characteristics. Such a system would make use of a range of computer vision techniques, such as image classification, segmentation (classifying an image at a pixel level), object detection, action recognition, scene awareness and understanding the underlying gist of a scene. By embedding these technologies within sensors, this approach would move beyond naturalistic studies towards automatic quantification and analysis of risk, and 7) Integrating all algorithms, which could allow causal inference.

Fig. 3.5 shows how deep learning and computer vision algorithms can be integrated to identify: 1) The risk factors, 2) near miss scenes, 3) types of near misses and 4) the impact of the different risk factors on the different types of near misses. We discuss each of the components in turn in the following subsections.

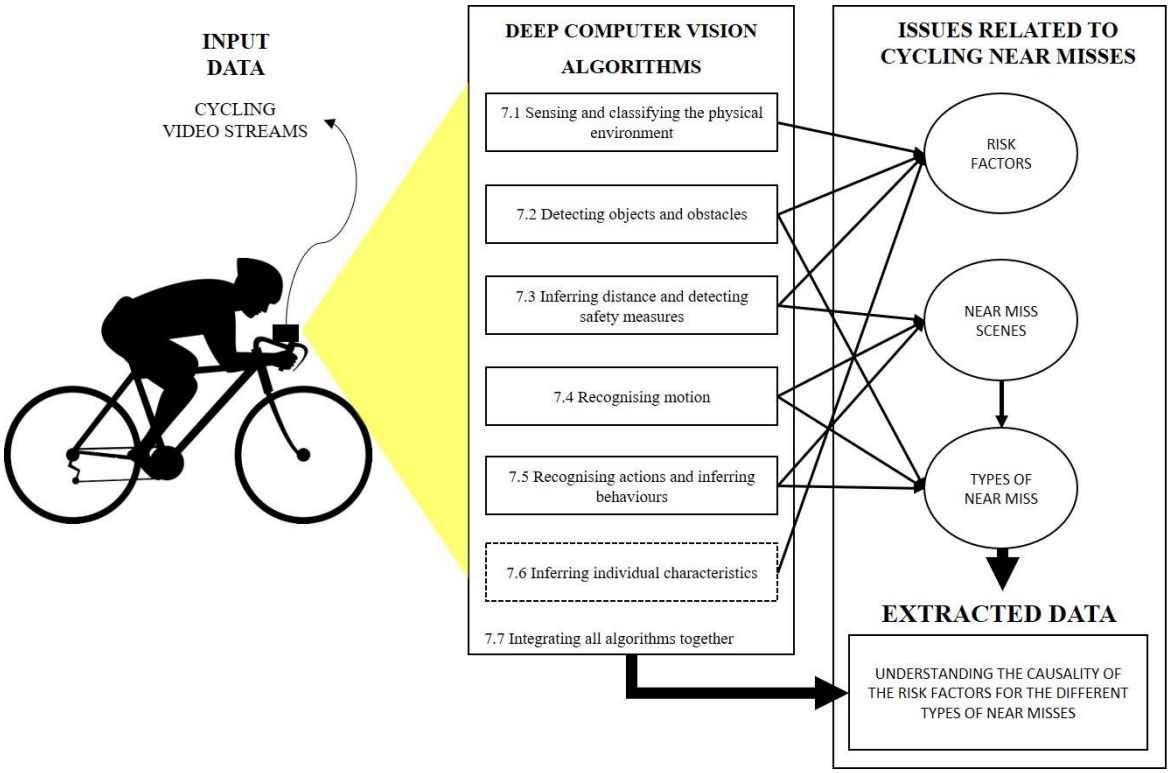

Figure 3.5: Conceptual framework for an embedded AI system to understand cycling near miss

3.7.1 Sensing and classifying the physical environment

As explained in chapter 2, CNN models make it possible to tackle risk factors related to cycling near misses, particularly those related to the physical environment and the visual conditions. ResNet (He et al., 2016) has shown substantial progress in recognising thousands of objects and by far represents the state-of-the-art in terms of scalable deep models for classification tasks. Using This method, different models can be developed to extract risk factors related to cycling near misses. These models can be capable of detecting weather conditions and time of day and understanding the overall deterioration of the environment and surface conditions. Given sufficient training data, bespoke models can be built to detect the different dynamics of the physical environment.

3.7.2 Detecting objects and obstacles

Detecting road users or objects, such as a car door, is indispensable for understanding the risk factors related to cycling near misses and for classifying the different types of near misses as previously discussed (Section 3.2). Localisation is the process of identifying multiple objects in a single frame and is also applicable to video streams. These models use a single deep model in an end-to-end fashion to localise different objects with a given confidence. Similar to classification, different models of different convolutional structures have been developed to segment and localise objects in a single frame of an image such as You Only Look Once (YOLO) (Redmon and Farhadi, 2017), and MultiBox Detectors (SSD) (W. Liu et al., 2016). Relying on this type of algorithm, extracting and mapping pedestrian and transport modes from complex urban settings can be achieved to recognise the different agents. This is the first step in understanding the interaction of people on bikes with other road users.

3.7.3 Inferring distance and detecting safety measures

Understanding what is a safe or unsafe distance when passing or overtaking a bicycle is crucial for detecting near miss scenes and identifying their types. CNN based computer vision techniques can be trained to infer the distance of objects from the camera just by looking at a still image or a video stream. For instance, Cao et al., (2018) trained deep CNN models to estimate the depth in a single image by labelling the different depths on the image. Also, He, Wang, & Hu (2018) trained a deep CNN model to estimate the depth of a monocular image. In the context of near misses, CNN algorithms can be trained with images labelled using Light Detection and Ranging (LiDAR), or ultrasonic range sensors. By utilising the depth estimation algorithms, a standard buffer distance can be measured and detected from cycling video streams, which would contribute to understanding the different types of near misses.

Furthermore, it is foreseeable that once trained, these algorithms will no longer require the range sensors and can be applied to video streams collected in isolation. This opens up near miss study to the vast amounts of data that are routinely collected by riders who use action cameras.

3.7.4 Recognising motion

Understanding the overall motions of the different objects in the scene is another key role in understanding scenes of near misses and identifying their types. Various computer vision models rely on estimating the change in motion for a sequential frame of images, or so-called optical flow (Alvarez et al., 2007; Andrade et al., 2006; Ayvaci et al., 2012; Baker et al., 2011; Butler et al., 2012; Enkelmann, n.d.; Mallot et al., 1991; Sun et al., 2010).

Estimating optical flow from cycling video streams would enable the differentiation of the moving objects from the stationary part of the scene, in addition to understanding obstacles and occlusion. Thus, it would allow a better understanding of such instant actions as near misses.

3.7.5 Recognising actions and inferring behaviours

Moving from tracking the motion of objects in a complex scene towards classifying multiple actions while tracking, deep models also have shown continuous progress (Bilen et al., 2016; Wang et al., 2015; B. Zhang et al., 2016). In fact, deep computer vision models have been successful in understanding human behaviours based on the poses of human skeletons and how they interact with other objects in a complex environment. While the issue of detecting cycling near misses from moving bicycles in real-world settings has not been addressed in the literature, there is a well-established body of knowledge on action recognition from video streams, as explained **in Chapter II**. Action recognition using computer vision typically involves two steps: 1) extracting and encoding features and 2) classifying features into action classes. By utilising action-recognition models to detect the overall motion, the unsafe riding scenes and near misses can be automatically detected and categorised. There are variants of action recognition models that focus mainly on understanding human activities rather than the overall perception of the interaction between different agents or the clue of the scene in the case of the stated issue of near misses.

In summary, not only do the architecture of action recognition models vary, but also the training process and the data fusion approach. Some models have been trained in an end-to-end network, whereas others are designed and trained in a two-stream network with an early or late-stage fusion of data types (RGB frames, optical flow data, etc.). While complex model structures have yielded higher accuracies in given tasks, specifically, the two-streams network, these differences have consequences on the trade-off between model accuracy, complexity, and time needed for inference.

3.7.6 Inferring individual characteristics

Computer vision coupled with deep learning shows good potential for extracting information related to individual characteristics (as mentioned in Section 5.1). However, this topic remains the most significant bottleneck in achieving the system outlined in figure 5. While it is feasible to collect personal characteristics of people using an embedded AI device, inferring the characteristics of those they come into contact with is more of a challenge and is a common issue for all near-miss study methodologies. Various deep models have shown good potential in extracting information in this regard, such as recognising gender (Levi and Hassncer, 2015; Narang and Bourlai, 2016), age (Levi and Hassncer, 2015), facial emotions (Minaee et al., 2021), and ethnicity (Narang and Bourlai, 2016). Building on these models could be beneficial for understanding the various characteristics of people on bikes and other road-users that would help to understand the impacts of near misses, along with their different types and the other risk factors.

3.7.7 Integrating all algorithms

There are different approaches for integrating different tasks, which depend on the availability of multi-label data, the ability of fusing data of different input parameters, or the availability of computational resources. Multitask and ensemble learning are two crucial approaches for learning multiple tasks (Goodfellow et al., 2017). Multitask learning refers to simultaneous training of several

tasks of the same input, in which tasks can share intermediate-level representation in some shared layers. This approach aims to improve generalisation by pooling the examples outputted by several tasks (Goodfellow et al., 2017). On the other hand, ensemble learning refers to combining multiple models to solve a given problem. There are different purposes of ensemble learning, most commonly, the bootstrap aggregating (or bagging) technique (Goodfellow et al., 2017). In this approach, several models are trained differently for a given task and combined to reduce generalisation error. Ensemble learning, however, is also used for other purposes such as data fusion or incremental learning (Parikh and Polikar, 2007). Certain problems can be too difficult for a given classifier to solve or too computationally expensive to conduct, in which case the divide-and-conquer approach can be utilised through incremental learning. Accordingly, ensemble learning seems suited to the diversity of computational tasks required to recognise cycling near misses and their risk factors. Different tasks can be learned by the representation of the input incrementally. This approach will allow flexibility in how the input data can be used and organised for each given task and minimise the computational requirements of training several models for various tasks at once. It would also allow modification and further development, at a later stage, of any given model without affecting the other assembled tasks.

By putting all algorithms together, a rich data set can be generated that includes information related to risk factors and the frequency and type of near misses. This would respond to the current knowledge gap in the applied methods used for studying cycling near misses. Moreover, regression models can be used to understand the causality and the impact of the wider range of extracted risk factors associated with the different types of near misses, leading to robust conclusions being drawn that could be more effective for both people on bikes and policy-makers.

3.8 Summary

In this chapter, we reviewed the literature on cycling near misses and demonstrated that because many near misses are a result of a combination of different factors that may not be transport-related, the current approaches to tackling these factors are not adequate to fully understand the genesis of a near miss. Here, we explore the potential of extracting data of different disciplines using a unified input (images/videos) that relies on computer vision methods to automatically extract the wide spectrum of risk factors that may cause potential risk for people on bikes.

Different studies have focused on analysing the perceived risk related to the various types of near misses, in which particular progress has been achieved related to understanding the most frequent types of near miss - close passes - when people on bikes are over-taken in unsafe or uncomfortable manoeuvres by other road-users. There are various challenges related to the current methods used to collect, analyse, and produce evidence that could assist policy-making towards minimising risky situations in cities. In this chapter, we reviewed the different studies, highlighting the methods and scope are to underline the knowledge gap in which further work is needed.

While the approach of the naturalistic study seems to be promising in understanding instant and risky situations such as near misses, the current methods related to collecting and analysing video streams remain a challenge for drawing significant conclusions. In the following chapters, we propose a framework based on artificial intelligence, relying specifically on the domain of computer vision, to overcome the current limitations and move towards a unified method of understanding near misses. Our conceptual framework explains how different deep models can be trained and utilised to reach

an embedded AI system that automates the detection of near miss scenes, analyses their types, their risk factors, and draws significant conclusions based on the causality of risk factors, behaviours of people on bikes, and their mutual interaction with other road-users.

4
OVERALL METHODOLOGY

4.1 Overview

This chapter presents the methodological framework for a system that makes use of computer vision to understand when, where, and why cycling near misses occur in cities. The main aims are threefold; 1) to detect the wide spectrum of risk factors that may be associated with or cause near misses, 2) to distinguish near miss scenes from safe scenes, 3) to understand the effect of the identified risk factors on cycling near misses. The overall research methodology consists of four phases, each of which develops a method to addresses part of the stated aims. Each method is independent of the others and works on a specific task. However, the overall methodology provides an integrated framework that functions as a pipeline for detecting various road-users and events in cities (people, transport modes, weather conditions, built environment conditions, action recognition) relying on deep learning and computer vision. This chapter also introduces a framework stringency index that aims to evaluate the overall performance of the proposed framework for a given task, such as detecting cycling near misses.

4.2 Proposed framework

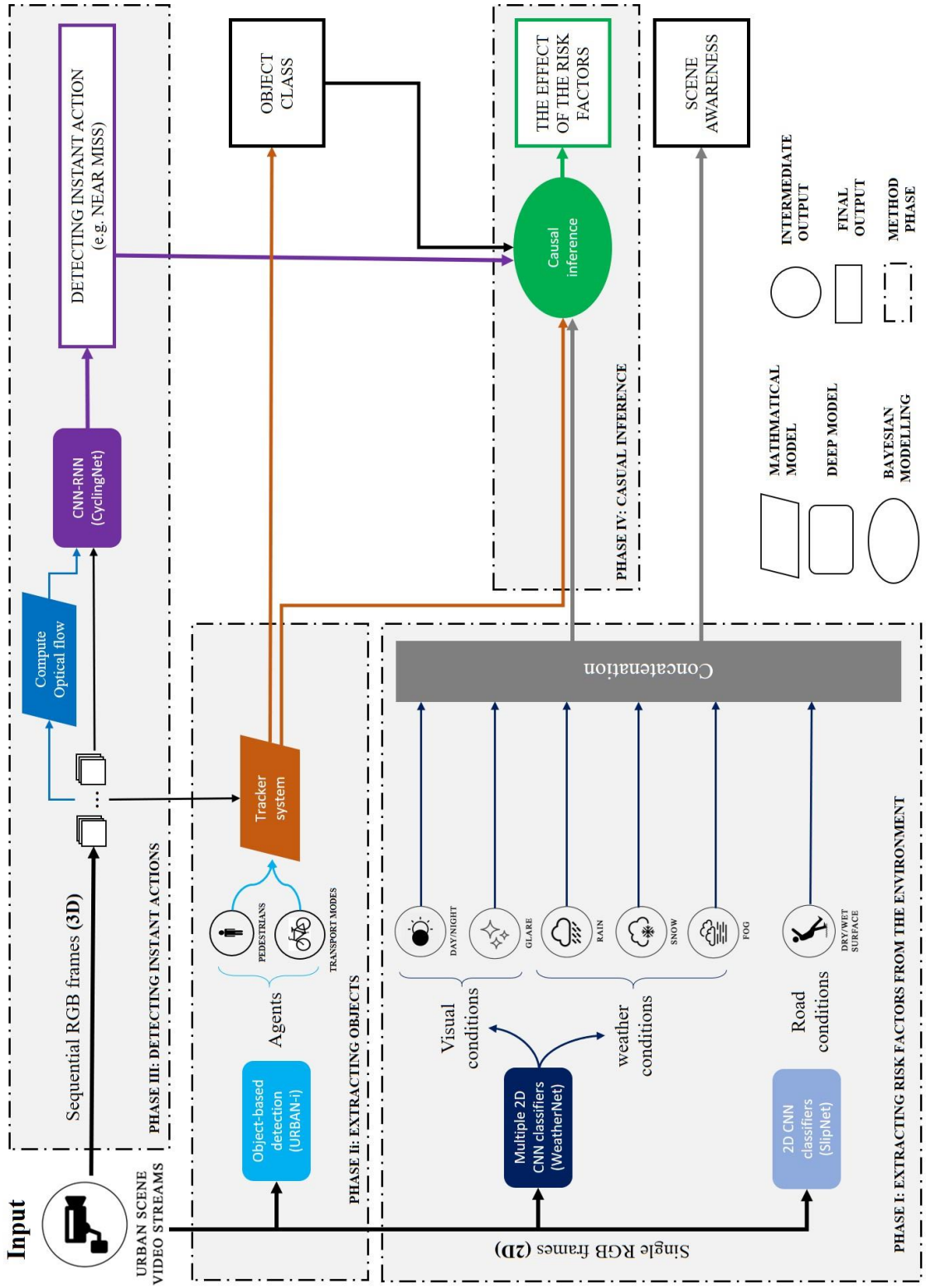

Figure 4.1: Research overall methodology

The framework is built based on ensemble learning (as discussed in section 3.7.7) with a single input of video streams. The framework outputs four outcomes: 1) critical event detection (in this case, near misses), 2) a list of detected risk factors and objects, and last 4) causal inference for the detected factors on the detected critical event. The pipeline is fully coded in Python programming, relying on three main libraries for deep learning; Tensorflow, Keras, and Pytorch. After training, testing, and validation, the pre-trained deep learning models are utilised for analysing future scenes as a pragmatic computer vision tool. For the objectives of this research, there are multiple advantages to selecting ensemble learning. During the training phase, this approach allows various tasks to be trained separately based on their input and computational requirements or the availability of data the might not be possible with other approaches such as Multitask learning. At inference, it allows the single input to be treated differently throughout the pipeline as either single-frame images or sequential images based on the specific tasks that will be explained in this chapter. In the post-production phase, ensemble learning allows the pipeline of the framework to be modified or expanded for a given task without affecting the other models in the pipeline.

Fig. 4.1 shows the overall workflow of the proposed pipeline when a video stream is received as input. First, phases I and II extract risk factors and agents (pedestrians, cycles, vehicles etc.), respectively, while phase III detects instant actions (near misses) in parallel. The outputs of all preceding phases are then fed into phase IV, where causal inference is performed. The four phases are described in detail in the following subsections:

4.2.1 Phase I: Sensing and detecting the conditions of the environment

This phase tackles the different factors related to the environment that may influence the safety of the cyclist. Alongside image classification, understanding the overall gist of a scene is crucial for understanding the built environment (Oliva and Torralba, 2006) and few studies have been done in this area. For instance, sensing the qualitative measures that are related to the built environment that may contribute to near-misses, such as road infrastructure, lighting and weather conditions. This phase comprises three models: 1) URBAN-i, 2) WeatherNet, and 3) SlipNet.

Model I-WeatherNet: Weather and visual conditions are often addressed individually. WeatherNet introduces a novel framework to automatically extract this information from street-level images relying on deep learning and computer vision using a unified method without any pre-defined constraints in the processed images (i.e. pre-determined field of view, angle, positioning, or cropping). The WeatherNet model comprises four deep Convolutional Neural Network (CNN) models and uses residual learning to extract various weather and visual conditions such as; Dawn/dusk, day and night for the time of day; glare for lighting conditions; and clear, rainy, snowy, and foggy for weather conditions.

Model II-SlipNet: Wet road conditions, combined with other factors related to visibility, weather and/or physical conditions may contribute to many risky situations and instant events when it comes to mobility in a complex environment. Whether driving, cycling, or even walking, a wet surface may cause potential near misses, or serious incidents. The classification of the road is often interpreted based on the perceived weather and precipitation conditions. However, in reality, there may be cases where the ground is wet enough to cause a critical event while the sun is shining, and conversely,

there may be rainy days where the ground is not yet wet. To tackle this subtle issue, we introduce SlipNet. SlipNet is a deep computer vision model based on Convolutional Neural Network (CNN) that classifies road conditions, independently of the current weather or overall visual conditions.

4.2.2 Phase II: Detecting and tracking objects

In this phase, we introduce simultaneous object detection and tracking of road users to the overall framework. The phase consists of two main models: 1) URBAN-i (Object-detection) and 2) multi-object tracking.

Model III-URBAN-i (Object-detection):

To detect road users (i.e. people, cars, trucks, buses, motorcycles, and bicycles) from extracted scenes, the framework uses a Single Shot Mulibox Detector (SSD) method. This method is discussed in terms of architecture, training, and validation in chapter VI. Unlike other object detection approaches, these methods rely on a single feed-forward deep CNN model. It produces bounding boxes and a confidence score for each category of objects presented in the image. There are three reasons for selecting this approach for object detection. First, the model relies on a single deep CNN model to do the prediction, which makes it easier and faster to train. Second, this state-of-the-art method for object detection shows competitive results when compared to the other object detection methods in many deep learning datasets, such as PASCAL VOC2007 (Everingham et al., 2015) and COCO (Lin et al., 2014).

Model IV- Multi-object tracker method:

After object detection, we adopted and implemented the Simple Online and Realtime Tracking (SORT) method (Bewley et al., 2016). The SORT method is suitable for online object-tracking because 1) Its speed allows fast computation without a huge drop on FPS, 2) it relies on simple techniques such as Kalman Filter, which makes it easy to implement online without previous training.

The models and methods addressed in Phase I and Phase II will be explained in detail in chapter V.

4.2.3 Phase III: Detecting instant actions

Model V-CyclingNet:

In this phase, we introduce a novel method called CyclingNet for detecting cycling near misses from video streams generated by a mounted frontal camera on a bike. CyclingNet is a deep computer vision model based on a convolutional structure embedded with Self-attention Bidirectional Long-short Term Memory (LSTM) blocks that aim to understand near misses from both sequential images of scenes and their optical flows. The model is trained on scenes of both safe rides and near misses. Action recognition, relying on the CyclingNet model will be discussed in detail in Chapter VI.

4.2.4 Phase IV: Causal inference

We aim, after precisely extracting a combination of risk factors, to understand: 1) the cause and effect of individual risk factors on the detected events, 2) the causality of these risk factors in the detected events in a time series. Accordingly, this phase relies on statistical modelling techniques to

uncover the causes and the effects of each extracted factor on the detected events. It includes two types of analysis, which will be covered, in detail, in Chapter VII. These types of analysis are:

1) Logistic Regression: We use the detected variable corresponding to critical events as the dependent variable, with detected objects and risk factors are independent or control variables in a logistic regression model. Besides the logistic model, we also include different parametric and non-parametric tests (i.e. t-test) to determine the strength and significance of the results.

2) Granger Causality: Granger causality is a probabilistic method for investigating the causality between two variables in a time series dataset. Unlike understanding the general cause and effect of the individual factors, causality, or the 'Granger-cause', focuses on highlighting when a particular variable comes before another in time series data.

4.3 Materials and datasets

Each deep model introduced in the overall methodology is trained separately on different datasets that serve the purpose of the given task of the model, which will be discussed in the upcoming three chapters (Chapter V, VI, and VII). However, it is essential to highlight that the proposed framework is validated on a single dataset that accumulates video streams of safe and unsafe rides in various weather, visual and built environment conditions. The dataset is also utilised and explained in detail in Chapter VI. In summary, this self-collected dataset from YouTube consists of 74,477 sequential frames, and we computed their equivalents of optical flows frames (74,469). 8,567 sequential frames belong to near miss cases (11.5% of the total sequential frames), which occur at sparse intervals. They represent 209 unique near miss videos of an average duration of 1.3 seconds (40.9 sequential frames).

The purpose of this dataset is to validate the performance of the overall methodology rather than the individual methods. It is worth mentioning that this dataset is manually labelled to ensure higher accuracy and relevance of the dataset to the stated issues of this research.

4.4 Framework Stringency Index

Similar to training and validation datasets, the performance of each model introduced in the pipeline of the overall methodology is evaluated with different metrics and loss functions depending on the types and scopes of the given task of the model, which will be discussed in detail in the upcoming chapters (Chapter V, VI, and VII). Nevertheless, the models not only vary based on their evaluation metrics but also in the resulting accuracies and precisions. On the other hand, as shown in **Fig. 4.1**, The relations between these different models vary. For instance, some models function consecutively, while others function in parallel to other phases. The goal of this research is to provide a stable framework to be used as a computer vision tool for the detection of near misses, risk factors, and their effects on near misses. This makes it a challenge for a pipeline of mixed models and different ensemble techniques to be evaluated as a whole. Traditionally, the performance can be measured based on the sum of the losses of each model when models are evaluated similarly and hold the same weights of utilisation in the entire pipeline. Given that we aim to develop a verified pipeline of the different pre-trained models, we introduce a new stringency index to indicate the performance of the entire framework on a given input that can draw a conclusion based on three aspects: 1) the individual loss

of each model, 2) the number of outputs of each model, and 3) the weight of the utilisation of each model in the framework. The framework Stringency Index (SI) is defined as:

$$SI = \sum_{1}^{t} \sum_{o=1}^{n} \sum_{i=1}^{j} ((\beta_i \cdot P_i)/t) \quad (4.6)$$

where j denotes the number of outputs per model, n denotes the number of models in the framework, t represents the total number of sequential frames, P represents the estimated average precision between the predicted and actual value for a single output o of a given model i, and β represents the normalised statistical weight of a given risk factor on a given task (i.e. detection of near miss), which will be addressed, in detail, in chapter VII.

4.5 Summary

This chapter introduces the overall methodology of the research. It shows the organisation of the different deep and mathematical models in an integrated pipeline. The general goals of this framework are to detect critical issues in cities, such as cycling near misses, while extracting their risk factors and their effect on these critical events. This chapter also introduces a framework stringency index that aims to evaluate the overall methodology, in addition to the evaluation metrics conducted on the individual methods and models. The importance of this index can be highlighted in evaluating the weights and the importance of the individual models in their function and utility in the overall methodology, nevertheless, the number of outputs that each sub-method contributes to the overall methodology. Last, the chapter also highlights the importance of the flexibility of the introduced pipeline that could allow and cope with any future adaptation, either in terms of refining methods or introducing new ones.

5

SENSING THE ENVIRONMENT AND EXTRACTING RISK FACTORS

5.1 Overview

Sensing the environment at a given time and space is indispensable for scene awareness and a better understanding of critical events.-This chapter describes a set of algorithms that are used to sense the environment and extract risk factors that may be associated with near misses, which are used in phases I and II of the methodological framework presented in figure 4.1 in Chapter 4. The algorithms are intended to detect the multi-labels of weather, visual and surface conditions with a unified method that can be easily used for practice. The proposed framework automatically extracts this information from street-level images relying on deep learning and computer vision using a unified method without any pre-defined constraints in the processed images. In phase one, a pipeline of five deep Convolutional Neural Network (CNN) models is trained, relying on residual learning using ResNet50 architecture, to extract various factors such as Dawn/dusk, day and night for time detection, and glare for lighting conditions, clear, rainy, snowy, and foggy for weather conditions, and lastly wet or dry for surface conditions. In phase two, an object detection model is introduced to detect and track the different road users (such as persons, cars, trucks, bicycles, trains, motorcycles and buses).

The materials and outcomes of this chapter are published in two journal articles and a conference paper as follows: 1) WeatherNet (Ibrahim et al., 2019a), 2) URBAN-i (Ibrahim et al., 2019a), and 3) SlipNet (Ibrahim et al., 2020a).

5.2 The overall framework map

This chapter covers the first two phases of the overall methodology introduced in Chapter IV. **Fig. 5.1** shows a thumbnail of the overall methodology, highlighting the study area covered in this chapter. The first phase covers sensing the overall environment and extracting risk factors, whereas the second one covers the localisation and tracking of road users. It is worth mentioning that these two phases are trained separately. However, their inference takes place, in parallel computation, simultaneously.

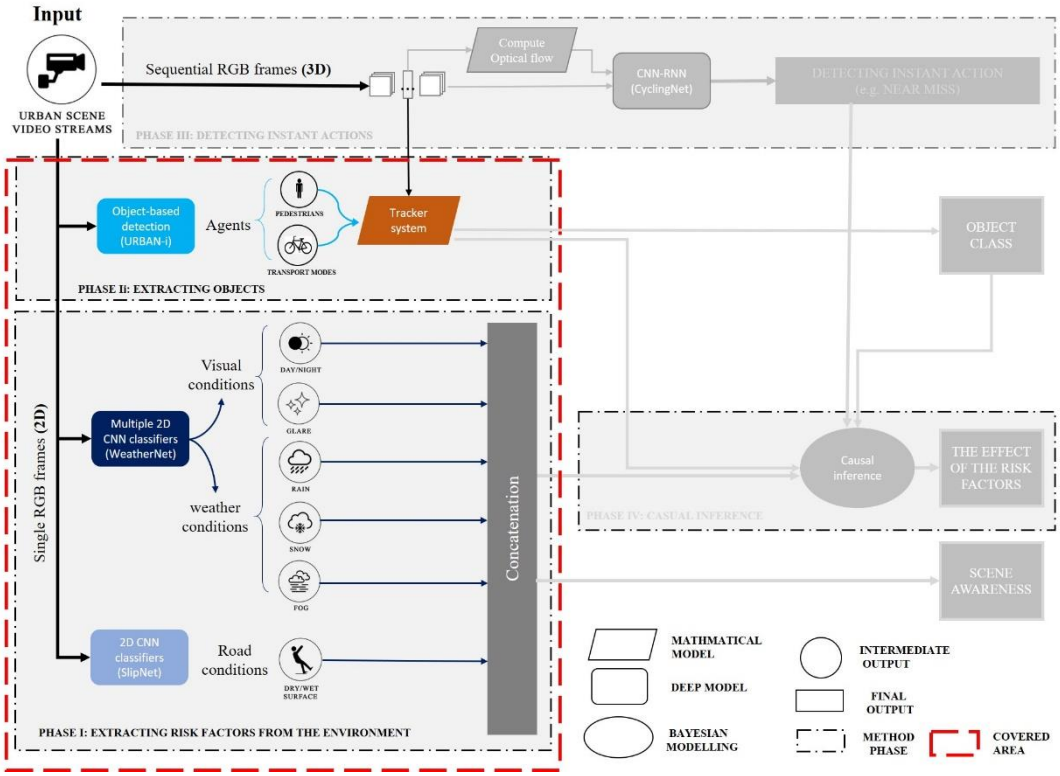

Figure 5.1: Keymap of the overall methodology covered in this chapter

5.3 WeatherNet

5.3.1 Architecture

WeatherNet is a framework of parallel deep CNN models trained to recognise weather and visual conditions from street-level images of urban scenes (See **Fig. 5.2**). The architecture comprises four deep CNN models to detect dawn/dusk, day, night-time, glare, rain, snow, and fog, respectively. The four models are: 1) NightNet detects the differences between dawn/dusk, day and night-time. It aims to understand the dynamics of time despite the dynamics of weather conditions and urban structure, 2) GlareNet detects images with glare regardless of its source (sun or artificial light) in all weather conditions and times of the day. Glare is defined as a direct light source that can be seen to cause rings or a star effect on the length of the camera without any correction, 3) PrecipitationNet detects clear, rainy, or snowy weather for day and night-time. It is worth mentioning that “clear” as a term is used to refer to no precipitation, 4) FogNet detects the presence of fog.

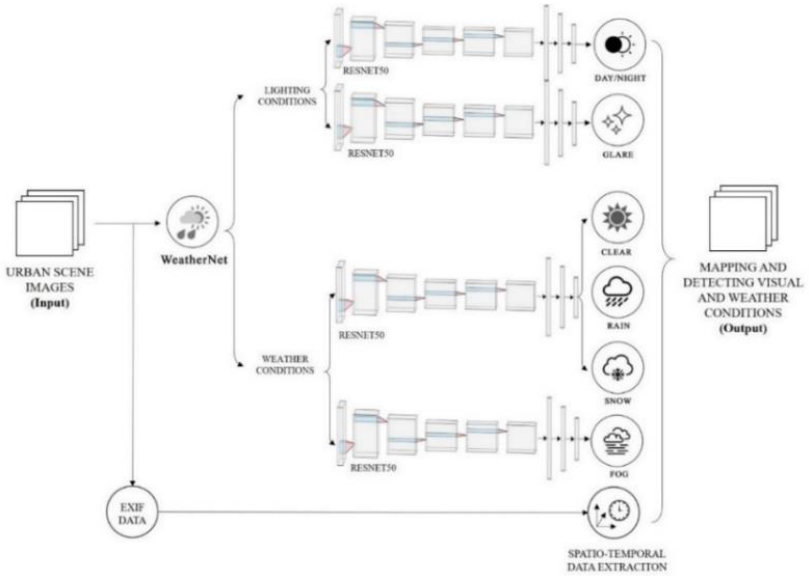

Figure 5.2: The framework of the WeatherNet.

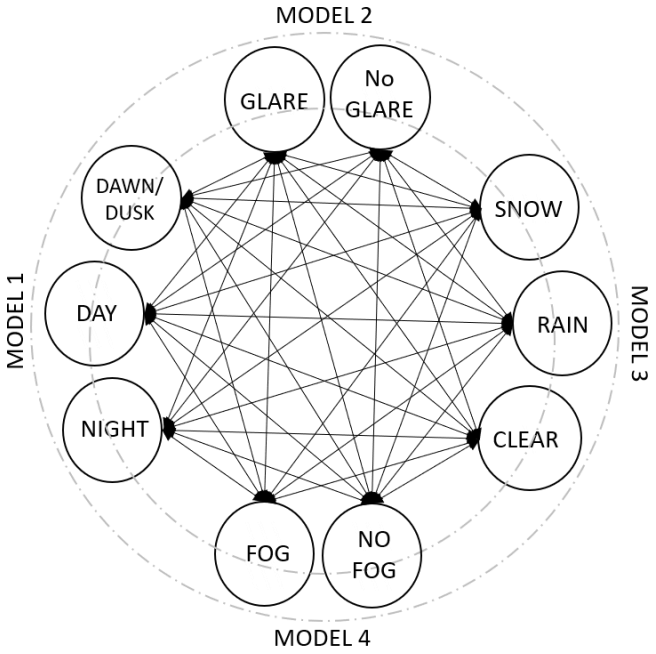

Figure 5.3: Exclusive vs co-existing classification classes.

Models 2 and 4 are trained as binary classifiers (0, 1) that detect whether one of the aforementioned events occurs, whereas models 1 and 3 are trained to output one of three classes. The main reasons for training different sets of CNN models then combining them in a framework are the complexity of the classification of urban scenes and the mutual occurrence of one or more of the events at the same time. **Fig. 5.3** explains the classes that may occur in one scene by solid arrows, whereas the mutually exclusive classes are not linked. For instance, it may be rainy and foggy during the daytime, while glare

is present. Therefore, combining separate models that detect a certain event in a binary/categorical fashion gives a better description of the events in a single image. This also lends modularity to the system, enabling simple use of all or part of the framework for different studies, depending on which factors are useful. Furthermore, the modular form makes the performance of the individual models independent of each other, which allows the modification or improvement of one classifier or more without changing the entire framework.

The architecture of the introduced models is based on residual blocks (for more details, see section 2.3). We have selected ResNet50, in addition to our base model, for designing our network. The main reason for selecting ResNet50 is due to the trade-off between accuracies, training, and inference runtime. The training and testing images are resized to (224 X 224 X 3) and fed-forward to the input layer of ResNet50 via transfer learning. The gradients, pre-trained on the ImageNet database (Krizhevsky et al., 2012; Russakovsky et al., 2015), of the different residual blocks of convolution, pooling, batch normalisation layers are set to false, whereas the gradient of the two fully-connected layers of 64 nodes are activated by a ReLU function (Dahl et al., 2013; Glorot et al., 2011), defined as:

$$f(x) = \max(0, \mathrm{x}) \tag{5.1}$$

where x is the value of the input neuron.

The output layer of the model gives a binary output of single neurons activated based on a sigmoid function, defined as:

$$\boldsymbol{\delta(x)} = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{-x}} \tag{5.2}$$

where x is the value of the input neuron.

The four CNN models are trained based on the back-propagation of error with a batch size of 32, with 'adam' optimiser (Kingma and Ba, 2015) and with an initial learning rate of 0.001 and momentum of 0.9. Each model is trained for 100 training cycles (epochs).

5.3.2 Base model architecture

In order to evaluate the introduced methods in this chapter, we have built a deep Convolutional Neural Network (CNN) model (Guo et al., 2016; Hinton et al., 2006; LeCun et al., 2015) that can be used as a base model. This model represents a lighter version of the Vgg16 model that requires less time for training from scratch with minimal computational resources and has been utilised previously for detecting slums and informal settlements from street-level images (Ibrahim et al., 2019a). The model transforms images into an input layer of size (200 x 200 x 3). The overall architecture of the base model is built based on 10 hidden layers of different types. After the input layer, the model consists of 4 convolutional layers. After the first two, each layer is followed by a Max-pooling layer. After the Flatten layer, two fully-connected layers are applied, followed by a single neuron output layer.

The First three layers consist of 32 Feature maps of subsampling (3 x 3), While the third one consists of 128 feature maps of subsampling (3 x 3). The four convolutional layers rely on Rectified Linear Unit (ReLU) as an activation function to increase the nonlinearity of the model and enhance the performance of the neurons (Dahl et al., 2013; Glorot et al., 2011).

Moreover, the three Max-pooling layers are of a downsampling size (2 x 2). These layers are responsible for reducing dimensionality, in addition, it allows the model to adapt to the variation of the scale, rotation, or skewing of samples that represent a certain feature (Scherer et al., 2010). After these convolutional and Max-pooling layers, the flatten layers allow the model to convert the feature maps to neuron vectors that can be feed-forwarded to the two fully-connected layers. The first fully-connected layer consists of 256 neurons, whereas the second one consists of 64 neurons. Both of these layers are activated based on a ReLU function. In order to avoid overfitting, we have applied features dropout regulation after several hidden layers (Dahl et al., 2013; Srivastava et al., 2014). The output layer is based on the number of outputs of each model. It is activated based on a sigmoid function.

5.3.3 Data

While Google Street-view images are a good source for various deep learning applications in cities, the images presented there only represent urban areas at a single weather condition, commonly clear weather, neglecting other visual and weather conditions that impact the appearance of cities. On the other hand, there are different datasets for detecting different weather conditions. For instance, the Image2Weather dataset consists of more than 180,000 images of global landmarks of four weather categories, such as sunny, cloudy, rainy, snowy and foggy (Chu et al., 2016). Similarly, the Multi-class Weather Image (MWI) dataset consists of 20,000 images of different weather conditions (Z. Zhang et al., 2016). Another example is a binary weather dataset that contains 10,000 images belonging to either sunny or cloudy weather (Lu et al., 2017). Also, a large dataset of images is presented to describe weather conditions from the view of cloud intensity, such as clear, partly cloudy, mostly cloudy, or cloudy, including time and location data (Islam et al., 2013). However, the dataset only represents cities at daytime for clouds intensity, neglecting the other factors**.**

Put together, creating our own dataset that represents the different environmental conditions of urban scenes becomes a crucial step to conduct this research. The dataset comprises 23,865 images that are gathered from the web, specifically Google images for training and testing, using different queries for each class of the weather and visual conditions that includes day and night-time, glare, fog, rain, snow and clear weather. These images can be accessed from Google image engine with a python script via a combination of ‘street view’, ‘urban area’, ‘highway’ or ‘road‘ with keywords for weather classes and an optional keyword for city names in a Boolean expression. For example, in the case of rain, these expressions are rainy AND road, rainy AND urban AND scene*, rainy AND street*, rainy AND highway, rainy AND London, rainy AND Paris, rainy AND Cairo, or rainy AND city. After repeating this process for each class and before reaching the final size of the dataset, the downloaded images are inspected qualitatively to remove duplicates and disregard images that do not belong to any of the subcategories of each given task.

Figure 5.4: Samples of WeatherNet dataset

It is worth mentioning that the process of manually labelling these images to the subcategories of each given task and verifying the outcome is time-intensive. This is because some images may include features belonging to two exclusive classes. After the first inspection, a thorough categorisation needed to be made, especially when images do not contain enough features to represent a visual class. A decision to disregard these images is needed. Accordingly, such a process increases the workload and time interval needed to make realistic labelling for the data beyond their meta-data.

On the other hand, selecting images based on their public accessibility without breaching any individuals' copyrights was also a key for selecting or disregarding images. Subsequently, the images collected were used only for training and testing, without publicly sharing or posting them elsewhere. **Table 5.1** summarises the classes and sample size of data sets used for each model. **Fig. 5.4** shows a sample of the different classes for training, testing, and validation. The datasets for each CNN model are subdivided into training and testing sets in an 80%-20% train-to-test fashion.

Table 5.1: Sample size and categories of the data sets

CNN model	Dataset classes	Sample size
Model1- NightNet	Dawn/Dusk	1673
	Day	2584
	Night	1848
Model2- GlareNet	Glare	1159
	No glare	3549
Model3- PrecipitationNet	Clear	4017
	Rain	2343
	Snow	2347
Model4- FogNet	Fog	718
	No fog	3627

The images used for training are still limited, nevertheless, deep models require a large data set to ensure better generalisation. Accordingly, we have applied a data augmentation technique, on the one hand, to enhance the training of the model, on the other, to account for the class imbalance of each model by generating extra images for the under-represented classes such as fog and glare. The algorithm allows the model to create random images based on four attributes; rescale, shear, zoom, and horizontal flips. These techniques are often common approaches for best practices to enhance the training process and the overall performance of deep learning models (Goodfellow et al., 2017; LeCun et al., 2015). While these approaches augment the training data, yet they do not change the class of the images. Nonetheless, they offer realistic possibilities that can represent various urban scenes of the same location. **Fig. 5.5** shows an example of an original image and a sample of 15 augmented images generated from the original image.

Figure 5.5: The urban scene for a planner area in London and examples of data augmentation

5.3.4 Evaluation metrics

We evaluate the performance of each CNN model using the following metrics: A cost function of Cross-Entropy to evaluate the model loss during training, testing, and validation. It is defined as:

$$\mathrm{E} = -\sum_i^n t_i \log(y_i) \tag{5.3}$$

where $\boldsymbol{t_i}$ is the target vector, $\boldsymbol{y_i}$ is the output vector, n represents the number of classes. We also calculated accuracy, precision and recall, false-positive rate, and F1-score for each model, defined as:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{5.4}$$

$$Precision = TP/(TP + FP) \tag{5.5}$$

$$Recall = TP/(TP + FN) \tag{5.6}$$

$$False - positive\ rate = FP/(FP + TN) \tag{5.7}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5.8}$$

Where TP are the predicted true-positive values, TN are the predicted true-negative values, FP are the predicted false-positive values, and FN are the predicted false-negative values.

Last, we compare the performance of our framework with other benchmarks in terms of scope and accuracy. This discussion is partly qualitative due to the absence of benchmark data sets to compare

all methods results. However, we also evaluate the performance of WeatherNet on two available datasets (Gbeminiyi Oluwafemi and Zenghui, 2019; B. Zhao et al., 2018) and compare the results of our framework with the original outputs.

5.3.5 WeatherNet results

Putting all the algorithms of WeatherNet together, the framework can enable the users to extract information of georeferenced weather and visual conditions to be used for multi-purpose research related to scene awareness, in which weather and visual conditions play a crucial role.

Table 5.2 summarises the evaluation metrics of each CNN model at the testing phase. After training the four CNN models for 100 epochs, the accuracies for the NightNet, GlareNet, PrecipitationNet, and FogNet on the test data sets are 91.6%, 94.8%, 93.2%, and 95.6%, respectively. The models also show high precision and F1-score with low false-positive rates of 6% or lower. These results achieved by transfer learning strongly outperform the performance of the introduced base model. To investigate further the performance and the fitness of each model during training and testing, **Fig. 5.6** shows the training and validation accuracies for each epoch, highlighting the overall performance and fitness of each model. It shows the consistency of the accuracies between the training and testing curves, in which no over-fitting is observed. However, due to the high variance in data and subtle differences among classes for the same model, the output for each training cycle show a high level of instability to converge and reach global minimum loss.

Table 5.2: Diagnoses of the CNN models for the test sets.

CNN model	Loss (Cross Entropy)	Accuracy (%)	Precision [(a)]	Recall/ True-positive rate [(a)]	False-positive rate [(a)]	F1-score
Model1- NightNet	0.098	91.6	0.885	0.825	0.045	0.854
Model2- GlareNet	0.040	94.8	0.883	0.895	0.035	0.889
Model3- PrecipitationNet[(b)]	0.077	93.2	0.959	0.932	0.068	0.947
Model4- FogNet	0.037	95.6	0.862	0.829	0.022	0.845

[(a)] The metrics are evaluated for the referenced class -indexed zero- for each model.
[(b)] This model contains three classes, in which the false-positive rate is shared with the classes prior to the referenced class.

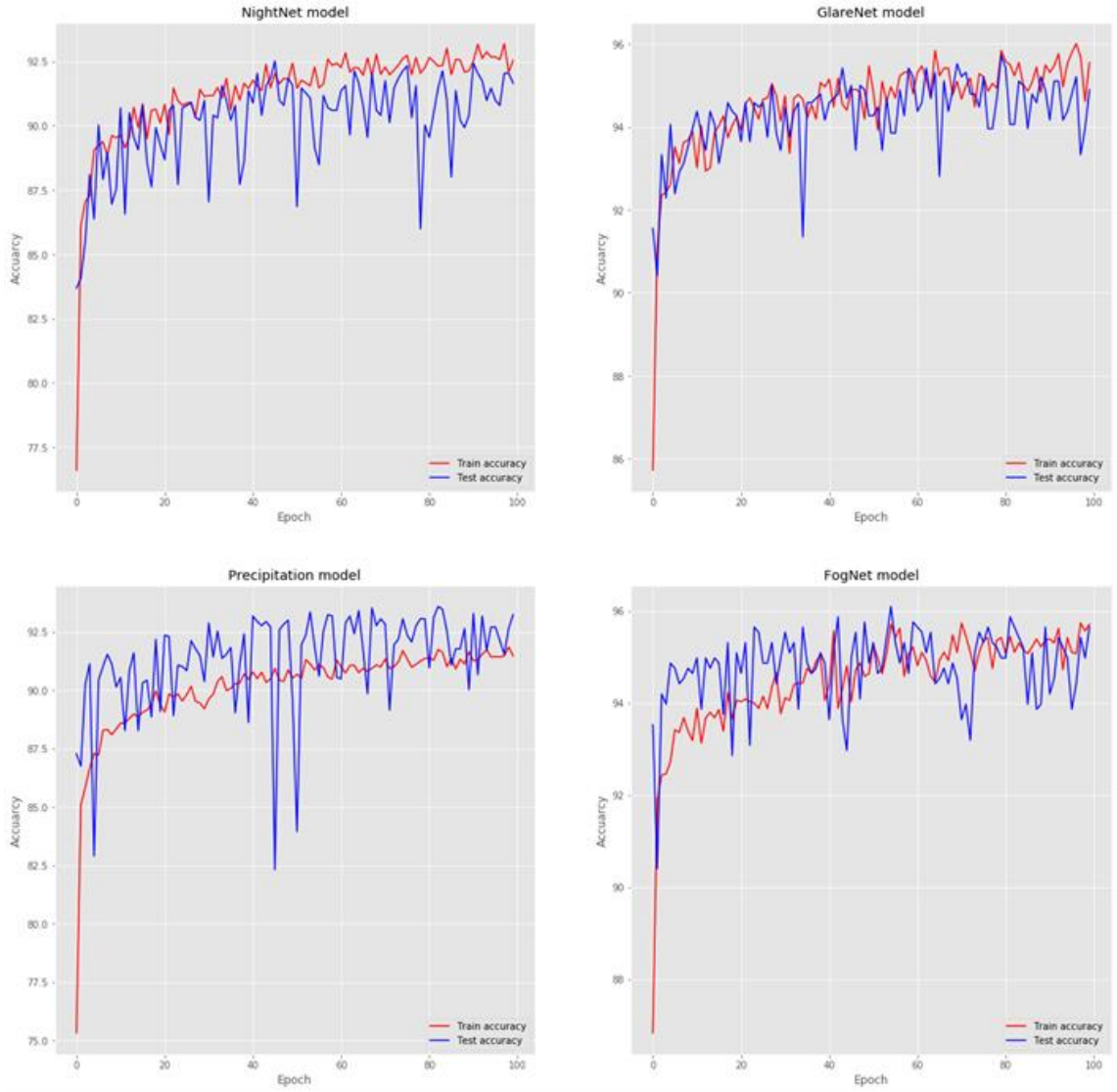

Figure 5.6: The training and test accuracies per training cycle for each CNN model

Table 5.3 evaluates our framework against other existing methods that deal with some aspects of weather and visual condition detection. The method used for each model and the yielded accuracy on the dataset used for each method is also shown. WeatherNet performs favourably in terms of accuracy compared with the other methods, but it should be noted that the datasets used are not the same.

Methods		Night-time detection (classes)	Glare detection	Fog detection	Weather detection (classes)	Overall score
(Roser and Moosmann, 2008)	Regions of interest-Histograms	-	-	-	X (clear, light rain, heavy rain)	0.85
(Islam et al., 2013)	Support Vector Regressor	-	-	-	X (clear, partly cloudy, mostly cloudy, cloudy)	NA
(Chu et al., 2016)	Random Forest Classifier	-	-	x	X (Sunny, cloudy, rainy, snowy)	0.70
(Lu et al., 2017)	CNN model	-	-	-	X (Sunny, cloudy)	0.91
(Villarreal Guerra et al., 2018)	Different types of CNN models	-	-	x	X (snowy, rainy)	0.80
(Gbeminiyi Oluwafemi and Zenghui, 2019)	SAID ENSEMBLE METHOD	-	-	-	X (sunny, cloudy, rainy)	0.86
(B. Zhao et al., 2018)	CNN-LSTM	-	-	x	X (sunny, cloudy, rainy, snowy)	0.91
WeatherNet	**Multiple Residual deep models**	**x (Dawn/dusk, day, night)**	**x**	**x**	**X (Clear, rain, snow)**	**0.93**

Table 5.3: Evaluations of WeatherNet framework on other open-sourced datasets.

Open-sourced benchmark datasets	Total images		Labels	Method	Testing scope	Original method score	WeatherNet score
Multi-class Weather Dataset for Image Classification	(Gbeminiyi Oluwafemi and Zenghui, 2019)	1,125	Cloudy, sunshine, rain, sunset	SAID ENSEMBLE METHOD II	Rain detection	Accuracy: 95.20%	**Accuracy: 97.69%**
Multi-label weather dataset (test-set)	(Zhao et al., 2019)	2,000	(Sunny, cloudy, rainy, snowy, foggy)	CNN-Att-ConvLSTM	Sunny/clear detection	(Precision/Recall/F1) 0.838/**0.843**/0.840	(Precision/Recall/F1): **0.924**/0.827/**0.872**
					Fog detection	(Precision/Recall/F1): **0.856**/0.861/0.858	(Precision/Recall/F1): 0.833/**0.940**/**0.883**
					Rain detection	(Precision/Recall/F1): 0.856/**0.758**/**0.804**	(Precision/Recall/F1): **0.958**/0.651/0.775
					Snow detection	(Precision/Recall/F1): **0.894**/0.938/**0.915**	(Precision/Recall/F1): 0.789/**1**/0.882
					Average scores	AP, AR, AF1 0.86, 0.85, 0.85	AP, AR, AF1 **0.88**, 0.85,0.85

To provide a quantitative comparison, we apply the WeatherNet to two open-source datasets used in previous studies (Gbeminiyi Oluwafemi and Zenghui, 2019; Zhao et al., 2018). **Table 5.4** describes the datasets used for evaluation in terms of size, labels and the original approach used for prediction. The outcomes and evaluation of our model scores on these datasets in comparison to the original are shown, using the same evaluation metrics used in the original research (accuracy for the first dataset as it the one measure mentioned in their article, precision, recall and F1-score for the second dataset). In the case of the first dataset, the table shows the labels used in the method introduced by Gbeminiyi Oluwafemi and Zenghui (2019), which are cloudy, sunshine, rain, sunset. It is worth mentioning that the comparison is only based on one class, rain, which is the only common class between WeatherNet and their method. WeatherNet achieves a higher accuracy score than their method using their test set. In the case of the second dataset, our model shows a higher precision than the original method when classifying clear and rain but a lower precision when classifying fog and snow. Nevertheless, a higher recall for only fog and snow classes has been achieved by WeatherNet. By comparing F1-scores, WeatherNet achieves higher scores when it comes to classifying clear and fog weather and lower ones when classifying rainy and snowy weather. Both methods achieve an equal average F1-score of 0.85 and an equal average recall of 0.85, whereas WeatherNet achieves a higher average precision. On the other hand, it is worth mentioning that the comparison has been made only for the classes that are common for both methods, whereas some classes, such as night-time, day-time, daw/dusk time, and glare, have not been included in the comparison due to their absence in the mentioned method introduced by Zhao et al. (2019), whereas one class, cloudy, has been excluded from the comparison due to its absence in WeatherNet.

Last, as we aim to use the proposed framework pragmatically for recognising and mapping weather and visual conditions in cities, **Fig. 5.7** shows a few examples of the different model predictions of a wide range of urban scene images taking from different cities globally. It highlights the diversity of the images used for prediction. Regardless of the change of urban structure, camera angles, scene lighting, and components, the proposed models show high accuracy for scene awareness related to visual and weather conditions.

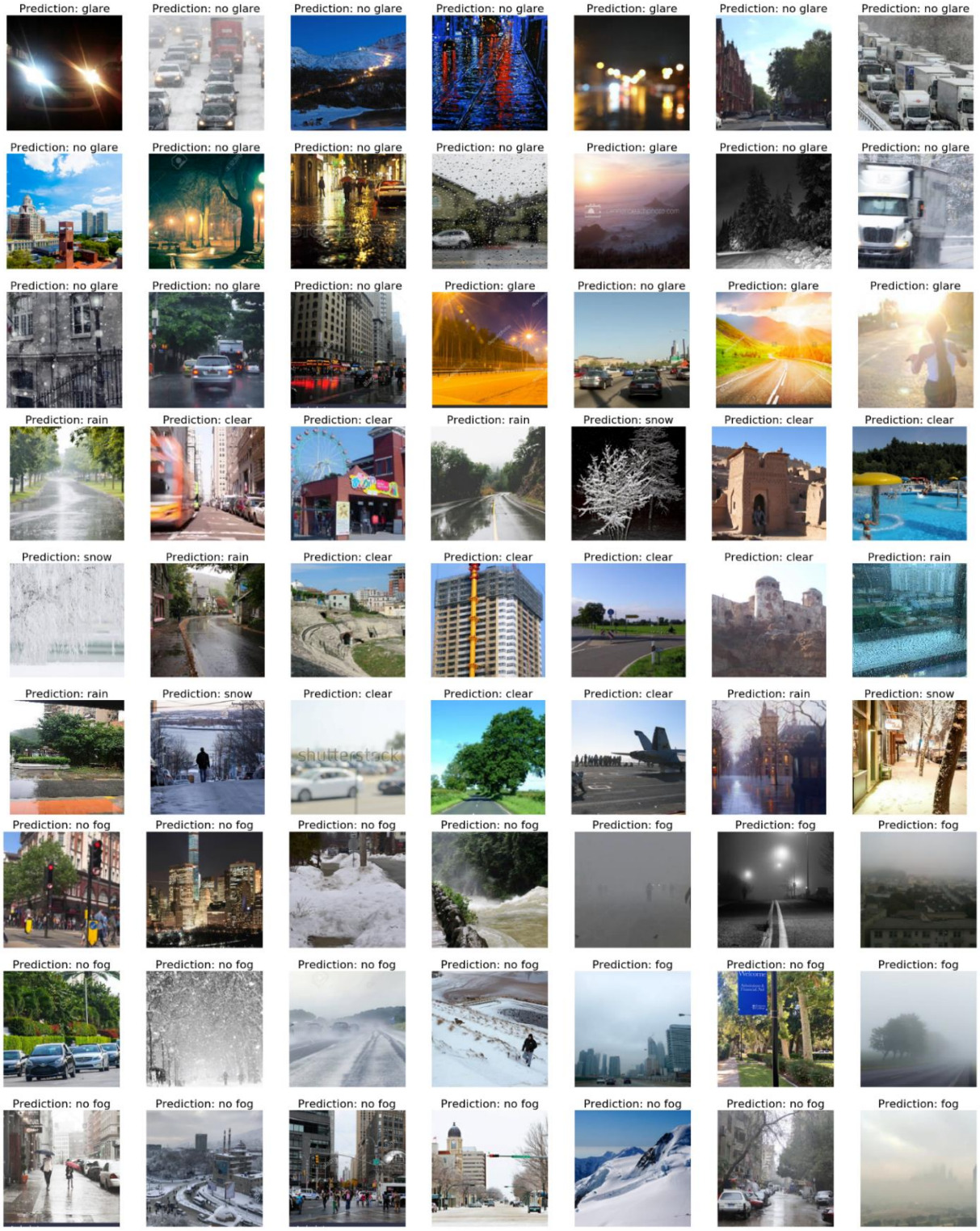

Figure 5.7: The results of the CNN models on street-level images from different cities globally

5.3.6 Limitations

The proposed framework shows novelty in analysing a wide range of street-level images of cities that belong to various urban structures, visual, and weather conditions globally. The precision of the framework in classification depends on the individual accuracy of each trained CNN model. While the miss-classification error for each classifier is below 8% on the test sets, as for future work, more

experiments with different architectures of CNN models or the way the framework is pipelined may enhance the accuracy.

While the trained deep models show high accuracy, precision, recall and F1-score in classifying scenes regardless of the position of the camera, weather, or lighting conditions, misclassification still can be encountered in scenes. This is due to various settings, such as scenes of heavy cloud that tends to seem rainy or scenes of heavy fog that tend to seem snowy. Similar to human eyes, single shots can be interpreted differently if they have only been seen once, and the quality of classification can be enhanced by seeing sequential images. Similarly, such an issue can be solved when video stream data is fed to the framework, where a threshold or a smoothing function is applied for a sequence of frames of short-time intervals. Subsequently, the best probabilities of the prediction can be taken into account for classification. The overall accuracy of multi frames can be enhanced by a threshold of multiple predictions.

Comparing the performance of the proposed models to previous work remains a limitation due to the absence of weather datasets that comprise similar classes as presented in this research (i.e. including images of weather at day and night-time and images with and without glare). However, this makes the proposed model indispensable in responding to the current knowledge gap in this research area and for analysing the variations of urban scene images by deep learning and computer vision that may be helpful for driver-assistance systems or planners and policy-makers in cities.

5.4 SlipNet

Road conditions, in terms of slippery, combined with other factors related to visibility, weather and/or physical conditions may hold responsible for many risky situations and instant events when it comes to mobility in a complex environment. Whether driving, cycling, or even walking, a wet surface may cause potential near misses or serious incidents. The classification of the road is often interpreted based on the perceived weather and precipitation conditions, however, in reality, they may be cases where the ground is wet while it is a sunny day which can cause critical events than on a rainy day, whereas the ground is not wet yet. To tackle this subtle issue, we introduce SlipNet. The SlipNet is a deep computer vision model based on CNN to only classify road conditions, despite the current weather or overall visual conditions. Combined with other state-of-the-art models, such as WeatherNet, we aim to precisely extract a combination of risk factors that can be used for understanding the causality of risky situations such as near misses or serious incidents. Accordingly, this model is indispensable for any safety-related research.

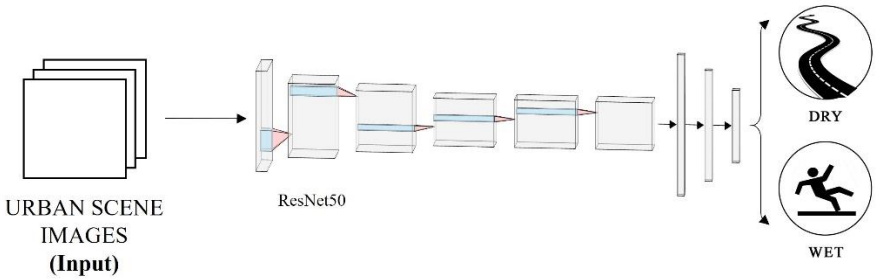

Figure 5.8: the architecture of the SlipNet model

5.4.1 Architecture

Fig. 5.8 shows the overall architecture of the SlipNet model. Similar to WeatherNet, we trained the SlipNet model based on residual learning, relying on ResNet50 architecture via transfer learning. Two fully-connected layers were added, each contains 64 neurons and activated with Linear Rectified Unit (ReLU), to output a single neuron. We applied a Dropout layer after each fully-connected layer of size 0.3 to avoid over-fitness. The binary classes are determined based on a threshold of 0.6 of the probability of the outputted neuron. Such a threshold is selected based on a trial and error to output a prediction with optimum Recall, False-negative, and Precision values. The accuracy of the trained model is based on the function of the cross-entropy of error. After 100 cycles of training (epochs), the training and testing accuracies are higher than 90%, whereas the loss is less than 0. 1. The outcomes of the training and testing curves for both accuracy and loss show relatively uniform results with no strong evidence of over-fitness. While the model can be improved in various ways from a data collection perspective and hyperparameter tuning, the current result of the SlipNet model shows a reliable outcome for generalisation and deployment purposes to be used for extracting data from images or video streams.

5.4.2 Data

There are no existing deep learning datasets that label and classify the surface conditions in terms of slippery. While Google Street View images are a good database for many urban analytics applications, they represent urban scenes in a single condition of clear weather during the daytime. This lacks the essential variations of the different conditions to train the introduced model. Given this limitation and the absence of any benchmark data set, collecting our database becomes the only reliable resource to conduct this research. Based on the Google images search engine, we collected more than 5000 images that belong to urban and non-urban scenes, which includes various types and conditions of the ground at different times and different weather conditions. Also, these images are collected without any restriction for the image size, existence of urban forms, components, or field of view. After visual inspection, we focused on 3367 images that belong to both dry and wet ground. Furthermore, these 3367 images are divided according to training and test with 0.8 to 0.2 proportions, respectively.

The ground truth for the model is defined based on three criteria. First, the obvious case of the surface conditions from where the urban scene image is taken. Second, the metadata associated with the images from search engines (such as Google search engine) when data is gathered. Put all together, the collected images are assessed for ensuring label relevancy and a wide representation of image orientation and lighting conditions by visual inspection and only then the images are labelled to either wet or dry surface. It is worth mentioning that these images are only used for training and validation, whereas the images presented in the results section for further validation are taken by the authors.

In order to allow a higher degree of freedom for analysing the status of a wide spectrum of urban scene images, we trained the model to understand both aerial perspective and street view images. Accordingly, this will allow the model to identify the status of the urban scenes, regardless of the angle and the elevation of the input image. On the other hand, the model can also classify images of daytime or night-time shots of different weathers, including sunny, foggy, rainy, or snowy weather. While this variation complicates the training process and adds limitations to the model, however, it allows the

proposed model to be widely used and furtherly developed to meet various mapping or sensing purposes.

5.4.3 Results

Table 5.5 summarises the evaluation metrics of the SlipNet model at the testing phase. After training the model for 60 epochs, the model's accuracies are 94.2% and 92.3% for training and testing, respectively, as shown in **Fig. 5.9**. The models also show high precision and F1-score with low false-positive rates of less than 1%.On the other hand, **Fig.5.10** shows the prediction results on a sample of the test set, highlighting the variations of urban scenes that include various weather, visual, and ground conditions.

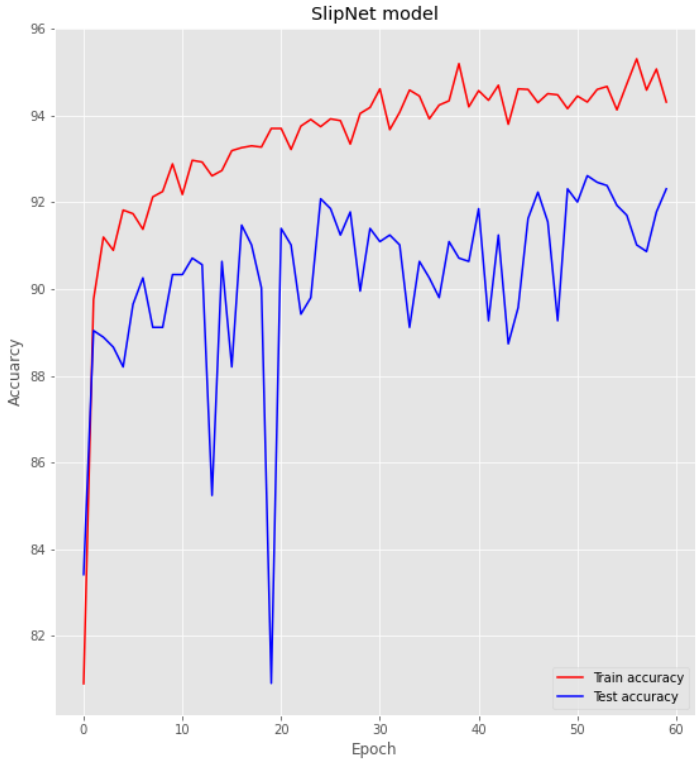

Figure 5.9: The training and test accuracies per training cycle for each CNN model

Table 5.4: Diagnoses of the SlipNet models for the test sets.

CNN model	Loss (Cross-Entropy)	Accuracy (%)	Precision	Recall/ True-positive rate	False-positive rate	F1-score
SlipNet	0.098	92.3	0.918	0.940	0.096	0.929

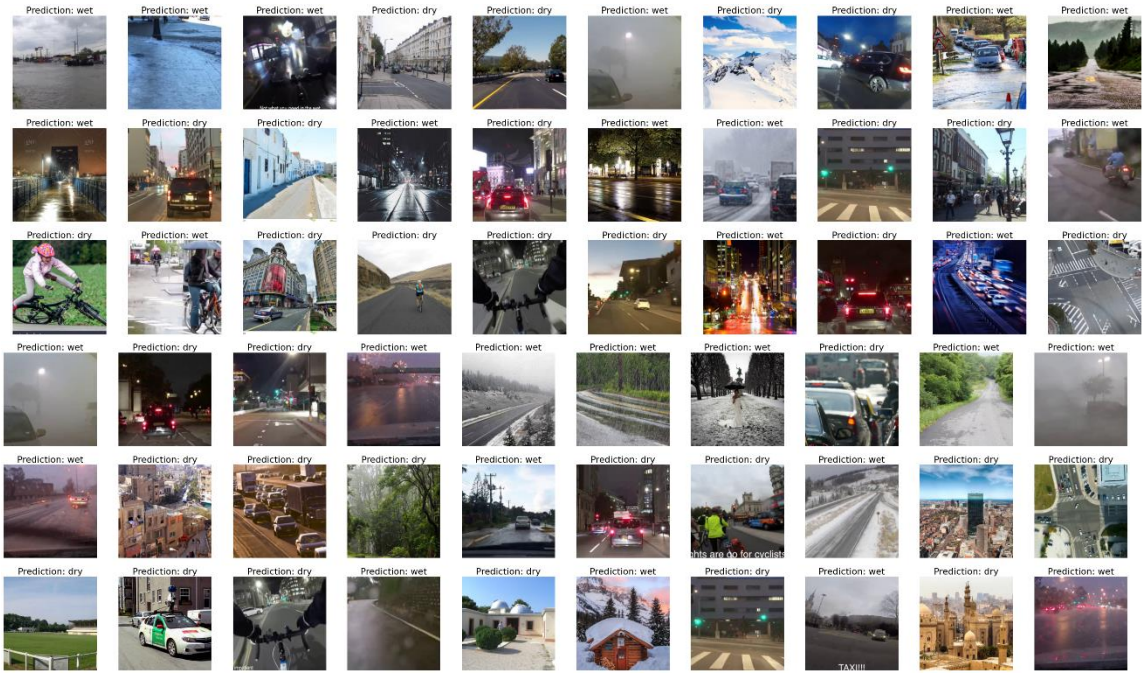

Figure 5.10: Sample of the inferred images by the SlipNet model

5.4.4 Limitations

While the introduced integrated model shows novelty in analysing a wide range of images that belong to different environmental conditions, the limitations of some models such as the SlipNet model appear in analysing images that are mixed with two classes in the same scene (i.e. mainly dry surface with a partly wet surface). In future work, a potential way to develop the model further is by using semantic segmentation and scene parsing. This pixel-level segmentation would allow the model to provide multiple categorizations and localisations of the predicted class for a single image. Accordingly, this will enhance the accuracy of the model when detecting complex scenes in the real world.

The model precision in detecting and classifying urban scenes depends on several factors. First, the individual accuracy of each pre-trained CNN model is a key factor. Each one can be fine-tuned to achieve better accuracy and results with larger training datasets, higher computational power, and deeper networks. However, the goal of this research is to show evidence on how to sense and tackle the complexity of urban issues with deep learning and computer vision with less effort and using data that are available and accessible by everyone anywhere in the globe, without the means of expensive sensors.

5.5 Object detection and tracking

5.5.1 Architecture

The introduced architecture consists of two consecutive phases that are trained separately, in which the output of the object detection model is an input for the tracking method. First, to detect people and transport modes from urban scenes, we have used a Single Shot Multibox Detector (SSD) method (W. Liu et al., 2016). **Fig. 11** shows the overall architecture of the SSD model. Unlike other object

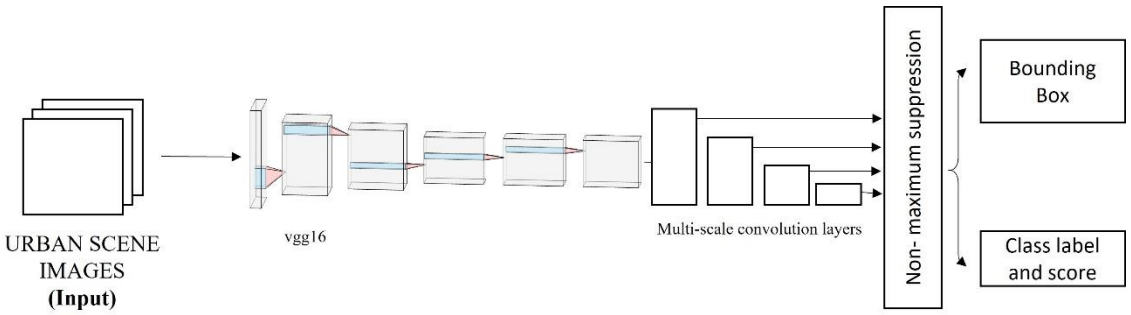

Figure 5.11: the architecture of the SSD model

detection approaches, SDD relies on a single feed-forward deep CNN model. It produces bounding boxes and a confidence score for each category of objects presented in the image. There are three reasons for selecting this approach for object detection. First, the model relies on a single deep CNN model to make the prediction, and this makes it easier and faster to train. Second, this state of the art method for object detection shows competitive results when compared to the other object detection methods in many deep learning datasets, such as PASCAL VOC2007 (Everingham et al., 2015), COCO (Lin et al., 2014), and ILSVRC.

The object-detection model architecture relies on a base network for high-quality image classification, as discussed in section 2, that is truncated before the layers of classification, and an additional structure to the network is added. The first base of this network is built on the architecture of the VGG16 model that deals with classifying several image categories (Simonyan and Zisserman, 2015), including people and the different transport modes. The second part of the model relies on multi-scale feature maps and convolutional layers for object detection. These added convolutional features enable the model to detect an object at different scales and give a confidence score for each bounding box for the occurrence of an object in the image. The major difference of this approach in comparison to the training of other detectors is that the model only requires an input image with a bounding box as a ground truth. This facilitates the training process of the model while maintaining high accuracy for object detection.

The objective loss function of the model is defined based on the weighted sum of the confidence loss (conf) and the localization loss (loc) (W. Liu et al., 2016). It is computed as:

$$L(x, l, g) = \frac{1}{N}\Big(L_{conf}(x, c) + \alpha\, L_{loc}(x, l, g)\Big) \tag{5.9}$$

where N is the number of the matched default bounding boxes, if $N = 0$, the loss is set to 0, α is set to 1 by cross-validating the model. The confidence loss (L_{conf}) is defined based on a softmax loss for multiple confident of classes (c).

$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^{p} \log\big(\hat{c}_i^{p}\big) - \sum_{i \in Neg} \log\big(\hat{c}_i^{0}\big),$$

$$\text{where } \hat{c}_i^{p} = \frac{\exp\,(c_i^{p})}{\sum_p \exp\,(c_i^{p})} \tag{5.10}$$

The localization loss (L_{loc}) is a smooth loss between the parameters of the predicted box (l) and the ground truth bounding box (g) where the centre of the default bounding box (d) is (cx, xy) and its width (w) and height (h). It is computed as:

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,xy,w,h\}} x_{ij}^{k}\, smooth\, L_1(l_i^m - \hat{g}_i^m) \quad (5.11)$$

We have used the weights of an open-source pre-trained SSD model implemented using the Pytorch library in Python programming (deGroot and Brown, 2017). The goal was not to train a new deep learning model for object recognition to achieve better results but rather to adapt the current state-of-the-art model to the other algorithms of the URBAN-i model.

In order to track the detected multi-objects in the sequential frames, we adopted the SORT method (Bewley et al., 2016). The advantages of using this method, besides its high performance, can be summarised in two main aspects: First, Its inference speed allows faster computation for the overall methodology without dropping multiple FPS. Second, Its simplicity for implementation without previous training is crucial for deploying the overall pipeline of the introduced framework. The objectives of the SORT method is threefold: 1) to propagate the states of an object into future frames, 2) correlating the detections of the current phase with the existing objects, and last 3) managing the lifespan of the bounding boxes of the tracked objects. To achieve these objectives, the SORT method relies on Kalman Filter (Kalman, 1960) and Hungarian algorithms that use a series of measurements observed over time to infer unknown variables of the displacements of an object in sequential frames. This identity of an object and displacements of the inferred frame is based on a linear constant velocity model, which is independent of the motions of other objects and camera movement. The state of each object can be identified as:

$$X = [\,\mathrm{u}, \mathrm{v}, \mathrm{s}, \mathrm{r}, u', v', s'\,]^T \quad (5.12)$$

where u, v denote the coordinates of a pixel centred at the object, S and r denote the scale and aspect ratio of the bounding box of an object, respectively. It is worth mentioning that the aspect ratio is considered a constant. When an object is detected, the detected bounding box is used to update the object state by solving the velocity component via a Kalman filter method.

The method is evaluated when an object enters and leave the image by two metrics: 1) Similar to object detection evaluation with the Localisation loss introduced in **equation 5.11**, Intersection Over Union (IOU) method is used, in which detection is considered only when the overlap is less the IOU_{min} and the tracking is initialised by the dimensions of the detected bounding box with velocity set to zero. After initialisations, the covariance of the velocity is evaluated by a loss metric (T_{loss}) that penalises the linear inference of the velocity after initialisation.

5.5.2 Data

In the case of object detection, the weights of the base network of the model, the VGG16 model, has been trained on the ILSVRC CLS-LOC dataset (Russakovsky et al., 2015). After truncating the base network by converting the last fully connected layers to convolutional layers and adapting its network with pre-discussed changes, the model is trained on PASCAL VOC 2007 dataset for image recognition (Everingham et al., 2015). For strengthening the model performance for different object sizes, a data augmentation technique has been conducted. For each image, the model computed several random

samplings based on various techniques, such as sampling the patch so that the minimum Jaccard overlapping the object is either numerically defined or randomly sampled.

5.5.3 Results

The average precision of the object detection model computed on a test set of PascalVOC dataset with a threshold of 0.1 is 71.51% for an Intersection Over Union (IOU) of 0.5. **Table 5.6** shows the average precision for each object class.

Table 5.5: Diagnoses of the Object detection models for the test sets.

Detected class	Person	Bicycle	Car	Bus	Motorbike	Truck
AP[1] (IOU=0.5)	0.75	0.79	0.81	0.77	0.81	0.77

[1]The average precision is calculated with an IOU value of 0.5

In the case of tracking, the SORT method is evaluated on Multi-Object Tracking (MOT) benchmark datasets (Leal-Taixé et al., 2015). The dataset consists of various sequential urban scenes taken by still and moving cameras. MOT 2015 -training set comprises 5500 sequential frames, with 39,905 bounding boxes of multi-objects, whereas the test set comprises 5,783 sequential frames, with 61,440 bounding boxes. After training the SORT model, **Table 5.7** summarises the model results. It achieved 33.4 Multi-Object Tracking Accuracy (MOTA) on the test set and Multi-Object Tracking Precision (MOTP) of 0.72. Besides its high performance, the model can be implemented in real-time, making it a reliable and fast-tracking system.

Table 5.6: Diagnoses of SORT model
The table is adapted from (Bewley et al., 2016).

Method	Type	MOTA	MOTP	False alarm/ frame
SORT	Online detection[1]	33.4	0.721	1.3%

[1]The average runtime of the model is 20 fps on GPU Titan v and CPU i7

5.6 Spatio-temporal data extraction

This sub-model deals with the extraction of the coordinates and the time data of where and when the urban scene images are taken. This information is extracted from the Exchangeable image file format (EXIF) data that is accompanied by an image that features GPS data. This will allow capturing changes of the urban world according to a wide range of temporal scales, from the scale of year to even a second, to cope with the nature and the interdisciplinary of urban modelling tasks.

The algorithms define different functions to extract the coordinates, date and time, where the URBAN-i model can iterate through images to identify and extract these data and write it to a file besides the data captured from the aforementioned models.

To extract the geographical coordinates, time, and date data, we have defined three functions that deal with each task separately. First, we have defined a function to extract the X and Y coordinates and convert them into latitude and longitude. Second, we have defined an array to extract date (year, month, and day), and Last, we have defined an array to extract time (hour, minute, second). The Python code is adapted and modified based on Sandler (2011).

5.7 What makes the integrated models the state-of-the-art

Cities are complex systems by nature, in which the dynamics of their appearance is highly influenced by multiple factors. Weather, visual and surface conditions are some of these prominent factors that not only impact the appearance of cities but also complicate the process of understanding them. In this chapter, we introduced the first two phases of the overall framework to tackle the variations and the dynamics of cities' appearances from the perspective of weather, visual and ground conditions, in addition to detecting road users. From a single street-level image of an urban scene, the framework can capture information related to visual conditions such as dawn, dusk, day or night time, in addition to detecting glare. While on the other hand, the framework can detect weather conditions such as clear, fog, rain, and snow. It can qualify the conditions of the road surface, in addition to detecting and counting road users that appeared in the scene. **Fig. 5.12** shows a sample of testing images of various urban settings, visual and weather conditions.

Figure 5.12: Samples of testing images using the framework

Several points can be discussed in detail regarding the introduced individual models. First, SlipNet is the first deep model that detects the conditions of the road surface from images, regardless of the weather conditions. Second, the innovation of the WeatherNet, in comparison to the current state-of-the-art, can be seen in three aspects:

1. The framework can tackle various weather and visual states, including detecting glare, which has never been tackled in previous deep learning and computer vision research. By using a unified and simple method, the WeatherNet framework is capable of classifying day or night-time, glare, fog, rain, and snow. Most of the previous models recognise only a limited number of weather conditions, neglecting other vital factors.
2. Unlike the current weather recognition models, the proposed framework does not require any pre-defined constraints such as applying filters, defining a camera angle, or defining an action area to the processed image. This simplicity of input makes the proposed framework user-friendly and a base for practical applications for both computer scientists and non-computer scientists to capture information related to weather and appearance of cities from user-defined datasets of street-level images.
3. Although weather and visual conditions depend on time and space, there are no weather stations in each location in cities, and the data forecasted and captured rather represent the agglomeration of locations rather than a precise condition for each location. This undermines the dynamics of the visual appearance of cities. Accordingly, the proposed framework captures weather and visual information. This can enable city planners to map the dynamics of cities according to their weather and visual appearance, which can be a useful tool to understand the dynamics of the appearance of locations and the impacts of these weather and visual dynamics on other aspects of cities (i.e. understanding locations in cities that most likely to cause accidents or risks under certain weather and visual conditions).

5.8 Summary

In this chapter, we presented new computer vision tools that can be utilised to model and sense the dynamics of urban areas. Besides contributing to tackling cycling near misses, these introduced tools exemplify the application of AI and deep learning in understanding cities. We present a novel framework that includes WeatherNet, The SlipNet and object detection and tracking models to detect and map weather, visual and surface conditions from single-images relying on deep learning and computer vision. After training several deep models, we obtained a validation accuracy of 91.6% or higher for all trained classifiers. For instance, the WeatherNet model can detect ten classes: Dawn/dusk, day, night, glare, no glare, fog, no fog, clear, rainy, and snowy weather. We aimed to exemplify the application of deep learning and computer vision for scene awareness and to understand the dynamics of the appearance of urban scenes that could be useful for autonomous applications in cities or elsewhere.

6
ACTION RECOGNITION

6.1 Overview

This chapter introduces a novel method called CyclingNet for detecting cycling near misses from video streams generated by a mounted frontal camera on a bike regardless of the camera position, the conditions of the built environment, the visual conditions and without any restrictions on the riding behaviour. CyclingNet is a deep computer vision model comprising algorithms based on convolutional structure embedded with Self-attention bidirectional Long-short Term Memory (LSTM) blocks that aim to understand near misses from both sequential images of scenes and their optical flows. The model is trained on scenes of both safe rides and near misses. After 42 hours of training on a single GPU, the model shows high validation on the training, validation and testing sets. The model is intended to be used for generating information that can draw significant conclusions regarding cycling behaviour in cities and elsewhere, which could help planners and policy-makers to better understand the requirement of safety measures when designing infrastructure or drawing policies. As for future work, the model can be pipelined with other state-of-the-art classifiers and object detectors simultaneously to understand the causality of near misses based on factors related to interactions of road users, the built and the natural environments.

The materials of this chapter are published as a journal article in the *IET intelligent transport systems journal*, entitled: *"CyclingNet: Detecting cycling near misses from video streams in complex urban scenes with deep learning"* (Ibrahim et al., 2021).

6.2 The overall framework map

This chapter covers the third phase of the overall methodology introduced in **Chapter IV**. **Fig. 6.1** shows a thumbnail of the overall methodology, highlighting the study area covered in this chapter. The input of this phase is the sequential frames of the video streams captured while cycling in urban areas.

The main contributions of this chapter are:

- Automating the detection of cycling near misses in a near real-time detection.
- A novel end-to-end deep model for recognising cycling near misses from untrimmed video streams in complex urban settings.
- A human-labelled large scale dataset for classifying video streams of moving bicycles, at a frame level, of near misses and safe rides.
- A comprehensive set of experiments to evaluate the different architectures of deep models that can be used as a baseline for future research in this study domain.

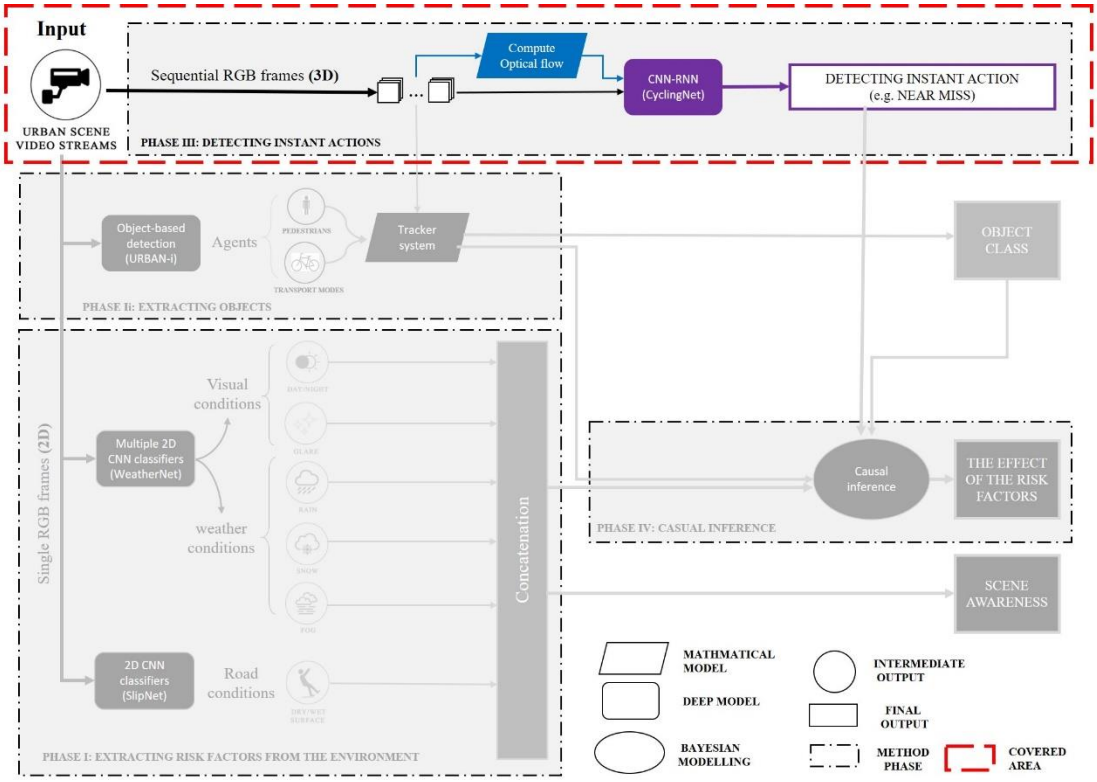

Figure 6.1: Keymap of the overall methodology covered in this chapter

6.3 Model requirements

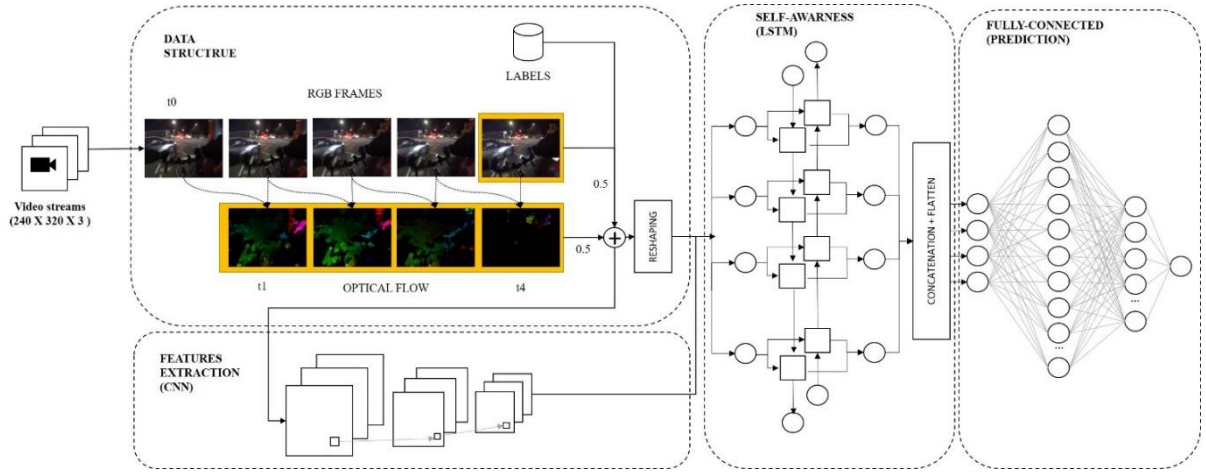

Figure 6.2: The architecture for the proposed CyclingNet

As stated **in Chapter III**, we define a near miss as *"a situation in which a person on a bike was required to act to avoid a crash, such as braking, speeding, swerving or stopping. In some cases, the definition may be extended to include those events that caused the person on the bike to feel unstable or unsafe, such as a close pass or tailgating"*. In order to identify these events, a computer vision algorithm must be capable of distinguishing such a set of instant actions from normal riding behaviour, which may also include actions similar to those taken during a near miss.

Near misses can be seen as instant actions that take place by other objects in the scene. Accordingly, there are three main elements that the model needs to learn in order to recognise near misses: 1) The relative motions of the elements in the scene, 2) the spatial structure of the scene, and 3) memory to recognise what happened in the past.

Subjectively, understanding the change in motion could lead to a better way of understanding the actions related to both safe and unsafe rides since each object conserves its motion between consecutive frames and neighbouring pixels are more likely to conserve similar motion. Accordingly, combining street-level frame images with their optical flow for a number of consecutive frames may lead to a better approach to recognise near misses from video streams.

6.4 Model architecture

In order to respond to the aforementioned requirements, we propose the CyclingNet model. The CyclingNet is a novel single-stream spatio-temporal deep model that is trained in an end-to-end fashion. It aims to include the features of two-streams networks by including the spatial and temporal aspects of the video stream while providing an inference in near real-time similar to the single-stream networks. Its algorithms comprise four main sections; data structure, feature extraction, self-awareness, and integration and prediction. **Fig. 6.2** shows the order of the main algorithms and how the model is structured.

6.4.1 Data structure

As inputs, the model takes two types of data, which are both are resized into 240 X 320 X 3 tensors of single-frame images. The first input comprises video streams typically produced by cameras mounted on a moving bike, which may have varying angles, fields of view, rider speeds and filtering processes (e.g. for stabilisation or extraction of a region of interest). The second input comprises a computed dense optical flow for each pixel in two consecutive frames. It is computed as follows:

For a given pixel $P_{(x,y,t)}$that moves a (d) distance of (dx, dy), the change in P, assuming that P does not change its intensity, can be calculated as:

$$P_{(x,y,t)} = p(x + dx, y + dy, t + dt) \quad (6.1)$$

By dividing the right side with dt and using the Taylor approximation technique, we estimate the optical flow as:

$$f_x u + f_y v + f_t = 0 \quad (6.2)$$

given that: $f_x = \frac{\partial f}{\partial x}$, $f_y = \frac{\partial f}{\partial y}$, $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$

Where (f_x) and (f_y) are the image gradients, (f_t) is the gradient over time, whereas (u) and (v) values are unknown.

To solve this equation with several unknown gradients, we used Gunner Farneback's algorithm (Farnebäck, 2003), in which he approximates each neighbourhood by a quadratic polynomial. Consequently, a new signal can be constructed based on a global displacement, in which it can be computed based on equating the coefficients of the yields of the quadratic polynomials.

The outputted optical flow vectors (u, v) are an array of two channels, in which it can be visualised in a colour image, given that its magnitude can be presented based on the value plane, and direction can be presented based on a Hue value of the image.

After computing the optical flow $(f_{o\prime(t)})$ for a given time (t), the data of the RGB images (f_{rgb}) are truncated for each video file to start with the 4th frame in the frame sequence and wrapped with the four timestamps of the frames of optical flows[t_i, t_{i-1}, t_{i-2}, t_{i-3}] in a portion of 0.5 to 0.5, respectively. The input $(x_{(t0)})$ is defined as:

$$x_{(t_i)} = \frac{f_{rgb\prime(t_i)}}{2} + \frac{f_{o\prime(t_i)} + f_{o\prime(t_{i-1})} + f_{o\prime(t_{i-2})} + f_{o\prime(t_{i-3})}}{8} \quad (6.3)$$

There are two reasons for selecting and optimising these hyperparameters (the proportions and count of previous optical flow frames): Firstly, to add the time dimension to the spatial structure of each street-level image, and secondly, to control and reduce the information and the number of features and textures that are not useful for detecting near misses (i.e. the textures of people, cars, building, etc.). We experimented with the values of the combined ratio based on trial and error to optimise the overall fitness and performance of the model when detecting near misses.

The output data is structured and reshaped in four-dimensional tensors (timestamps, width, height, channels), in addition to embedding the four optical flow steps with the single-frame images. Such an approach means the dimension of time can be utilised and seen either in the spatial structure of the image (fusion with four previous steps of the optical flows) or the series of the data (the length of timestamps). Both approaches will be utilised and discussed thoroughly in the algorithms of CyclingNet in the two upcoming sections.

6.4.2 Extracting features

This part of the model aims to extract mainly spatial features from the single-frame images, bearing in mind the fused data of the optical flows of the previous four steps. The architecture of this section comprises three consecutive blocks of convolutional structure representing the encoder component of the model. Each block has different sets of structure and hyperparameters and initialised by the "He normal" initialisation technique to provide more efficient and faster gradient descent (He et al., 2015b). Generally, the choices of the presented hyperparameters are made based on trials and errors and the most common practice for training Convolutional models. For instance, Simonyan and Zisserman (2015) reported that for the same depth networks, a larger kernel of convolution (3 X 3) performs better than a smaller one (1 X 1). A Batch-normalization layer is a common approach for accelerating the training of deep networks (Ioffe and Szegedy, 2015). It is worth mentioning that the choice of the number of layers of the encoder is a trade-off between extracting features and the computational limitations for training the entire model as a single-stage model. Moreover, models with different hyperparameters and layer components will be trained and presented as base models for further evaluating the introduced methods in the results section (**Section 6.7.2**).

Block one consists of two 2D convolution layers of a kernel size (24 X 5 X 5), (36 X 5 X 5) respectively, and a subsampling size of (2 X 2). They are activated based on a Rectified Linear Unit (ReLU). These two CNN layers are followed by a 2D Max-Pooling layer of pool size of (2 X 2) and a Batch-normalization layers of the momentum of 0.99 and epsilon of 0.001. It is feed with single-frame images with the embedded optical flow steps.

Similar to block one, block two consists of two 2D Convolution layers, however, a kernel size (48 X 5 X 5), (64 X 3 X 3) respectively, and a subsampling size of (2 X 2). They are activated based on a Rectified Linear Unit (ReLU). They are also followed by a 2D Max-Pooling layer of pool size of (2 X 2) and batch-normalization layers of the momentum of 0.99 and epsilon of 0.001.

Block three consists of a single convolution layer of a kernel size (128 X 3 X 3) subsampled with (2 X 2) and activated by a ReLU function. It is also followed by a 2D Max-Pooling layer of pool size of (2 X 2) and a Batch-normalization layers of the momentum of 0.99 and epsilon of 0.001.

6.4.3 Spatial and temporal awareness

If the algorithms of detecting near misses rely only on the outputted features of the previous section of the convolution structure, based on experiments, the results will be sensitive to the changes at the local context of the spatial structure of the fused single frames, despite the significances of the global context that can ensure stability and accuracy for training and inference. For this reason, designing the architecture of CyclingNet further to be aware of both local and global spatial and temporal

structure is indispensable. This temporal dependent element makes it a crucial part of detecting temporal features related to scenes of near misses.

This section comprises one bidirectional Long-Short term Memory (LSTM) block, followed by a regulated self-attention layer. The LSTM block consists of 128 units, and a dropout regulation of size 0.3 to avoid over the fitness of the model. However, the goal is not only considering the sequence of the defined timestamps but also considering the context for each timestamp. Therefore, a self-attention mechanism is essential to ensure the balance for both global and local context when describing a given scene.

Generally, a unidirectional LSTM has shown great progress in extracting features related to sequential data to predict future states (Goodfellow et al., 2017; LeCun et al., 2015). Unlike the traditional recurrent layer, LSTM can learn long-term dependencies without suffering from issues related to gradient vanishes. This internal recurrence, the so-called self-loop, enabled the previous vectors to create paths in which the gradient can move forward for a long duration without vanishes issues. Nevertheless, most recently, it has also been shown to improve the overall performance of the model when predicting even a given state without timestamps by learning not only the spatial structure of a given vector but also the short-term dependences among the inputted given vector as the time constants are outputted by the LSTM itself. Accordingly, this allows the time scale to change based on the input sequence, even if the LSTM units are with a fixed parameter.

To extract long-term dependences, the self-loops of the LSTM units can be controlled by three gated units: 1) forget gate ($f_i^{(t)}$), external input gate ($g_i^{(t)}$), and an output gate ($q_i^{(t)}$).

First, ($f_i^{(t)}$) can be explained for a given cell (i) and time (t), whereas it is fitted to a scaled value between 0,1 and an activation unit of sigmoid function (σ) as:

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right) \quad (6.4)$$

given that $h^{(t)}$represents a vector that contains the outputs of all the LSTM cells for the current hidden layer, $x^{(t)}$represents the current input vector, W^frepresents the recurrent weights for the forget gates, U^f represents the input weights and last, b^f represents the biases of the forget gates.

Second, to update the LSTM internal state, a conditioned weight of the self-loop ($f_i^{(t)}$) is computed as:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)})\right) \quad (6.5)$$

given that U is the input weights, b is the bias vector, W represents the current weights into the LSTM cell. Similar, to ($f_i^{(t)}$), the external input gate ($g_i^{(t)}$) is computed, however with it is a parameter:

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right) \quad (6.6)$$

Last, the output gate ($q_i^{(t)}$) is used to control and shut off the LSTM cell output ($h_i^{(t)}$) with a sigmoid unit, in which the $h_i^{(t)}$ is defined as:

$$h_i^{(t)} = \tanh(s_i^{(t)}) q_i^{(t)} \tag{6.7}$$

$$q_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right) \tag{6.8}$$

given that b^o is the model biases, U^o is the input weights, $W^o$is the current weight.

Unlike unidirectional LSTM units, a bidirectional LSTM layer allows the current hidden state to rely on two independent hidden states, one computed in a forward direction, named as a forward LSTM, and the latter is defined as a backward LSTM in the opposite direction. This allows the retaining of the historical information and the current ones simultaneously. This has a direct implication when detecting near misses, in which the predicted output for a given state is smoothed when compared to the previous ones without any post-prediction smoothing techniques.

Moreover, adding a self-attention mechanism to the bi-directional LSTM units allows the model to learn not only from the extracted features – whether spatial or temporal ones- but also to learn from the relations of the input sequences of the RGB image and optical flow ones by allowing the model to relate the positioning of each sequence and accordingly, learns the representation of its input (Goodfellow et al., 2017; Vaswani et al., 2017). Nevertheless, the model can learn which context to consider for a given scene to output the prediction (Xu et al., 2015). The context (l_t) can be computed as:

$$l_t = \sum_{t'} a_{t,t'} x_{t'} \tag{6.9}$$

given that:

$$h_{t,t'} = \tanh(x_t^T W_t + x_{t'}^T W_x + b_t) \tag{6.10}$$

$$e_{t,t'} = \sigma(W_a h_{t,t'} + b_a) \tag{6.11}$$

$$a_t = softmax(e_t) \tag{6.12}$$

where ($h_{t,t'}$) represents the hidden state of the previous step – in a given direction of the bidirectional LSTM- that is fitted to a simple forward neural model ($e_{t,t'}$), (a_t) is the amount of attention that the output at a given state should consider for the previous activation (σ).

6.4.4 Model initialization

After the LSTM block, the output is flattened and feedforward to two fully-connected layers of 180,64 neurons respectively. Both layers are activated by a ReLU function, in which a Dropout mechanism is applied for both layers with a size of 0.3. The final output layer consists of a single neuron and is activated with a sigmoid function.

6.5 Evaluation metrics

The model is penalised during training, testing and validation based on a cost function of Cross-Entropy of error. It is defined as:

$$E = -\sum_i^n t_i \, log(y_i) \tag{6.13}$$

given that $t_i t_i$ represents the target vector, $y_i y_i$ represents the predicted vector, and n represents the binary classes.

For further assessing the model performance, we computed accuracy, precision, recall, false-positive rate, and F1-score as explained in **Chapter V – 5.3.4.**

Last, it remains a challenge to compare our results with other models due to the absence of other models for detecting near misses for moving cycling from the street level. We created, however, different architecture to draw a baseline for the performance of the proposed method and to show how the different architecture and hyperparameters could yield different outcomes for a given task with the same material types.

6.6 Materials and data pre-processing

To the best of our knowledge, there is no benchmark data set of video streams that focus on the different types of cycling near misses that is open-sourced to conduct computer vision research. Therefore, collecting our own dataset becomes crucial to train the model to detect near misses in complex environments. We collected video clips that were made available online by people on bikes on two websites: YouTube and road.cc. In these clips, near misses are labelled manually in the embedded frames by the sharers, which represent the ground truth of the model. Two aspects make this data a significant one for understanding near misses: First, the variation in the perceptions of near misses as defined by the clips sharers. This could allow the model to extract features related to the common trends instead of being heavily directed or biased with a small group of participants or self-labelling. Second, the variation of equipment, camera position, context, visual, and weather conditions, along with the different behaviours and riding styles in these scenes, are crucial for the learning process of the model, generalisation, and deployment.

After qualitatively inspecting the quality and ground truth of the embedded information of the selected clips, we collected a dataset of 74,477 sequential frames, and we computed their equivalents of optical flows frames (74,469). Of these 8,567 sequential frames belong to near miss cases (11.5% of the total sequential frames) which occur at sparse intervals. They represent 209 unique near misses of an average duration of 1.3 seconds (40.9 sequential frames). We also used an additional dataset of 12,812 sequential frames for further testing after training and validation. This dataset comprises 81 unique near miss events.

These clips include complex urban settings of different visual and weather conditions and a variety of scene components. For example, 81.9% of the scenes in the dataset are during the daytime, 15.7% at night, and 2.4% at dawn/dusk time. Also, the dataset includes around 93.6% of scenes of clear weather, 5.9% rainy weather, and 0.5 % snowy weather. Around 2.5% of the dataset includes foggy scenes, 7.9% with glare, and 37.5% are scenes that include a cycling lane. 93.7% of scenes include

other humans, 46% includes scenes that comprise other cyclists, 67.9 % are scenes that include at least one car, 23% with one bus or more, and 12.6% with one truck or more.

The clips also consist of variations of near miss types of different temporal scales (the duration of near miss) and various interactions with different road users. The clips, for instance, include near misses such as a close pass, a near left or right hook, someone pulling in or out, swerve around an obstruction, pedestrian steps out, and someone approaching head-on. However, there is a lack of clips that include near-dooring and tailgating events. **Fig. 6.3** shows a sample of the sequential frames and their corresponding optical flows.

Data augmentation techniques, introduced in **Chapter V – section 5.3.3**, have also been utilised for this model for enhancing the training process and accuracy of the model and accounting for the class imbalance of safe and near miss scenes. However, we augmented the collected data by applying only normalisation, scaling, and horizontal flipping without applying shear to avoid any distortion to the overall motions of the sequential frames.

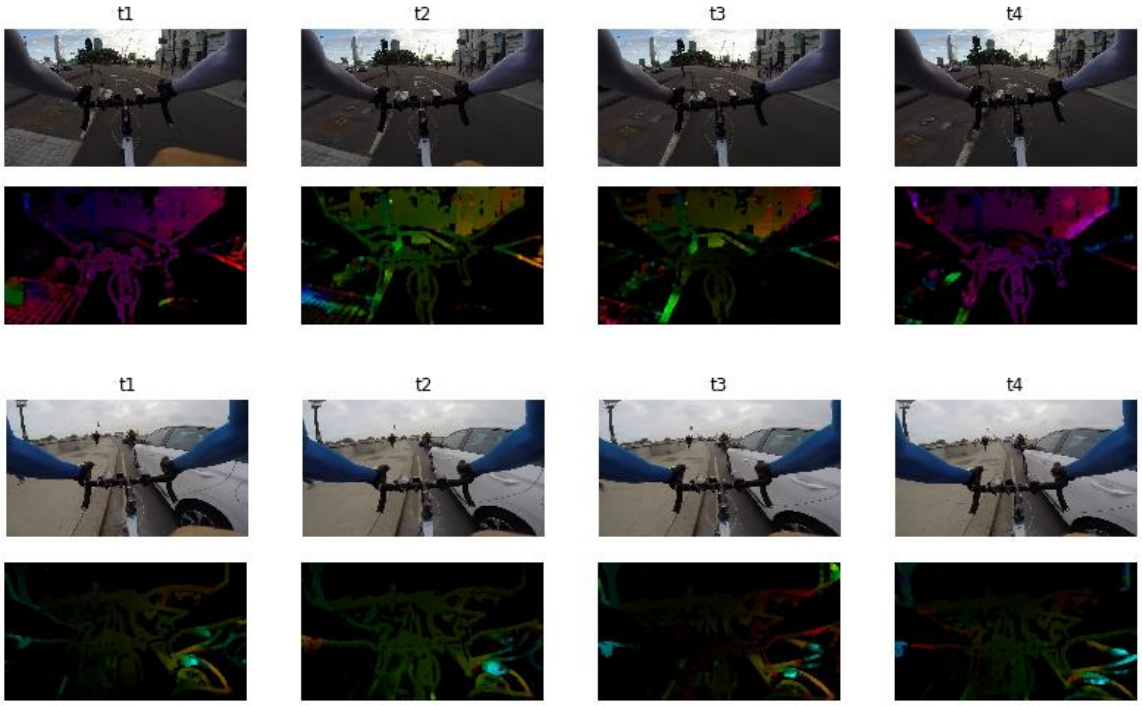

Figure 6.3: Sample of the dataset for the RGB frames and their optical flows

6.7 Results

6.7.1 CyclingNet evaluation

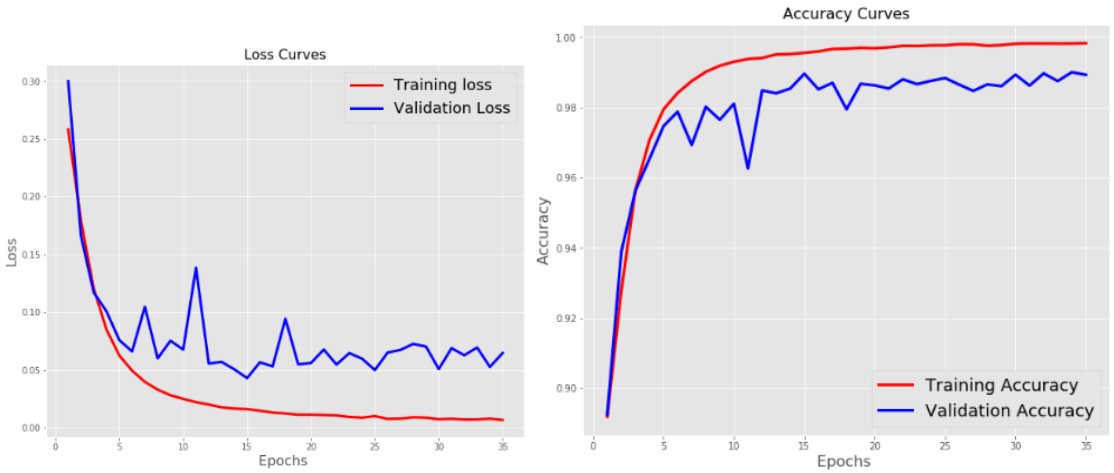

Figure 6.4: Training and evaluation of cyclingnet.
from left to right: (A) and (B)

Training the model on street-level images of both safe rides and near misses took almost 2 days (42 hours) on a single GPU (Titan V). **Fig. 6.4a -6.4b** show the losses and accuracies of the training and testing sets respectively for the Self-attention bidirectional CNN-LSTM architecture. After 35 training cycles of 100 ones, the model has converged and the training has stopped to avoid over-fitness after no significant change on the validation loss. In **table 6.1**, we expand further on evaluating the classification of CyclingNet. The table shows a high validation in terms of precision, recall, and an F1-score, with minimum false-positive rates. The model shows a high validation in terms of true positive of the area under the curve of 0.99, 0.84 for validation and testing sets respectively. However, the gap between the values of the validation and testing sets can be explained due to the wide range of variations of near miss events, or the limitations of similar events that the model can learn and extract features from them for future inference.

Table 6.1: Classification metrics for CyclingNet

Self-attention Bi CNN-LSTM	Precision	Recall	False-positive rate	F1-score
Validation set	0.994	0.995	0.041	0.994
Test set	0.842	0.927	0.418	0.883

6.7.2 Baseline evaluation

We experimented with adjusting the optical flow to images fusion ratio, model architecture, an optimisation technique, and post-prediction with the classification thresholds aiming to maximise temporal smoothing while reducing the global loss. We found that the global loss can be reduced even by a simple CNN architecture, however, the predicted values are prone to temporal instability. On the

contrary, after applying a CNN-LSTM architecture the temporal dependences improved whereas the model outputs a smoothed prediction throughout the clip. We also found that by including bidirectional and self-attention mechanisms in the architecture of the CNN-LSTM model, the losses at the local and global levels of the training and testing datasets have improved in comparison to a CNN-LSTM model. **Table 6.2** summarises the outcomes of the different studied architecture on the validation set, with a constant fusion ratio of 50% of the single images and optical flows.

Table 6.2: Baseline assessment of CyclingNet

Architecture comparison	Validation Accuracy	Validation Loss
CNN (Block 1-3)	86.5 %	0.73
CNN-LSTM	97%	0.15
Self-attention CNN-LSTM	96 %	0.20
Self-attention Bi CNN-LSTM	98.9 %	0.06

6.7.3 Scenes prediction

In **Fig. 6.5**, we show different clips of near misses predicted by CyclingNet. The model shows high accuracy in predicting a wide range of complex urban scenes at different times of the day and weather conditions. Nevertheless, the model shows high accuracy in predicting near misses including different types of near misses, such as close passes, pedestrian step in, or any risky situation with different road-users, including other people on bikes. Similarly, **Fig. 6.6** shows a variety of urban scenes that has been detected as a safe ride.

6.8 CyclingNet as the state-of-the-art method for detecting cycling near misses

Understanding safety as a clue from the overall scene and interaction of different road users remains a challenge. In this chapter, we introduced CyclingNet as a novel method for detecting cycling near misses from video streams of moving bicycles in a complex urban setting. The model has shown strong performance in detecting near misses, regardless of the complexity of the scene, time of the day, weather, visual conditions, or the placement of the camera on the bike. Due to the absence of other models or benchmark datasets for the stated purpose, it remains a challenge to compare our results to other models, besides the ones we developed as base models. This, however, makes The CyclingNet model a vital and indispensable model for the field of road safety and more specifically, for detecting near misses. Accordingly, this makes it good practice for generalisation, deployment, and transfer learning to detect near misses for other road-users or other safety-related domains.

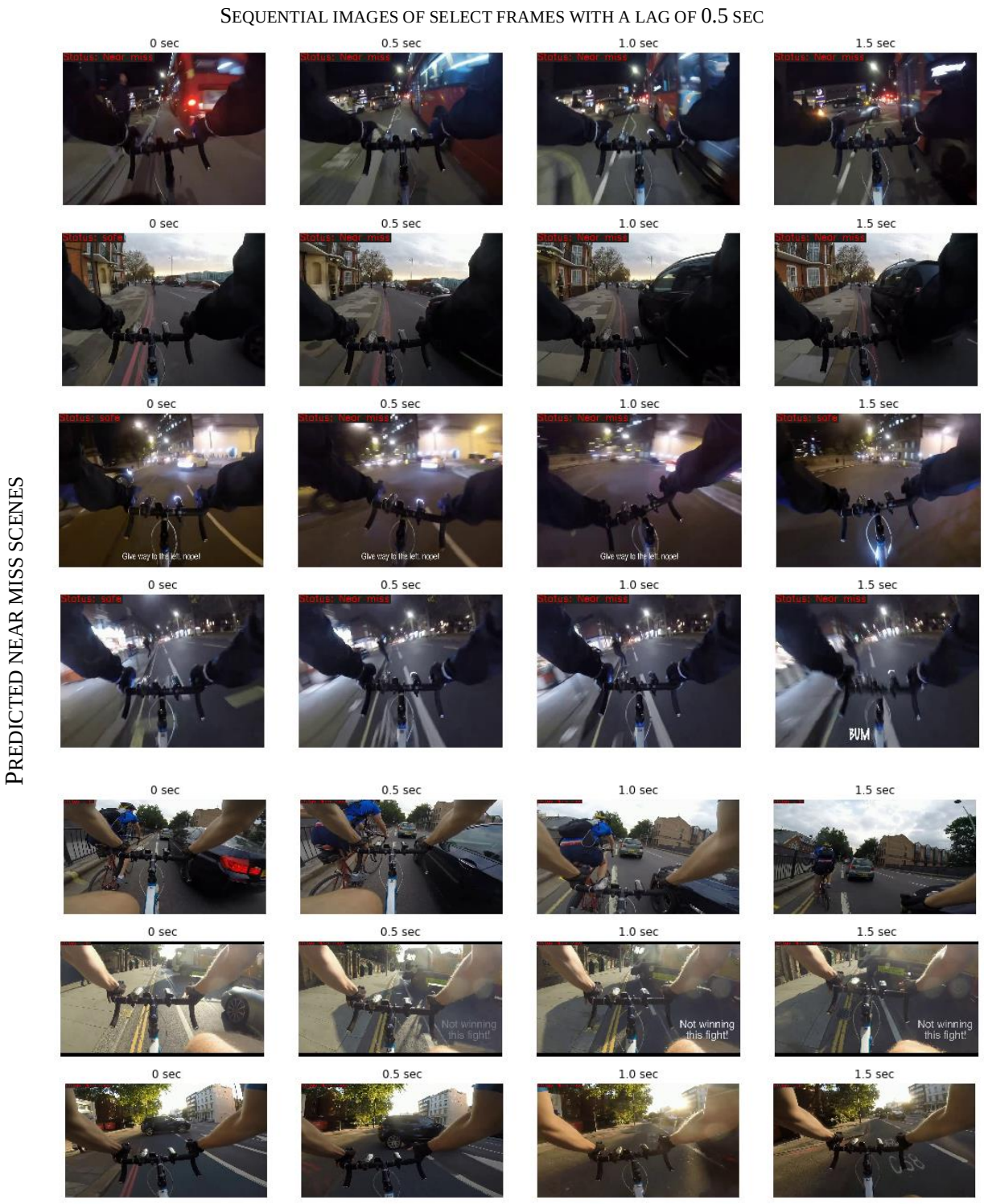

Figure 6.5: Examples of predicted cycling near misses by cyclingNet

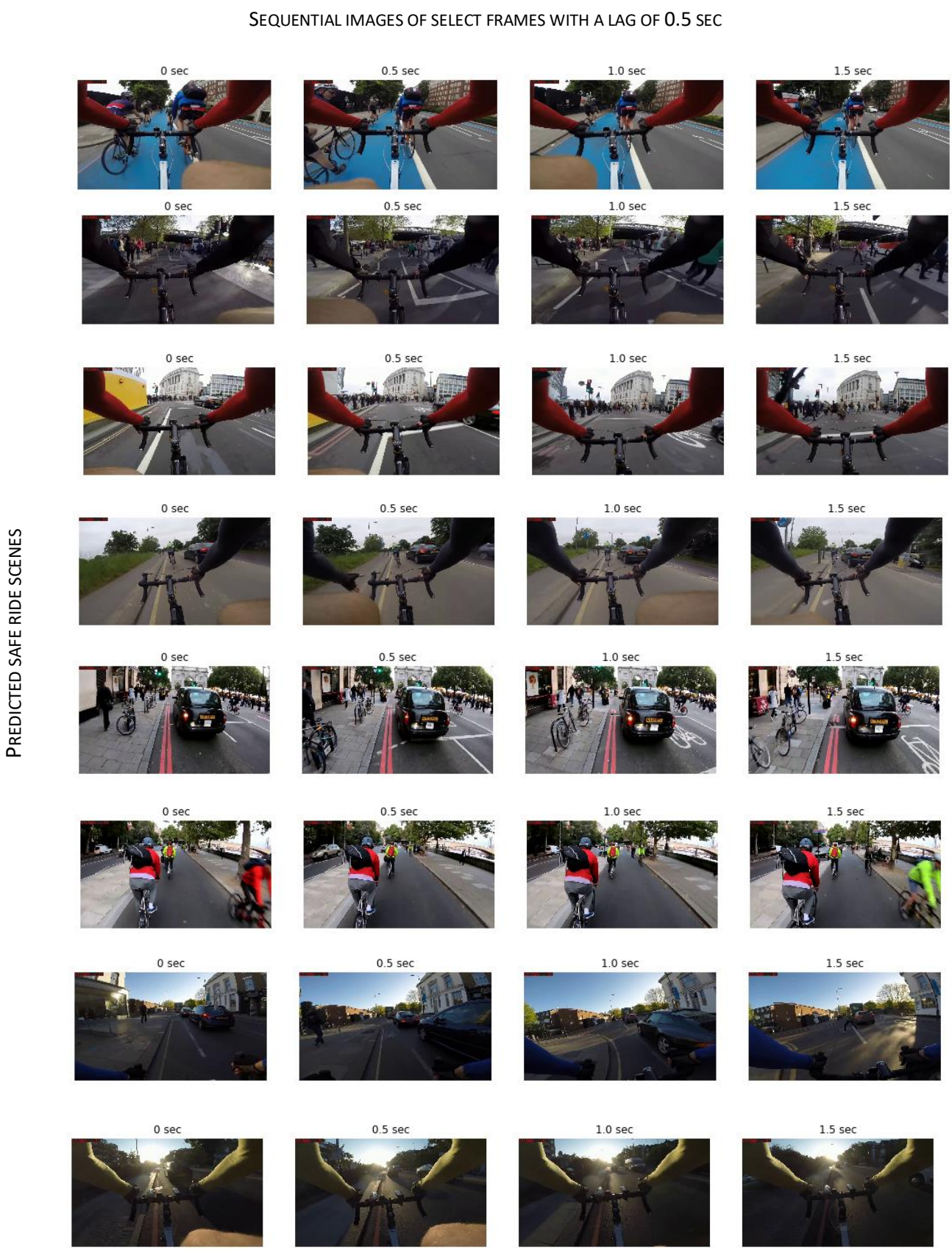

Figure 6.6: Examples of predicted cycling safe rides by cyclingNet

6.9 Model limitations and future work

The model shows a strong performance in the task of identifying near miss frames. However, there are still some limitations that need to be addressed in future work. Firstly, the model is currently applied on a frame by frame basis. In a practical sense, it would be useful to extract near miss scenes in their entirety in order to automatically label events from user-generated data. This could be accomplished by further developing the model to accurately extract the start and end of an event. These events could then be corroborated by the bike user, e.g. by sending them the clip and asking 'was this a near miss'. This approach would also enable the model to learn from experience as it receives a greater set of near miss scenes. Secondly, the model has been trained to identify near-miss events in general but does not currently discriminate between near-miss types. Introducing a new model to classify the different types of near misses after detection would allow a better understanding of the frequency of the different types of near misses and the different risk factors associated with them. For example, a close-pass is likely to have a different set of risk factors to a vehicle turning left at an intersection. Thirdly, the model has been trained using video data from online sources for which there is no information of the cyclist's characteristics (gender, age, experience level etc.). However, near misses are subjective and do depend on these characteristics. For example, an event that a new cyclist considers being a near-miss may not be considered as such by a more experienced cyclist. Furthermore, it is not known whether there are implicit biases in the population of users who upload near-misses to video sharing sites. Social media data has been shown to be biased in many cases and this may affect the range of near misses used to train the model. These issues can be addressed through the naturalistic approach described in chapter III, whereby a representative group of participants use instrumented bikes to collect data on near misses, which can then be linked to their personal characteristics. As the model receives near miss scenes of greater diversity, its generalisation performance will improve. Finally, applying similar models to detect safety measures and near misses for other road users such as pedestrians and car drivers would allow tailored-made policies or guidelines for the interaction of the different road-users according to the specific type of near misses for a given road-user.

6.10 Summary

Within the progress of the field of Artificial intelligence, specifically, the domain of deep learning and computer vision, different deep computer vision models have been developed to recognise a wide spectrum of human actions, activities, or their body poses in complex settings of untrimmed video streams.

The problem of detecting cycling near misses from video streams of moving bicycles in real-world settings has not been previously addressed in the literature. In this chapter, we utilised the advances in computer vision and deep learning to detect such events in a near real-time fashion. We introduced the CyclingNet model, a new deep computer vision for detecting cycling near misses from video streams of moving bicycles in complex urban environments. The model is structured as a single stream and trained in an end-to-end fashion, exploiting both single RGB frames, and optical flow data. After training the model using data of both near misses and safe rides, the results show strong performance on both training and validation data sets.

The model is intended to be used for generating information that can draw significant conclusions regarding cycling behaviour in cities and elsewhere, which could help planners and policy-makers to better understand the requirement of safety measures when designing infrastructure or drawing policies. As for future work, the model can be pipelined with other state-of-the-art classifiers and object detectors simultaneously to understand the causality of near misses based on factors related to interactions of road users, the built and the natural environments.

7
CAUSAL INFERENCE

7.1 Overview

This chapter analyses and quantifies the different risk factors of near misses and describes their contributions to the occurrence of near misses. It addresses 'why' near misses happen based on the set of risk factors extracted in the previous chapters. It covers four aspects: 1) descriptive analysis of near misses and safe ride scenes, 2) the statistical significance of the different risk factors, 3) the cause and effect of the different variables, and last 4) the causality of the different variables in cycling near misses. First, the descriptive analysis presents the data distribution and variable description, and the analysis of the data using non-parametric methods relying on correlation and t-test methods. The second section of this chapter addresses the impacts of the different risk factors on the occurrence of near misses using linear regression as a base model. The third section addresses the issue of class imbalance of the dependent variable. Last, the fourth section addresses the Granger causality in the data, highlighting which factors Granger-cause near misses, or in other words, which factors precede near misses and could assist in forecasting such events in near-real-time. This capability could be used in an early warning system for people on bikes.

7.2 The overall framework map

This chapter describes the final phase of the overall methodology introduced in **Chapter IV**.. **Fig. 7.1** shows a thumbnail of the overall methodology, highlighting the study area covered in this chapter. Besides the stated goals, in the overview section (7.1), the objective of this chapter is to extract the statistical weights of the different risk factors that contribute to near misses and utilise these weights for the framework stringency index introduced in **Chapter IV**.

The structure of this chapter is based on inductive reasoning or as consecutive steps in which the hypothesis, methodology and finding for each section is presented individually to build a foundation for the subsequent sections.

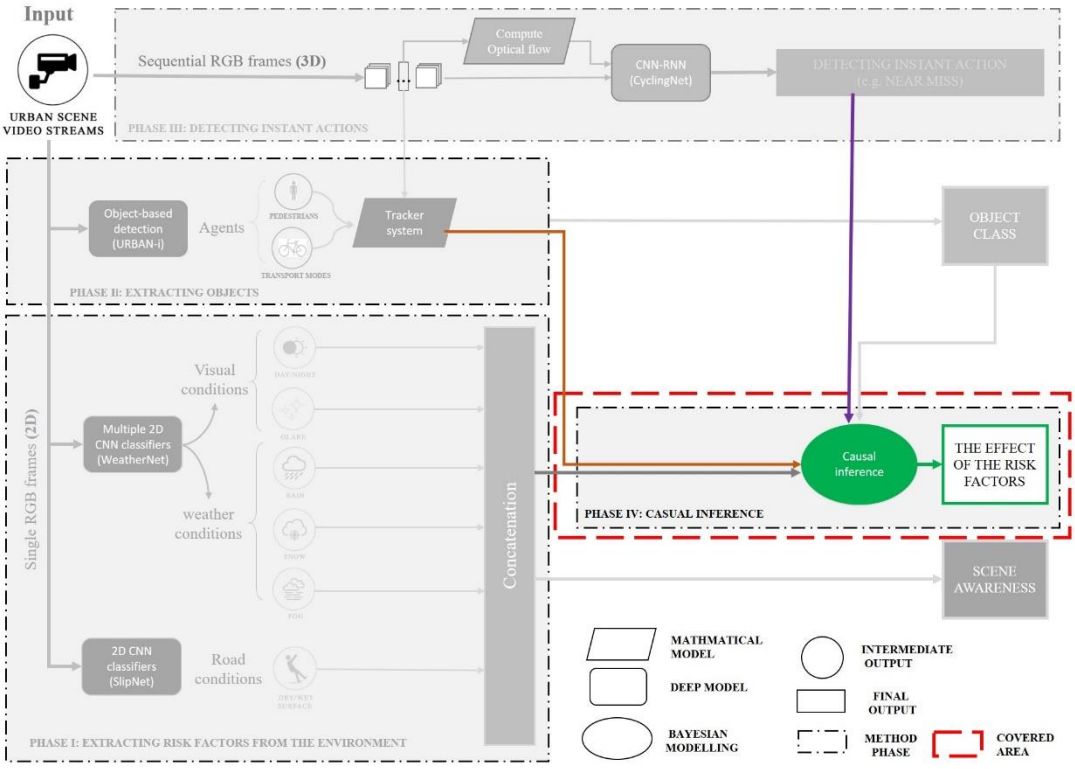

Figure 7.1: Keymap of the overall methodology covered in this chapter

7.3 Materials

The dataset used in this chapter is a subset of the data set used for training, testing and validating the CyclingNet model introduced in **Chapter VI.** It consists of 46,567 sequential frames extracted from video streams of cycling in different built environments, weather and visual conditions. The reason for only including a subset of the dataset is to balance the classes of the different variables of the data, including the observations that are relevant for statistical analysis, which was not necessary to train the deep models in previous chapters. Further data pre-processing techniques (such as random sampling, data transformations, etc.) will be described in the upcoming sections where relevant. There

are two purposes for using this subset: 1) Statistical analysis and quantifying the causes and the effects of the risk factors, 2) assessing the performance of the overall methodology, rather than the individual methods as explained in the overall methodology chapter (**Chapter IV**). The labels of this dataset are generated via the different deep models introduced in the previous chapters (**Chapter V and VI**). To ensure the validity of the results, the generated labels have been manually examined to correct errors. This was accomplished using visual inspection in batches. Accordingly, all presented data points represent the actual classes in the video streams.

Table 7.1 shows the 17 selected variables in the dataset used in this chapter, including their type and a brief description of the variable. Categorical variables have been transformed into dummy variables. For instance, one of the outputs of the WeatherNet model is a categorical variable of three conditions (clear, rain, and snow) and each one has been transformed to a binary variable of value 1 when a given condition exists and zero otherwise. Besides the dummy variables, the selected variables comprise six integer numerical variables describing the counts of the different road users in a given scene, from a person, bicycle, car to bus, motorbike and truck.

Table 7.1: CyclingNet layer structure and hyperparameters

ID	Variable name	Variable type	Variable description
1	Glare	Binary	A value of one represents the presence of glare, and a value of zero represents otherwise.
2	Fog	Binary	A value of one represents the presence of fog, and a value of zero represents otherwise.
3	Cycle_lane	Binary	A value of one represents the presence of a cycle lane, and a value of zero represents otherwise.
4	Person	numerical	A continuous integer value that represents the counts of people in a given data point
5	Bicycle	numerical	A continuous integer value that represents the counts of bicycles in a given data point
6	Car	numerical	A continuous integer value that represents the counts of cars in a given data point
7	Bus	numerical	A continuous integer value that represents the counts of buses in a given data point
8	Motorbike	numerical	A continuous integer value that represents the counts of motorbikes in a given data point
9	Truck	numerical	A continuous integer value that represents the counts of trucks in a given data point
10	Near_miss	Binary	A value of one represents the presence of a near miss, and a value of zero represents otherwise.

11	Nightime	Binary	A value of one represents the presence of nighttime, and a value of zero represents otherwise.
12	Daytime	Binary	A value of one represents the presence of Daytime, and a value of zero represents otherwise.
13	Dawn_dusk time	Binary	A value of one represents the presence of Dawn or dusk time, and a value of zero represents otherwise.
14	clear	Binary	A value of one represents the presence of clear weather, and a value of zero represents otherwise.
15	rain	Binary	A value of one represents the presence of rainy weather, and a value of zero represents otherwise.
16	snow	Binary	A value of one represents the presence of snowy weather, and a value of zero represents otherwise.
17	Wet_surface	Binary	A value of one represents the presence of a wet surface, and a value of zero represents otherwise.

7.4 Descriptive analysis

Table 7.2: Variables Descriptive Statistics

Variable	N	Minimum	Maximum	Mean	First quartile	Median	Second quartile	Std. Deviation	Variance
glare	46567	0	1	0.18	0	0	0	0.384	0.148
fog	46567	0	1	0.03	0	0	0	0.176	0.031
cycle_lane	46567	0	1	0.40	0	0	1	0.490	0.240
person	46567	0	20	2.77	1	2	4	2.445	5.978
bicycle	46567	0	7	0.61	0	0	1	0.912	0.832
car	46567	0	11	1.16	0	1	2	1.330	1.768
bus	46567	0	4	0.18	0	0	0	0.442	0.195
motorbike	46567	0	3	0.06	0	0	0	0.260	0.068
truck	46567	0	4	0.17	0	0	0	0.422	0.178
near_miss	46567	0	1	0.28	0	0	0	0.450	0.203
nighttime	46567	0	1	0.35	1	1	0	0.478	0.228
daytime	46567	0	1	0.60	0	0	1	0.489	0.239
dawn_dusk	46567	0	1	0.04	0	0	0	0.203	0.041
clear	46567	0	1	0.77	1	1	1	0.418	0.174

rain	46567	0	1	0.22	0	0	0	0.414	0.171
snow	46567	0	1	0.01	0	0	0	0.076	0.006
wet_surface	46567	0	1	0.06	0	0	0	0.245	0.060

Table 7.2 shows sample size, minimum, maximum, mean, standard deviation and variance values for each variable. One crucial finding of this table is the distribution of people in the data points. For instance, the person counts in the data points vary from 0-20, with an average of approximately three persons in a given scene. However, the person counts represent the highest variance in comparison to other variables of the value of 5.9, indicating that the data points are far spread from the mean values when it comes to person counts in comparison to the other variables.

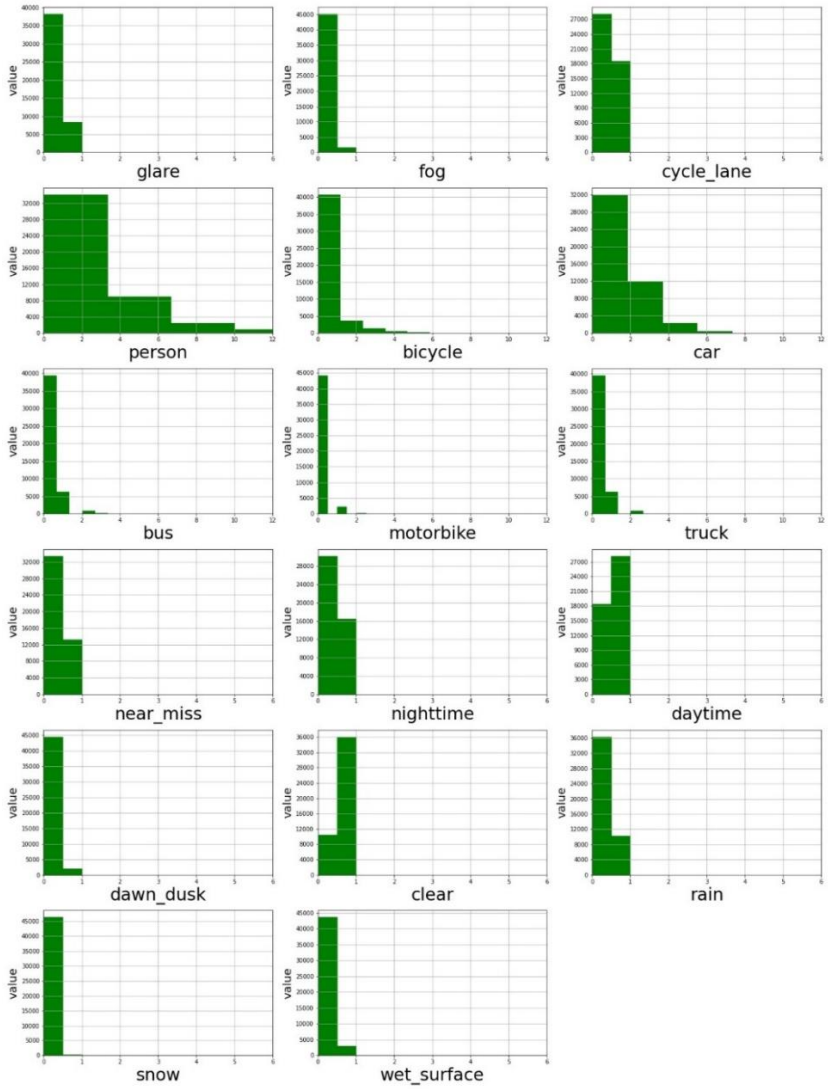

Figure 7.2: Histograms of the variables in the dataset

7.4.1 Data distribution

To understand their distributions, **fig. 7.2** shows the histograms of the individual variables. A class imbalance can be seen in the distribution of variables such as glare, fog, motorbike, dawn/dusk time, snow weather, and wet surface, which requires further statistical treatments to account for the skewed and imbalanced data before conducting any statistical models (Freedman, 2008; Riley et al., 2013).

7.4.2 Correlation between variables

Before running a regression model it is important to check for multicollinearity, which occurs when one or more of the independent variables can be predicted by some combination of the other independent variables. Multicollinearity can cause unstable regression coefficients, limiting the ability to draw conclusions from the model. We have used the Product Moment Correlation Coefficient (PMCC) to highlight the linear correlation in the data set. The PMCC measures the correlation between

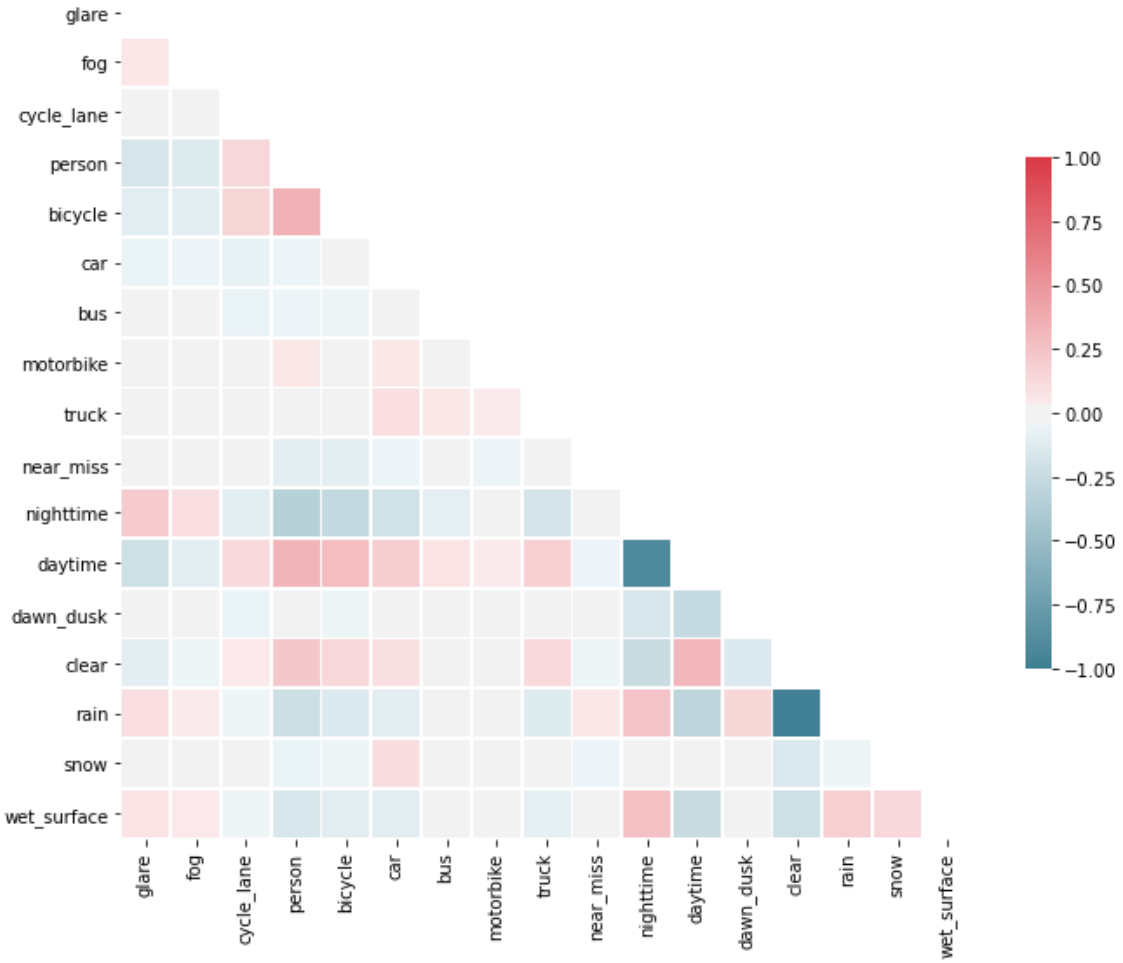

Figure 7.3: The results of the PMCC

two variables in the range [-1,1], where 1 represents a perfect positive correlation, -1 perfect negative correlation and 0 no correlation. For further explanation of the PMCC, see (Frey, 2018).

Fig. 7.3 shows the PMCC between each pair of variables. It indicates a different positive and negative correlation, in which some of them can be considered as new findings, whereas others can be seen as a logical and expected outcome. For instance, daytime is inversely correlated with night-time, and clear weather is inversely correlated with rain weather, which is logical and expected. Similarly, the presence of people is positively correlated with daytime, clear weather, and the presence of bicycles. The presence of glare is positively correlated with night-time, rain and fog. While glare is usually associated with sunny conditions, the detected glare in this dataset is due to headlights in darker conditions such as rain and fog. Wet ground is positively correlated with rain, snow, and night-time.

On the other hand, a crucial finding is that near misses are positively correlated with rain but uncorrelated with daytime. Furthermore, while there is a positive correlation between the presence of a cycling lane and the presence of people and bicycles, there is an absence of correlation between the presence of a cycling lane and the occurrence of near misses. It may seem counter-intuitive that dawn/dusk and daytime are not perfectly negatively correlated, so it is worth mentioning that this is because there are three mutually exclusive classes (Dawn/dusk-time, day-time, and night-time).

7.4.3 t-test method

As a step forward to further investigate the collinearity in the data set among the different variables, we used a t-test to highlight the significant differences between the near miss and safe scenes in terms of the selected variable. The t-test method is used to compare the means of the continuous variables for both groups; safe and near misses. When the p-value is less than 0.05, the null hypothesis can be rejected and the results of selected variables can be deemed statistically significant. It can be used to differentiate between safe and near miss scenes. For further explanation regarding t-test analysis, see Hoffman (2015); Smalheiser (2017).

Table 7.3 shows the statistically significant results of the t-test method for five significant independent variables, in which the near_miss variable is treated as a dependent variable. The results show that the occurrence of near misses is statistically significant with the counts of cars, buses, and motorbikes with positive coefficient values and statistical significance with the counts of people and bicycles with negative coefficient values.

Table 7.3: The significant results of the t-test method

Variable	F-Statistics	P-value
Car	15.497	0.000
Person	-32.137	0.000
Bus	12.192	0.000
Bicycle	-33.540	0.000
Motorbike	2.529	0.011

7.5 The Impacts of risk factors in cycling near misses

7.5.1 Assumptions

Here we focus on understanding directly the role of the different independent variables in the occurrence of near misses statistically in non-controlled experiments. This step can be perceived as a base model for the upcoming sections where we investigate further the role of selected variables after showing a general statistically significant association with the occurrence of near misses.

Fig. 7.4 shows a directed graph of all variables (factors and covariates) towards the dependent variables of near misses. It highlights the hypothesis for modelling the association between the dependent and independent variables. It shows the basic assumption that each variable has a direct association with near misses.

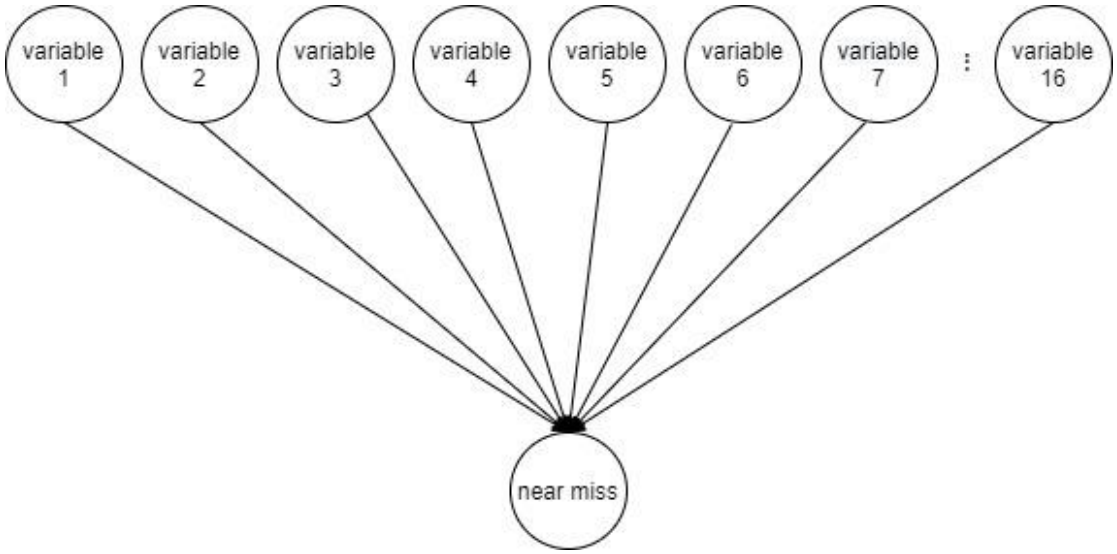

Figure 7.4: The base model assumption

7.5.2 Base model: Logistic regression model

To test the previous hypothesis and assumptions, a Logistic regression model is conducted to analyse the collinearity of the near_miss variable, as a dependent variable, with the other variables as independent variables, without any confounder assumptions nor a controlled variable. This model can be seen as a base model for further controlled studies in the following sections. For further explanation regarding logistic models and utility functions, see Ben-Akiva et al. (1997); Schroeder (2010).

The utility function of the near_miss category i in the occurrence of j is computed as:

$$v_{ij} = \varepsilon + \sum_{k \in T} b_k x_{ijk} \quad (7.1)$$

where x_{ijk} represents the attribute k for point j on near miss occurrence of i, b_k is a coefficient in the utility function, T represents the set of attributes, ε represents the stochastic part of the utility function.

The coefficient of the model is computed by estimating the maximum likelihood, whereas the stochastic part ε is computed by assuming it as a double exponential distribution. The logarithm of the likelihood of the model of the actual occurrence of near misses can be expressed as:

$$Log\ L = \sum_{i=1}^{N} \sum_{j=1}^{J} Y_{ij}\ ln\ P_i(Y = j/x, \beta),$$

$$\text{where } P_i(Y = j/x, \beta) = \frac{\exp\ (v_{ij}(x_{ij,b}))}{\sum_{h=1}^{j} \exp\ ((v_{ih}(x_{ih,b}))} \quad (7.2)$$

where Y is the binary dependent variable, X represents the independent variables, v_{ij} is the utility function for the jth alternative of ith choice (computed in equation 7.1), N represents the occasion of choices, j represents the number of alternatives, P_i represents the predicted probability of the occasion of i occurrence of a near miss, β represents the parameter vector of the model.

7.5.3 Results

Table 7.4: The summary of the logistic regression model (Base model)

Dependent Variable:	Near_miss	R-squared	0.023
Method:	Least Squares	Adj. R-squared:	0.023
No. Observations:	46567	Chi-Square	1089.246
Df Residuals:	46552	Log-Likelihood:	-20987.042
BIC:	21148.271	AIC:	21017.042

Table 7.5: The results of the logistic regression model (Base model)

						95% Confidence Interval for Exp(B)	
Variables[a]	Coefficient (B)	Standard error	Wald	p-value	Exp(B)	Lower Bound	Upper Bound
Intercept	1.099	0.108	102.961	**0.000**			
person	0.075	0.005	202.635	**0.000**	1.078	1.067	1.090
bicycle	0.210	0.014	220.335	**0.000**	1.234	1.200	1.269
car	0.089	0.009	106.746	**0.000**	1.093	1.075	1.112
bus	-0.097	0.023	17.295	**0.000**	0.907	0.867	0.950
motorbike	0.355	0.047	57.194	**0.000**	1.426	1.301	1.563
truck	0.015	0.026	0.327	0.567	1.015	0.965	1.068
[time=Dawn/dusk-time]	-0.360	0.050	51.637	**0.000**	0.698	0.632	0.770
[time=Day-time]	-0.088	0.026	11.318	**0.001**	0.916	0.870	0.964
[time=Night-time]	0[b]						
[glare=0]	-0.080	0.028	8.030	**0.005**	0.923	0.874	0.976
[glare=1]	0[b]						
[weather=Clear weather]	-0.144	0.084	2.979	0.084	0.866	0.735	1.020
[weather=Rainy weather]	-0.301	0.085	12.633	**0.000**	0.740	0.627	0.874

[weather=Snowy weather]	0[b]						
[fog=0]	-0.278	0.061	20.562	**0.000**	0.757	0.672	0.854
[fog=1]	0[b]						
[cycle_lane=0]	-0.019	0.022	0.776	0.378	0.981	0.940	1.024
[cycle_lane=1]	0[b]						
[wet_surface=0]	0.024	0.043	0.313	0.576	1.024	0.942	1.113
[wet_surface=1]	0[b]						

[a]The reference category for the independent variable (near misses) is 1.
[b]This parameter is not included in the model.

Table 7.4 summarises the statistics of the Logistic regression models, highlighting the dependent variable, computation method of least square, the total number of observations used in the model, in addition to the R-squared which represents a low value of 0.023. This shows a limitation in the strength and fitness of the proposed model.

In general, the model shows statistically significant results for different independent variables in which the sign of the coefficient (B-value) and the statistical significance level of the P-value varies (See **Table 7.5**). All risk factors are statistically significant except for four variables: 1) Truck counts, 2) Clear weather, 3) the presence of a cycle lane and the condition of the road surface.

7.6 Balancing near miss and safe ride cases

7.6.1 Random sampling

The number of the frames of near misses is 13,145, whereas the number of safe case frames is 33,422. To provide a better representation for the dependant variable, a new experiment is conducted to tackle class imbalance in the dependent variable. This has been conducted by selecting a random sample of size 13,145 from the safe case frames (33,422 cases). A second logistic regression model is produced and the result is compared with the base model. (See **Fig. 7.5**).

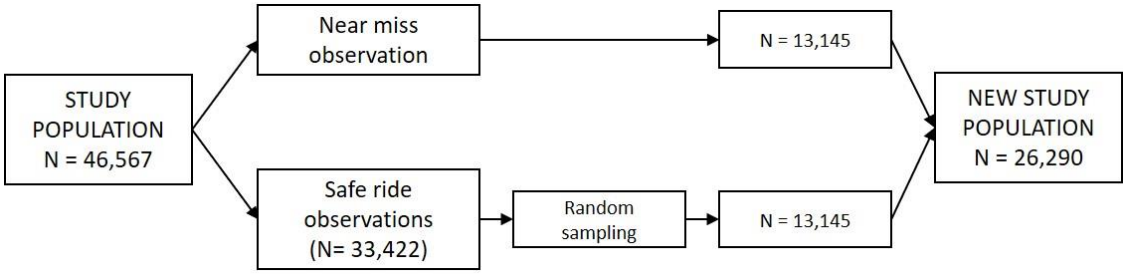

Figure 7.5: Random sampling technique for class imbalance in near_miss variable

7.6.2 Results

Table 7.6 summarises the statistics of the Logistic regression model that is used for the new experiment. The model shows a slight improvement in model fitness compared to the base model, with an R-squared of 0.027. The same independent variables are identified as statistically significant

as the base model, with slight changes in the variable coefficients and standard errors. Unlike the base model, this model shows a statistically insignificant intercept. **Table 7.7** shows the coefficients of all variables and their statistical significance with the occurrence of near misses in presence of a cycle lane.

Table 7.6: The summary of the logistic regression model (Balanced classes of near misses)

Dependent Variable:	Near_miss	R-squared	0.027
Method:	Least Squares	Adj. R-squared:	0.027
No. Observations:	26290	Chi-Square	725.606
Df Residuals:	26290	Log-Likelihood:	-15671.736
BIC:	15824.390	AIC:	15701.736

Table 7.7: The results of the logistic regression model Balanced classes of near misses)

						95% Confidence Interval for Exp(B)	
Variables[a]	Coefficient (B)	Standard error	Wald	p-value	Exp(B)	Lower Bound	Upper Bound
Intercept	0.164	0.129	1.613	0.204			
person	0.074	0.006	139.597	**0.000**	1.076	1.063	1.090
bicycle	0.201	0.016	153.051	**0.000**	1.223	1.184	1.262
car	0.089	0.010	73.561	**0.000**	1.093	1.071	1.115
bus	-0.100	0.029	11.928	**0.001**	0.905	0.855	0.958
motorbike	0.368	0.054	46.338	**0.000**	1.445	1.300	1.607
truck	0.034	0.031	1.196	0.274	1.035	0.973	1.100
[time=Dawn/dusk-time]	-0.417	0.063	43.899	**0.000**	0.659	0.583	0.746
[time=Day-time]	-0.120	0.032	14.394	**0.000**	0.887	0.834	0.944
[time=Night-time]	0[b]						
[glare=0]	-0.099	0.034	8.418	**0.004**	0.906	0.848	0.968
[glare=1]	0[b]						
[weather=Clear weather]	-0.117	0.098	1.433	0.231	0.889	0.734	1.078
[weather=Rainy weather]	-0.281	0.099	7.994	**0.005**	0.755	0.622	0.917
[weather=Snowy weather]	0[b]						
[fog=0]	-0.241	0.073	10.812	**0.001**	0.786	0.681	0.907
[fog=1]	0[b]						
[cycle_lane=0]	-0.044	0.026	2.790	0.095	0.957	0.909	1.008
[cycle_lane=1]	0[b]						
[wet_surface=0]	0.027	0.052	0.275	0.600	1.028	0.928	1.138
[wet_surface=1]	0[b]	0.129	1.613				

[a]The reference category for the independent variable (near misses) is 1.
[b]This parameter is not included in the model.

7.7 Granger causality of risk factors

7.7.1 Assumptions

Understanding the temporal causal structure of a given dataset is essential for interventions and decision-making for real-world applications (Bahadori and Liu, 2012). For a given risk factor to cause a near miss, it has to precede its occurrence. If the time lag between the risk factor being observed and the near-miss occurring can be modelled, then it has the potential to be used in an early warning system. **Fig. 7.6** shows the assumption of temporal causality, highlighting the scope that defines causality. To test for granger-causality, the figure shows that the tested variable must be in a sequential form and there is a defined lag between the selected variable and a near miss for causality to be significant.

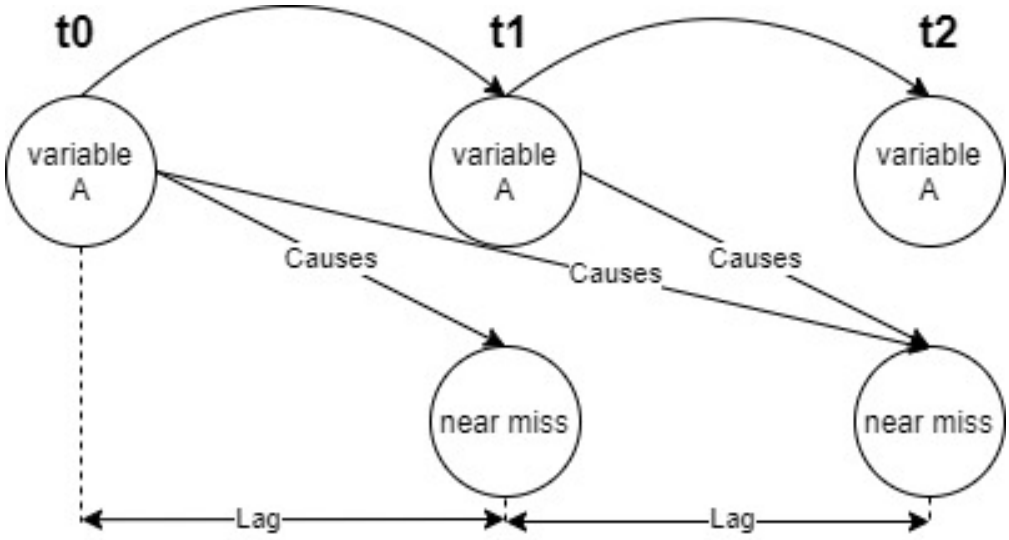

Figure 7.6: Temporal causality assumption

7.7.2 Methodology

To test the aforementioned assumption, the Granger causality method is employed (Bahadori and Liu, 2012; White and Lu, 2010). Granger causality is a statistical approach used to determine whether a given time series could be useful in predicting another one. The main hypothesis is that if a time-series X_1 Granger-causes a time-series X_2, then the past values of X_1 should contain information that assists in predicting X_2. To avoid the post hoc fallacy[1], Granger causality aims only to find predictive causality, whereas true causality is rather a philosophical argument.

Given that the variables are extracted from sequential frames, each variable can be seen as a time series and this approach can be useful to determine whether any risk factor Granger-causes near misses based on the time lag between the occurrence of a near miss, and the preceding existence of a given risk factor. To compute Granger causality, the variables have been transformed to stationary

[1] Given that an event (x) is *followed* by an event (y), event (x) must have caused event (y).

series, ensuring that the data distribution (mean, variance, and autocorrelation) of the variables do not change over time.

7.7.3 Results

To compute Granger causality for the different independent variables (16 variables), different experiments have been conducted for selecting a lag value. We experimented with values in the range of 1 to 120. This selection is made based on: 1) trial and error and; 2) the nature of the data set used, in which 30 data points represent one second.

The results show statistically significant outcomes for three independent variables (car, person, and glare), which means that the count of cars, persons or the occurrence of glare Granger-causes near misses for different lag intervals. In other words, these variables could be useful for forecasting the occurrence of cycling near misses.

Table 7.8 shows the result of the Granger causality for these three variables. Firstly, regarding the car variable, the results show significant Chi-squared and F-test values at a p-value less than 0.05 for a lag value that is 17 or lower (below 0.5 second). Besides the significant causality, this could also indicate the short-term effect or the rapid effect of the presence of a car in Granger-causing the occurrence of near misses. Secondly, regarding the person variable, similar to the car variable, the results show a statistically significant chi-squared test and f-test for various lags of p-value below 0.05. However, unlike the car variable, the causal effect of the person variable has a long-term effect in which the lag values range from 18 to 42 (approx. 1.5 seconds). Lastly, regarding the glare variable, similar to the two aforementioned variables, the occurrence of glare shows a statistically significant chi-squared test and f-test at different lags. Unlike these two aforementioned variables, however, the causal effect of glare on the occurrence of near misses remains significant both short-term (0.5 seconds) and long term (2 seconds). This could indicate how crucial the existence of glare is to the occurrence of cycling near misses.

Table 7.8: The significant results of the Granger causality

Variable	Lag	F-test	Chi-squared test	P-value
Car	5	2.5048	12.5269	0.0283
	7	2.4416	17.0964	0.0168
	8	2.2060	17.6547	0.0240
	9	1.9930	17.9443	0.0359
	14	1.8116	25.3778	0.0312
	15	1.6962	25.4597	0.0443
	16	1.7422	27.8947	0.0328

	17	1.6529	28.1212	0.0438
Person	18	1.6205	29.1928	0.0464
	25	2.1248	53.1480	0.0009
	42	1.4345	60.3207	0.0337
Glare	26	1.5611	41.0922	0.0342
	60	1.4308	85.9895	0.0160

7.8 Limitations

This chapter presented new approaches and outcomes for understanding the contributions of risk factors such as the counts of road users, visual, weather or surface conditions to the occurrence of cycling near misses. However, there are still limitations that need to be addressed in future work when it comes to assessing the cause and effect of the stated subject. First, data representation and distribution: Finding observation points that represent various types of events and conditions in the scope of the stated subject remains a critical issue for understanding and generalising the measured causes and effects. In this vein, for future studies, a naturalistic study needs to be carried out to include a representative sample of data that belongs to different types of near misses, and different visual, weather, and physical conditions. Second, addressing the behavioural aspects (as discussed in **Chapter III**) represents another limitation. Similar to addressing the issue of representative data in terms of scene types and conditions, the representation of strata that belongs to different socioeconomic structures needs to be considered. Last, even though the dataset presented in this chapter is large (N= 46,567), more data can be generated from more video streams. However, the process of generating this data via the deep models presented in the previous chapters (**Chapter V and VI**) remains computationally intensive. As for future work, more GPUs need to be considered.

7.9 Summary

In this chapter, we introduced different statistical approaches to understand the contributions of the risk factors to the occurrence of cycling near misses based on the observations generated by the deep models introduced in the previous methodological chapters. Firstly, the chapter analysed the data descriptively, highlighting the distribution of the selected 17 variables used in this chapter. Secondly, the chapter investigated the impacts of the different risk factors on the occurrence of near misses using a logistic regression model as a first step and base for further investigations of the underlying collinearly in the data set. Third, as a step further after introducing the base model, we conducted a new model that takes into consideration the class imbalance of the dependent variable. Lastly, the chapter addressed the temporal causality in the data set relying on Granger causality methods. After applying different experiments for the different independent variables at different lags, the results show statistically significant results for three variables (car, person, and glare) which granger-cause near misses and could be useful for forecasting near misses at different temporal lags.

8

DISCUSSION AND APPLICATIONS

8.1 Overview

This chapter represents the results of the overall framework and discusses them in the context of the current state-of-the-art. It also shows from a broader perspective how the introduced methods can be used in other applications in cities. Moreover, it discusses the findings thoroughly and shows the limitations and the potential future research work. The chapter also draws recommendations for practice and policy-making towards reaching AI-generated urban policies. Last, it discusses two vital applications (URBAN-i Box, and URBAN-i Cloud) to deploy and implement the research outcomes in the real world.

8.2 Framework Stringency Index (SI)

Even though each model presented in this research is validated individually, we introduced a Framework Stringency Index (SI) in chapter 4 to further evaluate the performance of the individual models for the given task of analysing cycling near misses. What makes this index unique is that it does not only include the performance of each model but it also takes into account its importance in understanding cycling near misses in terms of the weighting of the variables it generates the regression models introduced in chapter VII.

Table 8.1: The summary of the results of the introduced models

Deep models	Risk factors	Average precision[1]	Absolute statistical weight[2]	Normalised weight[3]
Model1- NightNet	nightime	0.885	0.268	0.101
	daytime	0.885	0.120	0.045
	dawn_dusk	0.885	0.417	0.157
Model2- GlareNet	Glare	0.883	0.099	0.037
Model3- PrecipitationNet(b)	clear	0.959	0.117	0.044
	rain	0.959	0.281	0.106
	snow	0.959	0.199	0.075
Model4-FogNet	Fog	0.862	0.241	0.091
Model5-SlipNet	Wet surface	0.918	0.241	0.091
Model6- Object_detection	person	0.75	0.074	0.028
	bicycle	0.79	0.201	0.076
	car	0.81	0.089	0.034
	bus	0.77	0.100	0.038
	motorbike	0.81	0.368	0.139
	truck	0.77	0.034	0.013

[1]The average precision calculated for each model on the test sets introduced in chapter V.
[2]The absolute value of the statistical weight of the second logistic regression model computed based on the coefficient (B) statistics, introduced in chapter VII.
[3]The normalised version of the statistical weight introduced in the previous column.

Table **8.1** shows the combined results of the individual models accumulated from the previous chapters. These results represent the average precision of each model and their absolute and normalised statistical weights. After computing SI based on the presented models' results, the overall performance of the pipeline achieved a SI of 0.81. The closer the SI value to 1, the more accurate the framework is in detecting the different risk factors in accordance with the different precisions of the deep models and the weight of a given factor on the occurrence of near misses. Based on the results of the normalised weights, it is worth mentioning that the SI index is highly influenced by the precisions of scenes that belong to clear and rainy weather and those that include people and bikes. Nevertheless, it is less influenced by the precision of scenes that include trucks, glare, and wet surfaces.

8.3 What makes the overall framework (URBAN-i) the state-of-the-art for understanding urban dynamics?

There is no doubt that advances in computer science in general, or geo-computational methods have led to several advances in geography and the understanding of urban systems (Arribas-Bel and Reades, 2018). While it is vital to understand the overall urban systems of cities from satellite images, seeing cities from the street view adds more dimensions of information and complexity. Capturing these

rapid urban changes in day-to-day life through images offers more opportunities to tackle urban dynamics towards a better understanding of cities. In the previous chapters, we introduced different methods to understand critical events such as cycling near misses and their risk factors, which we refer to here, as URBAN-i. The URBAN-i can be utilised as a base for generating urban data for multi-purpose urban and transport-related studies. The framework can capture information related to environmental, visual, and built environment conditions coupled with the spatiotemporal context.

The innovation can be seen in detecting critical events and understanding their causes. Also, by applying the same algorithms to active cameras in cities, the model can enable real-time capturing of data. Last, the same methodology can be applied to tackle and classify different urban issues from urban scenes. Coupled with remote sensing image classification methods, the proposed URBAN-i framework can reveal deeper insights into the dynamics of cities.

Putting all the algorithms of the URBAN-i model together, **Fig. 8.1** shows different examples of sequential frames of cycling in London with different scene conditions, including the presence of a cycling lane, or the occurrence of near misses. The figure also shows how information can be extracted from urban scene images to a database that can be used for various urban research and data visualisation purposes. This database, in addition to the aforementioned factors, shows the planning status of a scene, in addition to the depth of road users. This will enable urban modellers and researchers to collect and analyse their data sets based on the needs of their research.

8.4 Ancillary methods and input sensors for sensing the environment

By following the methods introduced in the previous chapter, and in addition to the deep models introduced, several additional models can be computed to assist and complement the perception of the surroundings of a given scene. For example, a model can be trained to classify scenes based on the image location: (i.e. indoor, outdoor, or in the transportation mode scenes). Also, a model can be trained to classify the scene by the pose in which the image or the video streams are captured (i.e. street level, or aerial view). And last, a model can be trained to classify scenes regarding the traffic conditions (i.e. heavy, medium or low, or no traffic).

In general, the field of machine learning is constantly evolving and methods develop rapidly, which necessitates future adaptation and adoption of new techniques to achieve better performance on a given task. It is crucial, therefore, to develop an overall methodology of a pipeline of deep models that allows future adaptation to either new methods or refined existing methods with minimal resources. Accordingly, the overall methodology is developed to adapt to new models that could refine the introduced ones, or introduce new input sensors for different types of data. For instance, adding a LiDAR unit could augment the video streams, after data fusion, with the estimated depth.

8.5 Covid-19 pandemic and the increase in the number of people on bikes

Whether temporarily or permanently, it has been debated that there is an increase in the number of people on bikes with different profiles and social characteristics (DfT, 2020). There is no doubt that this increase in cycling would have direct benefits for health and the environment. With this increase, however, cycling infrastructure needs further preparation to host the increased numbers and more safety-related measures need to be considered. Accordingly, automating the detection of near misses could lead to the design of new safety policies for cycling in cities, to understand the capacity of the

current cycling infrastructure for safe use, and to understand the tipping point for the increase in the number of near misses based on the interaction with other people on bikes.

8.6 Moving from prediction to decision-making to policy

After addressing the limitations of the stated models, the superior performance of modern computer vision algorithms is in little doubt. However, the extent to which model outputs can be used for automated and optimised policy and decision making remains an important research frontier. Big data, of which image data is a subset, is increasingly having an impact on decision and policymaking, whether explicitly or not. Government authorities rely on algorithmic outputs to inform their decisions daily. The practical, ethical and societal implications of this are still unclear and (Duarte and Álvarez, 2019) note the lack of synchronicity between the potential societal impact of AI technologies and our cultural discussions around them.

Alongside other sources of big data, images and video play a particularly important role in this effort because they capture the action and interaction of humans within their environment. This provides the opportunity to understand a range of issues, such as how the structure of the built environment affects pedestrian safety, or how street lighting influences crime. These issues are inextricably linked, and urban planning and policymaking must take a holistic view of them to avoid disadvantaging certain groups.

8.6.1 Enabling Technologies

Two enabling technologies will be important in this area. Firstly, multi-agent reinforcement learning will enable more realistic human agents to be simulated in more realistic urban environments. The behaviour of these agents can be learned and validated using images and videos data. Such models could support or supersede traditional land use and transport planning approaches, as well as optimise the performance of urban systems such as transportation. So far, this research shows evidence on how to extract agents and environmental conditions from street-level images. This gives the first milestones for building a virtual environment -based on realistic settings- to simulate the behaviours of urban multi-agents.

The second technology is GANs. It is not inconceivable that GANs, fed with images of a city, whether at a street view or aerial, could eventually be trained to design effective urban environments. In the same way that GANs can generate synthetic human faces that are indistinguishable from real faces (Karras et al., 2019), they could be used to plan new cities or neighbourhoods that perform like existing cities. This is certainly a long way off, but advancements in AI will enable predictions that are beyond what humans or social groups may achieve, or even conceive of (Duarte and Álvarez, 2019). The outcomes of the introduced framework would allow generative models to synthesise data conditioned based on multiple predictions of the introduced framework. For example, generating an image with clear weather when rainy and foggy weather is detected to enhance visibility.

8.6.2 Cameras with embedded AI in cities for real-time insights

The implementation of computer vision model pipelines in (near) real-time is a crucial issue for urban analytics and the Internet of Things (IoT) systems. This deployment at the edge in urban contexts can show a direct impact on the current research for developing urban theories and policies. For example, cameras with embedded AI may alert police or transport control rooms of incidents,

which they can verify and respond to. This type of system should be managed in a coordinated fashion so that the needs of various authorities can be met, which requires the integration of the different layers of the city. However, while this approach will enable fast decision making and response, it falls short of being a fully intelligent and automated system able to implement or generate policy.

8.6.3 Policy for AI and by AI

After the deployment of AI in cities based on accepted norms and ethics, their deployment in cities will also lead to the generation of adaptive urban policies by AI. AI has the potential to generate dynamic and place-based policies. However, challenges remain in the innovation and fusion of different domains of knowledge to reach this critical step where the machine not only predicts and makes decisions but generates short- and long-term plans. Most importantly, it is a mixture of the tackled deep learning and computer vision research in urban settings with Natural Language Processing (NLP) research and reinforcement learning. By merging these different knowledge domains and integrating models that are capable of addressing multiple tasks in cities, theories and more flexible place-oriented policies can be generated for cities and knowledge can be transferred from one city to another.

8.6.4 Conceptual framework towards AI-generated policy and decision making

Fig. 8.2 shows a conceptual framework and a recommended process for achieving the two crucial steps outlined in sections 6.2 and 6.3, and how they can be reached from the current perspective of deep computer vision research. It shows the overall system for policy-makers and developers showing the important aspect of this process and the domains that are still under-developed and require further integration with urban analytics research. This thesis has addressed the top part of the diagram, resulting in cameras with embedded AI that will be described in **section 8.8**. The bottom part of the diagram will be the focus of future work.

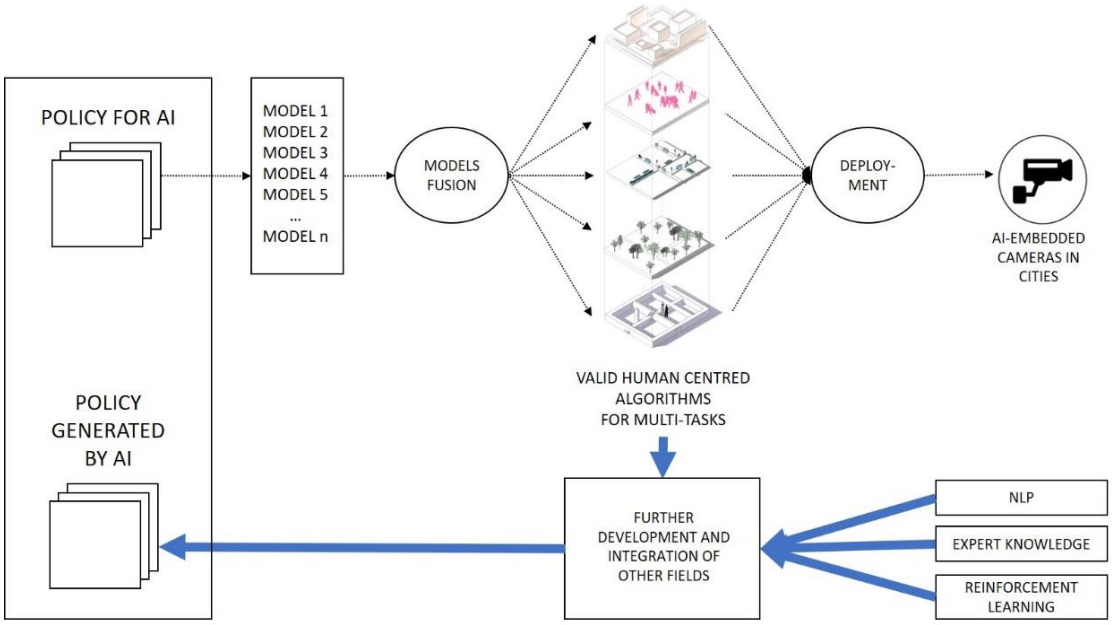

Figure 8.7: AI-generated urban policy conceptual framework

8.7 Ethics of AI

Before the framework in **Fig. 8.2** can be fully realised and trusted, it will be important to address the ethics of AI and ensure that our reliance on AI algorithms for decision making doesn't have unintended consequences. Understanding the comprehensive impact of AI on our daily lives remains a crucial subject of ongoing debate (Hagendorff, 2020; McLaren, 2003). It is still unanswered how we can provide an objective assessment to ensure the safety and fairness of AI when it comes to operation in cities while protecting the privacy and security of individuals. Subjectively, algorithms, in general, are not biased, however, when they are built by humans, different types of biases can be inherited whether intentionally due to misconduct, or unintentionally due to systematic errors and data misrepresentation. There are no doubts that the former is an issue that needs to be avoided by following the best practice and norms for creating a ground truth and developing algorithms that ensure fairness to those who are affected by it. For example, this research is funded by the Road Safety Trust and has been carried out from the perspective of improving cycling safety. Despite all measures being taken to ensure fairness in the research design, the thesis still examines risk factors for incidents from the perspective of a single group of road users. Furthermore, as stated in **section 6.9**, the data used are from a self-selected group of contributors to video sharing sites. If action is taken to reduce the presence of risk factors for people on bikes using this data, it may inadvertently increase risk or have other negative consequences for other groups of road users. Therefore, it is vitally important to continue to work towards systems that take a holistic view of the city and its interconnected components in order to prevent these unintended consequences.

In terms of algorithmic bias, this is a crucial issue that requires thorough analysis and assessment to be pinpointed and resolved due to the complexity of AI algorithms. Accordingly, a wide range of research has tackled biases in AI by focusing on understanding the rationale of machine behaviour, instead of how models are created by developers (Rahwan et al., 2019). On the other hand, noticeable progress has been made in domains such as explainable AI (Holzinger et al., 2018; Lundberg et al., 2020; Yang et al., 2021), where models are unpacked to assess their performance and highlight the weights of the individual hyperparameters of a given model when a prediction decision is made.

8.8 Research applications

This section introduces the implementation of URBAN-i for practical use in cities. The URBAN-i system is currently being used within the Road Safety Trust funded 100 Cyclists Project. URBAN-i is a suite of bespoke computer vision algorithms, based on this thesis, for detecting critical events and understanding their causes. As it watches, it learns how city systems interact to produce risk, proving actionable insights. The suite of URBAN-i comprises a camera with embedded AI (URBAN-i Box[2]) and a cloud-based system (URBAN-i Cloud) which both share the vision system. However, the URBAN-i Box comprises four other sub-systems (voice, environmental, communication, and data storage and encryption systems) that simultaneously function and interact with one another.

8.8.1 Application 1: URBAN-i Box

Generally, when critical events occur in cities, rapid detection and response are of utmost importance. However, critical events often result from the interaction between different systems that can be interdependent and complex. As has been established in this thesis, the occurrence of a traffic

[2] URBAN-i Box is a tangible prototype that has been tested and used in real-world settings.

incident could be due to a multitude of factors such as the weather, built or natural environment, or road users interactions. Current sensors are task-based (i.e. CCTV, traffic cameras, pollution sensors, help points) and output data that need processing and integration by experts before they can provide information for decision-making.

URBAN-i is operationalised through a camera with embedded AI that generates synchronized data. When a person is riding a bike and has a near miss, URBAN-i Box sees the whole scene, from how he/she was cycling to what the weather was like and the conditions of the road surface to work out the most likely contributing factors. The core elements of URBAN-i Box are the computer vision algorithms that are presented in this research and the sensor design.

The vision subsystem is the primary system and all other sub-systems either assist, complement, or verify its outcomes. It is divided into six phases, each one tackling different vision tasks: 1) Sensing location types and poses, 2) Sensing and detecting the conditions of the environment (including weather, visual, green spaces, deterioration, etc.), 3) detecting and tracking objects, 4) detecting instant actions (accidents, near misses, stabbing, etc.), 5) depth estimation and transformation into a top-view, and last, 6) causal inference. These phases are integrated and built as an end-to-end pipeline with a single input of video streams. The framework has four outputs: 1) critical event detection (in this case, near misses), 2) a list of detected risk factors and objects, 4) objects and their depths on a top-view image, and last 4) causal inference for the detected factors when critical events are detected.

The voice and environmental sub-systems are based on different input sensors (mic, air quality, pressure, temperature sensors) that are processed by Artificial Neural Networks to classify their inputs and if values are higher than designated thresholds warnings can be sent after an adjoined assessment of the vision system. The communication sub-system handles the user's authentication, inputs and interfacing between the sensor and the user. Finally, the storage and encryption sub-system handles how data is stored, accessed, and secured.

To achieve the sub-systems, the URBAN-i comprises 10 main hardware components, which are:

- An 8 MB camera with a wide-view and high-resolution (1296 X 972) calibrated sensor for video streaming while capturing single images with time and location reference. The camera algorithms rely on different AI models to detect different objects, classify scenes, and recognise instant actions.
- A Light Detection And Ranging (LiDAR) sensor that uses the direct flight time to measure the reflected light to a distance of up 12 metres for both indoor and outdoor. The sensor is calibrated to detect safe and unsafe distances for transport modes.
- An environmental sensor that integrates data related to temperature, humidity, pressure and gas sensing for air quality. The sensor is calibrated to automatically determine the concentration of particles of Carbon monoxide, Nitrogen dioxide, and Alcohol in indoor and outdoor scenes.
- A low powered screen to provide vital information to users such as Disk capacity, battery level, multi-sensor readings, and button interaction.
- A wide range of colour lights to be utilised for user interface and notification for the different tasks and scenarios.

- A GPS sensor with up to 10 updates per second. It consists of 66 channels. RTC battery, and built-in antenna that can capture data from multi satellites.
- An ultra-high-range luminosity sensor that relies on both infrared and full spectrum diodes to measure light temperature and intensity. It also detects different hand gestures that allow users to interface with the URBAN-i box.
- An inertia measurement unit (IMU) sensor of 9 axes is used to provide precise data regarding acceleration, heading and gyroscope. The sensor algorithms comprise AI models to understand different actions in cities to add further features to the camera functions.
- A Li-po chargeable battery of 3800 mAh capacity is used to provide a long time for streaming up to 4 and half hours in a single charge.
- A custom-designed holder that allows a flexible and easy mounting of the box in different types of surface and transport modes.

Fig. 8.3 shows the design of the URBAN-i Box, highlighting the organisations of the aforementioned components.

Figure 8.8: The model and specifications of the URBAN-i Box

8.8.2 Application 2: URBAN-i Cloud

Another application of the introduced methods is a prototype cloud-based system for computation on user-defined data sets. Unlike the URBAN-i Box, the cloud-based system only comprises the computer vision system for detecting critical events in addition to understanding their causes. However, the system can be developed further to include other tools and methods. **Fig. 8.4** shows the main user interface of the URBAN-i Cloud. On uploading a video stream or an image, the model can also compute on the cloud and post the prediction results to the users.

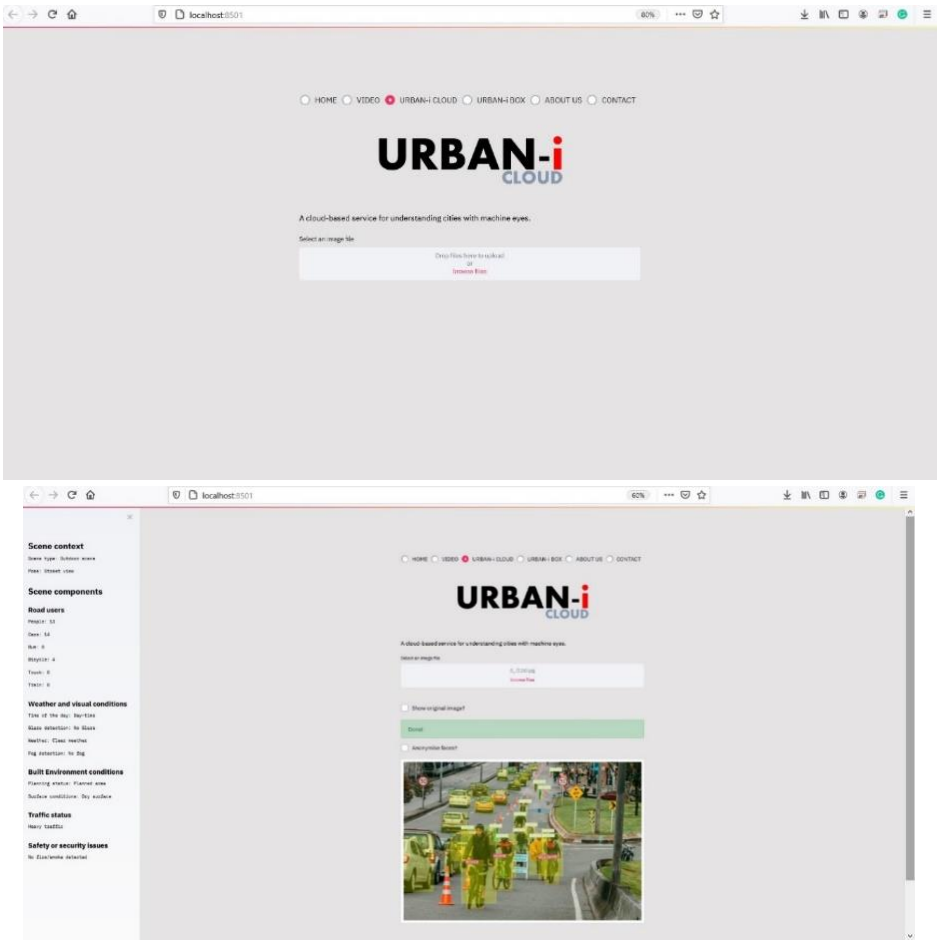

Figure 8.9: The user interface of URBAN-i Cloud

8.9 Limitations and future work

While the introduced framework shows novelty in analysing a wide range of the conditions of a given scene that belong to near misses or safe rides, the model limitation appears in analysing scene conditions that are mixed between several conditions in the same scene (i.e. a mixture of a wet and dry surface). In future work, a potential way to develop the model further is by using semantic segmentation and scene parsing. This pixel-level segmentation would allow the model to provide multiple categorizations and localisation of the surface conditions for a single image. Accordingly, this

will enhance the accuracy of the model when detecting complex scenes in the real world. However, it would increase the inference time needed to compute the entire framework.

Second, besides the accuracy of the utilised model for object detection, the counts of objects within a given scene are still limited by the field of view of the camera used to generate images. This may produce an under-representation of the actual road-users in a given region of interest which falls outside the camera's field of view. For instance, for a common camera with a field of view of 65 degrees, objects that are in the same line as the cyclist may not be seen in a given frame, despite their appearance in future image frames. This limitation, however, can be avoided in future research by introducing a 360-degree camera, or at least a 180-degree camera that covers the entire frontal view while cycling. In doing so, a better representation of road users can be extracted.

On the other hand, the framework precision in detecting and classifying urban scenes depends on several factors. First, the individual accuracy of each pre-trained CNN model is a key factor. Each one can be modified to achieve better accuracy and results with larger training datasets, higher computational power, and deeper networks. However, the goal of this research is to show evidence that the complexity of urban issues such as detecting near misses and their risk factors can be tackled by deep learning and computer vision with less burden on the researcher and using data that are available and accessible by everyone anywhere in the globe, without the means of expensive sensors.

8.10 Summary

In this chapter, we discussed the overall framework or URBAN-i as a new computer vision tool that can be utilised for various purposes of modelling the dynamics of urban areas, which has been used in this research to detect and analyse cycling near misses and their causes. We have shown the possibility of extracting information from cities and tackling critical events by using urban images. Accordingly, this tool exemplifies the application of AI and deep learning in understanding city dynamics and development. The outcomes of the thesis have been operationalised through two applications; the URBAN-i Box, and the URBAN-i Cloud, which directly contributes to the advancements in the methods of urban modelling to better understand cities. In the future, we expect devices with embedded AI such as the URBAN-i box to form part of an ecosystem of AI-generated policy, but this requires further research and development, both in terms of the AI algorithms and the ethics and regulation around their use.

9

SUMMARY AND CONCLUSION

It is evident that cycling is a growing mode of transportation in many cities. Whether for commuting or leisure, its advantages regarding public health and the reduction of environmental pollution have influenced planners and policy-makers to invest in its infrastructure in cities. However, the modal share of cycling among other transport modes remains low in many cities. Cycling is still perceived as a dangerous transport mode, not only because of the occurrence of incidents but also due to the frequent exposure to events that may not necessarily end up by a collision or so-called near misses. In this research, it is defined as a situation in which a person on a bike was required to act to avoid a crash, such as braking, speeding, swerving or stopping. This fear of getting hit or falling while cycling limits the modal share of cycling among other transport modes.

In this research, we aimed to identify when, where, and why cycling near-misses take place in cities to provide a safer environment for people on bikes. Due to the interdisciplinary nature of the addressed topic, the research includes an investigation on the different factors related to the conditions of the built and natural environments and road user interactions that may cause potential near mises. Accordingly, this research introduced novel methods, not only to detect and analyse near misses in complex urban scenes but also to analyse cities and advance the methods used for urban modelling. In a broad sense, the research aimed to map some of the agents of cities (pedestrian, transport modes, and environmental conditions) in order to understand their interactions at a given time and space with the respect to the complexity of the urban settings. The novelty of the proposed framework, or URBAN-i, lies in the application of computer vision and deep learning to understanding the conditions of the built environment and interaction among the different road-users in cities that leads to critical events such as near misses.

In practice, when critical events occur in cities, rapid detection and response are of utmost importance. However, critical events often result from the interaction between different systems that are often interdependent and complex. When a cycling near miss occurs, it could be because of the weather, built or natural environment, or road users interactions. Only by taking a holistic view can we develop a system to detect and understand these events and their causes. Current sensors are

task-based (i.e. CCTV, traffic cameras, pollution sensors, help points) that output data that need processing and integration by experts before they can provide information for decision-making.

The introduced overall framework (URBAN-i) is a suite of computer vision algorithms that can be utilised for detecting critical events and understanding their causes. As it watches, it learns how city systems interact to produce risk, proving actionable insights. Accordingly, when a person is riding a bike and has a near miss, URBAN-i sees the whole scene, from how he/she was cycling to what the weather was like and the conditions of the road surface to work out the most likely contributing factors.

The research aimed to answer one crucial question of *how can we tackle different scenarios of the interaction between people and transportation, bearing in mind the conditions and the dynamics of the built environment?* However, to cover the different dimensions of this question, we have subdivided it into sub-questions, these are:

1. How can we identify and predict urban systems that may influence near misses in cities?
2. To what extent can machine vision be used to understand the nuances of physical and non-physical elements from images/ videos?
3. To what extent can machine vision detect environmental conditions and visibility related factors from urban scenes?
4. To what extent can the machine recognise a safe or a near miss scene from the overall interactions of road users in complex scenes?
5. When and where do cycling near-misses take place in cities?
6. Which factors are more likely to cause cycling near misses?

Different types of computer vision methods are introduced to address the multi-faced nature of the aforementioned questions. In general, the logic behind selecting a given method is based on two main reasons; 1) to address the individual sub-question, and 2) to contribute to the bigger picture of the research topic. The research methodology consisted of five sections that respond to the research objectives and goals. The research provided several novel methods relying on deep learning and computer vision such as URBAN-i, WeatherNet, and CyclingNet. The URBAN-i model is a multipurpose model that can be used for various tasks related to urban modelling. After training several deep models to understand the nuances of various urban scenes we obtained a validation accuracy to prove that even the qualitative urban conditions in cities can be classified and detected relying on computer vision. The current version of the model can be used for mapping the occurrence of transport modes, pedestrian and planning status of urban scenes in cities, in addition to detecting and analysing the causes of cycling near misses. Nevertheless, the model can be developed further to be used for modelling the dynamics of traffic congestion, crowd, or crime detection. Therefore, better decisions can be taken by policy-makers and planners to optimise resources and improve the living conditions in the urban world. The introduced framework is fully coded in Python programming to be used as a tool to capture and understand urban dynamics in different corners of the globe.

After addressing the limitations of the stated models, the superior performance of modern computer vision algorithms is in little doubt. However, the extent to which model outputs can be used for automated and optimised policy and decision making remains an important research frontier. Big data, of which image data is a subset, is increasingly having an impact on decision and policymaking, whether explicitly or not. Alongside other sources of big data, images and video streams play a particularly important role in this effort because they capture the action and interaction of humans within their environment. This provides the opportunity to understand a range of issues, such as how the structure of the built environment affects pedestrian safety, or how street lighting influences crime. These issues are inextricably linked, and urban planning and policymaking must take a holistic view of them to avoid disadvantaging certain groups.

The implementation of computer vision model pipelines in (near) real-time is a crucial issue for urban analytics and the Internet of Things (IoT) systems. This deployment at the edge in urban contexts can show a direct impact of the current research for developing urban theories and policies. For example, cameras with embedded AI may alert police or transport control rooms of incidents, which they can verify and respond to. This type of system should be managed in a coordinated fashion so that the needs of various authorities can be met, which requires the integration of the different layers of the city. Accordingly, the introduced methods can be implemented and operationalised in cameras. In this research, we showed how URBAN-i can be implemented through a camera with embedded AI, the URBAN-i Box, that generates synchronized data or implemented in a cloud-based service, or so-called URBAN-i Cloud.

In a nutshell, this study contributes to science with theoretical and empirical foundations. Put all together, the research aims to provide human-centred evidence that may enable policy-makers and planners to provide a safer built-up environment for cycling in London, or elsewhere.

REFERENCES

Abay, K.A., 2015. Investigating the nature and impact of reporting bias in road crash data. Transp. Res. Part Policy Pract. 71, 31–45. https://doi.org/10.1016/j.tra.2014.11.002

Aldred, R., 2018. Inequalities in self-report road injury risk in Britain: A new analysis of National Travel Survey data, focusing on pedestrian injuries. J. Transp. Health 9, 96–104. https://doi.org/10.1016/j.jth.2018.03.006

Aldred, R., 2016. Cycling near misses: Their frequency, impact, and prevention. Transp. Res. Part Policy Pract. 90, 69–83. https://doi.org/10.1016/j.tra.2016.04.016

Aldred, R., Crosweller, S., 2015. Investigating the rates and impacts of near misses and related incidents among UK cyclists. J. Transp. Health 2, 379–393. https://doi.org/10.1016/j.jth.2015.05.006

Aldred, R., Goodman, A., 2018. Predictors of the frequency and subjective experience of cycling near misses: Findings from the first two years of the UK Near Miss Project. Accid. Anal. Prev. 110, 161–170. https://doi.org/10.1016/j.aap.2017.09.015

Alvarez, L., Deriche, R., Papadopoulo, T., Sánchez, J., 2007. Symmetrical Dense Optical Flow Estimation with Occlusions Detection. Int. J. Comput. Vis. 75, 371–385. https://doi.org/10.1007/s11263-007-0041-4

Amirkolaee, H.A., Arefi, H., 2019. Height estimation from single aerial images using a deep convolutional encoder-decoder network. ISPRS J. Photogramm. Remote Sens. 149, 50–66. https://doi.org/10.1016/j.isprsjprs.2019.01.013

Andrade, E.L., Blunsden, S., Fisher, R.B., 2006. Modelling Crowd Scenes for Event Detection, in: 18th International Conference on Pattern Recognition (ICPR'06). Presented at the 18th International Conference on Pattern Recognition (ICPR'06), IEEE, Hong Kong, China, pp. 175–178. https://doi.org/10.1109/ICPR.2006.806

Arribas-Bel, D., 2014. Accidental, open and everywhere: Emerging data sources for the understanding of cities. Appl. Geogr. 49, 45–53. https://doi.org/10.1016/j.apgeog.2013.09.012

Arribas-Bel, D., Reades, J., 2018. Geography and computers: Past, present, and future. Geogr. Compass 12, e12403. https://doi.org/10.1111/gec3.12403

Audebert, N., Le Saux, B., Lefèvre, S., 2018. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. ISPRS J. Photogramm. Remote Sens. 140, 20–32. https://doi.org/10.1016/j.isprsjprs.2017.11.011

Ayvaci, A., Raptis, M., Soatto, S., 2012. Sparse Occlusion Detection with Optical Flow. Int. J. Comput. Vis. 97, 322–338. https://doi.org/10.1007/s11263-011-0490-7

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39, 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615

Bahadori, M.T., Liu, Y., 2012. Granger Causality Analysis in Irregular Time Series, in: Proceedings of the 2012 SIAM International Conference on Data Mining. Presented at the Proceedings of the 2012 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, pp. 660–671. https://doi.org/10.1137/1.9781611972825.57

Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R., 2011. A Database and Evaluation Methodology for Optical Flow. Int. J. Comput. Vis. 92, 1–31. https://doi.org/10.1007/s11263-010-0390-2

Batty, M., 2009. Cities as complex systems: scaling, interaction, networks, dynamics and urban morphologies, in: Encyclopedia of Complexity and Systems Science. Springer, pp. 1041–1071.

Batty, M., 2008. The Size, Scale, and Shape of Cities. Science 319, 769–771. https://doi.org/10.1126/science.1151419

Batty, M., 1997. The computable city. Int. Plan. Stud. 2, 155–173. https://doi.org/10.1080/13563479708721676

Batty, M., 1976. Urban modelling: algorithms calibrations, predictions, Cambridge urban and architectural studies. Cambridge University Press, Cambridge ; New York.

Batty, M., Couclelis, H., Eichen, M., 1997. Urban systems as cellular automata. SAGE Publications Sage UK: London, England.

Batty, M., Torrens, P.M., 2005. Modelling and prediction in a complex world. Futures 37, 745–766. https://doi.org/10.1016/j.futures.2004.11.003

Batty, M., Torrens, P.M., 2001. Modelling complexity : The limits to prediction. Cybergeo. https://doi.org/10.4000/cybergeo.1035

Batty, M., Xie, Y., Sun, Z., 1999. Modeling urban dynamics through GIS-based cellular automata. Comput. Environ. Urban Syst. 23, 205–233.

Becattini, F., Uricchio, T., Seidenari, L., Del Bimbo, A., Ballan, L., 2021. Am I Done? Predicting Action Progress in Videos. ACM Trans. Multimed. Comput., Communications, and Applications 16, 1–24.

Beck, B., Chong, D., Olivier, J., Perkins, M., Tsay, A., Rushford, A., Li, L., Cameron, P., Fry, R., Johnson, M., 2019. How much space do drivers provide when passing cyclists? Understanding the impact of motor vehicle and infrastructure characteristics on passing distance. Accid. Anal. Prev. 128, 253–260. https://doi.org/10.1016/j.aap.2019.03.007

Beck, B., Stevenson, M., Newstead, S., Cameron, P., Judson, R., Edwards, E.R., Bucknill, A., Johnson, M., Gabbe, B., 2016. Bicycling crash characteristics: An in-depth crash investigation study. Accid. Anal. Prev. 96, 219–227. https://doi.org/10.1016/j.aap.2016.08.012

Ben-Akiva, M., McFadden, D., Abe, M., Böckenholt, U., Bolduc, D., Gopinath, D., Morikawa, T., Ramaswamy, V., Rao, V., Revelt, D., 1997. Modeling methods for discrete choice analysis. Mark. Lett. 8, 273–286.

Bettencourt, L., 2013. The origins of scaling in cities. science 340, 1438–1441.

Bettencourt, L., West, G., 2010. A unified theory of urban living. Nature 467, 912–913.

Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B., 2016. Simple Online and Realtime Tracking. 2016 IEEE Int. Conf. Image Process. ICIP 3464–3468. https://doi.org/10.1109/ICIP.2016.7533003

Bilen, H., Fernando, B., Gavves, E., Vedaldi, A., Gould, S., 2016. Dynamic Image Networks for Action Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, pp. 3034–3042. https://doi.org/10.1109/CVPR.2016.331

Blaizot, S., Papon, F., Haddak, M.M., Amoros, E., 2013. Injury incidence rates of cyclists compared to pedestrians, car occupants and powered two-wheeler riders, using a medical registry and mobility data, Rhône County, France. Accid. Anal. Prev. 58, 35–45. https://doi.org/10.1016/j.aap.2013.04.018

Bottino, A., Garbo, A., Loiacono, C., Quer, S., 2016. Street Viewer: An Autonomous Vision Based Traffic Tracking System. Sensors 16, 813. https://doi.org/10.3390/s16060813

Branion-Calles, M., Nelson, T., Winters, M., 2017. Comparing Crowdsourced Near-Miss and Collision Cycling Data and Official Bike Safety Reporting. Transp. Res. Rec. J. Transp. Res. Board 2662, 1–11. https://doi.org/10.3141/2662-01

Bretagnolle, A., Daudé, E., Pumain, D., 2006. From theory to modelling: urban systems as complex systems. CyberGeo Eur. J. Geogr.

Broach, J., Dill, J., Gliebe, J., 2012. Where do cyclists ride? A route choice model developed with revealed preference GPS data. Transp. Res. Part Policy Pract. 46, 1730–1740. https://doi.org/10.1016/j.tra.2012.07.005

Buch, N., Velastin, S.A., Orwell, J., 2011. A Review of Computer Vision Techniques for the Analysis of Urban Traffic. IEEE Trans. Intell. Transp. Syst. 12, 920–939. https://doi.org/10.1109/TITS.2011.2119372

Buch, S., Escorcia, V., Shen, C., Ghanem, B., Niebles, J.C., 2017. SST: Single-Stream Temporal Action Proposals, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, pp. 6373–6382. https://doi.org/10.1109/CVPR.2017.675

Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J., 2012. A Naturalistic Open Source Movie for Optical Flow Evaluation, in: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (Eds.), Computer Vision – ECCV 2012. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 611–625. https://doi.org/10.1007/978-3-642-33783-3_44

Cai, B.Y., Li, X., Seiferling, I., Ratti, C., 2018a. Treepedia 2.0: Applying Deep Learning for Large-scale Quantification of Urban Tree Cover. ArXiv180804754 Cs.

Cai, B.Y., Li, X., Seiferling, I., Ratti, C., 2018b. Treepedia 2.0: Applying Deep Learning for Large-scale Quantification of Urban Tree Cover, in: ArXiv:1808.04754 [Cs]. Presented at the IEEE International Congress on Big Data (BigData Congress), pp. 49–56.

Cao, Y., Wu, Z., Shen, C., 2018. Estimating Depth From Monocular Images as Classification Using Deep Fully Convolutional Residual Networks. IEEE Trans. Circuits Syst. Video Technol. 28, 3174–3182. https://doi.org/10.1109/TCSVT.2017.2740321

Cao, Y., Wu, Z., Shen, C., 2017. Estimating Depth from Monocular Images as Classification Using Deep Fully Convolutional Residual Networks. ArXiv160502305 Cs.

Cao, Z., Simon, T., Wei, S.-E., Sheikh, Y., 2017. Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, pp. 1302–1310. https://doi.org/10.1109/CVPR.2017.143

Caron, M., Bojanowski, P., Joulin, A., Douze, M., 2018. Deep Clustering for Unsupervised Learning of Visual Features 29.

Cha, Y.-J., Choi, W., Büyüköztürk, O., 2017. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks: Deep learning-based crack damage detection using CNNs. Comput.-Aided Civ. Infrastruct. Eng. 32, 361–378. https://doi.org/10.1111/mice.12263

Chao, Y.-W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R., 2018. Rethinking the Faster R-CNN Architecture for Temporal Action Localization. ArXiv180407667 Cs.

Chaurand, N., Delhomme, P., 2013. Cyclists and drivers in road interactions: A comparison of perceived crash risk. Accid. Anal. Prev. 9.

Chaurasia, A., Culurciello, E., 2017. LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation. 2017 IEEE Vis. Commun. Image Process. VCIP 1–4. https://doi.org/10.1109/VCIP.2017.8305148

Chen, J., Dowman, I., Li, S., Li, Z., Madden, M., Mills, J., Paparoditis, N., Rottensteiner, F., Sester, M., Toth, C., Trinder, J., Heipke, C., 2016. Information from imagery: ISPRS scientific vision and research agenda. ISPRS J. Photogramm. Remote Sens. 115, 3–21. https://doi.org/10.1016/j.isprsjprs.2015.09.008

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A., 2016a. SEMANTIC IMAGE SEGMENTATION WITH DEEP CONVOLUTIONAL NETS AND FULLY CONNECTED CRFS.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2016b. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. ArXiv160600915 Cs.

Chen, L.-C., Papandreou, G., Schroff, F., Adam, H., 2017. Rethinking atrous convolution for semantic image segmentation. ArXiv Prepr. ArXiv170605587.

Chen, W., Corso, J.J., 2015. Action Detection by Implicit Intentional Motion Clustering, in: 2015 IEEE International Conference on Computer Vision (ICCV). Presented at the 2015 IEEE International Conference on Computer Vision (ICCV), IEEE, Santiago, Chile, pp. 3298–3306. https://doi.org/10.1109/ICCV.2015.377

Chew, R., Jones, K., Unangst, J., Cajka, J., Allpress, J., Amer, S., Krotki, K., 2018. Toward Model-Generated Household Listing in Low- and Middle-Income Countries Using Deep Learning. ISPRS Int. J. Geo-Inf. 7, 448. https://doi.org/10.3390/ijgi7110448

Chew, R.F., Amer, S., Jones, K., Unangst, J., Cajka, J., Allpress, J., Bruhn, M., 2018. Residential scene classification for gridded population sampling in developing countries using deep convolutional neural networks on satellite imagery. Int. J. Health Geogr. 17. https://doi.org/10.1186/s12942-018-0132-1

Cho, G., Rodríguez, D.A., Khattak, A.J., 2009. The role of the built environment in explaining relationships between perceived and actual pedestrian and bicyclist safety. Accid. Anal. Prev. 41, 692–702. https://doi.org/10.1016/j.aap.2009.03.008

Chu, W.-T., Zheng, X.-Y., Ding, D.-S., 2016. Image2Weather: A Large-Scale Image Dataset for Weather Property Estimation, in: 2016 IEEE Second International Conference on Multimedia Big Data (BigMM). Presented at the 2016 IEEE Second International Conference on Multimedia Big Data (BigMM), IEEE, Taipei, Taiwan, pp. 137–144. https://doi.org/10.1109/BigMM.2016.9

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3213–3223.

Dahl, G.E., Sainath, T.N., Hinton, G.E., 2013. Improving deep neural networks for LVCSR using rectified linear units and dropout, in: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference On. IEEE, pp. 8609–8613.

de Hartog, J.J., Boogaard, H., Nijland, H., Hoek, G., 2010. Do the Health Benefits of Cycling Outweigh the Risks? Environ. Health Perspect. 118, 1109–1116. https://doi.org/10.1289/ehp.0901747

De Nadai, M., Vieriu, R.L., Zen, G., Dragicevic, S., Naik, N., Caraviello, M., Hidalgo, C.A., Sebe, N., Lepri, B., 2016. Are Safer Looking Neighborhoods More Lively?: A Multimodal Investigation into Urban Life, in: Proceedings of the 2016 ACM on Multimedia Conference - MM '16. Presented at the the 2016 ACM, ACM Press, Amsterdam, The Netherlands, pp. 1127–1135. https://doi.org/10.1145/2964284.2964312

De Rome, L., Boufous, S., Georgeson, T., Senserrick, T., Richardson, D., Ivers, R., 2014. Bicycle Crashes in Different Riding Environments in the Australian Capital Territory. Traffic Inj. Prev. 15, 81–88. https://doi.org/10.1080/15389588.2013.781591

deGroot, M., Brown, E., 2017. A PyTorch Implementation of Single Shot MultiBox Detector [WWW Document]. URL https://github.com/amdegroot/ssd.pytorch

Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., Hughes, F., Tuia, D., Raskar, R., 2018. DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images 10.

DfT, 2020. Transport Secretary's statement on coronavirus (COVID-19): 9 May 2020 [WWW Document]. Gov.uk. URL https://www.gov.uk/government/speeches/transport-secretarys-statement-on-coronavirus-covid-19-9-may-2020

Diba, A., Fayyaz, M., Sharma, V., Karami, A.H., Arzani, M.M., Yousefzadeh, R., Gool, L.V., 2017. Temporal 3D ConvNets Using Temporal Transition Layer 5.

Dozza, M., Bianchi Piccinini, G.F., Werneke, J., 2016a. Using naturalistic data to assess e-cyclist behavior. Transp. Res. Part F Traffic Psychol. Behav. 41, 217–226. https://doi.org/10.1016/j.trf.2015.04.003

Dozza, M., Schindler, R., Bianchi-Piccinini, G., Karlsson, J., 2016b. How do drivers overtake cyclists? Accid. Anal. Prev. 88, 29–36. https://doi.org/10.1016/j.aap.2015.12.008

Dozza, M., Schwab, A., Wegman, F., 2017. Safety Science Special Issue on Cycling Safety. Saf. Sci. 92, 262–263. https://doi.org/10.1016/j.ssci.2016.06.009

Dozza, M., Werneke, J., 2014. Introducing naturalistic cycling data: What factors influence bicyclists' safety in the real world? Transp. Res. Part F Traffic Psychol. Behav. 24, 83–91. https://doi.org/10.1016/j.trf.2014.04.001

Dozza, M., Werneke, J., Fernandez, A., 2012. Piloting the Naturalistic Methodology on Bicycles. Presented at the International Cycling Safety Conference 2012, p. 11.

Duarte, F., Álvarez, R., 2019. The data politics of the urban age. Palgrave Commun. 5, 1–7. https://doi.org/10.1057/s41599-019-0264-3

Dubey, A., Naik, N., Parikh, D., Raskar, R., Hidalgo, C.A., 2016. Deep learning the city: Quantifying urban perception at a global scale, in: European Conference on Computer Vision. Springer, pp. 196–212.

Elhoseiny, M., Huang, S., Elgammal, A., 2015. Weather classification with deep convolutional neural networks, in: 2015 IEEE International Conference on Image Processing (ICIP). Presented at the 2015 IEEE International Conference on Image Processing (ICIP), IEEE, Quebec City, QC, Canada, pp. 3349–3353. https://doi.org/10.1109/ICIP.2015.7351424

El-Nouby, A., Taylor, G.W., 2018. Real-Time End-to-End Action Detection with Two-Stream Networks. ArXiv180208362 Cs.

Enkelmann, W., n.d. Obstacle detection by evaluation of optical flow fields from image sequences 5.

Escorcia, V., Caba Heilbron, F., Niebles, J.C., Ghanem, B., 2016. DAPs: Deep Action Proposals for Action Understanding, in: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision – ECCV 2016. Springer International Publishing, Cham, pp. 768–784. https://doi.org/10.1007/978-3-319-46487-9_47

Eslami, S.M.A., Jimenez Rezende, D., Besse, F., Viola, F., Morcos, A.S., Garnelo, M., Ruderman, A., Rusu, A.A., Danihelka, I., Gregor, K., Reichert, D.P., Buesing, L., Weber, T., Vinyals, O., Rosenbaum, D., Rabinowitz, N., King, H., Hillier, C., Botvinick, M., Wierstra, D., Kavukcuoglu, K., Hassabis, D., 2018. Neural scene representation and rendering. Science 360, 1204–1210. https://doi.org/10.1126/science.aar6170

Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2015. The Pascal Visual Object Classes Challenge: A Retrospective. Int. J. Comput. Vis. 111, 98–136. https://doi.org/10.1007/s11263-014-0733-5

Faisal, A., Yigitcanlar, T., Kamruzzaman, Md., Currie, G., 2019. Understanding autonomous vehicles: A systematic literature review on capability, impact, planning and policy. J. Transp. Land Use 12. https://doi.org/10.5198/jtlu.2019.1405

Fang, H.-S., Xie, S., Tai, Y.-W., Lu, C., 2017. RMPE: Regional Multi-person Pose Estimation, in: 2017 IEEE International Conference on Computer Vision (ICCV). Presented at the 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, pp. 2353–2362. https://doi.org/10.1109/ICCV.2017.256

Farnebäck, G., 2003. Two-Frame Motion Estimation Based on Polynomial Expansion, in: Bigun, J., Gustavsson, T. (Eds.), Image Analysis, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 363–370. https://doi.org/10.1007/3-540-45103-X_50

Feng, C., Liu, M.-Y., Kao, C.-C., Lee, T.-Y., 2017. Deep Active Learning for Civil Infrastructure Defect Detection and Classification, in: Computing in Civil Engineering 2017. Presented at the ASCE International Workshop on Computing in Civil Engineering 2017, American Society of Civil Engineers, Seattle, Washington, pp. 298–306. https://doi.org/10.1061/9780784480823.036

Freedman, D.A., 2008. On regression adjustments in experiments with several treatments. Ann. Appl. Stat. 2, 176–196. https://doi.org/10.1214/07-AOAS143

Frey, B.B., 2018. Pearson Correlation Coefficient, in: The SAGE Encyclopedia of Educational Research, Measurement, and Evaluation. SAGE Publications, Inc., 2455 Teller Road, Thousand Oaks, California 91320. https://doi.org/10.4135/9781506326139

Fuller, D., Gauvin, L., Morency, P., Kestens, Y., Drouin, L., 2013. The impact of implementing a public bicycle share program on the likelihood of collisions and near misses in Montreal, Canada. Prev. Med. 57, 920–924. https://doi.org/10.1016/j.ypmed.2013.05.028

Gbeminiyi Oluwafemi, A., Zenghui, W., 2019. Multi-Class Weather Classification from Still Image Using Said Ensemble Method, in: 2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA). Presented at the 2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South

Africa (SAUPEC/RobMech/PRASA), IEEE, Bloemfontein, South Africa, pp. 135–140. https://doi.org/10.1109/RoboMech.2019.8704783

Gemert, J.C. van, Jain, M., Gati, E., Snoek, C.G.M., 2015. APT: Action localization proposals from dense trajectories, in: Procedings of the British Machine Vision Conference 2015. Presented at the British Machine Vision Conference 2015, British Machine Vision Association, Swansea, p. 177.1-177.12. https://doi.org/10.5244/C.29.177

Girdhar, R., Gkioxari, G., Torresani, L., Paluri, M., Tran, D., 2017. Detect-and-Track: Efficient Pose Estimation in Videos 10.

Girdhar, R., Ramanan, D., 2017. Attentional Pooling for Action Recognition, in: ArXiv:1711.01467 [Cs]. Presented at the Advances in Neural Information Processing Systems 30 (NeurIPS 2017).

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014a. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition. Presented at the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Columbus, OH, USA, pp. 580–587. https://doi.org/10.1109/CVPR.2014.81

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014b. Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv:1311.2524 [cs.CV].

Gkioxari, G., Girshick, R., Dollar, P., He, K., 2017. Detecting and Recognizing Human-Object Interactions 9.

Glorot, X., Bordes, A., Bengio, Y., 2011. Deep sparse rectifier neural networks, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. pp. 315–323.

Goodfellow, I., 2016. NIPS 2016 Tutorial: Generative Adversarial Networks. ArXiv170100160 Cs.

Goodfellow, I., Bengio, Y., Courville, A., 2017. Deep Learning, Adaptive computation and machine learning series. The MIT Press, Cambridge, Massachusetts.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014a. Generative Adversarial Networks, in: Neural Information Processing Systems Conference.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014b. Generative Adversarial Networks. ArXiv14062661 Cs Stat.

Gopalakrishnan, K., 2018. Deep Learning in Data-Driven Pavement Image Analysis and Automated Distress Detection: A Review. Data 3, 28. https://doi.org/10.3390/data3030028

Griffiths, D., Boehm, J., 2018. RAPID OBJECT DETECTION SYSTEMS, UTILISING DEEP LEARNING AND UNMANNED AERIAL SYSTEMS (UAS) FOR CIVIL ENGINEERING APPLICATIONS. ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLII–2, 391–398. https://doi.org/10.5194/isprs-archives-XLII-2-391-2018

Guler, R.A., Neverova, N., Kokkinos, I., 2018. DensePose: Dense Human Pose Estimation In The Wild 10.

Guo, M., Chou, E., Huang, D.-A., Song, S., Yeung, S., Fei-Fei, L., 2018. Neural Graph Matching Networks for Fewshot 3D Action Recognition 17.

Guo, S., Huang, W., Zhang, H., Zhuang, C., Dong, D., Scott, M.R., Huang, D., 2018. CurriculumNet: Weakly Supervised Learning from Large-Scale Web Images, in: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.), Computer Vision – ECCV 2018, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 139–154. https://doi.org/10.1007/978-3-030-01249-6_9

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M.S., 2016. Deep learning for visual understanding: A review. Neurocomputing 187, 27–48. https://doi.org/10.1016/j.neucom.2015.09.116

Gustafsson, L., Archer, J., 2013. A naturalistic study of commuter cyclists in the greater Stockholm area. Accid. Anal. Prev. 58, 286–298. https://doi.org/10.1016/j.aap.2012.06.004

Hagendorff, T., 2020. The Ethics of AI Ethics: An Evaluation of Guidelines. Minds Mach. 30, 99–120. https://doi.org/10.1007/s11023-020-09517-8

He, H., Yang, D., Wang, Shicheng, Wang, Shuyang, Li, Y., 2019. Road Extraction by Using Atrous Spatial Pyramid Pooling Integrated Encoder-Decoder Network and Structural Similarity Loss. Remote Sens. 11, 1015. https://doi.org/10.3390/rs11091015

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

He, K., Zhang, X., Ren, S., Sun, J., 2015a. Deep Residual Learning for Image Recognition. arXiv:1512.03385v1.

He, K., Zhang, X., Ren, S., Sun, J., 2015b. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision. pp. 1026–1034.

He, L., Wang, G., Hu, Z., 2018. Learning Depth from Single Images with Deep Neural Network Embedding Focal Length. IEEE Trans. Image Process. 27, 4676–4689. https://doi.org/10.1109/TIP.2018.2832296

Helbich, M., Yao, Y., Liu, Y., Zhang, J., Liu, P., Wang, R., 2019. Using deep learning to examine street view green and blue spaces and their associations with geriatric depression in Beijing, China. Environ. Int. 126, 107–117. https://doi.org/10.1016/j.envint.2019.02.013

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Dulac-Arnold, G., Osband, I., Agapiou, J., Leibo, J.Z., Gruslys, A., 2017. Deep Q-learning from Demonstrations. ArXiv170403732 Cs.

Hinton, G.E., Osindero, S., Teh, Y.-W., 2006. A fast learning algorithm for deep belief nets. Neural Comput. 18, 1527–1554.

Hoffman, J.I.E., 2015. Comparison of Two Groups, in: Biostatistics for Medical and Biomedical Practitioners. Elsevier, pp. 337–362. https://doi.org/10.1016/B978-0-12-802387-7.00022-6

Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (Eds.), 2018. Machine Learning and Knowledge Extraction, Lecture Notes in Computer Science. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-99740-7

Hong, S.-J., Han, Y., Kim, S.-Y., Lee, A.-Y., Kim, G., 2019. Application of Deep-Learning Methods to Bird Detection Using Unmanned Aerial Vehicle Imagery. Sensors 19, 1651. https://doi.org/10.3390/s19071651

Hou, R., Chen, C., Shah, M., 2017. Tube Convolutional Neural Network (T-CNN) for Action Detection in Videos, in: 2017 IEEE International Conference on Computer Vision (ICCV). Presented at the 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, pp. 5823–5832. https://doi.org/10.1109/ICCV.2017.620

Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L., 2017. Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. p. 3.

Ibrahim, Mohamed R., Haworth, J., Cheng, T., 2020. Understanding cities with machine eyes: A review of deep computer vision in urban analytics. Cities 96, 102481. https://doi.org/10.1016/j.cities.2019.102481

Ibrahim, M R., Haworth, J., Cheng, T., 2020a. SlipNet: Recognising surface conditions with deep learning. Presented at the SpaceTimeAI 2020, London.

Ibrahim, M.R., Haworth, J., Cheng, T., 2019a. WeatherNet: Recognising Weather and Visual Conditions from Street-Level Images Using Deep Residual Learning. ISPRS Int. J. Geo-Inf. 8, 549. https://doi.org/10.3390/ijgi8120549

Ibrahim, M.R., Haworth, J., Cheng, T., 2019b. URBAN-i: From urban scenes to mapping slums, transport modes, and pedestrians in cities using deep learning and computer vision. Environ. Plan. B Urban Anal. City Sci. 239980831984651. https://doi.org/10.1177/2399808319846517

Ibrahim, M.R., Haworth, J., Christie, N., Cheng, T., 2021. CyclingNet: Detecting cycling near misses from video streams in complex urban scenes with deep learning. IET Intell. Transp. Syst. itr2.12101. https://doi.org/10.1049/itr2.12101

Ibrahim, M R., Haworth, J., Christie, N., Cheng, T., Hailes, S., 2020b. Cycling near misses: a review of the current methods, challenges and the potential of an AI-embedded system. Transp. Rev. 1–25. https://doi.org/10.1080/01441647.2020.1840456

Imprialou, M., Quddus, M., 2017. Crash data quality for road safety research: Current state and future directions. Accid. Anal. Prev. https://doi.org/10.1016/j.aap.2017.02.022

Ioffe, S., Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Presented at the International Conference on Machine Learning (ICML), p. 9.

Islam, M.T., Jacobs, N., Wu, H., Souvenir, R., 2013. Images+Weather: Collection, Validation, and Refinement 7.

Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A., 2017. Image-to-Image Translation with Conditional Adversarial Networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, pp. 5967–5976. https://doi.org/10.1109/CVPR.2017.632

Jegou, S., Drozdzal, M., Vazquez, D., Romero, A., Bengio, Y., 2017. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, Honolulu, HI, USA, pp. 1175–1183. https://doi.org/10.1109/CVPRW.2017.156

Jestico, B., Nelson, T., Winters, M., 2016. Mapping ridership using crowdsourced cycling data. J. Transp. Geogr. 52, 90–97. https://doi.org/10.1016/j.jtrangeo.2016.03.006

Johnson, M., Charlton, J., Oxley, J., Newstead, S., 2010. Naturalistic Cycling Study: Identifying Risk Factors for On-Road Commuter Cyclists. Ann. Adv. Automot. Med. Annu. Sci. Conf. 54, 275–283.

Johnson, M., Newstead, S., Oxley, J., Charlton, J., 2013. Cyclists and open vehicle doors: Crash characteristics and risk factors. Saf. Sci. 59, 135–140. https://doi.org/10.1016/j.ssci.2013.04.010

Johnson, M., Oxley, J., Newstead, S., Charlton, J., 2014. Safety in numbers? Investigating Australian driver behaviour, knowledge and attitudes towards cyclists. Accid. Anal. Prev. 70, 148–154. https://doi.org/10.1016/j.aap.2014.02.010

Juhra, C., Wieskötter, B., Chu, K., Trost, L., Weiss, U., Messerschmidt, M., Malczyk, A., Heckwolf, M., Raschke, M., 2012. Bicycle accidents – Do we only see the tip of the iceberg? Injury 43, 2026–2034. https://doi.org/10.1016/j.injury.2011.10.016

Kale, G.V., Patil, V.H., 2016. A Study of Vision based Human Motion Recognition and Analysis. Int. J. Ambient Comput. Intell. 7, 18.

Kalman, R., 1960. A New Approach to Linear Filtering and Prediction Problems. J. Basic Eng. 12.

Kang, J., Körner, M., Wang, Y., Taubenböck, H., Zhu, X.X., 2018. Building instance classification using street view images. ISPRS J. Photogramm. Remote Sens. 145, 44–59. https://doi.org/10.1016/j.isprsjprs.2018.02.006

Kang, K., Ouyang, W., Li, H., Wang, X., 2016. Object Detection from Video Tubelets with Convolutional Neural Networks. 2016 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR 817–825. https://doi.org/10.1109/CVPR.2016.95

Karras, T., Laine, S., Aila, T., 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. Presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), p. 10.

Kingma, D.P., Ba, J., 2015. Adam: A Method for Stochastic Optimization, in: ArXiv:1412.6980 [Cs]. Presented at the International Conference on Learning Representations (ICLR).

Kocabas, M., Karagoz, S., Akbas, E., 2018. MultiPoseNet: Fast Multi-Person Pose Estimation using Pose Residual Network, in: Computer Vision – ECCV 2018. Presented at the ECCV 2018, Springer.

Krause, J., Sugita, G., Baek, K., Lim, L., 2018. *WTPlant* (What's That Plant?): A Deep Learning System for Identifying Plants in Natural Images, in: Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval - ICMR '18. Presented at the the 2018 ACM, ACM Press, Yokohama, Japan, pp. 517–520. https://doi.org/10.1145/3206025.3206089

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems. pp. 1097–1105.

Kuo, C.-C.J., 2016. Understanding convolutional neural networks with a mathematical model. J. Vis. Commun. Image Represent. 41, 406–413.

Lacherez, P., Wood, J.M., Marszalek, R.P., King, M.J., 2013. Visibility-related characteristics of crashes involving bicyclists and motor vehicles – Responses from an online questionnaire study. Transp. Res. Part F Traffic Psychol. Behav. 20, 52–58. https://doi.org/10.1016/j.trf.2013.04.003

Law, S., Paige, B., Russell, C., 2019. Take a Look Around: Using Street View and Satellite Images to Estimate House Prices. ACM Trans. Intell. Syst. Technol. 10, 1–19. https://doi.org/10.1145/3342240

Law, S., Seresinhe, C.I., Shen, Y., Gutierrez-Roig, M., 2018. Street-Frontage-Net: urban image classification using deep convolutional neural networks. Int. J. Geogr. Inf. Sci. 1–27. https://doi.org/10.1080/13658816.2018.1555832

Lawson, A.R., Pakrashi, V., Ghosh, B., Szeto, W.Y., 2013. Perception of safety of cyclists in Dublin City. Accid. Anal. Prev. 50, 499–511. https://doi.org/10.1016/j.aap.2012.05.029

Leal-Taixé, L., Milan, A., Reid, I., Roth, S., Schindler, K., 2015. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. ArXiv150401942 Cs.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539

Lehtonen, E., Havia, V., Kovanen, A., Leminen, M., Saure, E., 2016. Evaluating bicyclists' risk perception using video clips: Comparison of frequent and infrequent city cyclists. Transp. Res. Part F Traffic Psychol. Behav. 41, 195–203. https://doi.org/10.1016/j.trf.2015.04.006

Levi, G., Hassncer, T., 2015. Age and gender classification using convolutional neural networks, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Presented at the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, Boston, MA, USA, pp. 34–42. https://doi.org/10.1109/CVPRW.2015.7301352

Li, P., Wang, D., Wang, L., Lu, H., 2018. Deep visual tracking: Review and experimental comparison. Pattern Recognit. 76, 323–338. https://doi.org/10.1016/j.patcog.2017.11.007

Li, X., Jie, Z., Wang, W., Liu, C., Yang, J., Shen, X., Lin, Z., Chen, Q., Yan, S., Feng, J., 2017a. FoveaNet: Perspective-Aware Urban Scene Parsing. Presented at the IEEE International Conference of Computer Vision (ICCV 2017), IEEE, p. 9.

Li, X., Jie, Z., Wang, W., Liu, C., Yang, J., Shen, X., Lin, Z., Chen, Q., Yan, S., Feng, J., 2017b. FoveaNet: Perspective-aware Urban Scene Parsing. ArXiv170802421 Cs.

Li, Y., Lan, C., Xing, J., Zeng, W., Yuan, C., Liu, J., 2016. Online Human Action Detection using Joint Classification-Regression Recurrent Neural Networks, in: Lecture Notes in Computer Science. Presented at the ICCV 2016, Springer, pp. 203–220. https://doi.org/10.1007/978-3-319-46478-7_13

Li, Z., Shen, H., Cheng, Q., Liu, Y., You, S., He, Z., 2019. Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors. ISPRS J. Photogramm. Remote Sens. 150, 197–212. https://doi.org/10.1016/j.isprsjprs.2019.02.017

Lin, G., Milan, A., Shen, C., Reid, I., 2017. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollar, P., 2020. Focal Loss for Dense Object Detection. IEEE Trans. Pattern Anal. Mach. Intell. 42, 318–327. https://doi.org/10.1109/TPAMI.2018.2858826

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P., 2014. Microsoft COCO: Common Objects in Context, in: Computer Vision – ECCV 2014. Presented at the ECCV 2014, Springer, pp. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

Liu, C., Tsow, F., Zou, Y., Tao, N., 2016. Particle Pollution Estimation Based on Image Analysis. PLOS ONE 11, e0145955. https://doi.org/10.1371/journal.pone.0145955

Liu, L., Silva, E.A., Wu, C., Wang, H., 2017. A machine learning-based method for the large-scale evaluation of the qualities of the urban environment. Comput. Environ. Urban Syst. 65, 113–125. https://doi.org/10.1016/j.compenvurbsys.2017.06.003

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. Ssd: Single shot multibox detector, in: European Conference on Computer Vision. Springer, pp. 21–37.

Liu, W., Yang, Y., Wei, L., School of Automation, China University of Geosciences, 2017. Weather Recognition of Street Scene Based on Sparse Deep Neural Networks. J. Adv. Comput. Intell. Intell. Inform. 21, 403–408. https://doi.org/10.20965/jaciii.2017.p0403

Liu, Y., Racah, E., Correa, J., Khosrowshahi, A., Lavers, D., Kunkel, K., Wehner, M., Collins, W., 2016. Application of Deep Convolutional Neural Networks for Detecting Extreme Weather in Climate Datasets. Presented at the Int'l Conf. on Advances in Big Data Analytics, p. 8.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Boston, MA, USA, pp. 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965

Lu, C., Lin, D., Jia, J., Tang, C.-K., 2017. Two-Class Weather Classification. IEEE Trans. Pattern Anal. Mach. Intell. 39, 2510–2524. https://doi.org/10.1109/TPAMI.2016.2640295

Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.-I., 2020. From local explanations to global understanding with explainable AI for trees. Nat. Mach. Intell. 2, 56–67. https://doi.org/10.1038/s42256-019-0138-9

Lynch, K., 1960. The image of the city. The technology press and Harvard University press.

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., Omata, H., 2018. Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images: Road damage detection and classification. Comput.-Aided Civ. Infrastruct. Eng. 33, 1127–1141. https://doi.org/10.1111/mice.12387

Mahmud, S.M.S., Ferreira, L., Hoque, Md.S., Tavassoli, A., 2017. Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs. IATSS Res. 41, 153–163. https://doi.org/10.1016/j.iatssr.2017.02.001

Mallot, H.A., Bülthoff, H.H., Little, J.J., Bohrer, S., 1991. Inverse perspective mapping simplifies optical flow computation and obstacle detection. Biol. Cybern. 64, 177–185. https://doi.org/10.1007/BF00201978

Manen, S., Gygli, M., Dai, D., Gool, L.V., 2017. PathTrack: Fast Trajectory Annotation with Path Supervision, in: 2017 IEEE International Conference on Computer Vision (ICCV). Presented at the 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, pp. 290–299. https://doi.org/10.1109/ICCV.2017.40

Marcos, D., Volpi, M., Kellenberger, B., Tuia, D., 2018. Land cover mapping at very high resolution with rotation equivariant CNNs: Towards small yet accurate models. ISPRS J. Photogramm. Remote Sens. 145, 96–107. https://doi.org/10.1016/j.isprsjprs.2018.01.021

McLaren, B.M., 2003. Extensionally defining principles and cases in ethics: An AI model. Artif. Intell. 150, 145–181. https://doi.org/10.1016/S0004-3702(03)00135-8

Mettes, P., van Gemert, J.C., Snoek, C.G.M., 2016. Spot On: Action Localization from Pointly-Supervised Proposals, in: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision – ECCV 2016. Springer International Publishing, Cham, pp. 437–453. https://doi.org/10.1007/978-3-319-46454-1_27

Minaee, S., Minaei, M., Abdolrashidi, A., 2021. Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network. Sensors 21, 3046. https://doi.org/10.3390/s21093046

Mirowski, P., Grimes, M., Malinowski, M., Hermann, K.M., Anderson, K., Teplyashin, D., Simonyan, K., 2018. Learning to Navigate in Cities Without a Map. Presented at the NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 2424–2435.

Mnih, V., Badia, A.P., Mirza, M., Graves, A., Harley, T., Lillicrap, T.P., Silver, D., Kavukcuoglu, K., 2016. Asynchronous Methods for Deep Reinforcement Learning 10.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., 2013. Playing Atari with Deep Reinforcement Learning. ArXiv13125602 Cs.

Mohamed, A.N., Ali, M.M., 2013. HUMAN MOTION ANALYSIS, RECOGNITION AND UNDERSTANDING IN COMPUTER VISION: A REVIEW. J. Eng. Sci. 41, 19.

Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using Deep Learning for Image-Based Plant Disease Detection. Front. Plant Sci. 7. https://doi.org/10.3389/fpls.2016.01419

Naik, N., Kominers, S.D., Raskar, R., Glaeser, E.L., Hidalgo, C.A., 2017. Computer vision uncovers predictors of physical urban change. Proc. Natl. Acad. Sci. 114, 7571–7576. https://doi.org/10.1073/pnas.1619003114

Naik, N., Philipoom, J., Raskar, R., Hidalgo, C., 2014. Streetscore-predicting the perceived safety of one million streetscapes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 779–785.

Naik, N., Raskar, R., Hidalgo, C.A., 2016. Cities Are Physical Too: Using Computer Vision to Measure the Quality and Impact of Urban Appearance. Am. Econ. Rev. 106, 128–132. https://doi.org/10.1257/aer.p20161030

Narang, N., Bourlai, T., 2016. Gender and ethnicity classification using deep learning in heterogeneous face recognition, in: 2016 International Conference on Biometrics (ICB). Presented at the 2016 International Conference on Biometrics (ICB), IEEE, Halmstad, Sweden, pp. 1–8. https://doi.org/10.1109/ICB.2016.7550082

Narazaki, Y., Hoskere, V., Hoang, T.A., Jr, B.F.S., 2017. Vision-based Automated Bridge Component Recognition Integrated With High-level Scene Understanding. Presented at the The 13th International Workshop on Advanced Smart Materials and Smart Structures Technology, p. 10.

Nelson, T.A., Denouden, T., Jestico, B., Laberee, K., Winters, M., 2015. BikeMaps.org: A Global Tool for Collision and Near Miss Mapping. Front. Public Health 3. https://doi.org/10.3389/fpubh.2015.00053

Nguyen, Q.C., Sajjadi, M., McCullough, M., Pham, M., Nguyen, T.T., Yu, W., Meng, H.-W., Wen, M., Li, F., Smith, K.R., Brunisholz, K., Tasdizen, T., 2018. Neighbourhood looking glass: 360° automated characterisation of the built environment for neighbourhood effects research. J. Epidemiol. Community Health 72, 260–266. https://doi.org/10.1136/jech-2017-209456

Oliva, A., Torralba, A., 2006. Chapter 2 Building the gist of a scene: the role of global image features in recognition, in: Progress in Brain Research. Elsevier, pp. 23–36. https://doi.org/10.1016/S0079-6123(06)55002-2

Paganini, M., de Oliveira, L., Nachman, B., 2018. CaloGAN: Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. Phys. Rev. D 97. https://doi.org/10.1103/PhysRevD.97.014021

Parikh, D., Polikar, R., 2007. An Ensemble-Based Incremental Learning Approach to Data Fusion. IEEE Trans. Syst. Man Cybern. Part B Cybern. 37, 437–450. https://doi.org/10.1109/TSMCB.2006.883873

Parkin, J., Meyers, C., 2010. The effect of cycle lanes on the proximity between motor traffic and cycle traffic. Accid. Anal. Prev. 42, 159–165. https://doi.org/10.1016/j.aap.2009.07.018

Paschalidis, E., Basbas, S., Politis, I., Prodromou, M., 2016. '"Put the blame on. . .others!"': The battle of cyclists against pedestrians and car drivers at the urban environment. A cyclists' perception study 18.

Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J., 2017. Large Kernel Matters — Improve Semantic Segmentation by Global Convolutional Network, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, pp. 1743–1751. https://doi.org/10.1109/CVPR.2017.189

Poulos, R.G., Hatfield, J., Rissel, C., Grzebieta, R., McIntosh, A.S., 2012. Exposure-based cycling crash, near miss and injury rates: The Safer Cycling Prospective Cohort Study protocol: Figure 1. Inj. Prev. 18, e1–e1. https://doi.org/10.1136/injuryprev-2011-040160

PRISMA, 2015. TRANSPARENT REPORTING of SYSTEMATIC REVIEWS and META-ANALYSES [WWW Document]. PRISMA. URL http://prisma-statement.org/

Priya, G., Paul, S.N., Singh, Y.J., 2015. Human walking motion detection and classification of actions from Video Sequences 3, 6.

Pucher, J., Dill, J., Handy, S., 2010. Infrastructure, programs, and policies to increase bicycling: An international review. Prev. Med. 50, S106–S125. https://doi.org/10.1016/j.ypmed.2009.07.028

Quercia, D., O'Hare, N.K., Cramer, H., 2014. Aesthetic capital: what makes london look beautiful, quiet, and happy?, in: Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing - CSCW '14. Presented at the the 17th ACM conference, ACM Press, Baltimore, Maryland, USA, pp. 945–955. https://doi.org/10.1145/2531602.2531613

Radford, A., Metz, L., Chintala, S., 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. ArXiv151106434 Cs.

Rahwan, I., Cebrian, M., Obradovich, N., Bongard, J., Bonnefon, J.-F., Breazeal, C., Crandall, J.W., Christakis, N.A., Couzin, I.D., Jackson, M.O., Jennings, N.R., Kamar, E., Kloumann, I.M., Larochelle, H., Lazer, D., McElreath, R., Mislove, A., Parkes, D.C., Pentland, A. 'Sandy,' Roberts, M.E., Shariff, A., Tenenbaum, J.B., Wellman, M., 2019. Machine behaviour. Nature 568, 477–486. https://doi.org/10.1038/s41586-019-1138-y

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

Redmon, J., Farhadi, A., 2018a. YOLOv3: An Incremental Improvement 6.

Redmon, J., Farhadi, A., 2018b. YOLOv3: An Incremental Improvement 6.

Redmon, J., Farhadi, A., 2017. YOLO9000: Better, Faster, Stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, pp. 6517–6525. https://doi.org/10.1109/CVPR.2017.690

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H., 2016. Generative Adversarial Text to Image Synthesis, in: ICML'16: Proceedings of the 33rd International Conference on International Conference on Machine Learning. pp. 1060–1069.

Reed, S.E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., Lee, H., 2016. Learning What and Where to Draw 9.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems, pp. 91–99.

Riley, R.D., Kauser, I., Bland, M., Thijs, L., Staessen, J.A., Wang, J., Gueyffier, F., Deeks, J.J., 2013. Meta-analysis of randomised trials with a continuous outcome according to baseline imbalance and availability of individual participant data. Stat. Med. 32, 2747–2766. https://doi.org/10.1002/sim.5726

Robie, A.A., Seagraves, K.M., Egnor, S.E.R., Branson, K., 2017. Machine vision methods for analyzing social interactions. J. Exp. Biol. 220, 25–34. https://doi.org/10.1242/jeb.142281

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation, in: ArXiv:1505.04597 [Cs]. Presented at the Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer. https://doi.org/10.1007/978-3-319-24574-4_28

Roser, M., Moosmann, F., 2008. Classification of weather situations on single color images, in: 2008 IEEE Intelligent Vehicles Symposium. Presented at the 2008 IEEE Intelligent Vehicles Symposium (IV), IEEE, Eindhoven, Netherlands, pp. 798–803. https://doi.org/10.1109/IVS.2008.4621205

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. Int. J. Comput. Vis. 115, 211–252. https://doi.org/10.1007/s11263-015-0816-y

Saha, S., Singh, G., Cuzzolin, F., 2017. AMTnet: Action-Micro-Tube Regression by End-to-end Trainable Deep Architecture, in: 2017 IEEE International Conference on Computer Vision (ICCV). Presented at the 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, pp. 4424–4433. https://doi.org/10.1109/ICCV.2017.473

Saha, S., Singh, G., Sapienza, M., Torr, P.H.S., Cuzzolin, F., 2016. Deep Learning for Detecting Multiple Space-Time Action Tubes in Videos, in: British Machine Vision Conference (BMVC).

Salesses, P., Schechtner, K., Hidalgo, C.A., 2013. The Collaborative Image of The City: Mapping the Inequality of Urban Perception. PLoS ONE 8, e68400. https://doi.org/10.1371/journal.pone.0068400

Sanders, R.L., 2015. Perceived traffic risk for cyclists: The impact of near miss and collision experiences. Accid. Anal. Prev. 75, 26–34. https://doi.org/10.1016/j.aap.2014.11.004

Sandler, E., 2011. get_lat_lon_exif_pil.py.

Savan, B., Cohlmeyer, E., Ledsham, T., 2017. Integrated strategies to accelerate the adoption of cycling for transportation. Transp. Res. Part F Traffic Psychol. Behav. 46, 236–249. https://doi.org/10.1016/j.trf.2017.03.002

Sayed, T., Zaki, M.H., Autey, J., 2013. Automated safety diagnosis of vehicle–bicycle interactions using computer vision analysis. Saf. Sci. 59, 163–172. https://doi.org/10.1016/j.ssci.2013.05.009

Scherer, D., Müller, A., Behnke, S., 2010. Evaluation of pooling operations in convolutional architectures for object recognition, in: International Conference on Artificial Neural Networks. Springer, pp. 92–101.

Schleinitz, K., Petzoldt, T., Franke-Bartholdt, L., Krems, J., Gehlert, T., 2017. The German Naturalistic Cycling Study – Comparing cycling speed of riders of different e-bikes and conventional bicycles. Saf. Sci. 92, 290–297. https://doi.org/10.1016/j.ssci.2015.07.027

Schleinitz, K., Petzoldt, T., Franke-Bartholdt, L., Krems, J.F., Gehlert, T., 2015. Conflict partners and infrastructure use in safety critical events in cycling – Results from a naturalistic cycling study. Transp. Res. Part F Traffic Psychol. Behav. 31, 99–111. https://doi.org/10.1016/j.trf.2015.04.002

Schlögl, M., Stütz, R., 2017. Methodological considerations with data uncertainty in road safety analysis. Accid. Anal. Prev. https://doi.org/10.1016/j.aap.2017.02.001

Schroeder, D.A., 2010. Discrete choice models. Account. Causal Eff. 77–95.

Seresinhe, C.I., Preis, T., Moat, H.S., 2017. Using deep learning to quantify the beauty of outdoor places. R. Soc. Open Sci. 4, 170170. https://doi.org/10.1098/rsos.170170

Sharma, A., Liu, X., Yang, X., Shi, D., 2017. A patch-based convolutional neural network for remote sensing image classification. Neural Netw. 95, 19–28. https://doi.org/10.1016/j.neunet.2017.07.017

Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.-F., 2017. CDC: Convolutional-De-Convolutional Networks for Precise Temporal Action Localization in Untrimmed Videos. ArXiv170301515 Cs.

Simonyan, K., Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. Presented at the International Conference on Learning Representations (ICLR 2015).

Singh, G., Saha, S., Sapienza, M., Torr, P., Cuzzolin, F., 2017. Online Real-Time Multiple Spatiotemporal Action Localisation and Prediction, in: 2017 IEEE International Conference on Computer Vision (ICCV). Presented at the 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, pp. 3657–3666. https://doi.org/10.1109/ICCV.2017.393

Sirirattanapol, C., Nagai, M., Witayangkurn, A., Pravinvongvuth, S., Ekpanyapong, M., 2019. Bangkok CCTV Image through a Road Environment Extraction System Using Multi-Label Convolutional Neural Network Classification. ISPRS Int. J. Geo-Inf. 8, 128. https://doi.org/10.3390/ijgi8030128

Smalheiser, N.R., 2017. Null Hypothesis Statistical Testing and the t-Test, in: Data Literacy. Elsevier, pp. 127–136. https://doi.org/10.1016/B978-0-12-811306-6.00009-9

Soomro, K., Shah, M., 2017. Unsupervised Action Discovery and Localization in Videos, in: 2017 IEEE International Conference on Computer Vision (ICCV). Presented at the 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, pp. 696–705. https://doi.org/10.1109/ICCV.2017.82

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.

Srivastava, S., Vargas-Muñoz, J.E., Tuia, D., 2019. Understanding urban landuse from the above and ground perspectives: A deep learning, multimodal solution. Remote Sens. Environ. 228, 129–143. https://doi.org/10.1016/j.rse.2019.04.014

Steinbach, R., Green, J., Datta, J., Edwards, P., 2011. Cycling and the city: A case study of how gendered, ethnic and class identities can shape healthy transport choices. Soc. Sci. Med. 72, 1123–1130. https://doi.org/10.1016/j.socscimed.2011.01.033

Strauss, J., Miranda-Moreno, L.F., Morency, P., 2013. Cyclist activity and injury risk analysis at signalized intersections: A Bayesian modelling approach. Accid. Anal. Prev. 59, 9–17. https://doi.org/10.1016/j.aap.2013.04.037

Stubbings, P., Peskett, J., Rowe, F., Arribas-Bel, D., 2019. A Hierarchical Urban Forest Index Using Street-Level Imagery and Deep Learning 22.

Sun, D., Roth, S., Black, M.J., 2010. Secrets of optical flow estimation and their principles, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Presented at the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, San Francisco, CA, USA, pp. 2432–2439. https://doi.org/10.1109/CVPR.2010.5539939

Sun, Y., Liu, Y., Wang, G., Zhang, H., 2017. Deep Learning for Plant Identification in Natural Environment. Comput. Intell. Neurosci. 2017, 1–6. https://doi.org/10.1155/2017/7361042

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., 2015. Going Deeper with Convolutions. Computer Vision Foundation.

Tarigan, J., Nadia, Diedan, R., Suryana, Y., 2017. Plate Recognition Using Backpropagation Neural Network and Genetic Algorithm. Procedia Comput. Sci. 116, 365–372. https://doi.org/10.1016/j.procs.2017.10.068

Teschke, K., Frendo, T., Shen, H., Harris, M.A., Reynolds, C.C., Cripton, P.A., Brubacher, J., Cusimano, M.D., Friedman, S.M., Hunte, G., Monro, M., Vernich, L., Babul, S., Chipman, M., Winters, M.,

2014. Bicycling crash circumstances vary by route type: a cross-sectional analysis. BMC Public Health 14. https://doi.org/10.1186/1471-2458-14-1205

TfL, 2018. Cycling action plan: Making London the world's best big city for cycling. London.

Tian, K., Zhou, S., Guan, J., 2017. DeepCluster: A General Clustering Framework Based on Deep Learning, in: Ceci, M., Hollmén, J., Todorovski, L., Vens, C., Džeroski, S. (Eds.), Machine Learning and Knowledge Discovery in Databases. Springer International Publishing, Cham, pp. 809–825. https://doi.org/10.1007/978-3-319-71246-8_49

van Hasselt, H., Guez, A., Silver, D., 2015. Deep Reinforcement Learning with Double Q-Learning 7.

Vanhoey, K., Dai, D., Van Gool, L., de Oliveira, C.E.P., Riemenschneider, H., Bódis-Szomorú, A., Manén, S., Paudel, D.P., Gygli, M., Kobyshev, N., Kroeger, T., 2017. VarCity - the video: the struggles and triumphs of leveraging fundamental research results in a graphics video production, in: ACM SIGGRAPH 2017 Talks on - SIGGRAPH '17. Presented at the ACM SIGGRAPH 2017 Talks, ACM Press, Los Angeles, California, pp. 1–2. https://doi.org/10.1145/3084363.3085085

Vanparijs, J., Int Panis, L., Meeusen, R., de Geus, B., 2015. Exposure measurement in bicycle safety analysis: A review of the literature. Accid. Anal. Prev. 84, 9–19. https://doi.org/10.1016/j.aap.2015.08.007

Vansteenkiste, P., Zeuwts, L., Cardon, G., Lenoir, M., 2016. A hazard-perception test for cycling children: An exploratory study. Transp. Res. Part F Traffic Psychol. Behav. 41, 182–194. https://doi.org/10.1016/j.trf.2016.05.001

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention Is All You Need, in: ArXiv:1706.03762 [Cs]. Presented at the Proceedings of Conference on Neural Information Processing Systems (NIPS 201, Long Beach, CA, USA.

Villarreal Guerra, J.C., Khanam, Z., Ehsan, S., Stolkin, R., McDonald-Maier, K., 2018. Weather Classification: A new multi-class dataset, data augmentation approach and comprehensive evaluations of Convolutional Neural Networks, in: 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS). Presented at the 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), IEEE, Edinburgh, pp. 305–310. https://doi.org/10.1109/AHS.2018.8541482

Viola, P., Jones, M., 2001. Rapid object detection using a boosted cascade of simple features, in: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference On. IEEE, p. I–I.

Walker, I., Garrard, I., Jowitt, F., 2014. The influence of a bicycle commuter's appearance on drivers' overtaking proximities: An on-road test of bicyclist stereotypes, high-visibility clothing and safety aids in the United Kingdom. Accid. Anal. Prev. 64, 69–77. https://doi.org/10.1016/j.aap.2013.11.007

Wang, B., Zhao, W., Gao, P., Zhang, Y., Wang, Z., 2018. Crack Damage Detection Method via Multiple Visual Features and Efficient Multi-Task Learning Model. Sensors 18, 1796. https://doi.org/10.3390/s18061796

Wang, L., Qiao, Y., Tang, X., 2015. Action recognition with trajectory-pooled deep-convolutional descriptors, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Boston, MA, USA, pp. 4305–4314. https://doi.org/10.1109/CVPR.2015.7299059

Wang, L., Xu, X., Dong, H., Gui, R., Pu, F., 2018. Multi-Pixel Simultaneous Classification of PolSAR Image Using Convolutional Neural Networks. Sensors 18, 769. https://doi.org/10.3390/s18030769

Wang, S., Quan, D., Liang, X., Ning, M., Guo, Y., Jiao, L., 2018. A deep learning framework for remote sensing image registration. ISPRS J. Photogramm. Remote Sens. https://doi.org/10.1016/j.isprsjprs.2017.12.012

Wang, W., Yang, S., He, Z., Wang, M., Zhang, J., Zhang, W., 2018. Urban Perception of Commercial Activeness from Satellite Images and Streetscapes, in: Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18. Presented at the Companion of the The Web

Conference 2018, ACM Press, Lyon, France, pp. 647–654. https://doi.org/10.1145/3184558.3186581

Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., de Freitas, N., 2016. Dueling Network Architectures for Deep Reinforcement Learning, in: The 33rd International Conference on Machine Learning. Presented at the PMLR, pp. 1995–2003.

Weinzaepfel, P., Harchaoui, Z., Schmid, C., 2015. Learning to Track for Spatio-Temporal Action Localization, in: 2015 IEEE International Conference on Computer Vision (ICCV). Presented at the 2015 IEEE International Conference on Computer Vision (ICCV), IEEE, Santiago, Chile, pp. 3164–3172. https://doi.org/10.1109/ICCV.2015.362

Weinzaepfel, P., Martin, X., Schmid, C., 2016. Human Action Localization with Sparse Spatial Supervision. ArXiv160505197 Cs.

White, H., Lu, X., 2010. Granger Causality and Dynamic Structural Systems. J. Financ. Econom. 8, 193–243. https://doi.org/10.1093/jjfinec/nbq006

Williams, D., Britten, A., McCallum, S., Jones, H., Aitkenhead, M., Karley, A., Loades, K., Prashar, A., Graham, J., 2017. A method for automatic segmentation and splitting of hyperspectral images of raspberry plants collected in field conditions. Plant Methods 13. https://doi.org/10.1186/s13007-017-0226-y

Winters, M., Branion-Calles, M., 2017. Cycling safety: Quantifying the under reporting of cycling incidents in Vancouver, British Columbia. J. Transp. Health 7, 48–53. https://doi.org/10.1016/j.jth.2017.02.010

Wurm, M., Stark, T., Zhu, X.X., Weigand, M., Taubenböck, H., 2019. Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks. ISPRS J. Photogramm. Remote Sens. 150, 59–69. https://doi.org/10.1016/j.isprsjprs.2019.02.006

Xiao, Y., Tian, Z., Yu, J., Zhang, Y., Liu, S., Du, S., Lan, X., 2020. A review of object detection based on deep learning. Multimed. Tools Appl. 79, 23729–23791. https://doi.org/10.1007/s11042-020-08976-6

Xie, J., Girshick, R., Farhadi, A., 2016. Unsupervised Deep Embedding for Clustering Analysis 10.

Xu, H., Das, A., Saenko, K., 2017. R-C3D: Region Convolutional 3D Network for Temporal Activity Detection, in: 2017 IEEE International Conference on Computer Vision (ICCV). Presented at the 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, pp. 5794–5803. https://doi.org/10.1109/ICCV.2017.617

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y., 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, in: Proceedings of the 32nd International Conference on Machine Learning. Presented at the PMLR, pp. 2048–2057.

Yang, D., Liu, X., He, H., Li, Y., 2019. Air-to-ground multimodal object detection algorithm based on feature association learning. Int. J. Adv. Robot. Syst. 16, 172988141984299. https://doi.org/10.1177/1729881419842995

Yang, M., Yu, K., Zhang, C., Li, Z., Yang, K., 2018. DenseASPP for Semantic Segmentation in Street Scenes 9.

Yang, S.C.-H., Vong, W.K., Sojitra, R.B., Folke, T., Shafto, P., 2021. Mitigating belief projection in explainable artificial intelligence via Bayesian teaching. Sci. Rep. 11, 9863. https://doi.org/10.1038/s41598-021-89267-4

Yang, Z., Pun-Cheng, L.S.C., 2018. Vehicle detection in intelligent transportation systems and its applications under varying environments: A review. Image Vis. Comput. 69, 143–154. https://doi.org/10.1016/j.imavis.2017.09.008

Yu, F., Koltun, V., 2016. Multi-scale context aggregation by dilated convolutions, in: International Conference on Learning Representations (ICLR). Presented at the ICLR 2016.

Yu, H., Wu, Z., Wang, S., Wang, Y., Ma, X., 2017. Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks. Sensors 17, 1501. https://doi.org/10.3390/s17071501

Zaki, M.H., Sayed, T., 2013. A framework for automated road-users classification using movement trajectories. Transp. Res. Part C Emerg. Technol. 33, 50–73. https://doi.org/10.1016/j.trc.2013.04.007

Zaki, M.H., Sayed, T., Tageldin, A., Hussein, M., 2013. Application of Computer Vision to Diagnosis of Pedestrian Safety Issues. Transp. Res. Rec. J. Transp. Res. Board 2393, 75–84. https://doi.org/10.3141/2393-09

Zangenehpour, S., Miranda-Moreno, L.F., Saunier, N., 2015. Automated classification based on video data at intersections with heavy pedestrian and bicycle traffic: Methodology and application. Transp. Res. Part C Emerg. Technol. 56, 161–176. https://doi.org/10.1016/j.trc.2015.04.003

Zangenehpour, S., Strauss, J., Miranda-Moreno, L.F., Saunier, N., 2016. Are signalized intersections with cycle tracks safer? A case–control study based on automated surrogate safety analysis using video data. Accid. Anal. Prev. 86, 161–172. https://doi.org/10.1016/j.aap.2015.10.025

Zhang, B., Wang, L., Wang, Z., Qiao, Y., Wang, H., 2016. Real-Time Action Recognition with Enhanced Motion Vector CNNs, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, pp. 2718–2726. https://doi.org/10.1109/CVPR.2016.297

Zhang, F., Wu, L., Zhu, D., Liu, Y., 2019. Social sensing from street-level imagery: A case study in learning spatio-temporal urban mobility patterns. ISPRS J. Photogramm. Remote Sens. 153, 48–58. https://doi.org/10.1016/j.isprsjprs.2019.04.017

Zhang, F., Zhou, B., Liu, L., Liu, Y., Fung, H.H., Lin, H., Ratti, C., 2018. Measuring human perceptions of a large-scale urban region using machine learning. Landsc. Urban Plan. 180, 148–160. https://doi.org/10.1016/j.landurbplan.2018.08.020

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D., 2017. StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks, in: 2017 IEEE International Conference on Computer Vision (ICCV). Presented at the 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, pp. 5908–5916. https://doi.org/10.1109/ICCV.2017.629

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D., 2016. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. ArXiv161203242 Cs Stat.

Zhang, S., Wei, Z., Nie, J., Huang, L., Wang, S., Li, Z., 2017. A Review on Human Activity Recognition Using Vision-Based Method. J. Healthc. Eng. 2017, 1–31. https://doi.org/10.1155/2017/3090343

Zhang, X., Xia, G.-S., Lu, Q., Shen, W., Zhang, L., 2018. Visual object tracking by correlation filters and online learning. ISPRS J. Photogramm. Remote Sens. 140, 77–89. https://doi.org/10.1016/j.isprsjprs.2017.07.009

Zhao, B., Li, X., Lu, X., Wang, Z., 2018. A CNN–RNN architecture for multi-label weather recognition. Neurocomputing 322, 47–57. https://doi.org/10.1016/j.neucom.2018.09.048

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2017. Pyramid scene parsing network, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 2881–2890.

Zhao, J., Liu, X., Kuang, Y., Chen, Y.V., Yang, B., 2018. Deep CNN-Based Methods to Evaluate Neighborhood-Scale Urban Valuation Through Street Scenes Perception, in: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC). Presented at the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), IEEE, Guangzhou, pp. 20–27. https://doi.org/10.1109/DSC.2018.00012

Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D., 2017. Temporal Action Detection With Structured Segment Networks, in: 2017 IEEE International Conference on Computer Vision (ICCV). Presented at the ICCV.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A., 2017. Scene Parsing through ADE20K Dataset, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, pp. 5122–5130. https://doi.org/10.1109/CVPR.2017.544

Zhou, W., Li, Q., 2013. Complexity and Dynamic Modeling of Urban System. Int. J. Mach. Learn. Comput. 3, 440–444. https://doi.org/10.7763/IJMLC.2013.V3.356

Zhu, H., Vial, R., Lu, S., 2017. TORNADO: A Spatio-Temporal Convolutional Regression Network for Video Action Proposal, in: 2017 IEEE International Conference on Computer Vision (ICCV). Presented at the 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, pp. 5814–5822. https://doi.org/10.1109/ICCV.2017.619

Zhu, Y., Lan, Z., Newsam, S., Hauptmann, A.G., 2019. Hidden Two-Stream Convolutional Networks for Action Recognition, in: Computer Vision – ACCV 2018. Presented at the ACCV 2018, Springer, 11363. https://doi.org/10.1007/978-3-030-20893-6_23

APPENDIX

Training details of the introduced models

Base Model, WeatherNet and SlipNet: After applying the architecture for each model, we followed the standard procedures for training classification tasks based on the models described in Chapter II. The output layer is based on the number of outputs for each model. It is activated based on a sigmoid function for binary outputs or softmax function for non-binary outputs. The model is trained using a back-propagation of error algorithm to update the weights of the neurons of a batch size of 32, with a momentum of 0.9 and a learning rate of 0.01. It is compiled based on the optimization algorithm of stochastic gradient descent, relying on 'adam' optimizer (Kingma and Ba, 2015). The base model is trained, without freezing any layers' weights, by three epochs; each consists of 9000 steps for training and 2000 steps for validation. All models within the ensemble of WeatherNet and SlipNet are trained for 100 epochs after freezing all layers of ResNet50, except for the fully-connected layers. The accuracy of the model is based on the cost function of the Cross-Entropy error.

SSD Object detection model: After implementing the introduced architecture, the model has been trained using stochastic gradient descent with an initial learning rate of 0.001 that decays after the first 320k iterations. Further implementation details can be found by W. Liu et al. (2016).

CyclingNet: After implemented the introduced architecture, the final output layer consists of a single neuron and is activated with a sigmoid function. The final model is compiled with stochastic gradient descent, relying on 'adam' optimiser, with a momentum of 0.9 and a learning rate of 0.001. The model is set to be trained for maximum training cycles (epochs) of 100, with an early stopping technique, monitoring the change in loss with a patience value of 20 epochs. The same hyperparameters were applied to all other base models introduced in Chapter VI.