Doctoral Thesis

Responsibility Evaluation in Vehicle Collisions from Driving Recorder Videos

車両対車両衝突事故における
ドライブレコーダ映像に基づく責任評価手法について

Supervisor

Tadachika OZONO

Nagoya Institute of Technology

Department of Computer Science

Graduate School of Engineering

Admission: April 1, 2021

Student ID: 33514006

Full name: Helton Agbewonou YAWOVI

January 23, 2024

Doctoral Thesis

Responsibility Evaluation in Vehicle Collisions from Driving Recorder Videos

車両対車両衝突事故における
ドライブレコーダ映像に基づく責任評価手法について

Supervisor

Tadachika OZONO

Nagoya Institute of Technology

Department of Computer Science

Graduate School of Engineering

Admission: April 1, 2021

Student ID: 33514006

Full name: Helton Agbewonou YAWOVI

January 23, 2024

Abstract

Traffic collisions pose a significant global concern, necessitating innovative solutions to enhance safety. Post-collision, it becomes crucial for the police to determine the responsibilities of the involved parties, distinguishing between criminal acts and non-criminal incidents. Insurance companies also rely on such investigations to compensate victims. Assessing responsibility after a crash is a complex task requiring advanced knowledge of road rules. For straightforward scenarios like crashes with traffic lights, decisions are fast and easy. However, in situations such as crashes without any traffic signs, expert knowledge is essential. Automating such tasks demands innovative approaches, representing a necessity for the future of the automobile and insurance industries. Despite these critical needs, there has been limited research in the domain. This study introduces a system that is capable of detecting vehicle collisions within crash videos and implements an original responsibility assessment process to assess drivers' responsibilities. This system aims to provide accurate and timely evaluation of collision incidents, facilitating fair responsibility attribution. It employs object detection for collision detection along with an original algorithm and process that associate a knowledge rule-based system and open data for responsibility assessment. The entire responsibility assessment process involves four steps: (1) detecting the crash time within a crash video, (2) identifying all traffic lights within the video, (3) obtaining road information from the OpenStreetMap API, such as road width and the presence of other traffic signs if necessary, analyzing and processing the information, and (4) utilizing a rule-based knowledge system of road rules, vehicle speed, and orientation to deduce the probable responsibility of each party involved. The system focuses on head-on (front impact) collisions and angle collisions (left and right-side impacts) involving two cars and facilitates the seamless sharing of evaluation results with the police and insurance companies within minutes of a collision. By employing advanced image processing techniques, the system enables prompt detection and analysis of collision incidents. The integration of open data enhances the contextual understanding of the road environment, contributing to more accurate responsibility assessments by improving the performance of responsibilities evaluation mainly during nighttime with traffic lights. The first prototype of the system supported only head-on/angle crash scenarios with traffic lights within two weather conditions groups (bad weather/good weather). The second prototype was improved to support three types of head-on/angle crash scenarios without traffic lights (priority roads/one-way roads/roads with the same width) within three distinct weather conditions (sunny/cloudy/rainy), in addition to those with traffic lights. In its current version, the system now accommodates six different types of head-on/angle crash scenarios without traffic lights (priority roads/one-way roads/roads with the same width/roads with stop signs/roads with speed limit signs/roads with flashing red or yellow signals) within six harsh weather conditions (sunny/cloudy/rainy/stormy/snowy/foggy). The support of these additional scenarios and weather demanded retraining crash detection and traffic light detection models, adding more rules to the knowledge-based system, and enhancing the algorithm and process of responsibility assessment built in the first prototype. Additionally, extensive experiments are conducted with results showing that the system performs better than its previous versions, mainly during nighttime without traffic lights (up to 93% accuracy against up to 82.5% obtained previously). The significant difference and advantage of this system over existing ones Pis its automation of responsibilities evaluation for the police, claims adjusters, and victims themselves as well as its applicability for autonomous vehicles. Moreover, through case studies and comparisons with existing research, the effectiveness and superiority of this system are demonstrated. This study is among the first to enable machines to automatically assess the responsibility of drivers within a crash. It can serve as one of the precursors and foundations for automatic responsibility assessments in autonomous vehicles.

論文要旨

交通事故は重要で世界的な懸念事項であり，安全性を向上させるためには革新的な解決策が必要である．交通事故の発生後，警察にとって当事者の責任を判断し，犯罪行為と非犯罪の事例を区別することが非常に重要である．また，保険会社も被害者に対する補償のために警察の判断に頼っている．事故後の責任の評価は、道路規則の高度な知識を必要とする複雑なタスクである．道路と交通信号機のみのシンプルなシナリオの場合，その評価は迅速で容易である．しかし，交通標識のない状況における事故では，専門的な知識が不可欠である．こうしたタスクの自動化には，未来の自動車および保険業界のために必須の革新的なアプローチが求められる．しかし，これらは重要なニーズにもかかわらず，この分野では研究が限られている．本研究では，クラッシュ動画内の車両衝突を検出し，運転者の責任を評価するためのオリジナルの責任評価プロセスを実装できるシステムを紹介する．本システムは，衝突事件の正確で迅速な評価を提供し，公正な責任の帰属を容易にすることを目指す．本システムは，衝突検出に対してオブジェクト検出を使用し，知識ベースのルールシステムと責任評価のためのオリジナルのアルゴリズムとプロセスを組み合わせによって構成される．責任評価プロセス全体は次の 4 つのステップからなる．:（1）クラッシュ動画内での衝突時刻の検出，（2）クラッシュ動画内のすべての交通信号機の識別，（3）OpenStreetMap API からの道路情報の取得（必要に応じて道路の幅や他の交通標識の存在など），情報の分析と処理，および（4）道路規則，車両速度，および方向をもとに知識ベースシステムを使用して各当事者の責任を推定します．本システムは，2 台の車が関与する対向衝突を対象とし，衝突発生後数分以内に評価結果を警察と保険会社とシームレスに共有することを目指す．高度な画像処理技術を使用することで，本システムは迅速に衝突事件を検出し，分析する．オープンデータの統合により，道路環境の文脈をより考慮し，特に夜間の交通信号機のない状況での責任評価の性能を向上させた．本システムの最初のプロトタイプは，悪天候/良い天候の 2 つの天

候条件グループ内の交通信号機付きの対向衝突シナリオのみをサポートする．2 番目のプロトタイプは，交通信号機のない 3 つの対向衝突シナリオ（優先道路/一方通行道路/同じ幅の道路）をサポートするように改良され，最終的に，交通信号機のあるシナリオに加えて、晴れた日/曇りの日/雨の日の 3 つの異なる天候条件で、交通信号機のない 6 つの異なる対向衝突シナリオ（優先道路/一方通行道路/同じ幅の道路/停止標識のある道路/速度制限標識のある道路/赤または黄色の点滅信号のある道路）をサポートする．これらの追加のシナリオと天候のサポートには，クラッシュ検出および交通信号検出モデルの再トレーニング，知識ベースシステムへの追加のルール，責任評価のアルゴリズムとプロセスの向上が必要であった．さらに，多くの実験が行われ，結果として，本システムが以前のバージョンよりも特に夜間の交通信号機のない状況で優れた性能を発揮していることを示した（以前の 82.5% に対して 93% の正確性）．本システムの既存のものに対する重要な違いと利点は，警察・損害調整者・被害者自身のための責任評価の自動化だけでなく，車載システムとして適用可能であることである．さらに，事例研究や既存研究との比較を通じて，このシステムの有効性と優位性を実証した．本研究は，機械が自動的にクラッシュ動画内で運転者の責任を評価することを初めて実現するものの 1 つであり，自動車の自動責任評価の先駆けと基盤の一部として機能することが期待される．

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background of this Research

Vehicle collisions, also known as road traffic crashes, are global social issues that communities worldwide face on a daily basis. According to a recent report from the World Health Organization (WHO) (World Health Organization 2018), approximately 1.3 million lives are tragically cut short each year due to road traffic crashes. Additionally, between 20 and 50 million people suffer non-fatal injuries, many of whom are left with disabilities as a result. These incidents not only cause significant personal and familial hardships but also lead to substantial economic losses for individuals, families, and nations. Unfortunately, as long as vehicles exist, vehicle collisions will continue to occur.

Vehicle collisions can occur due to various factors and circumstances (Bucsuházy et al. 2020), involving multiple actors and transpiring in different situations (Buss, Abishev, and Baltabekova 2019). Following a collision between two vehicles, it becomes crucial to investigate the causes of the event and determine the responsibilities of each involved party. Typically, the police, responsible for conducting investigations, and claims adjusters, internal experts within insurance companies, proceed manually by visiting the crash site, gathering data (including collision videos recorded by driving recorders, if available), assessing responsibilities based on their knowledge of road rules, and finally generating crash reports. Unfortunately, this process typically takes between three (3) and fifteen (15) days to complete. Subsequently, insurance companies assign claims to their claims adjusters, who require approximately thirty (30) days to evaluate the insurance claims and determine the compensation amounts. Similar to the police, claims adjusters may gather their own data and evidence on the collision causes (including the collection of collision videos from driving recorders) to ascertain liability before giving compensation to the victims. Consequently, victims must patiently await the completion of this entire process before receiving their rightful compensation. This highlights the crucial need for innovative solutions that can shorten decision time and help victims obtain their compensations faster. Despite these critical needs, there has been limited research dedicated to evaluating responsibility in the aftermath of vehicle collisions.

1.2 Traditional Method of Road Accident Responsibilities' Evaluation

When an accident occurs, the police conduct an investigation to determine the responsibilities of the individuals involved. Various methods are employed for this purpose, with one commonly used approach being the **degree of negligence (or percentage of fault)**. The degree of negligence or percentage of fault in an accident refers to the extent to which each party involved contributed to the occurrence of the accident. This determination is crucial for legal and insurance purposes, as it helps allocate responsibility and liabilities among the involved parties. The police employ various methods to assess the degree of negligence in a traffic accident. The process may include:

- **Eyewitness Statements:** Gathering statements from witnesses to understand their perspectives on the events leading to the accident.
- **Traffic Violation Analysis:** Examining whether any party violated traffic laws, such as running a red light or exceeding the speed limit.
- **Vehicle Damage Analysis:** Assessing the extent of damage to each vehicle involved, which can provide insights into the force and angles of impact.
- **Skid Marks and Road Conditions:** Analyzing skid marks on the road and considering weather and road conditions to understand how the accident unfolded.
- **Statements from Involved Parties:** Obtaining statements from the drivers and passengers involved to gather their perspectives on the incident.

Consider a scenario where Driver A fails to yield at a stop sign, colliding with the vehicle of Driver B, who has the right of way. The police investigation may find that Driver A's failure to yield constitutes negligence. If Driver B was exceeding the speed limit, contributing to the severity of the collision, both parties might be assigned a percentage of fault. However, to further refine the determination of negligence, correction factors may be applied. These factors account for additional circumstances that could affect the degree of fault, such as:

- **Pre-existing Conditions:** If either driver had a pre-existing condition affecting their ability to drive safely.
- **Mechanical Failures:** If one of the vehicles experienced a mechanical failure that contributed to the accident.
- **Road Maintenance Issues:** If the accident was influenced by poor road maintenance or inadequate signage.
- **Emergency Situations:** Whether either driver was responding to an emergency, influencing their actions.

The determination of the degree of negligence or percentage of fault in an accident is a complex process that involves a thorough investigation by the police. By considering various factors and correctional elements, authorities aim to assign responsibility fairly and accurately, contributing to the resolution of legal and insurance matters related to the accident.

1.3 Problem Statement

As previously mentioned, assessing responsibility after a crash is a complex task requiring advanced knowledge of road rules. For straightforward scenarios like crashes with traffic lights, decisions are fast and easy. However, in situations such as crashes without any traffic signs, expert knowledge is essential. Automating such tasks demands innovative and high-level approaches.

The problem addressed in this research is as follows: **How can a machine automatically assess actors' responsibilities after a vehicle collision based on a crash video?** There are two main challenges that arise when attempting to solve this problem:

- The first challenge is how to automatically detect the crash time within a crash video: Prior to responsibility assessments, the machine must determine whether the input video depicts a crash or not. Furthermore, driving recorder videos are typically lengthy, making it crucial to establish a reference point within the crash video for an effective responsibility assessment. In this context, the crash time refers to the reference point.

- After determining the crash time, the second challenge is how to automatically evaluate actors' responsibilities.

To answer these questions, this study went through multiple iterations and approaches, requiring thinking outside the box to find a reliable, performant, and fast-response solution.

1.4 Purpose of this Research

The objective of this work is to automate the responsibility evaluation process for the police, claims adjusters, and victims themselves by developing a support system that utilizes driving recorders' videos to swiftly and automatically determine fault in the event of a vehicle collision. This automation aims to simplify and expedite the process, thereby reducing decision times for the police and insurance companies. For victims, it provides immediate clarity regarding their responsibility in the collision and minimizes the waiting period for compensation from insurance companies.

1.5 Structure of this Paper

The rest of this paper is organized as follows. Related work is explained in Chapter 2. The first approach to solve automatic responsibility prediction is presented in Chapter 3, the second approach in Chapter 4, and the improvement on the second approach in Chapter 5. Evaluation and results are reported and discussed in Chapter 6, and the limitations of the system as well as encountered difficulties discussed in Chapter 7. Conclusion and future work are finally stated in Chapter 8.

Chapter 2

Related Work

2.1 Basic Technologies

2.1.1 Car Accident Report

A car accident report is a comprehensive document used to record details of a road incident involving vehicles. Within this report, a crucial aspect is the assessment of drivers' responsibilities. This assessment is essential for legal documentation, insurance claims, and determining the appropriate party at fault. The report includes the following components related to drivers' responsibility:

- **Date and Time:** Precise details of when the accident occurred.
- **Location:** Exact location, including street names and landmarks.
- **Driver Details:** Names, addresses, contact numbers, and driver's license details of individuals involved.
- **Vehicle Identification:** Maker, model, year, color, and license plate numbers of vehicles.
- **Collision Description:** A narrative detailing how the accident occurred, with a focus on actions taken by each driver.
- **Traffic Violations:** Notation of any observed or reported traffic violations by involved drivers.
- **Drivers' Responsibility Assessment:** Detailed evaluation of each driver's actions leading to the accident.
- **Witness Information:** Statements and contact details of witnesses regarding the drivers' actions.
- **Reporting Officer:** Details of the responding police officer, including observations and assessment of drivers' behavior.
- **Injuries and Damages:** Assessment of injuries sustained and damages incurred, contributing to responsibility determination.
- **Diagram and Photographs:** Visual aids, such as diagrams and photographs, illustrating the accident scene and supporting responsibility assessment.

2.1.2 Deep Learning

Deep learning is a subfield of machine learning that focuses on the development and application of artificial neural networks, particularly deep neural networks. It involves training complex models with multiple layers (deep architectures) to learn hierarchical representations of data, enabling the automatic extraction of features and patterns. Key concepts in deep learning include:

- **Neural Networks:** The fundamental building blocks of deep learning, neural networks consist of interconnected layers of nodes (neurons) that transform input data into meaningful output.

- **Deep Neural Networks:** Models with multiple layers (deep architectures) that enable the learning of hierarchical representations. Common architectures include feedforward neural networks and convolutional neural networks (CNNs).

- **Backpropagation:** The training process in which the model adjusts its weights based on the difference between predicted and actual outputs, minimizing the error.

- **Activation Functions:** Non-linear functions applied to the output of neurons, introducing non-linearity into the model and allowing it to capture complex patterns.

- **Loss Functions:** Objective functions that measure the difference between predicted and actual outputs, guiding the training process.

Deep learning architectures are:

- **Convolutional Neural Networks (CNNs):** Specialized for processing grid-like data, such as images. CNNs excel in tasks like image classification and object detection.

- **Recurrent Neural Networks (RNNs):** Designed to handle sequential data, RNNs have connections that form directed cycles. They are suitable for tasks like natural language processing and time-series prediction.

- **Generative Adversarial Networks (GANs):** Comprising a generator and a discriminator, GANs are used for generating new data instances. They find applications in image synthesis and data generation.

Deep learning has various applications among which the are the following:

- **Computer Vision:** Deep learning powers image and video analysis, enabling tasks like image recognition, object detection, and facial recognition.
- **Natural Language Processing (NLP):** Deep learning models excel in language-related tasks, including sentiment analysis, language translation, and chatbot development.
- **Speech Recognition:** Deep learning algorithms are used for accurate speech-to-text conversion, enabling applications like virtual assistants.
- **Healthcare:** Deep learning contributes to medical image analysis, disease diagnosis, and drug discovery.

Deep learning has revolutionized machine learning by enabling the development of sophisticated models capable of learning intricate representations from data. Its applications span various domains, and ongoing research continues to advance the field, addressing challenges and unlocking new possibilities.

2.1.3 Object Detection

Object detection is a computer vision task that involves identifying and locating objects within an image or video. Unlike image classification, which assigns a label to an entire image, object detection goes further by outlining the precise location of each object and associating it with a corresponding class label. Key components of object detection are:

- **Bounding Box Generation:** Object detection typically involves drawing bounding boxes around detected objects, specifying their spatial extent in the image.
- **Class Labeling:** Each detected object is assigned a class label, indicating the type of object it represents (e.g., person, car, dog).

- **Feature Extraction:** Convolutional Neural Networks (CNNs) are commonly used for extracting features from images, allowing the system to learn discriminative features for object recognition.

- **Non-Maximum Suppression:** To refine the results and avoid duplicate detections, non-maximum suppression is often applied, keeping only the most confident bounding box for each object.

There are two object detection approaches:

- **Two-Stage Detectors:** These detectors first propose potential regions containing objects (region proposals) and then classify and refine these proposals. Examples include Faster R-CNN (Region-based Convolutional Neural Network) and R-FCN (Region-based Fully Convolutional Networks).

- **One-Stage Detectors:** These detectors directly predict bounding boxes and class probabilities for each region of the image in a single pass. Examples include YOLO (You Only Look Once) and SSD (Single Shot Multibox Detector).

Some applications of object detection are:

- **Autonomous Vehicles:** Object detection is crucial for identifying pedestrians, other vehicles, and obstacles in the environment.

- **Surveillance and Security:** In video surveillance, object detection helps identify and track people, objects, or unusual activities.

- **Medical Imaging:** Object detection assists in locating and analyzing specific structures or abnormalities in medical images.

- **Retail and Inventory Management:** Object detection is used for tracking product movements and managing inventory in retail environments.

Object detection plays a vital role in various computer vision applications, enabling machines to perceive and interact with their visual environment. As technology advances, the accuracy and efficiency of object detection algorithms continue to improve, opening up new possibilities for innovative applications.

2.1.4 Open Data

Open Data refers to data that is made available to the public without restrictions on its use, distribution, or modification. The philosophy behind Open Data is rooted in transparency, collaboration, and the belief that shared information can lead to social, economic, and technological advancements. Key characteristics of Open Data are:

- **Accessibility:** Open Data should be easily accessible to the public, fostering inclusivity and promoting equal opportunities for information access.
- **Reuse:** Users have the right to reuse the data for various purposes, including commercial and non-commercial activities.
- **Redistribution:** Open Data can be freely shared and distributed, enabling widespread dissemination of knowledge.
- **Formats:** Data should be available in machine-readable formats, enhancing its usability and interoperability across different platforms and applications.

The benefits of Open Data are multiple including:

- **Transparency:** Open Data promotes transparency in government operations, corporate activities, and various sectors of society.
- **Innovation:** Access to diverse datasets encourages innovation by enabling the development of new applications and solutions.
- **Collaboration:** Open Data fosters collaboration among individuals, organizations, and communities, leading to shared insights and collective problem-solving.

OpenStreetMap (OSM) is a prominent example of an Open Data initiative that focuses on mapping and geospatial data. OSM allows users to view, edit, and use map data collaboratively. Key aspects of OSM include:

- **User-Generated Mapping:** OpenStreetMap relies on contributions from a global community of mappers who voluntarily add, edit, and verify geographic information. This approach results in a constantly evolving and detailed map of the world.

- **Versatility of Data:** OSM provides a wide range of geospatial data, including information about roads, buildings, points of interest, and natural features. This versatility makes it a valuable resource for various applications, from navigation to urban planning.
- **Open Data Principles:** OSM follows the principles of Open Data, allowing users to freely use, modify, and distribute the map data. This openness has led to the integration of OSM data into countless projects and applications.
- **Community Collaboration:** The success of OpenStreetMap is attributed to the collaborative efforts of its community. Mappers, developers, and users work together to improve the accuracy and completeness of the map.

2.1.5 Knowledge System

Knowledge Systems, also known as Knowledge-Based Systems (KBS), are a type of artificial intelligence system designed to represent, store, and apply knowledge to solve complex problems. These systems leverage explicit knowledge, often in the form of rules and facts, to emulate human reasoning and decision-making processes. The key components of knowledge systems are :

- **Knowledge Base:** The central repository that stores domain-specific knowledge, including facts, rules, and heuristics. This knowledge is used by the system to draw inferences and make decisions.
- **Inference Engine:** The component responsible for reasoning and drawing conclusions based on the information stored in the knowledge base. It applies logical rules and heuristics to solve problems or answer queries.
- **User Interface:** The interface through which users interact with the Knowledge System. This can be a graphical user interface (GUI), a command-line interface, or integration with other applications.
- **Explanation Facility:** Knowledge Systems often include mechanisms to explain their reasoning processes and the logic behind specific decisions. This enhances transparency and user understanding.

- **Knowledge Acquisition System:** The process of capturing and inputting knowledge into the system. This may involve interviews with experts, documentation review, or automated methods for extracting information.

There are different types of Knowledge Systems:

- **Expert Systems:** Designed to emulate the decision-making ability of a human expert in a specific domain. Expert Systems use a knowledge base of rules and facts to provide expert-level advice.

- **Decision Support Systems (DSS):** Assist individuals or groups in making decisions by providing relevant information and analysis. DSS often integrate data-driven and knowledge-driven components.

- **Knowledge Management Systems:** Focus on capturing, organizing, and sharing organizational knowledge. These systems facilitate knowledge creation, storage, retrieval, and collaboration among employees.

Applications of Knowledge Systems are multiple:

- **Medical Diagnosis:** Expert Systems in healthcare assist in diagnosing medical conditions based on patient symptoms and historical data.

- **Financial Decision-Making:** Decision Support Systems help in financial analysis, risk assessment, and investment decision-making.

- **Troubleshooting and Maintenance:** Expert Systems are used to troubleshoot technical issues and provide maintenance recommendations in various industries.

- **Natural Language Processing:** Knowledge Systems play a crucial role in natural language understanding and generation, enabling applications like chatbots and language translation.

Knowledge Systems represent a significant advancement in artificial intelligence, contributing to problem-solving, decision-making, and knowledge management across diverse domains. The ongoing development of these systems holds promise for addressing increasingly complex challenges in various fields.

2.1.6 Rule-based Knowledge System

A rule-based knowledge system, also known as an expert system, is a type of artificial intelligence system designed to emulate human decision-making processes. It utilizes a knowledge base consisting of explicitly defined rules to make inferences, solve problems, and provide recommendations within a specific domain. Key components of a rule-based knowledge system are:

- **Knowledge Base:** The central repository that stores domain-specific knowledge in the form of rules, expressed as ”if-then” statements.
- **Inference Engine:** The reasoning mechanism responsible for applying rules to draw inferences and make decisions.
- **Rule-Based Reasoning:** The system uses rule-based reasoning to reach conclusions by matching conditions in rules with the current state of facts.
- **Forward and Backward Chaining:** Forward chaining starts with available facts, while backward chaining starts with a goal and works backward to determine necessary facts.
- **Knowledge Acquisition:** The process of capturing and inputting knowledge into the system, often facilitated by knowledge engineers working with domain experts.
- **Explanations:** Mechanisms for providing explanations of the system’s reasoning processes, enhancing transparency.

Rule-based knowledge systems play a vital role in emulating human decision-making processes within specific domains. Advances in artificial intelligence continue to enhance the capabilities and applications of these systems, contributing to problem-solving and decision support.

2.2 Related Work

2.2.1 Causes and Contributing Factors of Car Crashes

Numerous studies and research endeavors have been dedicated to identifying the causes and factors behind car crashes as well as exploring road accident likelihood and severity (Jianyu Wang et al. 2023; Mondal et al. 2023; Bhuiyan et al. 2022; Akin et al. 2022; F. Wang et al. 2022; Shahsavari et al. 2022; Kamalasekar et al. 2022; Tsala et al. 2021; Borucka et al. 2021; Casado-Sanz, Guirao, and Attard 2020; Cai 2020; Jima 2019; Belloumi and Ouni 2019; Rolison et al. 2018; Ditcharoen et al. 2018; Oralhan and Göktolga 2018). Jianyu Wang et al. 2023 explore risk factors influencing the at-fault party in traffic accidents and analyzes their impact on traffic accident severity. The study shows that travel mode, season, and road speed limit are more important risk factors for traffic accidents, with motor vehicle drivers as the at-fault parties. Another study (Bhuiyan et al. 2022) reveals that significant features associated with crash severities include driver characteristics (gender, license type, seat belts), vehicle characteristics (vehicle type), road characteristics (road surface type, road classification), environmental conditions (day of crash occurred, time of crash), and injury localization. From Tsala et al. 2021, it appears that, of the 382 accidents recorded during a period, six factors were identified and classified as follows: causes of accidents related to speed and carelessness, location of the accident, type of vehicle at fault, day the accident occurred, time of the accident and the age of drivers involved. A comprehensive study conducted by Ditcharoen et al. 2018 provides an overview of the various elements that influence the severity of car crashes. The paper also reviews commonly employed techniques such as logistic regression and power models utilized in previous studies. According to this research, the most frequently cited factors contributing to car crash severity are the speed at which the vehicle is traveling, followed by human characteristics. Additionally, significant factors include vehicle type, weather conditions, alcohol consumption, and driver fatigue.

2.2.2 Multi-Agent Systems

Multi-agent systems are widely recognized as a flexible and extensible system architecture for developing computer programs that address various problems. Multi-agent systems have been used as a solution to the vehicle collision avoidance control problem (C. Yuan et al. 2023; Muzahid et al. 2023; Sanogo et al. 2023), obstacle avoidance (Xiong, Z. Liu, and Y. Luo 2023), and autonomous delivery vehicles optimization problem (Ergün 2023). C. Yuan et al. 2023 provide a multi-agent coordinated control system to improve the real-time performance of intelligent vehicle active collision avoidance. The multi-agent coordinated control system can handle the conflict between the decisions of different agents according to the rules. Comparing with existing control strategies, the proposed system can realize multi decisions and planning at the same time; thus, it will reduce the operation time lag during active collision avoidance. Xiong, Z. Liu, and Y. Luo 2023 conduct research on collision and obstacle avoidance of multi-agent systems without mapping ability, while the constrained agent can only detect obstacles within a limited distance, then a velocity programming strategy is proposed considering the lack of a high-resolution map and the challenge of the modeling of complex obstacles. Kitajima et al. 2019 introduce a multi-agent traffic simulation methodology to estimate the potential improvements in road safety resulting from automated vehicle technologies. Their system applies traffic simulations to a designated area in Tsukuba city, Japan, by integrating road infrastructure data with a large number of vehicles, drivers, and pedestrians.

2.2.3 Crash Detection and Vehicle Identification

The automatic detection of car crashes through traffic monitoring cameras can significantly reduce response time, improve rescue efficiency, enhance traffic safety, and save lives. YOLO ("You Only Look Once"), a sophisticated convolutional neural network (CNN) for real-time object detection, has emerged as a powerful tool for achieving such detection. It is a widely adopted system for object detection in videos and images. Numerous studies have focused on detecting car crashes in real-time video feeds from traffic monitoring cameras utilizing YOLO (Mane et al. 2023; Adewopo et al. 2023; Pawar and Attar 2022; Lee et al. 2021; Naik et al. 2021; Hsu, Huang, and Han 2020; Tian et al. 2019; Gour and Kanskar 2019; Machaca Arceda and Laura Riveros 2018) or using other techniques (Ghahremannezhad, H. Shi, and C. Liu 2022; Hozhabr Pour et al. 2022; J. G. Choi et al. 2021; Boukerche and Hou 2021; Radu et al. 2021; C. Wang et al. 2020; Yao et al. 2019; S. Sharma and Sebastian 2019; Sultani, C. Chen, and Shah 2018; H. Sharma, Reddy, and Karthik 2016). Mane et al. 2023 propose an ensemble model that uses the YOLOv8 approach for efficient and precise event detection. The model framework's robustness is evaluated using YouTube video sequences with various lighting circumstances. Pawar and Attar 2022 propose a deep learning approach for automatic detection and localization of road accidents by formulating the problem as anomaly detection. The method follows one-class classification approach and applies spatio-temporal autoencoder and sequence-to-sequence long short-term memory autoencoder for modeling spatial and temporal representations in the video. Some other works have focused on vehicle identification and detection (Kutlimuratov et al. 2023; Sindhu 2021) and classification of traffic incidents (Basheer Ahmed et al. 2023). Kutlimuratov et al. 2023 propose a system that can identify and track objects inside the region of interest and count detected vehicles using a YOLOv5 model for vehicle identification. Basheer Ahmed et al. 2023 propose a real-time traffic incident detection and alert system that is based on computer vision. The proposed framework consists of three models, each of which is integrated within a prototype interface to fully visualize the system' s overall architecture. This study employed an innovative parallel computing technique for reducing the overall complexity and inference time of the AI-based system to run the proposed system in a concurrent and parallel manner.

2.2.4 Crash Risk Prediction and Anticipation

Some works have predominantly focused on crash risk prediction (Hu et al. 2023; Banerjee et al. 2022; Z. Luo et al. 2021; P. Li, Abdel-Aty, and J. Yuan 2020; Fawcett et al. 2017; Kumar and Toshniwal 2016; Park, Kim, and Ha 2016; L. Lin, Q. Wang, and Sadek 2015; Q. Shi and Abdel-Aty 2015), or crash anticipation (Bao, Yu, and Kong 2020; Suzuki et al. 2018; Chan et al. 2017; You, Junhua Wang, and Guo 2017; Y. Gu, Qian, and F. Chen 2016; D' Andrea et al. 2015). For instance, Hu et al. 2023 presents a novel method to predict crash risk proactively by combining these interactive factors: drivers' attention and environmental complexity. More than 200 high-risk zones and 300 noncrash zones were screened out through social media data. Corresponding environmental information was collected using the street view map. Spectral saliency mapping was applied to depict the driver' s attention distribution toward images. A featured vector was then constructed by fuzing the visual attention model and image semantics. The gradient boosting decision tree algorithm was applied to analyze the relationship between the multitype crash data and featured vectors.

2.2.5 Assessing Responsibility in Car Crashes

Human factors tend to rank highest in terms of being the main causes of crashes. Challenges arise when attempting to determine human responsibility during crash investigations and accurately assigning fault percentages to each party involved. Dirnbach et al. 2020 propose a new technical and analytical approach for handling expert reports on car crashes at intersections, specifically focusing on traffic light scenarios. In this work, a simulation program application is utilized to conduct a precise analysis of car crashes. Another approach involves employing a possibility theory-based classifier, specifically a possibility rule-based classifier that employs function approximation, to capture the uncertainty inherent in expert knowledge due to incompleteness. This approach infers a model from the 100-Car naturalistic driving dataset, showcasing the inherent uncertainty involved in making decisions based on expert evaluations.

To date, there have been limited studies, such as Sanjurjo-de-No et al. 2021; Dirnbach et al. 2020; Garcia et al. 2019; Chandraratna and Stamatiadis 2009, focusing on evaluating responsibility after vehicle collisions. In their work (Sanjurjo-de-No et al. 2021), the visual clustering technique of self-organizing maps (SOM) has been applied to better understand the multivariate structure in the data, to find out the most important variables for driver liability, analyzing their influence, and to identify relevant liability patterns. Garcia et al. 2019 estimate responsibility through a data-driven process with explicit rules. They compare various statistical learning methods (e.g., logistic regression with L1 penalty, random forests, and boosting) using cross-validation to provide responsibility attributions made by experts (considered as the gold standard) based on data routinely recorded by the police.

2.2.6 Positioning of this research

This study addresses the limitations of existing methods by expanding the scope of accident management beyond simple collision detection or prediction and responsibility attributions made by experts. It introduces a system that is capable of detecting vehicle collisions within crash videos and implements an original responsibility assessment process to assess drivers' responsibilities. This system aims to provide accurate and timely evaluation of collision incidents, facilitating fair responsibility attribution. It employs object detection for collision detection along with an original algorithm and process that associate a knowledge rule-based system and open data for responsibility assessment. The entire responsibility assessment process involves four steps: (1) detecting the crash time within a crash video, (2) identifying all traffic lights within the video, (3) obtaining road information from the Open-

StreetMap API, such as road width and the presence of other traffic signs if necessary, analyzing and processing the information, and (4) utilizing a rule-based knowledge system of road rules, vehicle speed, and orientation to deduce the probable responsibility of each party involved. The system focuses on head-on and angle crashes involving two cars and facilitates the seamless sharing of evaluation results with the police and insurance companies within minutes of a collision. The decision to focus on head-on and angle collisions is influenced by the technical specifications of driving recorders, primarily designed to capture the frontal view of vehicles. However, the usability and applicability in real-world situations of such a system supporting only head-on and angle collisions is broad, since according to a study [1], these two collision types collectively account for more than 70% of all car collisions.

By employing advanced image processing techniques, the system enables prompt detection and analysis of collision incidents. The integration of open data enhances the contextual understanding of the road environment, contributing to more accurate responsibility assessments by improving the performance of responsibilities evaluation mainly during nighttime with traffic lights. The first prototype of the system supported only head-on and angle crash scenarios with traffic lights within two weather conditions groups (bad weather/good weather). The second prototype was improved to support three types of head-on/angle crash scenarios without traffic lights (priority roads/one-way roads/roads with the same width) within three distinct weather conditions (sunny/cloudy/rainy), in addition to those with traffic lights. In its current version, the system now accommodates six different types of head-on/angle crash scenarios without traffic lights (priority roads/one-way roads/roads with the same width/roads with stop signs/roads with speed limit signs/roads with flashing red or yellow signals) within six harsh weather conditions (sunny/cloudy/rainy/stormy/snowy/foggy). The support of these additional scenarios and weather demanded retraining crash detection and traffic light detection models, adding more rules to the knowledge-based system, and enhancing the algorithm and process of responsibility assessment. Additionally, extensive experiments are conducted with results showing that the system performs better than its previous versions, mainly during nighttime without traffic lights (up to 93% accuracy against 82.5% obtained previously). The significant difference and advantage of this system over existing ones is its automation of responsibili-

[1] https://www.iii.org/table-archive/21904

ties evaluation for the police, claims adjusters, and victims themselves as well as its applicability for autonomous vehicles. Moreover, through case studies and comparisons with existing research, the effectiveness and superiority of this system are demonstrated. In addition, the detection models, the image dataset as well as the video dataset that were used to implement the system will be made publicly available to the scientific community. These resources can be used in the future by other researchers to implement more advanced systems in related fields. This study is among the first to enable machines to automatically assess the responsibility of drivers within a crash. It can serve as one of the precursors and foundations for automatic responsibility assessments in autonomous vehicles.

Chapter 3

Object Detection Based Responsibility Assessment System Using Videos from a Real Driving Recorder

3.1 Introduction

This chapter introduces a system based mainly on image processing that can support the police in evaluating actors' responsibilities automatically within a crossroad crash (one of the most common car crashes). The system uses the crash video recorded by the driving recorder of one of the vehicles involved in the crash as the input data source. It then assesses and outputs the evaluation of each actor's responsibility within the crash thanks to a rule-based knowledge system, which was introduced to make the reasoning about responsibility assessment explainable and enable users to understand the results easily. To assess responsibilities, the system is equipped in total with three different modules and goes through three different steps: (1) detecting crash time within the crash video thanks to the first module, (2) detecting all traffic signs within the crash video thanks to the second module, and (3) using a rule-based knowledge system of road rules to deduct each party's probable responsibility thanks to the third module. A head-on/angle crash is not an object on its own that a common vehicle detection model can recognize. Therefore, applying existing vehicle detection models such as (Z. Chen et al. 2022; Dong, Yan, and Duan 2022; Song and W. Gu 2021) will fail or output wrong results with many false positives. To solve the issue, we made our proposed model assume that if there is ahead-on/angle crash in an image with the angle of view of the driving recorder inside one of the vehicles involved in the crash, the crash can be recognized by taking into consideration only the collided vehicle, its shape, and its position within the video.

During the evaluation, the performance of each module of the system was tested with different parameters and under different road conditions (daytime and nighttime with good and bad visibility). The experiment's results show how the system performs in (1) detecting the crash time within a video using different vehicle types (cars, vans, and trucks), (2) detecting traffic signs within a crash video using different view distances (far, close, very close), and (3) assessing each party's responsibility.

3.2 Responsibilities' Evaluation within a Crossroad Crash with Traffic Lights

Using previously occurred car crash data with artificial intelligence, Internet of Things (S. Sharma and Sebastian 2019; Z. Yuan, Zhou, and Yang 2018), or machine learning (Theofilatos 2017) to predict future crashes or to detect factors leading to those fatalities is very important.

In this section, a heuristic approach was proposed to solve the problem of actors' responsibility determination when a crash occurs thanks to the usage of the driving recorder video of the crash as the data source. A system that helps evaluate each actor's responsibility within a car crash, especially a crossroad crash with traffic lights was implemented. The system consists of three different modules with a set of five distinct steps.

In an accident, there are many factors that should be taken into consideration. Some of these factors can be detected automatically, while some will need human input in the system. Here are some of the manual inputs, and some of the automatically detected factors, as well as some assumptions we made during the implementation of the system:

- Automatically detected factors: Crash time; Traffic light state; Daytime/Nighttime.
- Manual input factors: Road width; Vehicles' speed; Vehicles' direction; Accident location; Drivers' information (name, age...); Exact time (hh:mm:ss).
- Assumptions: Both actors involved in the crash are cars; Vehicle A is the car from which the crash video was recorded; Both actors come from different ways.

It is also important to note that the current version of the system is only for crossroad crashes with traffic lights that involve two cars and taking as input only one crash video at a time.

3.2.1 Design of the System

To implement a system that can evaluate actors' responsibilities within a crash using the crash video as the input data source, there is a need to implement distinct modules that can work together to achieve the final goal. The system is a set of three different modules, each having a specific job and working independently to achieve one sub-goal of the entire system. The first module is a module that can detect the crash time in a video and split the crash interval in that video into images for object detection. The second module is a module that can detect, in an image, crucial information used to evaluate responsibilities in a crash (such as traffic lights). The third and last module is a module based on a rule-based knowledge system of road rules that can use an inference engine to derive actors' responsibilities within a crash. Figure 5.1 shows the architecture of the system with the three modules.

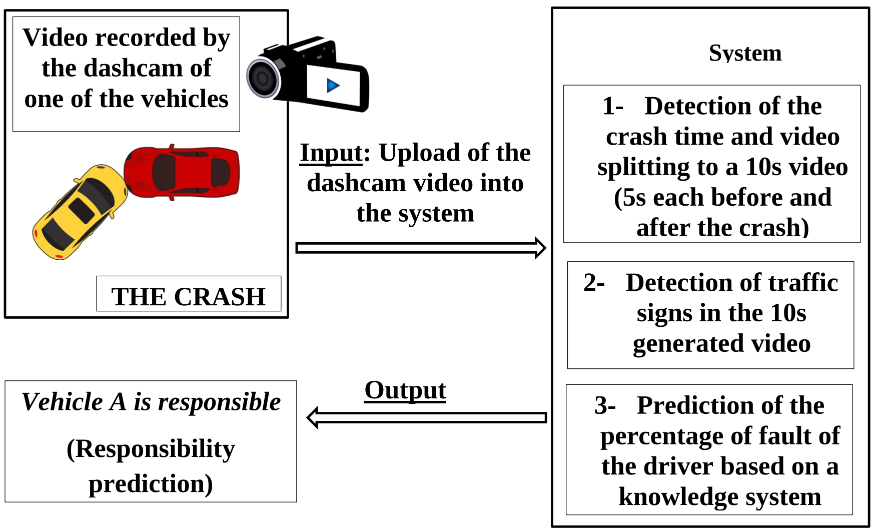

Figure 3.1: The architecture of the system with the collaboration between the three modules.

All of these three modules contribute to having a complete system that passes through five distinct steps to evaluate the responsibilities of actors within a crash: (1) The detection of the crash time in the video recorded by the driving recorder of the vehicle; (2) the split of that video into a 10 s video (5 s each, before and after the crash time); (3) the split of the 10 s video, frame by frame, into images that can be used by a model for object detection; (4) the detection of important objects in the video, such as traffic lights and cars; (5) the evaluation of the percentage of fault (degree of negligence) of each actor based on road rules using a rule-based knowledge system.

We implemented an easy-to-use and ergonomic user interface that allows the user to upload his crash video and get the result of the responsibilities' evaluation in three steps. Figure 3.2 shows a screenshot of the interface with the steps that the user must follow to get the final result of the evaluation.

Figure 3.2: Screenshot of the graphic interface of the system with the steps the user has to follow to get the final result of the responsibilities' evaluation with the list of the top-20 results displayed on the right side as thumbnails within a scrollable panel and the selected best result displayed on the left side.

3.2.2 Crash Time Detection Module

This module is the first one and the starting point of the system. It detects the time of the crash in the crash video, splits that video into a 10 s video (5 s each before and after the crash), and splits that 10 s video, frame by frame, into images. It consists of three main tasks: the detection of the crash time in the crash video, the split of the crash video into a 10s video, and the split of the 10 s video into images.

Detection of the Crash Time in the Crash Video

To detect the crash time in a video, we created a custom object detection model that can detect a crash using YOLOv5 (`https://doi.org/10.5281/zenodo.7002879` (accessed on 8 September 2022)). YOLOv5 is a family of object detection architectures and models pretrained on the COCO dataset (T.-Y. Lin et al. 2014). It is one of the fastest versions in the YOLO series. YOLO (Redmon et al. 2015) an acronym for 'You only look once', is a convolutional neural network for performing object detection in real time. It is an object detection algorithm that divides images into a grid system. Each cell in the grid is responsible for detecting objects within itself. YOLO is one of the most famous object detection algorithms due to its speed and accuracy. Many works related to car traffic have used YOLOv5 to detect traffic signals (Snegireva and Perkova 2021; W. Liu et al. 2021), obstacles (Murthy et al. 2022), traffic flow (Sun, Zhuoshen Li, and Zhuolin Li 2021), or to classify vehicles (Snegireva and Kataev 2021).

In this study, YOLOv5 was used to detect a crash within a video. We used 1530 images of head-on and angle crashes to train our YOLOv5 model and get it ready for crash detection.

Dataset Building

Our dataset has a particularity because of some criteria it has to meet:

- The images in the dataset have to be related to a head-on/angle crash;
- The dataset's images have to be from a video recorded by the driving recorder of one of the vehicles involved in the crash.

To build a custom dataset with enough images that meet these criteria, we had to extract these images from videos. We used YouTube as a source of data. YouTube is a well-known online platform where we can find many kinds of crash videos, including the ones of interest to us. However, it requires a manual and long job to get the ones that meet our dataset criteria. We started by searching compilation videos of crossroad accidents recorded by driving recorders that we could collect manually. The major part of the compilation videos we found was not dedicated only to head-on/angle crashes in the context of crossroad accidents. Most of them were a compilation of all kinds of car crashes that occurred in the United States of America, Thailand, Russia, and India with the driving recorders, not necessarily the ones of the vehicles involved in the crash. Therefore, we had to watch all the videos and select the parts containing the kinds of crashes and angles of recording of interest to us. In total, we watched 103 videos (each one having an average of 15 min in length) and finally selected 68 of them. We selected those compilation videos based on acceptable image quality (480 p and above), the fact that they have at least one video related to a head-on/angle crash in a crossroad accident context, and the fact that the crash details are clear enough to use the images during the labeling step. After collecting the compilation videos, we had to extract and resize the frames (images) of the parts that we were interested in. We used the library OpenCV (Open Source Computer Vision) to extract the frames and scikit-image (`https://scikit-image.org/` (accessed on 11 May 2022)) to modify and resize them so that they have the same size and structure. For homogeneity and consistency in the data, the frames of the videos were converted to a lower width (500 pixels). Figure 3.3 shows an example of one of the head-on crash images we finally got after the extraction and the resizing steps.

After all these steps, we got a total of 1530 crash images from different countries, such as the United States of America, Russia, and India, to train our model. To annotate the images, we used LabelImg (`https://github.com/tzutalin/labelImg` (accessed on 11 May 2022)) a graphical image annotation tool, and label object bounding boxes in images written in Python. We imported all of our images into the software and manually annotated them one by one, by creating rectangle bounding boxes around the crash. A crash is not an object on its own that an object detector can recognize. Therefore, we assumed that if there is a head-on/angle crash in an image with the angle of view of a driving recorder inside one of the vehicles involved in the crash, we can recognize the crash by taking into consideration the collided car. The shape and the position of the collided car can tell us if there is a crash or not. Therefore, we created annotation bounding boxes to

Figure 3.3: Example of one of the head-on crash images obtained after the extraction and the resizing phase.

surround collided objects (mainly cars, trucks, and vans) in the images. LabelImg allows for the creation of two types of annotation formats: PascalVOC (XML file-based format) and YOLO (TXT file-based format). We chose the YOLO format and exported our labeled images (with their corresponding annotation TXT files) to a directory ready to be split into training/validation/test data. We separated our annotated images into training data (1071 images equivalent to 70% of the dataset), validation data (306 images equivalent to 20% of the dataset), and test data (153 images equivalent to 10% of the dataset).

Model Training and Validation

After setting up the dataset, we then created the necessary configuration files and trained our custom detection model. To train the model, we used Google Colab (a Jupyter notebook environment created by Google to help disseminate machine learning education and research) because of its ease of use and its fast processing. We set the batch size to 16 and the number of training epochs to 100. After the training and validation steps, we exported the trained weights that can be used for future inferences for crash detection in any other videos and on devices that do not have a powerful processing unit.

In the early version of our system (Yawovi, Tadachika Ozono, and Shintani 2020), we implemented a service that can be used by the system to request the detection of a crash in a given video using the model inference. The service output is the list of the top-20 crash detection results with details about each detection's accuracy, the frame in which each detection was made, and the link to the image containing the bounding box for the corresponding detection. To be able to create such a service, we had to modify the detection algorithm of YOLOv5 to output the data we needed. The YOLOv5 detection algorithm outputs the result of the detection directly into videos or images. However, in our case, we needed to get the detection details in an array. Therefore, we created our custom detection algorithm based on the initial detection code of YOLOv5. We added features and methods to extract details about the detection's accuracy, the frame number in the image sequence of the video in which the detection was made, and the exportation of the image in which the crash was detected with the detection bounding box. Thanks to that service, the system was able to show the list of the top 20 results of the crash time detection to the user, who can manually select the best result to use for the next steps of the responsibilities evaluation.

In the current version of the system, to speed up the whole process and make things easier for users, we removed that manual selection step. Thanks to additional training in the crash time detection model, now the system automatically uses the best first result and processes to the next steps without asking the user for manual selection.

Split the Crash Video into a 10 s Video with Image Extraction

Generally, driving recorders record long sequences of videos. Depending on the brand and the available memory of the driving recorder, the raw recorded video can have an initial length of 30 min or more. When a crash occurs, we do not need the full-length recorded video. We only need a few seconds before and after the crash. Therefore, after the detection of the crash time within a video, the system splits that video automatically into a 10 s video (5 s each before and after the crash time).

After getting the 10 s video (which reduces processing time considerably), the module automatically extracts the frames from that short video to have a total of 240 images (1 s in a video may have 24 frames or images). Then, it modifies and resizes each of the frames to get frames with the same size and structure and makes all of them ready to be used by any object detection library. The output of this module is images of the environment of the crash a few seconds before and after the crash. This output will be used by the second module to detect traffic lights and other crucial information used to evaluate responsibilities within crashes.

3.2.3 Traffic Sign Detection Module

This module is the traffic sign detection module that helps the system know what to evaluate in the responsibilities' evaluation phase. The final purpose of this module is to detect any kind of traffic signs (stop signs, traffic lights, speed limit signs…), but in its current version it can only detect traffic lights. It is an intermediary module and serves as a middleware for the first module and the last module. To implement this module, whose main task is object detection, we used again the object detector YOLOv5 and the library Open CV. We trained a YOLO custom object detection model with a dataset of thousands of images of green, red, and yellow traffic lights.

Dataset Building

To build a custom dataset with enough images of traffic lights, we investigated and downloaded them one by one on Google Image Search. In addition, fortunately, we found a ready-to-use labeled traffic light image dataset on the public dataset of `Roboflow.com` (`https://public.roboflow.com/object-detection/self-driving-car` (accessed on 11 May 2022)). The dataset is a set of 15,000 images taken on roads in the United States of America. It contains not only signs for traffic lights, but also signs for pedestrians, bikers, and cars. Needing only traffic lights annotations, we extracted 1300 images of traffic lights from the original dataset (red, green, yellow, red left, green left, and yellow left). We finally got a total of 3000 images of traffic lights after adding the ones we downloaded manually from Google.

For labeling the dataset's images, we used LabelImg, to annotate them. We then separated the annotated images into training data (2100 images equivalent to 70% of the dataset), validation data (600 images equivalent to 20% of the dataset), and test data (300 images equivalent to 10% of the dataset).

Model Training and Validation

After the dataset building phase, we created the necessary configuration files and trained the model in Google Colab as we did previously for crash time detection module.

The output of this module is the list of all traffic lights detected in the crash environment's images a few seconds before and after the crash. This output is used by the third module to evaluate actors' responsibilities through an inference on a rule-based knowledge system.

3.2.4 Responsibility Evaluation Module

This module is the last module of the system. It uses the results of the previous module (detected traffic signs) to evaluate the responsibilities. To enable users to easily understand the reasons for the results, this module was implemented using a rule-based knowledge system and uses an inference engine to determine whether the first driver in the crash (the driver of the vehicle whose crash video was recorded by the driving recorder and inputted into the system via the first module) is responsible or not.

Usually, after a crash, the police determine responsibilities based on the evaluation of each actor's degree of negligence (or percentage of fault). In Japan, legal books on negligence offset rates in civil traffic proceedings are widely used to determine the degree of negligence. In these books, there are predefined degrees of negligence with different kinds of crashes. Since our system is based on Japanese traffic rules, we use those predefined degrees of negligence to obtain responsibilities (like the police generally do). For example, let's consider a crossroad crash involving two vehicles: Vehicle A (let X_A) and Vehicle B (let X_B) at an intersection (let P) with traffic lights (let T) as illustrated in Figure 3.4.

In such a case, according to Japanese traffic rules, we have basic degrees of negligence (let N) for each vehicle depending on the situation (Let the predicates $is\_intersection$ to check if a crash spot is an intersection or not, is_green to check if a traffic light is green or not, is_yellow to check if a traffic light is yellow or not, and is_red to check if a traffic light is red or not.):

> Situation 1: "In case the traffic light is green for Vehicle A and red for Vehicle B, the degree of negligence for Vehicle A is 0, and the one of Vehicle B is 100."

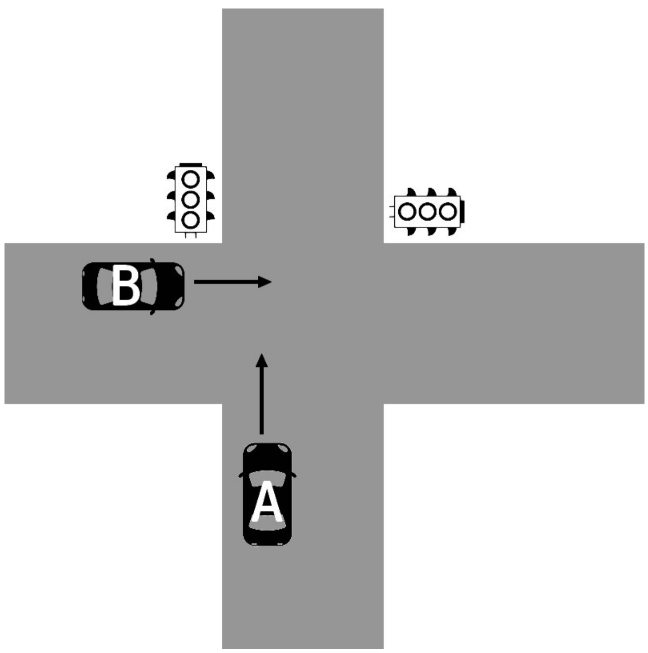

Figure 3.4: Illustration of a crash involving two vehicles at an intersection with traffic lights in Japan (left side driving).

In first-order logic, this can be expressed as:

$$(is\_intersection(P) \land is\_green(X_A, T) \land is\_red(X_B, T)) \\ \implies (N(X_A) = 0 \land N(X_B) = 100)$$

Situation 2: "In case the traffic light is yellow for Vehicle A and red for Vehicle B, the degree of negligence for Vehicle A is 20, and the one of Vehicle B is 80."

In first-order logic, this can be expressed as:

$$(is\_intersection(P) \land is\_yellow(X_A, T) \land is\_red(X_B, T)) \\ \implies (N(X_A) = 20 \land N(X_B) = 80)$$

Situation 3: "In case the traffic light is red for Vehicle A and red for Vehicle B, the degree of negligence for Vehicle A is 50, and the one of Vehicle B is 50."

In first-order logic, this can be expressed as:

$$(is\_intersection(P) \land is\_red(X_A, T) \land is\_red(X_B)) \\ \implies (N(X_A) = 50 \land N(X_B) = 50)$$

The responsibility evaluation module of the system uses this logic to deduct the corresponding degree of negligence of each actor involved in a crash and to output responsibilities. In some cases (such as low accuracy in detections, no result after multiple deductions from the knowledge system, or if the crash did not occur at a crossroad), the module outputs "unknown," which is set for unknown results. This is the output in case of failure in the responsibility prediction by the system.

3.3 Future Work

As mentioned in previous sections, our current system is specific to crossroad crashes with traffic lights only. To have more impact and contribution to solving problems related to responsibility assessment within vehicle crashes, we are currently working on improving the system to handle more crossroad cases. In addition, because not all law systems have clear rules for all car crash situations, the system (mainly the third module) will need some modifications when used with some law systems.

3.3.1 Other Crossroad Crash Cases to Handle

There are a wide variety of car crashes that occur every day with different scenarios, in different environmental conditions, and in different areas. Therefore, our system cannot handle all types of car crashes. We chose to focus only on crossroad crashes because of the fact that they have a variety of cases and they are the most recorded ones with driving recorders

Here is the list of other crossroad crash cases that we plan to implement to improve the system:

- Crossroad crash involving a car and a pedestrian: The handling of this case at/not at a pedestrian crossing involved taking into consideration different scenarios such as when there is a traffic light, when there is a safe zone, when there is no traffic light, when there is a traffic light near the pedestrian crossing, and when the collision occurs in front of the pedestrian crossing (as illustrated in Figure 3.5).

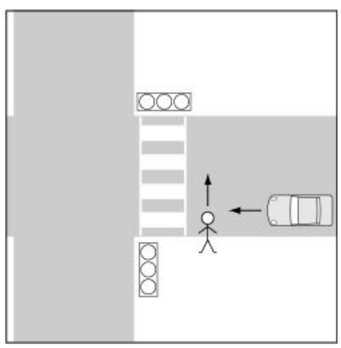

Figure 3.5: Illustration of a crash involving a car and a pedestrian at an intersection in front of the pedestrian crossing. From https://jiko-online.com/wp/kasitu/jiho-oudan/ (accessed on 8 September 2022).

- Crossroad crash involving two cars: This involved taking into consideration different scenarios such as intersection with a traffic light, intersection without traffic lights when there is a one-way violation when there is a stop sign or a red blinking/yellow blinking signal, and when there is a priority road;

- Crossroad crash involving a car and a motorcycle: This involved taking into consideration scenarios such as an intersection with a traffic light (as illustrated in Figure 3.6), an intersection of the same width without traffic lights, the motorcyclist clearly driving on a wide road, the car's driver clearly driving on a wide road the motorcyclist driving on a priority road, the car's driver driving on a priority road, the car's driver violates one way, and the motorcyclist violates one way.

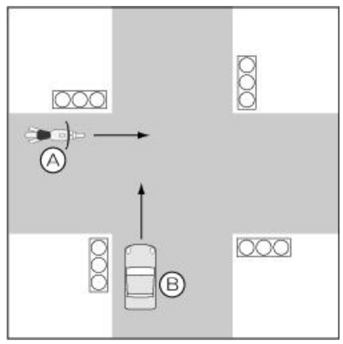

Figure 3.6: Illustration of a crash involving a car and a motorcycle at an intersection with traffic lights. From https://jiko-online.com/wp/kasitu/jiba-tyoku/ (accessed on 8 September 2022)

.

3.3.2 Data Collection

The improvement on our system is to enable it to handle more crossroad crash cases, we, therefore, need to collect data and train our models to recognize more elements in a given crash video dashcam footage. The data collection will be divided into four phases:

- **Data collection for the crash time detection model**: There are some existing car crash video datasets that were made from YouTube videos such as CCD (Bao, Yu, and Kong 2020), DAD (Chan et al. 2017), and A3D (Yao et al. 2019). These datasets contain all kinds of accidents with many scenarios. The ones of interest to us (containing crashes at intersections involving two cars /a car and a pedestrian /a car and a motorcycle with the vehicle from which the video was recorded involved) are very few. Therefore, because we can get only a few videos from these datasets, as we did before, we will look for additional crossroad crash videos on YouTube to have more data.

- **Data collection for the traffic sign detection model**: We will also collect data to train the existing traffic sign detection model to recognize not only traffic lights but also other traffic signs such as stop signs, pedestrian crosswalks, and speed limit signs.
- **Data collection for the knowledge system**: To enable the system to evaluate responsibilities within the new cases that will be implemented, we will need to collect more road rules and degree of negligence assessment logic to improve our knowledge base. A simple example may be:

  > "When the crash occurs near the pedestrian crossing with a traffic light, the degree of negligence is 70% for the pedestrian crossing in red and 30% for the car with green traffic light".

- **Data augmentation for all the collected data**.

3.3.3 Crash Simulation Sandbox Implementation

Getting video data of different crash cases with various scenarios is a complex and time-consuming task. Some crash cases' video data are easily available because they usually occur. On the other hand, some others are rare or very difficult to get because of their low level of occurrence. To have enough data in our dataset that are related to cases that our system will handle, we have to simulate some crashes for some cases that we cannot find enough data for. Therefore, we plan to add to our system a simulation sandbox that will enable a user to simulate a crash in 3D and generate a crash video that can be used to train our crash time detection model.

3.4 Conclusions

With the objective of shortening the decision time for the police and claims adjusters when a car crash occurs, we have developed a support system based on object detection and a rule-based knowledge system. The system can recognize a crash within a video, detect and list all traffic signs within that video and finally assess the responsibilities of actors within the crash. The reasoning for responsibility assessment should be explainable. Therefore, with the implementation of a rule-based knowledge system in the system, users can easily understand the responsibility assessment's reasons and confirm the results. To detect a crash within a video, we discovered that the simple application of existing vehicle detection models would result in wrong detections with many false positives. To solve the issue, we made our proposed model to take into consideration only the collided vehicle, its shape, and its position within the video. Moreover, because most of the existing sources of datasets related to car crashes or publicly available datasets are generally from surveillance cameras or from driving recorders of vehicles that are not necessarily involved in the crash, we built a custom dataset that fulfilled all the system's requirements. It was one of the most time-consuming tasks to complete the system's implementation. During the experimentation, results showed that the system performs well when the light's condition and the visibility of collided objects are good, and when traffic lights' view distances are close. That makes the system depends on the visibility of objects to perform well.

For improvement, we plan to train our crash time detection model with more data in different crossroad crash cases and train the traffic sign detection model with more traffic signs (stop signs, speed limit signs, etc.).

Chapter 4

Vehicle Collision Responsibility Evaluation System based on Object Detection and OpenStreetMap

4.1 Introduction

Nowadays, there is an increase in the usage of driving recorders (also called dashcams) by vehicle owners. There are even some insurance companies that make their usage mandatory for their customers. Videos recorded by driving recorders are valid proof of a collision and help to determine responsibilities faster. In addition, the usage of smartphones as recorders are becoming a cheap and best alternative because of their ability to do more than just recording and the fact that they can handle more complex or advanced features compared to traditional recorders.

In this chapter, we introduce a system based on object detection, knowledge system, and open data from OpenStreetMap API [1] a free, open geographic database updated and maintained by a community of volunteers via open collaboration. The system is composed of a mobile application and a server, and currently only supports head-on and angle crashes involving two cars at a crossroad.

In the previous chapter, we described one of the early versions of the system based mainly on image processing and a rule-based knowledge system. The system uses the crash video recorded by and taken from a real driving recorder of one of the vehicles involved in the crash as the input data source. It then assesses and outputs the evaluation of each actor's responsibility within the crash thanks to its rule-based knowledge system. The main limitations of this previous system are the fact that (1) it only supports head-on/angle crossroad crashes with traffic lights, (2) important details such as vehicle speed and crash location are missed and should be manually input by the user, and (3) its performance is totally dependent on the visibility of objects within the crash video. Toward realizing a general responsibility evaluation system, we examined the possibility of using open data to solve these limitation issues. This study proposes a novel approach of combining image detection (head-on/angle crash detection and traffic signs detection), open data from OpenStreetMap API, and a knowledge system to assess actors' responsibilities more efficiently from a crash video recorded by a mobile device. With this new approach, the system can handle more accident cases such as accidents without traffic lights, can perform better even during the night, and can get automatically information such as vehicle speed, crash location, and weather which are important for responsibility evaluation.

[1] https://wiki.openstreetmap.org/wiki/API

In this work, the research question we want to answer is: **How improved can a computer vision-based system be, by associating open data to assess automatically responsibilities after a head-on/angle vehicle collision occurs?** Through this study we answer this question by (1) making a review of related and existing works, highlighting their limitations, (2) describing our approach to the solution, experimenting how it performs better than a previously proposed solution, (3) highlighting its limitations and explaining our future work.

4.2 Vehicle Collision Responsibility Evaluation System

4.2.1 Design of the system

In a previous study, we implemented a system that uses the crash video recorded by a real driving recorder of one of the vehicles involved in the crash as the input data source. The system uses image detection to detect crash time, and traffic lights and uses a rule-based knowledge system to assess each actor's responsibility within the crash. This method was successful and helped evaluate responsibilities quickly for head-on/angle crossroad crashes with traffic lights during day time. However, it quickly reached its limits while using the system during the night or while expanding the system to support other crossroad head-on/angle crashes such as crashes with no traffic lights. To solve these limitation issues, we changed the architecture and working flow of the system and associated the usage of open data from OpenStreetMap API that helps to get information about roads and traffic signs independently of the weather, visibility, and time. With such a change, the system can handle more accident cases such as accidents without traffic lights, can perform better even during the night, and can get automatically information such as vehicle speed, crash location, and weather which are important for responsibility evaluation. The new system's architecture is based on a mobile application (the front end) and a server (the back end), each having a specific job and working independently to achieve one sub-goal of the entire system. The mobile application (as shown in Figure 5.2) is a vehicle recorder application that can be installed on a mobile device (a smartphone or tablet) and used to record driving experiences. It streams in real-time the recorded video to the server which saves it as a sequence of individual images and uses them later for responsibility assessment in case of incident. It also sends in real-time to the server information about the speed, the GPS location, and the orientation of the vehicle. The server receives in real time the video streams from the mobile application and saves them as sequences of individual data-tagged images (containing data about the speed, the GPS location, and the orientation of the vehicle) in the user's personal folders. Therefore, when the user sends a request for responsibilities evaluation through the mobile app, the server first uses its crash detection model to detect the crash time in the saved data-tagged images. It then uses its traffic light detection model to detect all traffic lights within the data-tagged images. After decoding the data-tagged images and getting back the data they contain about the speed, the GPS location, and the orientation of the vehicle, it uses the GPS location to retrieve road information from OpenStreetMap API such as road width, and the presence of other traffic signs (such as stop signs, pedestrian crosswalks, and speed limit signs). Finally, it uses a rule-based knowledge system of road rules and the vehicle's speed and orientation to deduct each party's probable responsibility for its explainability.

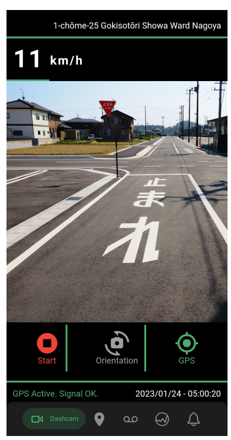

Figure 4.1: A screenshot of the Dashcam screen in the mobile app

4.2.2 Crash time detection and responsibility assessment

Once the mobile app starts streaming, the server receives in real-time the video streamed by the mobile app. It saves the video streams as sequences of individual data-tagged images (containing data about the speed, the GPS location, and the orientation of the vehicle) in the user's personal folders. To tag images and save additional data into them, we use their Exif (Exchangeable image file format) properties. Exif is a standard that specifies formats for images, sound, and ancillary tags used by digital cameras (including smartphones), scanners, and other systems handling image and sound files recorded by digital cameras. For crashes without traffic lights, the system gets from the images to process, the saved metadata (Exif header data about the vehicle's speed, its GPS location, and its orientation) and uses it to retrieve road information (see Table 4.1) from the OpenStreetMap API before accessing responsibilities. When a head-on/angle crash occurs, and a responsibility assessment request is sent by the mobile app, the server goes through three processes to get responsibilities.

Crash Time Detection

To detect the crash time in images, we retrained, updated, and used a previously created custom object detection model that can detect a crash using YOLOv5 a convolutional neural network for performing object detection in real time[2]. We used our previously created custom dataset (Yawovi, Kikuchi, and Tadachi Ozono 2022) of 1530 images of head-on/angle crashes to retrain our updated YOLOv5 model and get it ready for crash detection. The output of this process is images of the environment of the crash a few seconds before and after the crash.

Traffic Light Detection

This process is the traffic light detection phase. We retrained, updated, and used a previously trained YOLO custom object detection model with a dataset of thousands of images of green, red, and yellow traffic lights that we described in a previous work (Yawovi, Kikuchi, and Tadachi Ozono 2022). The output of this process is the list of all traffic lights detected in the crash environment's images a few seconds before and after the crash. This output is used by the third process to evaluate actors' responsibilities through an inference on a rule-based knowledge system.

---

[2] `https://doi.org/10.5281/zenodo.7002879`

Responsibility Evaluation

Usually, after a crash, the police determine responsibilities based on the evaluation of each actor's degree of negligence (or percentage of fault). Our system also uses this degree of negligence's logic to evaluate responsibilities. In Japan, legal books on negligence offset rates in civil traffic proceedings are widely used to determine the degree of negligence. In these books, there are predefined degrees of negligence with different kinds of crashes. Since our system is based on Japanese traffic rules, we use those predefined degrees of negligence to obtain responsibilities. For example, let's consider two crossroad head-on/angle crashes involving two vehicles Vehicle A (let X_A) and Vehicle B (let X_B) at an intersection (let P), one with traffic lights and the other without traffic lights. In such cases, according to Japanese traffic rules, we have basic degrees of negligence (let N) for each vehicle depending on the situation:

- **Situation 1 (with traffic lights)**: "*In case the traffic light is green for Vehicle A and red for Vehicle B, the degree of negligence for Vehicle A is 0, and the one of Vehicle B is 100. In case the traffic light is yellow for Vehicle A and red for Vehicle B, the degree of negligence for Vehicle A is 20, and the one for Vehicle B is 80. In case the traffic light is red for Vehicle A and red for Vehicle B, the degree of negligence for Vehicle A is 50, and the one for Vehicle B is 50.*"

  In first-order logic, this can be expressed as:

  Let T be traffic light, the predicates $is\_intersection$ to check if a crash spot is an intersection or not, is_green to check if a traffic light is green or not, is_yellow to check if a traffic light is yellow or not, and is_red to check if a traffic light is red or not.

$$(is\_intersection(P) \wedge is\_green(X_A, T) \wedge is\_red(X_B, T)) \implies (N(X_A) = 0 \wedge N(X_B) = 100)$$

$$(is\_intersection(P) \wedge is\_red(X_A, T) \wedge is\_red(X_B)) \implies (N(X_A) = 50 \wedge N(X_B) = 50)$$

- **Situation 2 (without traffic lights)**: "*If Vehicle B was going straight on a priority road and Vehicle A was coming from a small road, the* *degree of negligence for Vehicle A is 90, and the one for Vehicle B is 10.*"

In first-order logic, this can be expressed as:

Let R be the road, the predicates *is_intersection* to check if a crash spot is an intersection or not, *is_priority_road* to check if a road is a priority road or not, *is_small_road* to check if a road is a small road or not.

$$(is\_intersection(P) \wedge is\_priority\_road(X_A, R) \wedge \\ is\_small\_road(X_A, R)) \implies (N(X_A) = 90 \wedge N(X_B) = 10)$$

In Situation 1, there is only one piece of information needed to evaluate responsibilities: the status of the traffic light. Therefore, this situation can be easily handled if the traffic light status (either green, yellow, or red) is known. Our current system, as well as our previously proposed system, thanks to computer vision (its traffic light detection model) can detect the presence and status of the traffic light and evaluate responsibilities. However in Situation 2 in which there is no traffic light, there is a new type of information needed: the type of the road (priority road) and its width (small road). In such a situation, computer vision cannot be of any help, inducing the fact that our previously proposed system will fail to evaluate responsibilities. That is why the current system, instead of relying only on object detection to evaluate responsibilities, has an additional layout that fetches and uses open data from OpenStreetMap API.

The two situations described above are only a few of many scenarios of crossroad head-on/angle crashes that can occur (such as overtaking in an intersection where overtaking is prohibited and passing in intersections where passing is allowed), and all of them can not be handled by the current system. For each currently supported scenario (crossroad head-on/angle crashes in an intersection with/without traffic lights on a priority road, one-way road, or roads with the same width), the system fetches any necessary additional data (such as stop signs and speed limits) from OpenStreetMap API. Table 4.1 summarizes the type of information needed for supported crash scenarios and shows their availability and accessibility in our current system in comparison with the previous one. In addition, Table 4.2 describes all information required by the system for responsibility evaluation and it's source in the current system using the first-order logic representation where T, P, R, X, S, N represent traffic light, crash spot, road, vehicle, vehicle's speed and degree of negligence respectively and the predicates $is\_intersection$ to check if a crash spot is an intersection or not,$is\_t\_junction$ to check if a crash spot is a T-junction or not, $is\_priority\_road$ to check if a road is a priority road or not, is_small_road to check if a road is a small road or not, $is\_one\_way\_road$ to check if a road is a one-way road or not, $road\_width$ to get the width of a road, is_green to check if a traffic light is green or not, is_yellow to check if a traffic light is yellow or not, is_red to check if a traffic light is red or not, $is\_flashing\_red$ to check if a traffic light is flashing-red or not, $is\_flashing\_yellow$ to check if a traffic light is flashing-yellow or not, $is\_right\_turn\_green$ to check if a traffic light is right-turn-green or not, has_sign to check if there is a traffic sign or not, using the domain $SN = \{STOP, NONE\}$, $speed\_limit$ to check if there is a speed limit sign or not, $turn$ to check if a vehicle turns right or left or not, using the domain $DIR = \{LEFT, RIGHT, NONE\}$, $direction$ to get the side a vehicle is coming from, using the domain $SD = \{LEFT, RIGHT\}$, $orientation$ to get the orientation of a vehicle.

Table 4.1: List of information needed for responsibility evaluation and their accessibility/availability in our current and previous system

Information needed for crashes	**Accessibility & Availability**	
	Previous system	***Current system****
Situations with traffic lights		
Traffic lights presence	Yes ✓	Yes ✓
Traffic lights status	Yes ✓	Yes ✓
Situations without traffic lights		
Road width	No ✗	Yes ✓
Road type	No ✗	Yes ✓
Direction	No ✗	Yes ✓
Stop signs presence	No ✗	Yes ✓
Speed limit signs presence	No ✗	Yes ✓

Table 4.2: List of information required by the system for responsibility evaluation and it's source in our current system

Information Needed	**Information Source**		
	Detection model	***OpenStreetMap API***	*Vehicle*
T, P, R, X, S, N represent traffic light, crash spot, road, vehicle, vehicle's speed and degree of negligence respectively. The domains used in has_sign, $turn$, $direction$ are respectively $SN = \{STOP, NONE\}$, $DIR = \{LEFT, RIGHT, NONE\}$, $SD = \{LEFT, RIGHT\}$			
$is\_green(T)$	Yes ✓	No ✗	No ✗
$is\_yellow(T)$	Yes ✓	No ✗	No ✗
$is\_red(T)$	Yes ✓	No ✗	No ✗
$is\_flashing\_red(T)$	Yes ✓	No ✗	No ✗
$is\_flashing\_yellow(T)$	Yes ✓	No ✗	No ✗
$is\_right\_turn\_green(T)$	Yes ✓	No ✗	No ✗
$is\_intersection(P)$	No ✗	Yes ✓	No ✗
$is\_t\_junction(P)$	No ✗	Yes ✓	No ✗
$is\_priority\_road(X, R)$	No ✗	Yes ✓	No ✗
$is\_small\_road(X, R)$	No ✗	Yes ✓	No ✗
$is\_one\_way\_road(X, R)$	No ✗	Yes ✓	No ✗
$road\_width(X, R)$	No ✗	Yes ✓	No ✗
$has\_sign(X, SN)$	No ✗	Yes ✓	No ✗
$speed\_limit(X, S)$	No ✗	Yes ✓	No ✗
$turn(X, DIR)$	No ✗	No ✗	Yes ✓
$direction(X, SD)$	No ✗	No ✗	Yes ✓
$orientation(X)$	No ✗	No ✗	Yes ✓

4.3 Conclusions

Very few studies have gone beyond vehicle collision detection to resolve the problem of responsibility assessment after a crash. To solve that problem, in previous work, we proposed an approach based on image detection and a rule-based knowledge system to automatically assess actors' responsibilities when a crash occurs. With some limitations observed in this approach such as the difficulty to manage more crashes, we propose a new framework that combines open data (with OpenStreetMap API), image detection, and the rule-based knowledge system. The system can recognize a crash within images generated from video streaming of a mobile app, detect traffic signs and finally assess the responsibilities of actors within the crash by using the vehicle's speed, its GPS location, and its orientation as well as OpenStreetMap API to get information about the road. Our experiments showed promising results on how the system performs better than the previous system in assessing each party's responsibility when crashes with/without traffic lights occur. The system performs well when the crash occurs during daytime either with traffic lights, or without traffic lights; and during nighttime without traffic lights by using open data on a road map.

In future work, based on the promising future of our technique of combining image recognition and open data, we plan to improve our system to support more and more crossroad head-on/angle crash cases (such as when there is a red blinking/yellow blinking signal, in case of overtaking in an intersection where overtaking is prohibited, and passing in intersections where passing is allowed) and make additional experiments.

Chapter 5

Responsibility Evaluation in Vehicle Collisions from Mobile App Driving Recorder Videos Using Open Data

5.1 Introduction

Despite existing critical needs for innovative solutions that can shorten decision time and help victims get their compensations faster, there has been limited research dedicated to evaluating responsibility in the aftermath of vehicle collisions. Existing studies have focused only on individual aspects such as collision detection, crash risk prediction, crash anticipation, or responsibility attribution through data-driven processes. Assessing responsibility after a crash is a complex task requiring advanced knowledge of road rules. For straightforward scenarios like crashes with traffic lights, decisions are fast and easy. However, in situations such as crashes without any traffic signs, expert knowledge is essential. Automating such tasks demands innovative and high-level approaches, representing a necessity for the future of the automobile and insurance industry

This chapter introduces the improved version of the previous system that is capable of detecting vehicle collision and implements an original responsibility assessment process to assess drivers responsibilities. It employs object detection for collision detection and an association of a knowledge rule-based system and open data from the OpenStreetMap API for responsibility assessment. The responsibility assessment process involves four steps: (1) detecting the crash time within the crash video, (2) identifying all traffic lights within the video, (3) obtaining road information from the OpenStreetMap API, such as road width and the presence of other traffic signs if necessary, and (4) utilizing a rule-based knowledge system of road rules, vehicle speed, and orientation to deduce the probable responsibility of each party involved. The system focuses on head-on/angle crashes involving two cars and facilitates the seamless sharing of evaluation results with the police and insurance companies within minutes of a collision. It comprises a mobile application and a server. The mobile application, when installed on a smartphone or tablet, acts as a vehicle recorder that users can employ to document their driving experiences. It streams real-time recorded videos to the server, which saves them for later responsibility assessment in the event of an incident. Additionally, it transmits real-time information to the server regarding speed, GPS (Global Positioning System) location, and orientation.

In the previous chapter, the system supported only three types of crash scenarios without traffic lights (priority roads/one-way roads/roads with the same width) within three weather conditions (sunny/cloudy/rainy), in addition to those with traffic lights. This study further improves the system which now supports six different types of crash scenarios without traffic lights (priority roads/one-way roads/roads with the same width/roads with stop signs/roads with speed limit signs/roads with flashing red or yellow signals) within six harsh weather conditions (sunny/cloudy/rainy/stormy/snowy/-foggy). The support of these additional scenarios and weathers demanded retraining crash detection and traffic light detection models, adding more rules to our knowledge-based system, and enhancing the algorithm and process of responsibility assessment. Additionally, extensive experiments are also conducted with results showing that the system performs better than its previous version, mainly during nighttime without traffic lights (up to 93% accuracy against 82.5% obtained previously). The significant difference and advantage of this system over existing ones is its automation of responsibilities evaluation for the police, claims adjusters, and victims themselves as well as its applicability for autonomous vehicles.

Here is the summary of this study's contributions:

- Implementation of an advanced system that supports both collision detection and responsibility assessment, presenting a comprehensive and practical solution.
- Extension of the system to support a broader range of crash scenarios and weather conditions.
- Improvement in the performance of the system, particularly in nighttime scenarios.
- Improvement of a rule-based knowledge system with OpenStreetMap API for more accurate responsibility assessment.
- Conducting extensive experiments to validate the system's efficiency and reliability in real-world usage.
- Demonstration of the system's superiority through comparisons with existing research in the field.

5.2 Design of the system

The new system architecture consists of a mobile application (the frontend) and a server (the backend), each with specific roles and operating independently to achieve sub-goals within the overall system. The mobile application, illustrated in Figure 5.2, functions as a vehicle recording application that can be installed on a mobile device (smartphone or tablet) for capturing driving experiences. It streams the recorded video in real-time to the server, which saves it as a sequence of individual images. These images are later utilized for responsibility assessment in the event of an incident. The mobile application also transmits real-time information about vehicle speed, GPS location, and orientation to the server. The server receives the video streams from the mobile application in real-time and stores them as sequences of data-tagged images in the user's personal folders. These data-tagged images contain information about the vehicle's speed, GPS location, and orientation. When a user submits a request for responsibility evaluation through the mobile app, the server utilizes its crash detection model to identify the crash time within the saved data-tagged images. It then employs its traffic light detection model to locate all traffic lights in the data-tagged images. After decoding the data within the images and retrieving the vehicle's speed, GPS location, and orientation, the server employs the GPS location to extract road information from the OpenStreetMap API. This information includes road width and the presence of other traffic signs such as stop signs, pedestrian crosswalks, and speed limit signs. Finally, using a rule-based knowledge system of road rules along with the vehicle's speed and orientation, the server determines the probable responsibility of each party involved in the collision. Figure 5.1 illustrates the system architecture with its two components.

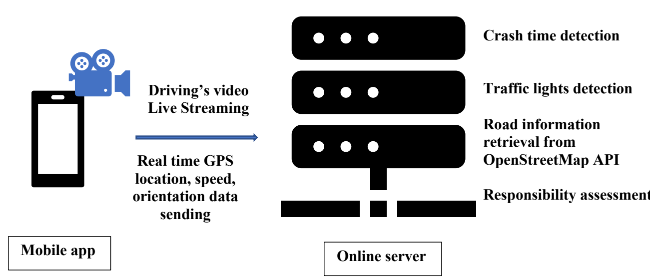

Figure 5.1: The architecture of the system with the collaboration between the front-end and the back-end

5.3 Implementation

5.3.1 The front-end: Driving recorder app

To develop a mobile application compatible with various devices such as smartphones and tablets, we opted to use Flutter (`https://github.com/flutter/flutter`), an open-source UI software development kit created by Google. Flutter allows for the creation of cross-platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web, all from a single codebase. Although our application has the capability to be compiled for multiple operating systems, we have chosen to focus on Android and iOS devices.

To enable video streaming within the system, we utilize WebRTC (`https://webrtc.org/`), a free and open-source project that provides real-time communication capabilities via application programming interfaces (APIs) for web browsers and mobile applications. This technology allows for direct peer-to-peer communication, enabling audio and video communication within web pages without the need for plugins or native app downloads.

To begin using the mobile application, it must be installed on an Android or iOS device. Due to some limitations with Flutter, the app can only be installed on devices running Android 5.0 (API level 21, released in October 2014) or later, and iOS 9 (released on September 16, 2015) or later. Fortunately, the majority of devices currently in use run on versions later than Android 5.0 and iOS 9, minimizing any impact on app usability.

The mobile app comprises various screens and functionalities, with the key ones being:

- Dashcam screen: This is the initial and primary screen of the app. Here, users can manage real-time recording (start/pause/stop), adjust the camera orientation, and activate/deactivate the GPS. The screen also provides real-time information about the vehicle's speed, location (street name), and local time. Figure 5.2 illustrates a screenshot of this screen.

- Map: Users can view the vehicle's current position and navigation details on a map implemented with OpenStreetMap. This screen serves as the navigation function of the app and enables users to search for places by address and obtain the corresponding route.

- Recordings screen: This screen allows users to view and manage the list of recorded videos (delete, download, share, etc.) and request responsibility assessments. After a collision occurs and the crash is recorded, users can quickly obtain responsibility evaluations through this screen, which displays all recorded videos and provides a ”Predict Responsibility” button for each recording.

Upon launching the app and initiating video streaming, the app starts recording and streaming the content to the server. Leveraging WebRTC technology, the app establishes direct peer-to-peer communication with the server, transmitting the streaming data in real-time. Prior to the app's full functionality, the device's GPS must be activated. This is necessary because each frame sent to the server must be tagged with crucial additional information that will be utilized in the event of a collision. This additional data includes the vehicle's speed, GPS location, and orientation. Consequently, in addition to the video data, the app sends this supplementary information to the server, which automatically processes and associates it with all saved frames.

As depicted in Figure 5.3, all recordings are displayed in a daily-based list format, filtered by creation date from newest to oldest.

5.3.2 Crash time detection and responsibility assessment

For the backend implementation, we used aiortc (`https://github.com/aiortc/aiortc`), a Python library for WebRTC (Web Real-Time Communication) and ORTC (Object Real-Time Communication). This library is built on top of asyncio, which is Python standard asynchronous I/O framework.

When the mobile app starts streaming, the Python-based backend server receives the video stream in real-time. It saves the incoming video streams as sequences of individual data-tagged images, with each image containing crucial data such as the vehicle's speed, GPS location, and orientation. To tag the images and store additional data, we utilize the Exif properties. Exif is a standard format that defines the specifications for image, sound, and ancillary tags used by digital cameras, including smartphones. By leveraging the Exif properties, we can easily retrieve and store important metadata recorded by digital cameras. For crashes occurring without traffic lights, the system processes the saved images and extracts the relevant metadata (Exif

Figure 5.2: A screenshot of the recording screen in the mobile app (Yawovi, Kikuchi, and Tadachika Ozono 2023)

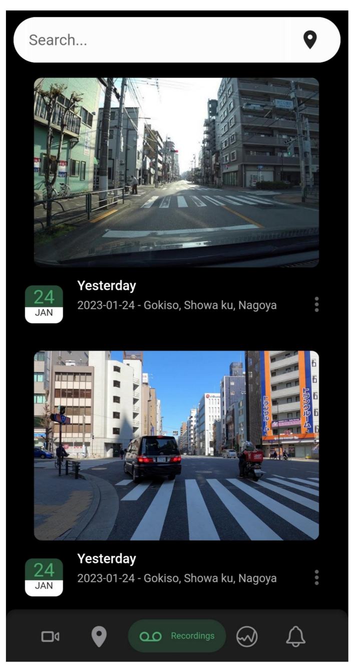

Figure 5.3: A screenshot of the recordings screen in the mobile app

header data) regarding the vehicle's speed, GPS location, and orientation. This information is used later during the responsibility assessment process. When a head-on/angle crash occurs and a responsibility assessment request is sent from the mobile app, the server undergoes three key processes to determine responsibilities. Firstly, the server analyzes all the saved frames within a specified time period, detects the beginning of the crash time, and narrows down the frames to be processed to 240. This selection corresponds to a 10-second video at 24 frames per second (5 seconds before and after the crash). Secondly, it analyzes the 240 images to determine the presence or absence of traffic lights. Finally, it uses the process descrived in Algorithm 1 and employs a rule-based knowledge system of road rules, combined with an inference engine, to deduce the responsibilities of the actors involved in the crash.

Crash Time Detection

This is the first process of the responsibility assessment flow. It analyzes all the saved frames during a given time period, detects the beginning of the crash time, and reduces the number of frames to work with to 240 (equivalent to a 10 seconds 24 fps (frame per second) video, 5 s each before and after the crash). To detect the crash time in images, we retrained, updated, and used a previously created custom object detection model that can detect a crash using YOLOv8 the latest version of a convolutional neural network for performing object detection in real time (`https://doi.org/10.5281/zenodo.7347926`). YOLOv8 is a family of object detection architectures and models pre-trained on the COCO dataset (T.-Y. Lin et al. 2014). It is one of the fastest versions in the YOLO series. YOLO (Redmon et al. 2015) an acronym for 'You only look once', is a convolutional neural network for performing object detection in real-time. We used our previously created custom dataset (Yawovi, Kikuchi, and Tadachi Ozono 2022) of 1530 images of head-on/angle crashes to retrain our updated YOLOv8 model and get it ready for crash detection. To build this dataset, we extracted crossroad head-on/angle crash images from videos. We used YouTube as a source of video data. We started by searching compilation videos of crossroad accidents recorded by driving recorders that we could collect manually. The major part of the compilation videos we found was not dedicated only to head-on/angle crashes in the context of crossroad accidents. Most of them were a compilation of all kinds of car crashes that occurred in the United States of America, Thailand, Russia, and India with the driving recorders, not necessarily the ones of the vehicles involved in the crash. Therefore, we had to watch all the videos and select the parts containing the kinds of crashes and angles of recording of interest to us. In total, we watched 103 videos (each one having an average of 15 min in length) and finally selected 68 of them. We selected those compilation videos based on acceptable image quality (480px and above), the fact that they have at least one video related to a head-on/angle crash in a crossroad accident context, and the fact that the crash details are clear enough to use the images during the labeling step. After collecting the compilation videos, we had to extract and resize the frames (images) of the parts that we were interested in. We used the library OpenCV (Open Source Computer Vision) to extract the frames and scikit-image ((`https://scikit-image.org/`) (accessed on 11 May 2022)) to modify and resize them so that they have the same size and structure. For homogeneity and consistency in the data, the frames of the videos were converted to a lower width (500px). After extracting and resizing frames from the videos, we got a total of 1530 head-on/angle crash images and annotated them. After all these steps, we got a total of 1530 crash images from different countries, such as the United States of America, Russia, and India, to train the model. To annotate the images, we used LabelImg (`https://github.com/tzutalin/labelImg`) (accessed on 11 May 2022) a graphical image annotation tool, and label object bounding boxes in images written in Python. We then separated the annotated images into training data (1071 images equivalent to 70% of the dataset), validation data (306 images equivalent to 20% of the dataset), and test data (153 images equivalent to 10% of the dataset). The output of this process is images of the environment of the crash a few seconds before and after the crash. This output is used by the second process to detect traffic lights and other crucial information used to evaluate responsibilities within crashes.

Traffic Light Detection

This process is the traffic light detection phase. To implement it, we used again the object detector YOLOv8 and the library Open CV. We retrained, updated, and used a previously trained YOLO custom object detection model with a dataset of thousands of images of green, red, and yellow traffic lights that we described in a previous work (Yawovi, Kikuchi, and Tadachi Ozono 2022). To build a custom dataset with enough images of traffic lights, we investigated and downloaded them one by one on Google Image Search. In addition, fortunately, we found a ready-to-use labeled traffic light image dataset on the public dataset of Roboflow.com (`https://public.roboflow.com/object-detection/self-driving-car`) (accessed on 11 May 2022). The dataset is a set of 15,000 images taken on roads in the United States of America. It contains not only traffic lights but also traffic signs for pedestrians, bikers, and cars. Needing only traffic lights annotations, we extracted 1300 images of traffic lights from the original dataset (red, green, yellow, red left, green left, and yellow left). We finally got a total of 3000 images of traffic lights after adding the ones we downloaded manually from Google and annotated them. For labeling the dataset images, we used LabelImg, to annotate them. We then separated the annotated images into training data (2100 images equivalent to 70% of the dataset), validation data (600 images equivalent to 20% of the dataset), and test data (300 images equivalent to 10% of the dataset). The output of this process is the list of all traffic lights detected in the crash environment's images a few seconds before and after the crash. This output is used by the third process to evaluate actors' responsibilities through an inference on a rule-based knowledge system.

Responsibility Evaluation

Typically, following a car accident, the police assign responsibilities by considering how much each person was at fault or negligent. Our system follows the same principle, using a logic based on the degree of negligence. In Japan, established legal guidelines, found in books used in civil traffic cases, help determine this degree of negligence. These books outline different levels of fault corresponding to various types of accidents. Since the system adheres to Japanese traffic regulations, we rely on these predetermined levels of negligence to assess responsibilities. For instance, let's examine three scenarios of head-on collisions at a crossroad involving two vehicles Vehicle A (let X_A) and Vehicle B (let X_B) at an intersection (let P). One intersection has traffic lights, and the other does not. In such cases, according to Japanese traffic rules, we have basic degrees of negligence (let N) for each vehicle depending on the situation:

- **Scenario 1 (with traffic lights)**: "*In case the traffic light is green for Vehicle A and red for Vehicle B, the degree of negligence for Vehicle A is 0, and the one of Vehicle B is 100. In case the traffic light is yellow* *for Vehicle A and red for Vehicle B, the degree of negligence for Vehicle A is 20, and the one for Vehicle B is 80. In case the traffic light is red for Vehicle A and red for Vehicle B, the degree of negligence for Vehicle A is 50, and the one for Vehicle B is 50.*"

In first-order logic, this can be expressed as:

Let T be traffic light, the predicates $is\_intersection$ to check if a crash spot is an intersection or not, is_green to check if a traffic light is green or not, is_yellow to check if a traffic light is yellow or not, and is_red to check if a traffic light is red or not.

$$(is\_intersection(P) \land is\_green(X_A, T) \land is\_red(X_B, T))$$
$$\implies (N(X_A) = 0 \land N(X_B) = 100)$$
$$(is\_intersection(P) \land is\_yellow(X_A, T) \land is\_red(X_B, T))$$
$$\implies (N(X_A) = 20 \land N(X_B) = 80)$$
$$(is\_intersection(P) \land is\_red(X_A, T) \land is\_red(X_B))$$
$$\implies (N(X_A) = 50 \land N(X_B) = 50)$$

- **Scenario 2 (without traffic lights)**: "*If Vehicle B was going straight on a priority road and Vehicle A was coming from a small road, the degree of negligence for Vehicle A is 90, and the one for Vehicle B is 10.*"

  In first-order logic, this can be expressed as:

  Let R be the road, the predicates $is\_intersection$ to check if a crash spot is an intersection or not, $is\_priority\_road$ to check if a road is a priority road or not, is_small_road to check if a road is a small road or not.

$$(is\_intersection(P) \land is\_priority\_road(X_B, R) \land$$
$$is\_small\_road(X_A, R)) \implies (N(X_A) = 90 \land N(X_B) = 10)$$

- **Scenario 3 (without traffic lights)**: "*If Vehicle A collides with and Vehicle B who was violating a one-way road, the degree of negligence for Vehicle A is 20, and the one for Vehicle B is 80.*"

  In first-order logic, this can be expressed as:

Let R be the road, the predicates $is\_intersection$ to check if a crash spot is an intersection or not, $is\_one\_way\_road$ to check if a road is a one-way road or not

$$(is\_intersection(P) \wedge is\_one\_way\_road(X_B, R)) \\ \implies (N(X_A) = 20 \wedge N(X_B) = 80)$$

In Scenario 1, evaluating responsibilities is straightforward; there is only one important information: the status of the traffic light -whether it is green, yellow, or red. However, in Scenarios 2 and 3, where there are no traffic lights, a different set of information becomes crucial: the road type (priority road or one-way) and its width (narrow road). These factors are necessary for determining responsibilities. These three scenarios represent only a few examples of head-on/angle crashes, with many various crash scenarios demanding different sets of information. Such a situation makes it difficult for automatically determining responsibilities. To address the challenge, we designed a novel process that helps to successfully assess responsibilities within different crash scenarios.

The following steps outline the proposed process as illustrated by Algorithm 1 and Algorithm 2:

1. First, the system extracts metadata from the main crash frame, including latitude and longitude coordinates, using Exif data of the saved images.

2. Then, it calculates a bounding box around the main crash frame coordinates, creating a search area of 20 meters by 20 meters.

3. It then constructs a specific query to fetch data within this defined area. An example of query is expressed by Listing 5.1

Listing 5.1: Formulated Query Example

```
[timeout:25];
node(%(35.1497066)s,%(136.9295019)s,
%(36.49236)s,
%(136.2956719)s) [highway];
//(._;>;);
out body;
```

4. The formulated query is executed through the OpenStreetMap API. This query retrieves nodes representing various elements like traffic signals, stop signs, and other relevant road features such as road type and width within the specified search area.

5. For each retrieved node, the system checks its type and tags. If the node represents a road element and is not yet recorded, the system saves it.

6. The detected traffic signs, along with any other relevant information, are aggregated and sent to the inference engine of road rules.

7. Multiple logic deductions (as described by Algorithm 2) are made and the result is finally output to the user. If there is no result after multiple deductions from the knowledge system, the user get ”unknown”. This is the output in case of failure in the responsibility assessment by the system.

5.4 Conclusions

Most studies only focused on detecting crashes, not on deciding who is at fault. We tackled this issue in a previous work by using image detection and a rule-based system. This helped assigning drivers responsibilities automatically. However, this approach had some limitations, especially when dealing with crashes in situations without traffic signs. To address these limitations, we introduced a new method that implements open data, image detection, and a rule-based system. This system can spot accidents in images from a mobile app, identify traffic signs, and determine who is at fault. It considers factors like vehicle speed, GPS location, orientation, and road’s information. Additional and extensive experiments demonstrated promising results and the system’s efficiency in performing well in various scenarios, both during the day and night, regardless of the presence of traffic lights. In addition, through detailed case studies and comparisons, the effectiveness and superiority of the system are demonstrated. Future work includes the further enhancement of the system. We plan to handle more complex cases, such as accidents involving overtaking in intersections where it is prohibited, and passing in intersections where it is allowed.

Algorithm 1: Algorithm for automatically assessing responsibilities

```
Data: Main crash frame metadata (Latitude, Longitude)
Result: Percentage of fault or unknown
Procedure EvaluateResponsibility
    Main crash frame metadata
Calculate search area: 20 meters × 20 meters around main crash
  frame coordinates;
Construct query using search area information;
Execute query through OpenStreetMap API ;  // Retrieve nodes
  within the search area
foreach node in queried nodes do
    if node represents a road element and type not recorded then
        Add node to detected traffic signs and relevant road features
          list;
    end
end
;        // Send Detected traffic signs and relevant road
  features as Facts to the inference engine
Facts: 'type_of_accident': 'head-on','road_width':
  '15','vehicle_speed': '75', 'road_type': 'one-way',
  'traffic_light': 'none';
;        // Use the inference engine to make a conclusion
Result: result = Infer(facts, rules);
return result;     // Result from the Function Infer that is
  more detailled in the next algorithm
return unknown;
```

Algorithm 2: Inference Engine for Assessing Responsibilities

Data: Rules, Facts
Result: Conclusion
Function *Infer*
    facts, rules
for *each rule in rules* **do**
    if *all conditions in facts match conditions in rule* **then**
        return conclusion from rule;
    end
end
return None;
; `// Define the rules and facts`
Rule 1:
$(\mathtt{is\_intersection}(P) \land \mathtt{is\_green}(X_A, T) \land \mathtt{is\_red}(X_B, T)) \implies (N(X_A) = 0 \land N(X_B) = 100)$;
Rule 2:
$(\mathtt{is\_intersection}(P) \land \mathtt{is\_yellow}(X_A, T) \land \mathtt{is\_red}(X_B, T)) \implies (N(X_A) = 20 \land N(X_B) = 80)$;
...
Rule 76: $(\mathtt{is\_intersection}(P) \land \mathtt{is\_one\_way\_road}(X_B, R)) \implies (N(X_A) = 20 \land N(X_B) = 80)$;
; `// Return the result`
Return: `result`;

Chapter 6

Experimental evaluation

6.1 First Round Evaluation

6.1.1 Introduction

To evaluate the system, we performed some experiments with a testing dataset containing 180 head-on and angle crash videos within the context of crossroad accidents that occurred in the United States of America and in India (as for now, the side of driving—either left or right—does not have any impact on the result of the system). The dataset contains 90 videos of crashes that occurred in good visibility conditions (45 each during daytime and nighttime) and 90 videos of crashes with poor visibility (45 each during daytime and nighttime). For each module, we got different results depending on the road environment conditions such as daytime and night time with good visibility (for example, a sunny day or a well-lighted road during the night), and daytime and night time with bad visibility (for example, a snowy day or a bad lighted road during the night).

6.1.2 Evaluation of the Crash Time Module

The crash time detection of the system within a video is achieved through the first module. To evaluate the performance of this module in detecting videos crashes, we carried out the experiment with 180 testing videos. Here, we want to evaluate how well our custom model can detect a crash so that the system can get the crash time. We chose different collided objects (the other actor in the crash) such as cars, vans, and trucks. First, we evaluated the performance of the system during the day with good visibility and bad visibility. Then we evaluated its performance at night with good visibility and bad visibility.

Table 6.1 shows the resulting accuracy of the module's model in detecting the crash in videos during these different environmental conditions with different collided objects. The model's accuracy calculation is expressed by the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100 \tag{6.1}$$

where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative.

Table 6.1: Results of the crash detection model's accuracy.

Collided object & performance of the model (accuracy)	Environment conditions			
	Day		Night	
	Good visibility	Bad visibility	Good visibility	Bad visibility
Car	93.5%	39.9%	82.1%	32.5%
Van	91.2%	38.7%	80.6%	28.3%
Truck	91.5%	38.3%	79.8%	28.1%

As shown in Table 6.1 during daytime and in good visibility conditions, the accuracy of the crash detection model is either 93.5% (for cars), 91.2% (for vans), or 91.5% (for trucks). During the day and in bad visibility, the accuracy drastically drops to reach 39.9% for cars, 38.7% for vans, and 38.3% for trucks. During nighttime and in good visibility conditions, the crash detection model performs relatively well when the collided object is either a car (82.1%), a van (80.6%), or a truck (79.8%). The lowest accuracy is reached when the environment condition is night with poor visibility. The model achieves an accuracy of 32.5% for cars, 28.3% for vans, and 28.1% for trucks.

The limitation of this module is mainly due to the variation of light and the visibility of objects in crash images. When the visibility is good enough, the module performs well with good accuracy. On the other hand, when the visibility is bad, the module suffers from making a good detection of the crash.

6.1.3 Evaluation of the Traffic Signs Detection Module

We evaluated the performance of the system in detecting traffic lights within the 180 test videos by evaluating the second module that is in charge of the task. Here, we want to evaluate how well our custom model can detect a traffic light (green, red, and yellow) in a given image. We made the experimentation in different road environment conditions as we did for the evaluation of the first module. We chose different view distances of the traffic lights from the driving recorder of the vehicle. We evaluated the system when the traffic light is far from the driving recorder (more than 30 m), close to the driving recorder (between 30 m and 10 m), or very close to the driving recorder (less than 10 m).

Table 6.2 shows the resulting accuracy of the module's model in detecting traffic lights in videos with different view distances. Here again, the model's accuracy calculation is expressed by formula (6.1).

Table 6.2: Results of the traffic sign detection model's accuracy.

View distance & performance of the model (accuracy)	Environment conditions			
	Day		Night	
	Good visibility	Bad visibility	Good visibility	Bad visibility
Far	83.5%	40.6%	68.1%	30.2%
Close	91.2%	40.8%	69.5%	30.6%
Very close	97.3%	41.2%	73.1%	36.8%

During the day and under good visibility conditions, the model of the module performs well when the distance from the traffic light from the driving recorder is either very close (97.3%), close (91.2%), or far (83.5%). During the day and in bad visibility, the accuracy drastically drops to 41.2% for very close distances, 40.8% for close distances, and 40.6% for far distances. During nighttime and in good visibility conditions, the traffic sign detection model performs relatively well when the view distance is either very close (73.1%), close (69.5%), or far (68.1%). The lowest accuracy is reached when the environment condition is night with poor visibility. The model achieves an accuracy of 36.8% for very close distances, 30.6% for close distances, and 30.2% for far distances.

The performance of this module depends on light conditions and traffic lights' view distance. When the visibility is good enough and the view distance is very close, the module performs well. However, when the visibility is bad with a far view distance, the module suffers from recognizing traffic lights.

6.1.4 Evaluation of the Responsibilities' Assessment Module

To evaluate the performance of the system in predicting each actor's responsibility within a crash, we evaluated the performance of the third module in different road environment conditions and in two different scenarios: (a) with the user's manual selection of the best result during the crash time detection phase, and (b) without the user's manual selection step. In both scenarios, we randomly partitioned the 180 testing videos as follows:

- With a car as the collided object (60 videos in total): 15 videos during the day in good visibility, 15 videos during the day in bad visibility, 15 videos during the night in good visibility, and 15 videos during the night in bad visibility;
- With a van as the collided object (60 videos in total): Same as the previous partition.
- With a truck as the collided object (60 videos in total): Same as the previous partition.

Evaluation Results with the User's Manual Selection Step

Table 6.3 shows the results of the successfully predicted responsibilities within the videos over the total number of videos tested when the user intervenes during the crash time detection phase and selects the best result to use among the top 20 results. The result is calculated by the proportion of videos in which the system has successfully predicted responsibilities over the total number of videos tested. The proportion formula is expressed below:

$$Proportion = \frac{VS}{VT} * 100 \tag{6.2}$$

where VS is the number of videos in which the system has successfully predicted responsibilities, and VT is the total number of tested videos.

As shown in the results, during the daytime, and in good visibility conditions, with the user's intervention, the module performs well in evaluating actors' responsibilities when the collided object is either a car (successful evaluation within 14 videos over 15 tested in total), a van (successful evaluation within 14 videos over 15 tested in total) or a truck (successful evaluation within 12 videos over 15 tested in total). During the day and in bad visibility, the performance drastically drops to reach good evaluation within 5 videos only over 15 tested in total for cars, 4 videos only over 15 tested in total for vans, and 5 videos only over 15 tested in total for trucks. During nighttime and in good visibility conditions, the system performs relatively

Table 6.3: Results of the module in evaluating responsibilities with user's manual selection.

Collided object & performance of the responsibilities' evaluation	Environment conditions			
	Day		Night	
	Good visibility	Bad visibility	Good visibility	Bad visibility
Car	93% (14/15 videos)	33% (5/15 videos)	93% (14/15 videos)	20% (3/15 videos)
Van	93% (14/15 videos)	26% (4/15 videos)	80% (12/15 videos)	20% (3/15 videos)
Truck	80% (12/15 videos)	33% (5/15 videos)	80% (12/15 videos)	13% (2/15 videos)

well when the collided object is either a car (successful evaluation within 14 videos over 15 tested in total), a van (successful evaluation within 12 videos over 15 tested in total), or a truck (successful evaluation within 12 videos over 15 tested in total). The lowest accuracy is reached when the environment condition is night with bad visibility. The system achieves a performance of good evaluation within 3 videos only over 15 tested in total for cars, 3 videos only over 15 tested in total for vans, and 2 videos only over 15 tested in total for trucks.

Evaluation Results without the User's Manual Selection Step

Table 6.4 shows the results of the successfully predicted responsibilities within videos over the total number of tested videos when the user does not intervene during the crash time detection phase to select the best result to use among the top 20 results. In this scenario, the system automatically takes the first best result of the crash time prediction to evaluate responsibilities. Here again, the result is calculated by the proportion expressed by Formula (6.2.2).

As shown in the results, during the daytime, and in good visibility conditions, without user intervention, the module performs relatively well in evaluating actors' responsibilities when the collided object is either a car (successful evaluation within 10 videos over 15 tested in total), a van (successful evaluation within 9 videos over 15 tested in total) or a truck (successful evaluation within 8 videos over 15 tested in total). During the day and in bad visibility, the performance drastically drops to reach good evaluation

Table 6.4: Results of the module in evaluating responsibilities without user's manual selection.

Collided object & performance of the responsibilities' evaluation	Environment conditions			
	Day		Night	
	Good visibility	Bad visibility	Good visibility	Bad visibility
Car	66% (10/15 videos)	20% (3/15 videos)	53% (8/15 videos)	13% (2/15 videos)
Van	60% (9/15 videos)	13% (2/15 videos)	46% (7/15 videos)	13% (2/15 videos)
Truck	53% (8/15 videos)	20% (3/15 videos)	46% (7/15 videos)	6% (1/15 videos)

within 3 videos over 15 tested in total for cars, 2 videos over 15 tested in total for vans, and 3 videos over 15 tested in total for trucks. During nighttime and in good visibility conditions, the system performs slightly well when the collided object is either a car (successful evaluation within 8 videos over 15 tested in total), a van (successful evaluation within 7 videos over 15 tested in total), or a truck (successful evaluation within 7 videos over 15 tested in total). Here again, the lowest accuracy is reached when the environment condition is night with bad visibility. The system achieves a performance of good evaluation within 2 videos over 15 tested in total for cars, 2 videos over 15 tested in total for vans, and 1 video over 15 tested in total for trucks.

Evaluation Results' Comparison for Both Scenarios

To compare the different evaluation results that were obtained with and without user intervention during the crash time detection process, we put the results in a simple column chart, as shown in Figure 6.1.

By analyzing the chart, we can see that the difference in accuracy in assessing responsibilities between cases with the user's intervention and the one without the user's intervention is in the range of 7% (best case) and 40% (worst case). This leads to the fact that although the system performs relatively well when there is no user intervention, it performs better when that intervention is involved. In addition, the performance of the system depends on the performance of the first and second modules of the system, either with or without the user's intervention. If one of the two modules or both do not perform well, it affects the ability of this module to evaluate responsibilities.

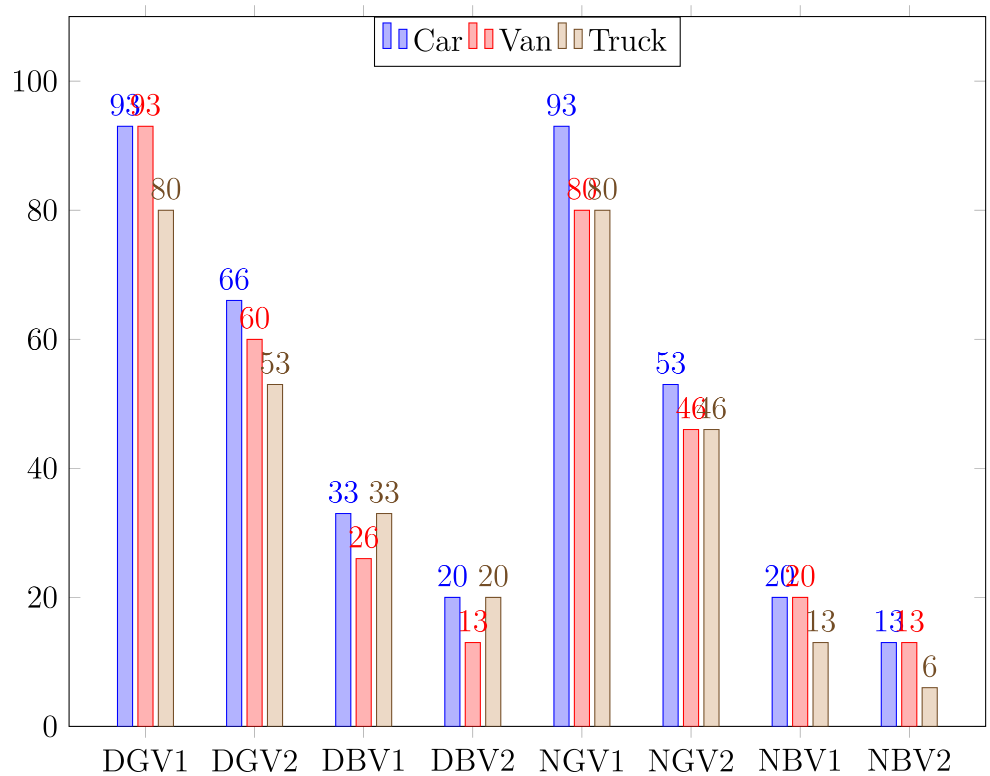

Figure 6.1: Comparison graph for evaluation results with and without user intervention

In some situations where there is no result after multiple deductions from the knowledge system, the third module could fail in assessing responsibilities even if both the first and second modules returned correct results. However, during the experiments, we did not get such a situation. Therefore, the limitations of our system are due to (a) the variation of light, (b) the visibility of collided objects, (c) the view distance of traffic lights within crash videos, and (d) the availability of matching rules in the knowledge system.

Legend:

- D: Day
- N: Night
- GV: Good visibility
- BV: Bad visibility
- 1: with user's intervention
- 2: without user's intervention
- For example, DGV1 means Day - Good visibility with user's intervention

6.2 Second Round Evaluation

6.2.1 Introduction

The purpose of this second round evaluation was to see the ability of the system to automatically detect the nature of the crash (crash with traffic lights or crash without traffic lights) and to assess responsibilities. To evaluate the system, we performed some experiments with 80 head-on/angle crash videos within the context of crossroad accidents with and without traffic lights during daytime and nighttime. We simulated the crash by playing the crash videos with a video player on a computer and recording them using the mobile app.

Table 6.7 shows the results of the successfully evaluated responsibilities within the videos over the total number of videos tested for the specific environmental condition. The result is calculated by the accuracy, which is the proportion of videos in which the system has successfully evaluated responsibilities over the total number of videos tested. The accuracy formula is expressed by Formula (6.2.2).

As shown in the results, during the daytime the system performs well in evaluating actors' responsibilities when the crash occurs either with traffic lights (successful evaluation within 18 videos over 20 tested in total) or without traffic lights (successful evaluation within 20 videos over 20 tested in

Table 6.5: Results of the current system in evaluating responsibilities within crashes with/without traffic lights in comparison with the previous one

Type of crash & Responsibility assessment	**Environment Conditions**	
	Day	***Night***
Accuracy: Previous system		
With traffic lights	53-66%	46-53%
Without traffic lights	Not available	Not available
Accuracy: Current system (with open data)		
With traffic lights	90%	55%
Without traffic lights	100%	80%

total). During nighttime the system performs relatively well without traffic lights (successful evaluation within 16 videos over 20 tested in total). The lowest accuracy is reached when the environment condition is night with bad visibility. Here, the system achieves a good performance within only 8 videos over 20 tested in total. Within crashes with traffic lights, the system uses its traffic light detection model to detect the state of existing traffic lights (green, yellow, or red). Generally, the light condition affects object detection (good light condition, better detection), that is why at night, the system does not perform very well. On the other hand, within crashes without traffic lights, the system gets the location of the device at the moment of the crash and uses the OpenStreetMap API to get information about the road before assessing the responsibilities. As a comparison with previous results obtained from the evaluation of our previous system, we see a significant accuracy improvement for crashes with traffic lights either during daytime (90% currently obtained against between 53 and 66% previously obtained) or nighttime (55% currently obtained against between 46 and 53% previously obtained). This shows how improved can our computer vision-based system be, by associating open data to assess responsibilities automatically.

6.2.2 Responsibility Assessment

The purpose of this evaluation is to see the ability of the system to automatically detect the nature of the crash (crash with traffic lights or crash without traffic lights) and to assess responsibilities.

In our prior study (Yawovi, Kikuchi, and Tadachika Ozono 2023), we conducted experiments involving 80 head-on and angle crash videos within the context of crossroad accidents during daytime and nighttime, considering scenarios both with and without traffic lights (40 videos with traffic lights and 40 videos without traffic lights). For situations without traffic lights, the 40 tested videos were mainly for 3 cases: (1) on priority roads (12 videos), (2) on one-way roads (16 videos) and (3) on roads with the same width (12 videos). However, it is pertinent to acknowledge the limitations of this testing dataset and the subsequent need for an expanded investigation to show the validation of our system's effectiveness and its use in real-world scenarios. In response to this necessity, the current experiment significantly augments the experimental dataset adding more diverse situations. We meticulously curated and analyzed 160 head-on and angle crash videos, specifically focusing on crossroad accidents with and without traffic lights (80 videos with traffic lights and 80 videos without traffic lights), spanning various lighting conditions during both daytime and nighttime and expanding the situation to 6 cases: (1) on roads with stop signs (15 videos), (2) on roads with speed limit signs (15 videos), (3) on roads with flashing red/flashing yellow signals (5 videos), (4) on priority roads (15 videos), (5) on one-way roads (15 videos) and (6) on roads with the same width (15 videos). In addition to increasing the number of videos in different scenarios, we made sure to include diverse weather conditions. We added videos taken during winter days with snow, rainy days, and foggy days. This variation allowed us to test the system's performance in challenging weather, providing a more comprehensive understanding of its capabilities. The comparison between the previous testing dataset and the current one is summarized in Table 6.6.

Table 6.6: Comparison between previous and current testing datasets

Details of & the dataset	**Availability**	
	Previous experiments	***Current experiments***
Number of videos	80	160
Scenarios	(1) on priority roads (2) on one-way roads (3) on roads with the same width	(1) on priority roads, (2) on one-way roads (3) on roads with the same width (4) on roads with stop signs ✓ (5) on roads with speed limit signs ✓ (6) on roads with flashing red/flashing yellow signals ✓
Weather conditions	Sunny day Cloudy day Rainy day	Sunny day Cloudy day Rainy day Stormy day ✓ Snowy day ✓ Misty day ✓ Foggy day ✓

Table 6.7 shows the results of the successfully evaluated responsibilities within the videos over the total number of videos tested for the specific environmental condition. The result is calculated by the accuracy, which is the proportion of videos in which the system has successfully evaluated responsibilities over the total number of videos tested. The accuracy formula is expressed by Formula (6.2.2)$Accuracy = \frac{VS}{VT}$ where VS is the number of videos in which the system has successfully evaluated responsibilities, and VT is the total number of tested videos.

As shown in the results, during the daytime the system performs well in evaluating actors' responsibilities when the crash occurs either with traffic lights (successful evaluation within 38 videos over 40 tested in total) or without traffic lights (successful evaluation within 40 videos over 40 tested in total). During nighttime the system performs relatively well without traffic lights (successful evaluation within 35 videos over 40 tested in total). The lowest accuracy is reached when the environment condition is night with bad visibility. Here, the system achieves a good performance within only 24 videos over 40 tested in total. Within crashes with traffic lights, the system uses its traffic light detection model to detect the state of existing traffic lights (green, yellow, or red). Generally, the light condition affects object detection (good light condition, better detection), that is why at night, the system does not perform very well. On the other hand, within crashes without traffic lights, the system gets the location of the device at the moment of the crash and uses the OpenStreetMap API to get information about the road before assessing the responsibilities.

As a comparison with previous results obtained from the evaluation of our previous system, we see a significant accuracy improvement for crashes with traffic lights either during daytime (95% currently obtained against between 53 and 66% previously obtained) or nighttime (60% currently obtained against between 46 and 53% previously obtained). This shows how improved can our computer vision-based system be, by associating open data to assess responsibilities automatically.

In addition, as a comparison with previous results obtained from the evaluation of our system that uses Yolov5, we see a significant accuracy improvement for crashes (1) with traffic lights either during daytime (95% currently obtained against 90 previously obtained) or nighttime (60% currently obtained against 55% previously obtained); and (2) without traffic lights during nighttime (87.5% currently obtained against 80% previously obtained).

Table 6.7: Results of the current system in evaluating responsibilities within crashes with/without traffic lights in comparison with the previous one

Type of crash & Responsibility assessment	Environment Conditions *Day*	Environment Conditions *Night*
Accuracy: Previous system		
With traffic lights	53-66%	46-53%
Without traffic lights	Not available	Not available
Accuracy: Current system with open data (YOLOv5)		
With traffic lights	90%	55%
Without traffic lights : On priority roads	100%	78.5%
Without traffic lights : On one-way roads	100%	82.5 %
Without traffic lights : On roads with the same width	100%	79%
Without traffic lights - Average	**100%**	**80%**
Accuracy: Current system with open data (YOLOv8)		
With traffic lights	95%	60%
Without traffic lights : On priority roads	100%	85.5%
Without traffic lights : On one-way roads	100%	89%
Without traffic lights : On roads with the same width	100%	85%
Without traffic lights : On roads with stop signs	100%	92%
Without traffic lights : On roads with speed limit signs	100%	93%
Without traffic lights : On roads with flashing red/yellow signals	100%	80.5%
Without traffic lights - Average	**100%**	**87.5%**

6.2.3 Comparison with Other Existing Systems

To gauge the significance of our contribution, we conducted a comparative analysis between our work and existing research within the domain of vehicle collision responsibility assessment.

Our system employs a unique approach by integrating image recognition, open data, and a knowledge system. This comprehensive combination ensures a robust methodology for accident responsibility assessment. In contrast, Jang-Hee et al.'s work (Yoo, Kang, and J.-U. Choi 1994) combines neural networks with fuzzy techniques, providing a different technological foundation. Cédric et al.(Garcia et al. 2019) utilize logistic regression with L1 penalty, random forests, and boosting, emphasizing statistical modeling. Chandraratna et al.'s approach (Chandraratna and Stamatiadis 2009) centers on not-at-fault drivers, focusing on specific driver behavior patterns. Regarding rule-based support, our system and Jang-Hee et al.'s work (Yoo, Kang, and J.-U. Choi 1994) both incorporate rule-based systems, enhancing their decision-making capabilities. However, Cédric et al. (Garcia et al. 2019) opt for a scoring system to assess responsibility, while Chandraratna et al. employ (Chandraratna and Stamatiadis 2009) quasi-induced exposure, each offering distinct evaluation methods. In terms of crash detection capacity, our system provides accurate identification of collision events within various scenarios. Jang-Hee et al.'s approach (Yoo, Kang, and J.-U. Choi 1994) lacks crash detection capabilities, focusing more on post-collision assessment methodologies. Cédric et al. (Garcia et al. 2019) and Chandraratna et al. (Chandraratna and Stamatiadis 2009)'s works also do not include crash detection features, concentrating on subsequent analysis methodologies. Regarding automatic traffic sign detection, our system integrates traffic sign detection, enhancing its ability to consider real-time road regulations and signals in responsibility assessment. Jang-Hee et al. (Yoo, Kang, and J.-U. Choi 1994), Cédric et al. (Garcia et al. 2019), and Chandraratna et al. (Chandraratna and Stamatiadis 2009) 's systems do not incorporate traffic sign detection, potentially limiting their contextual understanding in certain situations. In terms of user accessibility, our system caters to a broader user base, including victims, police, and insurance companies. Jang-Hee et al.'s system (Yoo, Kang, and J.-U. Choi 1994) primarily targets insurance companies, emphasizing a specialized user group. Cédric et al.'s approach (Garcia et al. 2019) is tailored for police use, focusing on law enforcement applications. Chandraratna et al.'s work (Chandraratna and Stamatiadis 2009) also caters to police, concentrating on specific driver behaviors within their assessment.

The summary of this comparison is encapsulated in Table 6.8. Notably, our system exhibited effectiveness, superiority and outperformed other works. It stands out for its versatile approach, wider user accessibility, and incorporation of rule-based support, making it a promising solution for accident responsibility assessment across diverse scenarios and user requirements. Each approach brings unique strengths, catering to specific contexts and user bases within the field of accident responsibility assessment.

Table 6.8: Comparison between our work and existing works in the field of vehicle collision responsibility assessment

Functionnalities	***Our system****	*Jang-Hee et al (Yoo, Kang, and J.-U. Choi 1994)*	*Cédric et al (Garcia et al. 2019)*	*Chandraratna et al (Chandraratna and Stamatiadis 2009)*
Approach	Image recognition + Open data + Knowledge system	Neural network + fuzzy techniques	Logistic regression with L1 penalty+ Random forests+ Boosting	Not-at-fault drivers
Rule based support	Yes ✓	Yes ✓	No ✗	No ✗
Crash detection	Yes ✓	No ✗	No ✗	No ✗
Traffic sign detection	Yes ✓	No ✗	No ✗	No ✗
Expert System	Yes ✓	Yes ✓	Yes ✓	Yes ✓
Responsibility detection method	Degree of negligence	Compensation rate	Scoring	Quasi-induced exposure
Users	Victims Police Insurance companies	Insurance companies	Police	Police

Chapter 7

Discussion

7.1 Introduction

The proposed system introduces novel aspects and advancements, setting it apart from state-of-the-art methods in the field such as Dirnbach et al. 2020; Dong, Yan, and Duan 2022; Garcia et al. 2019. Its unique approach of implementing a process that uses image detection, open data, and a rule-based knowledge system allows for a comprehensive and accurate assessment of responsibilities in vehicle collisions. By leveraging real-time video streaming and advanced image processing techniques, it enables prompt and precise evaluation of collision incidents. In addition, the innovative usage of Exif (Exchangeable image file format) properties to tag and store essential data within the captured images facilitates efficient retrieval of vital information such as vehicle speed, GPS location, and orientation. Incorporating this data, along with road information from the OpenStreetMap API, enhances the accuracy and contextuality of responsibility assessments. However, this approach has some limitations. In addition, the development of the system went through multiple iterations, encountering various difficulties. This chapter first discusses the encountered difficulties and then describes some of the limitations of the current system.

7.2 Difficulties Encountered During System Implementation

During the implementation of the system, we encountered several difficulties. The following summarizes some of them:

- **Changing YOLOv5 Code:** To obtain crash detection results in text format from YOLOv5, we had to modify the detection algorithm of YOLOv5 to output the required data. The original YOLOv5 detection algorithm outputs results directly in a video if the input is a video or directly in an image if the input is an image. However, this format was not suitable for our needs. In a video, we required detailed detection information, including each detection's accuracy, the frame in which each detection was made, and the image containing the bounding box for each detection. Therefore, we modified YOLOv5's original code and added features and methods to extract details about the detection's accuracy, the frame number in the image sequence of the video in which the detection was made, and the exportation of the image in which the crash was detected with the detection bounding box.

The following illustrates the modified code of the detection function, which now returns the results of the detection in addition to the direct output within images or videos:

```
def detect(save_img=False, return_result = False):
[...]
        # Process detections
        for i, det in enumerate(pred): # detections per image
            if webcam: # batch_size >= 1
                p, s, im0 = path[i], '%g: ' % i, im0s[i].copy()
            else:
                p, s, im0 = path, '', im0s

            save_path = str(Path(out) / Path(p).name)
            txt_path = str(Path(out) / Path(p).stem) + ('_%g' %
                dataset.frame if dataset.mode == 'video' else ''
                )
            s += '%gx%g ' % img.shape[2:] # print string
            gn = torch.tensor(im0.shape)[[1, 0, 1, 0]] #
                normalization gain whwh

            # Add the accuracy of the prediction to the list
            data_prediction_results.append(det)
[...]


    if return_result:
        return data_prediction_results

    if save_txt or save_img:
        print('Results saved to %s' % os.getcwd() + os.sep +
            out)
        if platform == 'darwin': # MacOS
            os.system('open ' + save_path)

    print('Done. (%.3fs)' % (time.time() - t0))
```

Listing 7.1: Modified YOLOv5's detection algorithm

- **Collecting Data with Specific Criteria:** To build a custom crash detection model, all images in the dataset had to be related to a head-on/angle crash and had to be from a video recorded by the driving recorder of one of the vehicles involved in the crash. This made data collection meticulous and difficult. YouTube was used as the primary source for various driving recorder crash videos. However, most compilation videos found were not dedicated solely to head-on/angle crashes in the context of crossroad accidents. Many of them were compilations of various car crashes captured by driving recorders of other vehicles, not necessarily the ones involved in the crash. Consequently, we had to watch numerous videos and select the parts containing the kinds of crashes and angles of recording that were of interest to us. This process took several months to complete.

- **Finding a Method Other Than Computer Vision:** The evolution of our system from the first prototype to its current version involved overcoming a significant challenge: finding an alternative method beyond conventional computer vision to address the limitations of the initial model. The primary limitation of the first prototype emerged in scenarios where traffic lights were absent, resulting in a less effective system in crash detection and responsibility assessment. Conventional computer vision heavily relies on visual cues, making it dependent on the presence of identifiable objects such as traffic lights. As we encountered this limitation, it became evident that a paradigm shift was necessary to find a solution that did not solely rely on visual elements. Finding an alternative method that could complement or even replace computer vision led us to the need to think ”out of the box.” We explored alternative fields such as sensor technology, data fusion, and signal processing. The goal was to identify a method that could provide reliable information even in the absence of visual cues like traffic lights. This exploration led to the consideration of sensor-based solutions, lidars, radars, and other non-visual technologies. However, each approach brought its own set of challenges and trade-offs. Finally, after various shifts, the integration of OpenStreetMap emerged as a compelling method.

7.3 Crash Time and Traffic Signs Detection

The experiment results reveal certain limitations of the system, with a notable concern being its sensitivity to light conditions during crashes. Similar to other technologies utilizing computer vision or object detection, variations in lighting present challenges to robust object detection. A potential solution to this challenge involves incorporating sensors such as lidars or radars. Alternatively, training the models with additional data collected under low lighting conditions is a viable approach. Given that many vehicles lack advanced technologies like lidar or radar but are equipped with driving recorders, the latter solution appears more practical for the system.

In addition to lighting conditions, the crash time detection module may occasionally misidentify objects, leading to inaccurate crash time evaluations. For instance, Figure 7.1 illustrates an instance where the module detects the shadow of a vehicle as the collided object. This misclassification occurs because the crash detection model is trained to identify a deformed vehicle as a crash, and any irregular shape of a vehicle in a driving recorder video is classified as a collided object. To address this challenge, exploration of sophisticated methods involving object tracking and the temporal evolution of scenes is planned, aiming to enhance the system's accuracy in crash detection.

Figure 7.1: Example of the detection of the vehicle's shadow as the collided object.

In real-world accidents, unforeseen and exceptional events can occur, surpassing the system's current capabilities. For instance, when a crash video displays both traffic lights for involved drivers simultaneously, the system may become confused, resulting in inaccurate outputs.

Furthermore, the robustness of the system depends on the performance of the collision detection and crash time detection. If either detection fails or makes an erroneous detection, the responsibility evaluation is adversely affected, leading to challenges in accurate assessments. Lastly, the system may face challenges when driving recorders sustain serious damage from impactful collisions. In such scenarios, the system's ability to evaluate responsibilities may be compromised if the recording of at least one second post-collision is unavailable.

7.4 Real-time Video Streaming

The incorporation of real-time video streaming is one of the most key points of a prompt responsibility assessment, providing unparalleled advantages in incident evaluation. By leveraging live video feeds, the system can promptly analyze and assess collision events as they occur. This immediate access to real-time data empowers the system to make fast decisions, potentially reducing response times during critical situations. The real-time video streaming component ensures that the responsibility assessment process is not affected by delays associated with traditional methods. Traditional post-incident analysis methods often rely on recorded footage, which may result in a temporal gap between the occurrence of an incident and its assessment. In contrast, real-time video streaming allows for instantaneous monitoring and evaluation, offering a more dynamic and proactive approach to responsibility assessment.

While real-time video streaming presents a significant strength, it is essential to acknowledge potential challenges such as bandwidth limitations, network latency, and the need for more robust video compression techniques. Addressing these challenges will be crucial to maintaining the efficiency and reliability of the real-time streaming component in diverse operational settings.

7.5 Open Data Usage

The utilization of location data from open sources, such as the OpenStreetMap, provides the system with a rich spatial context for each road crash. The detailed geospatial information, including road layouts, intersections, and geographical features, contributes to a more precise understanding of the environment in which the crash occurred. This enhanced location context is invaluable for assessing responsibilities accurately, especially in complex scenarios such as intersections or areas with specific traffic regulations. Another notable advantage is the ability to access real-time updates to road conditions and infrastructure. This real-time information ensures that the system stays current with the latest changes in the road network, including closures, construction, or modifications to traffic regulations. The dynamic nature of road data from open data sources contributes to the overall accuracy of responsibility assessments.

While the use of open data, particularly the OpenStreetMap API, brings substantial benefits, it is crucial to acknowledge potential limitations. One primary limitation is the system's dependency on the internet. The mobile component of the system necessitates a continuous internet connection to stream videos and transmit data to the server in real-time. In the event of a collision, if the driver's device is not connected to the internet, the server cannot receive the frames associated with the crash, resulting in a failure of responsibility assessment. Furthermore, the system relies on the OpenStreetMap API, which must always be accessible to guarantee uninterrupted responsibility assessment services.

7.6 Exif Properties for Data Tagging

The use of Exif properties plays a key role in enhancing the depth and richness of data available for responsibility assessment within the system. It provides a standardized way to store metadata within images, including crucial information such as timestamps, camera settings, and geospatial details.

By leveraging Exif properties, the responsibility assessment system gains the ability to tag captured images with essential contextual information. This metadata becomes a valuable resource for retrieving specific details related to the incident, such as vehicle speed, GPS location, and orientation. The systematic tagging of images ensures that the responsibility assessment process is not solely reliant on visual analysis but is augmented by a comprehensive set of associated data. The timestamp information embedded through Exif properties allows for chronological ordering of images, aiding in the reconstruction of events leading up to and following a collision incident. This temporal context is invaluable for understanding the sequence of events and contributing factors.

The orientation data captured in conjunction with GPS information and vehicle's speed, provides accurate location-based context. This additional layer of information are important in discerning environmental conditions, road layouts, and potential contributing factors specific to the geographical context of the incident.

However, it's important to note that the effectiveness of Exif properties is dependent on the accuracy of the mobile device's internal clock and GPS system. Discrepancies or inaccuracies in these components may lead to misinterpretations of the temporal and geospatial aspects, impacting the precision of responsibility assessments.

7.7 Responsibility Assessment

One of the key advantages of employing a rule-based knowledge system is the inherent transparency it brings to the responsibility assessment process. The rules governing the inference engine are explicitly defined, following the "if-then" structure, making it clear how conclusions are reached. This transparency is crucial, especially in scenarios where the assessment of responsibility may have legal or insurance implications. Stakeholders, including law enforcement and insurance agencies, can benefit from a clear understanding of how responsibility determinations are made. Our system's rule-based approach allows for the incorporation of domain-specific rules that capture the nuances of responsibility assessment in road crashes. For example, specific traffic regulations, local driving practices, or contextual factors can be easily integrated into the knowledge base. This adaptability ensures that the system can be fine-tuned to different environments, contributing to the accuracy and relevance of responsibility attributions Responsibility assessment in road crashes often involves dealing with uncertainty and ambiguity. The rule-based system allows us to explicitly model this uncertainty within the rules, providing a more nuanced evaluation. For instance, rules can include conditions that account for unclear situations or conflicting evidence. This capability aligns with the real-world complexity of crash scenarios where definitive responsibility determination might be challenging. Comparing our rule-based responsibility assessment with alternative methods, such as machine learning models or statistical approaches, reveals distinct advantages. While machine learning models may offer predictive capabilities, the lack of transparency in their decision-making processes can be a significant draw-back, especially when explaining responsibility attributions is essential. Our rule-based system strikes a balance by providing both accuracy and interpretability.

Despite the strengths of our rule-based responsibility assessment system, it is essential to acknowledge its limitations. The effectiveness of the system heavily relies on the comprehensiveness and accuracy of the defined rules. Incomplete or inaccurate rules may lead to suboptimal results. Future work could focus on refining and expanding the rule set based on continuous learning from real-world crash data.

7.8 Correction Factors

Correction factors play a crucial role in refining responsibility assessments by considering additional contextual elements that might influence the severity or degree of negligence in a collision incident. These factors can encompass a wide range of variables, including driver's fatigue, road maintenance, visibility, and other external influences that may not be explicitly covered by standard road rules. One notable limitation of the system is its reliance on the rule-based knowledge system without the incorporation of correction factors. This absence may lead to oversimplified assessments that do not fully encapsulate the complexity of real-world scenarios. For instance, road maintenance issues could significantly impact the dynamics of a collision, and neglecting these factors might result in a skewed evaluation of responsibilities. Integrating correction factors into the responsibility assessment process would enable a more nuanced and accurate determination of negligence. These factors can be dynamically adjusted based on real-time data sources, such as traffic reports, or surveillance systems, to provide a more comprehensive understanding of the incident context. While the current rule-based approach serves as a solid foundation, future iterations of the responsibility assessment system could benefit greatly from the inclusion of correction factors. This enhancement would align the assessment methodology more closely with the intricacies of real-world driving conditions, thereby improving the system's overall accuracy and reliability in assigning responsibilities.

7.9 Limitation on Crash Scenarios

Crashes at intersections, commonly referred to as road junctions or crossroads, constitute a common type of road collision. According to the National

Cooperative Highway Research Program in the U.S. [1], intersection-related crashes account for over 50% of all collisions in urban areas and more than 30% in rural areas. In 2021, the U.S. Department of Transportation, National Highway Traffic Safety Administration reported [2] that among collisions involving moving motor vehicles, angle collisions and head-on collisions were the most frequent, accounting for 45.46% and 27.18%, respectively. This is in contrast to rear-end collisions (18.66%), sideswipe collisions (7.5%), and other/unknown incidents (1.2%).

The proposed system fully supports head-on (front impact) collisions and angle collisions (left and right-side impacts). The system's usability and relevance in real-world application is highlighted by the fact that these supported collision types collectively account for more than 70% of all collisions, as mentioned earlier. However, the system's ability to handle angle collisions is under the condition that the impact with the other vehicle is visible in the crash video.

The decision to focus on head-on and angle collisions is influenced by the technical specifications of driving recorders, primarily designed to capture the frontal view of vehicles. Consequently, the system may not adequately address scenarios involving side impacts for both vehicles, rear-end collisions, or other non-frontal crash configurations, impacting its comprehensiveness in assessing other types of crash scenario.

While the current system excels in evaluating head-on and angle collisions, expanding its capabilities to cover a broader spectrum of crash scenarios would improve its applicability in real-world situations. Future iterations could explore technological enhancements or multi-camera setups to overcome limitations in handling diverse crash scenarios. By incorporating a wider field of view, the system could extend its support to side impacts, rear-end collisions, and other configurations, ensuring a more comprehensive and inclusive approach to responsibility assessments.

7.10 Crash Explainability

Currently, the system relies on a knowledge-based approach, utilizing predefined road rules to infer the basic degree of negligence. However, there is a potential avenue for further enhancement by incorporating advanced LLM

[1] https://web.archive.org/web/20061003032951/http://safety.transportation.org/doc/1P\%20Unsignalized\%20Intersection\%20Crashes.pdf

[2] https://www.iii.org/table-archive/21904

(Large Language Model) and NLP (Natural Language Processing) techniques to augment the explainability of the assessments. By doing so, the system will gain the ability to generate human-readable explanations that go beyond the constraints of rigid rule-based interpretations. LLM can understand intricate patterns within the data, while NLP enables the conversion of these patterns into coherent and interpretable narratives. Their incorporation will introduce the potential for dynamic and context-aware explanations. Unlike static rules, LLM can adapt to evolving circumstances, providing tailored justifications for each responsibility assignment. NLP can enable the system to articulate nuanced details, considering factors beyond strict rule adherence, such as driver behavior, environmental conditions, and the specific context of the collision.

However, it is important to acknowledge potential challenges in implementing LLM and NLP, including the need for extensive and diverse training data to ensure accurate language generation. Additionally, maintaining the balance between the interpretability of rule-based approaches and the flexibility of language models is crucial to avoid overly complex or ambiguous explanations.

Chapter 8

Conclusion

8.1 Summary of this Study

Many studies have primarily concentrated on crash detection rather than determining fault. In our approach, we initially employed image detection combined with a rule-based system to automatically assign driver responsibilities. However, this method encountered limitations, particularly in scenarios lacking traffic signs. To overcome these challenges, we introduced a novel process and algorithm that incorporates open data, image detection, and a rule-based system. Experimentation yielded promising outcomes, showcasing the system's effectiveness across diverse scenarios, day and night, irrespective of the presence of traffic lights.

8.2 Results and Contributions of this Research

Previous studies have focused on proposing methods for predicting road accidents or detecting traffic incidents in real-time. However, these approaches often have limitations, primarily addressing collision detection or prediction only. This study aims to overcome these limitations by expanding the scope of accident management. It introduces a new algorithm and process designed to automatically assess responsibility, going beyond simple collision scenarios, even in situations without traffic lights.

The proposed method enhances the system's ability to assess responsibilities in a broader range of crash situations, thereby improving its versatility and applicability in real-world scenarios. The entire responsibility assessment process involves four steps: (1) detecting the crash time within a crash video, (2) identifying all traffic lights within the video, (3) obtaining road information from the OpenStreetMap API, such as road width and the presence of other traffic signs if necessary, analyzing and processing the information, and (4) utilizing a rule-based knowledge system of road rules, vehicle speed, and orientation to deduce the probable responsibility of each party involved. The system focuses on head-on and angle crashes involving two cars and facilitates the seamless sharing of evaluation results with the police and insurance companies within minutes of a collision. By employing advanced image processing techniques, the system enables prompt detection and analysis of collision incidents. The integration of open data enhances the contextual understanding of the road environment, contributing to more accurate responsibility assessments by improving the performance of responsibilities evaluation mainly during nighttime with traffic lights. The significant difference and advantage of this system over existing ones is its automation of responsibilities evaluation for the police claims adjusters, and victims themselves. This automation is not only beneficial for accident management but also holds significant implications for the development of autonomous vehicles, driving assistant systems, and other cutting-edge research areas within the automobile and insurance industries.

Additionally, the detection models, image dataset, and video dataset used in implementing the system will be made publicly available to the scientific community. This valuable resource can be utilized by other researchers in the future to develop more advanced systems and contribute to related fields. This study is among the first to enable machines to automatically assess the responsibility of drivers within a crash. It can serve as one of the precursors and foundations for automatic responsibility assessments in autonomous vehicles.

8.3 Future Work

While the system has shown promising outcomes in assigning driver responsibilities using a combination of image detection and a rule-based system, there are several avenues for future work to enhance its capabilities and address identified limitations:

- **Integration of Advanced Machine Learning Techniques:** Explore the integration of advanced machine learning techniques, such as deep learning algorithms, to further improve the accuracy of object detection and scene understanding. This could involve training models on a larger and more diverse dataset to handle complex scenarios with greater efficiency.

- **Enhanced Object Recognition:** Investigate methods to enhance object recognition in scenarios where traditional image detection faces challenges, especially in the absence of clear traffic signs. This may involve exploring alternative computer vision approaches or leveraging additional sensor data, such as radar or lidar, to improve object identification.

- **Dynamic Rule-based System:** Develop a more dynamic and adaptive rule-based system that can evolve based on real-time data and environmental conditions. This would enhance the system's ability to handle diverse scenarios and adapt to changing road conditions or infrastructure.

- **Incorporation of Vehicle-to-Everything (V2X) Communication:** Explore the integration of V2X communication to gather real-time information from other vehicles and infrastructure. This could provide valuable contextual data for decision-making and contribute to a more comprehensive understanding of the traffic environment.

- **Extended Testing in Complex Traffic Scenarios:** Conduct extensive testing in complex traffic scenarios, including intersections with multiple lanes, diverse vehicle types, and intricate road layouts. This will help validate the system's robustness and reliability across a wide range of real-world situations.

- **Usability Studies and User Feedback:** Conduct usability studies involving drivers and stakeholders to gather feedback on the user interface and overall system performance. This user-centric approach can lead to refinements and improvements based on practical user experiences.

- **Legal and Ethical Considerations:** Address legal and ethical considerations associated with automated responsibility assignment. This includes exploring the regulatory landscape, liability implications, and ethical frameworks to ensure responsible deployment of the system.

- **Integration with Autonomous Vehicles:** Investigate the potential integration of the system with autonomous vehicles, considering how it can contribute to decision-making processes and enhance overall traffic safety in mixed traffic environments.

These future directions aim to further strengthen the system's capabilities, adaptability, and reliability in addressing challenges related to crash detection and responsibility assignment in diverse and complex traffic scenarios.

Acknowledgments

This research project would not have been possible without the support of many individuals.

I express my gratitude to my supervisor, Prof. Dr. Tadachika Ozono, who was abundantly helpful and offered invaluable assistance, support, and guidance. Without his knowledge and assistance, this study would not have been successful. Deepest gratitude is also extended to the members of the supervisory committee and to Dr. Masato Kikuchi.

I would like to thank Florence Gundidza, Hydara Ebrima, Ezoa Djangoran, and all other labmates, for their help with the laboratory, research, development, and day-to-day assistance. Warmest thanks also go to my friend and Ph.D. fellow in Network Security, Hiroki Inayoshi.

Gratitude is also directed to NGK Insulators Ltd. and all the staff in the Headquarters, especially Atsuko Sato and Takahiro Moriwaki, for providing the necessary financial support for this work and for being of great help and support.

I convey my thanks to Nagoya Institute of Technology for providing financial support, assistance, and access to laboratory facilities. Special thanks to the International Student Support Center staff members, especially Kaori Nagumo and Yuka Murase for their day-to-day support. Additionally, I would like to express my appreciation to Chihiro Takenaga, Tokuji Takagi, and all the members of the International Family Association, especially Keiko Ishida, Keiko Takeuchi, Ando, Hiroaki Inakuma, for their love and support in everyday life.

I express my love and gratitude to my beloved family for their understanding and endless love throughout the duration of this work. Thanks to my brother Darwin, and my two sisters Erna and Daina, as well as my aunts Monique, and uncles Andre, Maurice, Mignanou, and Boutamekpo, and the rest of the family.

Finally, warmest thanks to my best friends Adefolake Adeniyi, Tsubasa Matayoshi, Keiha Takagi, Moemi Suzumura, and Testu Matsubara who have always been there for me.

Bibliography

[1] World Health Organization. *Global status report on road safety 2018*. Tech. rep. Geneva: World Health Organization, 2018. [Statistics on deaths each year due to road traffic crashes.]

[2] Kateřina Bucsuházy et al. "Human factors contributing to the road traffic accident occurrence". In: *Transportation Research Procedia* 45 (2020), pp. 555–561. ISSN: 2352-1465. DOI: `https://doi.org/10.1016/j.trpro.2020.03.057`. URL: `https://www.sciencedirect.com/science/article/pii/S2352146520302192`. [Description of various factors and circumstances that cause vehicle collisions.]

[3] Davis Buss, Kairatolla Abishev, and Almagul Baltabekova. "Driver ‘s reliability and its effect on road traffic safety". In: *Procedia Computer Science* 149 (2019), pp. 463–466. ISSN: 1877-0509. DOI: `https://doi.org/10.1016/j.procs.2019.01.163`. URL: `https://www.sciencedirect.com/science/article/pii/S1877050919301711`. [Description of various circumstances and situations of vehicle collisions.]

[4] Jianyu Wang et al. "Analyzing the Risk Factors of Traffic Accident Severity Using a Combination of Random Forest and Association Rules". In: *Applied Sciences* 13.14 (2023). ISSN: 2076-3417. DOI: `10.3390/app13148559`. URL: `https://www.mdpi.com/2076-3417/13/14/8559`. [Explores risk factors influencing the at-fault party in traffic accidents and analyzes their impact on traffic accident severity.]

[5] Satyajit Mondal et al. "Identifying the Critical Risk Factors for Road Crashes Based on Large-Scale Safety Audits in India". In: *KSCE Journal of Civil Engineering* 27.11 (2023). TY: JOUR, pp. 4906–4918. ISSN: 1976-3808. DOI: `10.1007/s12205-023-0679-7`. URL: `https://doi.`

org/10.1007/s12205-023-0679-7. [Explores risk factors behind car crashes.]

[6] Hanif Bhuiyan et al. “Crash severity analysis and risk factors identification based on an alternate data source: a case study of developing country”. In: *Scientific Reports* 12 (Dec. 2022). DOI: 10.1038/s41598-022-25361-5. [Reveals that significant features associated with crash severities include driver characteristics, vehicle characteristics, road characteristics, environmental conditions, and injury localization.]

[7] Darcin Akin et al. “Identifying Causes of Traffic Crashes Associated with Driver Behavior Using Supervised Machine Learning Methods: Case of Highway 15 in Saudi Arabia”. In: *Sustainability* 14.24 (2022). ISSN: 2071-1050. DOI: 10.3390/su142416654. URL: https://www.mdpi.com/2071-1050/14/24/16654. [Explores causes and risk factors behind car crashes.]

[8] Fu Wang et al. “Analysis of the Causes of Traffic Accidents and Identification of Accident-Prone Points in Long Downhill Tunnel of Mountain Expressways Based on Data Mining”. In: *Sustainability* 14.14 (2022). ISSN: 2071-1050. DOI: 10.3390/su14148460. URL: https://www.mdpi.com/2071-1050/14/14/8460. [Explores causes and risk factors behind car crashes.]

[9] Soodeh Shahsavari et al. “Analysis of injuries and deaths from road traffic accidents in Iran: bivariate regression approach”. In: *BMC Emergency Medicine* 22.1 (2022). PMID: 35843936, p. 130. ISSN: 1471-227X. DOI: 10.1186/s12873-022-00686-6. URL: https://doi.org/10.1186/s12873-022-00686-6. [Explores causes and risk factors behind car crashes.]

[10] Athiappan Kamalasekar et al. “Identifying Influencing Factors of Road Accidents in Emerging Road Accident Blackspots”. In: *Advances in Civil Engineering* 2022 (Oct. 2022). DOI: 10.1155/2022/9474323. [Explores causes and risk factors behind car crashes.]

[11] Simon Tsala et al. “An In-Depth Analysis of the Causes of Road Accidents in Developing Countries: Case Study of Douala-Dschang Highway in Cameroon”. In: *Journal of Transportation Technologies* 11 (July 2021), pp. 455–470. DOI: 10.4236/jtts.2021.113030. [Explores causes and risk factors behind car crashes.]

[12] Anna Borucka et al. "Predictive analysis of the impact of the time of day on road accidents in Poland". In: *Open Engineering* 11.1 (2021), pp. 142–150. DOI: `doi:10.1515/eng-2021-0017`. URL: `https://doi.org/10.1515/eng-2021-0017`. [Explores causes and risk factors behind car crashes.]

[13] Natalia Casado-Sanz, Begoña Guirao, and Maria Attard. "Analysis of the Risk Factors Affecting the Severity of Traffic Accidents on Spanish Crosstown Roads: The Driver' s Perspective". In: *Sustainability* 12.6 (2020). ISSN: 2071-1050. DOI: `10.3390/su12062237`. URL: `https://www.mdpi.com/2071-1050/12/6/2237`. [Explores causes and risk factors behind car crashes.]

[14] Qiuru Cai. "Cause Analysis of Traffic Accidents on Urban Roads Based on an Improved Association Rule Mining Algorithm". In: *IEEE Access* 8 (2020), pp. 75607–75615. ISSN: 2169-3536. DOI: `10.1109/ACCESS.2020.2988288`. [Explores causes and risk factors behind car crashes.]

[15] Debela Jima. "Review on Factors Causes Road Traffic Accident In Africa". In: *Journal of Civil Engineering Research & Technology* 2 (Dec. 2019), pp. 41–49. DOI: `10.47363/JCERT/2019(1)101`. [Explores causes and risk factors behind car crashes.]

[16] Mounir Belloumi and Fedy Ouni. "Factors Affecting the Severity of Motor Vehicle Traffic Crashes in Tunisia". In: *SAE International Journal of Transportation Safety* 7 (Sept. 2019), pp. 1–30. DOI: `10.4271/09-07-02-0006`. [Explores causes and risk factors behind car crashes.]

[17] Jonathan J. Rolison et al. "What are the factors that contribute to road accidents? An assessment of law enforcement views, ordinary drivers' opinions, and road accident records". In: *Accident Analysis & Prevention* 115 (2018), pp. 11–24. ISSN: 0001-4575. DOI: `https://doi.org/10.1016/j.aap.2018.02.025`. URL: `https://www.sciencedirect.com/science/article/pii/S0001457518300873`. [Explores causes and risk factors behind car crashes.]

[18] A. Ditcharoen et al. "Road traffic accidents severity factors: A review paper". In: *2018 5th International Conference on Business and Industrial Research (ICBIR)*. 2018, pp. 339–343. DOI: `10.1109/ICBIR.2018.8391218`. [Explores road accident likelihood and severity.]

[19] Burcu Oralhan and Ziya Gökalp Göktolga. "Determination of the Risk Factors That Influence Occurrence Time of Traffic Accidents with Survival Analysis". In: *Iranian Journal of Public Health* 47 (2018), pp. 1181–1191. URL: https://api.semanticscholar.org/CorpusID:52166038. [Explores causes and risk factors behind car crashes.]

[20] Chaochun Yuan et al. "Research on Active Collision Avoidance and Hysteresis Reduction of Intelligent Vehicle Based on Multi-Agent Coordinated Control System". In: *World Electric Vehicle Journal* 14.1 (2023). ISSN: 2032-6653. DOI: 10.3390/wevj14010016. URL: https://doi.org/10.3390/wevj14010016. [Uses a multi-agent system as a solution to the vehicle collision avoidance control problem.]

[21] Abu Jafar Md Muzahid et al. "Multiple vehicle cooperation and collision avoidance in automated vehicles: survey and an AI-enabled conceptual framework". In: *Scientific Reports* 13.1 (Jan. 2023), p. 603. ISSN: 2045-2322. DOI: 10.1038/s41598-022-27026-9. URL: https://doi.org/10.1038/s41598-022-27026-9. [Uses a multi-agent system as a solution to the vehicle collision avoidance control problem.]

[22] Kader Sanogo et al. "A multi-agent system simulation based approach for collision avoidance in integrated Job-Shop Scheduling Problem with transportation tasks". In: *Journal of Manufacturing Systems* 68 (2023), pp. 209–226. ISSN: 0278-6125. DOI: https://doi.org/10.1016/j.jmsy.2023.03.011. URL: https://www.sciencedirect.com/science/article/pii/S0278612523000626. [Uses a multi-agent system as a solution to the vehicle collision avoidance control problem.]

[23] Zhigang Xiong, Zhong Liu, and Yasong Luo. "Collision and obstacle avoidance strategy for multi-agent systems with velocity dynamic programing". In: *Measurement and Control* 56.1-2 (2023), pp. 257–268. DOI: 10.1177/00202940221122195. eprint: https://doi.org/10.1177/00202940221122195. URL: https://doi.org/10.1177/00202940221122195. [Uses a multi-agent system as a solution to the obstacle collision avoidance problem.]

[24] Serap Ergün. "A study on multi-agent reinforcement learning for autonomous distribution vehicles". In: *Iran Journal of Computer Science* 6.4 (Dec. 2023), pp. 297–305. ISSN: 2520-8446. DOI: 10.1007/s42044-023-00140-1. URL: https://doi.org/10.1007/s42044-023-00140-

1. [Uses a multi-agent system as a solution to the autonomous delivery vehicles optimization problem.]

[25] Sou Kitajima et al. "Multi-agent traffic simulations to estimate the impact of automated technologies on safety". In: *Traffic Injury Prevention* 20 (June 2019), S58–S64. DOI: 10.1080/15389588.2019.1625335. [Introduce a multi-agent traffic simulation methodology to estimate the potential improvements in road safety resulting from automated vehicle technologies.]

[26] Deepak Mane et al. "Real-Time Vehicle Accident Recognition from Traffic Video Surveillance using YOLOV8 and OpenCV". In: *International Journal on Recent and Innovation Trends in Computing and Communication* 11 (May 2023), pp. 250–258. DOI: 10.17762/ijritcc.v11i5s.6651. [Focused on detecting car crashes in real-time video feeds from traffic monitoring cameras utilizing YOLO.]

[27] Victor A Adewopo et al. "A review on action recognition for accident detection in smart city transportation systems". In: *Journal of Electrical Systems and Information Technology* 10.1 (2023), p. 57. DOI: 10.1186/s43067-023-00124-y. URL: https://doi.org/10.1186/s43067-023-00124-y. [Focused on detecting car crashes in real-time video feeds from traffic monitoring cameras utilizing YOLO.]

[28] Karishma Pawar and Vahida Attar. "Deep learning based detection and localization of road accidents from traffic surveillance videos". In: *ICT Express* 8.3 (2022), pp. 379–387. ISSN: 2405-9595. DOI: https://doi.org/10.1016/j.icte.2021.11.004. URL: https://www.sciencedirect.com/science/article/pii/S2405959521001478. [Focused on detecting car crashes in real-time video feeds from traffic monitoring cameras utilizing YOLO.]

[29] Chaeyoung Lee et al. "A Study on Building a "Real-Time Vehicle Accident and Road Obstacle Notification Model" Using AI CCTV". In: *Applied Sciences* 11.17 (2021). ISSN: 2076-3417. DOI: 10.3390/app11178210. URL: https://www.mdpi.com/2076-3417/11/17/8210. [Focused on detecting car crashes in real-time video feeds from traffic monitoring cameras utilizing YOLO.]

[30] Umesh Parameshwar Naik et al. "Implementation of YOLOv4 Algorithm for Multiple Object Detection in Image and Video Dataset using Deep Learning and Artificial Intelligence for Urban Traffic Video

Surveillance Application”. In: *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. Sept. 2021, pp. 1–6. DOI: 10.1109/ICECCT52121.2021.9616625. [Focused on detecting car crashes in real-time video feeds from traffic monitoring cameras utilizing YOLO.]

[31] Hao-Hsuan Hsu, Nen-Fu Huang, and Chuan-Hsiang Han. “Collision Analysis to Motor Dashcam Videos With YOLO and Mask R-CNN for Auto Insurance”. In: *2020 International Conference on Intelligent Engineering and Management (ICIEM)*. 2020, pp. 311–315. DOI: 10.1109/ICIEM48762.2020.9160263. [Focused on detecting car crashes in real-time video feeds from traffic monitoring cameras utilizing YOLO.]

[32] Daxin Tian et al. “An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems”. In: *IEEE Access* PP (Sept. 2019), pp. 1–1. DOI: 10.1109/ACCESS.2019.2939532. [Focused on detecting car crashes in real-time video feeds from traffic monitoring cameras utilizing YOLO.]

[33] Deeksha Gour and Amit Kanskar. “Automated AI Based Road Traffic Accident Alert System: YOLO Algorithm”. In: *International Journal of Scientific & Technology Research* 8 (2019), pp. 574–578. [Focused on detecting car crashes in real-time video feeds from traffic monitoring cameras utilizing YOLO.]

[34] V. Machaca Arceda and E. Laura Riveros. “Fast car Crash Detection in Video”. In: *2018 XLIV Latin American Computer Conference (CLEI)*. Oct. 2018, pp. 632–637. DOI: 10.1109/CLEI.2018.00081. [Focused on detecting car crashes in real-time video feeds from traffic monitoring cameras utilizing YOLO.]

[35] Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. “Real-Time Accident Detection in Traffic Surveillance Using Deep Learning”. In: *2022 IEEE International Conference on Imaging Systems and Techniques (IST)*. Kaohsiung, Taiwan: IEEE Press, 2022, pp. 1–6. DOI: 10.1109/IST55454.2022.9827736. URL: https://doi.org/10.1109/IST55454.2022.9827736. [Focused on detecting car crashes in real-time video feeds from traffic monitoring cameras.]

[36] Hawzhin Hozhabr Pour et al. “A Machine Learning Framework for Automated Accident Detection Based on Multimodal Sensors in Cars”. In: *Sensors* 22 (2022), p. 3634. ISSN: 1424-8220. DOI: 10.3390/s22103634.

URL: https://www.mdpi.com/1424-8220/22/10/3634. [Focused on detecting car crashes.]

[37] Jae Gyeong Choi et al. "Car crash detection using ensemble deep learning and multimodal data from dashboard cameras". In: *Expert Systems with Applications* 183 (2021), p. 115400. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2021.115400. URL: https://www.sciencedirect.com/science/article/pii/S095741742100823X. [Focused on detecting car crashes in real-time video feeds from dashboard cameras.]

[38] Azzedine Boukerche and Zhijun Hou. "Object Detection Using Deep Learning Methods in Traffic Scenarios". In: *ACM Comput. Surv.* 54.2 (Mar. 2021). ISSN: 0360-0300. DOI: 10.1145/3434398. URL: https://doi.org/10.1145/3434398. [Focused on detecting car crashes.]

[39] Veronica Radu et al. "Car crash detection in videos". In: *2021 23rd International Conference on Control Systems and Computer Science (CSCS)*. 2021, pp. 127–132. DOI: 10.1109/CSCS52396.2021.00028. [Focused on detecting car crashes in videos.]

[40] Chen Wang et al. "A Vision-Based Video Crash Detection Framework for Mixed Traffic Flow Environment Considering Low-Visibility Condition". In: *Journal of Advanced Transportation* 2020 (Jan. 2020), pp. 1–11. DOI: 10.1155/2020/9194028. [Focused on detecting car crashes.]

[41] Yu Yao et al. "Unsupervised Traffic Accident Detection in First-Person Videos". In: *CoRR* abs/1903.00618 (2019). arXiv: 1903.00618. URL: http://arxiv.org/abs/1903.00618. [Focused on detecting car crashes in videos.]

[42] Shivani Sharma and Shoney Sebastian. "IoT based car accident detection and notification algorithm for general road accidents". In: *International Journal of Electrical and Computer Engineering (IJECE)* 9 (Oct. 2019), p. 4020. DOI: 10.11591/ijece.v9i5.pp4020-4026. [Focused on detecting car crashes using IoT.]

[43] Waqas Sultani, Chen Chen, and Mubarak Shah. "Real-world Anomaly Detection in Surveillance Videos". In: *CoRR* abs/1801.04264 (2018). arXiv: 1801.04264. URL: http://arxiv.org/abs/1801.04264. [Focused on detecting car crashes.]

[44] Harit Sharma, Ravi Kanth Reddy, and Archana Karthik. "S-CarCrash: Real-time crash detection analysis and emergency alert using smartphone". In: *2016 International Conference on Connected Vehicles and Expo (ICCVE)*. Sept. 2016, pp. 36–42. DOI: 10.1109/ICCVE.2016.7. [Focused on detecting car crashes.]

[45] Alpamis Kutlimuratov et al. "Applying Enhanced Real-Time Monitoring and Counting Method for Effective Traffic Management in Tashkent". In: *Sensors* 23.11 (2023). ISSN: 1424-8220. DOI: 10.3390/s23115007. URL: https://www.mdpi.com/1424-8220/23/11/5007. [Focused on vehicle identification and detection.]

[46] Vibhanshu Singh Sindhu. "Vehicle Identification from Traffic Video Surveillance Using YOLOv4". In: *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*. May 2021, pp. 1768–1775. DOI: 10.1109/ICICCS51141.2021.9432144. [Focused on vehicle identification and detection.]

[47] Mohammed Imran Basheer Ahmed et al. "A Real-Time Computer Vision Based Approach to Detection and Classification of Traffic Incidents". In: *Big Data and Cognitive Computing* 7.1 (2023). ISSN: 2504-2289. DOI: 10.3390/bdcc7010022. URL: https://www.mdpi.com/2504-2289/7/1/22. [Focused on classification of traffic incidents.]

[48] Chengyu Hu et al. "An image-based crash risk prediction model using visual attention mapping and a deep convolutional neural network". In: *Journal of Transportation Safety & Security* 15.1 (2023), pp. 1–23. DOI: 10.1080/19439962.2021.2015731. eprint: https://doi.org/10.1080/19439962.2021.2015731. URL: https://doi.org/10.1080/19439962.2021.2015731. [Focused on crash risk prediction.]

[49] Kakoli Banerjee et al. "Traffic Accident Risk Prediction Using Machine Learning". In: *2022 International Mobile and Embedded Technology Conference (MECON)*. 2022, pp. 76–82. DOI: 10.1109/MECON53876.2022.9752273. [Focused on crash risk prediction.]

[50] Zhenyu Luo et al. "A new traffic accident risk prediction method based on adaptive neural fuzzy inference system". In: *2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT)*. 2021, pp. 354–358. DOI: 10.1109/ICESIT53460.2021.9696905. [Focused on crash risk prediction.]

[51] Pei Li, Mohamed Abdel-Aty, and Jinghui Yuan. “Real-time crash risk prediction on arterials based on LSTM-CNN”. In: *Accident Analysis & Prevention* 135 (2020), p. 105371. ISSN: 0001-4575. DOI: `https://doi.org/10.1016/j.aap.2019.105371`. URL: `https://www.sciencedirect.com/science/article/pii/S0001457519311108`. [Focused on crash risk prediction.]

[52] Lee Fawcett et al. “A novel Bayesian hierarchical model for road safety hotspot prediction”. In: *Accident Analysis & Prevention* 99 (2017), pp. 262–271. ISSN: 0001-4575. DOI: `https://doi.org/10.1016/j.aap.2016.11.021`. URL: `https://www.sciencedirect.com/science/article/pii/S0001457516304341`. [Focused on crash risk prediction.]

[53] Sachin Kumar and Durga Toshniwal. “A data mining approach to characterize road accident locations”. In: *Journal of Modern Transportation* 24 (2016), pp. 62–72. [Focused on crash risk prediction.]

[54] Seong-hun Park, Sung-min Kim, and Young-guk Ha. “Highway traffic accident prediction using VDS big data analysis”. In: *The Journal of Supercomputing* 72 (July 2016). DOI: `10.1007/s11227-016-1624-z`. [Focused on crash risk prediction.]

[55] Lei Lin, Qian Wang, and Adel W. Sadek. “A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction”. In: *Transportation Research Part C: Emerging Technologies* 55 (2015), pp. 444–459. ISSN: 0968-090X. DOI: `https://doi.org/10.1016/j.trc.2015.03.015`. URL: `https://www.sciencedirect.com/science/article/pii/S0968090X15000947`. [Focused on crash risk prediction.]

[56] Qi Shi and Mohamed Abdel-Aty. “Big Data applications in real-time traffic operation and safety monitoring and improvement on urban expressways”. In: *Transportation Research Part C: Emerging Technologies* 58 (2015), pp. 380–394. ISSN: 0968-090X. DOI: `https://doi.org/10.1016/j.trc.2015.02.022`. URL: `https://www.sciencedirect.com/science/article/pii/S0968090X15000777`. [Focused on crash risk prediction.]

[57] Wentao Bao, Qi Yu, and Yu Kong. “Uncertainty-based Traffic Accident Anticipation with Spatio-Temporal Relational Learning”. In: *CoRR*

abs/2008.00334 (2020). arXiv: `2008.00334`. URL: `https://arxiv.org/abs/2008.00334`. [Focused on crash anticipation.]

[58] Tomoyuki Suzuki et al. "Anticipating Traffic Accidents with Adaptive Loss and Large-scale Incident DB". In: *CoRR* abs/1804.02675 (2018). arXiv: `1804.02675`. URL: `http://arxiv.org/abs/1804.02675`. [Focused on crash anticipation.]

[59] Fu-Hsiang Chan et al. "Anticipating Accidents in Dashcam Videos". In: *Computer Vision – ACCV 2016*. Cham: Springer International Publishing, 2017, pp. 136–153. ISBN: 978-3-319-54190-7. [Focused on crash anticipation.]

[60] Jinming You, Junhua Wang, and Jingqiu Guo. "Real-time crash prediction on freeways using data mining and emerging techniques". In: *Journal of Modern Transportation* 25 (2017), pp. 116–123. [Focused on crash anticipation.]

[61] Yiming Gu, Sean Qian, and Feng Chen. "From Twitter to detector: Real-time traffic incident detection using social media data". In: *Transportation Research Part C: Emerging Technologies* 67 (June 2016), pp. 321–342. DOI: `10.1016/j.trc.2016.02.011`. [Focused on crash anticipation.]

[62] Eleonora D'Andrea et al. "Real-Time Detection of Traffic From Twitter Stream Analysis". In: *IEEE Transactions on Intelligent Transportation Systems* 16 (Aug. 2015), pp. 1–15. DOI: `10.1109/TITS.2015.2404431`. [Focused on crash anticipation.]

[63] Igor Dirnbach et al. "Methodology Designed to Evaluate Accidents at Intersection Crossings with Respect to Forensic Purposes and Transport Sustainability". In: *Sustainability* 12 (Mar. 2020), p. 1972. DOI: `10.3390/su12051972`. [Proposes a new technical and analytical approach for handling expert reports on car crashes at intersections, specifically focusing on traffic light scenarios.]

[64] Almudena Sanjurjo-de-No et al. "Driver Liability Assessment in Vehicle Collisions in Spain". eng. In: *International Journal of Environmental Research and Public Health* 18.4 (Feb. 4, 2021), p. 1475. ISSN: 1660-4601 (Electronic), 1661-7827 (Print). DOI: `10.3390/ijerph18041475`. [One of the few studies focusing on evaluating responsibility after vehicle collisions.]

[65] Cédric Garcia et al. "Prediction of responsibility for drivers and riders involved in injury road crashes". In: *Journal of Safety Research* 70 (July 2019), pp. 9–21. DOI: 10.1016/j.jsr.2019.07.001. [One of the few studies focusing on evaluating responsibility after vehicle collisions.]

[66] Susantha Chandraratna and Nikiforos Stamatiadis. "Quasi-induced exposure method: Evaluation of not-at-fault assumption". In: *Accident Analysis & Prevention* 41.2 (2009), pp. 308–313. ISSN: 0001-4575. DOI: https://doi.org/10.1016/j.aap.2008.12.005. URL: https://www.sciencedirect.com/science/article/pii/S000145750800239X. [One of the few studies focusing on evaluating responsibility after vehicle collisions.]

[67] Ziwen Chen et al. "YOLOv5-Based Vehicle Detection Method for High-Resolution UAV Images". In: *Mob. Inf. Syst.* 2022 (Jan. 2022). ISSN: 1574-017X. DOI: 10.1155/2022/1828848. URL: https://doi.org/10.1155/2022/1828848. [Implemented a vehicle detection model.]

[68] Xudong Dong, Shuai Yan, and Chaoqun Duan. "A lightweight vehicles detection network model based on YOLOv5". In: *Engineering Applications of Artificial Intelligence* 113 (2022), p. 104914. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2022.104914. URL: https://www.sciencedirect.com/science/article/pii/S0952197622001415. [Implemented a vehicle detection model.]

[69] Xuyang Song and Wei Gu. "Multi-objective real-time vehicle detection method based on yolov5". In: *2021 International Symposium on Artificial Intelligence and its Application on Media (ISAIAM)*. May 2021, pp. 142–145. DOI: 10.1109/ISAIAM53259.2021.00037. [Implemented a vehicle detection model.]

[70] Zhuoning Yuan, Xun Zhou, and Tianbao Yang. "Hetero-ConvLSTM: A Deep Learning Approach to Traffic Accident Prediction on Heterogeneous Spatio-Temporal Data". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '18. London, United Kingdom: Association for Computing Machinery, 2018, pp. 984–992. ISBN: 9781450355520. DOI: 10.1145/3219819.3219922. URL: https://doi.org/10.1145/3219819.3219922. [Focused on traffic accident prediction.]

[71] Athanasios Theofilatos. "Incorporating real-time traffic and weather data to explore road accident likelihood and severity in urban arterials". In: *Journal of Safety Research* 61 (Mar. 2017). DOI: 10.1016/j.jsr.2017.02.003. [Explores road accident likelihood and severity.]

[72] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2014. DOI: 10.48550/ARXIV.1405.0312. URL: https://arxiv.org/abs/1405.0312.

[73] Joseph Redmon et al. *You Only Look Once: Unified, Real-Time Object Detection*. 2015. DOI: 10.48550/ARXIV.1506.02640. URL: https://arxiv.org/abs/1506.02640. [Description of one of the most famous object detection algorithms due to its speed and accuracy.]

[74] Daria Snegireva and Anastasiia Perkova. "Traffic Sign Recognition Application Using Yolov5 Architecture". In: *2021 International Russian Automation Conference (RusAutoCon)*. Sept. 2021, pp. 1002–1007. DOI: 10.1109/RusAutoCon52004.2021.9537355. [Used YOLOv5 to detect traffic signals.]

[75] Wentao Liu et al. "Real-time Signal Light Detection based on Yolov5 for Railway". In: *IOP Conference Series: Earth and Environmental Science* 769.4 (May 2021), p. 042069. DOI: 10.1088/1755-1315/769/4/042069. URL: https://doi.org/10.1088/1755-1315/769/4/042069.

[76] Jamuna Murthy et al. "ObjectDetect: A Real-Time Object Detection Framework for Advanced Driver Assistant Systems Using YOLOv5". In: *Wireless Communications and Mobile Computing* 2022 (June 2022), pp. 1–10. DOI: 10.1155/2022/9444360. [Used YOLOv5 to detect obstacles.]

[77] Fujie Sun, Zhuoshen Li, and Zhuolin Li. "A traffic flow detection system based on YOLOv5". In: *2021 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*. Oct. 2021, pp. 458–464. DOI: 10.1109/AINIT54228.2021.00095. [Used YOLOv5 to detect traffic flow.]

[78] Daria Snegireva and Georgiy Kataev. "Vehicle Classification Application on Video Using Yolov5 Architecture". In: *2021 International Russian Automation Conference (RusAutoCon)*. Sept. 2021, pp. 1008–1013. DOI: 10.1109/RusAutoCon52004.2021.9537439. [Used YOLOv5 to classify vehicles.]

[79] Helton Agbewonou Yawovi, Tadachika Ozono, and Toramatsu Shintani. "Crossroad Accident Responsibility Prediction Based on a Multi-agent System". In: *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2020, pp. 579–584. DOI: 10.1109/CSCI51800.2020.00103. [Implemented a service that can be used by the system to request the detection of a crash in a given video using the model inference.]

[80] Helton Agbewonou Yawovi, Masato Kikuchi, and Tadachi Ozono. "Who Was Wrong? An Object Detection Based Responsibility Assessment System for Crossroad Vehicle Collisions". In: *AI* 3 (2022), pp. 844–862. ISSN: 2673-2688. DOI: 10.3390/ai3040051. URL: https://www.mdpi.com/2673-2688/3/4/51. [Implemented a system based mainly on image processing and a rule-based knowledge system. The system uses the crash video recorded by and taken from a real driving recorder of one of the vehicles involved in the crash as the input data source.]

[81] Helton Agbewonou Yawovi, Masato Kikuchi, and Tadachika Ozono. "AiDashcam: A Vehicle Collision Responsibility Evaluation System Based on Object Detection and OpenStreetMap". In: *Advances in Systems Engineering*. Cham: Springer Nature Switzerland, 2023, pp. 12–21. ISBN: 978-3-031-40579-2. DOI: 10.1007/978-3-031-40579-2_2. [Implemented a system based mainly on image processing, open data and a rule-based knowledge system.]

[82] Jang-Hee Yoo, Byoung-Ho Kang, and Jong-Uk Choi. "A Hybrid approach to auto-insurance claim processing system". In: vol. 1. Oct. 1994, 537–542 vol.1. ISBN: 0-7803-2129-4. DOI: 10.1109/ICSMC.1994.399894. [One of the few studies focusing on evaluating responsibility after vehicle collisions.]

Research achievements

Journal (Peer-reviewed)

1. Yawovi Helton Agbewonou, Masato Kikuchi, and Tadachika Ozono: "Who Was Wrong? An Object Detection Based Responsibility Assessment System for Crossroad Vehicle Collisions", AI 3, no. 4, 2022, pp.844-862, doi https://doi.org/10.3390/ai3040051

International conferences (Peer-reviewed)

1. Yawovi Helton Agbewonou, Masato Kikuchi, and Tadachika Ozono: "AiDashcam: A Vehicle Collision Responsibility Evaluation System Based on Object Detection and OpenStreetMap". Selvaraj, H., Chmaj, G., Zydek, D. (eds) Advances in Systems Engineering. ICSEng 2023. Lecture Notes in Networks and Systems, vol 761. Springer, Cham, doi https://doi.org/10.1007/978-3-031-40579-2_2

2. Yawovi, Helton Agbewonou, Tadachika Ozono, and Toramatsu Shintani: "Crossroad Accident Responsibility Prediction Based on a Multiagent System", 2020 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2020, pp. 579-584, doi: https://doi.org/10.1109/CSCI51800.2020.00103.

Domestic conferences

1. Helton Agbewonou Yawovi, Masato Kikuchi, Tadachika Ozono: "Responsibility Assessment in Crossroad Accident Using Object Recognition and Knowledge System", Technical Committee on Artificial Intelligence and Knowledge-Based Processing(SIG-AI), 2022, pp.59-64.

2. YAWOVI Agbewonou Helton, OZONO Tadachika, SHINTANI Toramatsu: “Implementing an automatic and instant road accident report system using Knowledge System”, FIT2019, Vol. 2, F-039, 2019, pp. 353–354.

3. YAWOVI Agbewonou Helton, OZONO Tadachika, SHINTANI Toramatsu: “Implementing an Automatic Road Accident Report System with an Accident Simulator”, IPSJ SIG Technical Reports, Vol.2020-ICS-197, 2020, No.9, 4p.

4. YAWOVI Agbewonou Helton, OZONO Tadachika, SHINTANI Toramatsu: “Realizing an automatic responsibilities prediction system for road accident using 3D simulation and knowledge systems”, The proc. of JSAI2020, No. 3F1ES203, 2020, 3p. DOI:10.11517/pjsai.JSAI2020.0_3F1ES203

Others

1. Symposium of Multi Agent Systems for Harmonization 2020, Student Encouragement Award, 2020.

Appendix A

List of Road Rules of the Knowledge Base System

Let T, P, R be traffic light, crash spot, and road respectively, X_A Vehicle A, X_B Vehicle B, S_A the speed of Vehicle A, S_B the speed of Vehicle B, N the degree of negligence, the predicates $is\_intersection$ to check if a crash spot is an intersection or not,$is\_t\_junction$ to check if a crash spot is a T-junction or not, $is\_priority\_road$ to check if a road is a priority road or not, is_small_road to check if a road is a small road or not, $is\_one\_way\_road$ to check if a road is a one-way road or not, $road\_width$ to get the width of a road, is_green to check if a traffic light is green or not, is_yellow to check if a traffic light is yellow or not, is_red to check if a traffic light is red or not, $is\_flashing\_red$ to check if a traffic light is flashing-red or not, $is\_flashing\_yellow$ to check if a traffic light is flashing-yellow or not, $is\_right\_turn\_green$ to check if a traffic light is right-turn-green or not, has_sign to check if there is a traffic sign or not, using the domain $SN = \{STOP, NONE\}$, $speed\_limit$ to check if there is a speed limit sign or not, $turn$ to check if a vehicle turns right or left or not, using the domain $DIR = \{LEFT, RIGHT, NONE\}$, $direction$ to get the side a vehicle is coming from, using the domain $SD = \{LEFT, RIGHT\}$, $orientation$ to get the orientation of a vehicle.

Rule 1:

$is\_intersection(P) \wedge is\_green(X_A, T) \wedge is\_red(X_B, T)$

$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 0 \wedge N(X_B) = 100)$

Rule 2:
$is\_intersection(P) \wedge is\_yellow(X_A, T) \wedge is\_red(X_B, T)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 3:
$is\_intersection(P) \wedge is\_red(X_A, T) \wedge is\_red(X_B, T)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 50 \wedge N(X_B) = 50)$

Rule 4:
$is\_intersection(P) \wedge is\_green(X_A, T) \wedge is\_yellow(X_B, T)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 30 \wedge N(X_B) = 70)$

Rule 5:
$is\_intersection(P) \wedge (S_A = S_B) \wedge direction(X_B, \text{RIGHT})$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 40 \wedge N(X_B) = 60)$

Rule 6:
$is\_intersection(P) \wedge (S_A > S_B) \wedge direction(X_B, \text{RIGHT})$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 60 \wedge N(X_B) = 40)$

Rule 7:
$is\_intersection(P) \wedge (S_A < S_B) \wedge direction(X_B, \text{RIGHT})$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$

$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 8:
$is\_intersection(P) \wedge is\_one\_way\_road(X_B, R)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 9:
$is\_intersection(P) \wedge (road\_width(X_A, R) > road\_width(X_B, R)) \wedge (S_A = S_B)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 30 \wedge N(X_B) = 70)$

Rule 10:
$is\_intersection(P) \wedge (road\_width(X_A, R) > road\_width(X_B, R)) \wedge (S_A > S_B)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 40 \wedge N(X_B) = 60)$

Rule 11:
$is\_intersection(P) \wedge (road\_width(X_A, R) > road\_width(X_B, R)) \wedge (S_A < S_B)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 12:
$is\_intersection(P) \wedge has\_sign(X_A, STOP) \wedge (S_A = S_B)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 13:
$is\_intersection(P) \wedge has\_sign(X_A, STOP) \wedge (S_A > S_B)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 30 \wedge N(X_B) = 70)$

Rule 14:
$is\_intersection(P) \wedge has\_sign(X_A, STOP) \wedge (S_A < S_B)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 10 \wedge N(X_B) = 90)$

Rule 15:
$is\_intersection(P) \wedge has\_sign(X_A, STOP) \wedge (S_B <= 30)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 40 \wedge N(X_B) = 60)$

Rule 16:
$is\_intersection(P) \wedge is\_flashing\_red(X_A, T) \wedge (S_B = S_A)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 17:
$is\_intersection(P) \wedge is\_flashing\_red(X_A, T) \wedge (S_B > S_A)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 30 \wedge N(X_B) = 70)$

Rule 18:
$is\_intersection(P) \wedge is\_flashing\_red(X_A, T) \wedge (S_B < S_A)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 10 \wedge N(X_B) = 90)$

Rule 19:
$is\_intersection(P) \wedge is\_flashing\_red(X_A, T) \wedge (S_B <= 30)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 40 \wedge N(X_B) = 60)$

Rule 20:
$is\_intersection(P) \wedge is\_flashing\_yellow(X_A, T) \wedge (S_B = S_A)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 21:
$is\_intersection(P) \wedge is\_flashing\_yellow(X_A, T) \wedge (S_B > S_A)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 30 \wedge N(X_B) = 70)$

Rule 22:
$is\_intersection(P) \wedge is\_flashing\_yellow(X_A, T) \wedge (S_B < S_A)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 10 \wedge N(X_B) = 90)$

Rule 23:
$is\_intersection(P) \wedge is\_flashing\_yellow(X_A, T) \wedge (S_B <= 30)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 40 \wedge N(X_B) = 60)$

Rule 24:
$is\_intersection(P) \wedge is\_priority\_road(X_A, R)$
$\wedge\ has\_sign(X_A, NONE) \wedge has\_sign(X_B, NONE)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 10 \wedge N(X_B) = 90)$

Rule 25:
$is\_intersection(P) \wedge is\_green(X_A, T) \wedge is\_green(X_B, T)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 26:
$is\_intersection(P) \wedge is\_yellow(X_A, T) \wedge is\_green(X_B, T)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 70 \wedge N(X_B) = 30)$

Rule 27:
$is\_intersection(P) \wedge is\_yellow(X_A, T) \wedge is\_yellow(X_B, T)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 40 \wedge N(X_B) = 60)$

Rule 28:
$is\_intersection(P) \wedge is\_red(X_A, T) \wedge is\_red(X_B, T)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 50 \wedge N(X_B) = 50)$

Rule 29:
$is\_intersection(P) \wedge is\_red(X_A, T) \wedge is\_green(X_B, T)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 90 \wedge N(X_B) = 10)$

Rule 30:
$is\_intersection(P) \wedge is\_red(X_A, T) \wedge is\_yellow(X_B, T)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 70 \wedge N(X_B) = 30)$

Rule 31:
$is\_intersection(P) \wedge is\_red(X_A, T) \wedge is\_right\_turn\_green(X_B, T)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 100 \wedge N(X_B) = 0)$

Rule 32:

$is\_intersection(P) \wedge (road\_width(X_A, R) = road\_width(X_B, R))$
$\wedge (orientation(X_A) = orientation(X_B))$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 33:
$is\_intersection(P) \wedge (road\_width(X_A, R) = road\_width(X_B, R))$
$\wedge (orientation(X_A) != orientation(X_B)) \wedge direction(X_B, \text{LEFT})$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 40 \wedge N(X_B) = 60)$

Rule 34:
$is\_intersection(P) \wedge (road\_width(X_A, R) = road\_width(X_B, R))$
$\wedge (orientation(X_A) != orientation(X_B)) \wedge direction(X_B, \text{RIGHT})$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 30 \wedge N(X_B) = 70)$

Rule 35:
$is\_intersection(P) \wedge has\_sign(X_A, NONE) \wedge (road\_width(X_A, R) > road\_width(X_B, R))$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 36:
$is\_intersection(P) \wedge has\_sign(X_A, NONE) \wedge (road\_width(X_A, R) < road\_width(X_B, R))$
$\wedge\, direction(X_B, \text{LEFT}) \wedge turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 60 \wedge N(X_B) = 40)$

Rule 37:
$is\_intersection(P) \wedge has\_sign(X_A, NONE) \wedge (road\_width(X_A, R) < road\_width(X_B, R))$
$\wedge\, direction(X_B, \text{RIGHT}) \wedge turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 50 \wedge N(X_B) = 50)$

Rule 38:
$is\_intersection(P) \wedge has\_sign(X_A, STOP) \wedge (road\_width(X_A, R) = road\_width(X_B, R))$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 15 \wedge N(X_B) = 85)$

Rule 39:
$is\_intersection(P) \wedge has\_sign(X_A, STOP) \wedge (road\_width(X_A, R) = road\_width(X_B, R))$
$\wedge\ direction(X_B, \text{LEFT}) \wedge turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 70 \wedge N(X_B) = 30)$

Rule 40:
$is\_intersection(P) \wedge has\_sign(X_A, STOP) \wedge (road\_width(X_A, R) = road\_width(X_B, R))$
$\wedge\ direction(X_B, \text{RIGHT}) \wedge turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 60 \wedge N(X_B) = 40)$

Rule 41:
$is\_intersection(P) \wedge is\_priority\_road(X_A, R)$
$\wedge\ has\_sign(X_A, NONE) \wedge has\_sign(X_B, NONE)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 10 \wedge N(X_B) = 90)$

Rule 42:
$is\_intersection(P) \wedge is\_priority\_road(X_A, R)$
$\wedge\ has\_sign(X_A, NONE) \wedge has\_sign(X_B, NONE)$
$\wedge\ direction(X_B, \text{LEFT}) \wedge turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 80 \wedge N(X_B) = 20)$

Rule 43:
$is\_intersection(P) \wedge is\_priority\_road(X_A, R)$
$\wedge\ has\_sign(X_A, NONE) \wedge has\_sign(X_B, NONE)$

$\wedge\, direction(X_B, \mathrm{RIGHT}) \wedge turn(X_A, \mathrm{NONE}) \wedge turn(X_B, \mathrm{RIGHT})$
$\implies (N(X_A) = 70 \wedge N(X_B) = 30)$

Rule 44:
$is\_intersection(P) \wedge has\_sign(X_A, NONE) \wedge has\_sign(X_B, NONE)$
$\wedge\, (road\_width(X_A, R) = road\_width(X_B, R)) \wedge turn(X_A, \mathrm{LEFT}) \wedge turn(X_B, \mathrm{NONE})$
$\implies (N(X_A) = 50 \wedge N(X_B) = 50)$

Rule 45:
$is\_intersection(P) \wedge has\_sign(X_A, NONE) \wedge has\_sign(X_B, NONE)$
$\wedge\, (road\_width(X_A, R) < road\_width(X_B, R)) \wedge turn(X_A, \mathrm{LEFT}) \wedge turn(X_B, \mathrm{NONE})$
$\implies (N(X_A) = 70 \wedge N(X_B) = 30)$

Rule 46:
$is\_intersection(P) \wedge has\_sign(X_A, STOP)$
$\wedge\, turn(X_A, \mathrm{LEFT}) \wedge turn(X_B, \mathrm{NONE})$
$\implies (N(X_A) = 80 \wedge N(X_B) = 20)$

Rule 47:
$is\_intersection(P) \wedge is\_priority\_road(X_A, R) \wedge has\_sign(X_A, NONE)$
$\wedge\, turn(X_A, \mathrm{LEFT}) \wedge turn(X_B, \mathrm{NONE})$
$\implies (N(X_A) = 90 \wedge N(X_B) = 10)$

Rule 48:
$is\_t\_junction(P) \wedge has\_sign(X_A, NONE)$
$\wedge\, (road\_width(X_A, R) = road\_width(X_B, R))$
$\wedge\, turn(X_A, \mathrm{NONE}) \wedge turn(X_B, \mathrm{LEFT})$
$\implies (N(X_A) = 30 \wedge N(X_B) = 70)$

Rule 49:
$is\_t\_junction(P) \wedge has\_sign(X_A, NONE)$

$\wedge (road\_width(X_A, R) = road\_width(X_B, R))$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 30 \wedge N(X_B) = 70)$

Rule 50:
$is\_t\_junction(P) \wedge has\_sign(X_A, NONE)$
$\wedge (road\_width(X_A, R) > road\_width(X_B, R))$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{LEFT})$
$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 51:
$is\_t\_junction(P) \wedge has\_sign(X_A, NONE)$
$\wedge (road\_width(X_A, R) > road\_width(X_B, R))$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 52:
$is\_t\_junction(P) \wedge has\_sign(X_A, STOP)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{LEFT})$
$\implies (N(X_A) = 15 \wedge N(X_B) = 85)$

Rule 53:
$is\_t\_junction(P) \wedge has\_sign(X_A, STOP)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 15 \wedge N(X_B) = 85)$

Rule 54:
$is\_t\_junction(P) \wedge is\_priority\_road(X_A, R) \wedge has\_sign(X_A, NONE)$
$\wedge\, turn(X_A, \text{NONE}) \wedge turn(X_B, \text{LEFT})$
$\implies (N(X_A) = 10 \wedge N(X_B) = 90)$

Rule 55:
$is\_t\_junction(P) \wedge is\_priority\_road(X_A, R) \wedge has\_sign(X_A, NONE)$
$\wedge\ turn(X_A, \text{NONE}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 10 \wedge N(X_B) = 90)$

Rule 56:
$is\_t\_junction(P) \wedge has\_sign(X_A, NONE)$
$\wedge\ (road\_width(X_A, R) = road\_width(X_B, R))$
$\wedge\ turn(X_A, \text{RIGHT}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 40 \wedge N(X_B) = 60)$

Rule 57:
$is\_t\_junction(P) \wedge has\_sign(X_A, NONE)$
$\wedge\ (road\_width(X_A, R) > road\_width(X_B, R))$
$\wedge\ turn(X_A, \text{RIGHT}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 30 \wedge N(X_B) = 70)$

Rule 58:
$is\_t\_junction(P) \wedge has\_sign(X_A, STOP)$
$\wedge\ turn(X_A, \text{RIGHT}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 25 \wedge N(X_B) = 75)$

Rule 59:
$is\_t\_junction(P) \wedge is\_priority\_road(X_A, R) \wedge has\_sign(X_A, NONE)$
$\wedge\ turn(X_A, \text{RIGHT}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 60:
$is\_intersection(P) \wedge has\_sign(X_A, NONE)$
$\wedge\ (road\_width(X_A, R) = road\_width(X_B, R))$
$\wedge\ turn(X_A, \text{RIGHT}) \wedge turn(X_B, \text{RIGHT})$

$\implies (N(X_A) = 40 \wedge N(X_B) = 60)$

Rule 61:
$is\_intersection(P) \wedge has\_sign(X_A, NONE)$
$\wedge\, (road\_width(X_A, R) < road\_width(X_B, R))$
$\wedge\, turn(X_A, \text{RIGHT}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 70 \wedge N(X_B) = 30)$

Rule 62:
$is\_intersection(P) \wedge has\_sign(X_A, STOP)$
$\wedge\, turn(X_A, \text{RIGHT}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 75 \wedge N(X_B) = 25)$

Rule 63:
$is\_intersection(P) \wedge is\_priority\_road(X_A, R) \wedge has\_sign(X_A, NONE)$
$\wedge\, turn(X_A, \text{RIGHT}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 80 \wedge N(X_B) = 20)$

Rule 64:
$is\_intersection(P) \wedge has\_sign(X_A, NONE)$
$\wedge\, turn(X_A, \text{LEFT}) \wedge turn(X_B, \text{RIGHT})$
$\implies (N(X_A) = 30 \wedge N(X_B) = 70)$

Rule 65:
$is\_intersection(P) \wedge is\_green(X_A, T) \wedge is\_red(X_B, T)$
$\wedge\, turn(X_A, \text{RIGHT}) \wedge turn(X_B, \text{NONE})$
$\implies (N(X_A) = 0 \wedge N(X_B) = 100)$

Rule 66:
$is\_intersection(P) \wedge is\_yellow(X_A, T) \wedge is\_red(X_B, T)$
$\wedge\, turn(X_A, \text{RIGHT}) \wedge turn(X_B, \text{NONE})$

$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 67:
$\mathit{is\_intersection}(P) \wedge \mathit{is\_red}(X_A, T) \wedge \mathit{is\_red}(X_B, T)$
$\wedge\ \mathit{turn}(X_A, \mathrm{RIGHT}) \wedge \mathit{turn}(X_B, \mathrm{NONE})$
$\implies (N(X_A) = 50 \wedge N(X_B) = 50)$

Rule 68:
$\mathit{is\_intersection}(P) \wedge \mathit{is\_green}(X_A, T) \wedge \mathit{is\_yellow}(X_B, T)$
$\wedge\ \mathit{turn}(X_A, \mathrm{RIGHT}) \wedge \mathit{turn}(X_B, \mathrm{NONE})$
$\implies (N(X_A) = 30 \wedge N(X_B) = 70)$

Rule 69:
$\mathit{is\_intersection}(P) \wedge (S_A = S_B) \wedge \mathit{direction}(X_B, \mathrm{RIGHT})$
$\wedge\ \mathit{turn}(X_A, \mathrm{RIGHT}) \wedge \mathit{turn}(X_B, \mathrm{NONE})$
$\implies (N(X_A) = 40 \wedge N(X_B) = 60)$

Rule 70:
$\mathit{is\_intersection}(P) \wedge (S_A > S_B) \wedge \mathit{direction}(X_B, \mathrm{RIGHT})$
$\wedge\ \mathit{turn}(X_A, \mathrm{RIGHT}) \wedge \mathit{turn}(X_B, \mathrm{NONE})$
$\implies (N(X_A) = 60 \wedge N(X_B) = 40)$

Rule 71:
$\mathit{is\_intersection}(P) \wedge (S_A < S_B) \wedge \mathit{direction}(X_B, \mathrm{RIGHT})$
$\wedge\ \mathit{turn}(X_A, \mathrm{RIGHT}) \wedge \mathit{turn}(X_B, \mathrm{NONE})$
$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 72:
$\mathit{is\_intersection}(P) \wedge \mathit{is\_one\_way\_road}(X_B, R)$
$\wedge\ \mathit{turn}(X_A, \mathrm{RIGHT}) \wedge \mathit{turn}(X_B, \mathrm{NONE})$
$\implies (N(X_A) = 20 \wedge N(X_B) = 80)$

Rule 73:

$is\_intersection(P) \land (road\_width(X_A, R) > road\_width(X_B, R)) \land (S_A = S_B)$
$\land\, turn(X_A, \text{RIGHT}) \land turn(X_B, \text{NONE})$
$\implies (N(X_A) = 30 \land N(X_B) = 70)$

Rule 74:

$is\_intersection(P) \land (road\_width(X_A, R) > road\_width(X_B, R)) \land (S_A > S_B)$
$\land\, turn(X_A, \text{RIGHT}) \land turn(X_B, \text{NONE})$
$\implies (N(X_A) = 40 \land N(X_B) = 60)$

Rule 75:

$is\_intersection(P) \land (road\_width(X_A, R) > road\_width(X_B, R)) \land (S_A < S_B)$
$\land\, turn(X_A, \text{RIGHT}) \land turn(X_B, \text{NONE})$
$\implies (N(X_A) = 20 \land N(X_B) = 80)$

Rule 76:

$is\_intersection(P) \land has\_sign(X_A, STOP) \land (S_A = S_B)$
$\land\, turn(X_A, \text{RIGHT}) \land turn(X_B, \text{NONE})$
$\implies (N(X_A) = 20 \land N(X_B) = 80)$