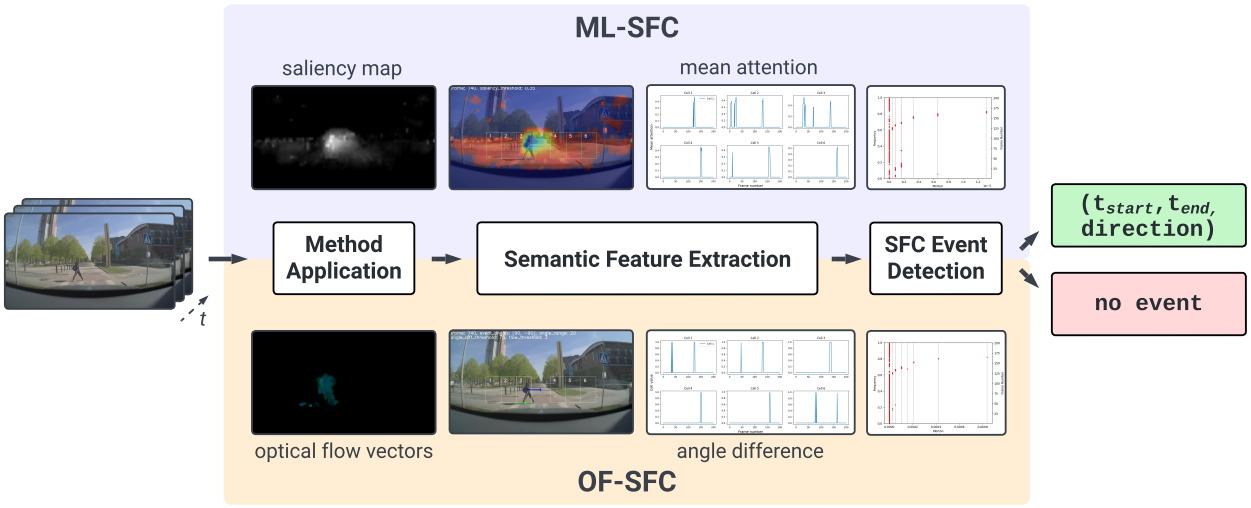
CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG



Space-Filling Curve-Based Traffic Event Detection Using Deep Learning and Optical Flow

A Conceptual Framework for Efficient Traffic Event Detection in Vehicle-Mounted Video Data

Master's thesis in Computer science and engineering

ERIK WESSMAN
ELIAS KJELLBERG CARLSON

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2024

Master's thesis 2024

Space-Filling Curve-Based Traffic Event Detection Using Deep Learning and Optical Flow

A Conceptual Framework for Efficient Traffic Event Detection in Vehicle-Mounted Video Data

ERIK WESSMAN
ELIAS KJELLBERG CARLSON

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2024

Space-Filling Curve-Based Traffic Event Detection Using Deep Learning and Optical Flow
A Conceptual Framework for Efficient Traffic Event Detection in Vehicle-Mounted Video Data
ERIK WESSMAN
ELIAS KJELLBERG CARLSON

Supervisor: Tayssir Bouraffa, Department of Computer Science and Engineering
Industrial Supervisor: Ali Nouri, Volvo Cars AB
Advisor: Christian Berger, Department of Computer Science and Engineering
Examiner: Hans-Martin Heyn, Department of Computer Science and Engineering

Master's Thesis 2024
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A visual representation of the framework developed to detect events in video data. Features are extracted, filtered, and converted to Morton codes, and an event is detected. The framework is described further in Chapter 4.

Typeset in LaTeX
Gothenburg, Sweden 2024

Space-Filling Curve-Based Traffic Event Detection Using Deep Learning and Optical Flow
A Conceptual Framework for Efficient Traffic Event Detection in Vehicle-Mounted Video Data
ERIK WESSMAN
ELIAS KJELLBERG CARLSON
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Identifying and analyzing traffic events in large-scale, unstructured video data from vehicle-mounted cameras is a significant challenge for enhancing advanced driver assistance systems (ADAS). This thesis presents a conceptual framework that leverages machine learning (ML) and optical flow (OF) for efficient traffic event detection, utilizing space-filling curves (SFCs) to reduce data dimensionality. Our first approach, ML-SFC, uses an ML model predicting human attention to identify events, while the second, OF-SFC, employs an OF algorithm to detect movement. Both methods are evaluated using the synthetic SMIRK dataset and validated on the real-world Zenseact Open Dataset (ZOD). The results show that OF-SFC performs better on the synthetic dataset, while ML-SFC is better on the real-world dataset. Both methods achieve comparable processing speeds, indicating their suitability for real-time applications. This framework could serve as a foundation for scalable solutions to analyze large volumes of unstructured data in the form of traffic event detection or other contexts.

The source code for our framework is available here:
https://github.com/erikwessman/ted-sfc.

Keywords: Computer science, software engineering, vehicle safety, neural networks, optical flow, space-filling curves, vehicle event detection

Acknowledgements

We would first like to thank our supervisor, Tayssir Bouraffa, for her guidance, advice, and collaboration throughout our thesis. We also thank our industrial supervisor, Ali Nouri, and Volvo Cars AB for providing an automotive industry perspective and welcoming us to their offices. Additionally, we thank Christian Berger for his genuine interest, valuable advice, and feedback on our work.

Thanks to Zenseact AB and AI Sweden for providing us with the data necessary for conducting this thesis.

Erik Wessman, Gothenburg, May 2024
Elias Kjellberg Carlson, Gothenburg, May 2024

Contents

List of Acronyms **xi**

List of Figures **xiii**

List of Tables **xv**

1 Introduction **1**
1.1 Problem Statement . . . 2
1.2 Purpose of the Study . . . 3
1.3 Research Questions . . . 3
1.4 Scope and Limitations . . . 4

2 Theory **5**
2.1 Neural Networks . . . 5
2.1.1 Convolutional Neural Networks . . . 5
2.1.2 Encoder-Decoder Networks . . . 6
2.1.3 Transformers . . . 6
2.2 Optical Flow . . . 7
2.2.1 Farnebäck . . . 7
2.2.2 Lucas-Kanade . . . 7
2.3 Space-Filling Curves . . . 7
2.3.1 Z-order Curve . . . 8
2.3.2 Hilbert Curve . . . 9
2.4 Evaluation Metrics . . . 10

3 Related Work **11**
3.1 Traffic Video Event Detection . . . 11
3.1.1 Machine Learning . . . 11
3.1.2 Optical Flow . . . 12
3.2 Saliency Prediction . . . 13
3.3 Event Retrieval in Large-Scale Datasets . . . 13

4 Methods **17**
4.1 Selection of Methods and Resources . . . 18
4.1.1 Machine Learning Models . . . 18
4.1.2 Optical Flow Algorithm . . . 19

4.1.3 Datasets and Event . . . 19
4.2 Semantic Feature Extraction . . . 21
4.2.1 Region of Interest Grid . . . 21
4.2.2 ML-enabled Saliency Prediction . . . 23
4.2.3 Optical Flow . . . 24
4.3 Dimensionality Reduction with Space-Filling Curves . . . 25
4.4 Computationally Efficient Event Retrieval . . . 27
4.5 Answering RQ1 . . . 29
4.6 Experiment Setup . . . 29
4.6.1 Ground Truth Annotation . . . 29
4.6.1.1 Zenseact Open Dataset . . . 30
4.6.1.2 SMIRK . . . 30
4.6.2 Metrics . . . 31

5 Results and Discussion **33**
5.1 Description of Evaluation Datasets . . . 33
5.2 Evaluation on Synthetic Dataset (SMIRK) . . . 34
5.3 Evaluation on Real-World Dataset (ZOD) . . . 35
5.3.1 Challenges in Analyzing Real-World Traffic Footage . . . 35
5.4 Comparison of ML-SFC and OF-SFC . . . 38
5.5 Implications for Software Engineering . . . 39
5.6 Threats to Validity . . . 40
5.6.1 Internal Threats to Validity . . . 40
5.6.2 External Threats to Validity . . . 40
5.6.3 Construct Threats to Validity . . . 40
5.6.4 Conclusion Threats to Validity . . . 40
5.7 Future Work . . . 41
5.7.1 Detection of Additional Events . . . 41
5.7.2 Exploring Different Models and Techniques . . . 41
5.7.3 Improved Grid . . . 41

6 Conclusion **43**

Bibliography **45**

A Computer Specifications **I**

B ZOD Test Sets Descriptions **III**

C Full Experiment Results **VII**

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AD	autonomous driving
ADAS	advanced driver assistance systems
CNN	convolutional neural network
CSP	characteristic stripe pattern
DSR	design science research
FPS	frames per second
IMU	inertial measurement unit
IoU	intersection over union
ML	machine learning
NN	neural network
OF	optical flow
RoI	region of interest
SFC	space-filling curve
VED	video event detection

List of Figures

2.1 A simple neural network (NN) with an input layer with two nodes, one hidden layer with five nodes, and an output layer with one node. [11], CC BY 1.0 6

2.2 Visualizations of three types of SFCs traversing a two-dimensional space at the third iteration, illustrating different degrees of data locality preservation. [19] 8

2.3 The first four iterations of a Z-order curve in a two-dimensional space. [21], CC BY-SA 3.0 9

2.4 The first four iterations of a Hilbert curve in a two-dimensional space. [22] 9

4.1 An overview of the traffic video event detection (VED) framework presented in this thesis. Semantic features are extracted for each frame in a video — using machine learning (ML) or optical flow (OF) — and are then filtered and converted to Morton codes using space-filling curves (SFCs). Then, event detection is performed to output either an event window or indicate that no event is present. The original frames in the figure are from ZOD [43]. 17

4.2 Example output of the two variants of our framework applying the region of interest (RoI) grid on a video of a pedestrian crossing from ZOD [43]. Approach (a) computes a saliency map predicting human attention, while approach (b) computes disturbances in the vector field compared to the generally expected optical flow (OF). The original image is taken from ZOD [43]. 22

4.3 Mean saliency cell values extracted using MLNET on a video of a pedestrian crossing from ZOD. 23

4.4 A visualization of the previous frames used for detecting sudden changes in the optical flow (OF). The effect of the flow vector in frame f_{t-7} is dampened due to the use of a mean flow vector, and the change in the mean flow in frames f_t, f_{t-1}, f_{t-2} is detected since it is compared to the previous mean flow. 25

4.5 A characteristic stripe pattern (CSP) from a video in ZOD containing a pedestrian crossing. The grey stripes denote the Morton codes with several being grouped in the same range near 0. The red dots indicate which frame the Morton codes are connected to. The horizontal band spanning from frame 130 to 159 is the event window where the crossing occurred. 26
4.6 A visualization showing the Morton code ranges corresponding to each cell ID in the region of interest (RoI) grid. 28

5.1 A comparison between optical flow on a scene in which the ego-vehicle is turning (a) and standing still while a cyclist is crossing (b). The original image is taken from ZOD [43]. 37
5.2 A comparison between saliency prediction with a bright (a) and dim (b) scene. Due to low visibility, the saliency model loses track of the person on the bike once they enter the shade. The original image is taken from ZOD [43]. 37
5.3 An example of a busy scene. The saliency model struggles to capture the pedestrian in cell 3 due to the number of visually prominent objects in the scene. Original footage from ZOD [43]. 38

List of Tables

4.1 Ranges of Morton codes mapped to cell IDs. These values were used to identify if a Morton code belongs to the range of any of the cells. . 27

5.1 Overview of the dataset partitions created for SMIRK and ZOD. . . . 33
5.2 Results from event retrieval using our framework on the synthetic dataset SMIRK. Three techniques were compared, one using optical flow (OF) (Farnebäck$^{\text{OF}}$) and two using machine learning (ML) (MLNET$^{\text{ML}}$ and TranSalNet$^{\text{ML}}$). Evaluation performed with PC_2, see Table A.1. Note that TASED-net was not applied to this dataset due to technical limitations. 34
5.3 Results from event retrieval using our framework on the real-world ZOD dataset. Four different variants were compared, optical flow (OF) (Farnebäck$^{\text{OF}}$) and three using machine learning (ML) (MLNET$^{\text{ML}}$, TranSalNet$^{\text{ML}}$, and TASED-Net$^{\text{ML}}$). Evaluation performed with PC_2, see Table A.1. 36

A.1 Computer hardware specifications used for experiments. I

B.1 Descriptions for the videos used in ZOD_{1-4} containing pedestrians crossings. III
B.2 Descriptions for the videos used in ZOD_1 containing no pedestrian crossings. IV
B.3 Descriptions for the videos used in ZOD_2 containing no pedestrian crossings. IV
B.4 Descriptions for the videos used in ZOD_3 containing no pedestrian crossings. V
B.5 Descriptions for the videos used in ZOD_4 containing no pedestrian crossings. V

C.1 The complete results from evaluating the detector on SMIRK. Note that TASED-net was not applied to this dataset due to technical limitations. VII
C.2 The complete results from evaluating the detector on ZOD. VIII

1

Introduction

Approximately 1.19 million people die annually in traffic accidents worldwide, with an additional 20-50 million sustaining injuries [1]. While improvements in vehicle safety to reduce these deaths and injuries have traditionally focused on passive, physical features such as seatbelts or crumple zones, technological advancements have increased the focus on digital, active safety systems. These advanced driver assistance systems (ADAS), including technologies like blind spot monitoring, forward collision warnings, and adaptive cruise control, actively assist drivers in preventing accidents, thereby decreasing the impact of human error and increasing overall traffic safety.

To augment drivers' reactions to common and unusual driving scenarios, ADAS leverages sophisticated technologies to detect traffic events such as traffic violations, sudden vehicle maneuvers, pedestrian crossings, and potential collisions. These systems increasingly depend on perception sensors — including LiDAR, inertial measurement units (IMUs), radar, cameras, and sonar — that gather extensive data from the vehicle's internal and external environment. To interpret this data and predict and adeptly respond to dynamic driving scenarios, machine learning (ML) models play a crucial role.

Perception sensors capture data high in volume and complexity, and vehicles equipped with these sensors can collect up to hundreds of megabytes of data per second. Companies within the automotive industry, such as Volvo Cars, accumulate vast quantities of unstructured, unlabelled video data [2]. This data may contain unusual driving patterns, events involving pedestrians, and near-miss accidents that could provide valuable information for advancing ADAS. However, leveraging this information often involves manually identifying and labeling events in the data, which is an expensive and time-consuming process [3]. Hence, there is a need for a more efficient approach to detect and retrieve traffic scenarios from unstructured data.

In collaboration with Volvo Cars [4], this study aims to contribute to traffic safety research by employing ML models and optical flow (OF) alongside computationally efficient approaches for traffic event detection. This effort aligns with Volvo's long-standing commitment to safety and innovation, providing potential options for the future real-world application of our research. Furthermore, the techniques developed in this study are not only of interest to the automotive industry but are also applicable to other industries dealing with large amounts of unlabeled, unstructured data.

This study explores a novel approach that involves extracting semantically relevant visual features from unstructured data, encoding those features for computational efficiency, and using the encoded features to detect traffic events. The feature extraction is done using two distinct methods, one by modeling human attention using ML models and one by computing the movement of objects in the scene using OF. The next phase involves encoding and exploring the extracted features using space-filling curves (SFCs) to enable efficient event detection by reducing the data's dimensionality, following the work by Berger and Birkemeyer [3].

To design our approach, we initially focus on identifying semantically relevant features through systematic experimentation using a synthetic dataset. After identifying these features, we validate our findings by applying the same approach to a real-world dataset, providing a practical context to confirm our observations and evaluate the effectiveness of the proposed concept.

The main contribution of this thesis is a conceptual framework for efficiently detecting and retrieving traffic events from large volumes of unstructured video data. By bridging the gap between video data processing and SFCs, we enhance the efficiency of event detection and enable a potential reduction in reliance on expensive sensor technology.

1.1 Problem Statement

Existing traffic event detection for ADAS primarily utilizes a combination of data from advanced sensors such as LiDAR, IMUs, radar, and arrays of cameras. While radar and LiDAR are less susceptible to environmental factors and offer reliable detection, their perception range is limited. Moreover, these sensors require specialized and costly equipment. On the other hand, video cameras, particularly dashcams, are cost-effective in comparison and present a unique opportunity for exploring event identification using just one of the various available perception sensors. To facilitate this, techniques must be developed to enable video event detection (VED) using vehicle-mounted video footage.

Detecting events by naively processing the video data is ineffective due to the sheer volume and complexity. One way to mitigate this is by leveraging ML models that can efficiently extract relevant features from large quantities of data. These models have been successfully applied in various image and video processing applications [5]–[7] and extensive research efforts have been conducted to apply ML models for VED in traffic scenarios. Despite these advancements, including studies by Bao, Yu, and Kong [8] and Yao, Xu, Wang, *et al.* [9], such models remain computationally demanding and their effectiveness is often constrained by the quality and quantity of their training data [3], [10]. Furthermore, existing ML VED models are typically designed to detect a single type of event, such as accidents [8], [9], meaning a model would have to be re-trained for each event type it's applied to. This would require significant time, computational resources, and datasets with annotations for the event.

While SFCs have enabled computationally efficient detection of automotive maneuvers

for structured data of vehicle acceleration [3], their application to unstructured video data remains unexplored. Applying SFCs to video could enhance the detail and accuracy of event detectors [3], but to implement it, the issue of extracting semantically relevant features from the video data on which the SFCs can be applied must be solved.

1.2 Purpose of the Study

This project explores the feasibility of using SFCs to efficiently detect events in forward-facing dashcam video data. Specifically, the goal is to first extract visual features from video footage using ML models and OF and then feed those features to a SFCs to reduce the dimensionality of the data and enable efficient event lookups. We will evaluate this approach's computational efficiency and performance, focusing on key metrics such as F1 score, sensitivity, specificity, mean intersection over union (IoU), and processing time in frames per second (FPS).

Ultimately, this study aims to provide a proof-of-concept of how to leverage SFCs for computational efficiency when working with video data. This work can benefit industries with large volumes of unstructured data, such as the automotive industry, by allowing efficient event detection. For the wider software engineering discipline, similar approaches could be used to extract relevant information and identify patterns in various forms of unstructured data, such as text documents or audio files, in various contexts.

Lastly, this work will outline the advantages and potential constraints of using SFCs for event retrieval from unstructured data and open opportunities to explore this technique further.

1.3 Research Questions

The aims of this study are defined in the form of two research questions. They are formulated as one open-ended and one close-ended question, focusing on the feasibility of using video data in combination with SFCs for traffic event retrieval, as suggested by Berger and Birkemeyer [3].

RQ1: How can machine learning models and optical flow be utilized to extract semantically relevant visual features from video data, with the objective of integrating these features into space-filling curves?

RQ2: What is the efficacy and efficiency of using space-filling curves to detect traffic events when applied to features extracted from video data?

RQ1 will be answered by developing a method to extract semantically relevant, spatiotemporal features from video data. This method will be developed by exploring existing ML-based VED models, as well as other potential solutions such as OF algorithms.

RQ2 will be answered by evaluating the approach on multiple datasets using F1 score, sensitivity, specificity, and mean IoU to measure the efficacy and the processing speed in FPS to measure the efficiency.

1.4 Scope and Limitations

Considering the study's engagement within the interrelated domains of ML, OF, and SFCs, we establish several limitations to maintain the research's feasibility.

First, the study will leverage existing, open-source ML models, concentrating the research on investigating the application of these models in combination with SFCs rather than the design of new models.

Second, new dashcam datasets will not be created due to time and privacy concerns. Instead, the study will rely on existing, anonymized real-world and synthetic datasets, ensuring ethical compliance and feasibility.

Lastly, to narrow the scope, the study will only focus on detecting one type of event: pedestrians and bikes crossing the road from left to right or right to left. However, the work aims to allow later extensions to other traffic events, like cut-in or merge scenarios.

2

Theory

This chapter covers the concepts and theories relevant to the study. First, neural networks (NNs) are explained along with relevant architectures used in the ML models employed in this study. Second, the technique of OF and a few algorithms are introduced and briefly described. Third, SFCs and their benefits for computation efficiency are explained. Lastly, the metrics used for evaluation in the study are defined along with what they measure.

2.1 Neural Networks

Neural networks (NNs) are sophisticated machine-learning models inspired by the structure and function of neurons in the human brain. These models consist of one or multiple layers of nodes (neurons) where each layer connects to the ones before and after, creating a network like the one in Fig. 2.1. The initial layer's nodes determine the network's input shape, and the final layer determines the shape of the output. Each node in the network has a weight and activation function that determine its output when it receives input from other nodes. The nodes' weights are gradually adjusted during training, typically using a backpropagation algorithm. At the end of the training, the final model is defined by the network's composition of layers and nodes, and the nodes' unique weight values.

NNs can be trained using several methods, of which two principal ones are supervised and unsupervised learning. Supervised learning relies on labeled datasets that pair each input with a correct output. In contrast, unsupervised learning uses unlabelled datasets and aims to identify inherent patterns and structures, as seen in clustering and anomaly detection tasks where specific outcomes are not predefined.

The architecture used to design a NN significantly impacts its behavior and performance. The remainder of this section focuses on three architectures used by models employed in this study: convolutional neural networks (CNNs), encoder-decoder networks, and transformers.

2.1.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a class of deep learning models well suited for image processing tasks such as object detection, facial recognition, and medical analysis [12], [13]. Inspired by the visual cortex of animals, CNNs utilizes

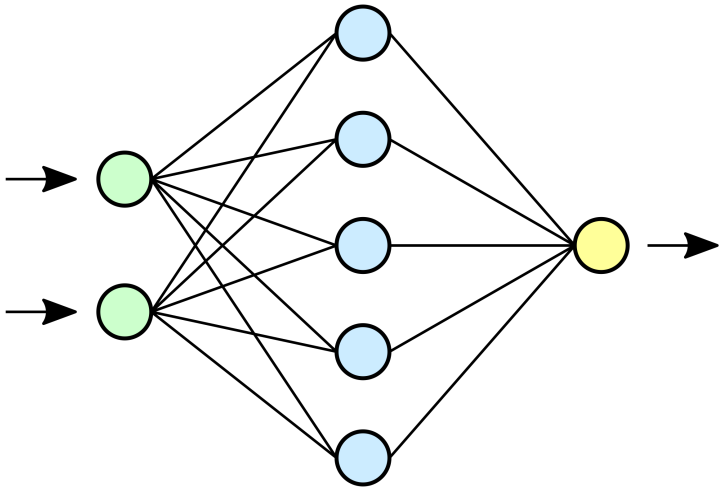


Figure 2.1: A simple NN with an input layer with two nodes, one hidden layer with five nodes, and an output layer with one node. [11], CC BY 1.0

a hierarchical structure comprising convolutional layers that detect patterns such as edges, textures, and shapes. These layers apply learnable filters across an input image, resulting in feature maps highlighting relevant features. Following convolutional layers, pooling layers reduce spatial dimensions while preserving important information. A series of convolutional and pooling layers can be stacked to learn increasingly complex patterns and perform tasks such as image classification [14].

2.1.2 Encoder-Decoder Networks

Encoder-decoder networks are a type of NNs designed to transform complex input sequences into structured output sequences. These networks have many applications, including machine translation, text summarization, and image captioning. The distinguishing characteristic of the encoder-decoder architecture is its use of two connected NNs, one for encoding and one for decoding. The encoding compresses the data into a latent representation of the input — a context vector — reducing the complexity while retaining the essential features of the input. This context vector is then processed in the decoding stage to produce an output.

2.1.3 Transformers

Transformers are a type of NNs designed to process sequential data using a self-attention mechanism instead of recurrence. Introduced by Vaswani, Shazeer, Parmar, *et al.* [15], they consist of an encoder-decoder structure with self-attention layers in both the encoder and decoder to allow the model to consider all tokens in the input simultaneously, instead of one-by-one like traditional models that use recurrence. This ability to see the whole context helps transformers perform well in tasks like translation, summarization, and question-answering. Transformers have played a large part in the recent rise and development of large-language models (LLMs), powering services like ChatGPT [16].

2.2 Optical Flow

Optical flow (OF) is a computer vision technique that estimates the motion of objects or camera movement from a sequence of consecutive frames. It captures the apparent motion of pixels by observing the camera's or scene objects' relative movement. Therefore, OF can represent a video as a set of motion vectors that describe the displacement of each pixel from one frame to the next. This is useful for video stabilization, motion detection, and object-tracking tasks.

2.2.1 Farnebäck

In 2003, Gunnar Farnebäck developed an algorithm for estimating the OF between two consecutive frames, henceforth referred to as Farnebäck OF [17]. It approximates a small region around each pixel using a quadratic equation 2.1, which essentially means fitting a curved surface to the pixel values.

$$f(x) \approx x^T Ax + b^T x + c \tag{2.1}$$

In this equation, x represents the pixel position, A is a matrix describing the surface's curvature, b is a vector describing the surface's slope, and c is a constant value. By observing how this surface changes between the two images, Farnebäck OF estimates how each pixel moves from the first image to the second. The algorithm calculates estimates at a lower resolution and then gradually refines the results at higher resolutions. This approach helps handle larger movements effectively [17].

2.2.2 Lucas-Kanade

Proposed in 1981, the Lucas-Kanade algorithm estimates OF between two consecutive frames by assuming that small neighboring regions move uniformly. It relies on the assumption that the intensity of a pixel remains constant as it moves between frames. For each pixel, the algorithm uses the intensity gradients in a local neighborhood to create a system of linear equations, which are then solved using least squares to find the displacement vector that best describes the pixel movement [18].

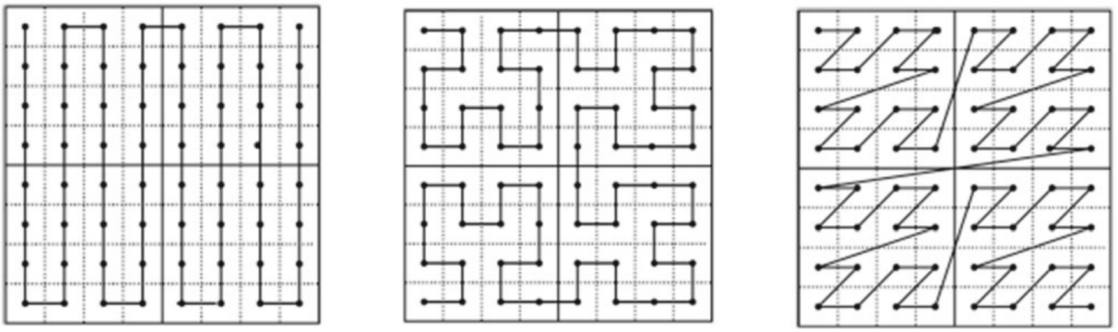
2.3 Space-Filling Curves

Space-filling curves (SFCs) provides a way to represent multi-dimensional values in a single dimension by creating a curve that explores every point in a multi-dimensional space without crossing itself. Doing so, each point in the multi-dimensional input space is mapped to a point on the single-dimensional SFC. By leveraging this behavior, it is possible to computationally efficiently explore large quantities of data [3].

Numerous types of SFCs exist, differentiated by the pattern with which they traverse their input space. The pattern used by a curve influences how the multi-dimensional data is mapped onto the SFC and, crucially, affects the data locality preservation.

Preserving the locality means keeping points that are in close proximity in the multi-dimensional space close to each other on the single-dimensional SFC, maintaining the structure and relations between points in the input data.

A two-dimensional visualization showing three different curve types is presented in Figure 2.2. As can be seen, the zigzag curve does not preserve locality effectively, unlike the Hilbert and Z-order curves, observable by the first and second points in the first row being adjacent on the grid but far apart on the resulting SFC.



(a) Zigzag curve. Low locality preservation.

(b) Hilbert curve. High locality preservation.

(c) Z-order curve. Moderate locality preservation.

Figure 2.2: Visualizations of three types of SFCs traversing a two-dimensional space at the third iteration, illustrating different degrees of data locality preservation. [19]

While Hilbert curves have superior locality preservation over Z-order curves, both types have been shown to perform similarly in querying applications of automotive sensor data, with Z-order curves having the benefit of a quicker encoding algorithm [20].

2.3.1 Z-order Curve

Studied in 1904 by Henri Lebesgue, the Z-order (or Lebesgue) curve consists of a series of Z-shaped lines, each Z connecting to the next and forming a larger structure, as seen in Fig. 2.3. This design maintains high locality preservation for the points within each Z-shape. The numerical representation of a Z-order curve is often called a Morton code, named after computer scientist Guy Macdonald Morton.

One drawback of the Z-order curve is that the transitions between clusters of Z's lead to sections of reduced preservation, most apparent horizontally across the middle of the fourth iteration (bottom right) in Fig. 2.3. Overall, however, the Z-order curve still retains a comparatively high level of preservation.

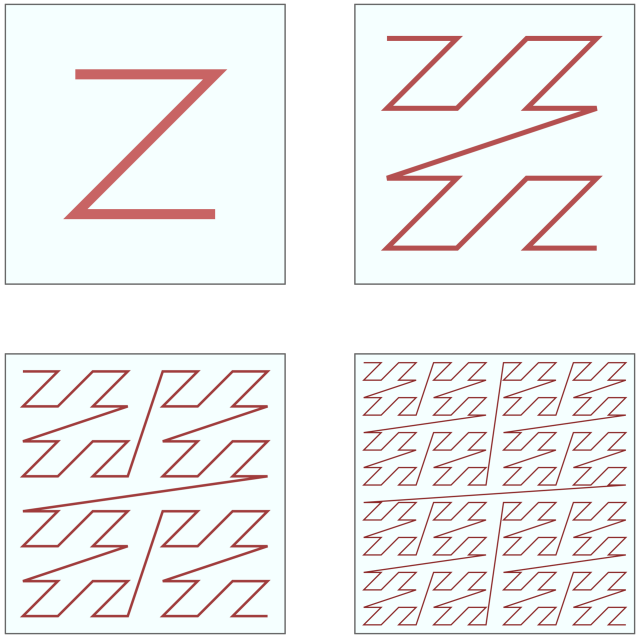


Figure 2.3: The first four iterations of a Z-order curve in a two-dimensional space. [21], CC BY-SA 3.0

2.3.2 Hilbert Curve

First described by mathematician David Hilbert in 1891, the Hilbert curve uses a pattern similar to that of a Peano curve, and the first four iterations in a two-dimensional space can be seen in Fig. 2.4.

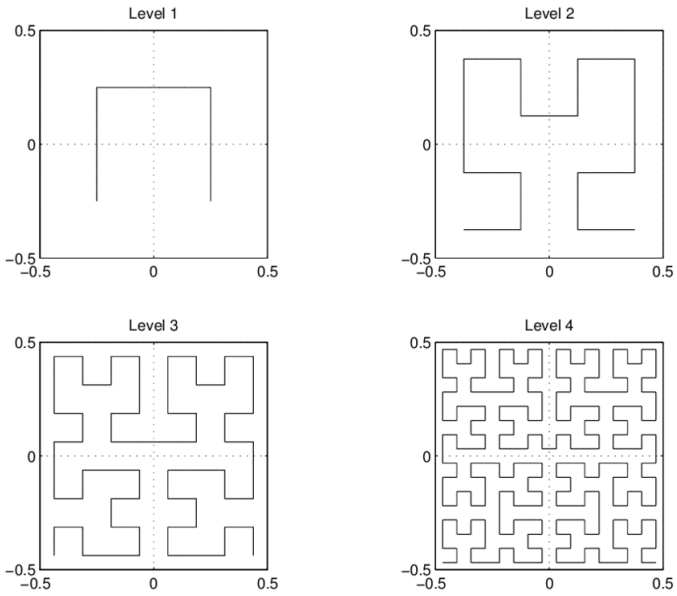


Figure 2.4: The first four iterations of a Hilbert curve in a two-dimensional space. [22]

This pattern is known for having high locality preservation and avoids the intermittent, lengthy jumps in representation found in Z-order curves. Thanks to their high locality preservation, Hilbert curves are commonly used in computer science applications.

2.4 Evaluation Metrics

This section defines and briefly describes the metrics used to evaluate efficacy and computational efficiency in this study.

1. **F1 Score**: The F1 Score is a harmonic mean of the model's precision and recall (sensitivity), providing a single score to balance false positives and false negatives.

$$\text{F1 Score} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}$$

2. **Sensitivity**: Sensitivity measures the proportion of correctly identified positives, indicating the ability to detect relevant events.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

3. **Specificity**: Specificity assesses the ability to correctly identify negatives, providing a measure of the effectiveness in avoiding false positives.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

4. **Intersection over Union**: The IoU measures the overlap between two ranges, such as two timespans, as a ratio of their combined range. Using IoU gives information about how well a predicted range matches the ground truth range.

$$\text{IoU} = \frac{\text{Overlap of frames}}{\text{Union of frames}}$$

5. **Frames per Second (FPS)**: The rate at which video frames are processed, indicating the method's efficiency in real-time contexts.

3

Related Work

This chapter reviews existing research relevant to this study, specifically focusing on traffic event detection using video footage, ML-based saliency prediction, and scalable event retrieval. First, previous approaches to traffic VED are examined, focusing on methods using ML and OF. Second, the field of saliency prediction using ML is explored. Lastly, various techniques for enabling event retrieval in scalable datasets are described.

3.1 Traffic Video Event Detection

Detecting events in traffic scenarios involves the identification of specific occurrences such as accidents, lane changes, emergency brakings, or pedestrian crossings. Various approaches have been developed that rely on different input sources, including cameras, LiDAR, ultrasonic sensors, and radar [23], each with its advantages and disadvantages, particularly regarding the price-to-accuracy trade-off and the complexity of the data.

In accordance with this study's focus, this section only covers approaches using video data from vehicle-mounted cameras that utilize ML or OF to detect events in the video footage.

3.1.1 Machine Learning

Unlike previous methods requiring extensive manual labeling, Yao, Xu, Wang, *et al.* developed an unsupervised anomaly detection model that analyzes deviations between predicted and actual vehicle trajectories to identify anomalies [9]. To avoid reliance on annotations, the model was trained on normal traffic footage to learn the patterns of such scenarios and thus be able to detect when abnormal situations occurred. This approach also addressed the long-tailed nature of traffic anomalies — that a small number of scenarios make up the vast majority of traffic situations, with a practically infinite amount of rare scenarios making up the rest — since it doesn't attempt to identify or classify the specific anomaly type, but rather treats all anomalies as one.

Bao, Yu, and Kong introduced an innovative model, UString, predicting traffic accidents using graph convolution and recurrent networks for relational feature learning and Bayesian neural networks for addressing uncertainty [24]. In a later

work, Bao, Yu, and Kong presented the DRIVE model for anticipating traffic accidents using deep reinforcement learning and prediction of human attention [8]. DRIVE uses the generated attention heatmaps to identify high-risk regions, which a reinforcement learning agent leverages to predict accident probability and the focal point for the next frame. In addition, the attention heatmaps provide visual explainability of the model's prediction, which is crucial in increasing trust from human users of AI systems.

3.1.2 Optical Flow

An alternative approach to ML to enable event detection is by using OF algorithms. By leveraging the detection of movement in images from OF techniques, one can avoid several challenges often faced when using ML-based approaches, particularly reliance on annotation and the quality of the dataset used during training.

In 2008, Diaz Alonso, Ros Vidal, Rotter, *et al.* proposed a system to detect overtakings by utilizing the fact that from the perspective of the ego vehicle — the vehicle on which the camera is mounted — everything behind it moves away from the car, with the notable exception of an overtaking vehicle [25]. They implemented this idea by placing a camera on the ego vehicle side rear-view mirror and computing OF in real-time using a custom digital signal processor (DSP). The computed OF was then used to track objects based on motion patterns and detect overtaking vehicles.

To detect anomalies in traffic video footage, Yuan, Wang, and Wang presented an approach that measures the abnormality of motion orientation and magnitude to detect anomalies [26]. The paper's novelty is mainly that measuring the OF motion orientation and magnitude is done separately. These results are then fused using a Bayesian model to make the method more robust and capable of handling more types of events. The method also treats motion differently depending on where in the image the motion occurs to ensure more focus is placed on relevant objects.

Kilicarslan and Zheng presented a real-time approach for the computation of time-to-collision using a vehicle-mounted camera [27]. The approach computes the motion in each frame using OF and uses this to calculate horizontal and vertical motion divergence in different areas of the frame. By analyzing motion divergence in specific, collision-sensitive zones, they could detect potential collisions and enable collision avoidance.

As an improvement of using (ML-based) appearance detectors to compensate for deficiencies when objects are distorted near the edges of the screen, Ramirez, Ohn-Bar, and Trivedi proposed the incorporation of OF [28]. Using front- and rear-facing cameras, they computed OF with compensation for the ego-vehicle's movement. Then, they performed clustering to group similar motion vectors and produced vehicle candidates, which were used together with an appearance detector to filter out false positives and output the final vehicle detections.

3.2 Saliency Prediction

Several studies have been published in the last decade on saliency prediction both for generic use and specifically for use in traffic contexts. Saliency prediction aims to identify the areas of an image or video that are of interest, either based on visual prominence [29]–[31] or where a human's attention would be placed [32]–[34]. This section will describe a selection of models presented for the latter type of saliency prediction.

For general use, Cornia, Baraldi, Serra, *et al.* presented MLNET that uses a CNN to extract and combine salient features at different levels of the network [32]. MLNET consists of three main parts: the CNN that extracts features at a low, medium, and high level; an encoding network that creates a saliency-specific map; and lastly, a learned prior that is applied to create the final saliency prediction map. A version of MLNET trained on eye tracking data in a traffic context was later provided by Bao, Yu, and Kong and used in their DRIVE model [8].

To model how drivers allocate their attention, Deng, Yan, Qin, *et al.* collected an eye-tracking dataset from having 28 drivers view 16 traffic driving videos [35]. This dataset was then used to train a convolutional-deconvolutional neural network (CDNN) to predict drivers' eye fixations.

To predict driver attention during critical driving situations, Xia, Zhang, Kim, *et al.* presented an attention dataset, Berkeley DeepDrive Attention, of braking events from a previously available large-scale dataset on which they collected eye tracking information from 45 participants [36]. Using this dataset, they presented a saliency prediction model that achieves state-of-the-art accuracy and exhibits behaviors such as correctly ignoring pedestrians on the sidewalk while retaining focus on crossing pedestrians.

TASED-net is a video saliency prediction model with a fully convolutional network architecture [33]. The model consists of two main parts: an encoder network that extracts spatiotemporal features from an input of several consecutive frames and a decoder network that decodes the encoded features. Thus, TASED-net produces a single saliency map from several input frames and can be applied sequentially on a video to achieve frame-by-frame saliency maps. The model performs especially well on moving salient objects.

To more accurately capture the viewing behavior of humans, Lou, Lin, Marshall, *et al.* presented a CNN-based prediction model, TranSalNet, incorporating transformer components to provide long-range visual context information [34]. Using this approach, TranSalNet achieved state-of-the-art performance on several saliency prediction benchmarks.

3.3 Event Retrieval in Large-Scale Datasets

While real-world driving data offers unmatched insight into factors behind safety-critical situations such as crashes or near-crashes, identifying these situations in

large-scale datasets poses a significant challenge. Several studies have been presented to address this issue of efficiently traversing and identifying patterns of events in both structured and unstructured data.

Perez, Sudweeks, Sears, *et al.* presented a method that identified events by processing on-board kinematic data with threshold queries for parameters such as acceleration, yaw rate, swerve, or the activation anti-lock braking systems (ABS) and airbags [37]. Using this approach, they identified a set of parameters that could reduce the number of candidate events and thus reduce the need for manual validation. However, the brute-force nature of the approach means it may not be sufficiently efficient for real-world application.

In analyzing database technologies to handle large volumes of spatial data (such as real-time feeds of geographical locations of smart cars), Hulbert, Kunicki, Hughes, *et al.* [38] investigated GeoMesa [39]. GeoMesa is an open-source technology that can be applied to several types of databases. It uses Z-order SFCs to reduce multi-dimensional spatio-temporal data into the single-dimensional key space for the database on which it is applied. In their analysis, Hulbert, Kunicki, Hughes, *et al.* conclude that while other approaches can match GeoMesa for smaller datasets, they cannot do so for spatiotemporal queries in the terabyte size range, displaying the benefit of using SFCs.

In a more general sense, Bader discussed how SFCs can reduce data dimensionality [40]. By mapping multidimensional data to a one-dimensional space while preserving locality, he showed that SFCs can improve cache efficiency, data clustering, and computational algorithms [40].

In a work by Zhou, Johnson, and Weiskopf, the authors build upon SFCs to visualize 2D and 3D data in one dimension [41]. They propose a ”data-driven” SFC that adapts to the data characteristics and an approach that optimizes a Hamiltonian path to traverse the data while preserving locality efficiently. This preserves important patterns and clusters in the input data, which helps visualize complex datasets more effectively than traditional SFCs [41].

Chen, Yao, Zhang, *et al.* [42] addressed the computational challenges of SFCs with the use of a deep learning-based SFC generation method using an efficient graph convolutional network (EGCN). The EGCN reduces memory usage and computational costs by exploiting the grid graph structure and predicting graph weights to identify important features and spatial relationships. They introduced a multi-stage minimum spanning tree algorithm and a Siamese network learning scheme to adapt to image compression and data transmission tasks. Their approach significantly outperforms traditional SFCs like the Hilbert curve and data-adaptive methods like neural SFC, achieving higher autocorrelation and better compression efficiency with less computational overhead Chen, Yao, Zhang, *et al.*

In the context of ADAS, SFC-enabled event retrieval has been explored by Berger and Birkemeyer to identify automotive maneuvers using kinematic data [3]. This was done by collecting a vehicle's x- and y-acceleration in a confined test area while executing lane changes and emergency braking and reducing the data to a single dimension

using SFCs. Then, it was identified in what regions of the single-dimensional space the two types of events were located to successfully identify events of both types in a real-world example.

In subsequent work, Berger, Cabrero-Daniel, Kaya, *et al.* applied SFCs to detect roundabouts using multiple Z-order and Hilbert curves configurations [2]. While a configuration using a Hilbert curve achieved the highest F1 score, one using a Z-order curve could detect a greater part of the roundabouts, although with more false detections.

Berger and Birkemeyer suggest further research using SFCs with different input sources, such as unstructured video footage, to potentially enable more detailed event detectors. This remains an open research area that this study will explore by incorporating SFCs on video data using ML and OF.

4

Methods

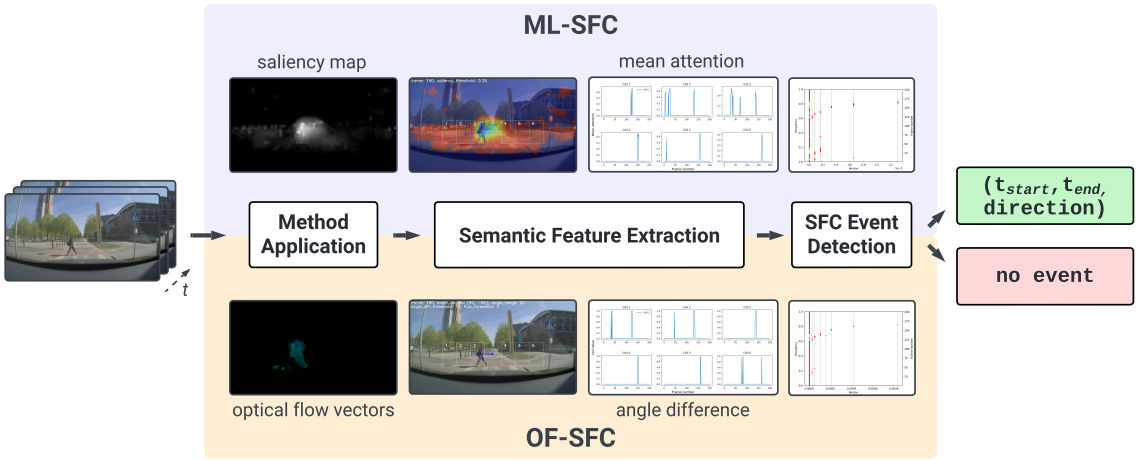


Figure 4.1: An overview of the traffic video event detection (VED) framework presented in this thesis. Semantic features are extracted for each frame in a video — using ML or OF — and are then filtered and converted to Morton codes using SFCs. Then, event detection is performed to output either an event window or indicate that no event is present. The original frames in the figure are from ZOD [43].

This chapter presents our framework for computationally efficient traffic VED in large volumes of unlabeled video footage. Our framework, shown in Fig. 4.1, applies ML-enabled saliency prediction or OF to extract semantically relevant features from video footage and then leverages the dimensionality-reducing properties of SFCs to enable computationally efficient event retrieval. The variant using saliency prediction is referred to as ML-SFC, and the variant using OF as OF-SFC.

The framework was implemented as a semi-automated pipeline to allow for iterative evaluation and improvement of our approach on datasets containing hundreds or thousands of videos, in accordance with design science research (DSR) principles [44]. The source code for the pipeline and our framework is available at https://github.com/erikwessman/ted-sfc.

The approach developed in this study expands on previous work by Berger and Birkemeyer [3] by using unstructured data from vehicle-mounted cameras rather than vehicle kinematics. Since SFCs expect structured data as input, using video footage requires information in each frame to be extracted to a structured form, for

which this study compares two alternative techniques: ML-based saliency prediction models and OF algorithms.

In developing our framework, we adopted several core principles of DSR, as outlined by Hevner, March, Park, *et al.* [44]. Particularly effective when creating innovative technological solutions, DSR stresses iterative design, evaluation, and improvement cycles. In our study, this iterative process involved continuous improvement of our framework based on systematic evaluation with synthetic and real-world datasets. This process aligned with DSR and ensured our framework was rigorously tested and validated in simulated and real-world scenarios.

This chapter will first describe the identification and selection of the methods and resources used to develop our framework. Then, the framework's components will be detailed, including the feature extraction, the application of SFCs, and how events were detected. After describing the framework, RQ1 is answered. Lastly, the experiment setup used in evaluating the framework will be described.

4.1 Selection of Methods and Resources

This chapter outlines the process of selecting the ML models for saliency prediction and OF algorithm used by the framework. The arguments used in the decision process are described, along with the requirements and tools used to narrow the search.

4.1.1 Machine Learning Models

Several inclusion and exclusion criteria that aligned with our project goals and needs were specified to identify and select the appropriate ML models for the study.

A primary inclusion criterion was that the model had to have an accompanying research paper. This ensured the models were rigorously tested and provided benchmarks against other models, facilitating comparative analysis. To identify models connected to research, a search was performed using the Google Scholar search engine with the search string: "(anomaly OR accident) detection AND (first-person OR dashcam) videos."

The first exclusion criterion was based on the models' recency, dismissing models older than 5 years to ensure focus on state-of-the-art research. It was also crucial for models to be open-source and well-documented to enable their use in the framework and allow for potential modifications or extensions.

Initially, three VED models were identified that met these criteria: FOL [9], UString [24], and DRIVE [8]. FOL is an unsupervised model that predicts the future locations of objects in dashcam footage, using deviations from these predictions to detect anomalies. UString integrates spatiotemporal relational learning to anticipate traffic accidents, and DRIVE combines a visual saliency model with deep reinforcement learning to predict accidents and human attention.

Compatibility and preprocessing challenges with FOL and UString were encountered upon implementation, and these models were thus dismissed. DRIVE proved more compatible due to its minimal preprocessing requirements and comprehensive documentation, making it the only model considered for further use.

Although the initial goal was to use a complete VED model to extract features, it was discovered that a more flexible approach could be achieved by using DRIVE's underlying saliency prediction models. DRIVE supports and provides pre-trained model weights for two saliency models: MLNET [32] and TASED-Net [33]. The models are trained on traffic data from the DADA-2000 dataset [45], which includes fixation point ground truth for each video, representing where a human wearing an eye-tracker fixates their attention. This means they are highly suitable for use in traffic contexts. The DRIVE code and pre-trained models were reused and repurposed in this study's feature extraction implementation.

For further comparative analysis, TranSalNet [34], a state-of-the-art saliency model not pre-trained on traffic data, was included to evaluate its performance in the context of traffic event detection.

4.1.2 Optical Flow Algorithm

The Farnebäck algorithm was selected for computing OF due to its optimal balance between computational efficiency and accuracy in flow estimation. This algorithm is particularly effective for analyzing motion patterns across a scene, unlike the Lucas-Kanade method, which focuses on tracking specific features in sparse OF [18]. Farnebäck is advantageous for processing the complex and dynamic backgrounds found in real-world driving footage, adeptly managing variations in weather, lighting, and traffic conditions. Unlike deep learning models such as FlowNet [6], which require extensive training datasets, the Farnebäck algorithm operates independently of training data. This allows it to be deployed across diverse scenes and scenarios without retraining.

4.1.3 Datasets and Event

This section describes identifying and selecting datasets used to design and evaluate the approach. Similarly to the model selection process, several inclusion and exclusion criteria were chosen. Applying these criteria, three suitable datasets were identified: ZOD [43], SMIRK [46], and VAS HD [47].

The first and most fundamental criterion was that the dataset must consist of vehicle-mounted, forward-facing traffic video footage. This was a necessity due to the purpose of the study, which was to investigate traffic event retrieval.

Licensing, anonymization, and availability for research purposes were other crucial criteria to ensure the study's approach and findings' full transparency and reproducibility.

The datasets' annotations were another important factor, though not crucial. To simplify the work, particularly regarding manual annotation for evaluation, existing

annotations of objects present in each frame later proved highly beneficial. The ideal would be a dataset with annotations for any events in a video, such as cut-in, pedestrian crossing, or others, but no datasets investigated provided this.

Based on our criteria, datasets from three different sources were filtered. The first source included the datasets referenced and used in the aforementioned papers for ML VED, see Section 4.1. The second source contained datasets published on GitHub, specifically in two collections [48], [49]. The final source was datasets available through the universities' partnerships and relations, specifically with AI Sweden [50] and Zenseact [51]. This led to the identification of seven datasets matching the criteria: ZOD [43], SMIRK [46], VAS HD [47], DADA-2000 [45], CCD [52], DAD [53], D²-City [54], and DoTA [55].

Due to licensing, privacy concerns, and time constraints, the seven datasets were further narrowed down to SMIRK [46] and ZOD [43], as they were the most appropriate for the studied event type.

The SMIRK dataset is a synthetic dataset developed for the experimental pedestrian automatic emergency braking system called SMIRK, led by RISE. It comprises 4,928 scenarios involving pedestrians crossing or moving near a straight road at various speeds and trajectories relative to the camera, along with similar scenarios featuring basic geometric shapes crossing the road. The dataset was generated using ESI Pro-SiVIC and is designed to support advancements in ML-based systems for automotive safety. It includes images and segmentation maps for moving objects collected from a virtual forward-facing camera.

The Zenseact Open Dataset (ZOD) is a multi-modal dataset for autonomous driving (AD) development created and provided by Zenseact. The dataset contains anonymized data from 14 different European countries, and the sensor data was collected using a high-resolution camera, three LiDAR sensors, and one GNSS/IMU sensor. The dataset is divided into three subsets – Frames, Sequences, and Drives – respectively containing individual frames, 20-second frame sequences, and multi-minute sequences. The subset explored and used in this thesis was sequences, which consist of 1473 20-second video sequences. The annotations for the dataset contain data from each of the sensors, of which only the camera data was used, and the annotations are available per sequence and not per frame.

As mentioned in Section 1.4, the focus was placed on detecting only pedestrians and bikes crossing the road to limit the study's scope. This decision was largely influenced by the synthetic dataset SMIRK, which consists solely of pedestrian crossings. This allowed the implementation of event detection in a simulated environment to limit disturbances and learnings to be transferred to the real-world ZOD dataset. Additionally, a pedestrian crossing is a relatively easy and consistent event to detect, making it suitable for developing a proof-of-concept framework.

4.2 Semantic Feature Extraction

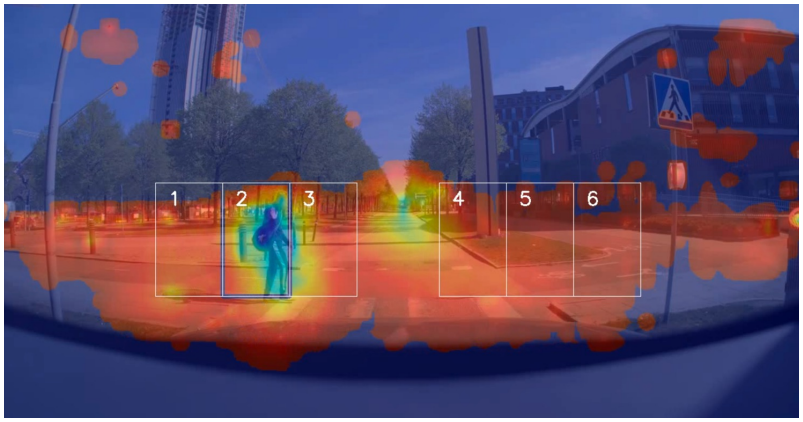
This section outlines the methods for extracting semantically relevant features from video data using ML and OF. Extracting these values enabled analysis of elements' movements across the grid over time and, thus, event detection. To effectively capture the spatiotemporal features within each video, a domain-specific region of interest (RoI) grid was applied to each video frame, partitioning it into cells and capturing values for each cell. Using this grid, information could be extracted while retaining information about its location in the frame. For ML-SFC, the extracted value for each cell was the mean attention, while the angle difference between the OF vector and the horizontal axis was used for OF-SFC.

4.2.1 Region of Interest Grid

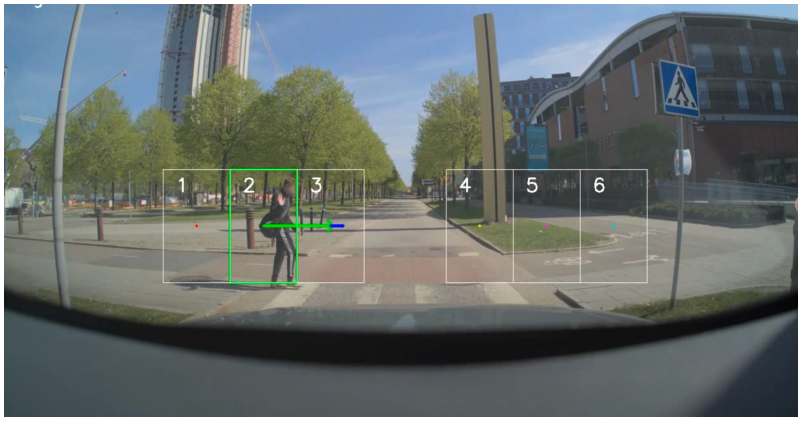
Different events occur in different areas of a frame; thus, different regions have varying levels of relevance. The area with the greatest importance for a specific event and dataset is referred to as the domain-specific RoI. When determining the position and shape of the grid used for feature extraction, this RoI was crucial.

For pedestrian crossings, the RoI is situated along the horizontal axis and vertically in the middle of the frame, as that is the area pedestrians move across. Therefore, a vertically centered grid was used, which covered most of the frame's width and roughly a third of the height, as seen in Fig. 4.2. Additionally, a driver's gaze is often focused on the vanishing point of the scene, meaning the saliency prediction models trained on eye tracking data predict near-constant attention here. Thus, a gap was placed in the middle of the grid to exclude this noise and was also used for OF-SFC for consistency.

The RoI also depends on the ego vehicle and camera position used to collect the data. For example, a larger ego vehicle may mean that the RoI for pedestrian crossings will be lower in the frame relative to the camera's position. Thus, the grid's positioning was also influenced by the characteristics of the datasets used.



(a) ML-based saliency prediction approach.

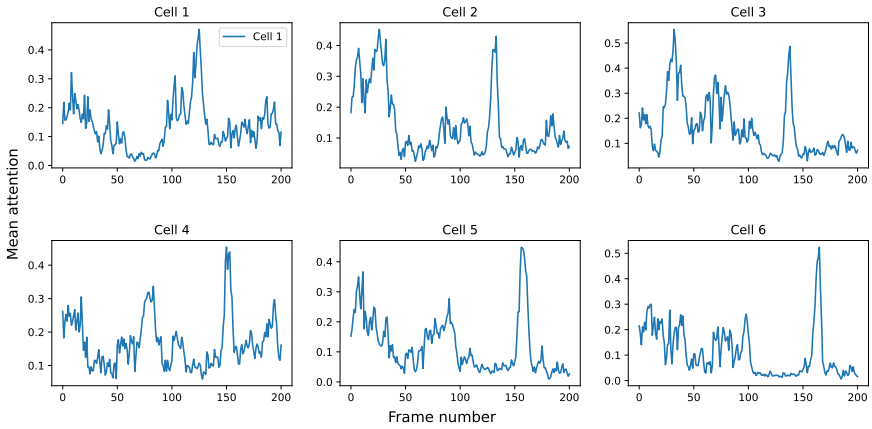


(b) Optical flow approach.

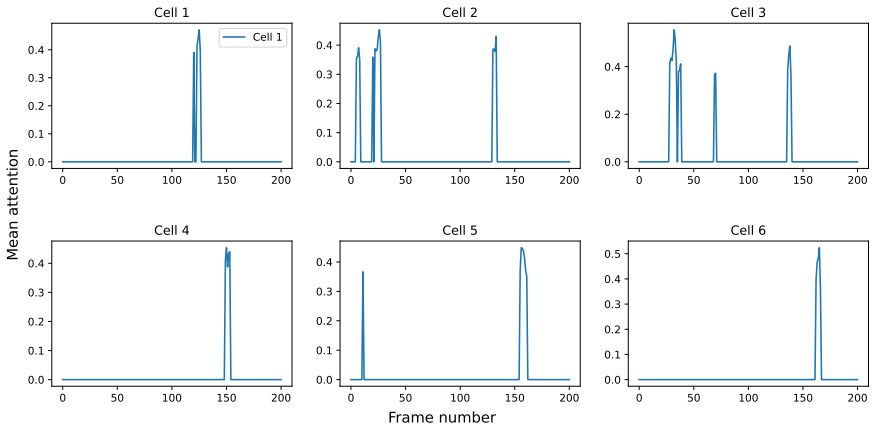
Figure 4.2: Example output of the two variants of our framework applying the RoI grid on a video of a pedestrian crossing from ZOD [43]. Approach (a) computes a saliency map predicting human attention, while approach (b) computes disturbances in the vector field compared to the generally expected OF. The original image is taken from ZOD [43].

4.2.2 ML-enabled Saliency Prediction

For the saliency prediction approach of the framework, ML models were used to compute saliency maps for each frame. Then, the mean saliency for each cell in the domain-specific RoI grid was computed to extract the cell values. In this stage, two filters were applied to reduce noise. First, only the cell with the largest mean saliency value was kept for each frame, and the others were set to 0. Second, a domain-specific saliency threshold was used to nullify all values under it. Fig. 4.3 shows unfiltered and filtered example cell value output for a video from ZOD.



(a) Unfiltered mean saliency value for each cell.



(b) Filtered mean saliency value for each cell.

Figure 4.3: Mean saliency cell values extracted using MLNET on a video of a pedestrian crossing from ZOD.

Three different ML models were implemented and compared in our framework to thoroughly test the saliency prediction approach: MLNET [32], TASED-Net [33], and TranSalNet [34].

The MLNet algorithm [32] is applied across an entire video frame to comprehensively observe the scene, leveraging the attention mechanism to identify pertinent driving events. The process begins with an input image, from which a CNN based on the

VGG-16 architecture [56] computes features at low, medium, and high levels. The extracted feature maps are subsequently fed into an encoding network tasked with learning a feature weighting function, which is crucial for generating feature maps specifically tailored for saliency detection. Following this, a prior image is learned and strategically applied to enhance the accuracy of the resulting saliency map. Saliency maps are generated for each video frame where a pixel's intensity reflects the level of saliency.

The second saliency model tested was the temporally-aggregating spatial encoder-decoder network (TASED-Net) [33]. TASED-net is a 3D convolutional encoder-decoder model designed to leverage collective spatiotemporal information to improve saliency prediction. This contrasts earlier models that performed the spatial decoding and temporal aggregation separately. The model architecture comprises two main components: an encoder and a prediction network. The encoder extracts low-resolution spatiotemporal features from a sequence of frames. The prediction network then reconstructs high-resolution spatial details from the compressed, encoded feature representation, resulting in the saliency map.

The third and last saliency model that was used is TranSalNet [34]. This model integrates transformers into a CNN-based architecture to learn long-range spatial relationships within images. The model first processes images through a CNN to extract multi-scale spatial features. These features lack contextual information, and to overcome this, TranSalNet uses transformers that apply a self-attention mechanism to capture broader contextual details. This allows TranSalNet to model local and non-local visual cues effectively, enriching the saliency maps.

4.2.3 Optical Flow

The mean flow vector within each grid cell, representing the average motion, is computed using Farnebäck OF. Using the mean flow vector, the direction of motion in each cell can be calculated as the angle of the mean flow vector in that cell. This direction was used as the basis for the cell value computations for OF-SFC. The mean flow vector was computed and stored for each frame for each cell.

To denote a crossing event, the angle of the flow vector at the current frame t should approximate an event angle $\alpha = 90°$ for left-to-right crossings and $\beta = -90°$ for right-to-left. This is equivalent to an object moving along the horizontal axis in either direction. To account for noise, the acceptable range for these event angles includes a threshold θ. By comparing the direction of the average mean flow vector from frames $\{f_t, f_{t-1}, \ldots, f_{t-m}\}$ to $\{f_{t-m-1}, f_{t-m-2}, \ldots, f_{t-n}\}$, any unexpected changes in the flow motion relative to its expected pattern can be identified. In our study, we set $m = 2$ and $n = 9$, and a visual representation of this is presented in Fig. 4.4. A cell is considered part of an event when the angle difference between the previous n average mean flow vector and the m average mean flow vector of the preceding frames exceeds a threshold δ.

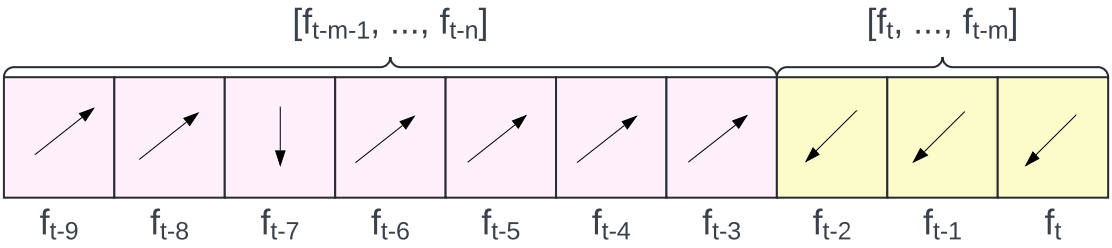


Figure 4.4: A visualization of the previous frames used for detecting sudden changes in the OF. The effect of the flow vector in frame f_{t-7} is dampened due to the use of a mean flow vector, and the change in the mean flow in frames f_t, f_{t-1}, f_{t-2} is detected since it is compared to the previous mean flow.

4.3 Dimensionality Reduction with Space-Filling Curves

After the feature extraction using either saliency prediction or OF and the 1x6 RoI grid, the resulting output was a six-dimensional vector for each frame in each video. While it would have been possible to detect events in this format, leveraging SFCs enabled a significant reduction in complexity and, thus, computational demand. Using the `zCurve` package [57], the six-dimensional input vectors were transformed to a single number, a Morton code, and following this SFC application, each video in the dataset could be represented by a characteristic stripe pattern (CSP), as shown in Fig. 4.5.

A CSPs can give significant information about the spatiotemporal properties of movements in a video. For example, the video represented by the CSP shown in Fig. 4.5 contains a pedestrian crossing, which can clearly be distinguished as the sequence of red dots within the event window, starting around frame 125. It is this information that was later leveraged for the event retrieval.

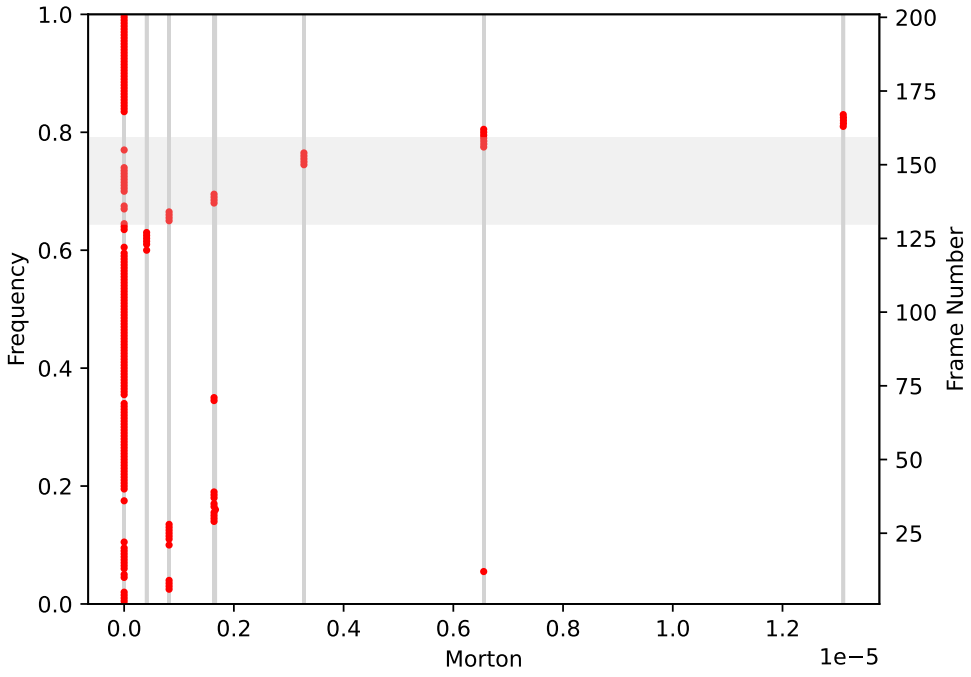


Figure 4.5: A CSP from a video in ZOD containing a pedestrian crossing. The grey stripes denote the Morton codes with several being grouped in the same range near 0. The red dots indicate which frame the Morton codes are connected to. The horizontal band spanning from frame 130 to 159 is the event window where the crossing occurred.

4.4 Computationally Efficient Event Retrieval

To detect methods using the Morton codes generated using SFCs, a method of finding how objects move across the grid and a set of spatiotemporal constraints that define how a pedestrian crosses a road was required. Essentially, this method aimed to distinguish Morton code sequences of pedestrian crossings, like the red dots within the highlighted range in the CSP seen in Fig. 4.5 from sequences of non-crossings, such as the noise occurring elsewhere in the figure.

Systematic experimentation was first performed on SMIRK to identify the constraints and validate various detection methods. The key parameters and learnings were later transferred to the real-world dataset, ZOD. This was essential in validating the algorithm's effectiveness in more complex and variable conditions.

Two primary considerations influenced the decision to perform the initial experimentation with the synthetic dataset. First, its smaller size and less complex scenes made it faster to load the dataset and extract features. Second, the synthetic nature of the dataset meant minimal noise, further simplifying the feature extraction process. These factors simplified the evaluation and comparison of different approaches and the refinement of the framework in a more simple environment.

Each combination of active cells in the RoI grid leads to a unique Morton code in the CSPs for each video. This fact was leveraged to identify ranges of Morton codes corresponding to each specific cell. By observing the change in Morton codes for each cell in the synthetic dataset, SMIRK, these Morton code ranges were defined and mapped to their corresponding cell ID. The ranges are presented in Table 4.1 and visualized in Fig. 4.6. For example, a Morton code of 3.0×10^{-7} would correspond to activity within cell 1 when using one of the saliency models, and the same cell would activate with a Morton code of 2.5×10^{-5} when using OF.

Cell ID	**Saliency ranges**	**Optical flow ranges**
1	$[2.7 \times 10^{-7},\ 5.5 \times 10^{-7}]$	$[1.8 \times 10^{-5},\ 3.5 \times 10^{-5}]$
2	$[6.5 \times 10^{-7},\ 1.0 \times 10^{-6}]$	$[4.7 \times 10^{-5},\ 6.0 \times 10^{-5}]$
3	$[1.44 \times 10^{-6},\ 1.84 \times 10^{-6}]$	$[9.5 \times 10^{-5},\ 1.2 \times 10^{-4}]$
4	$[3.1 \times 10^{-6},\ 3.5 \times 10^{-6}]$	$[1.95 \times 10^{-4},\ 2.25 \times 10^{-4}]$
5	$[6.3 \times 10^{-6},\ 6.8 \times 10^{-6}]$	$[4.05 \times 10^{-4},\ 4.35 \times 10^{-4}]$
6	$[1.28 \times 10^{-5},\ 1.34 \times 10^{-5}]$	$[8.2 \times 10^{-4},\ 8.5 \times 10^{-4}]$

Table 4.1: Ranges of Morton codes mapped to cell IDs. These values were used to identify if a Morton code belongs to the range of any of the cells.

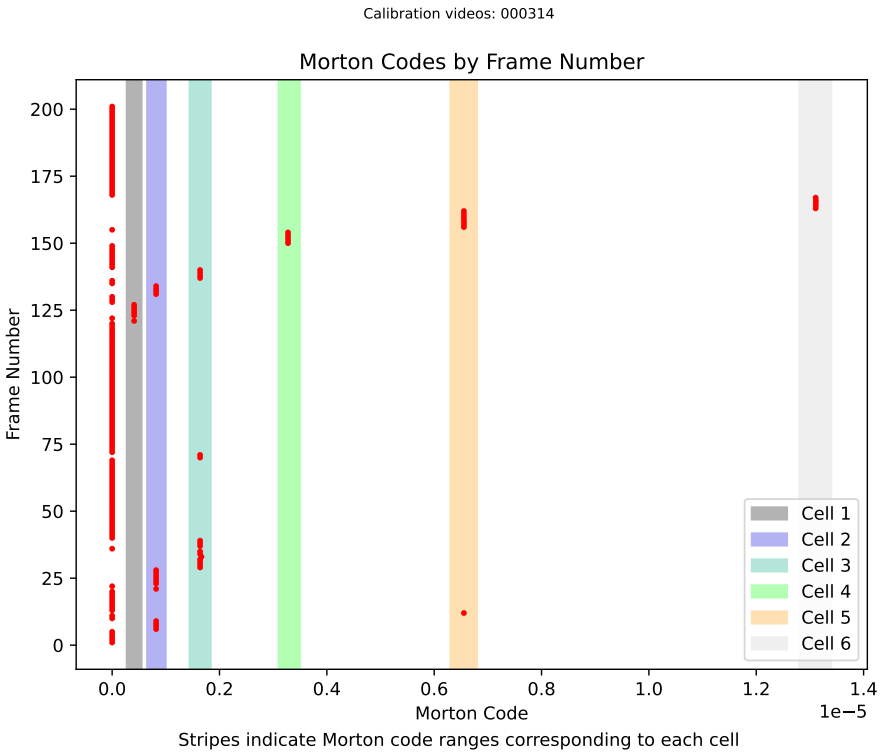


Figure 4.6: A visualization showing the Morton code ranges corresponding to each cell ID in the RoI grid.

A set of constraints for a pedestrian crossing was defined, outlining the specific movement patterns detected using the aforementioned Morton code ranges required for a crossing to be considered valid. These constraints were:

1. **Ordered sequence:** The sequence of cell activations must be ordered, either increasing or decreasing with at most $n = 1$ interruption between any two cells. An interruption is an unexpected value that disrupts the natural consecutive ordering. For example, $1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 4$ is treated as two sequences: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ and $1 \rightarrow 4$.
2. **Event time:** The total time of the event window must be within a defined minimum and maximum event time.
   - Min: 1.25 seconds
   - Max: 10 seconds
3. **Transition times:** The time t (seconds) between each activated cell must be within defined limits.
   - 1 & 2: $0.1 \leq t \leq 3$
   - 2 & 3: $0.1 \leq t \leq 3$
   - 3 & 4: $0.5 \leq t \leq 5$
   - 4 & 5: $0.1 \leq t \leq 3$

  - 5 & 6: $0.1 \leq t \leq 3$

4. **Enter and exit cells:** There must be at least 3 activated cells, whereof at least one must be in the left part of the grid and at least 1 in the right part.
5. **No jumps:** The ordering of the activated cells must not contain sudden jumps by more than 2 cells. For example, $1 \rightarrow 3$ is OK, while $1 \rightarrow 4$ is not OK.

If a sequence of Morton codes satisfies the constraints, that sequence is considered a valid pedestrian crossing. The constraints are checked twice to detect crossings from both left and right. If more than 1 crossing is detected in any video, the sequence that traversed the largest amount of cells is kept since that sequence is most likely to align with the ground truth since it has more chances to fail but doesn't.

4.5 Answering RQ1

> **RQ1:** How can machine learning models and optical flow be utilized to extract semantically relevant visual features from video data, with the objective of integrating these features into space-filling curves?

To address RQ1, the study adopted the DSR methodology to develop a framework for detecting vehicle events in forward-facing cameras. An artifact in the form of a semi-automated pipeline was designed to extract semantically relevant, spatiotemporal features in a domain-specific RoI grid. These features were then successfully fed to an SFC to create a one-dimensional representation of the features.

The answer to RQ1 found by this study is that semantically relevant features can be extracted by computing visual properties in each frame using techniques such as ML-based saliency prediction or OF algorithms and applying a RoI grid to capture the features in the relevant areas of the frame.

4.6 Experiment Setup

This section details the methods, evaluation metrics, and benchmarking procedures employed to assess the effectiveness and efficiency of the different approaches. In accordance with the principles of DSR, the experiments were performed in cycles, evaluating and improving the approach in every iteration. Each ML model (MLNET, TASED-Net, and TranSalNet) and the OF algorithm (Farnebäck) were evaluated on subsets of the SMIRK and ZOD datasets. The experiments were performed on two machines, the specifications of which are presented in Table A.1.

4.6.1 Ground Truth Annotation

A systematic process of annotating the datasets was applied to evaluate the accuracy of the event detection. This section details the criteria for a crossing event and the annotation methods used.

For this study, a pedestrian crossing event is characterized by specific criteria ensuring the event's relevance to the research objectives:

1. **Proximity of the ego vehicle:** The event must occur close enough to the ego vehicle to ensure the crossing is within a clear visual range and relevant to the vehicle's immediate context.
2. **Positioning of the ego vehicle:** The ego vehicle should be the foremost vehicle in the lane closest to the crossing, providing an unobstructed view of the event.
3. **Complete crossing:** The video must capture the entire crossing duration, from entry to exit, of the pedestrian on the road.
4. **Isolation of the event:** Ideally, the video should document only one crossing event. If multiple pedestrians or bikes are present, they are considered part of the same event only if they move in reasonably close proximity and the same direction.

For each video containing a qualifying pedestrian crossing, the following annotations were recorded:

1. **Event window**: The frames where the pedestrian or bike entered and exited the road were recorded, defining the start and end of the crossing event.
2. **Crossing category**: The direction of the movement across the road was noted and categorized into 'left' for movements from left to right and 'right' for movements from right to left.

4.6.1.1 Zenseact Open Dataset

ZOD provides extensive video data capturing various traffic scenarios, which includes interactions involving pedestrians and vehicles. However, ZOD does not offer comprehensive frame-by-frame annotations; it only provides annotations for a single frame in each video. Therefore, a subset of the data was manually annotated to identify videos containing pedestrian crossings. An initial filtering step was performed using the ZOD annotations to remove all scenarios containing no pedestrians, reducing the dataset size to 358 videos, and these were manually annotated using the criteria defined above.

4.6.1.2 SMIRK

SMIRK provides frame-by-frame annotations, which were used to automatically generate ground truth annotations for the entire dataset. To do this, the camera parameters used to generate SMIRK were defined. SMIRK uses a monocular camera with a 752×480 resolution, a 3.13 cm × 2.00 cm sensor, a 3.73 cm focal length, and an angle of view of 45 degrees [46]. Next, the pedestrian X and Y coordinates in camera space were calculated using their bounding boxes. The pedestrian X, Y, and Z coordinates in the virtual environment could be approximately calculated using this information. To maintain consistency with the previously defined pedestrian crossing,

any crossing 26 meters or further from the camera was marked as non-crossing. Finally, by comparing the pedestrian's X coordinate with the X coordinates that define the bounds of the road, we were able to calculate the event window in which the pedestrian enters and exits the road, as well as the direction of the crossing.

4.6.2 Metrics

As described in Section 2.4, four metrics were used for measuring efficacy: mean IoU, sensitivity, specificity, and F1 score, and one for efficiency: FPS. The definitions and benefits of each can be found in Section 2.4.

Since IoU is calculated per video, the study uses the mean IoU across all videos containing a crossing that was correctly predicted (true positives).

During the evaluation, the definitions of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) were as follows:

- **TP**: Any overlap between the predicted and ground truth event windows and the same crossing category (direction).
- **TN**: No event window is predicted when no crossing is present in the video.
- **FP**: An event window predicted for a video with no crossing, a predicted event window with no overlap with the ground truth event window, or an overlap but mismatching direction.
- **FN**: No event window predicted in a video with a pedestrian crossing.

To ensure a fair comparison regarding processing speed, FPS was only compared between the same machine. A version of the Farnebäck OF algorithm compiled with GPU support was utilized to ensure both the machine learning and OF methods benefit from parallel processing capabilities provided by NVIDIA CUDA [58].

5

Results and Discussion

This section presents and analyzes the results and findings of the study based on the data obtained from the experiments, for which the setup is detailed in Section 4.6. These results form the answer to RQ2, and the significance of the results in terms of the performance of the different variants is highlighted.

> **RQ2:** What is the efficacy and efficiency of using space-filling curves to detect traffic events when applied to features extracted from video data?

5.1 Description of Evaluation Datasets

SMIRK and ZOD were partitioned into subsets to create test sets, presented in Table 5.1 with a balanced number of videos with crossings and non-crossings, and to ensure a fair evaluation. Two partitions were created for SMIRK, one containing the full dataset, a total of 4928 videos, and a smaller version with an equal amount of crossings and non-crossings, 25 each, sampled randomly. Similarly, 4 subsets were created for ZOD, each containing the same 16 crossing videos and 17 unique non-crossings. An initial set of 358 ZOD videos was created by finding all videos that contained at least one pedestrian (meaning that the video contains a pedestrian but not necessarily a pedestrian crossing the road) using the provided annotations, and from this, the 4 subsets were created. This was done to create datasets representative of driving environments where crossings would be typically encountered.

Name	**Number of videos**	**Crossings**	**Non-crossings**
SMIRK_{full}	4928	640	4288
SMIRK_{small}	50	25	25
ZOD_1	33	16	17
ZOD_2	33	16	17
ZOD_3	33	16	17
ZOD_4	33	16	17

Table 5.1: Overview of the dataset partitions created for SMIRK and ZOD.

After creating the partitions, a description of each ZOD video was manually created to make the study reproducible and aid the discussion of the results. The descriptions included weather (cloudy, sunny, raining, snowing, foggy), lighting (bright, dim, night), setting (urban, highway, countryside), and country code (SE, DE, FR). The descriptions are provided in Tables B.1 to B.5.

5.2 Evaluation on Synthetic Dataset (SMIRK)

The results obtained after evaluating the approaches on the SMIRK subsets are presented in Table 5.2. MLNET$^{\text{ML}}$, TranSalNet$^{\text{ML}}$, Farnebäck$^{\text{OF}}$ exhibit comparative results regarding efficacy for SMIRK$_{full}$. TranSalNet$^{\text{ML}}$ performed the best, achieving a 0.9091 F1 score, 1.0 sensitivity, and 0.8 specificity, but had the lowest mean IoU of 0.6538. Regarding computational efficiency, TranSalNet$^{\text{ML}}$ performed the best with 33.98 FPS.

From evaluating SMIRK$_{full}$, we see that Farnebäck$^{\text{OF}}$ achieves results consistent with its performance on SMIRK$_{small}$. Here, Farnebäck$^{\text{OF}}$ achieves the highest F1 score of 0.8083 and specificity of 0.9956 but with low sensitivity, meaning it tends to miss TPs. In contrast to Farnebäck$^{\text{OF}}$'s consistency, the efficacy of the variants of ML-SFC exhibit more variation between SMIRK$_{full}$ and SMIRK$_{small}$, as shown by MLNET$^{\text{ML}}$ and TranSalNet$^{\text{ML}}$'s F1 scores dropping significantly. TranSalNet$^{\text{ML}}$ was the fastest (46.60 FPS) but also had the lowest F1 score (0.5304) and specificity (0.7738) in this subset, indicating a trade-off between speed and accuracy.

Subset	Variant	F1 Score	Sensitivity	Specificity	Mean IoU	FPS
SMIRK$_{small}$	MLNET$^{\text{ML}}$	0.8372	0.72	**1.0**	**0.7298**	25.42
	TranSalNet$^{\text{ML}}$	**0.9091**	**1.0**	0.8	0.6538	**33.98**
	Farnebäck$^{\text{OF}}$	0.8889	0.8	**1.0**	0.7242	22.91
SMIRK$_{full}$	MLNET$^{\text{ML}}$	0.7374	0.6297	0.9883	0.705	28.26
	TranSalNet$^{\text{ML}}$	0.5304	**0.9078**	0.7738	0.6699	**46.60**
	Farnebäck$^{\text{OF}}$	**0.8083**	0.6984	**0.9956**	**0.7328**	21.88

Table 5.2: Results from event retrieval using our framework on the synthetic dataset SMIRK. Three techniques were compared, one using OF (Farnebäck$^{\text{OF}}$) and two using ML (MLNET$^{\text{ML}}$ and TranSalNet$^{\text{ML}}$). Evaluation performed with PC_2, see Table A.1. Note that TASED-net was not applied to this dataset due to technical limitations.

The synthetic nature of the SMIRK dataset likely contributes to the high accuracy observed, as it contains low noise and well-defined features, making it easier for the models to perform accurately. Additionally, SMIRK scenes are not complex and have relatively low resolution, resulting in less information for all variants of ML-SFC and OF-SFC to process, contributing to the high FPS observed. The scenes in SMIRK are also consistent regarding lighting and other environmental factors, simplifying the detection process but not reflecting the variability encountered in real-world scenarios.

We identify several factors that negatively impact the techniques' performance. Firstly, the grid placement for pedestrian detection is only effective at certain distances. For example, a pedestrian far away does not align well with the grid, whereas one closer does, leading to lower mean IoU. Therefore, the chosen distance of 26 meters to conform to our pedestrian crossing definition, see Section 4.6, may have impacted the efficacy of the techniques. Third, pedestrians' rapid and erratic movement can disrupt the OF calculations, leading to inaccuracies. Finally, the worse results on the SMIRK_{full} subset can be attributed to the increased volume of data, providing more opportunities for incorrect predictions.

5.3 Evaluation on Real-World Dataset (ZOD)

The ZOD dataset containing real-world footage was evaluated with three different variants of ML-SFC (MLNET^{ML}, $\text{TranSalNet}^{\text{ML}}$, and $\text{TASED-Net}^{\text{ML}}$) alongside OF-SFC ($\text{Farnebäck}^{\text{OF}}$). As described in Section 5.1, four test sets were created with a balanced number of positives (crossing videos) and negatives (non-crossing videos).

The results in Table 5.3 show that MLNET^{ML} performs best in terms of F1 score on ZOD_{2-4}, while $\text{TranSalNet}^{\text{ML}}$ performs best on ZOD_1. This indicates MLNET^{ML} is effective at minimizing both FP and FN, making it the best choice when it's important to reduce both types of errors. On the other hand, $\text{TranSalNet}^{\text{ML}}$ achieves the highest sensitivity in each of the subsets, meaning it correctly identifies the most TPs. However, its low specificity indicates that this is due to a general tendency to predict positives, leading to a high FP rate. This means $\text{TranSalNet}^{\text{ML}}$ may be suitable when a greater amount of FPs is acceptable to minimize the amount of missed TPs. Opposite of $\text{TranSalNet}^{\text{ML}}$ is $\text{Farnebäck}^{\text{OF}}$, which achieves the highest specificity with lower sensitivity, meaning it may be viable when reducing the number of FPs is important. $\text{Farnebäck}^{\text{OF}}$ also scores the highest on mean IoU, indicating it does better in detecting when an event starts and ends. Last, $\text{TASED-Net}^{\text{ML}}$ performs the worst in most metrics except for specificity, in which it places second behind $\text{Farnebäck}^{\text{OF}}$ on $\text{ZOD}_{1,3,4}$.

Regarding computational efficiency and FPS, $\text{TranSalNet}^{\text{ML}}$ performs best by a significant margin, with its average FPS (27.78) being roughly 31.5% and 32.2% faster than the evenly matched MLNET^{ML} (21.12) and $\text{Farnebäck}^{\text{OF}}$ (20.99) variants, respectively. In this regard, $\text{TASED-Net}^{\text{ML}}$ (14.94) once again performs worst.

The variants' performance was consistent across the different test sets, with only minor variance. However, some cases had greater differences, such as $\text{TranSalNet}^{\text{ML}}$'s F1 score being significantly better on ZOD_1 than ZOD_{2-4}.

5.3.1 Challenges in Analyzing Real-World Traffic Footage

Because it contains video footage from real-world traffic scenarios where the ego vehicle is moving, light conditions vary, and many scenes are busy, ZOD presents significant challenges for the ML-SFC and OF-SFC approaches.

The greatest of these challenges affects the $\text{Farnebäck}^{\text{OF}}$ approach and is due to the

Subset	Variant	F1 Score	Sensitivity	Specificity	Mean IoU	FPS
ZOD_1	$MLNET^{ML}$	0.6207	0.6429	0.6842	0.4988	20.71
	$TranSalNet^{ML}$	**0.6429**	**0.75**	0.6667	0.4827	**28.59**
	$Farneb\ddot{a}ck^{OF}$	0.4545	0.3125	**0.9412**	**0.5143**	21.05
	$TASED\text{-}Net^{ML}$	0.2727	0.2143	0.7368	0.3481	14.75
ZOD_2	$MLNET^{ML}$	**0.6429**	0.6429	0.7368	0.4988	21.24
	$TranSalNet^{ML}$	0.4865	**0.75**	0.2381	0.4827	**28.14**
	$Farneb\ddot{a}ck^{OF}$	0.4545	0.3125	**0.9412**	**0.5143**	21.00
	$TASED\text{-}Net^{ML}$	0.2609	0.2143	0.6842	0.3561	15.18
ZOD_3	$MLNET^{ML}$	**0.5806**	0.6429	0.5789	0.4988	21.43
	$TranSalNet^{ML}$	0.5625	**0.75**	0.4762	0.4827	**25.84**
	$Farneb\ddot{a}ck^{OF}$	0.4167	0.3125	**0.8235**	**0.5143**	20.92
	$TASED\text{-}Net^{ML}$	0.25	0.2143	0.6316	0.3481	15.14
ZOD_4	$MLNET^{ML}$	**0.5806**	0.6429	0.5789	0.4988	21.08
	$TranSalNet^{ML}$	0.5143	**0.75**	0.3333	0.4827	**28.56**
	$Farneb\ddot{a}ck^{OF}$	0.4348	0.3125	**0.8824**	**0.5143**	20.97
	$TASED\text{-}Net^{ML}$	0.25	0.2143	0.6316	0.3561	14.70

Table 5.3: Results from event retrieval using our framework on the real-world ZOD dataset. Four different variants were compared, OF ($Farneb\ddot{a}ck^{OF}$) and three using ML ($MLNET^{ML}$, $TranSalNet^{ML}$, and $TASED\text{-}Net^{ML}$). Evaluation performed with PC_2, see Table A.1.

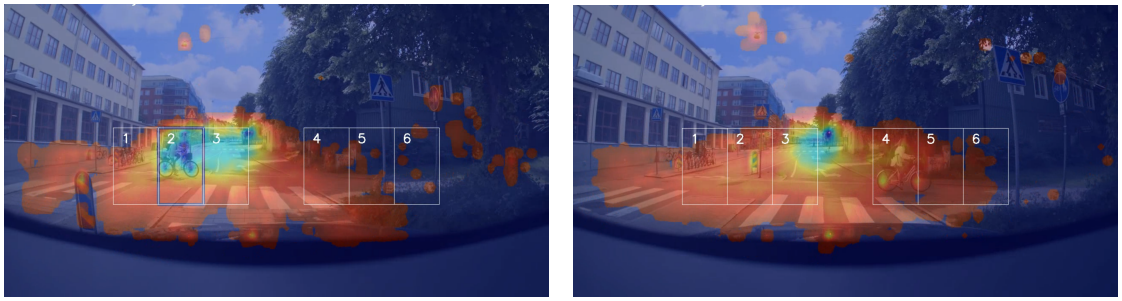
ego-vehicle movement. Since OF measures the pixel-wise changes between frames, it treats movement similarly regardless of whether an object in the environment moves or if the ego vehicle causes the camera to move. A clear consequence is that the ego vehicle turning causes the OF to behave similarly to when a pedestrian or bicyclist moves across the frame horizontally, as can be seen in Fig. 5.1. This similarity greatly increases the difficulty distinguishing noise, such as turning, from events like pedestrian crossings.

Another issue for $Farneb\ddot{a}ck^{OF}$ caused by the ego vehicle movement is that the difference when a pedestrian enters the frame differs significantly when the ego vehicle is standing still compared to when it is moving or turning. When standing still, the OF is undisturbed except in areas where objects are moving, compared to the constant OF present when the ego vehicle is in motion. This makes filtering the angle difference values described in Section 4.2.3 especially challenging.

(a) (b)

Figure 5.1: A comparison between optical flow on a scene in which the ego-vehicle is turning (a) and standing still while a cyclist is crossing (b). The original image is taken from ZOD [43].

One major challenge for all variants of ML-SFC is the reduced contrast in environments with poor or inconsistent lighting, which makes it difficult for the models to distinguish between salient objects and the background. In low-light environments, noise and artifacts are more prevalent, potentially leading models to make incorrect predictions. An example of this is presented in Fig. 5.2.



(a) (b)

Figure 5.2: A comparison between saliency prediction with a bright (a) and dim (b) scene. Due to low visibility, the saliency model loses track of the person on the bike once they enter the shade. The original image is taken from ZOD [43].

The ML-SFC variants also struggle with busy or complex scenes containing several salient objects, even if they are outside the RoI. This is apparent in Fig. 5.3, where the saliency model fails to highlight a pedestrian crossing the road in cell three since the model's attention mechanism is distributed focus across the entire scene, diluting the saliency strength attributed to any single object. This suggests that ML-SFC may be less effective in environments containing many salient objects, such as urban areas or on highways.

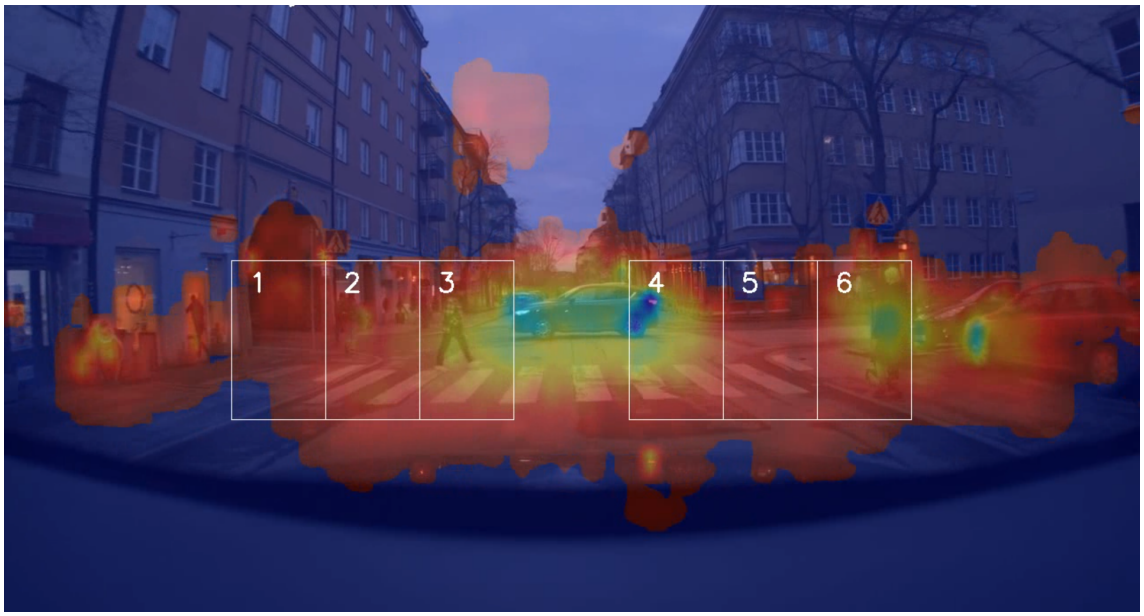


Figure 5.3: An example of a busy scene. The saliency model struggles to capture the pedestrian in cell 3 due to the number of visually prominent objects in the scene. Original footage from ZOD [43].

For this dataset, Zenseact AB has taken all reasonable measures to remove all personally identifiable information, including faces and license plates. To the extent that you like to request removal of specific images from the dataset, please contact privacy@zenseact.com.

Lastly, a general challenge for the ML-SFC variants is distinguishing between different types of objects. Since the output is simply a saliency map of attention, no information is provided about what objects are in the highlighted areas. Practically, this means that a significant amount of noise is created by vehicles in adjacent lanes driving through the RoI grid and attracting a large amount of attention, exceeding the threshold filter described in Section 4.2.2. Several incorrect TPs are due to this factor.

5.4 Comparison of ML-SFC and OF-SFC

The results of our evaluation show that while both ML-SFC and OF-SFC produced comparable outcomes on the synthetic SMIRK dataset, MLNET$^{\text{ML}}$ demonstrated superior performance on the real-world ZOD dataset. Even though the ML-SFC approach also struggled with challenges posed by real-world data, such as varying lighting conditions and busy scenes, the difficulties imposed on Farnebäck$^{\text{OF}}$ through the ego-vehicle movement meant its performance suffered greatly.

Regarding processing speed and efficiency, the variants of ML-SFC were faster than Farnebäck$^{\text{OF}}$ across both datasets. This speed advantage was most apparent on SMIRK, where TranSalNet$^{\text{ML}}$ was significantly faster than Farnebäck$^{\text{OF}}$ but was also present on ZOD. However, the ML-SFC variant with the best efficacy performance, MLNET$^{\text{ML}}$, was only slightly faster than Farnebäck$^{\text{OF}}$ on SMIRK and equivalent on ZOD.

One benefit of Farnebäck^{OF} was its greater reliability and proficiency in tracking object movement. Whereas the variants of ML-SFC could be "distracted" due to other objects in the scene intermittently drawing attention away from, e.g., a pedestrian, Farnebäck^{OF} was able to track multiple objects at once uninterruptedly. This suggests that with the implementation of methods to better distinguish noise caused by ego-vehicle movement, Farnebäck^{OF} may possess greater potential than ML-SFC for future applications.

Overall, the ML-SFC variants showed superior performance regarding both efficacy and efficiency since, despite performing slightly worse in F1 score than Farnebäck^{OF} on SMIRK, it markedly outperformed it on the real-world ZOD footage.

5.5 Implications for Software Engineering

This section will outline a few potential implications our study may have on the broader field of software engineering beyond the scope of traffic event detection and the automotive industry.

This study provides proof-of-concept for how features can be effectively extracted from unstructured data. While specifically employed on traffic video data in this study, the same concepts for feature extraction can be adapted to other contexts or other forms of unstructured data, such as audio. This can be valuable in various applications, including video surveillance, medical imaging, or analysis of satellite footage. By applying similar techniques, software engineers in these areas can reduce the dimensionality of the data and achieve more efficient data processing.

One practical use case of our framework is its potential to generate relevant test scenarios from unstructured data. Creating systems can be challenging without real-world examples to test them on. Future improvements to our approach and its implementation in the industry can enable extracting more relevant test scenarios from unstructured data. This may allow for more realistic and relevant tests to validate systems, such as AD/ADAS in the automotive industry, or other types of systems in other software engineering contexts.

Our research underscores the value of interdisciplinary approaches in addressing software engineering challenges, such as using ML models or computer vision techniques. We hope this leveraging of ML in a software engineering context may encourage software engineers to consider how they can incorporate ML models in their work.

Lastly, this study illustrates that using machine learning models specifically trained for a particular use case, such as pedestrian crossings, is not always necessary. Instead, more general models, like saliency prediction models based on traffic eye-tracking data, can be tailored to specific domains using techniques like the RoI grid.

5.6 Threats to Validity

This section addresses potential threats to the validity of our study, categorized into internal, external, construct, and conclusion threats. Addressing these threats is crucial for transparency of the study's shortcomings and understanding our findings' limitations.

5.6.1 Internal Threats to Validity

One internal threat to validity is the inability to confirm all the videos we evaluated manually. Given the large volume of video data, manual annotation and verification of our evaluation was impractical, which could affect the accuracy of our results. Additionally, the specific videos used from the ZOD dataset might not be a fair representation of the full dataset, even though they were chosen randomly. Another factor is the resolution of the videos, which could have impacted the feature extraction process. Variations in resolution might yield different results, suggesting that either lower- or higher-resolution videos could lead to alternate outcomes.

5.6.2 External Threats to Validity

The generalizability of SMIRK is limited due to its consistent environment across all videos, which may not adequately reflect real-world scenarios. However, this risk is somewhat mitigated by our subsequent experiments on the ZOD dataset. Similarly, the ZOD dataset itself may not be generalizable. Most examples were recorded during the afternoon in sunny and bright conditions in Sweden, with some from Germany and France. These factors may mean that the performance of ML-SFC and OF-SFC does not apply to other parts of the world or other weather and driving conditions.

5.6.3 Construct Threats to Validity

The definition of pedestrian crossing events and our manual annotation process may have introduced inconsistencies within and between the ZOD and SMIRK datasets, particularly when determining if a pedestrian crossing was close enough to be considered a positive event.

5.6.4 Conclusion Threats to Validity

The sample size of the test sets used in the ZOD experiments threatens our findings' statistical power. Considering the trend observed with SMIRK, it is possible that performance may decrease when applied to the full dataset due to a higher likelihood of falsely predicting events in non-crossing videos. Additionally, without manually analyzing all event retrievals, important nuance regarding what caused incorrect prediction might have been missed, potentially affecting the interpretation of the results and causing suboptimal refinement to the framework following the DSR process.

5.7 Future Work

The study's results have opened up several directions for future work. This section will highlight some of the framework's limitations and ways that those limitations could be addressed to improve or extend the work.

5.7.1 Detection of Additional Events

Future work could include extending the work to detect additional traffic events such as cut-ins, merges, near accidents, sudden braking, lane departures, and traffic congestion. Each event requires specific detection strategies and feature extraction techniques to be accurately identified.

This work would involve identifying and defining new RoIs and event detection algorithms specific to the type of event. Additionally, new datasets containing the events and ground-truths would need to be identified.

5.7.2 Exploring Different Models and Techniques

Currently, the framework supports three ML-SFC models and one OF-SFC algorithm. The work could be extended to support more techniques, including ML-based OF such as [6], or it could be extended to explore other OF algorithms and saliency models. Furthermore, other features, such as object detection, bounding boxes, or 3D depth estimation, could be extracted or combined to enhance event detection. To mitigate the issues OF had with ego-vehicle movement, described in Section 5.3.1, it may be possible to compute OF with compensation for the camera movement, like Ramirez, Ohn-Bar, and Trivedi did in their work on OF overtaking detection [28].

5.7.3 Improved Grid

A major limitation of our study that future work may attempt to address is the relation between the RoI grid's shape and placement and distance of events. As addressed previously, the current grid's effectiveness at extracting relevant information depends on the distance of the pedestrian. One potential way to mitigate this issue would be a grid that dynamically changes based on the distance of the objects in the frame, perhaps by using data from depth sensors or depth prediction models.

6

Conclusion

This study has presented a framework for efficiently detecting events in unstructured video data using SFCs. An artifact in the form of a semi-automated pipeline was developed, capable of detecting pedestrians and bikes crossing the road in large volumes of forward-facing dashcam video data.

The work comprised two main parts: extracting semantically relevant, spatiotemporal features and then reducing the dimensionality of these using a SFC for efficient event retrieval. The study investigated two approaches for feature extraction — ML-SFC using ML-based saliency prediction and OF-SFC using OF — and evaluated the findings on two datasets to measure the efficacy and computational efficiency.

Evaluations on a synthetic and real-world dataset, SMIRK [46] and ZOD [43] respectively, demonstrated the framework's capability to identify pedestrian crossing events with promising efficacy and computational efficiency. On SMIRK, ML-SFC using MLNET^{ML} achieved an F1 score of 0.7374, mean IoU of 0.705 and 28.26 FPS and on ZOD an F1 score of 0.5806, mean IoU of 0.4988 and 21.08 FPSt. $\text{Farnebäck}^{\text{OF}}$ achieved an F1 score of 0.8083, mean IoU of 0.7328 and 21.88 FPS on SMIRK, and an F1 score of 0.4545, mean IoU of 0.5143 and 21.05 FPS on ZOD.

The results obtained from the evaluation show that the framework is capable of traffic VED, but that its efficacy is not yet at a level suitable for real-world application. The results for computational efficiency, however, show that the framework can be suitable for real-time application, provided the efficacy is improved and sufficiently powerful hardware is available.

To continue improving and exploring our framework, we recommend several avenues of future work. The first is to extend its application to additional traffic events, like cut-ins or near accidents, by adjusting the RoI grid and the domain-specific configurations. Additionally, the techniques and models used for feature extraction could be expanded upon, perhaps by using different saliency models, ML-based OF models, or extracting other features like bounding boxes. Lastly, addressing the existing limitations of a static RoI grid could allow for more dynamic and flexible event detection, increasing the framework's efficacy.

Bibliography

[1] World Health Organization, *Road traffic injuries*, Accessed on 28 February 2024, Dec. 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

[2] C. Berger, B. Cabrero-Daniel, M. C. Kaya, M. E. Darestani, and H. Shiels, "Systematic evaluation of applying space-filling curves to automotive maneuver detection," 2023.

[3] C. Berger and L. Birkemeyer, *ZEBRA: Z-order curve-based event retrieval approach to efficiently explore automotive data*, 2023. arXiv: 2304.10232 [cs.IR].

[4] Volvo Cars, *Volvo cars*, http://www.volvocars.com, Accessed: 2024-05-06, 2024.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[6] A. Dosovitskiy, P. Fischer, E. Ilg, *et al.*, "FlowNet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015.

[7] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012. DOI: 10.1109/TPAMI.2011.239.

[8] W. Bao, Q. Yu, and Y. Kong, "DRIVE: Deep reinforced accident anticipation with visual explanation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7619–7628.

[9] Y. Yao, M. Xu, Y. Wang, D. J. Crandall, and E. M. Atkins, "Unsupervised traffic accident detection in first-person videos," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 273–280, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:67855477.

[10] A. Jain, H. Patel, L. Nagalapatti, *et al.*, "Overview and importance of data quality for machine learning tasks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '20, Virtual Event, CA, USA: Association for Computing Machinery, 2020, pp. 3561–3562, ISBN: 9781450379984. DOI: 10.1145/3394486.3406477. [Online]. Available: https://doi.org/10.1145/3394486.3406477.

[11] M. v. W. C. Dake, *Neural network*, https://commons.wikimedia.org/wiki/File:Neural_network.svg, Accessed: May 7, 2024, 2006.

[12] A. S. Yakimov, A. Morgun, A. B. Salmina, M. G. Dorrer, A. Tolmacheva, and D. Ogurtsov, "The use of convolutional neural networks to identify artifacts of cells micrographs in biomedical research," *Journal of Physics: Conference Series*, vol. 1399, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:213785374.

[13] A. Esteva, B. Kuprel, R. A. Novoa, *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[14] Y. LeCun, B. Boser, J. S. Denker, *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989. DOI: 10.1162/neco.1989.1.4.541.

[15] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[16] OpenAI, *Chatgpt | openai*, https://openai.com/chatgpt, Accessed: 2024-05-21, 2024.

[17] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*, Springer, 2003, pp. 363–370.

[18] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision (ijcai)," vol. 81, Apr. 1981.

[19] Z. Ren, G. Chen, and W. Lu, "Malware visualization methods based on deep convolution neural networks," *Multimedia Tools and Applications*, vol. 79, pp. 1–19, Apr. 2020. DOI: 10.1007/s11042-019-08310-9.

[20] A. Nordin and A. Telles, "Comparing the locality preservation of z-order curves and hilbert curves," Software Engineering and Management, Bachelor of Science Thesis, University of Gothenburg and Chalmers University of Technology, Gothenburg, Sweden, 2023.

[21] v. W. C. David Eppstein, *Four-level z*, https://commons.wikimedia.org/wiki/File:Four-level_Z.svg, Accessed: May 9, 2024, 2008.

[22] Y. Sergeyev, P. Pugliese, and D. Famularo, "Index information algorithm with local tuning for solving multidimensional global optimization problems with multiextremal constraints," *Mathematical Programming*, vol. 96, pp. 489–512, Jun. 2003. DOI: 10.1007/s10107-003-0372-z.

[23] H. A. Ignatious, M. Khan, *et al.*, "An overview of sensors in autonomous vehicles," *Procedia Computer Science*, vol. 198, pp. 736–741, 2022.

[24] W. Bao, Q. Yu, and Y. Kong, "Uncertainty-based traffic accident anticipation with spatio-temporal relational learning," in *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*, Oct. 2020.

[25] J. Diaz Alonso, E. Ros Vidal, A. Rotter, and M. Muhlenberg, "Lane-change decision aid system based on motion-driven vehicle tracking," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 5, pp. 2736–2746, 2008. DOI: 10.1109/TVT.2008.917220.

[26] Y. Yuan, D. Wang, and Q. Wang, "Anomaly detection in traffic scenes via spatial-aware motion reconstruction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1198–1209, May 2017, ISSN: 1558-0016. DOI: 10.1109/tits.2016.2601655. [Online]. Available: http://dx.doi.org/10.1109/TITS.2016.2601655.

[27] M. Kilicarslan and J. Y. Zheng, "Direct vehicle collision detection from motion in driving video," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1558–1564. DOI: 10.1109/IVS.2017.7995931.

[28] A. Ramirez, E. Ohn-Bar, and M. Trivedi, "Integrating motion and appearance for overtaking vehicle detection," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 96–101. DOI: 10.1109/IVS.2014.6856598.

[29] J. Wang, H. Jiang, Z. Yuan, M.-M. Cheng, X. Hu, and N. Zheng, "Salient object detection: A discriminative regional feature integration approach," *International Journal of Computer Vision*, vol. 123, no. 2, pp. 251–268, Dec. 2016, ISSN: 1573-1405. DOI: 10.1007/s11263-016-0977-3. [Online]. Available: http://dx.doi.org/10.1007/s11263-016-0977-3.

[30] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1155–1162. DOI: 10.1109/CVPR.2013.153.

[31] G. Li and Y. Yu, *Visual saliency based on multiscale deep features*, 2015. arXiv: 1503.08663 [cs.CV].

[32] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "A deep multi-level network for saliency prediction," *CoRR*, vol. abs/1609.01064, 2016. arXiv: 1609.01064. [Online]. Available: http://arxiv.org/abs/1609.01064.

[33] K. Min and J. J. Corso, "Tased-net: Temporally-aggregating spatial encoder-decoder network for video saliency detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019.

[34] J. Lou, H. Lin, D. Marshall, D. Saupe, and H. Liu, "Transalnet: Towards perceptually relevant visual saliency prediction," *Neurocomputing*, vol. 494, pp. 455–467, 2022.

[35] T. Deng, H. Yan, L. Qin, T. Ngo, and B. S. Manjunath, "How do drivers allocate their potential attention? driving fixation prediction via convolutional neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 2146–2154, 2020. DOI: 10.1109/TITS.2019.2915540.

[36] Y. Xia, D. Zhang, J. Kim, K. Nakayama, K. Zipser, and D. Whitney, *Predicting driver attention in critical situations*, 2018. arXiv: 1711.06406 [cs.CV].

[37] M. Perez, J. Sudweeks, E. Sears, *et al.*, "Performance of basic kinematic thresholds in the identification of crash and near-crash events within naturalistic driving data," *Accident; analysis and prevention*, vol. 103, pp. 10–19, Mar. 2017. DOI: 10.1016/j.aap.2017.03.005.

[38] A. Hulbert, T. Kunicki, J. N. Hughes, A. D. Fox, and C. N. Eichelberger, "An experimental study of big spatial data systems," in *2016 IEEE International Conference on Big Data (Big Data)*, 2016, pp. 2664–2671. DOI: 10.1109/BigData.2016.7840909.

[39] J. N. Hughes, A. Annex, C. N. Eichelberger, A. Fox, A. Hulbert, and M. Ronquest, "GeoMesa: a distributed architecture for spatio-temporal fusion," in

Geospatial Informatics, Fusion, and Motion Video Analytics V, M. F. Pellechia, K. Palaniappan, P. J. Doucette, S. L. Dockstader, G. Seetharaman, and P. B. Deignan, Eds., International Society for Optics and Photonics, vol. 9473, SPIE, 2015, 94730F. DOI: 10.1117/12.2177233. [Online]. Available: https://doi.org/10.1117/12.2177233.

[40] M. Bader, *Space-filling Curves. An Introduction With Applications in Scientific Computing*. Jan. 2013, vol. 9, ISBN: 978-3-642-31045-4. DOI: 10.1007/978-3-642-31046-1.

[41] L. Zhou, C. R. Johnson, and D. Weiskopf, *Data-driven space-filling curves*, 2020. arXiv: 2009.06309 [cs.GR].

[42] W. Chen, X. Yao, X. Zhang, and B. Yu, "Efficient deep space filling curve," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2023, pp. 17 525–17 534.

[43] M. Borg, J. Henriksson, K. Socha, *et al.*, *Zenseact open dataset: A large-scale and diverse multimodal dataset for autonomous driving*, The Zenseact Open Dataset (ZOD) is the property of Zenseact AB (©2022 Zenseact AB) and is licensed under the permissive CC BY-SA. For this dataset, Zenseact AB has taken all reasonable measures to remove all personally identifiable information, including faces and license plates. To the extent that you like to request removal of specific images from the dataset, please contact privacy@zenseact.com., 2023. arXiv: 2305.02008.

[44] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004, ISSN: 02767783. [Online]. Available: http://www.jstor.org/stable/25148625 (visited on 05/13/2024).

[45] J. Fang, D. Yan, J. Qiao, J. Xue, H. Wang, and S. Li, "DADA-2000: Can driving accident be predicted by driver attention? analyzed by a benchmark," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 4303–4309. DOI: 10.1109/ITSC.2019.8917218.

[46] M. Borg, J. Henriksson, K. Socha, *et al.*, *Ergo, SMIRK is safe: A safety case for a machine learning component in a pedestrian automatic emergency brake system*, 2022. arXiv: 2204.07874.

[47] AI Sweden, *Highway dataset*, https://www.ai.se/en/labs/data-factory/datasets/highway-dataset, Accessed: 2024-03-04.

[48] D. Bogdoll, *AD4AD: Anomaly Detection for Autonomous Driving*, https://github.com/daniel-bogdoll/phd, 2024.

[49] F.-J. Chang, *awesome-video-anomaly-detection*, https://github.com/fjchange/awesome-video-anomaly-detection, 2024.

[50] *AI Sweden | Nationellt center för tillämpad artificiell intelligens*, https://www.ai.se/sv, Accessed: 2024-05-22.

[51] *Zenseact – Towards zero. Faster.* https://zenseact.com/, Accessed: 2024-05-22.

[52] W. Bao, Q. Yu, and Y. Kong, "Uncertainty-based traffic accident anticipation with spatio-temporal relational learning," in *ACM Multimedia Conference*, May 2020.

[53] O. Kopuklu, J. Zheng, H. Xu, and G. Rigoll, "Driver anomaly detection: A dataset and contrastive learning approach," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 91–100.

[54] Z. Che, B. Jiang, Y. Meng, *et al.*, *D²-City: A Large-Scale Dashcam Video Dataset of Diverse Traffic Scenarios*, version V1, Feb. 2021. DOI: 10.11922/sciencedb.00603.

[55] Y. Yao, X. Wang, M. Xu, *et al.*, "DoTA: Unsupervised detection of traffic anomaly in driving videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 444–459, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:246826180.

[56] S. Karen and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *arXiv preprint arXiv:1409.1556*, 2014.

[57] R. Schubotz, *zCurve: Multi-dimensional indexing using Morton space filling curves.* Version 0.0.4, May 2021. DOI: 10.5281/zenodo.4777584. [Online]. Available: https://github.com/rmrschub/zCurve.

[58] NVIDIA, *Cuda toolkit*, Accessed: 2024-05-17, 2024. [Online]. Available: https://developer.nvidia.com/cuda-toolkit.

A

Computer Specifications

Computer	OS	CPU	GPU	Memory	Storage
PC_1	Ubuntu 22.04.3	Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz	GeForce GTX 1660 Ti	16GB DDR3	120GB SSD
PC_2	Ubuntu 23.10	12th Gen Intel(R) Core(TM) i7-12700K	GeForce RTX 3080	32GB DDR5	512GB M.2 SSD

Table A.1: Computer hardware specifications used for experiments.

B

ZOD Test Sets Descriptions

Video ID	Weather	Lighting	Setting	Bike?	Country code	Busy?
000011	Sunny	Day (bright)	Urban	true	SE	false
000046	Sunny	Day (bright)	Urban	true	SE	false
000113	Cloudy	Day (dim)	Urban	false	SE	false
000169	Sunny	Day (bright)	Urban	false	SE	true
000237	Cloudy	Day (dim)	Urban	false	SE	true
000292	Cloudy	Day (bright)	Urban	false	SE	false
000314	Sunny	Day (bright)	Urban	false	SE	false
000316	Cloudy	Day (dim)	Urban	true	SE	false
000383	Sunny	Day (bright)	Urban	false	DE	true
000389	Sunny	Day (bright)	Urban	false	DE	true
000398	Sunny	Day (bright)	Urban	false	SE	true
000433	Sunny	Day (dim)	Urban	false	SE	true
000521	Cloudy	Day (dim)	Urban	false	SE	true
000653	Sunny	Day (bright)	Urban	false	SE	false
000860	Sunny	Day (bright)	Urban	false	SE	false
000893	Cloudy	Day (dim)	Urban	false	SE	true

Table B.1: Descriptions for the videos used in ZOD_{1-4} containing pedestrians crossings.

Video ID	Weather	Lighting	Setting	Country code	Busy?
000082	Cloudy	Day (bright)	Urban	SE	true
000084	Raining	Day (dim)	Urban	SE	false
000098	Sunny	Day (bright)	Urban	SE	false
000137	Sunny	Day (bright)	Urban	SE	false
000161	Sunny	Day (bright)	Urban	SE	true
000162	Sunny	Day (bright)	Urban	SE	true
000306	Cloudy	Day (dim)	Urban	SE	false
000327	Sunny	Day (bright)	Urban	SE	true
000684	Cloudy	Day (dim)	Highway	SE	false
000864	Sunny	Day (bright)	Urban	SE	false
000865	Sunny	Day (bright)	Urban	SE	false
000870	Cloudy	Day (bright)	Urban	SE	true
000900	Cloudy	Day (dim)	Urban	SE	true
000934	Sunny	Day (bright)	Urban	SE	true
001326	Cloudy	Day (dim)	Urban	SE	false
001352	Raining	Day (dim)	Urban	SE	false
001457	Sunny	Day (bright)	Urban	SE	true

Table B.2: Descriptions for the videos used in ZOD_1 containing no pedestrian crossings.

Video ID	Weather	Lighting	Setting	Country code	Busy?
000007	Sunny	Day (dim)	Urban	SE	false
000019	Sunny	Day (bright)	Highway	SE	false
000236	Cloudy	Day (dim)	Urban	SE	false
000238	Cloudy	Day (dim)	Urban	SE	false
000390	Sunny	Day (bright)	Urban	DE	true
000414	Sunny	Day (bright)	Highway	SE	false
000530	Sunny	Day (bright)	Urban	SE	true
000583	Sunny	Day (bright)	Urban	DE	true
000869	Sunny	Day (bright)	Urban	SE	false
000905	Cloudy	Day (dim)	Urban	SE	false
000935	Sunny	Day (bright)	Urban	SE	false
001012	Sunny	Day (bright)	Urban	DE	true
001091	Raining	Day (dim)	Urban	SE	false
001199	Sunny	Day (bright)	Urban	SE	true
001273	Cloudy	Day (dim)	Urban	SE	true
001294	Sunny	Day (bright)	Urban	SE	false
001245	Sunny	Day (bright)	Urban	SE	false

Table B.3: Descriptions for the videos used in ZOD_2 containing no pedestrian crossings.

Video ID	Weather	Lighting	Setting	Country code	Busy?
000143	Sunny	Day (bright)	Urban	SE	false
000217	Cloudy	Day (bright)	Urban	SE	true
000229	Cloudy	Day (bright)	Urban	SE	true
000230	Cloudy	Day (bright)	Urban	SE	true
000232	Cloudy	Day (bright)	Urban	SE	true
000296	Sunny	Day (bright)	Highway	SE	false
000461	Cloudy	Day (dim)	Urban	SE	true
000541	Sunny	Day (bright)	Urban	SE	true
000603	Cloudy	Day (dim)	Urban	FR	true
000871	Cloudy	Day (bright)	Urban	SE	false
000881	Sunny	Day (bright)	Urban	SE	true
000956	Raining	Day (dim)	Urban	SE	false
000977	Sunny	Day (bright)	Urban	SE	true
001011	Sunny	Day (bright)	Urban	DE	true
001067	Cloudy	Day (dim)	Urban	SE	false
001200	Sunny	Day (bright)	Urban	SE	true
001295	Sunny	Day (bright)	Urban	SE	true

Table B.4: Descriptions for the videos used in ZOD_3 containing no pedestrian crossings.

Video ID	Weather	Lighting	Setting	Country code	Busy?
000024	Sunny	Day (bright)	Urban	SE	false
000168	Sunny	Day (bright)	Urban	SE	true
000231	Cloudy	Day (bright)	Urban	SE	true
000234	Cloudy	Day (bright)	Urban	SE	true
000411	Sunny	Day (bright)	Highway	SE	true
000464	Cloudy	Day (dim)	Urban	SE	false
000614	Cloudy	Day (dim)	Highway	FR	true
000680	Cloudy	Day (dim)	Urban	SE	false
000705	Sunny	Day (dim)	Highway	DE	false
000877	Sunny	Day (bright)	Urban	SE	true
000880	Sunny	Day (bright)	Urban	SE	true
001049	Sunny	Day (bright)	Urban	SE	true
001300	Sunny	Day (bright)	Urban	SE	true
001307	Sunny	Day (bright)	Urban	SE	true
001328	Cloudy	Day (dim)	Urban	SE	false
001341	Sunny	Day (bright)	Urban	SE	true
001412	Sunny	Day (bright)	Urban	SE	false

Table B.5: Descriptions for the videos used in ZOD_4 containing no pedestrian crossings.

C

Full Experiment Results

Subset	Computer ID	Variant	F1 Score	Sensitivity	Specificity	Mean IoU	FPS
SMIRK_{small}	1	MLNET^{ML}	0.8372	0.72	1.0	0.7298	11.70
		$\text{TranSalNet}^{\text{ML}}$	0.9091	1.0	0.8	0.6538	12.41
		$\text{Farnebäck}^{\text{OF}}$	0.8889	0.8	1.0	0.7342	15.16
	2	MLNET^{ML}	0.8372	0.72	1.0	0.7298	25.42
		$\text{TranSalNet}^{\text{ML}}$	0.9091	1.0	0.8	0.6538	33.98
		$\text{Farnebäck}^{\text{OF}}$	0.8889	0.8	1.0	0.7242	22.91
SMIRK_{full}	1	MLNET^{ML}	0.7374	0.6297	0.9883	0.705	12.19
		$\text{TranSalNet}^{\text{ML}}$	0.5304	0.9078	0.7738	0.6699	11.85
		$\text{Farnebäck}^{\text{OF}}$	0.8083	0.6984	0.9956	0.7328	16.19
	2	MLNET^{ML}	0.7374	0.6297	0.9883	0.705	28.26
		$\text{TranSalNet}^{\text{ML}}$	0.5304	0.9078	0.7738	0.6699	46.60
		$\text{Farnebäck}^{\text{OF}}$	0.8083	0.6984	0.9956	0.7328	21.88

Table C.1: The complete results from evaluating the detector on SMIRK. Note that TASED-net was not applied to this dataset due to technical limitations.

Subset	Computer ID	Variant	F1 Score	Sensitivity	Specificity	Mean IoU	FPS
ZOD_1	1	TranSalNet^{ML}	0.4375	0.7	0.3478	0.3553	9.68
		MLNET^{ML}	0.6207	0.6429	0.6842	0.4988	8.83
		Farnebäck^{OF}	0.4545	0.3125	0.9412	0.5143	14.91
		TASED-Net^{ML}	0.2727	0.2143	0.7368	0.3561	5.14
	2	TranSalNet^{ML}	0.6429	0.75	0.6667	0.4827	28.59
		MLNET^{ML}	0.6207	0.6429	0.6842	0.4988	20.71
		Farnebäck^{OF}	0.4545	0.3125	0.9412	0.5143	21.05
		TASED-Net^{ML}	0.2727	0.2143	0.7368	0.3481	14.75
ZOD_2	1	TranSalNet^{ML}	0.4375	0.7	0.3478	0.3553	9.71
		MLNET^{ML}	0.6429	0.6429	0.7368	0.4988	8.92
		Farnebäck^{OF}	0.4545	0.3125	0.9412	0.5143	14.93
		TASED-Net^{ML}	0.2727	0.2143	0.7368	0.3481	5.15
	2	TranSalNet^{ML}	0.6429	0.6429	0.7368	0.4988	21.24
		MLNET^{ML}	0.4865	0.75	0.2381	0.4827	28.14
		Farnebäck^{OF}	0.4545	0.3125	0.9412	0.5143	21.00
		TASED-Net^{ML}	0.2609	0.2143	0.6842	0.3561	15.18
ZOD_3	1	TranSalNet^{ML}	0.4375	0.7	0.3478	0.3553	9.75
		MLNET^{ML}	0.5806	0.6429	0.5789	0.4988	8.90
		Farnebäck^{OF}	0.4167	0.3125	0.8235	0.5143	14.94
		TASED-Net^{ML}	0.25	0.2143	0.6316	0.3561	5.13
	2	TranSalNet^{ML}	0.5806	0.6429	0.5789	0.4988	21.43
		MLNET^{ML}	0.5625	0.75	0.4762	0.4827	25.84
		Farnebäck^{OF}	0.4167	0.3125	0.8235	0.5143	20.92
		TASED-Net^{ML}	0.25	0.2143	0.6316	0.3481	15.14
ZOD_4	1	TranSalNet^{ML}	0.4	0.7	0.2174	0.3553	9.76
		MLNET^{ML}	0.5806	0.6429	0.5789	0.4988	8.87
		Farnebäck^{OF}	0.4348	0.3125	0.8824	0.5143	14.96
		TASED-Net^{ML}	0.25	0.2143	0.6316	0.3561	5.13
	2	TranSalNet^{ML}	0.5806	0.6429	0.5789	0.4988	21.08
		MLNET^{ML}	0.5143	0.75	0.3333	0.4827	28.56
		Farnebäck^{OF}	0.4348	0.3125	0.8824	0.5143	20.97
		TASED-Net^{ML}	0.25	0.2143	0.6316	0.3561	14.70

Table C.2: The complete results from evaluating the detector on ZOD.