A Framework for Automated Traffic Safety Analysis from Video Using Modern Computer Vision

Bornø Jensen, Morten; Ahrnbom, Martin; Kruithof, Maarten; Åström, Karl; Nilsson, Mikael; Ardö, Håkan; Laureshyn, Aliaksei; Johnsson, Carl; Moeslund, Thomas

Published in:
Transportation Research Board Annual Meeting 2019

2019

Document Version:
Other version

A FRAMEWORK FOR AUTOMATED TRAFFIC SAFETY ANALYSIS FROM VIDEO USING MODERN COMPUTER VISION

Morten B. Jensen
Visual Analysis of People Lab, Aalborg University
Aalborg, Denmark. Email: mboj@create.aau.dk

Martin Ahrnbom
Mathematical Imaging Group, Lund University
Lund, Sweden. Email: ahrnbom@marths.lth.se

Maarten Kruithof
TNO
The Hague, The Netherlands. Email: maarten.kruithof@tno.nl

Kalle Åström
Mathematical Imaging Group, Lund University
Lund, Sweden. Email: kalle@marths.lth.se

Mikael Nilsson
Mathematical Imaging Group, Lund University
Lund, Sweden. Email: micken@marths.lth.se

Håkan Ardö
Mathematical Imaging Group, Lund University
Lund, Sweden. Email: ardo@marths.lth.se

Aliaksei Laureshyn
Transport and Roads, Lund University
Lund, Sweden. Email: aliaksei.laureshyn@tft.lth.se

Carl Johnsson
Transport and Roads, Lund University
Lund, Sweden. Email: carl.johnsson@tft.lth.se

Thomas B. Moeslund
Visual Analysis of People Lab, Aalborg University
Aalborg, Denmark. Email: tbm@create.aau.dk

Word Count: 7493 words + 11 figure(s) x 0 + 0 table(s) x 250 = 7493 words

Submission Date: July 30, 2018

ABSTRACT

Traffic surveillance and monitoring are gaining a lot of attention as a result of an increase of vehicles on the road and a desire to minimize accidents. In order to minimize accidents and near-accidents, it is important to be able to judge the safety of a traffic environment. It is possible to perform traffic analysis using large quantities of video data. Computer vision is a great tool for reducing the data, so that only sequences of interest are further analyzed. In this paper, we propose a cross-disciplinary framework for performing automated traffic analysis, from both a computer vision researcher's and traffic researcher's point-of-view. Furthermore, we present STRUDL, an open-source implementation of this framework, that computes trajectories of road users, which we use to automatically find sequences containing critical events of vehicles and vulnerable road users in an traffic intersection, which is an otherwise time-consuming task.

Keywords: Computer vision, data reduction, computer aided analysis, deep learning, surveillance, tracking, detection, traffic analysis

INTRODUCTION

In 2017 more than 25,000 people died and approximately 135,000 people were seriously injured on the roads in the European Union (EU) (*1*). While the numbers are still very high, both injuries and fatalities have been decreasing for decades. Paradoxically, road safety experts worry about the problem of "too few crashes", referring to the difficulties using the traditional safety diagnosing methods as crash counts registered at individual sites become very low (*2*)(*3*). The situation is aggravated by the unresolved problems of crash under-reporting, scarce information about the crash details and conditions in standard police reports and the general retro-active nature of the crash analysis (before safety problem can be diagnosed, it has to manifest itself in form of crashes with people killed or injured).

An alternative or a complementary approach to crash analysis is to use surrogate measures of safety (SMoS). The method rests on the assumption of a continuous relation between the severity of events in traffic and their frequency (*4*), visualized in Figure 1. The fatal and injury crashes are the most severe events and occur relatively seldom, while the events of "normal" severity can be observed in hundreds or thousands every day. The SMoS are normally derived from non-crash events that are close enough to crashes on the severity scale to possess sufficient similarities and thus be relevant for the safety, but much more frequent compared the actual crashes.

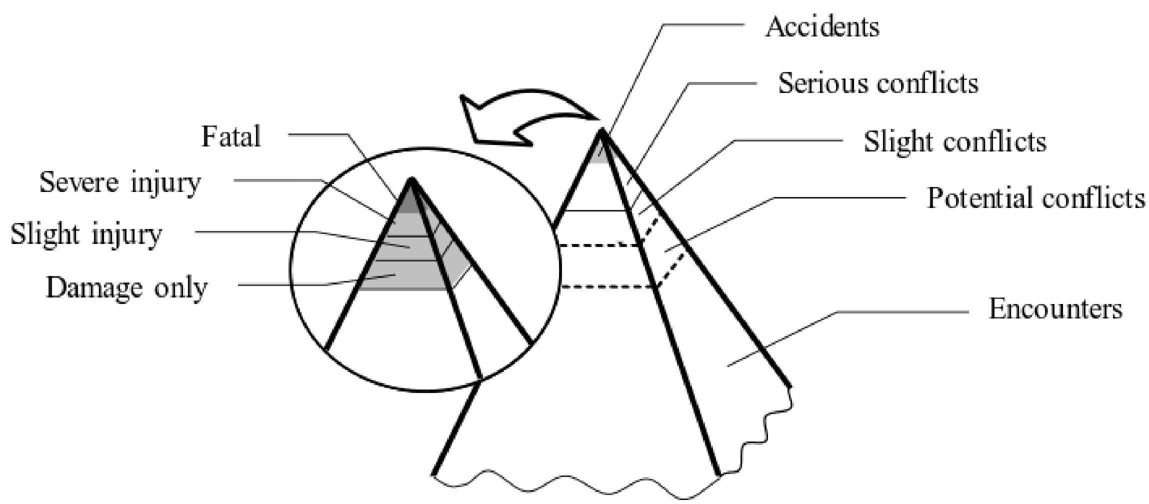

FIGURE 1: "Safety pyramid", adopted from (*4*).

While the idea has been known for decades (*5*)(*6*)(*7*), the lack of an efficient tool to reliably and accurately measure SMoS hindered the method from being used on a large scale. Previously, human observers were tasked to detect, classify and record the relevant events, all in real-time while being in the traffic environment. The high costs of using human observers, as well as some doubts in their reliability were too discouraging.

Automated tools like computer vision are about to change the situation. It is already a very common practice in safety studies based on SMoS to use video recordings either as a complementary documentation for field observations or as a main data source (*8*). With a proper camera perspective and resolution, the measurements of road user positions and speeds taken from video can be very accurate (*9*). Fully automated tools able to detect and track road users in video do already exist and are used (*10*) (*11*) (*12*). The generel concepts of such tools are illustrated in Figure 2. The next challenge is to make the computer vision algorithms more stable while processing longer video sequences that include less favourable conditions such as congested traffic, precipitation, twilight and night, etc. (*13*) while making them more practically usable for traffic research.

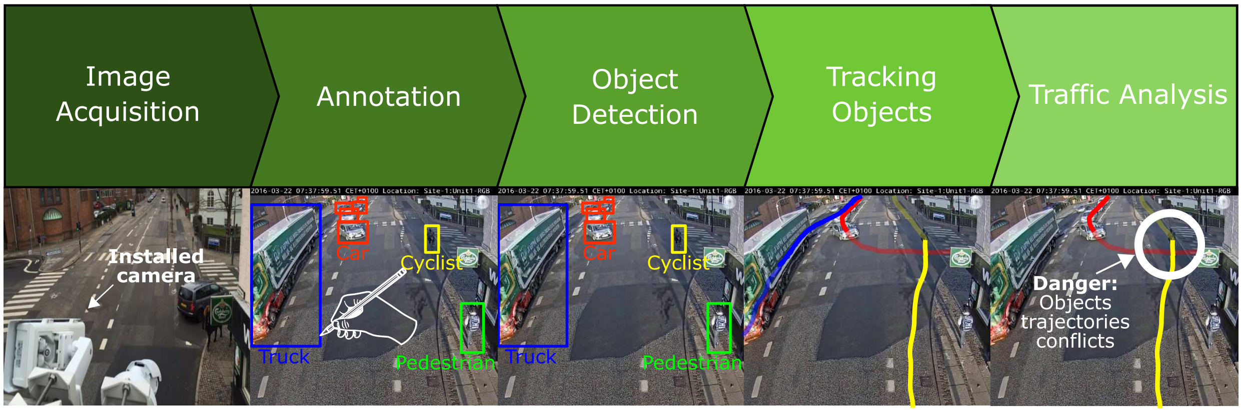

FIGURE 2: General concept for automated traffic analysis. Videos are captured via installed cameras. Humans then annotate some images with bounding boxes, used to train an object detector, which is then run on all the collected videos. The detected objects are then tracked across time, to generate trajectories which can be analyzed to find times of interest or computing SMoS.

Traffic safety and computer vision are two different worlds and the communication between the researchers of these two domains is not always straightforward. The following list summarizes the specific “expectations” from the traffic side that has to be taken into consideration while developing a computer vision tool:

- Majority of the indicators suggested to measure the severity of a traffic event are based on temporal and spatial proximity of the road users. Thus, the most important data to extract from video are the positions and speeds of the road users, complemented with at least rough estimate of their type and size.
- Traffic analysis requires measurements related to the road surface (e.g. speed is to be measured in meters per second rather than pixels per frame) requiring an accurate calibration model.
- Though more frequent than crashes, the events used to calculate SMoS are still relatively rare. Depending on the definition of SMoS chosen, the observation period necessary to collect a sufficient number of the relevant events might vary from 8-10 hours to several weeks.
- The observation period is limited in time, making it common to use temporary installations for the recording equipment but not for the analysis. This put less constrains on the complexity of the video analysis algorithms as they can be processed off-line.
- Traffic environment is a public space and special rules to how the data collected should be handled apply. Ideally, some pre-processing should be done during the recording such the images are cleared from the sensitive information while keeping the relevant data.

Current frameworks trying to bridge the gap between traffic research and computer vision are all based on more traditional computer vision approaches (*14*)(*15*)(*16*)(*17*)(*18*)(*19*). The tra-

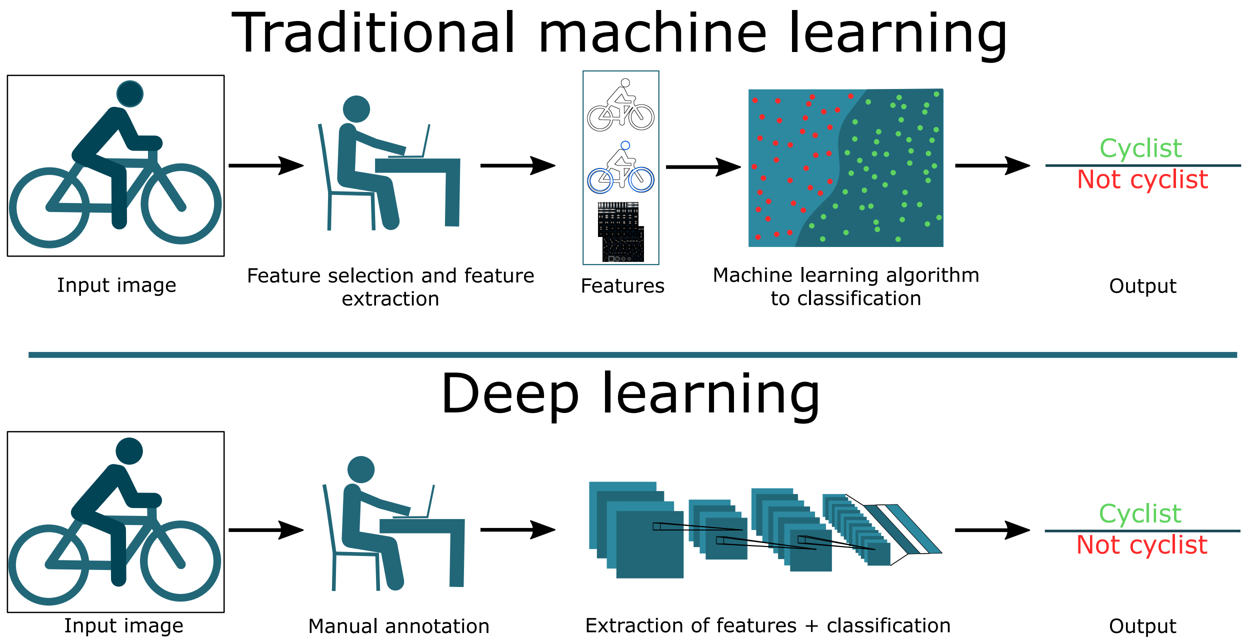

FIGURE 3: Simplified comparison of traditional machine learning approach and deep learning approach for cyclist detection. Note that manual annotations are generally faster and require less specialized knowledge than in traditional machine learning approaches.

ditional approach involve looking for movement in the image or calculating the foreground image which depending on the scene, tells something about the moving objects or foreground objects. To classify the objects, distinctive features, e.g. width, height, color, etc. are used to separate the localized objects. The features varies a lot from object to object, so the used features chosen for classification various correspondingly, but are in this case always manually selected. A traditional machine learning algorithm will then examine all the selected features and maximize the distinction between each of the object's subset of features with the purpose of classifying them. This traditional workflow is illustrated in the upper half of Figure 3.

Computer vision have generally seen a tremendous boost as a result of past decade's hardware improvements, in particular the graphical processing units (GPU) improvements, which have lead to a large use of the well-performing data-driven methods. A very popular data-driven method is deep learning(*20*)(*21*), which is based upon artificial neural networks, which to some extent is an imitation of the human brain. In a computer vision perspective, deep learning is a sub-field of the aforementioned machine learning. It differs from traditional machine learning as it does not require manually selected features. Deep learning is able to learn features that represents a given object automatically from large quantities of annotated data, which is illustrated in Figure 2 and the lower part of Figure 3.

In this paper, we investigate and propose a general data-driven framework to help and ease the cross-disciplinary communication of going from capturing video sequences and automate the traffic analysis generation using deep learning. Furthermore, we present an open-source implementation of the introduced framework.

The contributions of this paper are thus two-fold:

- Introducing and defining a data-driven cross-disciplinary framework for performing automated traffic analysis, from video acquisition to traffic analysis.

- An implementation of this framework that can detect, classify, track, and create a traffic analysis of data from an intersection.

Our implementation is released as open source and is available here: https://github.com/ahrnbom/strudl. This program is designed to be easy to use for traffic researchers, without extensive knowledge in computer vision.

RELATED WORK

In this section, we present recent and relevant work done related to defining a cross-disciplinary framework for easing collaborations between traffic researchers and computer vision researchers. The section will be split into 2 parts: a part containing general established frameworks followed by relevant work and applications where computer vision has aided traffic researchers.

General frameworks

From a computer vision perspective, several frameworks have been proposed to fit the development of most general computer vision systems. In (*14*), a general framework is defined which is applicable for most systems working with video. The framework consists of the following blocks: camera, image acquisition, pre-processing, segmentation, representation, and classification. Given a set of images acquired with one or several cameras, is is possible to classify e.g. objects and actions, by the use of various mathematical operations. In (*15*) a video-based system for automated pedestrian conflict analysis is introduced following 5 basic components: video pre-processing, feature processing, grouping, high-level object processing, and information extraction. Compared to (*14*), these components are more angled towards a high-level information extraction which can be considered more applicable for a traffic researcher.

In (*16*) a comprehensive review of computer vision techniques used for analysis in urban traffic is presented. They propose two different approaches to automated traffic analysis. Both of them takes an input frame as starting point, but differ in structure by one being a top-down approach and the other a bottom-up approach. The top-down approach consist of estimating the foreground of the frame, e.g. by frame differentiation (*17*). A grouping of connected foreground pixels is done, e.g. connected component analysis, which constitutes the objects. These objects are classified (*18*), which can be based on heuristically predefined rules or by use of training data. Finally, tracking translate the objects into spatial-temporal domain, which provides the user with object trajectories (*19*). As described in (*16*), the top-down approach analyzes the objects as a whole, whereas the bottom-up approach takes its starting point in using smaller patches of the image to detect a part of the objects, e.g. scale invariant feature transform (*22*) and histogram of oriented gradients (*23*). The detected parts of the objects are afterwards grouped together to form an object constituting the object detection step. Object detection can be extended with a classification step, where the individual object is assigned to a specific class label. Finally, the objects is tracked with the purpose of creating object trajectories.

The available cross-disciplinary frameworks are in general feature-based and model-based, which have been very common prior to the hardware improvements made the past decade. GPUs in particular have made training of complex artificial neural networks possible. The artificial neural network is inspired by the neural networks found in the human brain. The recent trend in computer vision is the usage of artificial neural networks, often referred to as deep learning, to do object detection by learning and adjusting the parameteres and weights in the network by exposing it to large quantities of annotated data. Genreally, deep learning is outperforming traditional methods by a large margin (*20, 21*). The current available cross-disciplinary frameworks do not use deep learning, making our proposed framework the first to take advantage of this significant improvement in technology.

Automated video-based traffic applications

The related video-based traffic applications are split into two categories, which are object detecting and conflict-based data reduction.

Object counting

Object counting in relation to the traffic domain primarily consists of firstly detecting and classifying the object of interest, e.g. cars, trucks, pedestrians and cyclists, followed by tracking them to prevent counting the same object multiple times and to cope with potential occlusion. A lot of work has been done in especially detecting and classifying objects, in (*24*) they build upon the well-known Haar-like features (*25*), which have traditionally been used for single-frame detection. By computing such features in temporal space, the motion can estimated by comparing the absolute differences between the values in the spatial-temporal domain. Detecting and classifying objects have traditionally been based upon the RGB modality. In relation to traffic analysis, this can cause challenges as RGB is vulnerable changing weather and light conditions. To make a system more robust, the thermal modality can be introduced to complement the RGB camera, in a so-called multi-modal setup (*13*). In (*26*), object classification is done based on images captured from multiple visual traffic surveillance sensors, providing a multi-view setup which is less prone to occlusion.

As previously mentioned, the recent years of object detection has followed the hardware improvements, leading to a large use of the well-performing deep learning methods (*20*). Most of the object detectors using deep learning methods, e.g. convolutional neural networks (CNN), relies on supervised learning, meaning that large quantities of annotated data is needed to train the CNN (*27, 28*). In (*29*) a CNN was applied to the popular ImageNet Large-Scale Visual Recognition Challenge, which is a popular object recognition benchmark containing 1.2 million training images, 50,000 validation images, and 150,000 testing images. The CNN nearly halved the top-5 error rate of object recognition generated from traditional computer vision methods (*21*).

In general, for most of the aforementioned methods, the found objects can be tracked by using nearest neighbour, Kanade-Lucas-Tomasi feature tracker (*15, 30, 31*), or by the use of more complex feature based methods such as a Kalman filter (*32*) or Hungerian tracking (*33*), which have proven quite useful in a wide variety of applications.

Conflict-based data reduction

Computer vison software can greatly speed-up the process of reducing a captured video dataset to only the sequences of interest, as manually analyzing large quantities of data is a time-consuming task. In (*15*) pedestrians and motorized traffics are detected, tracked and classified, and then used to identify critical events in the video. The critical events are in (*15*) defined as "*any event that involves a crossing pedestrian and a conflicting vehicle in which there exists a conceivable chain of events that could lead to a collision between these road users*", resulting in all the detected objects intersecting trajectories triggering an important event, which is similar to the illustration to the traffic analysis step in figure 2. All the triggered events can be split into a subset of important conflict indicators: Time-to-Collision, Post-Encroachment Time, Gap Time, and Deceleration-to-Safety Time, which can be used the measure the severity of the event.

In (*34, 35*) the human-in-the-loop framework is further cultivated as a graphical user interface is developed to enable traffic researchers to utilize computer vision methods. The traffic researchers can annotate areas of interest on the input video, which are triggered by activity. Combining multiple of these annotated areas in a timed logic, e.g. potential conflict between cyclist and a right-turning vehicle, can then be used to trigger an interesting event flag.

To the best of our knowledge, our proposed framework is the first cross-disciplinary framework to use modern deep learning for data reduction in traffic surveillance, with an open source and free implementation designed to be used by traffic researchers with limited knowledge of deep learning and computer vision.

FRAMEWORK OVERVIEW

In order for a data reduction framework for traffic surveillance to be useful in practise, it needs to be general enough to be able to handle different kinds of queries and criteria. A single cross-disciplinary computer vision framework can work for multiple applications, as the main steps that need to be performed are typically the same. The proposed framework in this paper takes its spawn in a top-down approach, as presented in (*16*), and some of the general concepts presented in (*14, 15*). In Figure 4, the proposed framework is illustrated in a block flow diagram. Each block forms the structure for the following of this section and will thus be described accordingly.

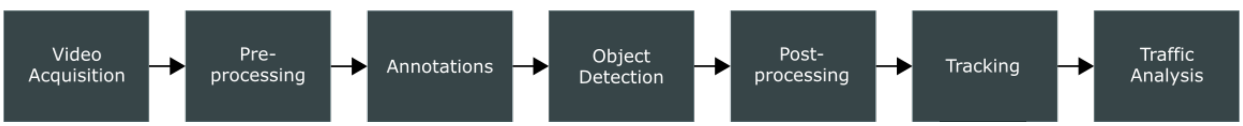

FIGURE 4: The proposed cross-disciplinary framework for automated traffic analysis. While Video Acquisition and Traffic Analysis can be considered to belong to the field of traffic research, the remaining central blocks belong to the field of computer vision.

Video Acquisition

The first step in the general framework, seen in figure 4, is video acquisition. In this step the primary goal to acquire video data to the pipeline. Essential considerations to do this is presented in the following subsection.

Modalities

The most common sensor for acquiring video data is a traditional RGB camera, which is similar to the human eye making the videos easy to interpret and work with. As mentioned in the related work, other options include using a thermal camera, which during the last decade have seen a price reduction making it feasible to use in traffic surveillance applications (*36*). A thermal camera is a passive sensor that captures the infrared radiation emitted by all objects, which can be translated to "seeing" the temperature in a given scene. The thermal camera are thus usable in the night which can be an advantage compared to RGB, but can also be considered a disadvantage as the lack of color information can make classification challenging. An example of the two modalities is seen in Figure 5, where both modalities are used in a challenging rainy night-time scene.

The choice of modality, e.g. RGB or thermal, does not affect the rest of the suggested framework, the choice comes down to a matter of cost, expected light and weather conditions, and privacy concerns. Specifications of the sensor should be taken into consideration, e.g. FPS and resolution.

(a) RGB Camera. (b) Thermal Camera.

FIGURE 5: Comparison of the RGB modality and thermal modality captured at a traffic intersection doing a rainy night. Notice the strong reflections in the RGB image, as well as the poor contrast in the thermal image. This is an example of a situation where none of the modalities are optimal.

Camera calibration

By carefully measuring the positions of some points visible in the camera, the camera can be calibrated, allowing positions in pixel coordinates in the images to be translated to world coordinates. If this step is omitted, any results found by computer vision algorithms are significantly more difficult to interpret and use since they cannot be converted to world coordinates. Detailed search queries and SMoS typically need to be computed in world coordinates to be useful.

Pre-processing

Modern object detectors using CNNs do not need much pre-processing. The only form of pre-processing used in our framework is masking. Often, the entire scene captured by the camera is not of interest; if an application is to find interesting situations in a crossing, then it is of no importance what happens far from that crossing. For these cases, a manually drawn "do-not-care" zone is created as an overlaying mask as exemplified in figure 6. This speeds up annotations and helps training a reliable object detector. If this step is omitted, and only parts of the images are annotated, this may confuse the detector during training, possibly leading to reduced accuracy.

(a) RGB Camera. (b) Overlaying mask.

FIGURE 6: Manually annotated "do-not-care" zone which is overlayed on the input image as a mask.

Annotations

In order to run modern object detectors based on deep learning, manually creating annotations is necessary. CNNs learn by examples, so a human needs to define and annotate this example many times before the network can be trained to do the same. In this pipeline, neural networks are used for object detection only, so the annotations consist entirely of marking objects in images, by bounding boxes and assigning a class label to each box, as illustrated in Figure 7.

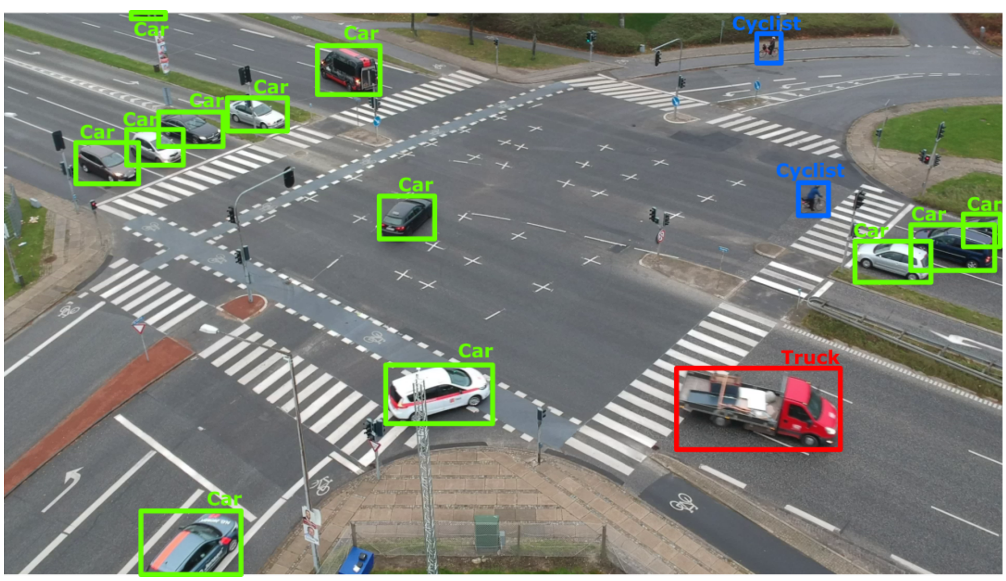

FIGURE 7: Bounding box annotations with belonging class label.

It is important that all visible objects (after applying the mask defined in Section 4.2) are annotated. Otherwise, these will be considered negative examples when the detector is trained. For example, if one car is marked as a "car" and another one is not, then the detector will have a hard time understanding why one is considered a "car" while the other one is not, and accuracy may suffer as a result.

If a large dataset of traffic images were annotated and made publicly available, they could be used if their viewing angle, lightning conditions etc. are reasonably similar to the new data. In such a case, a smaller amount (or none at all) new data may need to be annotated. Despite the current lack of such a dataset, pre-training the networks on general images reduces the number of image annotations needed to a couple of hundred, as opposed to thousands or more.

Object Detection

The goal of object detection is to find "objects", e.g. road users, as axis-aligned bounding boxes with class labels in an image. Traditionally this step has been split into two steps: localization (finding the bounding box) and classification (assigning a class to the bounding box), but due to recent years' advancements in CNN designs, both can be performed in a single step. The choice of class labels is application dependent and is not limited to a specific amount. Multiple can be used, if they are of particular interest, but it should be noted that a significant amount of examples has to appear in the annotated images for the detector to become accurate.

Post-processing

In the post-processing step, the movement direction for each of the detected objects can be computed and converted to world coordinates. Movement directions are useful cues when connecting the detections into tracks. Performing the tracking in world coordinates has benefits, mainly being more independent of the viewing angle, and working directly in natural units and world coordinates allows more detailed and natural track analysis.

Tracking

The tracking step consists of connecting the detected objects in spatio-temporal space, meaning that each detected object in the video needs to be either associated with a previously existing track or as a completely new track in the video. Though this might sound as a somewhat easy task, several challenges are introduced when objects radically change direction or if multiple objects get too close to each other in the sensor's field-of-view.

The performance of the object detection is critical for proper tracking as trajectories cannot be generated for objects that are not detected. Tracking can, however, compensate for some issues in the detector. For example, if a vehicle is detected in only 1 frame, but not in any of prior or following frames, there is a high probability that this is a false detection. If an object is not detected in a small number of frames, but is detected before and after, the tracking algorithm may be able to understand that it is indeed the same object.

Selecting a sensor with too low FPS results in objects in the scene moving a large distance between the consecutive captured frames, which can make it harder to connect the detected objects in spatio-temporal space. Using a high FPS, the objects' movement between consecutive captured frames becomes less, which generally makes tracking easier. Videos with 15 FPS seem to work in our experiments.

Traffic Analysis

The final step of the proposed framework is to analyze the road user tracks with respect to safety. For example, indicators like Time-to-Collision and Post-Encroachment Time can be calculated and events with severity above a certain threshold can be detected and presented to the user for further examination. The data about the distribution of events within different severity categories can be used by special statistical methods such as extreme-value theory in order to estimate the expected number of crashes (*37*) (*38*). Also, trajectory data can be used for calculations of advanced exposure measures, for example a number of encounters between road users of a certain type and performing a certain manoeuvre (*39*). Clustering of the trajectories and detection of deviant trajectories do not fit into any of clusters may reveal the abnormal incidents such as movement in wrong direction or stop at an unusual place.

Since the tracks are computed in world coordinates, thresholds, safety measures and other criteria can be expressed in natural terms and units. While traditional computer vision systems allow only simple criteria (typically expressed in pixel coordinates), world coordinate tracks allow for arbitrarily complex queries, that are more meaningful from a traffic analysis perspective

EXPERIMENTS

As a part of a traffic analysis project, an intersection with a crossing of interest in Malmö, Sweden was filmed for 24 hours with a thermal camera. TSAI calibration(*40*) was computed by measuring 57 points visible in the videos. People were hired to watch through the entire 24 hours of video, tasked to find times in the video where both a car and either a pedestrian or a bicyclist are visible at the same time, where the car will at some point make a turn to pass the crossing of interest, while the pedestrian/bicyclist will at some point pass the crossing. See Figure 8 for a visual explanation of the task. These times were then inspected in more detail by traffic analysts. We stress that this is not a "toy problem"; the human watchers were required as a starting point for further traffic research at this intersection, and we hope that the existence of this framework can reduce the need for human labor in situations like these in the future.

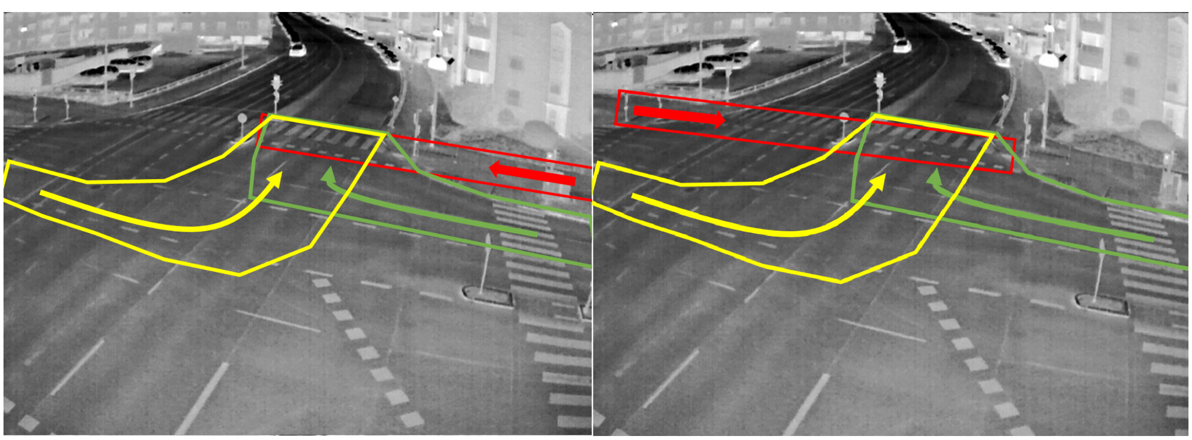

FIGURE 8: The goal is to find times when a vulnerable road user is moving through the red regions in the marked directions, while a car is moving either through the green or yellow regions simultaneously.

An implementation of the suggested framework was used to perform the same task, using the human observer's results as ground-truth. As a baseline, the Road User Behaviour Analysis

(RUBA) software (*34*), which is a traffic analysis tool based on traditional computer vision technology, was also tested for the same task.

There is some ambiguity in when exactly during an encounter it is detected by an observer or computer vision tools. Therefore, it was allowed for some time discrepancy for a detection to be counted as correct. By testing multiple time distance thresholds between the ground truth and the output of the automatic systems, a trade-off between precision and recall can be observed. We use precision and recall curves to visualize this trade-off and compare the automatic systems.

STRUDL: description of implementation

This section describes how our framework following the definitions in Section 4 was implemented, in order to solve the problem described above. The implementation is called **S**urveillance **Tr**acking **U**sing **D**eep **L**earning (STRUDL). It can be used in any context where objects seen from a static camera need to be tracked. Those tracks can be analyzed to for example find times of interest. While thermal videos were used in this experiment, the STRUDL system works with RGB as well (and should in fact perform better with RGB as the pre-training of the object detector is made with RGB images). The remaining parts of this section will describe in more detail how STRUDL implements the computer vision parts of the suggested framework.

Pre-processing

With modern object detection algorithms based on CNN, very little pre-processing of images is necessary. The only pre-processing done is applying a visual "do-not-care" mask.

Annotation

500 frames were selected from the collected videos and annotated manually with bounding boxes and class labels. The frames were taken from 25 randomly selected 5 minute clips, and from each such clip, 20 frames were sampled evenly. This way, there should be a large variety in the road users appearing in the images. A variant of Extreme Clicking (*41*) was implemented to make the annotation process fast. The reason why 500 frames is sufficient to get decent object detection performance is that the detector is pre-trained on a general objects detection task. Training the object detector from scratch would require drastically many more images.

Object Detection

The object detector SSD (*42*) was used. It is a commonly used CNN for the object detection task for its reasonable trade-off between accuracy and execution speed. On a powerful modern GPU, it runs in around real-time. The objects found are presented as axis-aligned bounding boxes. The SSD network was pre-trained on the large MS COCO dataset(*43*), which contains a large amount of images with bounding box annotations of many different kinds of objects (not only traffic-related ones), made by human annotators. Then, the network was fine-tuned on images from the videos for which the experiment is conducted, as described in Section 5.1.2. Finally, the object detector is applied to every single image, and detected objects are stored.

Post-processing

For the videos, the OpenCV function `goodFeaturesToTrack` was used to find points which can be tracked, and then by repeatedly using the OpenCV function `calcOpticalFlowPyrLK`(*44*), those points were turned into point tracks. These tend to follow how objects move in the scene. For each detected bounding box, the average movement direction of point tracks moving through the box were computed, giving each box a movement direction.

Then, using a TSAI camera calibration model (*40*), each such box and movement direction were converted to world coordinates. Because of the pixel-aligned nature of bounding boxes, only the center point was converted.Because the orientation of road users can be computed from their movements directions, and the class labels allow approximate 3D models to be inserted in their place, any information about the movements, position and spatial extent of the road users should be possible to obtain, at least approximately, from this simple representation.

Tracking

A simple Hungarian tracker (*33*) was used, using class consistency, position in world coordinates and movement direction to compute the association cost. World coordinate detections that were not associated to any existing tracks, were made into tracks of their own unless they were too close to some already existing track. When no detection were associated with a given track, the track continues along its previous direction for some time until being removed, unless it is associated with a new detection before that. Tracks that were short-lived, that were only associated with one or two detections were removed, as they are often false or unreliable tracks.

The tracking requires tuning of 13 parameters, which were optimized using a blackbox optimization scheme for a short video clip (15 seconds long) where ground truth tracks in world coordinates were created for each road user, which took around 30-40 minutes of human labor to create. Because the tracks are in world coordinates, it is believed to be possible to re-use the optimized parameters for a different viewpoint, perhaps with minor changes.

Traffic Analysis

The goal was to find times when at least two tracks are visible at the same time while the two tracks intersects at some point, e.g. car move to turn and cross the vulnerable road user track. To implement this as a traffic analysis program, mask images were drawn which mark the interesting regions, and the tracks were tested to see if they at some points move through the marked regions. The mask images can be seen in Figure 9.

Results

The results are seen in Figure 10, where the proposed system is compared to RUBA (*34*). RUBA's raw output was compared directly, and after seeing that the number of false positives were very high (leading to a low precision), time was spent to remove 967 of RUBA's found situations by manually examination ("RUBA+human" in the figure). Most of all the removed events were indeed false positives, as the recall drops very little in this process. Even so, the number of false positives remain high for left-turning cars. The manual time spent with RUBA was around three hours, where around 90 minutes were spent manually removing false positives. Our system, on the other hand, required only around two hours of manual work constructing the detection annotations, and around 30-40 minutes spent on tracking ground truth. Also note that the human time can decrease as the software becomes more used, allowing training images from similar viewing angles to be re-used, and tracking parameters might be possible to transfer with little to no changes, because they are expressed in world coordinates. Furthermore, for a human to make annotations, little training is required, while designing hitboxes and thresholds for RUBA requires experience and familiarity with the software.

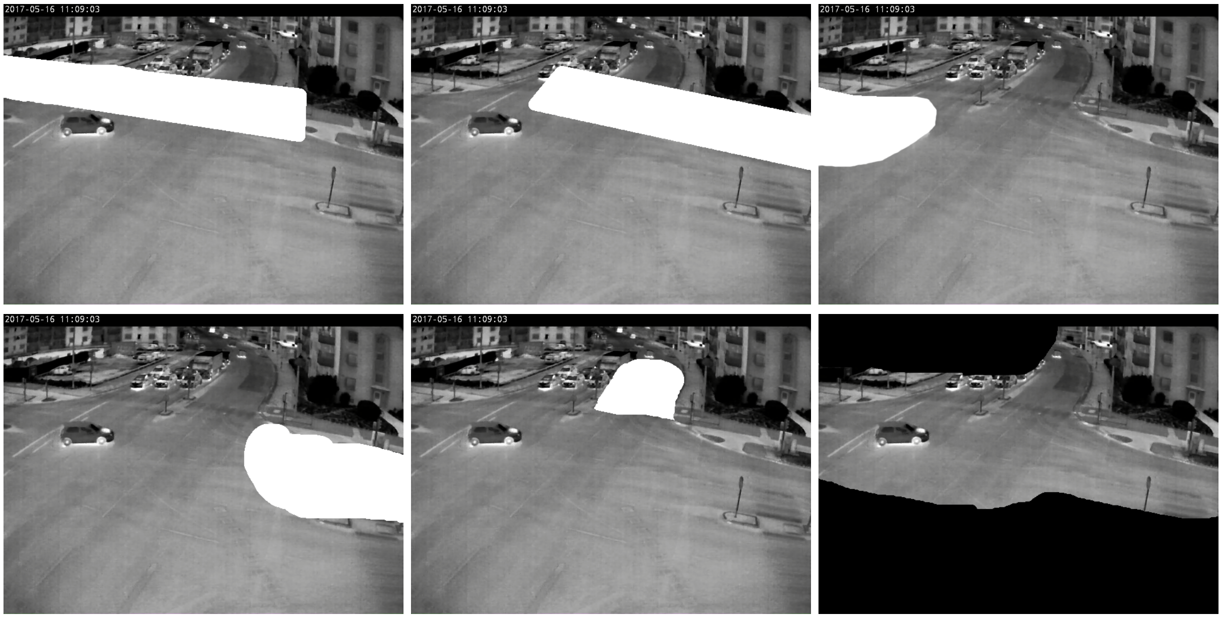

FIGURE 9: Checking masks used for the experiment. Top left: VRUs moving to the right. Top center: VRUs moving to the left. Top right: first required position of left-turning cars. Bottom left: first required position for right-turning cars. Bottom center: the last required position of both left-turning and right-turning cars. Bottom right: mask used during object detection annotations, in order to save annotation time. The same mask is used when running the object detector.

It should be noted that the problem was significantly more difficult for left-turning cars than for right-turning cars. The exact cause for this is not yet known. Only 10 situations with left-turning cars were marked as interesting by the human annotators, compared to 331 for right-turning cars during this one day of video.

We stress that this comparison between STRUDL and RUBA does not include the main difference between the two; while RUBA provides only "take-it-or-leave-it" times of interest, STRUDL provides full tracks in world coordinates that can be further analyzed, by e.g. computing SMoS, sorting by severity or further filtered.

Some tracking example results can be seen in Figure 11. The tracking generally works well, but there is also some room for improvement in its robustness for some tracks.

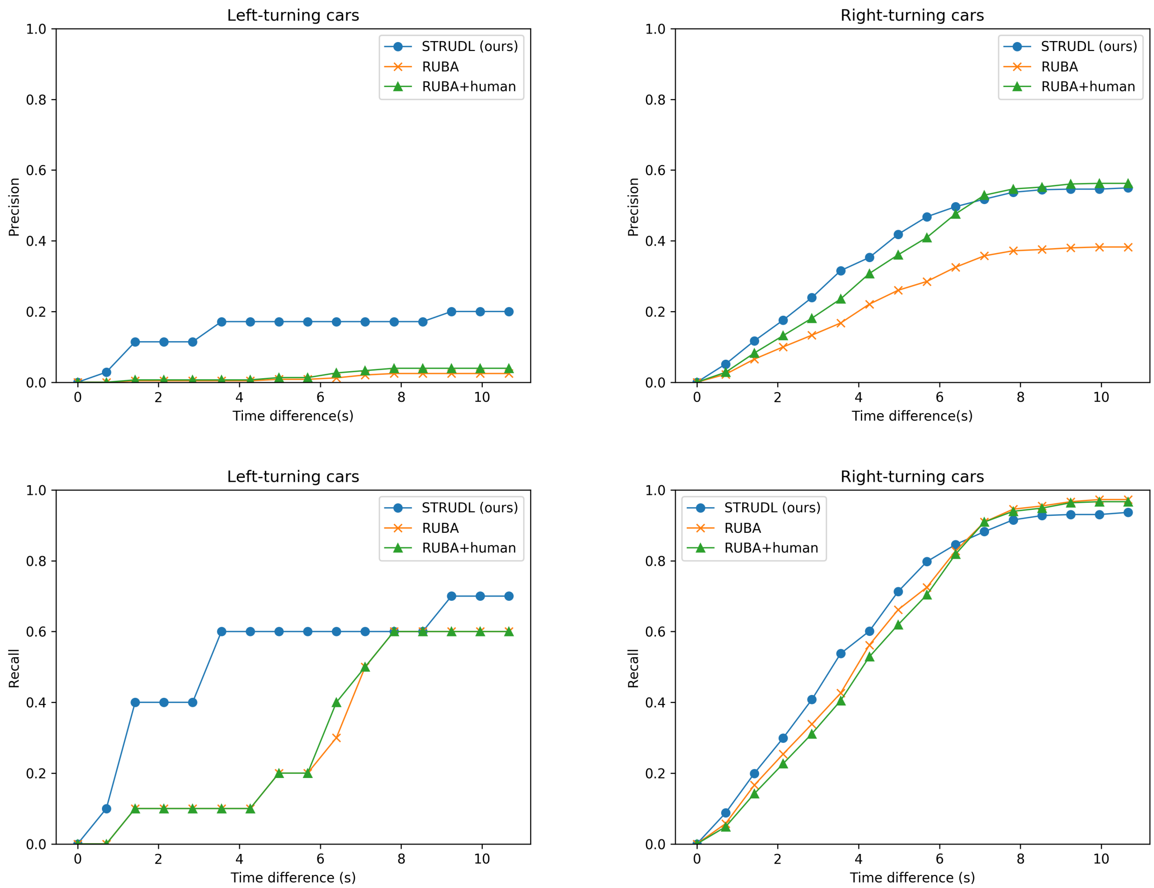

FIGURE 10: Precision and recall for the experiment, against the time difference for which a detected time can differ from the ground truth time and still be considered correct. RUBA+human reaches STRUDL's precision for right-turning cars, while STRUDL is still better for left-turning cars. For recall, they perform similarly for right-turning cars, and are able to find more than 95% of the ground truth times within ± 10 s, while for left-turning cars, STRUDL's recall is clearly better for short time differences, while being only slightly better for longer time differences.

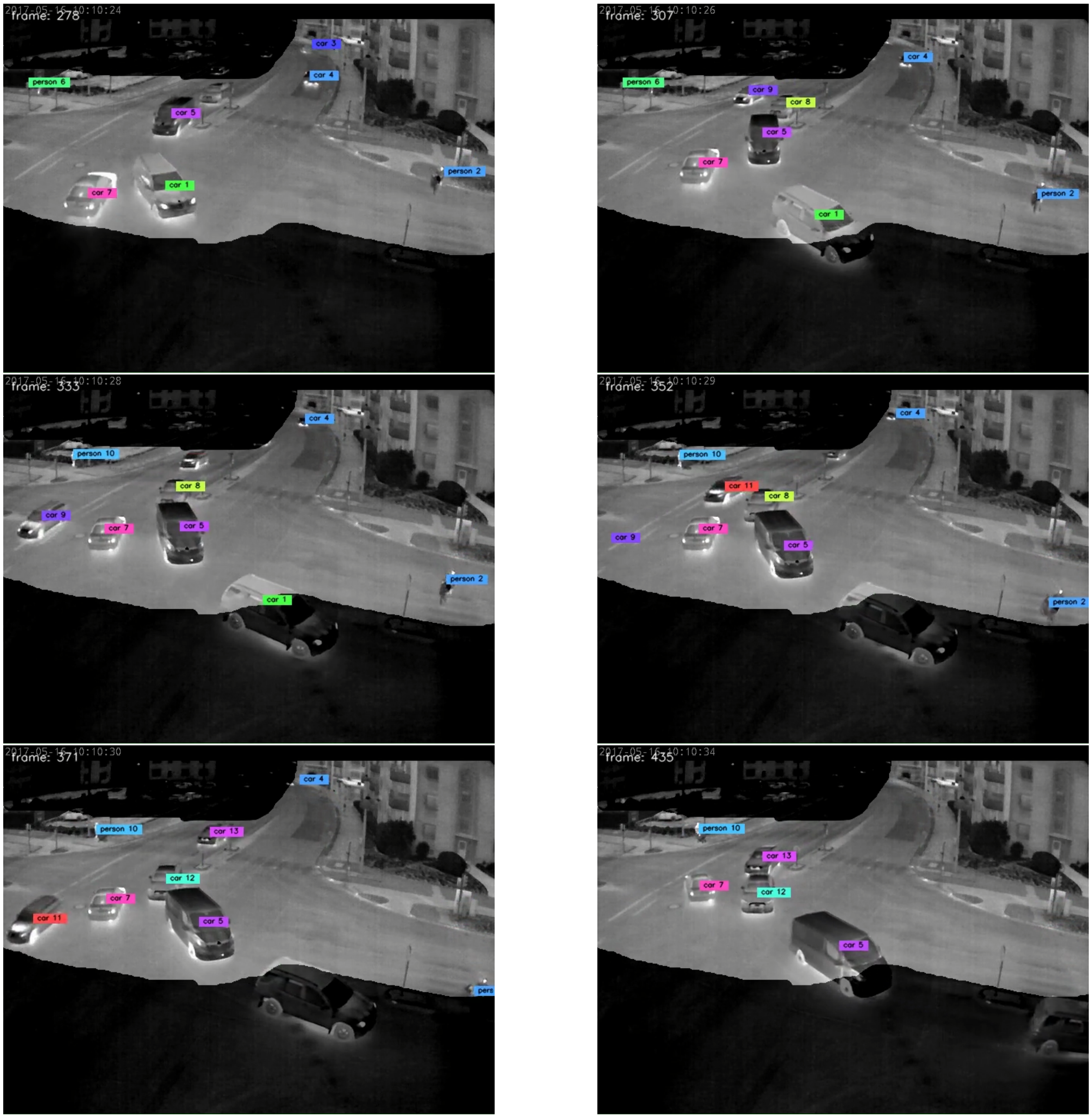

FIGURE 11: Example of tracking results from our experiments. An example of bad tracking is the person walking with a stroller, who is broken up into two tracks. "Car 9" is lost and it takes a few frames for the tracking algorithm to remove this track. For the most part however, tracking works as expected. The dark areas are the masked "do-not-care" zones. Best viewed in color.

DISCUSSION

The experimental results show that the proposed framework works, and the STRUDL implementation is better than traditional approaches for tasks of this kind. It is flexible, meaning that if one is dissatisfied with the results for a given problem, the path forward is often clear. If the object detector makes too many mistakes, more training data can be provided. If the tracking fails too often, the parameters can be tuned, manually or via data-driven optimization. If there are too many false positives, the analysis criteria can be modified with relatively little effort. Visualizations of the different steps of the computer vision pipeline make it easy to pinpoint where issues arise.

More importantly, where traditional computer vision system have a limited range of possible operations, the richness of full trajectories allow for much more freedom. It is possible to compute SMoS or other measures of interest, to filter or sort the detected situations by severity. The proposed system can therefore be seen as a starting point for arbitrarily complex traffic analysis, whereas traditional methods are essentially of a take-it-or-leave-it nature, impossible or difficult to further analyze, filter, sort and work with.

The proposed automatic system needs some human assistance, mainly in annotating image data to train the object detector. When looking at new video data, the amount of new annotations necessary will depend heavily on previously available data. Manually annotating some images seem like a good trade-off, as opposed to traditional methods requiring time-consuming parameter tuning, as it is relatively simple and fast, and if multiple somewhat similar views are studied, annotations from one view can be re-used, reducing the annotation time per intersection.

One limitation of the proposed framework is the tracking algorithm, which is quite simple in nature. It is known to sometimes make mistakes when tracks get too close to each other, or if the detector fails to locate an object for many frames. These flaws could possibly be fixed or reduced by letting a neural network perform the tracking, but that would require a large amount of annotated ground-truth tracks for training which take time to produce. Our implementation requires little to no annotated ground-truth tracks, since tracking parameters should be mostly transferable between views. Still, it would be of interest to test and compare different tracking algorithms for this setting. The modular implementation of STRUDL makes it relatively simple to replace the current tracking algorithm, should so be needed. Another limitation is the lack of uncertainty measures in the STRUDL software. There is no universally accepted standard for how to measure the certainty of detections and tracks, but some combination of detection confidence and the similarities between every track and typical trajectories could probably be used for this purpose. This is one promising direction for future work.

The implementation of STRUDL is designed with flexibility in mind and it is our intention to continually improve the software. For example, it would be useful to have built-in support for computing SMoS, or make improvements to its computer vision algorithms, tracking in particular. We also hope that other implementations of the proposed framework will arise, to suit the specific needs of different traffic analysis problems.

CONCLUSION

We present the, to the best of our knowledge, first cross-disciplinary framework for automated traffic surveillance analysis to take advantage recent improvements in data-driven deep-learning. Through experiments with our open-source implementation, STRUDL, we show better results than traditional systems, while opening new possibilities by providing full trajectories in world coordinates, allowing arbitrarily complex traffic analysis. Promising future works includes computing certainty measures and SMoS automatically and improving the stability of tracking.

ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 635895. This publication reflects only the authors' view. The European Commission is not responsible for any use that may be made of the information it contains.

AUTHOR CONTRIBUTION STATEMENT

The authors confirm contribution to the paper as follows: study conception and design: Morten B. Jensen, Martin Ahrnbom, Maarten Kruithof; the framework design: Morten B. Jensen, Martin Ahrnbom, Maarten Kruithof; the STRUDL experiment: Martin Ahrnbom; analysis and interpretation of results: Martin Ahrnbom, Carl Johnsson; draft manuscript preparation: Morten B. Jensen, Martin Ahrnbom, Aliaksei Laureshyn. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

[1] European Commission, *2017 road safety statistics: What is behind the figures? - Fact Sheet*. European Commission, 2018.

[2] Chang, A., N. Saunier, and A. Laureshyn, Proactive methods for road safety analysis. Vol. SAE International. White paper, 2017.

[3] Laureshyn, A., C. Johnsson, T. Madsen, A. Várhelyi, M. de Goede, A. Svensson, N. Saunier, and W. van Haperen, Exploration of a method to validate surrogate safety measures with a focus on vulnerable road users. *International Conference on Road Safety and Simulation*, 2017.

[4] Hydén, C., The development of a method for traffic safety evaluation: the Swedish traffic conflict technique. *Department of Traffic Planning and Engineering. Bulletin 70*, Vol. Doctoral thesis. Lund University, 1987.

[5] Kraay, J. H., Proceedings of the third international orkshop on traffic conflicts techniques. *SWOV, R-82-27*, 1982.

[6] Older, S. J. and J. Shippey, Proceedings of the second international traffic conflicts technique workshop. *Transport and Road Research laboratory*, 1980.

[7] Amundsen, F. H. and C. Hyden, Proceedings from the first workshop on traffic conflicts, 1977.

[8] Laureshyn, A., C. Johnsson, T. D. Ceunynck, A. Svensson, M. de Goede, N. Saunier, P. Wlodarek, A. R. A. van der Horst, and S. Daniels, Review of current study methods for VRU safety. Vol. Appendix 6 – Scoping review: surrogate measures of safety in site-based road traffic observations. InDeV, Horizon 2020 project. Deliverable 2.1 – part 4, 2016.

[9] Laureshyn, A. and M. Nilsson, How accurately can we measure from video? Practical considerations and enhancement of the camera calibration procedure. *Transportation Research Record*, 2018.

[10] Knake-Langhorst, S., K. Gimm, T. Frankiewicz, and F. Köster, Test site AIM – toolbox and enabler for applied research and development in traffic and mobility. Vol. Transportation Research Procedia 14, pp. 2197–2206, 2016.

[11] Ismail, K., T. Sayed, and N. Saunier, Automated safety analysis using video sensors: technology and case studies. Vol. Canadian Multidisciplinary Road Safety Conference, 2010.

[12] Laureshyn, A., Application of automated video analysis to road user behaviour. *Department of Technology and Society, Faculty of Engineering, LTH. Bulletin 253*, Vol. Doctoral thesis. Lund University, Transport and Roads, 2010.

[13] Alldieck, T., C. H. Bahnsen, and T. B. Moeslund, Context-Aware Fusion of RGB and Thermal Imagery for Traffic Monitoring. *Sensors*, Vol. 16, No. 11, 2016.

[14] Moeslund, T., *Introduction to video and image processing: Building real systems and applications*. Springer, 2012.

[15] Ismail, K., T. Sayed, N. Saunier, and C. Lim, Automated analysis of pedestrian-vehicle conflicts using video data. *Transportation Research Record: Journal of the Transportation Research Board*, , No. 2140, 2009, pp. 44–54.

[16] Buch, N., S. A. Velastin, and J. Orwell, A Review of Computer Vision Techniques for the Analysis of Urban Traffic. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 3, 2011, pp. 920–939.

[17] Cheung, S.-C. S. and C. Kamath, Robust background subtraction with foreground validation for urban traffic video. *EURASIP Journal on Advances in Signal Processing*, Vol. 2005, No. 14, 2005, p. 726261.

[18] Buch, N., J. Orwell, and S. A. Velastin, Urban road user detection and classification using 3D wire frame models. *IET Computer Vision*, Vol. 4, No. 2, 2010, pp. 105–116.

[19] Leibe, B., K. Schindler, N. Cornelis, and L. Van Gool, Coupled object detection and tracking from static cameras and moving vehicles. *IEEE PAMI*, Vol. 30, No. 10, 2008, pp. 1683–1698.

[20] Schmidhuber, J., Deep learning in neural networks: An overview. *Neural networks*, Vol. 61, 2015, pp. 85–117.

[21] LeCun, Y., Y. Bengio, and G. Hinton, Deep learning. *nature*, Vol. 521, No. 7553, 2015, p. 436.

[22] Lowe, D. G., Object recognition from local scale-invariant features. In *Computer vision*, IEEE, 1999, Vol. 2, pp. 1150–1157.

[23] Dalal, N. and B. Triggs, Histograms of oriented gradients for human detection. In *CVPR*, IEEE, 2005, Vol. 1, pp. 886–893.

[24] Jones, M. J. and D. Snow, Pedestrian detection using boosted features over many frames. In *ICPR*, IEEE, 2008, pp. 1–4.

[25] Viola, P., M. J. Jones, and D. Snow, Detecting pedestrians using patterns of motion and appearance. IEEE International Conference on Computer Vision, 2003, p. 734.

[26] Liu, W., M. Zhang, Z. Luo, and Y. Cai, An Ensemble Deep Learning Method for Vehicle Type Classification on Visual Traffic Surveillance Sensors. *IEEE*, Vol. 5, 2017, pp. 24417–24425.

[27] Oñoro-Rubio, D. and R. J. López-Sastre, Towards Perspective-Free Object Counting with Deep Learning. In *ECCV*, Springer, Cham, 2016, pp. 615–629.

[28] Ahrnbom, M., M. B. Jensen, K. Åström, M. Nilsson, H. Ardö, and T. Moeslund, Improving a Real-Time Object Detector with Compact Temporal Information. In *IEEE International Conference on Computer Vision Workshops*, 2017, pp. 190–197.

[29] Krizhevsky, A., I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[30] Saunier, N. and T. Sayed, A feature-based tracking algorithm for vehicles in intersections. In *Computer and Robot Vision*, IEEE, 2006, pp. 59–59.

[31] Tomasi, C. and T. Kanade, *Detection and Tracking of Point Features*. International Journal of Computer Vision, 1991.

[32] Hanif, A., A. B. Mansoor, and A. S. Imran, Performance Analysis of Vehicle Detection Techniques: A Concise Survey. In *Trends and Advances in Information Systems and Technologies*, Springer, Cham, 2018, pp. 491–500.

[33] Bourgeois, F. and J.-C. Lassalle, An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices. Vol. 14, 1971, pp. 802–804.

[34] Madsen, T. K. O., P. M. Christensen, C. Bahnsen, M. B. Jensen, T. B. Moeslund, and H. S. Lahrmann, RUBA-Videoanalyseprogram til trafikanalyser. *Trafik and Veje*, , No. 3, 2016, pp. 14–17.

[35] Madsen, T. K. O., C. H. Bahnsen, M. B. Jensen, H. S. Lahrmann, and T. B. Moeslund, Watchdog System, 2016.

[36] Gade, R. and T. B. Moeslund, Thermal cameras and applications: a survey. *Machine Vision and Applications*, Vol. 25, No. 1, 2014, pp. 245–262.

[37] Songchitruksa, P. and A. P. Tarko, The extreme value theory approach to safety estimation. *Accident Analysis & Prevention*, Vol. 38, No. 4, 2006, pp. 811–822.

[38] Tarko, A. P., Surrogate measures of safety. In *Safe Mobility: Challenges, Methodology and Solutions*, Emerald Publishing Limited, 2018, pp. 383–405.

[39] Elvik, R., Some implications of an event-based definition of exposure to the risk of road accident. *Accident Analysis & Prevention*, Vol. 76, 2015, pp. 15 – 24.

[40] Y. Tsai, R., An efficient and accurate camera calibration technique for 3D machine vision, 1986.

[41] Papadopoulos, D. P., J. R. R. Uijlings, F. Keller, and V. Ferrari, Extreme clicking for efficient object annotation. *CoRR*, Vol. abs/1708.02750, 2017.

[42] Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, SSD: Single Shot MultiBox Detector. *CoRR*, Vol. abs/1512.02325, 2015.

[43] Lin, T., M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, Microsoft COCO: Common Objects in Context. *CoRR*, Vol. abs/1405.0312, 2014.

[44] Bouguet, J.-Y., Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm. Vol. 1, 2000.