A Semi-Automated Software Pipeline for Assembling Collision and Near-Collision Dashcam Footage Datasets

Ben Upenieks, Chaitanya Varier,
Curtis Duy Kha Phan, Jack David Roberts Williamson,
Nicholas Geofroy, Chengzhong Meng, Vincent-Olivier Roch

Faculties of Mathematics and Engineering, University of Waterloo
ben.upenieks@uwaterloo.ca, cvarier@uwaterloo.ca,
cdkphan@uwaterloo.ca, jdrwilli@uwaterloo.ca,
nicholas.geofroy@uwaterloo.ca, c24meng@uwaterloo.ca, vroch@uwaterloo.ca

***Abstract*— Over the last decade, emphasis has been made on the need for autonomous vehicles to reduce road collisions and reckless driving. However, research into collision prediction has been slowed by the lack of egocentric collision-oriented datasets. Currently, this type of data is often created through computer generated scenarios [1] due to the dangerous and costly nature of road collisions and data collection equipment. We provide a semi-automated software pipeline to assemble datasets of real vehicle collision events that have been published online to various video repositories. We specifically focus on collecting footage of collisions recorded from front-mounted dashcam video cameras and demonstrate the importance of learning from complex road scenes and conditions globally rather than in one region. Further, we present a small prototype dataset that we have produced using our pipeline which includes egocentric dashcam vehicle footage from all over the world, subject to varying visibilities, varying weather conditions, and varying settings.**

I. INTRODUCTION

In the advent of connected vehicles and autonomous transportation, it has become commonplace to equip vehicles with high-end equipment to facilitate automated collision avoidance systems [2]. Using these vehicles to record collision situations has a two-fold cost: both the car and expensive data-gathering equipment are put at risk. However, we can collect collision data naturally in real driving scenarios through dashcams – cheap video cameras that can be mounted to a car's dashboard. Typically used for insurance purposes, dashcams have the added benefit of recording a wide variety of real collisions that are difficult to simulate effectively. This has led to the creation of many online communities to share videos of these rare situations on websites such as Reddit, YouTube, and Imgur. We propose that this egocentric dashcam video stream can be leveraged by computer vision systems to act as an affordable and commonplace sensor for automated collision prediction.

In this paper we present a software pipeline to facilitate the collection, processing and annotation of egocentric dashcam videos from various online video repositories to produce datasets that can be used to train deep dashcam collision prediction models. We aim to generate datasets that are not constrained to any regions or countries but rather provide location-agnostic data that is representative of general, global road and vehicle conditions. These datasets will also be collision-oriented as the majority of these videos contain collisions for people's viewing. Current autonomous vehicle data collection techniques typically require astronomically expensive hardware and equipment. Consequently, they are rarely involved in collisions and intentionally not put in high risk environments that may lead to a collision. Thus, our goal is to capture this data by taking advantage of existing videos of real-world collisions.

The pipeline uses web-scraping techniques to fetch the video data from these websites, along with a set of video metadata, describing the video's contents. We use this data, in combination with the manually annotated data in order to create a prototype collision dataset of dashcam video samples. This combination allows us to take advantage of the wide variety of videos available on the Internet while still having reliable data that has been curated manually. As part of this data, we identify *participants* and *bystanders*, both known as *agents*. Participants are objects that are involved in a collision identified in the metadata, while bystanders are uninvolved objects.

It is worthwhile to note that in collecting samples for our prototype dataset, we do not impose a minimum video resolution threshold. We mitigate this by only collecting samples which contain clearly identifiable agents. This allows us to maximize the amount of data that we can feasibly include in our dataset.

II. RELATED WORK

Chan et al. [3] approach the same problem of collision prediction via egocentric dashcam videos. They propose a novel Dynamic-Spatial-Attention Recurrent Neural Network (DSARNN) that distributes attention to localized and detected vehicles within the video and models temporal dependencies to predict the earliest frame of a potential impending collision. Along with their paper, they have released a dataset of 678 dashcam videos capturing areas in Taiwan. Each video is annotated with vehicle tracking bounding boxes, the vehicles that were involved in the collision and the frame of the collision. We hope to allow researchers to expand on this dataset with location-agnostic data and similar annotations.

III. METHODOLOGY

Our pipeline is built from several modular components called executors that each process or handle the data corresponding to a single video before piping the resulting data to the next executor. Figure 10 illustrates the organization of the pipeline in its default configuration.

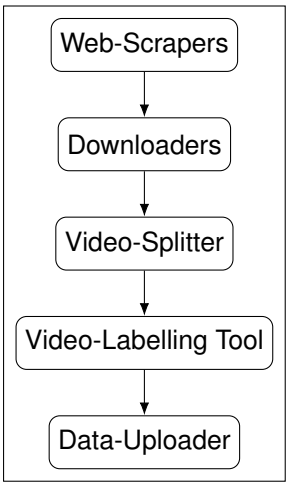

Fig. 1. Architecture of the pipeline in its default configuration. Each node in the diagram represents a stage of the pipeline, consisting of a single executor or series of executors.

A. Video Collection

The entry into our pipeline is a set of web scraper executors that each target and scrape a specific website or video repository based on a user-defined set of key-terms, and a *database source executor* that pulls unlabelled videos from the user's database. Videos that are pulled by the scrapers are immediately added to the database with relevant metadata and marked as unlabelled. Subsequently, they are loaded into memory and sequentially fed into the pipeline. If the pipeline terminates before labelling a video, it remains flagged as unlabelled in the database and the video will be pulled in a subsequent run by the *database source executor*. Crawler state is maintained via database lookup of specified key-value pairs to ensure duplicate data is not pulled. No video preprocessing such as resizing, trimming or cropping is automatically performed.

Users may define their own search terms to be used when querying the video repository. This value can be set in a YAML pipeline configuration file. By default, our scrapers query using the keywords: `Car crash dashcam`. Currently we support crawling Reddit and YouTube to locate videos that are of interest, and downloading videos hosted on YouTube and Imgur. Users that want to pull from a video repository that is not currently implemented should create a custom executor and implement the `iCrawler` interface. This new executor should be added as an entry point in a new YAML configuration and should generate and return a `MetaDataItem` with video source information to be used by a subsequent and corresponding downloader. `MetaDataItem` objects are fed into the `UniversalDownloader` which aggregates `iDownloader` implementations. The `UniversalDownloader` then distributes `MetaDataItem` objects to their corresponding downloader by matching source URLs through regular expressions. New downloaders should be registered with the `UniversalDownloader` via a YAML pipeline configuration along with a regular expression string for URL matching.

B. Anonymization

An important facet of automobile data aggregation is in eliminating biases and protecting the identities of depicted individuals. Consequently, our pipeline contains a post-processing stage in which we anonymize personally identifiable information. Specifically, we provide the executor `FaceBlurrer`, which blurs faces. This executor first uses a MobileNet Single Shot MultiBox Detector [4] trained on the WIDERFACE dataset [5] to obtain the bounding boxes of all visibly identifiable individuals' faces at each frame of a given video. The executor then applies a blurring mask over each bounding box and writes the modified frame back to the video stream. We have chosen to use a MobileNet SSD for this task because it achieves a good balance between accuracy and performance, while requiring relatively little memory [4].

This software module is encapsulated as a library and can be reused in other projects by importing our `anonymization` package. The accuracy, amount and type of blurring applied can be tweaked using the exposed API.

C. Annotation

In addition to collection and preprocessing, a core component of our pipeline is a GUI tool that facilitates the manual annotation of the dashcam videos, and quick manual filtering of non-dashcam and irrelevant videos. Following Chan et al.[3], if a collision occurs, we provide tools to annotate the frame where the collision occurs and bounding boxes to localize, identify and track agents in the video segment. Object tracking bounding boxes can be linearly interpolated from manual start and end points to improve ease of labelling. We also save the video resolution at which the video was annotated to ensure bounding box annotations can always be scaled to fit variable download resolutions. For long-form videos that may contain multiple collision events or irrelevant video segments, such as compilation videos or videos with substantial editing and transitions, we provide a video clipping feature to demarcate the collision events from which we extract and generate distinct data points. Each video segment will eventually become a metadata item. Figure 2 shows an example of the GUI tool in usage.

D. Metadata

After all of the previous steps have been completed, the pipeline is able to produce a complete metadata object. There are many fields in the metadata object (as can be seen in Figure 3):

- "description": Description of the video as scraped from the video hosting website
- "state": String that describes the current labelling state of the video. One of *processed*, *in-progress*, *NULL*
- "download_src": The site that the video was sourced from (e.g. YouTube)
- "tags": Object that contains custom user-defined tags on videos and additional data that is site- specific
- "title": The title of the video from the video hosting website
- "url": URL that the video was fetched from
- "is_split_url": Boolean that is true if the video has been segmented into multiple MetaDataItem objects during labelling, false otherwise
- "enum_tags": List of tags identifying the content of the video. New tags can be added while labelling and can identify anything important in the video.
- "start_i": The start frame of the video segment that the bounding boxes are created for (if is_split_url is true)
- "end_i": The end frame of the video segment that the bounding boxes are created for (if is_split_url is true)
- "bb_fields": All bounding-box specific data
  - "collision_locations": List of frame numbers indicating the collision start frames
  - "resolution": List with 2 elements (width, height) representing the resolution in pixels of the video the bounding boxes were created for
  - "objects": List of all of the relevant metadata pertaining to each agent in the video (either participant or bystander). Each item in this list includes the following:
    * "id": Unique numerical identifier that represents this object
    * "bboxes": List of bounding boxes for this specific object. Each bounding box is represented as a 1×5 array with elements in the following order: frame number, x1, y1, x2, y2
    * "class": The class of the object (eg. "car", "pedestrian")
    * "has_collision": Boolean that is true if the object is in a collision, false otherwise

IV. EXPERIMENTS

A. Prototype Dataset

In order to evaluate the importance of acquiring data from multiple geographical locations and under varying driving conditions around the world, we have built a prototype dataset of 223 video clips using our pipeline on which we will test the existing collision anticipation model, DSARNN. This dataset is comprised of 97 positive and 126 negative videos. Most of these videos represent road scenes from North America, Europe and Russia which have noticeably different characteristics than those in Taiwan. To prepare our data we follow the steps outlined by Chan et. al [3].

Fig. 2. GUI tool to facilitate annotation

```
{
  "description":"",
  "download_src":"YouTube",
  "tags": {
    "reddit_post_info":{
      "id":"<post_id>"
      "title":"title"
    }
  },
  "title":"Title of the Video",
  "url":"http://www.myvideo.com",
  "is_split_url": false,
  "enum_tags": ["NoCollision"],
  "start_i": 0,
  "end_i": 60,
  "bb_fields": {
    "collision_locations": [],
    "objects": [
      {
        "bboxes": [
          [0, 1096, 377, 1279, 634]
        ],
        "class": "car",
        "has_collision": false,
        "id": "1"
      }
    ],
    "resolution": [1280, 720]
  }
}
```

Fig. 3. Example metadata produced by the pipeline

Namely, we trim videos such that they are 5 seconds long, sampled at 20 FPS, and the collision occurs at frame 90. This trimming and sampling functionality is provided within the pipeline configuration `smallcorgi_exporter.yml`. In our experiments, we train the DSARNN model on features produced by running a pre-trained VGG on each full frame and the cropped objects within the frame [6]. This is denoted VGG+RNN+F+D-con. in the original publication. We note that this data includes manually annotated bounding boxes, so there will be no error imposed by using automated vehicle detection and localization models.

We have aggregated some statistics about our database and dataset to help clients understand the pertinent distributions in the data. While these distributions may be representative of online collision footage, they are not necessarily representative of real world collisions. The total number of videos represented by the following tables and histograms is a superset of the number of videos in the final prototype dataset as it includes videos that were ultimately unsuitable for our collision prediction experiments. Thus, we denote this superset as the *pre-processed prototype dataset*.

TABLE I

DISTRIBUTION OF VIDEOS IN DATABASE BY DOWNLOAD SOURCE

Download source	Percentage of videos
youtube.com	**98.40%**
imgur.com	0.85%
gfycat.com	0.47%
streamable.com	0.28%

TABLE II

DISTRIBUTION OF VIDEOS IN PRE-PROCESSED PROTOTYPE DATASET BY GEOGRAPHICAL REGION

Region	Percentage of videos
North America	48.13%
Europe	24.30%
Russia	19.16%
Asia	4.67%
Australia	3.74%

From Table I, it is clear that the vast majority of videos that have been processed through our pipeline and cataloged in our database were obtained from YouTube. It can also be seen that the YouTube and Reddit downloaders are capable of scraping various other online repositories such as Imgur, Gfycat and Streamable.

From Table II, we can see that our pre-processed prototype dataset consists of dashcam footage from a reasonably well-balanced distribution of geographical regions. The dataset mostly consists of samples from North America, Europe and Russia, which complements the dataset presented by Chan. et. al. [3] well. This also implies a diverse representation in driving conditions due to the various differences in climate and infrastructure amongst these regions.

From Figure 4, we see that there are several types of objects which have been identified as agents in our pre-processed prototype dataset. Among these, the most common are motorized vehicles and pedestrians as one would intuitively expect. A number of stationary, inanimate objects such as industrial machinery were worth noting as they were involved in collisions. These have been aggregated into the "other" category.

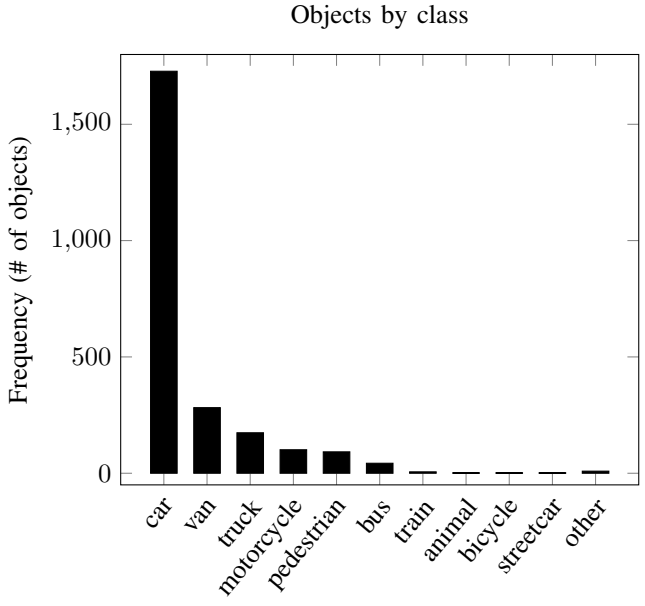

Fig. 4. Distribution of videos in pre-processed prototype dataset by object class

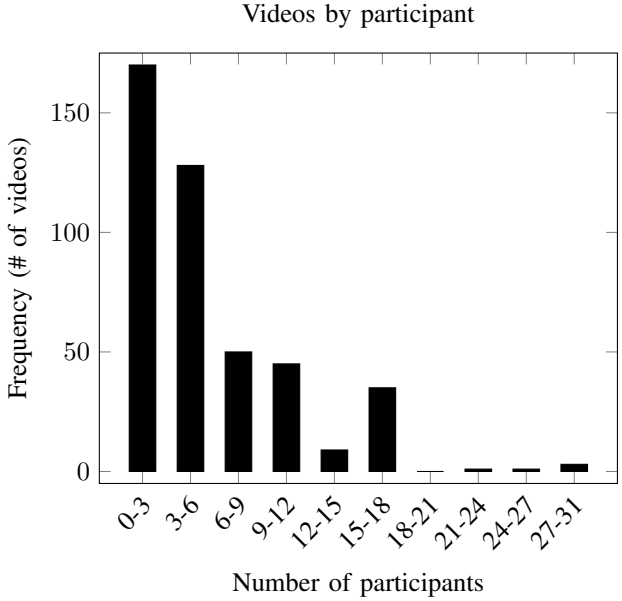

Fig. 5. Distribution of videos in pre-processed prototype dataset by number of participants

From Figure 5, it can be seen that there is a wide range in the distribution of the number of participants in each sample of our dataset. We also observe that it is more common for no collisions to occur than any particular non-zero number of participants to be present in a sample.

From Figure 6, we observe that it is most common for samples to contain either exactly 2 agents or no agents at all. This is implied by the fact that videos with only 1 agent cannot contain a collision. These are less likely to be interesting and in turn, scraped by our pipeline.

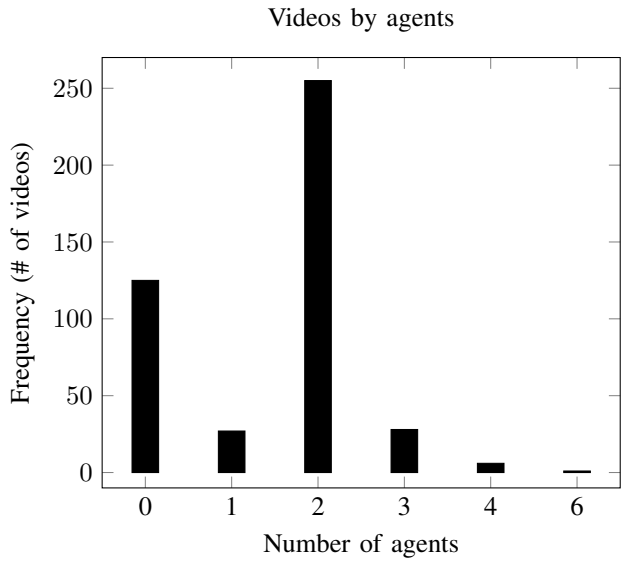

Fig. 6. Distribution of videos in pre-processed prototype dataset by number of agents

From Figure 7, we can clearly see that there is a roughly standard normal distribution in the number of videos by duration, with a mean, median and mode at around 352 frames. This suggests that the pipeline favors videos which tend to be shorter on average. At an average video framerate of 30 FPS, this equates to a mean sample duration of just over 11 seconds. In section IV-A, it was mentioned that the samples were cropped to 5 seconds in the final prototype dataset. Therefore, this is a favorable characteristic as it helps reduce the amount of necessary processing.

Figures 8 and 9 outline key characteristics about our database at large. Figure 8 shows that from the video repositories we targeted, we were able to obtain a relatively well-balanced distribution of collision types in our data. Here, the distinction between *Simple Collision* and *Multi Collision* is that the former involves exactly 2 participants and the latter involves more than 2 participants. Figure 9 shows that the majority of the videos the pipeline scraped were indeed dashcam videos. This suggests that the pipeline is quite effective in identifying suitable candidates for a dataset consisting solely of dashcam collision footage.

B. Collision Prediction Model Training and Analysis

In our first experiment, we merge our prototype dataset with the dataset provided by Chan et. al. We augment their training set with 77 positives and 106 negatives, and their testing set with 20 positives and 20 negatives. We use the same training procedure as Chan et. al. Namely, we train for 40 epochs with a learning rate of 10^{-4} and a batch size of 10. Although our data contribution here is relatively

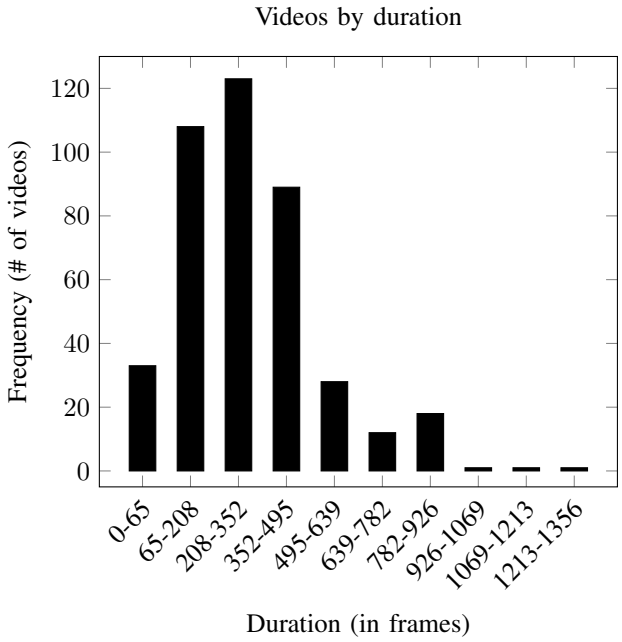

Fig. 7. Distribution of videos in pre-processed prototype dataset by duration

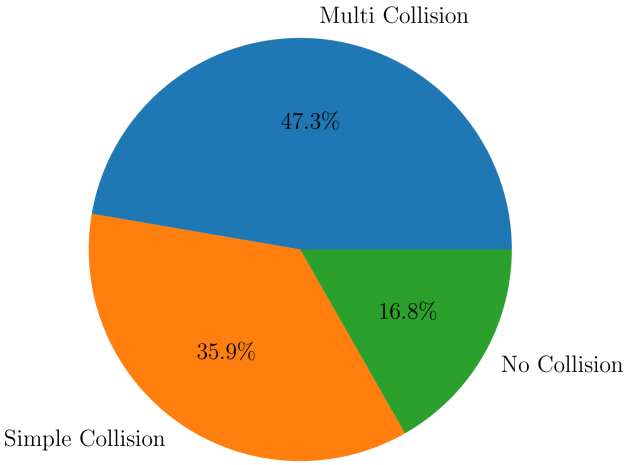

Fig. 8. Distribution of videos in database by collision type

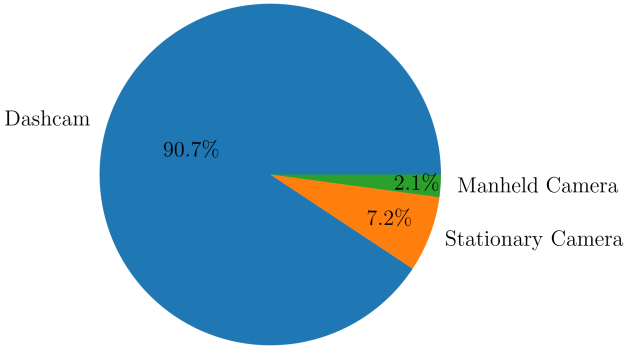

Fig. 9. Distribution of videos in database by camera type

small, we observe a mean time-to-collision of **2.198s** at **80%** recall which is very similar to the existing results from

Chan et. al. However, we observe an average precision of 66.22% which is appreciably lower than the 73.53% average precision reported by Chan et. al. [3]

In our second experiment, we use the provided pre-trained DSARNN model and test the performance on our entire dataset of 223 videos. We see very poor prediction performance at an average precision of 34.29%, and in Figure 10 we observe poor precision at all recall values.

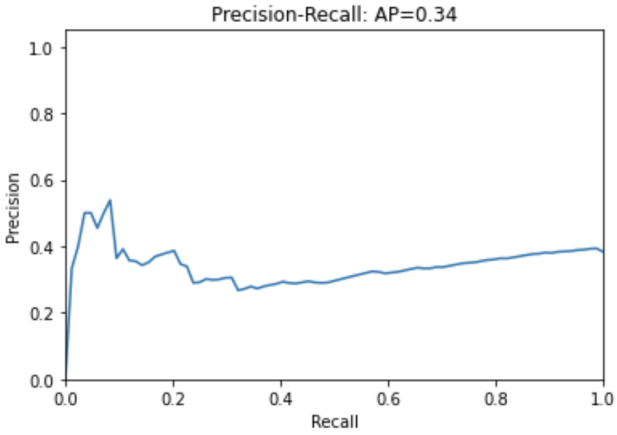

Fig. 10. Pre-trained DSARNN Precision-Recall curve on prototype dataset

These experiments support the notion that in order for a collision prediction model to be generally effective in practice, it must be trained on collisions and complex roads scenes from varying regions and road conditions around the world.

V. FUTURE WORK

There are many ways in which we could improve our system, namely the following:

1) **Labelling precision improvements:** The current video labelling process in the pipeline involves manual bounding box drawing with linear interpolation between input frames. This is prone to inserting mistaken bounding boxes that do not exactly match up with the location of an object on certain frames. Automated labelling and custom interpolation would speed up the process of labelling and minimize the cases of misplaced bounding boxes. Labelling would be much faster if we had a semi-automated process of detecting objects and tracking them over frames. There are models for object detection that behave reasonably well for our purposes, but they would still need to be augmented by a review process. We could easily solve the problem of object tracking by implementing simple heuristic algorithms. Being able to modify the method of interpolation between frames beyond the default linear interpolation would allow for more bounding box precision between large input frame windows. Moreover, if we store the interpolation type in the metadata, we would likely gain a significant improvement in performance and memory usage by not storing all the interpolated frames rather than just the ones the user has entered. While linear interpolation works well for entering one frame for every 10 frames in a video clip, supporting additional types of interpolation would facilitate quicker data entry and enable the possibility of supporting affine transformations on more complex bounds.
2) **Complete Location-Varied Dataset:** Our next step is to expand our existing dataset of automobile collisions to include samples from additional regions across the globe. We would then use this to form a training and testing set for our next experiment. Finally, we would compare the results of this experiment with location-varied data to the results from the dataset collected in Taiwan by Chan et. al.

REFERENCES

[1] H. Kim, K. Lee, G. Hwang, and C. Suh, "Crash to not crash: Learn to identify dangerous vehicles using a simulator," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 978–985, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/3887

[2] P. Perumal, "Lidar based intelligent obstacle avoidance system for autonomous ground vehicles," 2020.

[3] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun, "Anticipating accidents in dashcam videos," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 136–153.

[4] Y. H. Hu, K. Wa, J. Bruzaud, X. Y. Lim, and Z. Medley, "Tensorflow face detector," Mar. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4642275

[5] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[7] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[8] M. Pitropov, D. Garcia, J. Rebello, M. Smart, C. Wang, K. Czarnecki, and S. Waslander, "Canadian adverse driving conditions dataset," 2020.

[9] J.-A. Pattinson, H. Chen, and S. Basu, "Legal issues in automated vehicles: critically considering the potential role of consent and interactive digital interfaces," *Humanities and Social Sciences Communications*, vol. 7, no. 1, p. 153, Nov 2020. [Online]. Available: https://doi.org/10.1057/s41599-020-00644-2