Capstone Project Report

Car Crash Detection using 3D Convolutional Neural Networks

LAU

الجامعة اللبنانيّة الأميركيّة

Lebanese American University

Yasmine Mnafki

Advisor: Dr. Danielle Azar

Department of Computer Science and Mathematics
Lebanese American University

[Spring 2023]

Abstract

In this study, we aimed to develop an effective car crash detection method using a 3D Convolutional Neural Network (CNN) architecture. Car crash detection plays a crucial role in improving road safety and facilitating timely emergency response. Our approach involved training a 3D CNN to analyze spatiotemporal features in video data, thereby detecting car crashes in real-time. The model achieved an accuracy of 90.62% and an AUC score of 93.07% in detecting car crashes. These results demonstrate the potential of our proposed method for enhancing surveillance systems and aiding in accident prevention. Despite some limitations, such as dataset variability and the omission of trajectory analysis due to time constraints, our study contributes valuable insights to the field of car crash detection and offers a foundation for future research.

Table of contents

Chapter 1

Introduction

Anomaly detection is the process of detecting patterns or occurrences in the data that differ from what would be considered typical or anticipated behavior. A problem, threat, or opportunity may be indicated by rare or unexpected occurrences that can be found through this technique. Finding anomalies is crucial for solving a variety of problems in a variety of fields. For instance, in the financial industry, identifying fraudulent transactions can stop financial losses and protect the organization's reputation. Early disease detection in the medical field can result in better treatment outcomes and patient health. In cybersecurity, spotting network intrusions can stop data breaches and safeguard private data. Recognizing unusual behaviors in video surveillance can improve public safety. Overall, anomaly detection is a crucial technique for identifying unusual and potentially harmful events in various scenarios. One notable application of this technique is the detection of car accidents in surveillance videos. Road accidents affect 20–50 million people annually, killing 1.3 million people (WHO 2022). With prompt intervention from authorities and emergencies, traffic accidents, injuries, and fatalities can be significantly reduced. Given the abundance of surveillance cameras in public places, there is a wealth of data that can be used to create smart computer systems to decrease the amount of time and labor required to monitor these cameras. These systems can assist in the quick real-time detection of traffic accidents, alert the appropriate authorities for quick intervention, and lower the risk of fatality. The existing systems can be categorized into two types in terms of the type of cameras used- CCTV cameras or dashboard cameras. In this paper, we propose a car crash detection method using a novel 3D convolutional neural network (CNN) architecture. The proposed approach effectively detects car crashes by feeding videos into a 3D CNN, which analyzes the spatiotemporal features of the generated video frames and detects the occurrence of an accident.

Chapter 2

Literature Review

2.1 Anomaly Detection Methods

In recent years, anomaly detection research has gained a lot of attention. Anomaly detection methods involve identifying events that are abnormal or unexpected, such as car crashes. Researchers have approached the identification of anomalies in videos mainly as a likelihood problem. However, an approach that represents a benchmark in video anomaly detection is using weakly-supervised learning techniques, such as multiple instance learning (MIL). This method involves dividing surveillance videos into a fixed number of segments during training and treating them as instances in a bag. The proposed deep MIL ranking loss is then used to train an anomaly detection model using positive (anomalous) and negative (normal) bags. In this approach, anomaly detection is formulated as a regression problem, and a binary SVM classifier is used to distinguish between normal and abnormal instances (Sultani et al. 2018). This method can effectively detect car crashes in real-time, but it may also generate false alarms due to other variables such as crowded scenes or darkness. Although this method was trained on a large amount of data, only 150 videos were allocated for road accidents. A dataset that focuses on car crashes may increase the accuracy of this approach.

2.2 Multimodal Data Methods

Multimodal data methods involve using data from multiple sources, such as video, audio, and sensor data, to detect car crashes.

M et al. (2021) propose a system to detect and predict road accidents by analyzing traffic behavior and identifying abnormal traffic flow using GPS data. The study uses a variety of methods to accomplish this, including hardware devices like GPS, vibrating sensors, and ultrasonic sensors, as well as three different algorithms like SVM, Random Forest, and CNN. To track and identify accidents, the system divides data into groups using grouping techniques. A brief description of the accident is also provided to authorities by the system via the GPS tracking system.

The system has the advantage of sending an automatic message to the designated email id. An alert system can be further implemented to send a message to the police and emergency services, potentially saving lives. The system deals with the detection and forecasting of traffic accidents, which can significantly increase traffic safety. However, the system's reliance on hardware devices, which may not always be available or reliable, is a disadvantage. Nonetheless, it can be used to provide additional information such as the severity of the crash using different sensors.

In the study by Choi et al. (2021), an ensemble deep learning model based on both video and audio data obtained from dashboard cameras is used to detect car crashes. A weighted average ensemble technique is later used to combine both data types. A CNN-and-GRU-based classifier is proposed for car crash detection using video data. In contrast, a GRU-based classifier is developed for car crash detection using audio features, and a CNN-based classifier is used for spectrogram images.

The proposed car crash detection system has a high classification performance, with audio data having a higher accuracy than video data. According to the study, the complexity of video representation in comparison to audio makes it difficult to achieve high performance in car crash detection. The ensemble model can classify crashes, near crashes, and non-crashes with high accuracy. However, future work is needed to improve the performance of the classification of crashes and near-crashes.

2.3 Video-Only Methods

Both studies by Ijjina et al. (2019) and Machaca Arceda and Laura Riveros (2018) used a common approach of breaking down the process into three stages: car detection, vehicle tracking, and accident detection. The study by Ijjina et al. (2019) proposed a supervised deep-learning framework to detect roadside objects for accident detection. The proposed algorithm consists of three tasks: vehicle detection, vehicle tracking, feature extraction, and accident detection using three new parameters (Acceleration Anomaly, Trajectory Anomaly, and Change in Angle Anomaly). The Mask R-CNN object detection framework and Centroid Tracking algorithm are used for vehicle detection and tracking, respectively. Lastly, the combined anomalies in the different parameters are used to determine if the accident occurred taking into account the weights of each threshold. These methods have shown promising results, with high accuracy rates. However, the study highlighted the proposed algorithm's ineffectiveness in high-density traffic due to inaccuracies in vehicle detection and tracking. Large obstacles in the camera's field of view may also interfere with vehicle tracking and collision detection.

The approach proposed by Machaca Arceda and Laura Riveros (2018) uses a convolutional neural network (CNN) with the You Only Look Once (YOLO) network in the first stage. The second stage employs a tracker to focus on each car, and the final stage employs the Violent Flow (ViF) descriptor in conjunction with a Support Vector Machine (SVM) to detect car accidents. Despite the dataset containing videos in bad conditions like wind or poor resolution, the system showed good results with an 80% probability of detection. However, the dataset can be further improved as it is relatively small. The study also suggests the use of vector direction in Vif as a way to improve this method's accuracy in the future.

2.4 Trajectory Feature Extraction

In previous work done in action recognition and anomaly detection, including car crash detection, trajectory feature extraction has been used to identify and analyze motion patterns in order to extract the relevant information or features. Researchers have employed different approaches to trajectory analysis, including supervised and unsupervised methods.

Unsupervised

Due to the difficulty of collecting and labeling data, it was easier to adopt an unsupervised method in some techniques used. The approach by Sekh et al. (2020) used unsupervised clustering and multi-criteria ranking. The proposed method, called t-Cluster, prepares indexes of object trajectories using high-level features such as origin, destination, path, and deviation. The clusters are then fused using multi-criteria decision making and trajectories are ranked based on abnormality scores. The method can effectively identify abnormal patterns in trajectory data. The t-Cluster technique utilizes a set of trajectories extracted using Multi-Object Tracking and returns a set of clusters, assigning each trajectory to three clusters. The method is based on a clustering and ranking approach using entry/exit regions and entry-to-exit paths. It generates trajectory abnormality scores for each moving object with respect to various factors including path deviation and local ranks. The inclusion of path deviation is found to split the clusters and produce larger movement patterns. Trajectory-based clustering is a widely utilized method for discovering traffic data in video. It is also used in work by Athanesious et al. (2019). They propose a method for detecting abnormal trajectories in traffic scenes, such as road intersections and highways, intending to analyze vehicle behavior. The approach involves two stages: in the first stage, the Gaussian process dynamical model with spectral clustering is estimated to group trajectories. In the second stage, Bayes decision theory is applied to detect abnormal patterns from the results of stage 1. The authors found that the proposed trajectory clustering method outperformed existing models in terms of accuracy. However, due to pre-defined cluster size and distance measures between trajectories, the technique faces difficulties in trajectory-based clustering for complex scenarios such as junctions. The use of GPU-based implementation has drastically reduced the execution time in detecting anomalies. The authors suggest that the method can be extended with deep learning methodologies. The proposed approach is effective for arbitrary shape data and does not rely on assumptions about the statistics of clusters. The results show that the method achieves 12% better accuracy in detecting abnormalities compared to state-of-the-art techniques.

Supervised

The study by Shi et al. (2015) aims to improve human action recognition by developing a more effective way to encode trajectories and fully utilize them, motivated by the success of deep neural networks. The proposed system consists of three modules: dense trajectory extraction, deep trajectory descriptor (DTD) generation, and classification using a linear SVM. The study employs HOG, HOF, and MBH descriptors and encodes them with the Fisher vector. To improve dense trajectories, the authors use ViBe, a background subtraction technique that incorporates several innovative mechanisms (Barnich and Van Droogenbroeck 2011). The results show that DTD statistically outperforms several state-of-the-art approaches, with an average accuracy of 95.6% on KTH and 92.14% on UCF50 datasets. However, the study notes that the background-subtraction-based method may remove trajectories of slight movements or tiny objects, leading to the partial loss of statistical information and potentially leading to worse results for IDT. Nevertheless, the proposed method significantly reduces disk usage while maintaining the performance of dense trajectories.

2.5 Spatiotemporal Feature Extraction With Neural Networks

Neural networks have been widely used for spatiotemporal feature extraction in various applications, including car crash detection and anomaly detection in videos. In addition to trajectory feature extraction, spatiotemporal feature extraction using 3D-CNNs has gained attention in recent years. 3D-CNNs are deep learning algorithms that can capture both spatial and temporal information from videos, making them suitable for analyzing dynamic events. Previous work has demonstrated the effectiveness of spatiotemporal feature extraction using 3D-CNNs for car crash detection. Tran et al. (2015) propose a spatiotemporal feature learning method using deep 3-dimensional convolutional networks. Their proposed 3D ConvNet consists of 8 convolution layers, 5 pooling layers, two fully connected layers, and a softmax output layer. This approach takes full video frames as inputs without relying on preprocessing, which makes it easy to scale for large datasets. The authors extract C3D features and input them into a multi-class linear SVM for classification. Experimental results show that 3D convolutional deep networks are effective feature-learning machines that can model appearance and motion simultaneously. The authors find that a $3 \times 3 \times 3$ convolution kernel for all layers works best among the limited set of explored architectures. Furthermore, their proposed features with a simple linear model outperform or approach the current best methods on 4 different tasks and 6 different benchmarks. However, it is noted that no GPU implementation of this method has been found, and it is not easy to implement a parallel version of this algorithm on GPU. The authors conclude that 3D CNNs are conceptually straightforward and easy to train and use.

Chapter 3

Methodology

3.1 Dataset

A large-scale dataset of traffic videos was collected from various sources, including traffic cameras and dashcams. We used the Road Accident video included in the Anomaly Detection dataset (Sultani et al. 2018). A part of the anomalous video was also taken from the CADP dataset (Shah et al. 2018). The normal videos are a mix of highway traffic dataset (Shah 2021) and anomaly dataset normal videos (Sultani et al., 2018) that involve traffic and cars.

3.2 Alternative Method - Trajectory Feature Analysis

In the initial stages of our research, we proposed a novel car crash detection method that combined trajectory feature clustering with a 3D convolutional neural network (CNN) architecture. The approach aimed to effectively detect car crashes by first clustering the trajectory features of vehicles in the vicinity of the target vehicle, and then analyzing the spatiotemporal features of the clustered trajectory data using a 3D CNN. The trajectory features were to be extracted from video frames using the YOLOv3 object detection model trained on the COCO dataset, which provided bounding boxes of cars in the video frames. These bounding boxes were to be used to extract trajectory features from the video frames, which were then preprocessed to calculate velocity and acceleration and concatenated with the original data before normalization.

During the development process, we identified several disadvantages and challenges associated with this approach. The trajectory features may be affected by noise and occlusion, particularly in scenes with a high density of vehicles or varying conditions such as weather. These factors could introduce errors in the extracted features, leading to decreased performance in car crash detection. Additionally, he approach relies heavily on the performance of the object detection model. Any errors or inaccuracies in the object detection could negatively impact the trajectory feature extraction and eventually, the crash detection performance.

Extracting and preprocessing the trajectory features from the video frames can be computationally expensive and complex. This could lead to longer processing times and increased resource requirements, making it difficult to implement. The approach also requires extensive feature engineering and preprocessing, such as calculating velocity and acceleration and normalizing the data. This process also can be time-consuming and challenging, especially considering the high-dimensional nature of the data.

Considering these challenges and the tight project timeline, we decided to change our approach and adopt a more feasible method. The current method, which employs a 3D CNN architecture, offers a more straightforward solution and has shown promising results in detecting car crashes.

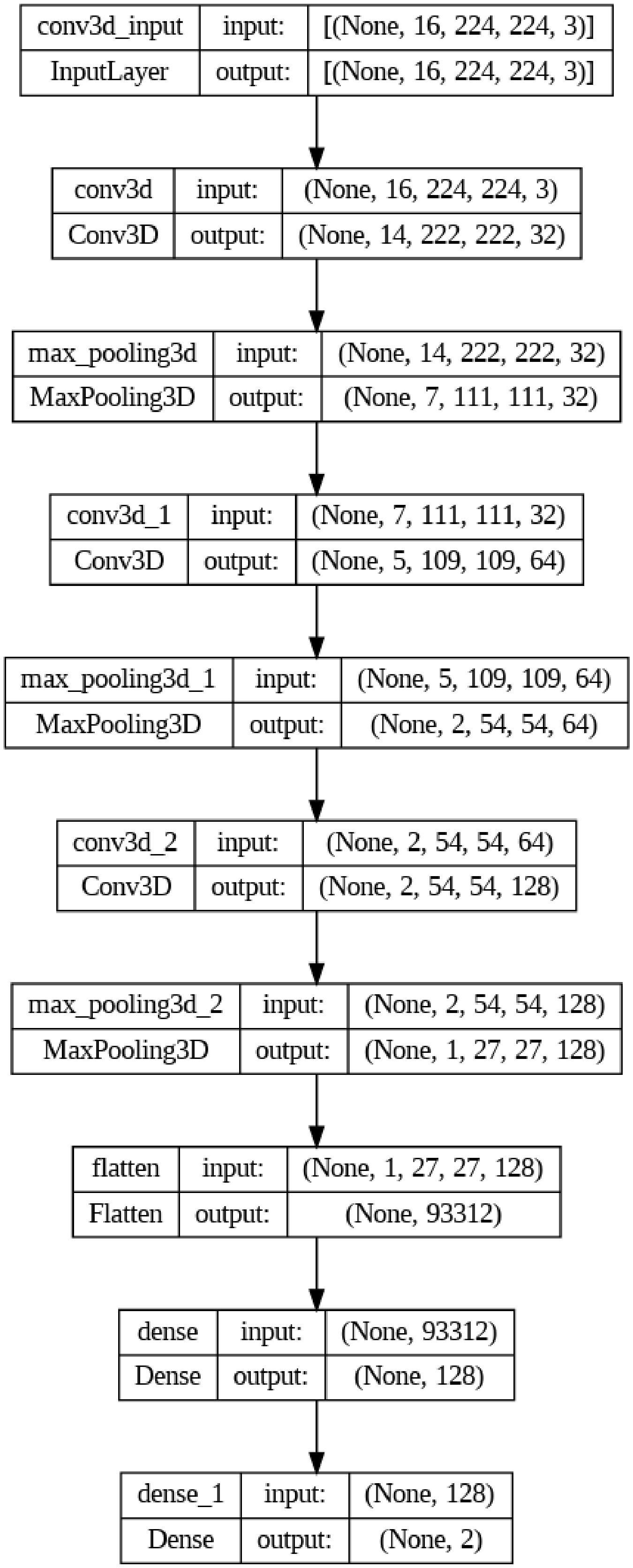
3.3 3D Convolutional Neural Network

The video features are extracted separately using a 3D CNN. The 3D CNN consists of three convolutional layers and is designed to efficiently extract features from the video frames.

The model is then trained using the extracted features and labels indicating whether a crash occurred in the video or not. The 3D CNN model uses a binary cross-entropy loss function and metrics such as accuracy, AUC, precision, and recall to evaluate the performance of the model.

3.4 The proposed architecture

The model consists of three 3D convolutional layers with ReLU activation functions, interleaved with three 3D max-pooling layers to reduce spatial and temporal dimensions of the feature maps, controlling overfitting and reducing computation. Following these layers, a flattened layer converts the 3D feature maps into a 1D vector, which is the input for two fully connected layers. The first fully connected layer has 128 neurons with a ReLU activation function to learn higher-level features. In contrast, the second has 2 neurons with a softmax activation function to produce the final output probabilities for each class.


conv3d_input | input: | [(None, 16, 224, 224, 3)]
InputLayer | output: | [(None, 16, 224, 224, 3)]
conv3d | input: | (None, 16, 224, 224, 3)
Conv3D | output: | (None, 14, 222, 222, 32)
max_pooling3d | input: | (None, 14, 222, 222, 32)
MaxPooling3D | output: | (None, 7, 111, 111, 32)
conv3d_1 | input: | (None, 7, 111, 111, 32)
Conv3D | output: | (None, 5, 109, 109, 64)
max_pooling3d_1 | input: | (None, 5, 109, 109, 64)
MaxPooling3D | output: | (None, 2, 54, 54, 64)
conv3d_2 | input: | (None, 2, 54, 54, 64)
Conv3D | output: | (None, 2, 54, 54, 128)
max_pooling3d_2 | input: | (None, 2, 54, 54, 128)
MaxPooling3D | output: | (None, 1, 27, 27, 128)
flatten | input: | (None, 1, 27, 27, 128)
Flatten | output: | (None, 93312)
dense | input: | (None, 93312)
Dense | output: | (None, 128)
dense_1 | input: | (None, 128)
Dense | output: | (None, 2)

Chapter 4

Evaluation and testing

4.1 Machine specification

The machine specifications for this project are as follows: We use a virtual GPU-enabled machine with Google Colab.

Memory: Disk size: 27.9 / 78.2 GB

System Ram: 7.1 / 12.7 GB

GPU Used: 9.85 / 15.0 GB

Data storage: The dataset was uploaded on Google Drive which is mounted in Colab to be used for training and testing. The size of the dataset is 6.16 GB.

4.2 Results

In this section, we present the results and performance of our proposed car crash detection model. As a general assessment, we use accuracy as a measure, which is the proportion of correct predictions out of the total number of predictions made. Besides accuracy, the evaluation metrics used to assess the performance are namely the F1 score and the ROC AUC score. The F1 score is the harmonic mean of precision and recall. It provides a balanced measure of the model's accuracy in identifying both car crash and non-crash instances. The AUC score measures the ability of the model to distinguish between crash and non-crash classes by evaluating the trade-off between the true positive rate (sensitivity) and the false positive rate (specificity) at various classification thresholds. We achieved an accuracy of 90.62% and an AUC of 93.07%. To verify that our data split did not introduce any bias, we conducted multiple experiments with fine-tuned hyper-parameters. The variations observed in the performance metrics across these experiments indicate that our results are not biased due to data split. In the table below (table4.1) we report the results of 5 distinct runs with the same parameters.

Table 4.1: Results of the experiments

Accuracy	**AUC**	**F1 measure**
87.5%	89.36%	89.36%
84.38%	88.67%	85.71%
84.38%	86.23%	83.33%
90.62%	93.07%	89%
87.5%	80.47%	90.62%

4.3 Comparision to existing work

We compare the performance of our model with other models in pre-existing literature. We choose to depend on the performance indicators as provided in the original papers rather than attempting to duplicate these models and run them on our own dataset.

The complexity involved with accurately reproducing these established models, data privacy considerations, computational resource limitations, and other issues all played a role in this choice.

Based on the reported results, as shown in the table 4.2 below, our model appears to perform competitively in relation to these established models. This comparative analysis strengthens the credibility of our model and underscores its potential utility in real-world applications for detecting car crashes in surveillance videos.

Nevertheless, it is crucial to keep in mind that the datasets and the specific conditions that influenced the reported metrics in these studies may not align perfectly with ours.

Table 4.2: Comparison with exising work

The work	**Accuracy**	**AUC**	**F1 measure**
(Suzuki et al. 2018)	58.54%	-	59.86
(Ki and Lee 2007)	50%	-	-
(Ijjina et al. 2019)	71%	-	-
(Singh and Mohan 2019)	77.5%	-	-
(Machaca Arceda and Laura Riveros 2018)	75%	76%	-
Proposed framework	90.62%	93.07%	89%

Chapter 5

Limitations and future work

Our study has some limitations that can be addressed in future work such as the use of a more consistent dataset, with uniform quality and resolution.This would likely improve the model's performance. The current dataset exhibits considerable variability in terms of road and location types (highways, intersections, gas station...), which may affect the model's consistency. A single source of data (particular camera location) for training the model might produce more accurate results, making it more appropriate for implementation into a surveillance system. To further improve the efficiency and resilience of the model, the dataset size may also be increased. A larger dataset would provide the model more varied examples to learn from, potentially improving generalization to new data. Future work can consider working with videos extracted from dashboard cameras considering that there is more datasets available. Finally, due to time restrictions, we were unable to include trajectory analysis in our research. Future studies could examine the possible advantages of adding trajectory feature extraction to the model, which might enhance its capacity to accurately identify car accidents.

Chapter 6

Acknowledgments

I would like to sincerely thank everyone who supported me in completing this project.

I would like to express my sincere gratitude and appreciation to my professor and capstone advisor, Dr. Azar, for her patience and invaluable feedback.

Finally, I must express my heartfelt thanks to my family and friends for their consistent support and encouragement throughout this project.

Bibliography

Athanesious, J. J., Chakkaravarthy, S. S., Vasuhi, S., and Vaidehi, V. (2019). Trajectory based abnormal event detection in video traffic surveillance using general potential data field with spectral clustering. *Multimedia Tools and Applications*, 78(14):19877–19903.

Barnich, O. and Van Droogenbroeck, M. (2011). Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724.

Choi, J. G., Kong, C. W., Kim, G., and Lim, S. (2021). Car crash detection using ensemble deep learning and multimodal data from dashboard cameras. *Expert Systems with Applications*, 183:115400.

Ijjina, E. P., Chand, D., Gupta, S., and Goutham, K. (2019). Computer vision-based accident detection in traffic surveillance. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–7. IEEE.

Ki, Y.-K. and Lee, D. H. (2007). A traffic accident recording and reporting model at intersections. 8(2):188–194.

M, B. K., Basit, A., MB, K., R, G., and SM, K. (2021). Road accident detection using machine learning. *2021 International Conference on System, Computation, Automation and Networking (ICSCAN).*

Machaca Arceda, V. and Laura Riveros, E. (2018). Fast car crash detection in video. In *2018 XLIV Latin American Computer Conference (CLEI)*, pages 1–8. IEEE.

Sekh, A. A., Dogra, D. P., Kar, S., and Roy, P. P. (2020). Video trajectory analysis using unsupervised clustering and multi-criteria ranking. *Soft Computing*, 24(21):16643–16654.

Shah, A. (2021). Highway traffic videos dataset. Kaggle.com.

Shah, A. P., Lamare, J.-B., Nguyen-Anh, T., and Hauptmann, A. (2018). Cadp: A novel dataset for cctv traffic camera based accident analysis. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*.

Shi, Y., Zeng, W., Huang, T., and Wang, Y. (2015). Learning deep trajectory descriptor for action recognition in videos using deep neural networks. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*.

Singh, D. and Mohan, C. K. (2019). Deep spatio-temporal representation for detection of road accidents using stacked autoencoder. 20(3):879–887.

Sultani, W., Chen, C., and Shah, M. (2018). Real-world anomaly detection in surveillance videos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Suzuki, T., Kataoka, H., Aoki, Y., and Satoh, Y. (2018). Anticipating traffic accidents with adaptive loss and large-scale incident db.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*.

WHO (2022). Road traffic injuries.