Comparison of Object Detection Algorithms for the Task of Detecting Possible Road Accident

Oleksandr Byzkrovnyi [1], Anastasiya Chupryna [1], Kirill Smelyakov [1], Natalia Sharonova [2], Vadim Repikhov [1]

[1] *Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine*
[2] *National Technical University "KhPI", Kyrpychova str. 2, Kharkiv, 61002, Ukraine*

Abstract

This article is an investigation of the best suited algorithm for object detection in terms of finding a potential road accident case. The determination of the best algorithm is based on comparative analysis of evaluation and testing results and metrics. Also, the max FPS of processing video during detection will be considered. There are a lot of examples of road accident situations, but in this study the turn left across one and more oncoming road lanes will be explained. There are two classes for recognition: danger and not-danger which explain case. Other accident types have not been considered due to wish for making simple models with further sophistication. The Yolov7 and Detectron2 algorithms are compared.

Keywords

Detectron2, machine learning, object detection and tracking, road accidents prevention, Yolov7

1. Introduction

Road accidents are a major problem today, as the number of cars and their speed increase, and the number of victims and injured as a result of road accidents increases.

Modern methods of preventing dangerous situations on the roads are focused on the safety of each individual car, whose systems do not consider other road users who are beyond the reach of the built-in mechanisms for ensuring the safety of the car. There is a need to create methods for notifying the driver of dangers that may be detected in traffic areas that cannot be scanned by existing ADAS systems due to the impossibility of analyzing traffic for obstacles of various kinds. Another important safety factor is the driver's appropriate response to changes in traffic and the flow of information received while driving. Thus, based on the above, the development and research of new methods for monitoring driver behavior, predicting dangerous situations while driving, and notifying the driver about them is a popular and relevant task. The goal of this research is estimation of an algorithm which has the smallest time for prediction, smallest size and good accuracy for the problem of identifying and prevention of road accidents based on the outdoor video stream.

2. Related works

Many studies and reviews are devoted to various aspects of detection, prediction and prevention of accidents using modern information technologies [1-3]. Including the most important aspects of

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine
EMAIL: oleksandr.byzkrovnyi@nure.ua (O.Byzkrovnyi); anastasiya.chupryna@nure.ua (A. Chupryna); kyrylo.smelyakov@nure.ua (K. Smelyakov); nvsharonova@ukr.net (N. Sharonova); vadym.repikhov@nure.ua (V. Repikhov)
ORCID: 0000-0001-9335-442X (O. Bizkrovnyi); 0000-0003-0394-9900 (A. Chupryna); 0000-0001-9938-5489 (K. Smelyakov); 0000-0002-8161-552X (N. Sharonova); 0000-0002-1274-4205 (V. Repikhov)

forecasting the development of situations [2] and the use of modern neural networks and deep learning models [3].

In this regard, the work [4] describes the features of building an autonomous decision support system in the sense of safety based on the analysis and evaluation of road accidents. The work [5] presents a solution with respect to the recognition of the signs of traffic accidents in cities Based on GA-XGBoost for Big Data conditions. In [6, 7], a detailed situation analysis is given and the application of machine learning models to classify the severity level of road accidents involving two-wheeled vehicles, mainly bicycles and motorcycles, is described. In [8, 9], adaptive models of image gradation correction for high and low field-of-view illumination conditions are described. The main value of the models is the ability to be bound to an arbitrary brightness scale (distribution) and file format. In [10] the authors have proposed a new structure of probabilistic convolutional neural network and fuzzy logic model, which includes a person in the loop and takes into account multiple input data streams generated by certain sensors; in fact, the model is based on the analysis of human emotions and traffic data. In [11], a unified latent space model is proposed that links extreme weather conditions with traffic safety and accidents. The model is based on the use of nested time series. In [12], a model and algorithms for crash detection using synthetically generated videos of crashes from different angles are proposed. The model is based on the use of deep convolutional neural networks for feature extraction. In [13], the authors proposed a high-precision method of segmentation and vectorization of three-dimensional road boundaries to eliminate the gap between point clouds, which are obtained by unstructured mobile laser scanning technology and vector representation of road boundaries. The article shows high efficiency of the proposed method. The paper [14] describes the key aspects of the development of super-precision neural networks, which are used for image analysis, shows their advantages and disadvantages. In [15], a model for crash detection based on self-learning consistency in driving scenarios is proposed. In fact, the model is based on finding inconsistencies in the sequence of video frames. In [16], the authors proposed a model for preventing crashes and accidents that are caused by impairment. The model makes decisions based on the detection and analysis of disease symptoms and driver behavior. In [17], a convolutional neural network model is proposed for the purpose of predicting crash severity with high prediction accuracy. The proposed model considers the mutual relations between the characteristics and the participants of the crash. In [18], a model and algorithm for analyzing the road situation and risk analysis in the context of vehicle lane change. The paper [19] presents the results of research on modern models of driving and on-board training based on feedback. The main purpose of the research is proactive monitoring of road safety and timely decision-making in the conditions of danger. In [20], the problem of real-time crash detection using speed sensors that are distributed on the freeway is solved. A new Bayesian formula for rapid change detection and an optimal decision-making strategy based on its use using dynamic programming is proposed. In [21], models and methods for end-to-end risk analysis in autonomous driving situations are proposed. An approach to behavioral anomaly detection and decision making based on the application of convolutional neural networks is described.

The development of systems and technologies for detecting, predicting, and preventing traffic accidents is now inextricably linked to the timely detection and prevention of cyber threats [22, 23], as well as to the effective expert analysis and evaluation of technical and IT solutions in the subject area [24, 25].

Also, there is a big number of datasets which contain different info about road accidents: time, weather, weekday, month, road surface condition, etc. All these factors are collected and may be used for analysis. Particularly, analysis may consist of determining factors which may have the biggest influence on probability of road accidents happening [26]. The mentioned study uses linear regression for analyses of different factors and metrics before accidents happen, to reduce their number and define which one has the biggest impact on safety in Haryana state. The dataset there covers a period of 21 years: 1996 – 2016. The methodology for the mentioned article is on the Figure 1.

The results of the research are nice. There are three models for different types of accident results (accident in general, fatal accident, injury accident) which have the following R2 metric value corresponded: 0.945, 0.854, 0.988. Which means that factors have a relationship between dependent and independent variables.

Another research [27] provides different dashboards based on road accidents datasets to identify rules based on which the accidents may happen (Figure 2).

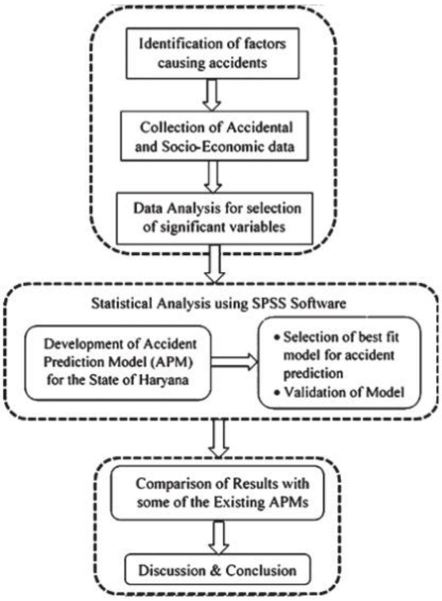


Figure 1: Methodology of research [26]

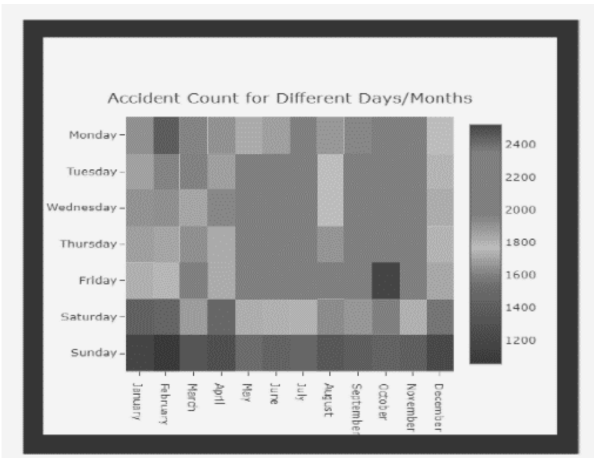


Figure 2: Accidents count for different days and month [27]

These dashboards are very useful in terms of identifying possible danger days for roading. But this and previous works do not investigate a possibility of real time response on violation defined rules. Current study aims to research whether it is possible to make a highly efficient model for analyzing the environment at the place and respond immediately to potential accident participants about found danger. This investigation is based on object detection and tracking on the potential danger places. So, the statistics of accidents may help to find such places.

3. Methods and materials

The following presented input data, models and models evaluation metrics are used for experiments running.

3.1. Data description

Problem requires usage of the raw data. The raw data for models training and testing is a video stream of potential danger places which captures road traffic. The computer game Beamng [28] is used for forming video stream and dataset of images for training. The Figure 3 display a training image for dataset. This picture shows the “ok” case, when all oncoming lanes are stopped to allow running the car which would like to turn left.

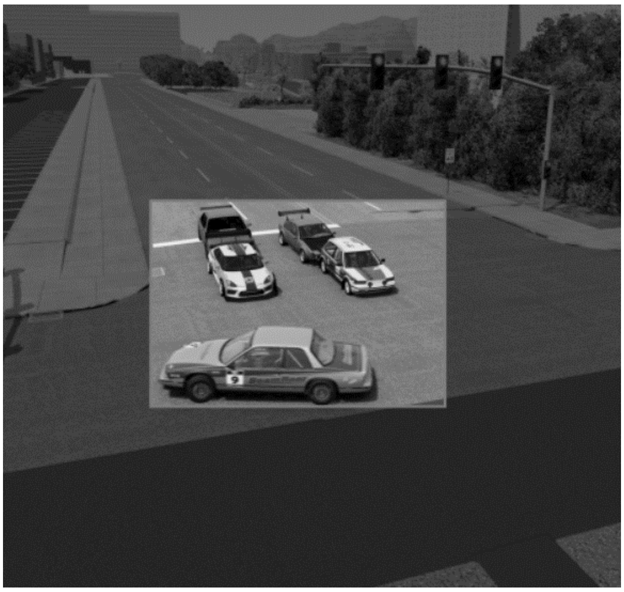


Figure 3: Example of training dataset [29]

In general, the dataset consists of two types of images: danger (which displays danger situation) and non-danger. Designed model is considered to be simple, so the only one type of danger situation is taken into account: turning left across oncoming lanes, when one of the lines is empty and other lanes contain stopped vehicles to skip a driver which would like to turn left. This situation describes a case, when some drivers are aware of another driver to turn left, but the driver is going on the empty lane and doesn't understand why other lanes are stopped. In case, if a driver, who would like to turn left, continues moving across the lines, he may unexpectedly appear in front of the driver which is going on the ‘empty’ lane. This leads to road accidents. Described case is displayed on the Figure 4.

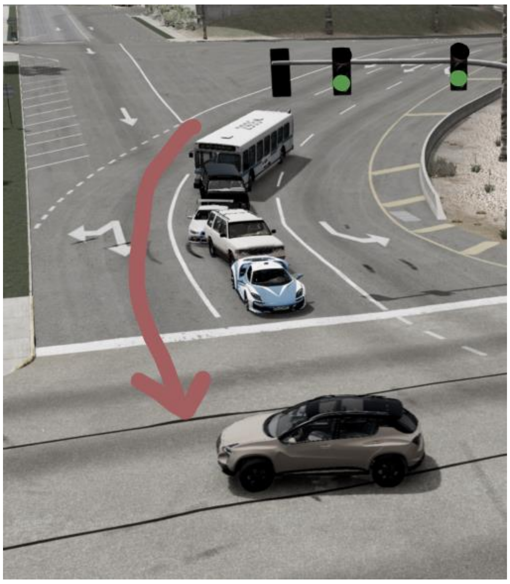


Figure 4: Description of danger situation pattern [29]

This research uses a dataset which includes 300 images. These images have 640x640 resolution and represent danger and non-danger patterns (Figure 5) in equals parts, means 150 images per case.

Figure 5: Safe turning left across oncoming lanes [29]

Also, the dataset contains images with different car setup, different cross roading places, different number of lanes and oncoming vehicles.

The dataset is prepared using a useful tool – Roboflow [30]. This web service allows users to upload, label and export dataset in a particular format, which suits for different machine learning object detection algorithms. The dataset is publicly available by the following link [29].

3.2. ML model validation and metrics

Each machine learning algorithm has a way to somehow measure the performance and effectiveness of trained models.

Current research uses Average Precision with IoU, Time to train, Occupied GPU memory, Model size, Loss *CLS*, Loss box, Inference time metrics.

The IoU is a value of correctness filling of a predicted box. Figure 6 displays in detail that metric.

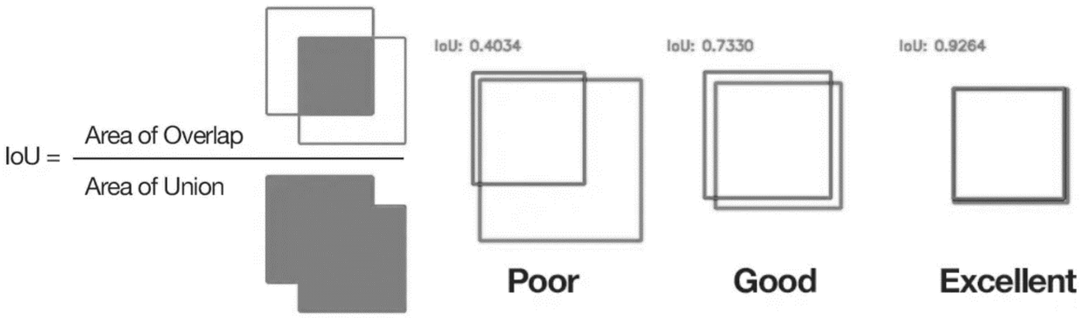


Figure 6: Explanation of IoU [31]

The formula of IoU is the following

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}. \tag{1}$$

The IoU (Intersection over union) is a good way to measure the degree of overlap between two bounding boxes or segmentation masks. If the prediction is perfect, IoU = 1, and if it misses completely, IoU = 0. The degree of overlap will give the IoU value between these two values. So, the precision usually is measured for different IoU values to identify the place of best performance.

The Loss Box: a loss that measures how "tight" the predicted bounding boxes are adjacent to the true object (usually regression loss, L1, smoothL1, etc.). The formula is the following

$$L_{box}(t^u, v) = \sum_{i \in \{x,y,w,h\}} L_1^{smooth}(t_i^u - v_i). \tag{2}$$

The bounding box loss should measure the difference between t_i^u and v_i using a robust loss function.

The Loss *CLS* (classification) is a loss that measures the correct classification of each predicted bounding box: each box may contain an object class or "background". This loss is usually called cross-entropy loss. The formula is the following

$$L_{cls}(p,u) = -\log\log p_u\,. \qquad (3)$$

The Precision is calculated in the following way

$$P = \frac{TruePositive}{(TruePositive + FalsePositive)}. \qquad (4)$$

3.3. ML models and methods

The recent development of object detection and tracking algorithms present Yolov7 [32] and Detectron2 [33], which allow the detection of an object and trace it with high frame rate. Current investigation uses Detectron2 with Faster R-CNN as backend model.

The following developers (Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao) present in the paper for Yolov7.

The YOLOv7 surpasses all known object detectors in both speed and accuracy in the range from 5 FPS to 160 FPS and has the highest accuracy 56.8% Average Precision among all known real-time object detectors with 30 FPS or higher on GPU V100.

YOLOv7-E6 object detector (56 FPS V100, 55.9% Average Precision) outperforms both transformer-based detector SWINL Cascade-Mask R-CNN (9.2 FPS A100, 53.9% Average Precision) by 509% in speed and 2% in accuracy, and convolutional based detector ConvNeXt-XL Cascade-Mask R-CNN (8.6 FPS A100, 55.2% Average Precision) by 551% in speed and 0.7% Average Precision in accuracy, as well as YOLOv7 outperforms: YOLOR, YOLOX, Scaled-YOLOv4, YOLOv5, DETR, Deformable DETR, DINO-5scale-R50, ViT-Adapter-B and many other object detectors in speed and accuracy. Moreover, we train YOLOv7 only on MS COCO dataset from scratch without using any other datasets or pre-trained weights. [32] The following comparison diagram exists in the paper (Figure 7).

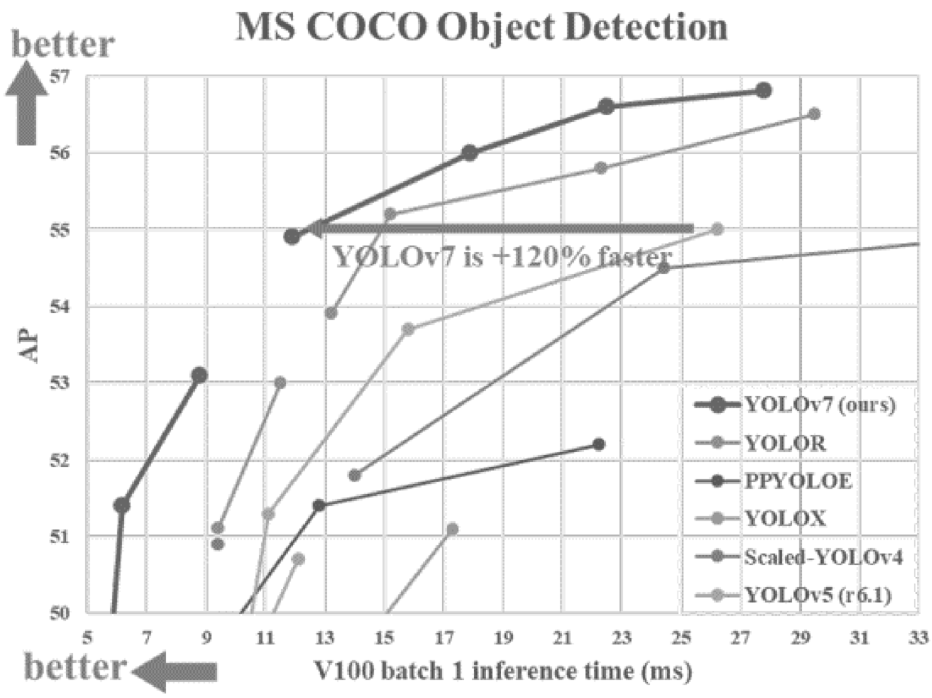


Figure 7: Comparison of YOLO algorithms performance [32]

The Yolov7 developers propose to use the Extended ELAN architecture (Figure 8) to achieve controlling the shortest longest gradient path, a deeper network can learn and converge effectively [32].

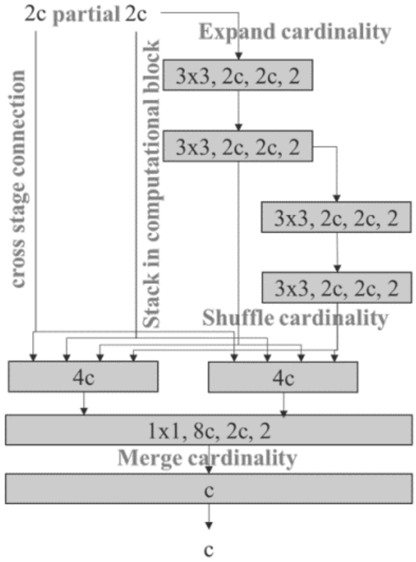


Figure 8: E-ELAN architecture [33]

The Detectron2 provides different models under the hood. One of the provided models is Faster R-CNN. A few words about Faster R-CNN: this algorithm has three parts [33]:

1. Backbone Network: extracts feature maps from the input image using different scales. Base RCNN-FPN's output features are called P2 (1/4 scale), P3 (1/8), P4 (1/16), P5 (1/32) and P6 (1/64).
2. Region Proposal Network: detects object regions from the multi-scale features. The 1000 box proposals with confidence scores are obtained.
3. Box Head: crops and warps feature maps using proposal boxes into multiple fixed-size features, and obtains fine-tuned box locations and classification results via fully-connected layers. Finally, 100 boxes (by default) in maximum are filtered out using non-maximum suppression (NMS). The box head is one of the subclasses of ROI Heads. For example, Mask R-CNN has more ROI heads such as a mask head.

The architecture overview presented on Figure 9.

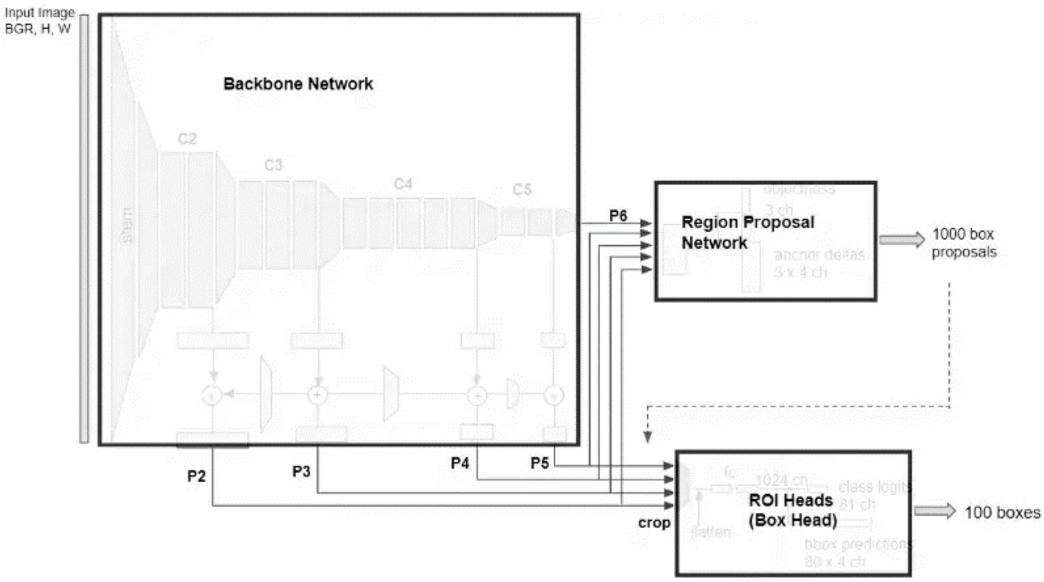


Figure 9: Faster R-CNN architecture [34]

4. Experiment

The experiment includes model training on a defined dataset [34] for understanding which one model will be better in comparison by presenting before metrics. The inference speed and accuracy are the most important here, because time to detect dangerous situations may cost a life.

In general, experiment consists of:

1. Experiment planning and code preparations;
2. Model training;
3. Evaluation and testing;
4. Results comparison.

4.1. Experiment planning

First, any training requires the creation of a dataset. The Dataset is collected by the Capture function in the Beamng game. Also, the car running scenario is defined by Script AI Manager, which may be selected by tapping on F11 with the following selection Window button. Running path is recorded when the user is driving a car. Also, the recorded path may be looped, this means that only pressing the capture button requires to collect the screenshots of the scenario.

After photos are collected, it may be uploaded into the Roboflow web service, which gives an ability to upload, label, and download a dataset in a big number of different ways: starting with selection of dataset format and finishing with code snippets of usage such dataset.

The Gradient Notebooks [35] service was considered a possible provider of hardware for model training. It has paid tiers. The Figure 10 displays training cost per 100 epochs.

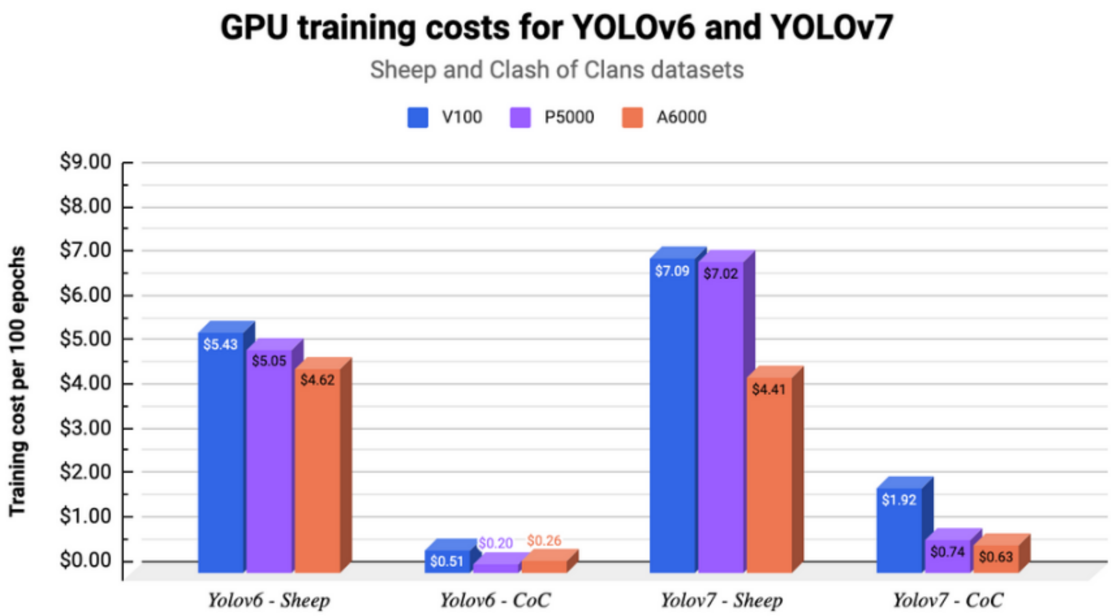


Figure 10: GPU training cost for Yolov6 and Yolov7 on Gradient Notebooks [36]

The available tiers with On-Demand cost are presented on Table 1.

Table 1
Paid tiers cost [35]

Tire name	Cost per Hour
P5000	0.78$
A6000	1.89$
V100	2.3$

The Google Colab service is used to run training, because this allows users to avoid installation of everything on their own machine and, also, the convenient way of usage of the Jupyter Notebook they provided. This service provides a free tier with GPU, so model training may not take a lot of time.

The code examples are located here [37-38].

4.2. ML models training

Selected models are trained over dataset which contains near 300 images which represent a danger and non-danger situations. The Yolov7 model has batch size = 16, image size = 640x640, epoch count = 55. The Detectron2 model has batch size 64, image size = 640x640 and epoch count = 1500. The hardware is provided by Google Colab: Python 3 Google Compute Engine backend (GPU: Tesla T4). The following Figure 11 displays resource consumption during Detectron2 model training.

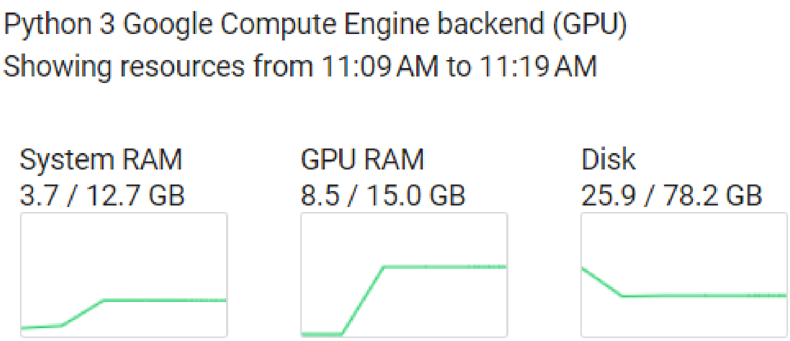


Figure 11: Resource consumption during training of Detectron2 model

As you can see, there RAM and GPU RAM consumption is not so high, but training on less powerful hardware may lead to training time increasing or even out of memory errors. The bigger consumption Yolov7 model has (Figure 12).

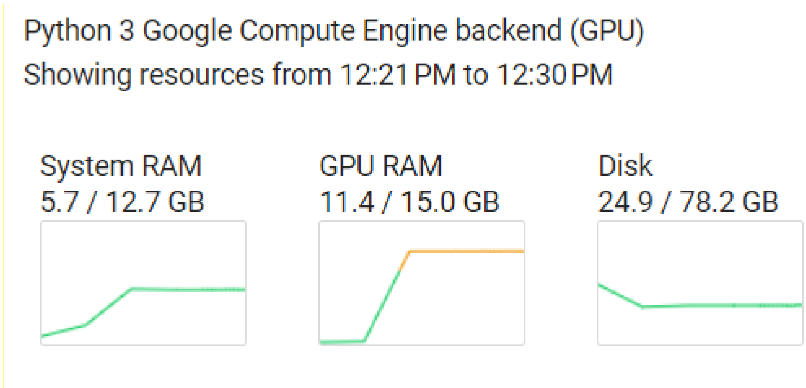


Figure 12: Resource consumption during training of Yolov7 model

It takes much more resources for training in comparison with Detectron2. The time for models training is displayed in Table 2.

Table 2
Average training time

Time, mins	Algorithm
51	Detectron2
15.9	Yolov7

5. Results

The evaluation results during training and testing are presented in and chart view.

5.1. Yolov7

The Yolov7 model has the following evaluation result presented in Table 3.

Table 3
Evaluation result during training Yolov7

Value	Metric
0.583	AP@.5:.95
0.926	AP@.5
0.191668 s / img per device	Inference time metrics
74.8 MB	Model size

In general, the 58% is not such a good result, but accuracy of 50% overlapping precision is quite good – 92%. The next figure (Figure 13) represents the loss metrics.

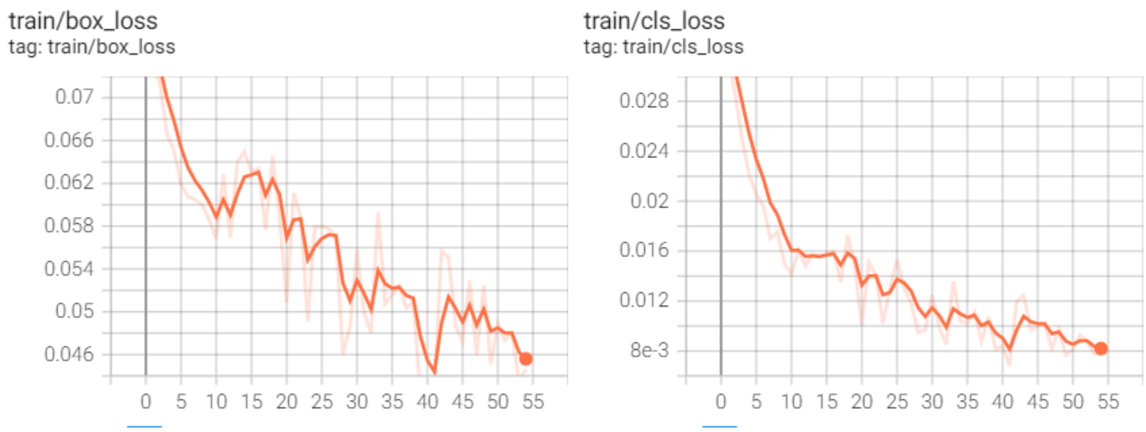


Figure 13: Loss Box and Loss *Cls* charts of Yolov7

The interesting finding that precision significantly decreased during running of ~42 epoch (Figure 14).

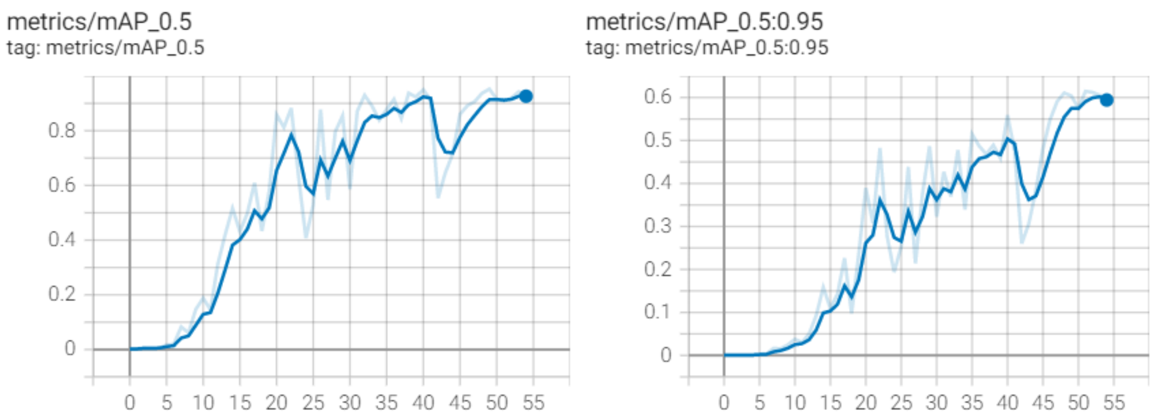


Figure 14: Mean AP metrics of Yolov7

The inference results are on Figure 15. There are no mistakes or incorrect predictions observed.

Figure 15: Inference result of Yolov7 [28]

5.2. Detectron2

The Detectron2 model has the following evaluation result presented in Table 4.

Table 4
Evaluation result during training Detectron2

Value	Metric
0.651	AP@.5:.95
0.909	AP@.5
0.198095 s / img per device	Inference time metrics
815 MB	Model size

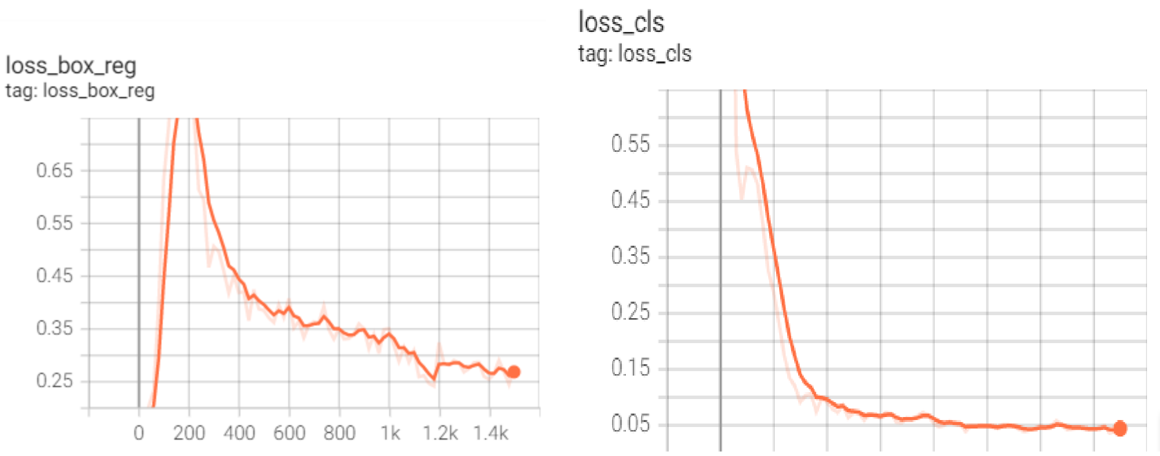


Figure 16: Loss Box and Loss *Cls* charts of Detectron2

The 65% precision result is bigger than in Yolov7, and accuracy of 50% overlapping precision is near on the same level. The next figure (Figure 16) represents the loss metrics.

Also, the Faster R-CNN metrics are available (Figure 17).

The inference results are presented on Figure 18. I would like to mention, Detectron2 results contain some mistakes during predictions, which are not observed during prediction with Yolov7. The model is taking care of another car, which is going straight forward.

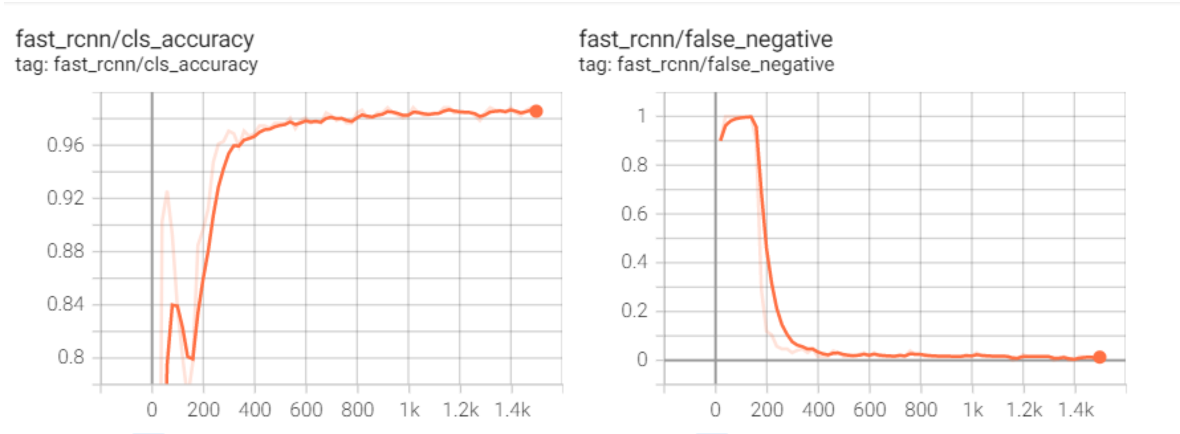
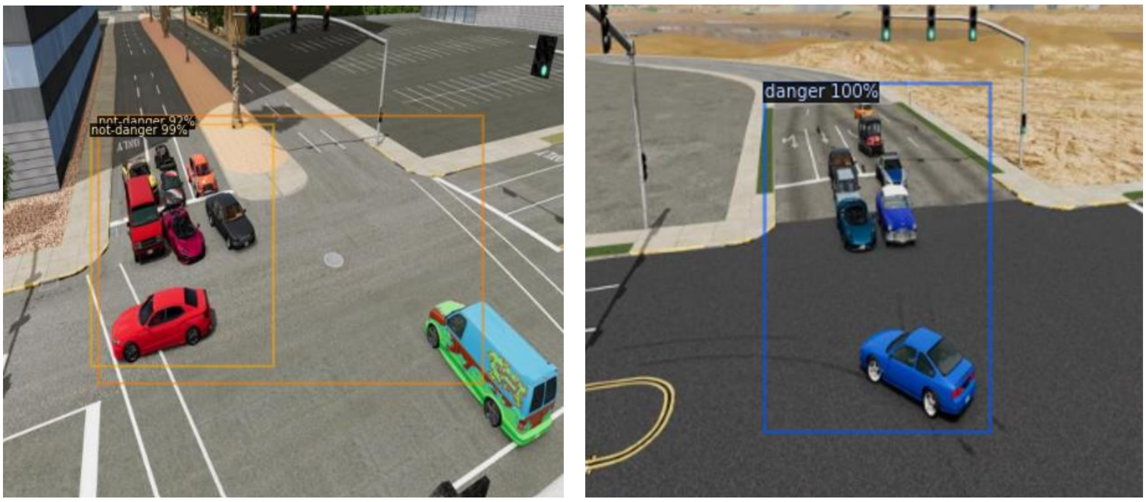


Figure 17: Faster R-CNN metrics



a) b)

Figure 18: Inference result with a covering of a car which is going straight forward (a); Inference result with great accuracy (b) [28]

6. Discussions

Discussion section contains the model's results comparison and identifying the best algorithm which is best suited for the problem. The Table 5 displays compilation results for mentioned algorithms.

Table 5
Compilation results

Metric Name	Detectron2 (Faster R-CNN)	Yolov7
AP@.5:.95	0.651	0.583
AP@.5	0.909	0.926
Inference time metrics	0.198095 s / img per device	0.191668 s / img per device
Model size	815 MB	74.8 MB
GPU RAM	8.5 Gb	11.4 Gb
Time to train	51 mins	15.9 mins
Epoch	1500	55

Results display the following: with small benefit in accuracy, Detectron2 is worse than Yolov7 in comparison by other parameters. Because near the same accuracy is achieved in even bigger time and used resources.

The better result may be obtained using a bigger dataset, because 300 images is quite a small number of pictures to create a well-performing model.

6.1. Recommendations

The multi-resolution training may increase model performance of Yolov7, especially, when different image sizes may be. In case of multi-resolution training, the images will be resized to +-50%. So, for 640×640 images, the minimum resolution will be 320×320 and the maximum resolution will be 1280×1280. This helps to train a more robust model.

The Yolov7's model reparameterization may significantly improve model performance. The model reparameterization method combines several computational modules into one at the inference stage, which gives us better inference throughput. It can be considered an ensemble technique and can be divided into two categories: module-level ensemble and model-level ensemble. There are two methods of model-level reparameterization. One is to train several identical models with different training data and then average the weights of all models. The second method is to calculate the weighted average of the model weights across different iterations. Whereas the module-level method splits a module into several identical or different branches during training and combines the branched modules into one during inference.

Other approaches may be used for increasing the performance of Detectron2. The one way to increase FPS is to reduce the image resolution. The lower the image resolution, the faster inference per image will be obtained.

An image resolution lower than the one specified in the Zoo Model will help speed up the process, especially if you are using it for video, where the difference can be a few milliseconds.

Another way is to skip frames, if you are using a video input and there is no fast movement in the video, skipping a few frames does not cause any harm as the main information is not lost. In addition to skipping frames, use streaming when reading frames. Make sure that only the last frame is processed. To do this, you may need to make some changes before making predictions in predictor.py.

If you want to speed up the visualization process, Detectron2 provides parallel processing. In the predictor.py file: change the parallel=True parameter to run the model in different processes.

6.2. Research result usage

The research result is a background of future investigations for creation of a well-trained model, which may have the ability to be run on a one board computer, in case it is possible from hardware view. Also, investigation results may be used for researchers which need to decide what model is best suited for their problem. So, if the investigation problem is two class classification, which requires a little amount of time to train and detect, the Yolov7 is the best.

7. Conclusions

This research provides a source for thinking about what image detection and tracking algorithm is better for quick classification of danger situations on road. Road accidents are a great problem nowadays based on statistics of injury and death, where the root cause is a road accident.

Related works cover slightly another side of the problem – statistical approaches for road accidents prediction and using different regression algorithms for prediction. The following factors may impact the road accident happening from their point of view: weather, road condition, day time, day of week, season, speed, car condition, etc.

Dataset has been prepared by Beamng Drive game, AI script manager and Roboflow service. Beamng Drive is a playground for building everything related to cars braking. The 300 images with different combinations of road intersections, outdoors, cars setup, etc were captured and labeled using

Roboflow webservice. Also, Roboflow allows image resizing and dataset splitting in some parts. Different dataset types exist to be used for dataset extraction and usage in different models. The 640x640 pixels size of the picture is used.

The Average Precision, Inference time, Time for training, Model size, GPU RAM usage metrics have been used in this research. Also, the Union over intersection metric is considered during calculation of Average Precision.

The Detectron2 (with Faster R-CNN) and Yolov7 algorithms have been compared in current research. They have different architecture and their comparison is a main goal of this research.

The Google Colab GPU environment is used for model training and evaluation. The Tesla T4 GPU is used for models training. The 51 minutes Detectron2 model training takes in comparison to Yolov7, where training time is 15.9 minutes.

Based on evaluation results, the Yolov7 has better performance than Detectron2. Also, this research is a base research for further investigations in order to understand whether current approaches may be applied for one-board computer. And researchers which have the same two-classes classification may consider these research results.

8. References

[1] C. Zhang, Y. Shu and L. Yan, "A Novel Identification Model for Road Traffic Accident Black Spots: A Case Study in Ningbo, China," in IEEE Access, vol. 7, pp. 140197-140205, 2019, doi: 10.1109/ACCESS.2019.2942647.

[2] D. E. Benrachou, S. Glaser, M. Elhenawy and A. Rakotoniraigny, "Use of Social Interaction and Intention to Improve Motion Prediction Within Automated Vehicle Framework: A Review," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 12, pp. 22807-22837, Dec. 2022, doi: 10.1109/TITS.2022.3207347.

[3] M. Manzoor et al., "RFCNN: Traffic Accident Severity Prediction Based on Decision Level Fusion of Machine and Deep Learning Model," in IEEE Access, vol. 9, pp. 128359-128371, 2021, doi: 10.1109/ACCESS.2021.3112546.

[4] G. Xie, A. Shangguan, R. Fei, X. Hei, W. Ji and F. Qian, "Unmanned System Safety Decision-Making Support: Analysis and Assessment of Road Traffic Accidents," in IEEE/ASME Transactions on Mechatronics, vol. 26, no. 2, pp. 633-644, April 2021, doi: 10.1109/TMECH.2020.3043471.

[5] Y. Qu, Z. Lin, H. Li and X. Zhang, "Feature Recognition of Urban Road Traffic Accidents Based on GA-XGBoost in the Context of Big Data," in IEEE Access, vol. 7, pp. 170106-170115, 2019, doi: 10.1109/ACCESS.2019.2952655.

[6] N. S. Hadjidimitriou, M. Lippi, M. Dell'Amico and A. Skiera, "Machine Learning for Severity Classification of Accidents Involving Powered Two Wheelers," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 10, pp. 4308-4317, Oct. 2020, doi: 10.1109/TITS.2019.2939624.

[7] F. Tabei, B. Askarian and J. W. Chong, "Accident Detection System for Bicycle Riders," in IEEE Sensors Journal, vol. 21, no. 2, pp. 878-885, 15 Jan.15, 2021, doi: 10.1109/JSEN.2020.3021652.

[8] K. Smelyakov, A. Chupryna, M. Hvozdiev and D. Sandrkin, "Gradational Correction Models Efficiency Analysis of Low-Light Digital Image," 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream), 2019, pp. 1-6, doi: 10.1109/eStream.2019.8732174.

[9] K. Smelyakov, M. Hvozdiev, A. Chupryna, D. Sandrkin and V. Martovytskyi, "Comparative Efficiency Analysis of Gradational Correction Models of Highly Lighted Image," 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 2019, pp. 703-708, doi: 10.1109/PICST47496.2019.9061356.

[10] M. Usman, A. Carie, B. Marapelli, H. D. Bedru and K. Biswas, "A Human-in-the-Loop Probabilistic CNN-Fuzzy Logic Framework for Accident Prediction in Vehicular Networks," in IEEE Sensors Journal, vol. 21, no. 14, pp. 15496-15503, 15 July15, 2021, doi: 10.1109/JSEN.2020.3023661.

[11] J. Fior and L. Cagliero, "Correlating Extreme Weather Conditions with Road Traffic Safety: A Unified Latent Space Model," in IEEE Access, vol. 10, pp. 73005-73018, 2022, doi: 10.1109/ACCESS.2022.3190399.

[12] T. K. Vijay, D. P. Dogra, H. Choi, G. Nam and I. -J. Kim, "Detection of Road Accidents Using Synthetically Generated Multi-Perspective Accident Videos," in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 2, pp. 1926-1935, Feb. 2023, doi: 10.1109/TITS.2022.3222769.

[13] X. Mi, B. Yang, Z. Dong, C. Chen and J. Gu, "Automated 3D Road Boundary Extraction and Vectorization Using MLS Point Clouds," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 6, pp. 5287-5297, June 2022, doi: 10.1109/TITS.2021.3052882.

[14] A. Arsenov, I. Ruban, K. Smelyakov and A. Chupryna, “Evolution of convolutional neural network architecture in image classification problems”, Selected Papers of the XVIII International Scientific and Practical Conference “Information Technologies and Security” (ITS 2018), Kyiv, Ukraine, November 27, 2018. In CEUR Workshop Proceedings, Vol-2318, 2018, pp. 35-45. https://ceur-ws.org/Vol-2318/

[15] J. Fang, J. Qiao, J. Bai, H. Yu and J. Xue, "Traffic Accident Detection via Self-Supervised Consistency Learning in Driving Scenarios," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 9601-9614, July 2022, doi: 10.1109/TITS.2022.3157254.

[16] H. Hayashi, M. Kamezaki and S. Sugano, "Toward Health–Related Accident Prevention: Symptom Detection and Intervention Based on Driver Monitoring and Verbal Interaction," in IEEE Open Journal of Intelligent Transportation Systems, vol. 2, pp. 240-253, 2021, doi: 10.1109/OJITS.2021.3102125.

[17] M. Zheng et al., "Traffic Accident’s Severity Prediction: A Deep-Learning Approach-Based CNN Network," in IEEE Access, vol. 7, pp. 39897-39910, 2019, doi: 10.1109/ACCESS.2019.2903319.

[18] W. Qi, W. Wang, B. Shen and J. Wu, "A Modified Post Encroachment Time Model of Urban Road Merging Area Based on Lane-Change Characteristics," in IEEE Access, vol. 8, pp. 72835-72846, 2020, doi: 10.1109/ACCESS.2020.2987959.

[19] L. Pozueco et al., "Analysis of Driving Patterns and On-Board Feedback-Based Training for Proactive Road Safety Monitoring," in IEEE Transactions on Human-Machine Systems, vol. 50, no. 6, pp. 529-537, Dec. 2020, doi: 10.1109/THMS.2020.3027525.

[20] Y. W. Liyanage, D. -S. Zois and C. Chelmis, "Near Real-Time Freeway Accident Detection," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 2, pp. 1467-1478, Feb. 2022, doi: 10.1109/TITS.2020.3027494.

[21] C. Ryan, F. Murphy and M. Mullins, "End-to-End Autonomous Driving Risk Analysis: A Behavioural Anomaly Detection Approach," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 3, pp. 1650-1662, March 2021, doi: 10.1109/TITS.2020.2975043.

[22] O. Lemeshko, O. Yeremenko, M. Yevdokymenko, A. Shapovalova, A. M. Hailan and A. Mersni, "Cyber Resilience Approach Based on Traffic Engineering Fast ReRoute with Policing," 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Metz, France, 2019, pp. 117-122, doi: 10.1109/IDAACS.2019.8924294.

[23] Tereshchenko, G., Gruzdo, I., "Overview and Analysis of Existing Decisions of Determining the Meaning of Text Documents," 2018 International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 2019, pp. 645–653, 8632014, doi: 10.1109/INFOCOMMST.2018.8632014.

[24] Sharonova, N., Kyrychenko, I., Gruzdo, I., Tereshchenko, G., “Generalized Semantic Analysis Algorithm of Natural Language Texts for Various Functional Style Types”, 2022 6th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2022), 2022. – CEUR-WS 3171, 2022, ISSN 16130073. - Volume I: Main, PP. 16 - 26.

[25] H. -J. Jeon, V. D. Nguyen, T. T. Duong and J. W. Jeon, "A Deep Learning Framework for Robust and Real-Time Taillight Detection Under Various Road Conditions," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 11, pp. 20061-20072, Nov. 2022, doi: 10.1109/TITS.2022.3178697.

[26] N. Mor and T. Goyal, Application of machine learning technique for prediction of road accidents in Haryana-A novel approach. [Online]. URL:

https://www.researchgate.net/publication/340824266_Application_of_machine_learning_technique_for_prediction_of_road_accidents_in_Haryana-A_novel_approach. [Accessed: 14-Feb-2023].
[27] N. Soni and M. Patel, "An efficient road accident prediction using Machine Learning Techniques," An Efficient Road Accident Prediction Using Machine Learning Techniques. [Online]. URL: https://www.researchgate.net/publication/359814996_An_Efficient_Road_Accident_Prediction_Using_Machine_Learning_Techniques. [Accessed: 14-Feb-2023].
[28] BeamNG.drive. [Online]. URL: https://www.beamng.com/game/. [Accessed: 14-Feb-2023].
[29] O. Byzkrovnyi, "Roboflow," Roboflow Dataset. [Online]. URL: https://app.roboflow.com/ds/polJ55iQ4a?key=zYXz2EmCeK. [Accessed: 14-Feb-2023].
[30] Roboflow. [Online]. URL: https://app.roboflow.com/. [Accessed: 14-Feb-2023].
[31] "Intersection over union (IOU)," Hasty.ai, 07-Feb-2023. [Online]. URL: https://hasty.ai/docs/mp-wiki/metrics/iou-intersection-over-union. [Accessed: 14-Feb-2023].
[32] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv.org, 06-Jul-2022. [Online]. URL: https://arxiv.org/abs/2207.02696. [Accessed: 14-Feb-2023].
[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," arXiv.org, 06-Jan-2016. [Online]. URL: https://arxiv.org/abs/1506.01497. [Accessed: 14-Feb-2023].
[34] O. Byzkrovnyi, "Roboflow," Roboflow Dataset. [Online]. URL: https://app.roboflow.com/ds/polJ55iQ4a?key=zYXz2EmCeK. [Accessed: 14-Feb-2023].
[35] "Gradient notebooks," Paperspace. [Online]. URL: https://www.paperspace.com/gradient/notebooks. [Accessed: 15-Feb-2023].
[36] J. Robison, "Benchmarking yolov6 and Yolov7 on Paperspace gradient with the help of Roboflow datasets," Paperspace Blog, 23-Aug-2022. [Online]. URL: https://blog.paperspace.com/benchmarking-yolov6-yolov7-paperspace-roboflow-datasets/. [Accessed: 15-Feb-2023].
[37] "Google colaboratory," Google Colab. Detectron2. [Online]. URL: https://colab.research.google.com/drive/1vTh-bC31puePqn68K8lY41eyCAm5DrUt?usp=share_link. [Accessed: 14-Feb-2023].
[38] "Google Colab. Yolov7," Google Colab. [Online]. URL: https://colab.research.google.com/drive/1kaVsuWK6DfXykeae1PazYH0Yw-ZVzZ-l?usp=share_link. [Accessed: 14-Feb-2023].