Applying Transfer Learning to Traffic Surveillance Videos for Accident Detection

Ajeet R. Pathak & Anne C. Elster

ICAPAI 2022 - PREPRINT!!

***Abstract*—Automated traffic video surveillance is a crucial research domain in the field of computer vision due to the need for enabling highway safety. It is very important to detect incidents of road accidents from traffic surveillance videos in an automated manner in order to take necessary actions and save the lives of people and properties. Motivated by the same, this paper proposes a method for detecting the road accidents from traffic surveillance videos in an automated manner. Specifically, we use an object-centric accident detection model using the YOLOv2 architecture based on transfer learning technique. The YOLOv2 model is a homogeneous convolutional architecture that makes it faster to predict bounding boxes. In this work, we include a brief description of the YOLOv2 architecture and how we fine-tune a 32-layer variant pre-trained on the VOC dataset to our custom accident dataset. Our experiments using a real-world anomaly detection dataset, shows the significant results in terms of mean average precision. Moreover, our model works in real-time achieving 60FPS on an NVIDIA Tesla K80 GPU and ~16.67 FPS on a standard laptop with a 4GB GT GPU. Our implementation can thus provide a near real-time accident localization with 76% mAP on the road accident dataset.**

Keywords—Accident detection, machine learning, transfer learning, real-time surveillance, YOLOv2

I. Introduction

Due to proliferation of Internet of Things and Artificial Intelligence, smart city based projects have begun flourishing with an aim to enhance the quality of human lives [1]. Smart and sustainable traffic management and surveillance is one of the major goals in smart city based projects. Roads plays a crucial role in transport of people and goods. In the category of various traffic hazards, road accident is a serious issue in transportation domain. According to the World Health Organization (WHO), motorcyclists, pedestrians, and cyclists are the most vulnerable road users [2].

To mitigate the chances of losing lives and properties by taking appropriate actions in case of accident fatalities, the premises covering the main roads, highways, and squares are deployed with closed-circuit television (CCTV) cameras. However, continuously checking such videos is not practical. It is, therefore, necessary to automate the task of identifying the accidents. Typically, techniques in computer vision are used to automate the task of analyzing such traffic surveillance videos which involve segmentation of entities, classification of all types of vehicles, and extracting the spatio-temporal features. These features collectively are then used for related tasks such as tracking and counting of vehicles, detecting the crashes and accidents, road surface monitoring, to mention a few.

There are two categories of tasks to mitigate the chances of accidents. These are 1) *Proactive tasks* in which probable conditions before occurrence of actual crash incidents are detected. Such systems predict the risk of happening accidents (For example, drowsiness detection of drivers, road surface monitoring) 2) *Corrective tasks* in which corrective actions are undertaken to control the situation after occurrence of unusual incidents such as accidents. However, to take corrective actions by human personnel, it is necessary to detect the accidents in a real-time manner. Motivated by the latter, this paper describes how to apply a transfer learning approach for object-centric accident detection using the You Only Look Once (YOLO) architecture [3]. Specifically, the model performs the task of real-time detection of accident scenarios from videos.

The contributions of the paper are given as follows.

1. This paper addresses an issue of detecting the accident scenes from surveillance videos in a real-time manner.
2. Transfer learning and YOLOv2 model have been utilized for accident detection and the performance of the model in terms on mean Average Precision (mAP) and frames per second (FPS) has been analyzed.
3. A new accident detection dataset has been created by combining road accident scenes from UCF-Crime [4] and IITH accident datasets [5] and annotated with bounding box coordinates covering accident events for training and evaluation.

The rest of the paper is organized as follows. Related work on accident detection approaches is discussed in Section II. Section III deals with problem formulation and transfer learning approach based on YOLOv2 model for accident detection. Section IV gives experimentation details encompassing dataset creation, model training and fine-tuning. Quantitative and qualitative results have been described in Section V. The future enhancements and conclusion have been given in Section VI.

II. Related Work

Substantial research in the domain of accident detection has been carried out in the past decade and is currently ongoing [25-29]. However, very few approaches have supported real-time performance in addition to detecting the accidents. A few of the related research works we found most interesting is discussed in this section. Jiang et al. [6] proposed traffic accident detection model based on Single Shot MultiBox Detector (SSD), multi-feature fusion technique and adaptive default frame selection algorithm. The model is evaluated using manually collected traffic dataset. This approach supports accuracy and real-time performance.

Two-stream Convolutional Neural Network (CNN) based architecture proposed in [7] has spatial network for detecting the objects and temporal network for tracking the multiple objects by utilizing the motion features. This approach supports three tasks mainly object detection, classification, and tracking. Spatial pixel segmentation technique is used for predicting the bounding boxes of multiple objects.

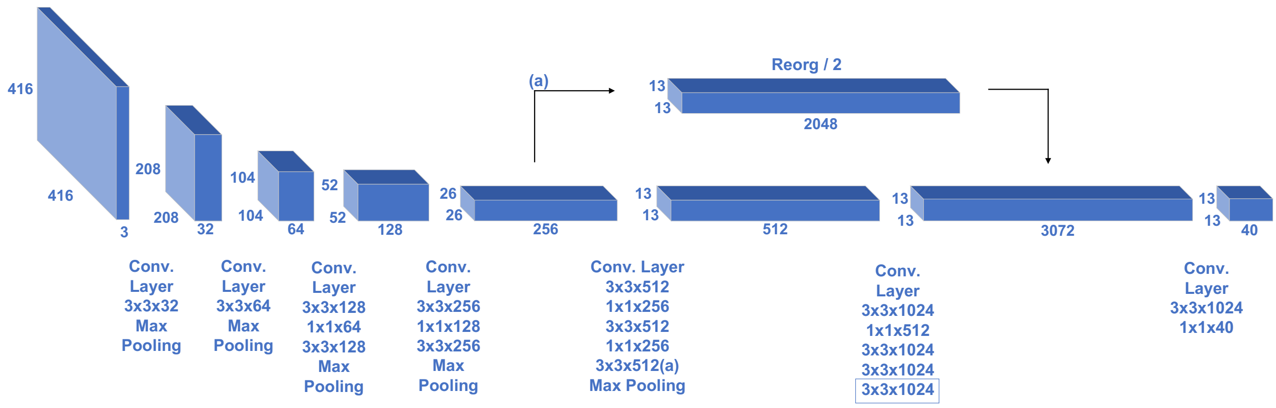

Fig.1. Default architecture of YOLOv2 [30]

Choi et al. [8] proposed car crash detection system based on multimodal data which uses ensemble technique based on gated recurrent unit (GRU) and convolutional neural network (CNN). YouTube videos of car accidents have been used for validating the model. The scope of their approach is limited, as only dashboard cameras have been used and dashboard cameras does not provide whole view which is generally provided by surveillance cameras. Our approach, on the other hand, works on surveillance videos to get full notion of the accident. Accident detection model proposed by Vishnu et al. [9] follows Hybrid Support Vector Machine (SVM), multinomial logistic regression and histogram of gradient flow features. Ijjina et al. [10] applied Mask Region-Based Convolutional Neural Networks (R-CNN) for object detection followed by centroid-based object tracking algorithm. Ren et al. [11] proposed a recurrent neural network approach based to predict traffic accident risk based on spatial-temporal patterns of accident data. Tian et al. [12] proposed a deep neural network model based on one-stage YOLO model and leveraged multi-scale feature fusion scheme and loss function having dynamic weights.

There are many research works which formulate the problem of accident detection as anomalous activity detection [13],[14],[15],[16]. Maaloul et al. [14] applied anomaly detection approach in which traffic motion flow modelling technique is applied. Initially, normal traffic flow pattern is modelled. After occurrence of accident, sharp change in velocity vector occurs which implies accident event. Sultani and Choi [15] used interaction among the cars as a pointer for accident detection. They applied driver model based on Latent Dirichlet Allocation (LDA) and particle advection technique for detecting the abnormalities in videos. Yun et al. [16] applied Physics-inspired approach based on motion interaction field to detect abnormalities such as accidents from the videos.

There are some works which used YOLO architectures for detecting the car accidents [7], [12], [31]. The model proposed in [31], combines YOLO v3 model and Canny edge detection algorithm for detecting highway accidents. However, these works address the task of accident detection using hybrid approach in which YOLO model and some other feature extractor is used. Our approach, on the contrary, is simple which uses YOLOv2 model and applies transfer learning approach giving good detection performance. Moreover, there have been various works on accident detection from videos, very few works deal with real-time traffic surveillance, which is the goal of our work.

III. Methodology

As mentioned earlier, the YOLOv2 model [3] is a homogeneous convolutional architecture that makes it faster to predict bounding boxes. A simplified architecture diagram of this model is shown in Figure 1. However, for this research work, we used a 32 layer variant of YOLOv2 model pre-trained on VOC dataset and fine-tuned it on our custom accident dataset. Table I enlists a clear description about each layer and their outputs based on architecture mentioned in figure 1. The YOLOv2 model quickly became a milestone model for object detection when it was introduced by Redmon and Farhadi in 2016 in their milestone paper [3]. The model showcased high performance (76.8 mAP on VOC dataset) paired with real-time predictions at 67 FPS when run on Geforce GTX Titan X GPU.

In our work, we showed that the problem of accident detection can be formulated as an object-centric detection of vehicle-crashes using the YOLOv2 architecture and transfer learning. With reference to insights obtained research work related to object detection [17], [18], [19], we chose to apply object-centric detection approach for detecting the accidents using YOLOv2 framework [3] and transfer learning [20]. Various research works have been further conducted in order to investigate the applications of YOLOv2 model. Although newer versions of YOLO models such as YOLOv3 [21] and YOLOv4 [22] are already available, we decided to utilize the YOLOv2 model for accident detection task. This is because that the architecture of YOLOv2 model is much simpler than its newer versions which allows the model to perform in real time on modern computers. In terms of identifying the accidents, achieving high performance at real-time using limited resource would be highly crucial which ultimately led us to proceed our research using YOLOv2 model.

In the rest of this section, we will give a brief description of the YOLOv2 architecture. In YOLOv2, the input layer accepts an image and convolutional blocks are applied on the images to extract feature maps at different levels. In the middle of the network, we also have a passthrough module that takes feature maps from intermediate layers and concatenate with a feature map towards the end of the network as shown in Figure 1. In YOLOv1, fully connected layers are present at the end of the network that predicts the bounding box coordinates. However, the major difference in YOLOv2 is the usage of anchors or anchor boxes.

Unlike Region proposal network based models such as Faster R-CNN, YOLOv2 uses anchor boxes in order to predict bounding box of an object.

Anchor boxes are fixed set of bounding boxes present in the final convolutional layers of the architecture. They are applied on the feature map and bounding boxes are learned by training the model to learn their offsets when compared with ground truth.

TABLE I. LAYERWISE DETAILS OF YOLOV2 ARCHITECTURE

Layer description	**Output size**
input	(None,416,416, 3)
cony 3x3p1_1 +bnorm	(None,416,416, 32)
maxp 2x2p0_2	(None,208,208, 32)
cony 3x3p1_1 +bnorm	(None,208,208, 64)
maxp 2x2p0_2	(None,104,104, 64)
cony 3x3p1_1 +bnorm	(None,104,104, 128)
cony lx1p0_1 +bnorm	(None,104,104, 64)
cony 3x3p1_1 +bnorm	(None,104,104, 128)
maxp 2x2p0_2	(None,52,52, 128)
cony 3x3p1_1 +bnorm	(None,52,52, 256)
cony lx1p0_1 +bnorm	(None,52,52, 128)
cony 3x3p1_1 +bnorm	(None,52,52, 256)
maxp 2x2p0_2	(None,26,26, 256)
cony 3x3p1_1 +bnorm	(None,26,26, 512)
cony lx1p0_1 +bnorm	(None,26,26, 256)
cony 3x3p1_1 +bnorm	(None,26,26, 512)
cony lx1p0_1 +bnorm	(None,26,26, 256)
cony 3x3p1_1 +bnorm	(None,26,26, 512)
maxp 2x2p0_2	(None,13,13, 512)
cony 3x3p1_1 +bnorm	(None,13,13, 1024)
cony lx1p0_1 +bnorm	(None,13,13, 512)
cony 3x3p1_1 +bnorm	(None,13,13, 1024)
cony lx1p0_1 +bnorm	(None,13,13, 512)
cony 3x3p1_1 +bnorm	(None,13,13, 1024)
cony 3x3p1_1 +bnorm	(None,13,13, 1024)
cony 3x3p1_1 +bnorm	(None,13,13, 1024)
concat [16]	(None,26,26, 512)
cony lx1p0_1 +bnorm	(None,26,26, 64)
local flatten 2x2	(None,13,13, 256)
concat [27, 24]	(None,13,13, 1280)
cony 3x3p1_1 +bnorm	(None,13,13, 1024)
cony lx1p0_1 linear	(None,13,13, 40)

Hence, when using the YOLOv2 model, we are training the model to learn these offsets of anchors so that an anchor box's shape and position could be adjusted to contain the object under study. By converting the problem of object detection into bounding box regression problem, YOLOv2 has been able to perform faster in localizing the objects as the feature maps of input image could be efficiently used with higher mAP values [3].

The YOLOv2 model also encompasses other modules such as high-resolution classifier, dimensionality cluster, fine-grained features, and multi-scale training. Another contribution given by the authors in the YOLOv2 architecture is the usage of Batch Normalization at each of the convolutional block. This step has eliminated the need for implementing other types of normalization to the training dataset to eliminate noise. Additionally, by using batch normalization at each convolutional block, it enabled the model to achieve convergence faster while also preventing overfitting.

IV. IMPLEMENTATION AND EXPERIMENTATION DETAILS

Our implementation is done in Python. We used an open-source neural network framework – Darknet in which YOLOv2 has been implemented by Redmon and Farhadi [3]. FFmpeg, OpenCV and Pandas are few significant packages used.

A. Datasets

For the task of accident detection, two datasets have been selected. The road accident scenes from UCF-Crime dataset [4] have been used for training the model and IITH road accident dataset [5] is used for testing the model.

1) UCF-Crime detection dataset

In UCF-Crime detection dataset, there are surveillance videos of 13 categories such as road accidents, burglary, assault, explosion, vandalism which can be utilized for anomaly detection. Out of 13 categories, we selected videos pertaining to the road accidents only to train the model.

2) IITH road accident dataset

The IITH road accident dataset [5] is a real-world surveillance video dataset containing road accidents captured by CCTV cameras deployed in the city of Hyderabad, India. This dataset is used for assessment (testing) of the model in this paper. The major challenge with this dataset is that accident scenes are captured at varying environment conditions such as bright sunlight, night, and early morning. Moreover, videos are captured with different cameras and view angles. These challenges make accident detection a difficult task. Out of 33,281 frames in dataset, we considered 668 frames which have road accident scenes. We executed the model on these video frames. Figure 2 shows sample images from IITH road accident dataset used for testing.

B. Preprocessing

When using the YOLOv2 model, the training data should be given as images along with corresponding bounding box annotations in the required YOLOv2 format. Since both the training and test datasets are available in video format, we first extracted the frames containing road accident scenes in it. For this, we defined road accident frames in a video as frames where the vehicle starts to deviate from its intended moving direction and end at the frame where the vehicle gets crashed on to another vehicle or object or humans. Using this definition, we first marked the starting and ending frame number of road accident in each of the train and test videos and then extracted the marked frames using FFMPEG tool in python. After extracting the frames from videos, the next step is to get bounding box coordinates of road accident in each frame.

Fig. 2. Sample images from IITH road accident dataset

Bounding boxes are needed to highlight the scenes of accidents for training purpose, whereas, for comparing the predicted bounding boxes with the ground truth bounding boxes, manually marking the bounding boxes is necessary. Therefore, we manually marked the bounding box coordinates of each extracted frame using the tool labellmg [23] from both train and test videos in the following format.

```
<Category ID, center w.r.t. x coordinate, center
w.r.t. y coordinate, width, height>
```

Category ID represents presence of accident incident (0 for accident incident). The second and third entities in above format represent center x and center y coordinate of the bounding box respectively. The fourth and fifth entities respectively represent width and height of the bounding box. Figure 3 shows examples of road accident video frames from the UCF-Crime dataset with annotated bounding boxes over accidents. After annotating the training and test dataset frames with bounding box coordinates, we fine-tuned the YOLOv2 model. The video frames do not require any additional preprocessing steps such as normalization or scaling with the exception of annotation. As mentioned before, YOLOv2 is comprised of a batch normalization step at each convolutional block. This further reduces the need to normalize the images separately. In order to augment the dataset, the images are resized during training time by the Darknet framework itself.

C. Model training and fine-tuning

After installing the necessary programs in a Python anaconda environment, we first downloaded the pre-trained YOLOv2 model. It has been widely established that fine-tuning a deep learning model that has previously been trained on a large-scale dataset may greatly improve the model's performance [20], [24]. Transfer Learning is the process of initially training the model on a large-scale dataset and then fine-tuning it on another dataset. The schematic of transfer learning technique applied in this paper is depicted in figure 4. The weights of each layer are changed in a pre-trained model by numerous rounds of model training on a large-scale dataset. When we refine a model using these updated weights rather than training with randomly initialized weights, the model's ability to learn contributing features of each class improves [24].

After configuring the environment, we used our training dataset to fine-tune the YOLOv2 32 layer model. The values of model hyperparameters are listed in Table II.

Based on the training technique described in the original YOLOv2 study [3], we set the initial learning rate to 0.01. We lowered the learning rate by a factor of 10 per 1000th iteration to force the model to learn the local minimum. Additionally, we adjusted the weight decay to 0.0005 in accordance with the original learning technique. Saturation of learning occurred during the 4000th iteration of training, at which time the average loss had remained constant for the previous 200 iterations. As a result, we forced the model training to halt due to the lack of a meaningful improvement in the model loss. As previously stated, YOLOv2's design incorporated a Batch Normalization layer at each convolutional block. As the input data is normalized at each convolutional block of the model, this reduces the likelihood of overfitting.

Fig. 3. Samples of annotated frames from UCF-Crime dataset

The model was trained using an NVIDIA Tesla K80 GPU equipped with 12 GB of RAM. We discovered that completing 4000 iterations on the training set took four hours on road accident scenes of UCF-crime dataset [4].

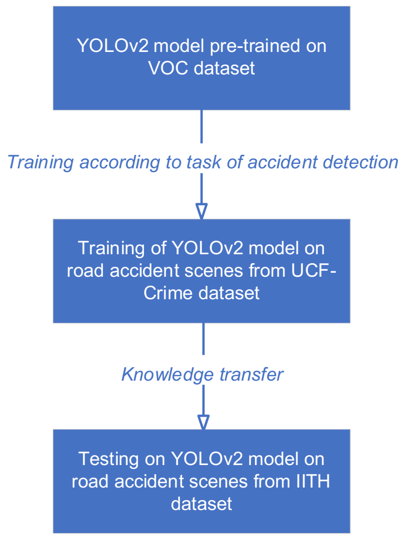

Fig. 4. Notion of transfer learning paradigm applied in the model

TABLE II. HYPTERPARAMETERS FOR MODEL TRAINING

Hyperparameter	Value
Learning rate	0.01
Weight Decay	0.005
Step Epoch	1000
Steps	0.1
Iterations	4000

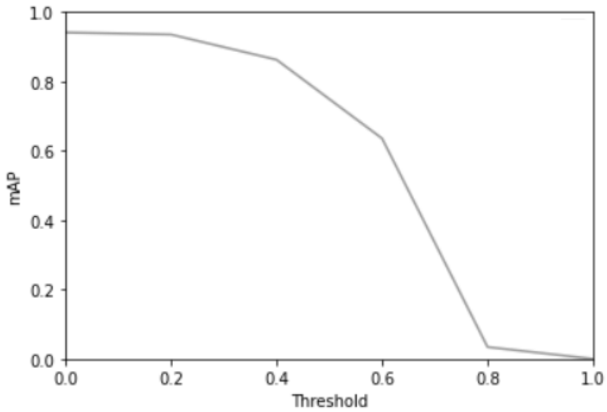

Fig. 5. Plot of mean Average Precision with varying theshold

V. MODEL EVALUATION AND RESULTS DISCUSSION

The results obtained for quantitative (mAP and FPS) and qualitative (performance in terms of predicted bounding boxes) evaluation of the model are discussed in this section.

A. *Quantitative Evaluation*

The performance of the model is assessed using mAP and frames processed per second.

- Estimation of Mean-Average Precision (mAP) is a widely used method for evaluating bounding box predictions. In this metric, we take the average of precision across recall values in the range 0 to 1 through various thresholds and finally take the mean of the average precision for each of the class. Since we had only one class, *accident*, our average precision value became the mean average precision value as well. Each threshold represented the percentage of overlap between predicted bounding box and ground truth bounding box. We tabulated the mAP values for an Intersection over Union (IoU) threshold range ranging from 0.1 to 1.0. We set the IoU threshold of 0.5. Figure 5 shows the mAP curve for the threshold range. We achieved mAP of 76%.
- We evaluated the model on test videos by running it on the GPU which was used for training the model and model processed the frames at 60 frames per seconds for accident detection on IITH road accident dataset. The model took 11 seconds to process 668 accident frames from IITH dataset. This demonstrated the YOLOv2 model's real-time performance on a high-end GPU. For practical purposes, we also evaluated the model on a system equipped with an NVIDIA GTX 4GB GPU, where it predicted outcomes at a rate of ~16.67 frames per second, providing detection at near real-time.

B. *Qualitative Evaluation*

From the qualitative results mentioned in figure 6 in which bounding box predictions by the model detecting accident incidents are given, it is observed that model generalized well on unseen dataset, namely, IITH road accident dataset using transfer learning.

Fig. 6. Bounding box predictions by the model detecting accident incidents

VI. CONCLUSION

Smart and sustainable traffic management and surveillance are one of the major goals in smart city-based projects. Particularly an interest is analyzing and avoiding traffic accidents. In this paper, we implemented a deep-learning based single-shot object detection model to perform accident detection in real-time from traffic surveillance camera. We first worked on creating a video dataset that contained road accident scenes by using corresponding videos from the UCF-Anomaly dataset.

For the extracted frames, we provided bounding box annotations to both training and test videos. Once the dataset was ready, we proceeded to implement the concept of transfer learning. This was done by using a pre-trained YOLOv2 model. We then fine-tuned it using our custom road accident dataset. Our quantitative and qualitative results obtained by fine-tuning this model, were also presented. Our model was able to generalize well for unknown video dataset (IITH road accident dataset). We achieved mAP of 76% by executing the model on IITH road accident dataset. Moreover, our model implementation achieved 60 FPS on a system with an NVIDIA Tesla K80 GPU and about 16.67 FPS on a standard laptop with a 4GB GTX GPU. This shows that our model can be deployed in the real-world for detection of unusual activities such as accidents.

In this paper, we implemented a deep-learning based single-shot object detection model to perform accident detection in real-time from traffic surveillance camera. We first worked on creating a video dataset that contained road accident scenes by using corresponding videos from UCF-Crime dataset.

Our model is currently limited to accident detection from videos. It can be extended to support whole traffic surveillance system including vehicle tracking, road surface monitoring and upgraded to support multimodal data also such video and audio for more robust model. For future work, it would also be interesting to extend the model to support drone images and covert the existing code to hardware specific language to achieve even higher speed for processing the frames. We would also focus on annotating public dataset with varying notions of accident scenes.

References

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014

[2] Road traffic injuries infographic by WHO, https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

[3] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[4] UCF-Crime, https://www.crcv.ucf.edu/research/real-world-anomaly-detection-in-surveillance-videos/

[5] IITH accident dataset, https://sites.google.com/site/dineshsinghindian/iith_accident-dataset

[6] H. Jiang, Y. Wang, and Y. Yang, "Fast Traffic Accident Identification Method Based on SSD Model," in *Proceedings of the 2019 International Conference on Robotics Systems and Vehicle Technology*, 2019, pp. 177–181.

[7] X. Huang, P. He, A. Rangarajan, and S. Ranka, "Intelligent intersection: two-stream convolutional networks for real-time near-accident detection in traffic video," *ACM Trans. Spat. Algorithms Syst.*, vol. 6, no. 2, pp. 1–28, 2020.

[8] J. G. Choi, C. W. Kong, G. Kim, and S. Lim, "Car crash detection using ensemble deep learning and multimodal data from dashboard cameras," *Expert Syst. Appl.*, p. 115400, 2021.

[9] V. C. M. Vishnu, M. Rajalakshmi, and R. Nedunchezhian, "Intelligent traffic video surveillance and accident detection system with dynamic traffic signal control," *Cluster Comput.*, vol. 21, no. 1, pp. 135–147, 2018.

[10] E. P. Ijjina, D. Chand, S. Gupta, and K. Goutham, "Computer vision-based accident detection in traffic surveillance," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2019, pp. 1–6.

[11] H. Ren, Y. Song, J. Wang, Y. Hu, and J. Lei, "A deep learning approach to the citywide traffic accident risk prediction," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 3346–3351.

[12] D. Tian, C. Zhang, X. Duan, and X. Wang, "An automatic car accident detection method based on cooperative vehicle infrastructure systems," *IEEE Access*, vol. 7, pp. 127453–127463, 2019.

[13] K. K. Santhosh, D. P. Dogra, and P. P. Roy, "Anomaly detection in road traffic using visual surveillance: A survey," *ACM Comput. Surv.*, vol. 53, no. 6, pp. 1–26, 2020

[14] B. Maaloul, A. Taleb-Ahmed, S. Niar, N. Harb, and C. Valderrama, "Adaptive video-based algorithm for accident detection on highways," in *2017 12th IEEE International Symposium on Industrial Embedded Systems (SIES)*, 2017, pp. 1–6.

[15] W. Sultani and J. Y. Choi, "Abnormal traffic detection using intelligent driver model," in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 324–327.

[16] K. Yun, H. Jeong, K. M. Yi, S. W. Kim, and J. Y. Choi, "Motion interaction field for accident detection in traffic surveillance video," in *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 3062–3067.

[17] A. R. Pathak, M. Pandey, and S. Rautaray, "Application of Deep Learning for Object Detection," *Procedia Comput. Sci.*, vol. 132, pp. 1706–1717, 2018, doi: https://doi.org/10.1016/j.procs.2018.05.144.

[18] A. R. Pathak, M. Pandey, S. Rautaray, and K. Pawar, "Assessment of object detection using deep convolutional neural networks," Springer, vol. 673. 2018.

[19] A. R. Pathak, M. Pandey, and S. Rautaray, "Deep Learning Approaches for Detecting Objects from Images: A Review," in *Progress in Computing, Analytics and Networking*, 2018, pp. 491–499.

[20] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010, doi: 10.1109/TKDE.2009.191.

[21] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv Prepr. arXiv1804.02767*, 2018.

[22] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv Prepr. arXiv2004.10934*, 2020

[23] Tzutalin. Labelimg. Free Software: MIT License, 2015.

[24] F. Zhuang *et al.*, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[25] S. Robles-Serrano, G. Sanchez-Torres, and J. Branch-Bedoya, "Automatic Detection of Traffic Accidents from Video Using Deep Learning Techniques," *Computers*, vol. 10, no. 11, p. 148, 2021.

[26] M. Ozbayoglu, G. Kucukayan, and E. Dogdu, "A real-time autonomous highway accident detection model based on big data processing and computational intelligence," in *2016 IEEE International Conference on Big Data (Big Data)*, 2016, pp. 1807–1813.

[27] HawkAI:Monitoring Road Accidents, https://www.ischool.berkeley.edu/projects/2019/hawkai-monitoring-road-accidents-real-time-using-deep-learning

[28] J. Hagenauer and M. Helbich, "A comparative study of machine learning classifiers for modeling travel mode choice," *Expert Syst. Appl.*, vol. 78, pp. 273–282, 2017.

[29] N. Dogru and A. Subasi, "Traffic accident detection by using machine learning methods," in *Third International Symposium on Sustainable Development (ISSD'12)*, 2012, p. 467.

[30] S. Seong, J. Song, D. Yoon, J. Kim, and J. Choi, "Determination of vehicle trajectory through optimization of vehicle bounding boxes using a convolutional neural network," *Sensors*, vol. 19, no. 19, p. 4263, 2019.

[31] Y.-L. Chung and C.-K. Lin, "Application of a model that combines the YOLOv3 object detection algorithm and canny edge detection algorithm to detect highway accidents," *Symmetry (Basel).*, vol. 12, no. 11, p. 1875, 2020.