

Accident vehicle types classification: a comparative study between different deep learning models

Mardin Abdullah Anwer, Shareef M. Shareef, Abbas M. Ali

Software and Informatics Department, College of Engineering, Salahaddin University-Erbil, Iraq

Article Info

Article history:

Received Sep 24, 2020

Revised Nov 27, 2020

Accepted Dec 16, 2020

Keywords:

Accident recognition
Deep learning
GAI and KAI datasets
Transfer learning
Vehicle accidents image classification

ABSTRACT

Classifying and finding type of individual vehicles within an accident image are considered difficult problems. This research concentrates on accurately classifying and recognizing vehicle accidents in question. The aim to provide a comparative analysis of vehicle accidents. A number of network topologies are tested to arrive at convincing results and a variety of matrices are used in the evaluation process to identify the best networks. The best two networks are used with faster recurrent convolution neural network (Faster RCNN) and you only look once (YOLO) to determine which network will identifiably detect the location and type of the vehicle. In addition, two datasets are used in this research. In consequence, experiment results show that MobileNetV2 and ResNet50 have accomplished higher accuracy compared to the rest of the models, with 89.11% and 88.45% for the GAI dataset as well as 88.72% and 89.69% for KAI dataset, respectively. The findings reveal that the ResNet50 base network for YOLO achieved higher accuracy than MobileNetV2 for YOLO, ResNet50 for Faster RCNN with 83%, 81%, and 79% for GAI dataset and 79%, 78% and 74% for KAI dataset.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Mardin Abdullah Anwer
Software and Informatics department
Salahaddin University- Erbil, Iraq
Email: mardin.anwer@su.edu.krd

1. INTRODUCTION

Intelligent transportation system has become one of the civilizational developments that the world is witnessing. Monitoring and analyzing the road has become quite necessary to control some areas instead of human beings-this breakthrough has brought into vision what is known as the smart city. Auto accident recognition and reporting system is a crucial topic; however, it has some problems such as multiple object tracking, object detection, and video surveillance in addition to some other issues related to the real-time monitoring system.

The main objective of this research is to find the best deep learning network for describing vehicle accident images. Since use of roads by vehicles is on the increase, numerous studies on accident detection and classification are conducted utilizing signals such as acoustic where cross-correlation processing of radiated tire noise is used [1]. Radar [2] and ultrasonic signals [3] have been used to solve the problem of computational complexity in vehicle detection. Infrared thermal images taken from infrared thermal cameras are used to analyse road traffic flow surveillance in various circumstances, including conditions of poor visibility [4]. Many approaches using convolutional neural networks (CNNs) based on deep learning have been applied for vehicle classification and detection [5, 6]. At the time of the accident, vehicle data such as tag, picture, time, area, volume, and type are reported in written form by a police officer while in developed countries such information is passed on using intelligent transportation systems [7]. Research in this area has

used machine learning technology for image classification. Five pre-trained convolutional neural networks have been worked on to classify vehicle accident images and compare the results achieved. There is accomplished research that has used machine vision for detecting the damaged vehicles. This research examines some CNN detectors to see which one best classifies images of the accident.

The key contributions of this research are: 1) Identifying the best pre-trained CNN from a deep learning perspective for classifying the categories of the accident images. 2) Identifying the best network for detecting the vehicle location and its type whether it is a car, a motorcycle, a bus, or a truck. 3) Creating two datasets for vehicle accident images where the images are categorized according to the accident type. The categories can be car-car, car-bus, car-motorcycle, bus-motorcycle, and bus-truck accidents. This paper consists of five sections. Section one is literature review and section two covers the classification of accidents and vehicle detection CNNs. Section three sets up the experiment, whereas section four tackles the evaluation of the matrices and discusses the results. The last section is the conclusion which sums up the main findings of the research and includes recommendations for further research. It is common for vehicle accident detection and recognition system to analyse and retrieve accident images incorrectly from the accident image investigation. The problem might increase if the preprocess stages such as conversion, scaling, and rotation are applied to the images.

Many algorithms are used to process and extract the significant features of visual vehicle accident or filter the background for easier analysis. This process is beneficial for images with low quality or for transfer learning. The preprocessing relies on the accident images if it is colored or grey as each process requires different methods and algorithms. Methods mainly are histogram equalization [8], wavelet transform [9], and homomorphic filtering [10] to enhance low-quality images. The efficiency of these methods is limited under certain conditions since it requires a large number of parameters and thresholds to be adjusted. Accident images classification using deep learning has become a debatable issue. The research in [11] offers a novel deep learning-based vehicle area that figures and calculates with 2D deep convolution organizes. The 2D-DBN design utilizes second-request planes rather than first-request vector as input and utilization bilinear projection for holding dis-criminative data to decide the size of the deep architecture which improves the achievement rate of vehicle identification. He proposes a different productive vehicle identification and classification approach which depends on the convolutional neural network. The features extracted by using this approach beat those created by customary methodologies [12]. Yi applies a pre-trained AlexNet model for checking vehicles in an image patch in wide area motion imagery (WAMI) investigation [13]. While Kumeda proposes a deep convolutional neural network (CNN) which consists of four convolutional layers and three fully connected layers [14].

They conclude that their CNN model was effective for image classification and performed the task with an accuracy of 94.4% on four target accident classes (Traffic-Net dataset). However, they did not apply their CNN to different datasets or compare their results with another research. In [15], a single 2D end-to-end fully convolutional network is introduced to predict the vehicle and the bounding boxes simultaneously. Object Detection includes images or video frames as inputs, while outputs are lists where each element represents the location and category of candidate object information [16]. Generally, object detection aims to extract discriminatory features to help distinguish the classes. The process of object recognition using a convolutional neural network is accomplished in two phases: feature extraction and training [17]. In the first phase, the common features are extracted using a base network which can be any pre-trained CNN. The second phase is achieved by using models such as support vector machines [18], Neural Networks [19], AdaBoost [20], histogram of flow gradient (HFG), hidden markov model (HMM) and gaussian mixture model (GMM). Nevertheless, they did not compare their results with any deep learning models. Nowadays, convolutional neural networks (CNNs) are used to extract features from area proposals or end-to-end object detection without specifying the unique features of a certain class. The well-performed deep learning-based approaches of object detection include region proposals (R-CNN) [21], Fast R-CNN [22], Faster R-CNN [23], single shot multi-box detector (SSD) [24], and you only look once (YOLO) [25]. Mudumbi combines Faster R-CNN and MobileNet to detect different kinds of logos. They improve the regional proposal network (RPN) by selecting appropriate anchors which have achieved 92.4% detection accuracy. Nonetheless, the authors did not test a suitable resolution for generating proposals that may improve the performance of the RPN considerably [26].

2. METHODOLOGY

2.1. CNN

Pre-trained deep CNN image-based classifications were trained on a large number of images to be able to classify other images within a big range of objects [27]. Therefore, the features of many images were learned by pre-trained CNN. The networks used in this research are GoogleNet, which contains 22 layers,

ResNet50, containing 50 layers, MobileNetV2, containing 53 layers, AlexNet, containing 25 layers, and SqueesNet with 18 layers. To classify the kind of vehicle in the images of the accident, these CNN models have been fine-tuned using transfer learning. Thus, the five used CNNs have been retrained with images from the SCGAI and SCKAI datasets. A deep learning toolbox in MATLAB was used to develop and train the networks. For both datasets, 66% of images for training and 34% for testing have been employed. As the dimension of an input image in each pre-trained CNN is different, analyze network function has been used to examine the network architecture and change the input layer dimension to match the architecture of the pre-trained CNN. For example, SqueesNet and AlexNet require 227-by-227 image size, while MobileNetV2 and ResNet50 input layer is 224-by-224. Augmented image datastore has been used to achieve the process of automatically resizing the images and avoid overfitting of the network. The preprocessing step is shown in the workflow of CNN image-based systems in Figure 1.

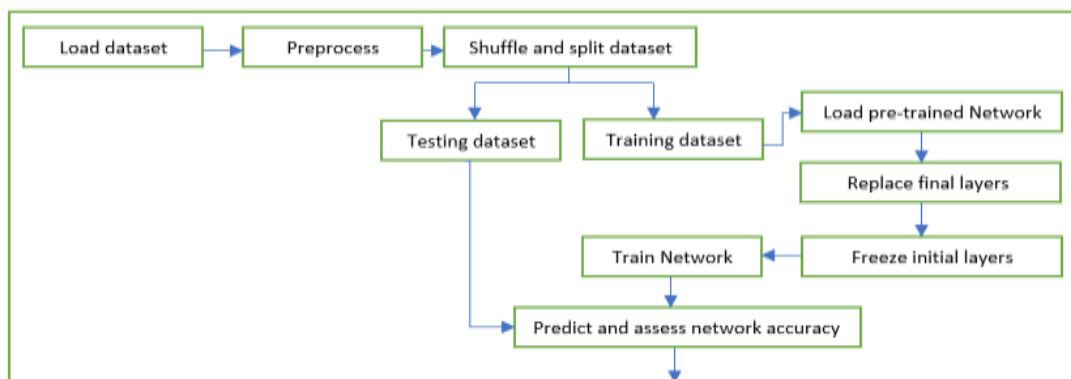


Figure 1. Workflow of CNN-Image based systems

2.2. Accident classification

The last classification and learnable layers use the extracted features resulted from convolutional layers for classification. These layers embrace the details on how to combine the features into loss value, class probabilities and predicted label [28]. To retrain the pre-trained deep CNN, these layers were replaced to be able to classify the new images. Another important function that was helpful in this stage was **findLayersToReplace** function. This function finds the last layer that holds learnable weights and replaces it with the new two output layers which are fully connected. No changes have been made to the parameters of the rest of the layers as the main goal of this research is to compare the accuracy of the networks. Therefore, the layers were frozen. Before training the network using the **trainnetwork** function, MATLAB gave us the freedom of changing the training option using training options function. The main parameters which need to be changed are **InitialLearnRate**, **MiniBatchSize**, and **MaxEpochs**. In transfer learning, **InitialLearnRate** prefers to be of very low value to decrease the training time in non-frozen layers. Besides, there is no need for performing the learning process for many epochs. Setting the mini-batch size means updating the weights and evaluating the gradient of the loss function.

One of the main contributions of this research is evaluating the performance of the networks and comparing the accuracy results. The value of accuracy is between 0 and 1. The two most efficient classifier networks will be used as a base network for the second stage that is detecting the type of the vehicle in the accident image.

2.3. Vehicle detection and recognition

YOLO V2 [29] and Faster Recurrent Convolutional Neural Network [26] models have been used for detecting multi-class vehicles in the images. Vehicles in accident image datasets are truck, bus, motorcycle, and cars. The network architecture of YOLO consists of 24 convolutional layers, the last one outputs a tensor with shape (7,7,1024), which is flattened, and with two fully connected layers. The number of fully connected layers is changed to four because we have four classes in the GAI dataset and three in the KAI dataset. The final loss adds localization, confidence, and classification losses together. The function of the two fully connected layers is to perform a linear regression to create boundary box predictions and a final prediction using a threshold of box confidence scores. YOLO is an end-to-end regression problem and uses a single convolutional network to predict the boundary boxes and the corresponding class probabilities. By

splitting the image into the $S \times S$ grid and taking m of boundary boxes from each grid, the convolutional neural network outputs a class probability and offsets values for the boundary box. It then selects boundary boxes that have the class probability above a threshold value and uses them to locate the object within the image [16]. However, YOLO does not perform well if the objects involved in the image are very small. Figure 2 illustrates the architecture of the YOLO detection system.

Instead of using boxes approach, Faster R-CNN detector adds a region proposal network (RPN) to generate regions of interest directly from the network. The RPN works by moving a sliding window over the CNN feature map and at each window generating k potential boundary boxes and scores associated with how good each of those boxes is expected to be. This k represents the common aspect ratios that candidates to objects could fit caller anchor boxes. For each anchor box, the RPN outputs a boundary box and score per position in the image. Consequently, the speed and detection process are increased by using this method. Figure 3 demonstrates the implementation of the Faster RCNN detection algorithm.

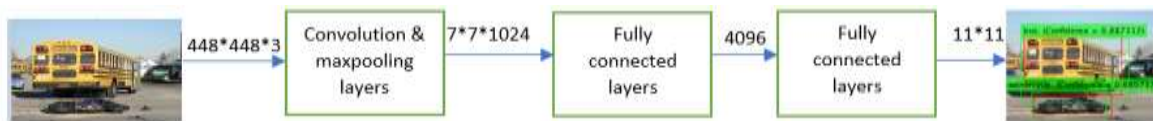


Figure 2. YOLOV2 detection system, the input is an accident image of GAI dataset and the output the same image with class probability and offset values for the bounding box

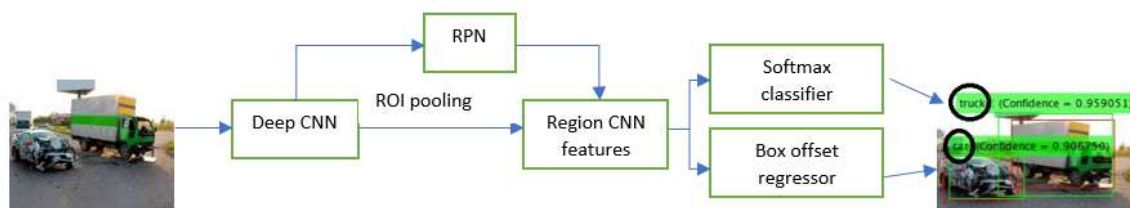


Figure 3. Faster-RCNN object detection system

As it can be seen from the architecture, the region proposal network and the classifier use convolutional layers that are trained jointly. The region proposal network behaves as an attention director, determining the optimal boundary boxes across a wide range of scales and using nine candidate aspect ratios to be evaluated for object classification. In other words, the RPN tells the unified network where to look [30]. The detector identified two vehicles in the accident image, a truck, and a car with a score confident 0.95, 0.90 respectively.

3. EXPERIMENTAL SETUP

The experiments were conducted on a computer with specification Intel(R) Core (TM) i7-8700T CPU @ 2.40 GHz (2400Mhz), 6 Cores, 8 GB RAM, using MATLAB 2019a.

3.1. Dataset

In this research, vehicle accident images have been collected from different resources since there are no public datasets for vehicle accidents that would align with our research methodology. The images have generated two datasets. The first dataset is named general accident images (GAI) where images are taken from DCD [31], accident dataset [32], and online google images. The second dataset is kurdish accident images (KAI) where images are taken directly from the ministry of interior of the Iraqi kurdistan regional government (KRG). The images are of accidents that occurred from 2015-2020 in different locations and on roads such as highways and intersections. The raw dataset contains 2000 images as well as 100 videos of the accidents. We screenshotted specific parts of the videos to be used as images of the accident. After creating the datasets, the procedure of preprocessing and categorizing the images according to the accidents and vehicles in the images begins. In the GAI dataset, six categories are formed, which are: cars, car-bus, car-motorcycle, bus-motorcycle, bus-truck, and car-truck. To maintain the balance of the image numbers in each category, 118 images in each category are selected to have a total of 708 images. Most of the accidents in the KAI dataset involve a single car with a few ones that occurred as bus-bus, bus-truck, bus-car, and bus-

motorcycle. This could be due to the transportation system in place in Kurdistan region. As an example, bus transportation between the cities does not exist in Kurdistan and buses operate locally only during the daytime. As a result, only three categories in KAI are created, which are cars, car-motorcycle, and car-truck. There are 130 images in each category to become 390 images. Image augmentation has been used to augment the datasets. Regarding the number of vehicles in each image, images of accidents between vehicles are selected. Although the number of vehicles is not balanced in both datasets, this will not affect the purpose of the research. Detecting the type of vehicle will enrich the classification process in terms of description. Table 1 illustrates the number of vehicles in each dataset. All the images are preprocessed, filtered, and scaled from their original sizes to 224×224 pixels. Figure 4 illustrates some images and their category.



Figure 4. Images from GAI and KAI after preprocessing: (a) bus-truck, (b) cars accidents, (c) car-bus, (d) car-motorcycle, (e) bus-motorcycle, (f) car-truck accident images

Table 1. Number of vehicles in each dataset

Dataset	Number of cars	Number of bus	Number of motorcycles	Number of trucks
GAI	590	354	236	236
KAI	520	-	130	130

3.2. Classification results

After processing the images and adding them to the categories they each belong to, deep learning was utilized for the classification stage. In computer vision, adopting deep and pre-training CNN in classification proves the ability to accomplish high precision in low-quality images and real-time applications. To assess the classification process of vehicle accidents, five pre-trained Convolutional Neural Network models are used. The models are: AlexNet, GoogleNet, ResNetv2, MobileNetV2 and SqueeNet. Each of the listed models has different architecture and parameters. Selection of the models depends on the most used ones in the image-based vehicle classification and detection research. A remarkable accuracy of 89.69%, 88.45% 88.72%, and 89.11% has been achieved while applying ResNet50 and MobileNetV2, respectively for GAI and KAI datasets. And this has resulted from the architecture and number of layers discussed in section 2. Figure 5 is the comparison of the training process of the listed pre-trained CNNs on the datasets. Figure 6 demonstrates examples of the prediction results for the classification of the datasets. Table 1 illustrate the number of vehicles in each dataset and Table 2 illustrates The number of images used for experimental results.

Table 2. The number of images used for experimental results (CC: cars accidents, CB: car-bus, CM: car-motorcycle, BM: bus-motorcycle, BT: bus-truck, CT: car-truck) accident images

Dataset	Training dataset						Total	Testing dataset						Total
	CC	CB	CM	BM	BT	CT		CC	CB	CM	BM	BT	CT	
GAI	78	78	78	78	78	78	468	40	40	40	40	40	40	240
KAI	97	-	80	-	-	80	257	53	-	40	-	-	40	133

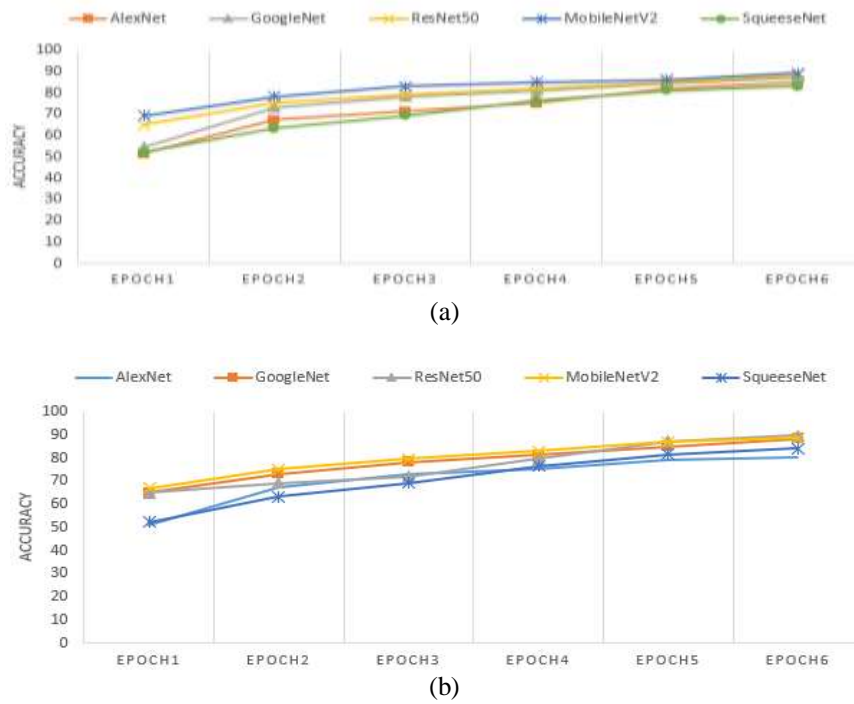


Figure 5. Comparison of the accuracy of training process between different pre-trained CNN using (a) GAI dataset (b) KAI dataset

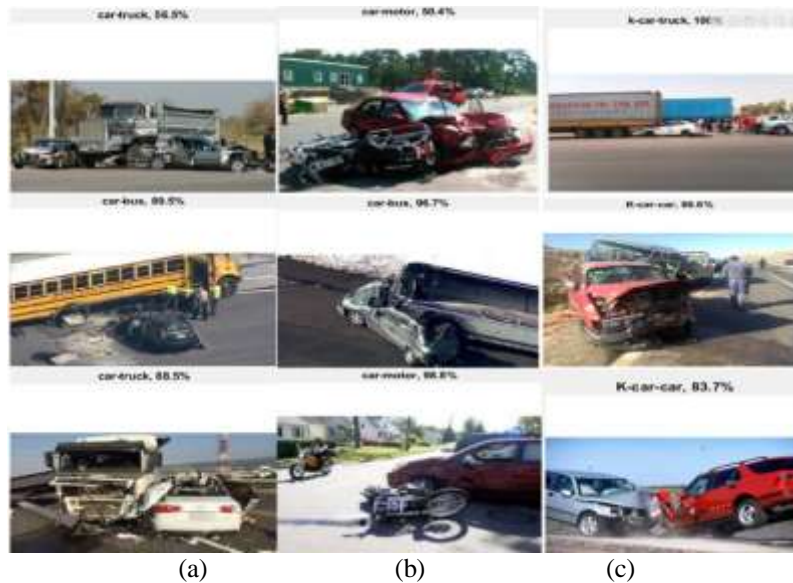


Figure 6. Prediction results for classification using: (a) GoogleNet (b) MobileNetV2 (c) GoogleNet predefined CNN on GAI and KAI datasets

3.3. Vehicle detection results

This stage locates the site of the accident and accurately recognizes the type of vehicle in the accident images. The results from the deep-learning-based classification stage show that ResNet50 and MobileNetV2 have higher accuracy than the rest of the models. Therefore, ResNet50 and MobileNetV2 as base networks for both Faster RCNN and YOLOV2 object detection networks are used. YOLO and RCNN depend on the object annotation technique. Annotating object boundary boxes is necessary for the detection process. In this research, a total of 2,196 boundary boxes are annotated using the image labeler app in MATLAB for GAI and KAI datasets.

- a) Vehicle detection experiment with YOLO: YOLO is composed of two subnetworks: feature extraction network followed by detection network. The feature extraction network is typically a convolutional neural network, which is ResNet50 and MobileNetV2 from section 3. Conversely, detection subnetwork is a small CNN if compared to the feature extraction network and consists of a few convolutional layers and layers specific for YOLO. The subnetwork makes the grid of the image and does the prediction of the detected object class if it exists and that is by separating boundary boxes and associating class probabilities. It utilizes a custom network based on the GoogLeNet architecture, using 8.52 billion activities for a forward pass [33]. The advantage of this prediction is that it can be implemented independently since the first detection. However, the disadvantage is the difficulty of detecting the smallest objects as well as objects that are overlapping. Figure 7 displays the detection and prediction results using the YOLOV2 network.



Figure 7. YOLOV2 detection and prediction score using (a) ResNet50, 2 vehicles available, 2 correctly detected, 2 true positive (b) MobileNetV2, 2 vehicles available, 1 correctly detected and 1 false negative

- b) Vehicle Detection Experiment with RCNN: The Faster R-CNN is the first detector tested for the detection of apples as well as obtaining the highest precession of PASCAL VOC 2007 and 2012. In the COCO challenges 2015 [23] and ImageNet detection and localization at ILSVRC 2016 [34], faster R-CNN was the basis of more than 125 proposed entries and was the foundation of the winners in several categories by sharing the fully convolutional layers in the region proposal network and the object classifier [35]. These layers are trained jointly. The region proposal network (RPN) behaves as an attention director, determining the optimal boundary boxes across a wide range of scales and using nine candidate aspect ratios to be evaluated for object classification [30]. In other words, the RPN employs anchor boxes for object detection. Producing region proposals in the network is considered faster and better tuned to the data. Figure 8 shows the detection and prediction score using Faster RCNN.



Figure 8. Vehicle detection using (a) Faster RCNN resnet50 on GAI test image dataset, 2 vehicles available, 1 correctly detected, 1 true positive and 1 false negative (b) Faster RCNN resnet50 on KAI test image dataset, 2 vehicles available, 1 correctly detected, 1 true negative

4. DISCUSSION

This research is about accident images classification and vehicle type detection processes. The images used in this experimental research are of modest quality and possess good resolution. This is consistent with the aim of the research which is to compare the classification and performance detection of the selected networks. In order to evaluate the performance of the detectors and compare their results, performance matrices, which are listed in [36, 37], have been used. However, these matrices have been modified because of using multiclass instead of binary class. Common benchmarks used are boundary box detection, mAP, Log Average Miss Rate detection, F1-measure and accuracy.

a) Intersection over Union (IoU): The process of detecting uses the image as an input to the network and output a detected BB (BBdt) and a ground truth BB (BBgt) form a potential match if they overlap sufficiently by performing multiscale detection and any necessary non-maximal suppression (NMS) for merging nearby detections. The IoU score ranges from 0 to 1 and the closer the two boxes, the higher the IoU score [37]. Figure 9 shows the estimated anchor box for the class bus.

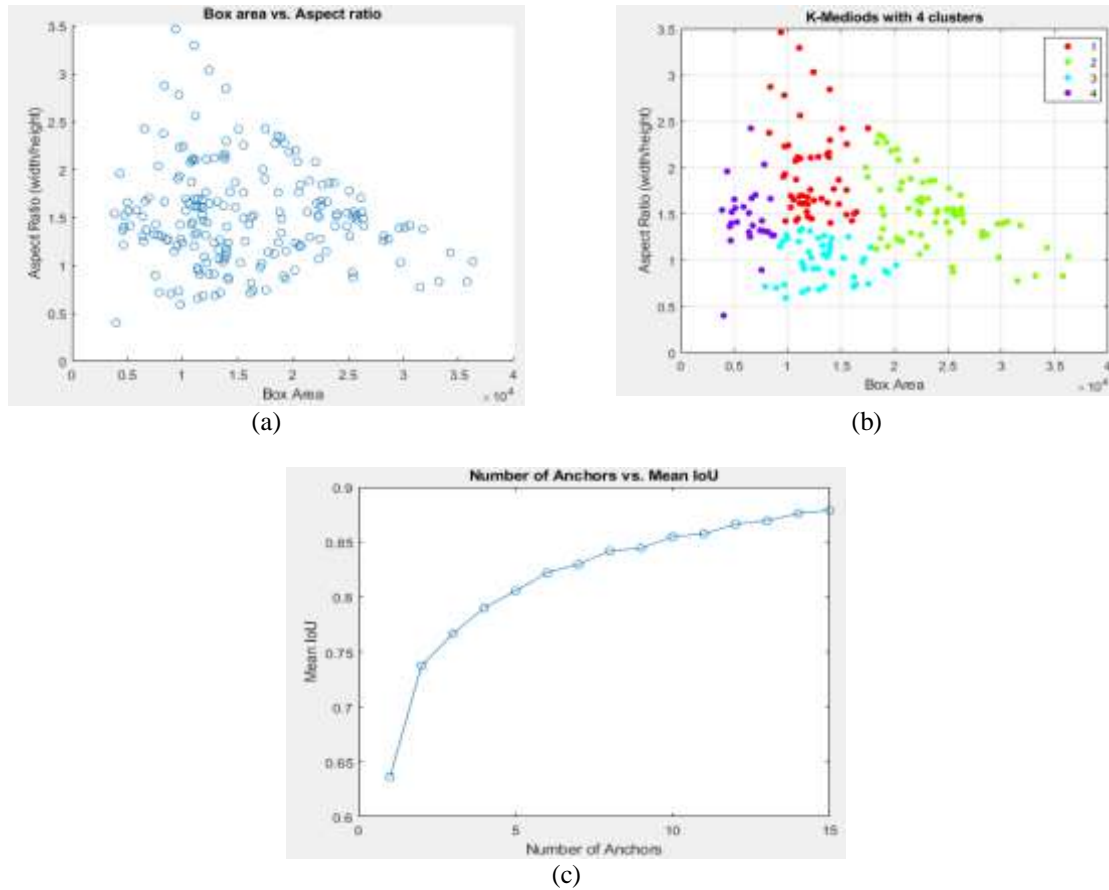


Figure 9. (a) box area verses aspect ratio (b) K-mididos with 4 clusters (c) number of anchors verses Mean IoU for the vehicle bus

$$IoU_{dt}^{gt} = \frac{area(BB_{dt} \cap BB_{gt})}{area(BB_{dt} \cup BB_{gt})} \tag{1}$$

To compare the detection results of the models we applied them on the same accident image as shown in Figure 10 that is clearly shown that Yolo ResNet50 BBgt and BBdt are closer than other models. The red box represents the BBgt and the green box represents the BBdt with the confidence score. Figures 10 and 11 shows the BBgt and BBdt on car and truck accident after applying YOLOV2 MobileNetV2.

b) mAP: to evaluate the performance of object detectors, AP utilized metric as it is a single number metric that typifies both precision and recall as well as summarizes the Precision-Recall curve by averaging precision over recall values from 0 to 1. Mean average precision (mAP) is the averages of AP over the N classes. It measures how well the model generates a bounding box [36].

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \tag{2}$$

As shown Figure 10, the precision of a given class in classification, known as the positive predicted value, is given as the ratio of true positive (TP) and the total number of predicted positives (true positive + false positive) [37].

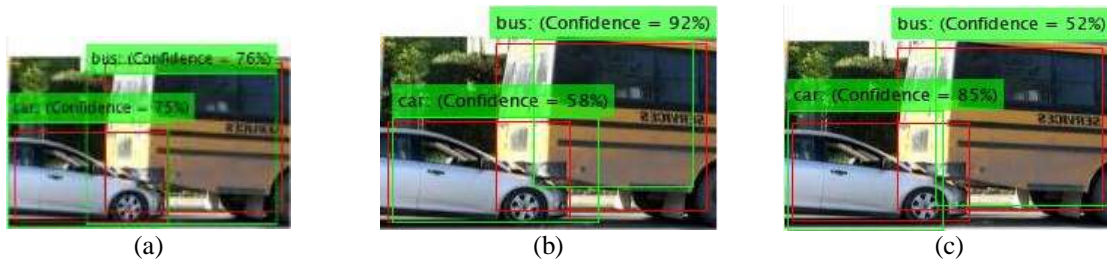


Figure 10. BBgt and BBdt on a random image of GAI using (a)Yolo ResNet50, (b)Faster RCNN Res-Net50, (c) YOLOV2 MobileNetV2



Figure 11. (a) Car and truck accident image in KAI dataset (b) the BBgt of the vehicles (c) BBdt detection and prediction score using YOLOV2 MobileNetV2 network

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3}$$

Similarly, the recall, also called a true positive rate or sensitivity, of a given class in classification, is defined as the ratio of TP and total of ground truth positives [36].

$$\text{Recall (TPR)} = \frac{TP}{TP+FN} \tag{4}$$

- c) Log Average Miss Rate: it uses to summarize detector performance, computed by averaging miss rate at nine FPPI rates evenly spaced in log-space in the range 10^2 to 100 because there is a limitation on (FPPI) [37]. Table 3 illustrates the results of average precision and log average miss rate for all the classes in the datasets.

$$\text{FPPI} = \text{FP/number of tested images} \tag{5}$$

Table 3. Experimental result of average precision and log average miss rate for each vehicle in the datasets

Dataset	CNN	Backbone CNN	Average precision				Log Average Miss Rate			
			Bus	Motorcycle	car	Turk	Bus	Motorcycle	car	Turk
GAI	YOLOV2	ResNet50	0.99	0.93	0.98	0.99	0.01	0.07	0.02	0.01
	YOLOV2	MobileNetV2	0.93	0.61	0.89	0.84	0.07	0.33	0.13	0.16
	Faster RCNN	ResNet50	0.71	0.58	0.65	0.84	0.30	0.39	0.42	0.17
KAI	YOLOV2	ResNet50	-	0.90	0.95	0.93	-	0.10	0.05	0.07
	YOLOV2	MobileNetV2	-	0.63	0.87	0.73	-	0.37	0.13	0.27
	Faster RCNN	ResNet50	-	0.59	0.67	0.78	-	0.41	0.33	0.22

- d) Accuracy: is the percentage of correctly predicted examples out of all predictions [32]. Table 4 shows the number of detecting each vehicle in the test datasets. While Table 5 shows the accuracy results on the testing image dataset.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \tag{6}$$

Table 4. presents the occurrence of the vehicle in test images. NCDC: number of correct detection-Car, NCDB: number of correct detection-Bus, NCDM: number of correct detection-Motorcycle, NCDT: number of correct detection-Truck

Vehicle accident dataset	YOLOV2/ ResNet50				YOLOV2/MobileNetV2				Faster RCNN/ResNet50			
	NCDC	NCDB	NCDM	NCDT	NCDC	NCDB	NCDM	NCDT	NCDC	NCDB	NCDM	NCDT
GAI	84/93	42/45	37/52	37/50	78/93	40/45	33/52	45/50	73/93	40/45	37/52	34/50
KAI	69/78	-	18/30	19/25	65/78	-	20/30	20/25	60/78	-	19/30	20/25

Table 5. Accuracy of Faster RCNN and YOLO on test images datasets

Datasets	CNN Model	Backbone CNN	Total number of vehicles in test images	Number of vehicles detected	Accuracy
GAI	YOLOV2	ResNet50	240	200	0.83
	YOLOV2	MobileNetV2	240	196	0.81
	Faster RCNN	ResNet50	240	184	0.76
KAI	YOLOV2	ResNet50	133	106	0.79
	YOLOV2	MobileNetV2	133	105	0.78
	Faster RCNN	ResNet50	133	99	0.74

5. CONCLUSION AND FUTURE WORK

This research has made it certain that convolutional neural networks (CNNs) can achieve accurate classification and detection results. As there are different kinds of CNN used for recognizing different images, it has chosen several CNN to recognize images of vehicle accidents. To accomplish this goal, KAI and GAI datasets were proposed. Following analysis of the result, these datasets can now be recommended to fill in the gap in the intelligent transportation systems, especially when it comes to images of vehicle accidents. To classify the images, five deep learning models such as; GoogleNet, ResNet50, MobileNetV2, AlexNet, and SqueesNet have compared. The selected networks have provided different classification rates (MobileNetV2 and ResNet50 achieved 4% more than the rest networks). Many pre-trained network topologies are used for image classification, but for space concerns, this research has chosen only five. The research findings revealed that using detector networks with deep CNN topologies increases the accuracy of accident classification by determining the location and vehicle type in the accident image. Although training time of YOLOV2 was much less than Faster RCNN, the experiment results also show that pre-trained Resenet50, which is used as an extractor of features with YOLOv2 model, achieves much better in detecting the type of vehicles (more than 6%) in both datasets, according to the accuracy equation. Nonetheless, these results fall short of expectations as the research aims to discover a network or method that can describe the accident images precisely and detect the vehicle type with full accuracy. Therefore, at present time, deep semantic segmentation is under research to help with the precise classification of object types or classes and the checking of results. Hence our future research will focus on this issue.

REFERENCES

- [1] J. F. Forren, D. Jaarsma, "Traffic monitoring by tire noise", in : *Proceeding of Conference on Intelligent Transportation Systems*, Boston, MA, USA, 1997, pp. 177-182, doi: 10.1109/ITSC.1997.660471.
- [2] Duzdar, G. Kompa, "Applications using a low-cost baseband pulsed microwave radar sensor," *IMTC 2001, in: Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics* (Cat. No.01CH 37188), Budapest, vol. 1, pp. 239-243, 2001. Doi: 10.1109/IMTC.2001.928819.
- [3] Y. Jo, I. Jung, "Analysis of vehicle detection with WSN-based ultra-sonic sensors", *Sensors*. vol. 14, no. 8, pp. 14050-14069, 2014.
- [4] Y. Iwasaki, M. Misumi, and T. Nakamiya, "Robust vehicle detection under various environments to realize road traffic flow surveillance using an infrared thermal camera", *The Scientific World Journal*, 947272, 2015.
- [5] Y. Hans, T. Jiang, Y. Ma and C. Xu, "Pretraining convolutional neural networks for image-based vehicle classification". *Advances in Multimedia*, 2018.
- [6] H. Leung, X. Chen, C. Yu, H. Liang, J. Wu, Y. Chen, "A Deep-Learning-Based Vehicle Detection Approach for Insufficient and Nighttime Illumination Conditions", *Applied Sciences*, vol. 9, no. 22, p. 4769, 2019.
- [7] M. Anwer, S. Shareef, A. Ali, "Smart traffic incident reporting system in e-government," in: *Proceedings of the of the ECIAIR 2019 European Conference on the Impact of Artificial Intelligence and Robotic*, 2019. DOI: 10.34190/ECIAIR.19.061.
- [8] Y. Wang, Z. Pan, "Image contrast enhancement using adjacent-blocks-based modification for local histogram equalization," *Infrared Physics and Technology*, vol. 86, pp. 59-65, 2017.
- [9] W. He, Q. Wu, and S. Li, "Medical X-Ray image enhancement based on wavelet domain homo-morphic filtering and CLAHE," in: *Proceeding of the 2016 International Conference on Robots & Intelligent System (ICRIS)*, IEEE, Zhangjiajie, China, pp. 249-254, 2016.

- [10] B. Shalchian, H. Rajabi, and H. Soltanian-Zadeh, "Assessment of the wavelet transform in re-duction of noise from simulated PET images," *Journal of Nuclear Medicine Technology*, vol. 37, no. 4, pp. 223-228, 2009.
- [11] H. Wang, Y. Cai, and L. Chen, 2014. "A vehicle detection algorithm based on deep belief net-work," *The Scientific World Journal*, p. 647380.
- [12] D. He, C. Lang, S. Feng, X. Du, and C. Zhang, "Vehicle detection and classification based on convolutional neural network," In: *Proceedings of the of the 7th International Conference on Internet Mul-timedia Computing and Service*, pp. 1-5, 2015.
- [13] M. Yi, F. Yang, E. Blashch, "Vehicle Classification in WAMI Imagery using Deep Network," In: *Proceedings of the of the SPIE 9838: Sensors and Systems for Space Applications IX*, 2016.
- [14] B. kumeda, Z. fengli, A. oluwasanmi, F. owusu, M. assefa and T. amenu, "Vehicle Accident and Traffic Classification Using Deep Convolutional Neural Networks," in : *Proceeding 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing*, Chengdu, China, pp. 323-328, 2019 doi: 10.1109/ICCWAMTIP47768.2019.9067530.
- [15] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3D lidar using fully convolutional network," in: *Proceedings of the of the Robotics: Science and Systems*, 2016.
- [16] X. Huang, P. He, A. Rangarajan and S. Ranka. "Intelligent Intersection: Two-stream Convolu-tional Networks for Real-time Near-accident Detection in Traffic Video," *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, vol. 6, no. 2, pp. 1-28, 2020.
- [17] X. Wen, L. Shao, W. Fang, and Y. Xue, "Efficient feature selection and classification for ve-hicle detection, | IEEE Trans," *Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 508-517, 2015.
- [18] Y. Tang. "Deep learning using linear support vector machines," In *Workshop on Representation Learning, ICML*, Atlanta, USA, 2013.
- [19] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural net-works- A review, Pattern recognition," vol. 35, no. 10, pp. 2279-2301, 2002.
- [20] P.A. Viola and M.J. Jones, "Rapid object detection using a boosted cascade of simple fea-tures," *Proc. CVPR*, no. 1, pp. 511-518, 2001.
- [21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," In: *Proceedings of the of the IEEE conference on computer vision and pattern recognition*, 580-587, 2014.
- [22] R. Girshick, "Fast-cnn," In: *Proceedings of the of the IEEE international conference on computer vi-sion*, pp. 1440-1448, 2015.
- [23] S. Ren, K. He, R. Girshick, and J. Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks," In *Advances in neural information processing systems*, pp. 91-99, 2015.
- [24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," 2016. Doi:10/1007/978-3-319-46448-0_2
- [25] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," arXiv preprint (2017).
- [26] T. Mudumbi, N. Bian, Y. Zhangand, F. Hazoum, "An Approach Combined the Faster RCNN and Mobilenet for Logo Detection," In *Journal of Physics: Conference Series*, vol. 1284, no. 1, p. 012072, 2019. IOP Publishing.
- [27] Image Category Classification Using Deep Learning, <https://www.mathworks.com/help/vision/examples/image-category-classification-using-deep-learning.html>, (accessed 20 June 2020).
- [28] H. El-Khatib, D. Popescu and, L. Ichim, "Deep Learning-Based Methods for Automatic Diagnosis of Skin Lesions," *Sensors (Basel, Switzerland)*, vol. 20, no. 6, p. 1753, 2020, <https://doi.org/10.3390/s20061753>.
- [29] Real time vehicle detection using YOLO, W. Mengxi, 2017, <https://medium.com/@xslittlegrass/almost-real-time-vehicle-detection-using-yolo-da0f016b43de> (accessed 20 June 2020).
- [30] J. Espinosa, S. Velastin, J. Branch, "Vehicle detection using alex net and faster R-CNN deep learning models: a comparative research," In *International Visual Informatics Conference 2017*, pp. 3-15. Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-70010-6_1
- [31] R. Vaishnavi, V. Lavanya and R. Shanta," A Novel Approach to Automatic Road-Accident De-tection using Machine Vision Techniques," *International Journal of Advanced Computer Science and Applications*. vol. 7, no. 10.14569/IJACSA.2016.071130,2016. <https://github.com/vaishnavi29/DCD>
- [32] Accident Data. <https://www.kaggle.com/parthmehta15/accidentdata>, 2020 (accessed 20 June 2020).
- [33] YOLO_v2 and YOLO9000 Part 2, <https://medium.com/adventures-with-deep-learning/yolo-v2-and-yolo9000-part-2-5aa24b6d32e2>, Divakar Kapil, 2018, (accessed 1 January 2021).
- [34] 'ILSVRC2016'.<http://imagenet.org/challenges/ILSVRC/2016/results>. 2017, (accessed 20 June 2020).
- [35] J. Espinosa, S. Velastin, J. Branch, "Motorcycle detection and classification in urban Scenarios using a model based on Faster R-CNN," 2018, 1808.02299.
- [36] Evaluating performance of an object detection model, what is mAP? How to evaluate the per-formance of an object detection model?, <https://towardsdatascience.com/evaluating-performance-of-an-object-detection-model-137a349c517b> ,2020 (accessed 20 June 2020).
- [37] Evaluating Object Detection Models: Guide to Performance Metrics,<https://manalelaidouni.github.io/manalelaidouni.github.io/Evaluating-Object-Detection-Models-Guide-to-Performance-Metrics.html>, 2019 (accessed 20 June 2020).