Emergency Vehicle Detection with Computer Vision

Rawand Rizgar Sulayman, Salih Yaseen Rajab, Bawer Azad Kareem,

Department of Computer Engineering, Faculty of Engineering, Tishk International University, Erbil, Iraq
Correspondence: Rawand Rizgar Sulayman, Tishk International University, Erbil, Iraq.
Email: rawandrzgar1899@gmail.com

ABSTRACT: In this work, we have proposed a model that can detect emergency cars on a heavy traffic road. A populated region like Kurdistan faces too much traffic on the road and because of that emergency car like ambulance and fire-service fall into trouble middle of the road. Our model will solve this problem. It can be embedded with CCTV to track emergency can and give priority in that road to pass the emergency can. With this automated process, no human effort will be required to manually help such scenario. In our project we used a customized Yolov5 object detection algorithm. YOLO an acronym for (You Only Look Once), it is an object detection algorithm that divides images into a grid system. Each cell within the grid is responsible for detecting objects within itself. YOLO models are used for Object detection with high performance which consists of 84 classes to detect and differentiate between 84 different objects. Our model is based on 4 classes which are (Firetrucks, Ambulance, Police Car, and Normal Cars) classes. Our model has achieved impressing results in detecting and identifying emergency cars of all kinds, for Police Cars we have the result of 98%, for Fire Trucks 96%, for Ambulances we've got 89% and for Normal Cars 97% results.

Introduction

Emergency vehicles including ambulances, fire trucks, and police cars, provide emergency response to help people, save lives and ensure security in our society. During emergency conditions, the response time of the emergency vehicle is a significant importance, and, in some cases, a few seconds of timesaving may save lives. For example, in the case of an ambulance, each minute of delay can reduce the survival rate of patients with cardiac arrest by 7-10%. If the ambulance arrives after 8-10 minutes, the chance of survival for those patients will be exceptionally low. The importance of response time for firefighters and police cars is needless to explain.

A well-populated region like Kurdistan faces too many traffic jams. Most of the times emergency vehicles (Ambulance, Firetrucks, etc..) get stuck in traffic causing threats to live. It's mattering much to give priority to these vehicles and to aid to clear its path, but often it is hard or nearly impossible for traffic police to handle these situations and the rapid increase in the number of cars in cities makes it too much harder. Similar systems are now being used in European countries and some American Latin countries and with the aid of the government, the process can become much simpler.

Therefore, an emergency vehicle detection system by using CCTV (Cameras) will be useful to reduce traffic jams and help to save live threats. Emergency vehicles are one of the most important innovations in the modern era it plays their role in life-threatening situations, In scenarios when a patient needs to go to the hospital immediately, it will be very dangerous for the patient if the ambulance car gets stuck in traffic jams or fire truck if it can't get in the time the fire will be out of control, so in our application, we play important role situations like that which is will be by detecting emergency vehicles and opening traffic lane ahead because many people don't bother for opening lanes and Emergency vehicle drives don't know which lane has low traffic and much quieter than other roads.

Automatic traffic control systems can enhance the response time of emergency vehicles bygiving preference to them over others. An effective strategy for realizing this objective is emergency vehicle pre-emption (EVP), which controls the traffic lights in a way that the emergency vehicles can reach their destination in a safer and faster way [1].

There are some solutions of existing applications (systems) made by other researchers, which can be classified into 3 major computer visions (Machine learning, Deep learning, and neural networking) each of them has different algorithms for solving the problem but in the end, all of them do the same idea to solve the problem.

A deep convolutional neural network is the most advanced technology of classification and detection in the image. These days the object detection and classification capabilities have dramatically improved. The convolutional neural network uses the concept of a kernel to process the image. The values of these kernels are learned during training. A deep convolutional neural network is constructed stacking many of such convolutional layers. And kind of hard to train such a network like that because it requires a lot of data sets to train it [2].

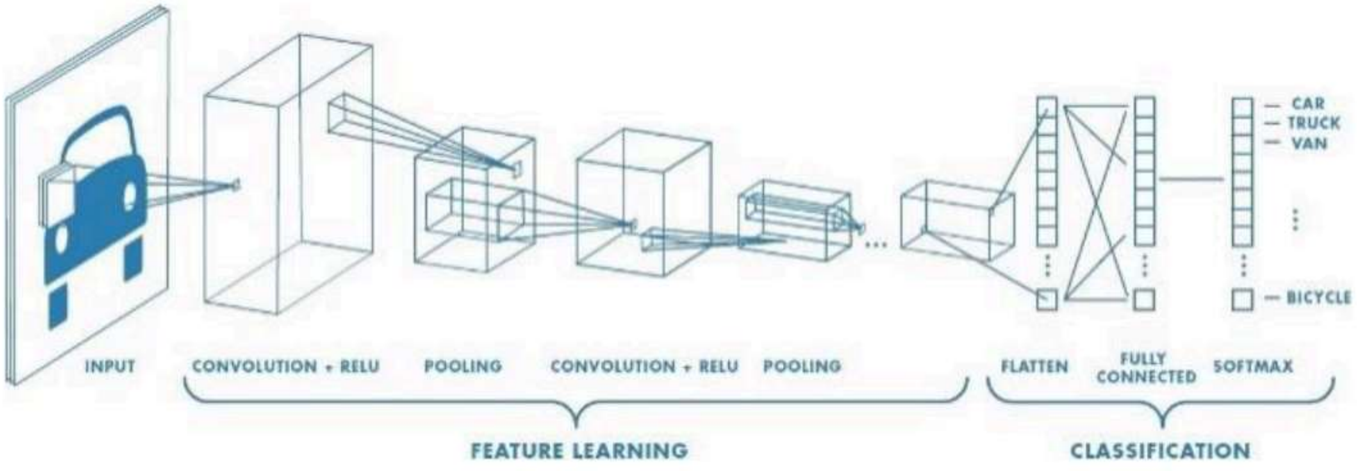


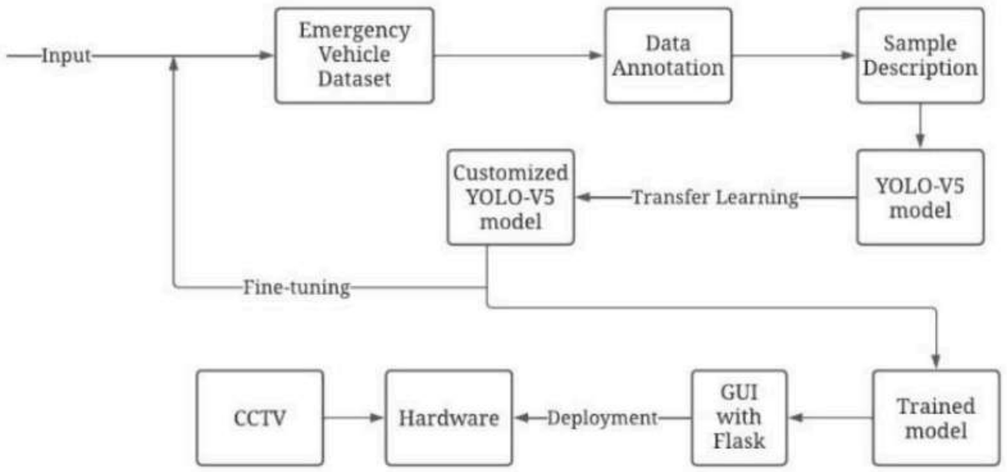
Figure 1: Deep Convolutional Neural Network

Deep learning is a class of machine learning algorithms. Deep learning uses a multi-layer approach to extract high-level features from the data that is provided to it, and it doesn't require to be provided manually for classification. Most of the methods of deep learning use neural networking to archive the results. Some models for detection objects by deep learning are (R-CNN, Fast R- CNN, Faster R-CNN, and Yolo family model).[2]

Machine learning is a part of artificial intelligence which is using computers to learn from the data sets given to it and then make decisions on their own similar to humans. It gives the ability to the computer to be able learning learn and make predictions based on the data and information given to like fed to it. Is it a process of using algorithms to analyze data and then learn from it to make predictions? Some examples of machine learning these days are voice assistants, object detection, etc..

Methodology

In this project, an image processing- based emergency vehicle detection algorithm is proposed. By utilizing the images from exiting traffic cameras and taking advantage of modern low cost but



powerful modern signal processors.

Figure 2: Flow-Chart

Emergency Vehicle Detection uses computer vision and Machine learning, We collected dataset and organized it then we annotated all of the images in the dataset and we have put the labels into sample description folder using YOLOv5 format, Then we have modified the YOLOv5 according to our needs, then we added weight and epochs to the modified YOLOv5 then we trained the data. In case if the results weren't good enough, we will remodify the YOLOv5 and add more epochs and weight then the trained model will be available. After that we have made a GUI using flask to see the results and test the trained model and download the trained model to hardware using raspberry pi and we have connected it to a camera to test our trained model in real life.

Dataset

We downloaded the dataset from the internet and combined it with our Data, then created two separate folders one for training which the algorithm will train on it and one for test to check the training results, in each folder there are 4 different categories of images (Ambulance, Fire truck, Police car, Normal car), which corresponds to the modified version of YOLOv5 then we will annotate and label train folder dataset. Examples of the dataset can be seen in Table below.

Table 1: Datasets

Firetruck	Ambulance	Police	Normal Cars

Training

For the training part, the model needs three things which are: the datasets (images) are notated with box container then separate the images(datasets) into a folder starting with train, after that adding two more folders like train and validation, next inside train and validation folder add labels folder and images, in images folder will put the images and for the label-folder will put .txt files which are the coordinates. This part was most difficult one because the training requires a powerful computer such as high GPU's, more RAM to be able to do the train and took **5 days** to finish **1000 images** for our train which is mean more images(datasets) need more time.

YOLOv5

CNN-based Object Detectors are mainly applicable for proposal systems. YOLO models are used for Object detection with high performance. YOLO divides a picture into a grid system, and every grid detects objects within itself. they'll be used for real-time object detection supported by the info streams. They require only a few computational resources [19].

For understanding how Yolov5 has improved its performance and its architecture as presented in Figure 3, high-level Object detection architecture must be cleared:

General Object Detector will obtain a backbone for pre-training it and ahead to foresee classes and bounding boxes. The Backbones might be running on GPU or CPU platforms. the head will be either one-stage for example: (YOLO, SSD & RetinaNet) for Dense prediction, or two-stage for example: (Faster R-CNN) for the Sparse prediction object detector. New

Object detectors have some layers (Neck) to gather feature maps, and it's between the backbone and therefore the Head.

In YOLOv4, CSPDarknet53 is employed as a backbone and SPP block for increasing the receptive field, which separates the numerous features, and there's no devaluation of the network operation speed. PAN is employed for parameter aggregation from distinctive backbone levels. YOLOv3 (anchor-based) head is employed for YOLOv4. In YOLOv4 new approaches of knowledge augmentation, Mosaic, and SAT (Self-Adversarial Training) have been imported. Self-Adversarial Training conducts in two stages, forward and backward stages. within the 1st stage, the network adjusts the sole image rather than the weights. The network is trained to detect an object on the modified image in the second stage. And Mosaic combines four training images [19].

Yolov5 and Yolov4 are almost similar despite some of the following differences: Yolov4 is released in the Darknet framework, which is written in C. But Yolov5 is based on the PyTorch framework.

Yolov4 uses .cfg for configuration whereas Yolov5 uses .yaml file for configuration.

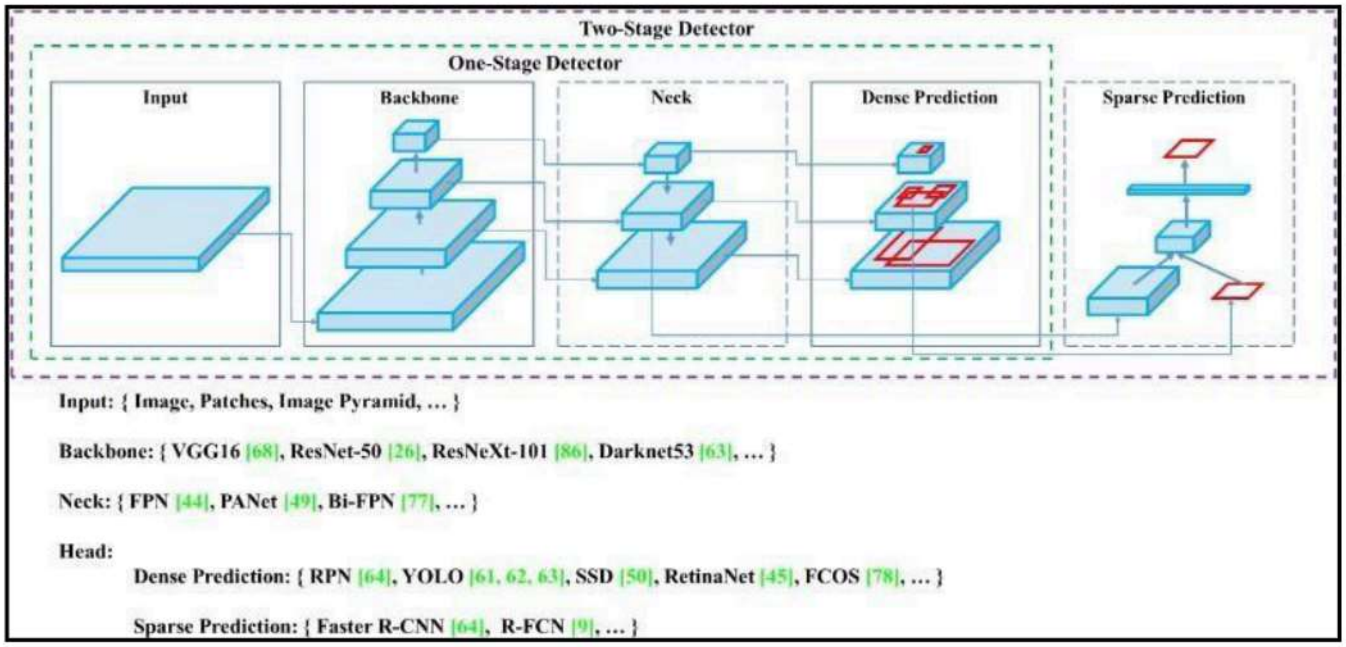

Two-Stage Detector
One-Stage Detector
Input
Backbone
Neck
Dense Prediction
Sparse Prediction
Input: { Image, Patches, Image Pyramid, … }
Backbone: { VGG16 [68], ResNet-50 [26], ResNeXt-101 [86], Darknet53 [63], … }
Neck: { FPN [44], PANet [49], Bi-FPN [77], … }
Head:
Dense Prediction: { RPN [64], YOLO [61, 62, 63], SSD [50], RetinaNet [45], FCOS [78], … }
Sparse Prediction: { Faster R-CNN [64], R-FCN [9], … }

Figure 4: Yolov5 Architecture

Our Model

Our model is based on 4 classes which are (Firetrucks, Ambulance, Police Car, and Normal Car) as shown in Figure 5, the model will train by overtraining the yolov5 model medium weights these weights already trained before, do to create our weight have to reusing this weight to get our new one.

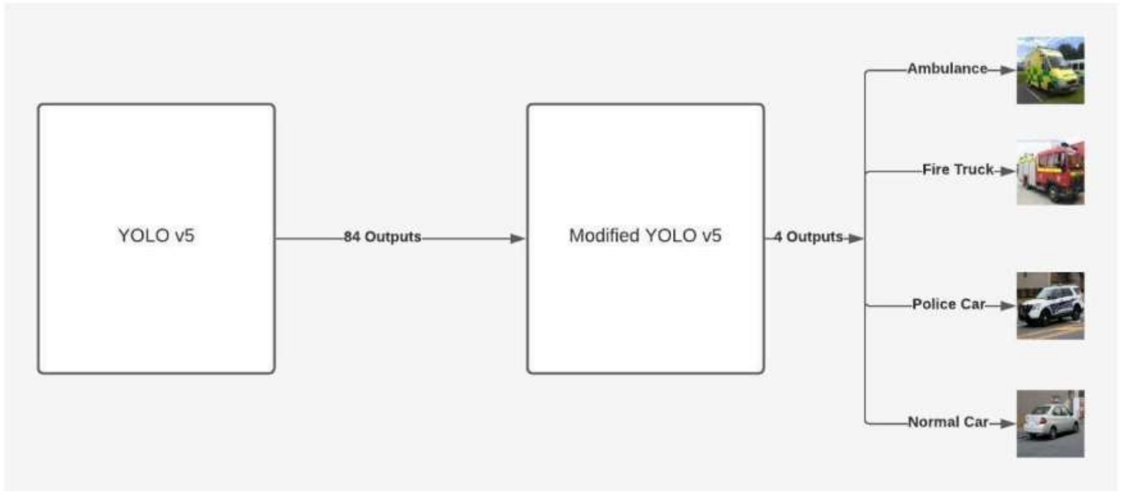


Figure 5: Our Model Structure

GUI

For the User Interface We used Flask (Python framework) for the backend and for the frontend we used plane HTML and Bootstrap (CSS framework). Flask is a Framework of Python that permits us to make up web applications. We can see the API of Flask in Figure attached below. Flask's framework is more exact than Django's framework and is additionally easier to find out because it's less base code to implement an easy web-Application. Flask is predicated on WSGI (Web Server Gateway Interface) toolkit and Jinja2 template engine. In this research we build a web interface based on three things: first detecting the object by live streaming, second playing video files which is also object detection, and the third one is running(detecting) the objects from a folder. We added bootstrap through CDN link which makes the size of the web smaller.

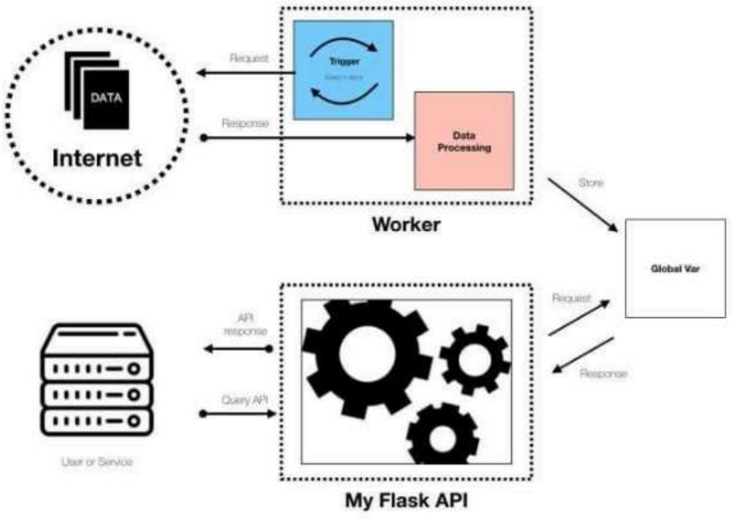


Figure 6: Flask API

Hardware

For our hardware we'll be using Raspberry Pi minicomputer. Raspberry Pi plugs into a computer monitor or TV and uses an ordinary keyboard and mouse. it's a capable little device that permits people to explore computing, and to be able to program in languages like Scratch and Python. It's capable of doing everything you'd expect a computer to accomplish, from surfing and browsing the net and playing high-definition videos, to creating spreadsheets, word processing, and playing games.

Figure 8: Raspberry-Pi Hardware

Training and Validation Results

For training and testing purposes for our model, we should have our data broken into three distinct datasets. These datasets will consist of the following:

- Training Set
- Validation Set
- Testing Set

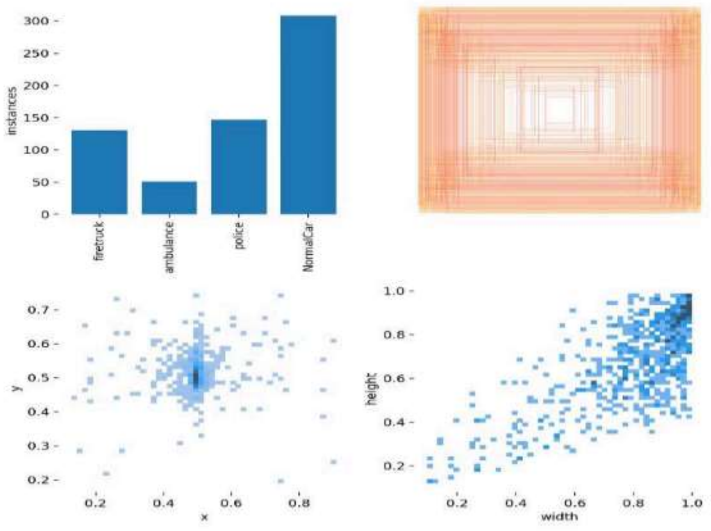


Figure 9: Training and Validation

Figure 9 is labels of our classes which are four classes including firetruck, ambulance, police, normal vehicles. the top left shows that the numbers of images(datasets) during the training and validation, the left bottom is about the coordinates of the images that are notated. the top right shows the output of the detected objects during the training and validation.

Training Results

The training set is what it sounds like. It' the set of data used to train the model. During each epoch, our model will be trained repeatedly on this same data in our training set, and it will continue to learn about the features of this data. The hope with this is that later we can deploy our model and have it accurately predicted on new data that it's never seen before. It will be making these predictions based on what it's learned about the training data. Fig 4.3 below is the results of training and validation.

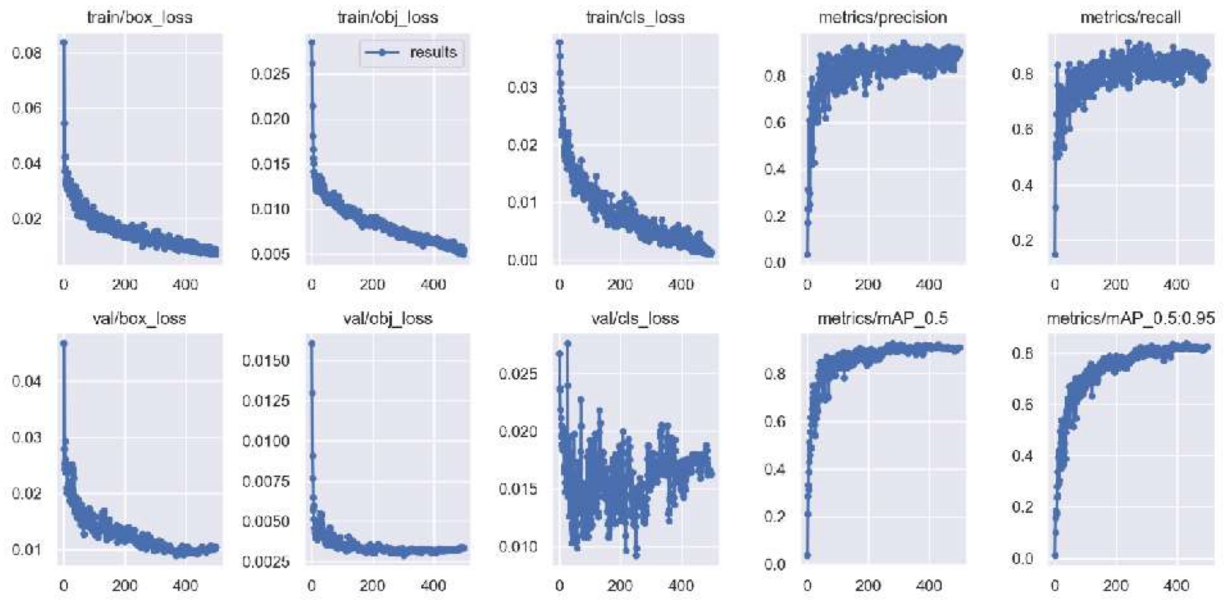
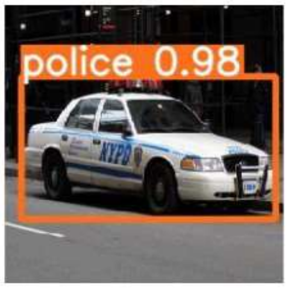
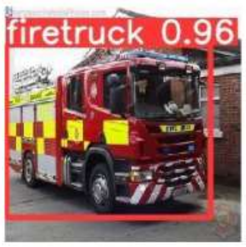
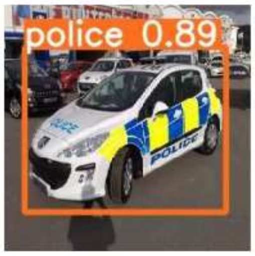
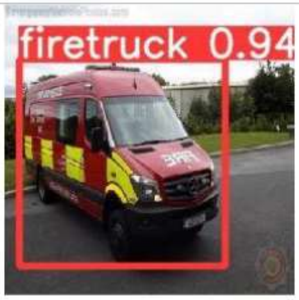
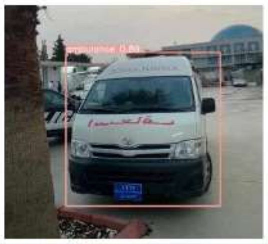
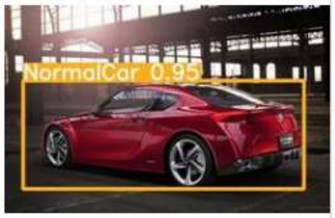
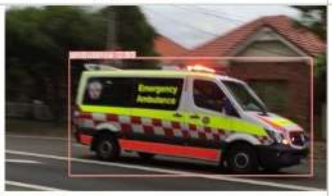
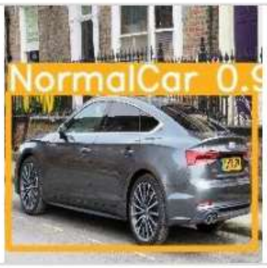


Figure 10: Training Results

Testing Results

Table 2: Testing Results

Before Detecting	After Detected	Before Detecting	After Detected
			
			
			
			

Future Work

Probing deeper, the results in this research also provide a strong foundation for future work in awareness and in peripheral displays. Our area of future work is to feature Sound Recognition System for emergency vehicles which are recognized by their siren sound. The Siren Sound is a special signal sounded by alarm systems or emergency vehicles. When an emergency vehicle performs its task, the siren sound is issued to alert other drivers or pedestrians on the road. However, private cars' drivers may sometimes not hear nearby siren sounds because of the interference of the in-car audio signal, cars nowadays have soundproofing ability, or maybe the distraction of drivers themselves. This problem may lead to a delay in providing emergency services or maybe traffic accidents thanks to inappropriate communication and cooperation. Therefore, our future work will be adding the sound recognition system to our project to develop it and make it more useful and practical to our region

References

[1] M. Asaduzzaman, "A traffic signal control algorithm for emergency vehicles," Memorial University of Newfoundland, 2017.

[2] F.Chollet, "Xception: Deep learning with depthwise separable convolutions," arXiv preprint, pp. 1610–02 357, 2017A. m. traffic. *SCATS Package Configurations*. Available:

[3] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.H. X. Liu, W. Ma, X. Wu, and H. Hu, "Real-time estimation of arterial travel time under congested conditions," *Transportmetrica,* vol. 8, no. 2, pp. 87- 104, 2012.

[4] Micek, J., Kapitulik, J. (2011). Wireless Sensor Networks in Road Transportation Applications. Proceedings of the VII-the International Conference in MEMS Design. 114- 119.

[5] D. Cire san, U. Meier, and J. Schmidhuber. Multi-column deep neural networks forimage classification. Arxiv preprint arXiv:1202.2745, 2012.B. Fazenda, H. Atmoko, F. Gu, L. Guan, and A. Ball, "Acoustic based safety emergency vehicle detection for intelligent transport systems," in *2009 ICCAS-SICE*, 2009, pp. 42504255: IEEE.

[6] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. IJCV, 61(1):55–79, 2005.

[7] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated Recognition, Localization, and Detection using Convolutional Networks. In ICLR, 2014.

[8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale

image recognition. In ICLR, 2015.

[9] J. Zhu, X. Chen, and A. L. Yuille, “DeePM: A deep part-based model for object detection and semantic part localization,” arXiv:1511.07131, 2015.

[10] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In ECCV, 2016.

[11] P. Sermanet, D. Eigen, S. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “OverFeat: Integrated recognition, localization and detection using convolutional networks,” in ICLR, April 2014.

[12] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.

[13] Szegedy, C., Reed, S., Erhan, D., Anguelov, D.: Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441 v3 (2015).

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

[15] He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification (2015).

[16] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015.

[17] R. A. Dobre, C. Negrescu, and D. Stanomir, ‘‘Improved low computational method for siren detection,’’ in Proc. IEEE 23rd Int. Symp. Design Technol. Electron. Packag. (SIITME), Oct. 2017, pp. 318–323.

[18] T. GreyB, “Top 30 autonomous vehicle technology and car companies,” Jun 2021.

[Online]. Available: https://www.greyb.com/autonomous-vehicle-companies/

[19] Surya Gutta, “Object Detection Algorithm — YOLO v5 Architecture”, Aug 2021. [Online]. Available: Object Detection Algorithm — YOLO v5 Architecture.