Road Peculiarities Detection using Deep Learning for Vehicle Vision System

M.R. Rani¹*, M.Z.C. Mustafar¹, N.H.F. Ismail¹, M.S.F. Mansor¹, Z. Zainuddin¹
¹Safety & Intelligent Drive, PROTON, Shah Alam, Malaysia

*E-mail: rahairirani@proton.com

Abstract. Recent development of Advance Driver Assistance System (ADAS) has seen various advancement in object detection for vehicle vision system, particularly on the detection of other vehicles, pedestrians, road lane and signage. While these detections can provide assistant to avoid road accidents, they still lack to include road condition factors that also contributed to road accidents in Malaysia. This paper proposes a detection of the road peculiarities such as pothole and road bumps to act as additional safety feature in ADAS. With the breakthrough of deep learning in solving image recognition problems, this work takes advantage of Single Shot Detector (SSD)-MobileNetV2 as the detection algorithm, implemented on the real-time. The training images for potholes and road bumps taken from the Malaysia roads are fed into the detection model, and then the pre-trained weights are fine-tuned over the training process. The results show that the detection algorithm can predicts the potholes and road bumps, while exhibit the detection accuracy and confidence limitation due to the variety of shape and pattern of potholes and road bumps. Testing the detection algorithm with NVIDIA Jetson Nano yielded about 20 frames per second (fps), suitable for real-time applications.

1. Introduction
Over the last decade, safety has been a noteworthy concern in late vehicle innovation and tremendous researches and works has been conducted guided towards the course. Investigations regarding the safety of vehicle occupant have established two standards [1]: passive and active safety systems. Formerly, the passive safety system was developed to reduce any injury to the vehicle occupant through passive systems such as seat belts and air bags subsequent the unfortunate incidences. They have become standard vehicle safety equipment and have been successful in reducing injuries or fatalities in accidents. However, the air bags are deployed after the accident has already occurred and it would be great if the accident can be fully avoided. There has been an increasing interest in active safety and have seen a growing trend among automotive industries since then. The main purpose of active systems is to assist the driver in avoiding the accidents before they occur. These systems are also known as Advance Driver Assistant Systems (ADAS) and design to help the driver by taking the pressure off the driver in standard situations. A notable example of ADAS are Forward Collision Warning (FCW), Autonomous Emergency Braking (AEB), Lane Keeping System (LKS) and Lane Change Assist (LCA). In recent years, these ADAS has been integrated to enable full autonomous driving. To develop such system, robust and accurate object detection is the important area to be covered.

The role of object detection in understanding and analysing the driving scene is of great importance in order to build more intelligent driver assistance systems. This feature give ability to the road signs
and speed limits in road signage detection. Object detection also can be incorporating with other sensors such as radar and light detection and ranging (LIDAR) to form the vehicle perception.

Road peculiarities such as potholes and road bumps will change the vehicle dynamic motion. Potholes will cause the vehicle to accelerate suddenly and further decreases the friction between the tyres and the road. These situations also lead to a dangerous vehicle deviation or out of control. In addition, potholes in the road can cause damage to the wheel, suspension and chassis components. Pothole size and vehicle speed were the main factors for the damage. The larger pothole can cause severe damage or alignment issues. Road bump acts for slowing traffic in low speed limit areas such as car parks, residential areas, office complexes and industrial areas. Detection of road peculiarities such as potholes and road bumper are crucial for vehicle driving dynamics and road safety. The objective of this paper tries to extend the capability of vehicle object detection system with the detection of road peculiarities. According to [2], the road condition factors correspondingly about 0.4 times less compared to the absence of such factors to the road accidents. This detection expected to increase the driving safety especially in the developing country roads such ASEAN.

2. Methodology

Object detection is the computer vision technique for locating the instance of objects in image or video. Object detection algorithms typically leverage machine learning or deep learning to produce meaningful results. When humans look at images or video, they can recognize and locate objects of interest within a matter of moments. The goal of object detection is to replicate this kind of ability using a computer.

Method for implementing object detection generally fall into either machine learning (ML) based approaches or deep learning (DL) based approaches. In ML, it is necessary to first define features and then using method such as Support Vector Machine (SVM) to do the classification. This feature selection technique is best for classification of clear feature different in classes, however, is less likely to work in distinguish between a car and a truck.

Contrary to ML, DL able to do end-to-end object detection without specifically defining the features as show in Figure 2. This means the algorithm only take the images and their bounding boxes as input in the learning process. DL in object detection is typically based on convolution neural network (CNN). There are a number of popular object detection algorithm based on deep learning such as:

i. Region-based Convolutional Neural Network (R-CNN) [3]
ii. Single Shot Detector (SSD) [4]
iii. You Only Look Once (YOLO) [5]
iv. Retina-Net [6]
Due to its accuracy and performance in real-time, this paper uses SSD as the object detector in the algorithm and MobileNetV2 as the feature extractor.

Figure 2. The difference between machine learning and deep leaning lies on the feature extraction technique

2.1 Single Shot Detector (SSD) MobileNetV2
SSD MobileNetV2 consists of two major part in its algorithm, SSD as the object detector and MobineNetV2 as the base network or feature extractor. Feature extractor works to extract the significant information from the image before being passed to object detector in order to determine the object classes and bounding boxes in the image.

2.1.1 MobileNetV2 - MobileNetV2 is the successor of MobileNetV1 [7] that intends to design an efficient neural network based on the mobile devices. This refinement from the V1 that makes it even more efficient and powerful [8].

The idea behind MobileNetV1 is that convolutional layers, which are essential to computer vision tasks but are quite expensive to compute, can be replaced by so-called depth wise separable convolutions.

The job of the convolution layer is split into two subtasks: first there is a depth wise convolution layer that filters the input with rectified linear unit (ReLU), followed by a 1×1 (or pointwise) convolution layer that combines these filtered values to create new features.

Figure 3. Depthwise separable convolution block in MobileNet

MobileNetV2 still uses depth wise separable convolutions, but there are major changes to its main building block as show in Figure 4.
There are three convolutional layers in the block. The last two are: a depth wise convolution that filters the inputs, followed by a 1×1 pointwise convolution layer. However, this 1×1 layer now has a different job.

In V1 the pointwise convolution either kept the number of channels the same or doubled them. In V2 it does the opposite: it makes the number of channels smaller. This is why this layer is now known as the projection layer, it projects data with a high number of dimensions (channels) into a tensor with a much lower number of dimensions.

The second new thing in MobileNetV2’s building block is the residual connection. This works just like in ResNet [9] and exists to help with the flow of gradients through the network.

The idea behind V1 was to replace expensive convolutions with cheaper ones, even if it meant using more layers. That was a great success. The main changes in the V2 architecture are the residual connections and the expansion/projection layers.

2.1.2 Single Shot Detector (SSD) - SSD is designed to be independent of the base network (feature extractor), and so it can run on top of pretty much anything, including Mobile Net V2 or V1. The complete network can be shown as in Figure 6.

The network not only takes the output of the last base network layer but also the outputs of several previous layers, and these outputs are fed into the SSD layers. The job of the Mobile Net layers are to convert the pixels from the input image into features that describe the contents of the image, and pass
these along to the other layers. Hence, Mobile Net is used here as a feature extractor for a second neural network.

![Figure 6. Complete network of SSD-Mobile Net](image)

**Figure 6.** Complete network of SSD-Mobile Net

In the case of object detection with SSD, not just these high-level features are processed but also lower-level ones, which is why the previous layers are also forwarded. Since object detection is more complicated than classification, SSD adds many additional convolutional layers on top of the base network. Thus, it is crucial to have a base network that is fast, and that exactly what MobileNetV2 is.

### 2.2 Training the Dataset

In deep learning, there are two main processes involved, which are training and inference. In training, the network is guided to converge and archive a desired accuracy based on the labelled dataset. This requires running a thousand or perhaps millions of experiments by adjusting the model weights in the network. The training process is compute-extensive and often done in data centre or a powerful computer.

Deep learning inference is a process of using a trained neural network model to make prediction against previously untrained data. The inference process typically running on a smaller device and required further optimization of the trained model.

![Figure 7. The training and inference process in deep learning](image)

**Figure 7.** The training and inference process in deep learning

#### 2.2.1 Creating the Labelled Dataset -
In order to perform deep learning training, a dataset of road peculiarity must first be established. The dataset consists of 500 images of road bumps and potholes for each class that taken along the Malaysia roads with camera was mounted at vehicle windscreen. The images have varied in shape and structure of the road bumps and potholes to make it a robust dataset. These images went through a manual labelled process in order to specify the bounding box of the classes in the images according to its class.
3. Results

Training a new dataset in deep learning from scratch is competitively expensive. One way to speed up the training is to employ the transfer learning technique, which uses the weights that has been trained for other classes. For this purpose, a SSD MobileNetV2 model that has been trained for COCO dataset [10] will be used as the initial weights. The training configuration for the transfer learning as Table 1.

The objective of the learning process is minimizing the total loss, which is the combination of the classification loss and the localization loss. By readjusting the neural network parameters in the next step, the algorithm tries to reduce the total loss over time.

Table 1. Training configuration for SSD MobileNetV2 Transfer Learning

| Parameters               | Setting                      |
|--------------------------|------------------------------|
| No. of steps             | 200 000                      |
| Batch size               | 4                            |
| No. of classes           | 2                            |
| Initial learning rate    | 0.004                        |
| Learning rate decay steps| 800720                       |
| Deep learning library    | Tensor flow v1.5             |
| Central Processing Unit (CPU) | Intel i5 2400 – 12 GB       |
| Graphic Processing Unit (GPU) | NVIDIA GTX 1060 6GB        |
| Initial weights          | SSD MobileNetV2-COCO – 2018-03-29 |

Classification loss measures the predicted class probability towards the ground truth. For the classification loss function, these probabilities are measures towards the true probability distribution with cross entropy, which quantifies the difference between two probability distributions. Equation 1 presents cross entropy, where $\hat{y}$ is the predicted value, $y$ is the ground truth value and $M$ is the number of classes.

$$\text{classification loss} = \sum_{i}^{M} y_i \log (\hat{y}_i)$$  \hspace{1cm} (1)

The transfer learning of the classification loss versus the number of steps is shown as in Figure 9. The faded signal is the actual loss while the lighten signal is the smoothed loss.
Localization is the ability of the network to locate an object in an image. The localization loss is the computed error between the ground truth and the predicted bounding boxes. Smooth L1 localization loss is defined by Equation 2 and as a penalty term to the loss function with the purpose to avoid overfitting [6].

\[
\text{localization loss}(x) = \begin{cases} 
0.5x^2 & \text{if } |x| < 1 \\
|x| - 0.5 & \text{if otherwise}
\end{cases}
\]  

(2)

The localization loss for the transfer learning is shown as in Figure 10. Like the classification loss, the localization loss also made decrease as the step increases.

The total loss is summation of the classification and localization loss. Equation 3 defines the total loss and the total loss of the transfer learning is shown as in Figure 11.

\[
\text{Loss}_{\text{Total}} = \frac{1}{N} (\text{Loss}_{\text{classification}} + \alpha \text{Loss}_{\text{localization}})
\]

(3)

where N is the number of matched bounding boxes and \(\alpha\) balances the weights between two losses, picked by cross validation.
At the end of the transfer learning, the training results can be summarized by Table 2.

| Results          | Results       |
|------------------|---------------|
| Classification loss | 0.9812        |
| Localization loss  | 0.3604        |
| Total loss        | 1.932         |
| Total training duration | 8 days 8 hour 44 minutes |

4. Discussion
The trained model produced in the training process is infeasible to run in the vehicle. Typically, the vision controller in the vehicle uses small form computer, exactly what NVIDIA Jetson Nano is. In order to optimize the trained model, the batch size is reduced to single batch and the model is convert to low precision model using NVIDIA TensorRT software.

![Figure 11. Total loss of SSD-MobileNetV2 transfer learning](image)

![Figure 12. The optimization process from the trained model to inference model in NVIDIA TensorRT](image)

4.1 Testing the model
In order to test the accuracy and robustness of the optimized inference model, the test images that different in the dataset are used as input. The example of the results of the testing can be shown as in Figure 13.
Figure 13. Road peculiarities detection with SSD MobileNetV2 for (a) pothole and (b) road bump

The summary of the detection result from 10 test images for each class can be illustrated in the Table 3.

| Road Peculiarity | Accuracy Percentage | Average Confidence |
|------------------|---------------------|--------------------|
| Potholes         | 60%                 | 0.65               |
| Road bumps       | 70%                 | 0.43               |

In potholes detection, the model is able to detect the potholes even the holes are filled with water. This is due to the training dataset also contains the wet potholes. However not all the potholes are detected, only 60% accuracy in the test images since large and distant potholes could not be recognized. This may be due to the dataset does not contains enough data for that situations.

Road bumps detection is a quite challenging task due to the variety of their patterns and the bigger size of bounding box. Bigger the size of bounding box causes more unwanted noise present in the dataset. Although a clear road bump is using as the test image, the detection only yields 0.43 of average confident level.

Running the optimized SSD MobileNetV2 with NVIDIA Jetson Nano yields ~20 frame per seconds (fps), good enough to be considered as real time applications. The detection speed is crucial in this work since the detection system is built for vehicle system. The vehicle able to react quickly when the detection speed is increasing.

5. Conclusion
SSD MobileNetV2 has been implemented to detect road peculiarities such as potholes and road bumps in this paper. The results show the detection algorithm able to recognize the road peculiarities but with the accuracy limitations in potholes and detection confidence limitation in road bumps. From our observation, the dataset size plays an important role in order to make the detection more robust and accurate. The detection model could adapt more patterns and situations when the dataset size is increased.

In future, we would like to increase the detection performance by increasing the training dataset size (over 5000 images for each class) in order to make detection algorithm more robust.

Acknowledgment
The authors would like to thank to Perusahaan Otomobil Nasional Berhad (PROTON) for funding this work.
References

[1] Asean N. ASEAN NCAP Roadmap 2018 Malaysian Institute of Road Safety Research (MIROS), Kajang; 2021-2025.

[2] Musa, M. F., Hassan, S. A., & Mashros, N. 2020 The impact of roadway conditions towards accident severity on federal roads in Malaysia. PLoS one, 15(7), e0235564.

[3] Girshick, R., Donahue, J., Darrell, T., & Malik, J. 2014 Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition pp. 580-587.

[4] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. 2016 Ssd: Single shot multibox detector. In European conference on computer vision pp. 21-37. Springer, Cham.

[5] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition pp. 779-788.

[6] Goyal, P. and Kaiming, H., 2018. Focal loss for dense object detection. IEEE transactions on pattern analysis and machine intelligence, 39, pp.2999-3007.

[7] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[8] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition pp. 4510-4520.

[9] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. 2016. p. 770-8.

[10] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L. 2014 Microsoft coco: Common objects in context. In European conference on computer vision pp. 740-755