

An Image-Classifying-Based Diagnostic Method Applied to Electronic Fuel Injection System for Gasoline Engine

Quang T. D. Pham^{1,2}, Phat Lam Huynh^{1,2}, Quan Pham Van^{1,2}, Duong Nguyen Thai^{1,2}, and Long Tran Dang^{1,2*}

¹ Faculty of Transportation Engineering, Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Viet Nam

² Vietnam National University Ho Chi Minh City, Ho Chi Minh City, Viet Nam

*E-mail: trandanglong@hcmut.edu.vn

Abstract. The electronic fuel injection (EFI) system significantly governs modern gasoline engine performance. Dirty sensors, weak actuators, broken electric wires, shorted circuits, and mechanical issues are most causes of abnormal system operations. Technicians can use the integrated onboard diagnostic (OBD) function to monitor system behaviour and then apply a standard Symptom-System-Component-Cause (SSCC) diagnostic process to determine what are causing rough idling, hesitate acceleration, excess exhaust gas emissions, and reduced fuel economy. This is a complicated and time-consuming procedure that is prone to human errors, particularly when mechanical causes occur, requiring skilled and experienced technicians. This study aims to demonstrate the feasibility of an automatic diagnostic solution for identifying causes of bad symptoms in EFI systems. Engine live data are collected from the OBD system while the engine is running through a diagnostic engine speed cycle. Live data features are then extracted and visualized as a radar-chart-type diagnostic image, which contains the signatures of normal or abnormal engine operations and can represent a specific cause of failure. By classifying the obtained diagnostic image, the cause of failure is identified. Experiments were conducted with a 4A91 engine, which calculates intake mass air flow from engine speed and intake air density instead of using an air flow meter. Mechanical issues such as cracked intake manifold, evaporative purge valve leak, and weak fuel pump were generated with different degrees of failures. Various pruned versions of the VGG-16 deep learning model were applied to classify diagnostic images thereby can indentify the causes of failure. The results showed that in the case VGG-16 model reduces 2 convolutional layers and 1 dense layer, it is most suitable for this automatic diagnostic application. The proposed diagnostic method can be further improved to detect multi-cause failures as well as predict system degradation trends for early warning and preventive maintenance.

Keywords: gasoline engine, electronic fuel injection, automatic diagnosis, image classification, OBD-II.

1. Introduction

To meet modern automotive standards, fuel injection systems have continuously evolved, leading to the development of the Electronic Fuel Injection (EFI) system. The EFI system precisely meters fuel into the combustion chamber with respect to the amount of air charged in the intake stroke, governing engine performances in terms of fuel efficiency, emissions, and responses [1]. Instead of using a sophisticated air mass flow meter which increases overall cost, the amount of charged-air can be determined by the speed-density method basing on engine speed, manifold absolute pressure, and intake air temperature. The overall diagram of a typical EFI system using the speed-density method for intake air mass measurement is illustrated in Figure 1.

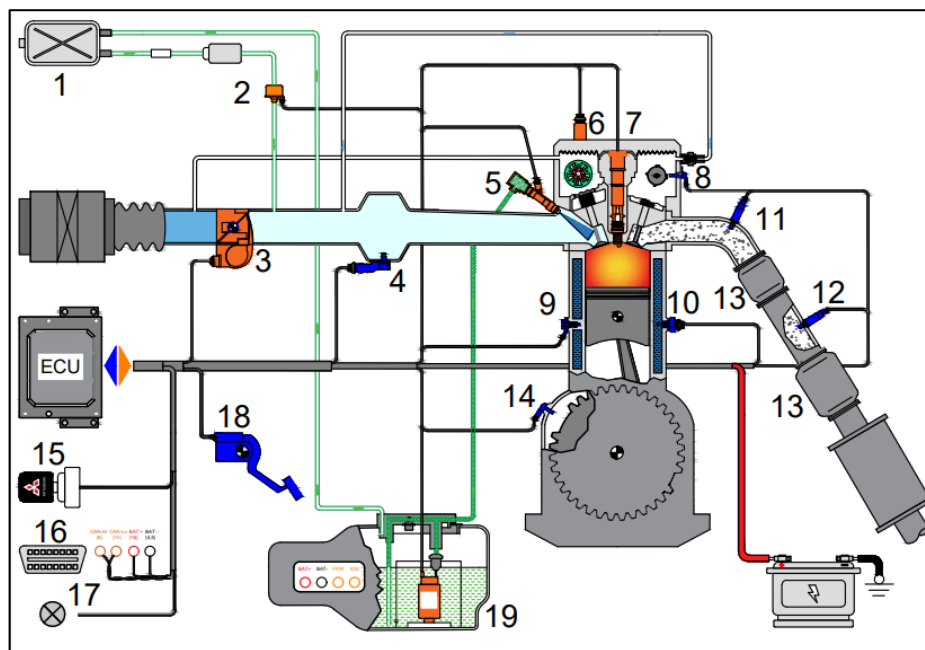


Figure 1. Overall diagram of a typical EFI system using the speed-density method for intake air mass measurement. 1 - charcoal canister; 2 - evaporative purge valve; 3 - electronic throttle control; 4 - manifold absolute pressure and intake air temperature sensor; 5 - injector; 6 - oil feeder control valve; 7 - spark plug and coil-on plug; 8 - camshaft position sensor; 9 - knock sensor; 10 - engine coolant temperature sensor; 11 - pre-oxygen sensor; 12 - post-oxygen sensor; 13 - three-way catalyst; 14 - crankshaft position sensor; 15 - ignition switch; 16 - OBD-II data link connector; 17 - malfunction indicator lamp; 18 - accelerator pedal position sensor; 19 - fuel pump.

During operation, the system faces challenges due to various potential failures such as: contaminated sensors, weak actuators, damaged wires, short circuits, and mechanical issues. For fault diagnosis, technicians utilize On-Board Diagnostics (OBD) to monitor the system via diagnostic trouble codes (DTCs) and follow the Symptom-System-Component-Cause (SSCC) process [2]. However, this procedure is complex and time-consuming, requiring skilled and experienced technicians, particularly in the cases of mechanical failures which cannot be detected via abnormal electric signals.

Many studies have been conducted to address these challenges. Notable among these was the work of Du Danfeng et al., who developed a fault diagnosis method for EFI engines using the MATLAB software package, which facilitated failure detection without relying solely on codes [3].

Tran Thanh Khiet et al. introduced a quick diagnostic method by comparing characteristic datasets between causes [4]. Alexandros Mouzakitis et al. proposed new methodologies and tool developments for automated onboard vehicle diagnostics testing [5]. Despite demonstrating promising results, these methods presented challenges in implementation. Fortunately, the advent of Machine Learning (ML) has opened new avenues for diagnostics development. ML has been applied across various domains, including security, image analysis, deep learning, object recognition, and pattern recognition [6]. In the automotive field, studies have leveraged ML, such as Mohammed Al-Zeyadi et al., who utilized ML to automate diagnostics across a range of cause types [7]; Hong Guo et al., who developed techniques for signal decomposition, diagnostic feature extraction, and intelligent diagnosis using ML [8]; Ali Khodadadi et al., who used ML to classify service requests to improve the vehicle diagnostics process [9]; and Alex Gong et al., who used various machine learning algorithms, which were applied to the failure prediction and warning [10].

This study proposed a novel diagnostic concept for EFI systems by using image classification model. By performing a specific diagnostic engine test cycle, engine live data was collected from the OBD system, and then visualized into radar-chart-type diagnostic images representing normal or abnormal engine conditions, helping to pinpoint failure causes.

The paper was structured as follows. Section 2 described the concept of image-classification-based diagnostics. Section 3 elaborated on the image-classification model. Section 4 outlined the results, and Section 5 concluded the study.

2. Image-classification-based diagnostic concept

The image classification-based diagnostic method represented an innovative approach in automotive diagnostics utilizing an image classification model, as illustrated in Figure 2.

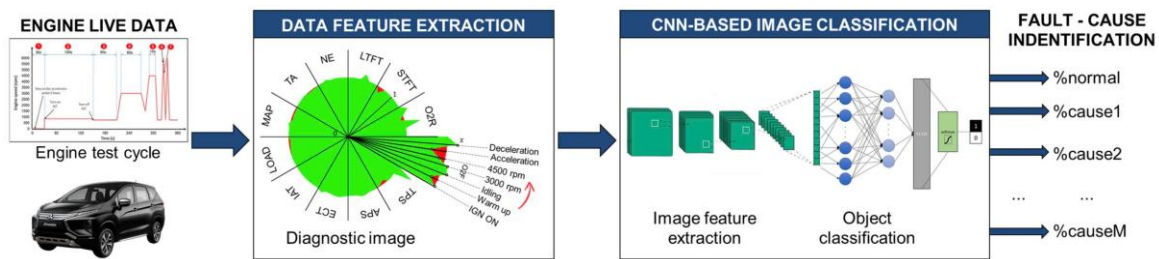


Figure 2. Overall layout for the image-classifying-based diagnostic method.

2.1 Engine test cycle

Each cause exhibited distinct characteristics and impacts on the engine under different operating conditions. To enhance cause detection ability, the diagnostic engine speed cycle in this study was structured based on basic engine operating stages including start-up, warm-up, idling, low and high speeds, acceleration, deceleration, coasting, and speeds beyond the optimal range with a specific performing time. To facilitate stable data collection, the cycle was conducted with the vehicle stationary and under no-load conditions. Therefore, the coasting and exceeding optimal speed stages were not considered. Although the warm-up stage also occurred under load when

turning on the air conditioning, it was included to lengthen the warm-up time. The complete engine speed cycle was shown in Figure 3.

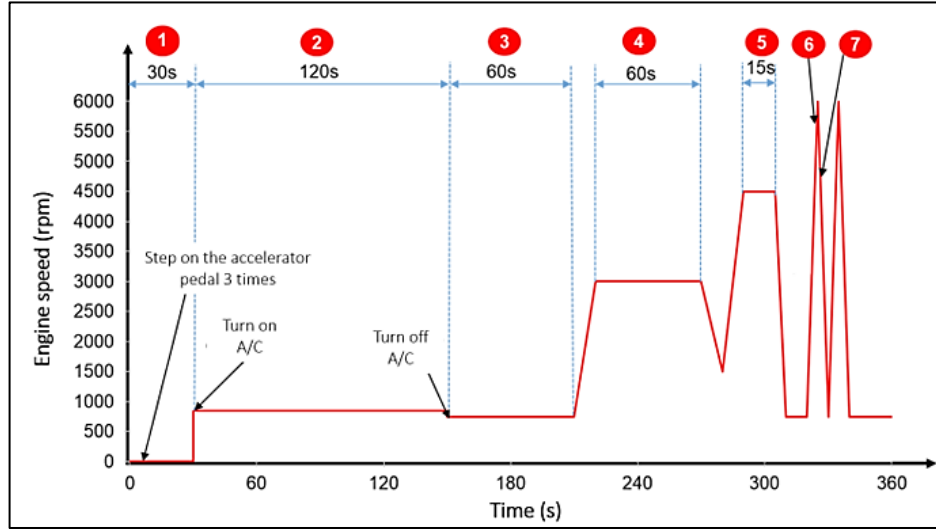


Figure 3. Engine test cycle for diagnostic purposes used in this study, including 7 stages: 1-Ignition on; 2-warm up; 3-idling; 4-3000rpm; 5-4500rpm; 6-Acceleration; 7-Deceleration.

In this study, live data were collected under two conditions, including normal engine operation and intentionally simulated faults. A dataset contained a total of 84 values, with 12 live data parameters obtained across 7 stages, including: Engine speed (N_e), manifold absolute pressure (MAP), engine load (Load), timing advance (TA), accelerator pedal position (APS), throttle position (TPS), front oxygen sensor voltage (O2F), and rear oxygen sensor voltage (O2R), intake air temperature (IAT), engine coolant temperature (ECT), long-term fuel trim (LTFT), and short-term fuel trim (STFT).

2.3 Diagnostic image

The diagnostic image was structured as a radar chart with multiple axes, each representing live data collected at 7 engine operating stages, arranged counterclockwise. However, since these data differed in units and value ranges, it was necessary to normalize these data to a common scale, ranging from 0 to 1. The process was based on normalizing values from a standard data set, which was compiled from 10 datasets under normal engine conditions. For each data set, appropriate processing methods were applied based on the characteristics of the data, which are divided into three groups.

Group 1: Live data collected from warm-up to 4500 rpm stages, excluding STFT and LTFT. This group denoted data continuously fluctuating around specific values. For this group, the average value of each stage was determined:

$$x_{\text{average}} = \frac{\sum_{i=1}^n x_i}{n} \quad (1)$$

where:

x_{average} : The average value of live data parameter at each stage

x_i : i^{th} value in the sample

n : The number of values in the sample

Group 2: STFT and LTFT at all stages. These data had significant changes between normal and failure conditions, leading to a notable difference in the diagnostic image. To avoid distortion in the image, conventional values were assigned to the data based on conversion ranges determined through experiment, as illustrated in the Table 1.

Table 1. Conversion table of STFT and LTFT at all stage.

Live data	Conversion range	Conventional value
STFT	$x_{average} \leq -25\%$	0.5
	$-25\% < x_{average} < -9\%$	0.9
	$-9\% \leq x_{average} \leq 9\%$	1
	$9\% < x_{average} < 25\%$	1.1
	$x_{average} \geq 25\%$	1.5
LTFT	$x_{average} \leq -10\%$	0.5
	$-10\% < x_{average} < -0.78\%$	0.9
	$-0.78\% \leq x_{average} \leq 3.91\%$	1
	$3.91\% < x_{average} < 10\%$	1.1
	$x_{average} \geq 10\%$	1.5

Group 3: Live data collected in ignition-on, acceleration and deceleration stages, excluding STFT and LTFT. For this group, the maximum values during ignition and acceleration, as well as the minimum values during deceleration, were determined to evaluate engine performance at each stage.

Figure 4. An illustration of the diagnostic image structure proposed in this study.

observation. The second observation, representing the experimental-normalized dataset, was filled in green and constructed by connecting adjacent data points, as determined by the following formula:

$$x = \frac{a_i}{A_i} \times 1 \quad (3)$$

where:

A_i : The value of live data in standard dataset

a_i : The value of live data corresponding to A_i in experimental dataset

1: The normalized value of A_i

x : The normalized value of a_i for data point

3. Image-classification model

3.1 Convolutional neural network

The VGG16 architecture is a well-established convolutional neural network, characterized by its straightforward yet powerful design, encompassing 16 operational layers equipped with trainable parameters, as illustrated in the Figure 5.

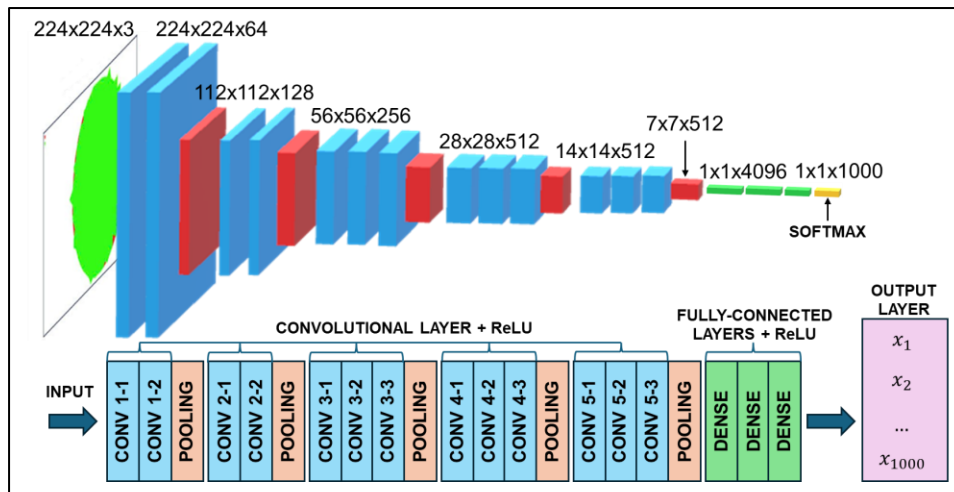


Figure 5. The VGG-16 model architecture.

This architecture incorporates 13 convolutional layers, each employing small 3×3 kernels, and is methodically segmented into blocks. Its uniform design, featuring consistent convolutional layers followed by max-pooling, helps in reducing the spatial dimensions effectively. VGG16 employs ReLU activations for non-linearity and faster convergence, while max-pooling layers simplify feature maps for reduced computational demands. Following the convolutional sequence, the architecture integrates three fully connected layers; the initial two layers consist of 4096 channels, while the final layer outputs a distribution over 1000 classes, aligned with the ImageNet dataset's classification task. The network extensively utilizes the ReLU activation function across its hidden layers to introduce non-linearity. Cumulatively, the model possesses approximately 138 million parameters, predominantly concentrated in the fully connected layers. To simplify the VGG16 model, pruning techniques are described in the following section.

4.2 Pruning convolutional blocks in VGG16

In VGG16, the convolutional blocks could be effectively pruned to streamline the image classification process. The earlier layers excelled at extracting fundamental features such as edges and textures, which were crucial for initial image analysis. In contrast, the deeper blocks in VGG16, responsible for capturing more abstract features, were less essential for the application of this study. Furthermore, reducing the number of layers in the higher blocks did not have negative impact on the performance, as the abstract features they captured were unnecessary for tasks requiring simple classification or those relying primarily on low to mid-level visual features.

Table 2. The architectures of pruned VGG16 models used in this study.

Layers		Feature Map (N)				Output size	Kernel Size	Stride	Activation
		VGG16	VGG16_PM1	VGG16_PM2	VGG16_PM3				
	Input image					224×336×3			
1	Convolution	64	64	32	16	224×336×N	3×3	1	ReLU
2	Convolution	64	64	32		224×336×N	3×3	1	ReLU
	Pooling	64	64	32	16	112×168×N	2×2	2	ReLU
3	Convolution	128	128	64	32	112×168×N	3×3	1	ReLU
4	Convolution	128	128	64		112×168×N	3×3	1	ReLU
	Pooling	128	128	64	32	56×84×N	2×2	2	ReLU
5	Convolution	256	256	128	64	56×84×N	3×3	1	ReLU
6	Convolution	256	256	128		56×84×N	3×3	1	ReLU
7	Convolution	256	256			56×84×N	3×3	1	ReLU
	Pooling	256	256	128	64	28×42×N	2×2	2	ReLU
8	Convolution	512				28×42×N	3×3	1	ReLU
9	Convolution	512				28×42×N	3×3	1	ReLU
10	Convolution	512				28×42×N	3×3	1	ReLU
	Pooling	512				14×21×N	2×2	2	ReLU
11	Convolution	512				14×21×N	3×3	1	ReLU
12	Convolution	512				14×21×N	3×3	1	ReLU
13	Convolution	512				14×21×N	3×3	1	ReLU
	Pooling	512				7×10×N	2×2	2	ReLU
14	Fully connected	✓	✓	✓	✓	4096			ReLU
15	Fully connected	✓	✓	✓	✓	4096			ReLU
16	Fully connected	✓	✓	✓	✓	M+1			ReLU
	Output	✓	✓	✓	✓	M+1			Softmax

^a M represents the number of causes detectable by the model.

Table 2 presented the architectures of the VGG16 model, and its pruned versions: VGG16_PM1 (pruned model #1), VGG16_PM2 (pruned model #2), and VGG16_PM3 (pruned model #3). These pruned versions were designed to reduce computational complexity by progressively decreasing the number of feature maps in the convolutional layers. The original VGG16 architecture employed 64 filters in the first two convolutional layers, increasing to 128,

256, and 512 filters in the deeper layers, with a total of 13 convolutional layers, 3 fully connected layers, and 51,432,290 parameters.

In VGG16_PM1, the feature maps were moderately pruned, retaining 64 filters in the initial layers, 128 in the middle layers, and 256 in the deeper layers, reducing the parameters to 51,404,402. VGG16_PM2 further reduced the feature maps to 32 in the initial layers, 128 in the middle layers, and 256 in the deeper stages, resulting in 24,673,124 parameters. The most significantly pruned version, VGG16_PM3, reduced the feature maps to 16 in the initial layers, 64 in the middle layers, and 128 in the deeper layers, bringing the number of parameters down to 12,829,106.

Despite these substantial reductions in convolutional filters, all pruned models retained the original fully connected layers, making them suitable for resource-constrained environments while maintaining competitive performance.

4. Results

4.1. Experimental diagnostic images for model training

In this study, three mechanical failures including cracked intake manifold (CIM), leaked evaporative purge valve (LEPV), and weak fuel pump (WFP) were simulated on a Mitsubishi 4A91 engine to collect diagnostic images for training the cause-classification model. The CIM simulation method involved in disconnecting the vacuum booster hose from the intake manifold. To simulate various degrees of this failure, a manual valve and an airflow meter were used to control the volumetric flow rate of intruded air, expressed in 30%, 60%, 100%, 160%, 240%, and 280% of the normal intake air flow rate during idling, which was approximately 68 liters per minute at 40°C. For simulating various degrees of LEPV, the purge valve was replaced by a manual valve connected in series with an airflow meter to control air flow and fuel vapor unintentionally drawn into the intake manifold. The volumetric flow rate of intruded air-fuel mixture was expressed in 30%, 60%, 100%, 125%, and 150% of the normal intake air flow rate during idling. To simulate a weak fuel pump, a resistor was connected in series with each injector coil, reducing coil current and thereby decreasing the amount of the fuel injected. The reduction in fuel supply was categorized as 10%, 20%, 30%, and 40% of normal operation.

Several sets of engine live data were collected for each degree of cause, covering the variations of ambient conditions. Features of each engine live data set were extracted and then corresponding diagnostic image was generated.

Figure 6 shows the typical diagnostic images of normal condition, CIM, LEVP, and WFP experimentally obtained from the Mitsubishi 4A91 engine. Feature differences among causes were apparent. Features of causes at low degrees appeared to be almost similar to those of normal condition. The features of CIM were clearly reflected in live data during idling, particularly in TPS, LTFT, Ne, and TA. While TPS and LTFT for LEPV during idling showed the most noticeable changes, in the cases of WFP, the most affected parameter was LTFT during idling.

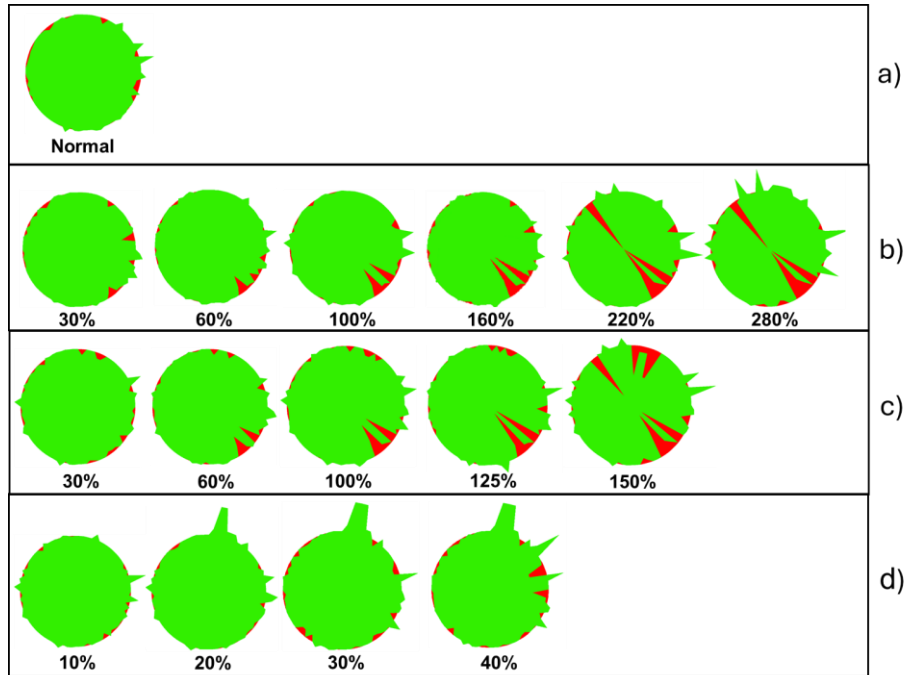


Figure 6. Typical diagnostic images of (a) normal condition; (b) cracked intake manifold; (c) leaked evaporative purge valve, and (d) weak fuel pump experimentally obtained from a Mitsubishi 4A91 engine. Percentages indicated the degrees of failure.

4.2. Model of diagnostic image classification

The dataset used for model training contained 16 classes, comprising a total of 1,321 images, as shown in Table 3. The dataset was divided into training sets for model training and validation sets for evaluating the model's actual performance. The training set, which accounted for 80% of the total images, included 1,053 images, while the validation set, representing 20%, contained 268 images.

Table 3. Training data sets of 15 failure condions including: Cracked intake manifold (CIM); Leaked evaporative purge valve (LEPV); Weak fuel pump (WFP) at various degrees, along with 1 normal condition. Each class contains a different number of images, with a total of 1321 images across all classes

	Normal	CIM 30%	CIM 60%	CIM 100%	CIM 160%	CIM 220%	CIM 280%	LEPV 30%
Total	89	87	87	91	91	91	91	91
Valid	18	18	18	18	18	18	18	18
Train	71	69	69	73	73	73	73	73
	LEPV 60%	LEPV 100%	LEPV 125%	LEPV 150%	WFP 10%	WFP 20%	WFP 30%	WFP 40%
Total	90	101	85	107	59	59	60	58
Valid	18	20	17	21	12	12	12	12
Train	72	81	68	86	47	47	48	46

Accuracy and loss were employed as evaluation metrics to assess the model's performance across 100 test cases, encompassing three mechanical causes with varying degrees of severity. The training results for the VGG16 models and their variants including VGG16_PM1, VGG16_PM2, VGG16_PM3, exhibited a consistent and positive learning trajectory across all architectures, as illustrated in Figure 7.

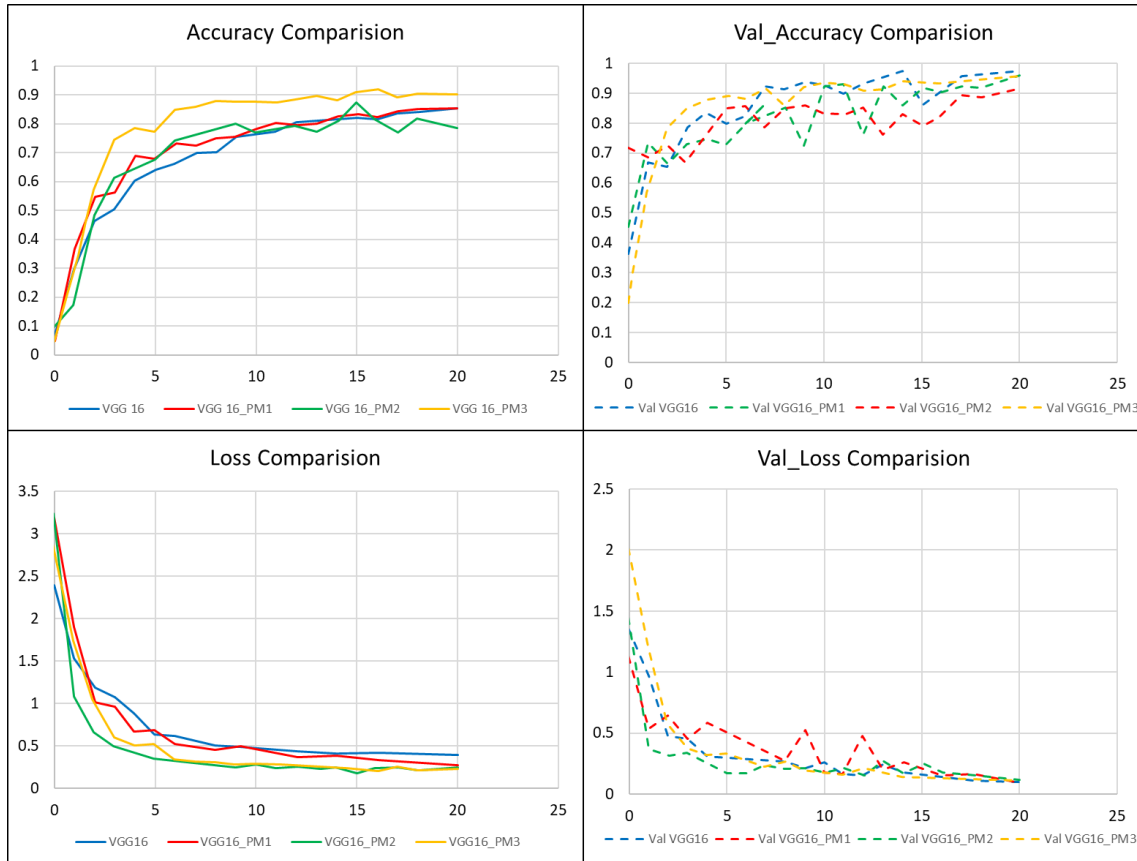


Figure 7. The accuracy and loss curves for both training and validation phases during the custom CNN training process.

In this figure, it was evident that the VGG16_PM3 model outperformed the other models across key metrics, demonstrating superior accuracy and reduced error rates. In particular, its validation accuracy began to increase and the model achieved an accuracy of 91.6 % at epochs 15. The training accuracy for each model demonstrated a steady increase throughout the 20 epochs. However, between epochs 15 and 20, no further improvement in accuracy was observed, leading to the training model stopped at epoch 20, indicating effective learning from the training dataset. Notably, the validation accuracy paralleled this trend, suggesting that the models not only fit the training data well but also generalized effectively to unseen examples.

Furthermore, the loss function curve at each epoch showed a significant decrease over time, approaching 0 after epoch 15. After that, the loss values stabilized until epoch 20, as illustrated in the two graphs in Figure 7. This trend supports the conclusion that the models progressively improved their predictive capabilities throughout training.

In addition to the comparison of models, including the VGG16 model and its variants VGG16_PM1, VGG16_PM2, and VGG16_PM3, training time, model size, parameter count, and accuracy were used as evaluation metrics, as illustrated in Figure 8.

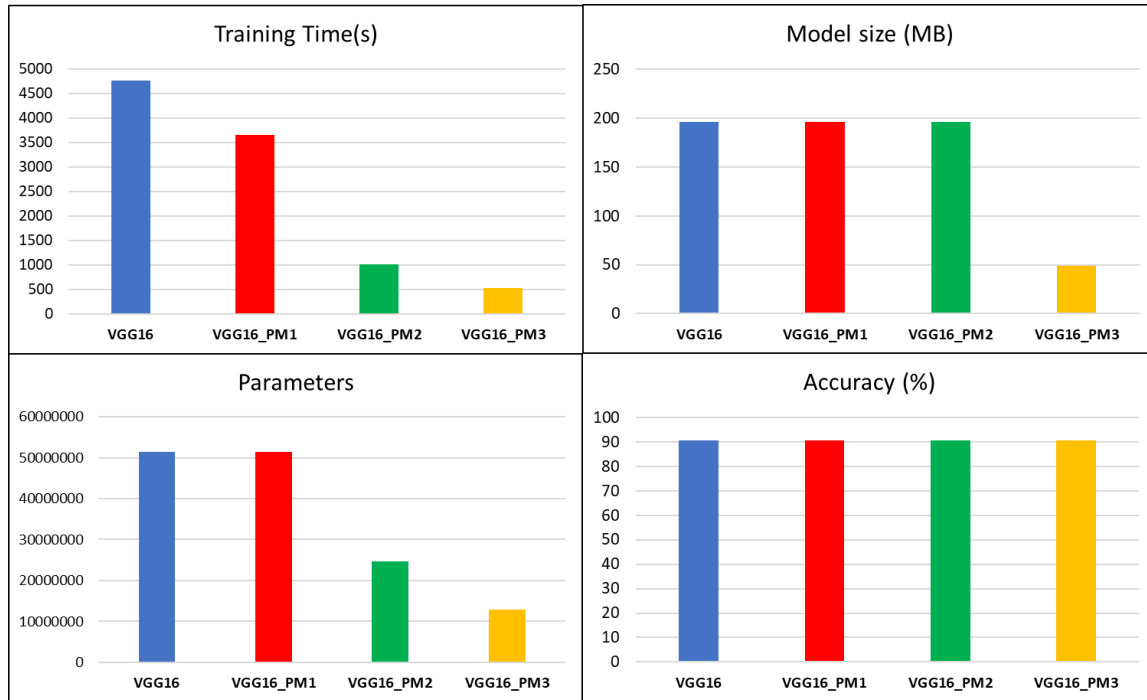


Figure 8. Comparison of models, including the VGG16 model and its pruned versions VGG16_PM1, VGG16_PM2, and VGG16_PM3 in terms of training time, model size, number of parameters, and accuracy.

The training times for the models showed considerable variation, as depicted in the upper left of Figure 8. The VGG16 model took 4,759 seconds to complete training, while its variants exhibited a notable reduction in training time: VGG16_PM1 required 3,645 seconds, VGG16_PM2 took 1,003 seconds, and VGG16_PM3 completed in 519 seconds. This marked decrease in training time for the pruned models suggested enhanced computational efficiency, resulting from architectural modifications and optimizations that facilitated faster convergence.

The parameter counts revealed significant differences among the models, as illustrated in the lower left of Figure 8. Both the VGG16 and VGG16_PM1 models contained approximately 51.4 million parameters, resulting in a model size of 196 MB. In contrast, VGG16_PM2 and VGG16_PM3 had substantially fewer parameters, with 24.7 million and 12.8 million, respectively, leading to model sizes of 196 MB for VGG16_PM2 and 48.9 MB for VGG16_PM3, as shown in the upper right of Figure 8. This reduction in parameters reflected a more streamlined architecture, potentially enhancing model efficiency and reducing the risk of overfitting.

All models achieved the same accuracy of 90.6%, as shown in the lower right of Figure 8. This consistency in performance across different architectures and parameter counts indicated that the modifications made in VGG16_PM2 and VGG16_PM3 did not adversely affect the models' ability to generalize to the validation dataset. The VGG16_PM3 model, as shown in Figures 7 and 8, exhibited superior performance compared to the other versions. Figure 9 presents the confusion matrix, which provided valuable insights into the model's ability to differentiate among

the 15 failure conditions, including various degrees of CIM, LEPV, and WFP, as well as one normal condition. The matrix also revealed several instances of misidentification across these conditions.

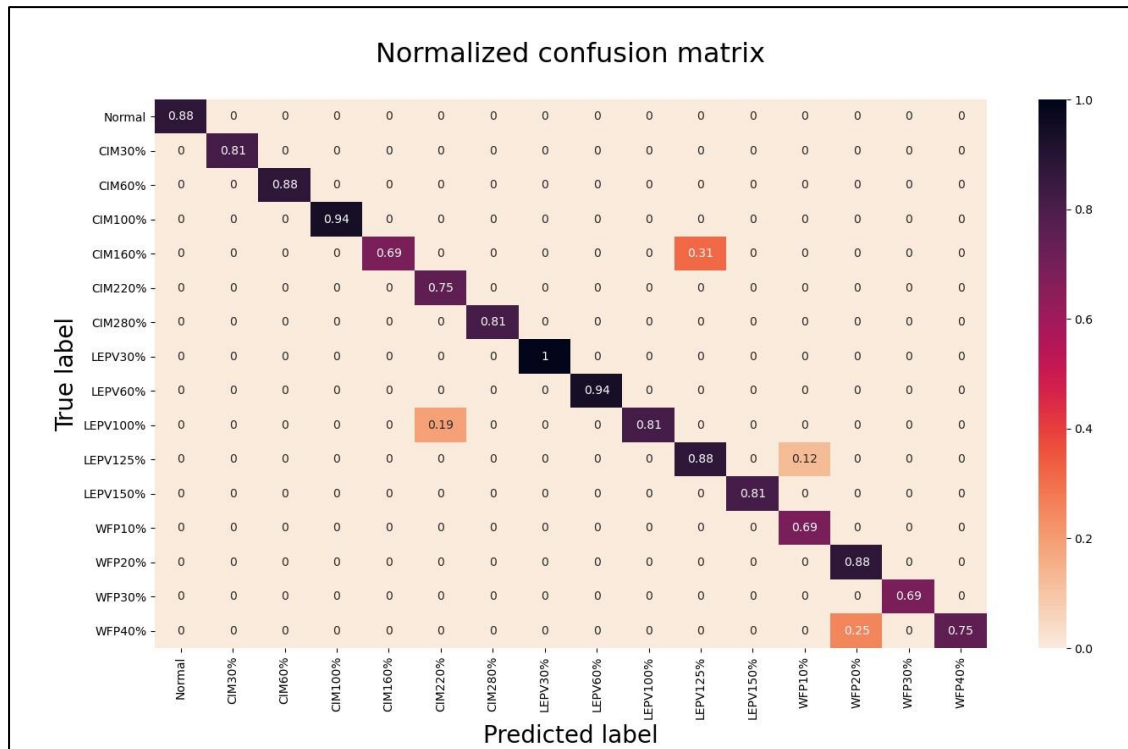


Figure 9. Confusion matrix displaying the classification performance of the VGG16_PM3 model on validation diagnostic images, highlighting its ability to distinguish between 15 failure conditions and a normal condition.

This indicated a need to incorporate more diagnostic images for each corresponding condition to enhance the model's performance. By expanding the dataset, the model could learn more effectively from a diverse range of examples, thereby improving its ability to accurately distinguish between similar conditions and reducing the rate of misclassification.

5. Conclusion

This study demonstrated the feasibility of using image classification model in troubleshooting electronic fuel injection systems for gasoline engines, especially mechanical causes. By expressing the features of engine live data collected at different operating states as a radar-chart-type diagnostic image, abnormal values of live data could be clearly displayed. The diagnostic images of mechanical causes such as cracked intake manifold, leaked evaporative purge valve, and weak fuel pump were effectively classified by using a pruned VGG16 deep-learning model with a precision of 0.85420 and a recall of 0.84400. This diagnostic concept could therefore be useful to develop automatic diagnostic tools as well as monitor the degradation of the fuel system for preventive maintenance.

Acknowledgements

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

References

- [1] Manger, H 1982 Electronic Fuel *Injection SAE Technical Paper* 820903
- [2] Oliveira L P d, Wehrmeister M A and Oliveira A d 2017 Systematic Literature Review on Automotive Diagnostics *VII Brazilian Symposium on Computing Systems Engineering (SBESC)*
- [3] Du D, Guo X and Guan Q 2008 A fault diagnostic method for EFI engine based on MATLAB software package *Proceedings - International Conference on Intelligent Computation Technology and Automation*
- [4] Tran K T, Cao D T, Duong D V, Pham Q T D and Tran L D 2024 A Quick Diagnostic Method for the Electronic Control System of the Mitsubishi 4A91 Engine *Proceedings of the International Conference on Sustainable Energy Technologies*
- [5] Mouzakitis A, Muniyappa N, Parker R and Puthiyapurayil S 2010 Advanced Automated Onboard Vehicle Diagnostics Testing *UKACC International Conference on Control*
- [6] Sharma N, Sharma R and Jindal N 2021 Machine Learning and Deep Learning Applications-A Vision *Global Transitions Proceedings* **2** 24–8
- [7] M Al-Zeyadi *et al* 2020 Deep Learning Towards Intelligent Vehicle Cause Diagnosis *International Joint Conference on Neural Networks*
- [8] Hong Guo, Crossman J A, Murphey Y L and Coleman M 2000 Automotive signal diagnostics using wavelets and machine learning *IEEE Transactions on Vehicular Technology* **49** 1650-1662
- [9] Khodadadi A, Ghandiparsi S and Chuah C-N 2022 A Natural Language Processing and deep learning-based model for automated vehicle diagnostics using free-text customer service reports *Machine Learning with Applications* **10** 100424
- [10] Gong C S A, Su C H S, Chen Y H and Guu D Y 2022 How to Implement Automotive Fault Diagnosis Using Artificial Intelligence Scheme *Micromachines* **13** 1380