IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 26, 2022, accepted February 10, 2022, date of publication February 24, 2022, date of current version March 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3153340

License Plate Detection Using Convolutional Neural Network–Back to the Basic With Design of Experiments

YANG YANG LEE[1], ZAINI ABDUL HALIM[1], (Member, IEEE), AND MOHD NADHIR AB WAHAB[2], (Member, IEEE)

[1]School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Nibong Tebal, Penang 14300, Malaysia
[2]School of Computer Sciences, Universiti Sains Malaysia (USM), Gelugor, Penang 11800, Malaysia

Corresponding author: Zaini Abdul Halim (zaini@usm.my)

This work was supported in part by the Universiti Sains Malaysia under Grant RUI 1001/PELECT/8014152.

ABSTRACT Automatic License Plate Recognition (ALPR) is one of the applications that hugely benefited from Convolutional Neural Network (CNN) processing which has become the mainstream processing method for complex data. Many ALPR research proposed new CNN model designs and post-processing methods with various levels of performances in ALPR. However, good performing models such as YOLOv3 and SSD in more general object detection and recognition tasks could be effectively transferred to the license plate detection application with a small effort in model tuning. This paper focuses on the design of experiment (DOE) of training parameters in transferring YOLOv3 model design and optimising the training specifically for license plate detection tasks. The parameters are categorised to reduce the DOE run requirements while gaining insights on the YOLOv3 parameter interactions other than seeking optimised train settings. The result shows that the DOE effectively improve the YOLOv3 model to fit the vehicle license plate detection task.

INDEX TERMS Convolutional neural network, design of experiments, license plate detection.

I. INTRODUCTION

Automatic License Plate Recognition (ALPR) has been an active field of research in computer vision applications. With the emerging Machine Learning (ML) method, specifically Convolutional Neural Network (CNN), ALPR has become much more robust and reliable than traditional hard-coded image processing techniques. Recent innovations and research focus on real-time CNN inferencing benefited ALPR applications, such as YOLOv3 [1] and SSD [2] methods. Meanwhile, much ALPR research focuses on custom CNN models or post-processing methods to tackle different ALPR problems by the format or geo-specific conditions of license plate (LP). ALPR can be classified into LP detection and character recognition, each with application implementation challenges. In the LP recognition task, the characters on the LP could have different languages, such as Arabic and Chinese characters, or other formats such as Italic or non-standard fonts. Meanwhile, for LP detection (also known as localisation), vehicle LPs could have different sizes, shapes, orientations, conditions, and colours. The process is similar to You-Only-Look-Once (YOLO) or Single-Shot detector (SSD) algorithm that improves CNN localisation performance and could be effectively transferred to the ALPR task with some efforts on data engineering.

A complete ALPR system relies on both LP detection and character recognition, with some works only focusing on the LP detection stage. Newer research attempts to eliminate the traditional cascaded processing, i.e. LP detection then character recognition. A CNN model for achieving one-pass end-to-end LP detection and character recognition is favourable for real-time processing. There are several performance metrics in the ALPR task. Average precision (AP) is the primary interest for bounding box regression in LP detection to localise the LP. Character recognition performance is based on the usual accuracy or recall metrics by comparing the ground truth LP labels. The algorithm's execution time is another comparable performance metric for real-time processing, but

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wang.

it is highly dependent on the computing hardware. Many ALPR techniques are not precisely apple-to-apple comparison due to geo-specific LP datasets, not to mention the variety of proposed CNN models and processing techniques further limit the comparable metrics. Despite the major innovations in the ML model specifically for the ALPR task, no one has attempted to investigate the effect of the ML model training parameters, which could play a significant role in the ALPR performance itself.

This article intends to bring the ALPR research back to the basics with Design of Experiments (DOE) by understanding the training parameters' correlation and optimising the training in the DOE process. YOLOv3 algorithm would be utilised, and the train parameters would be studied for the LP detection task without modifying the backbone CNN model. It is shown that the DOE effectively tunes the YOLOv3 algorithm to fit the LP detection task across a wide variety of LP conditions.

II. RELATED WORKS

A. TRANSITION OF ALPR TO DEEP LEARNING ALGORITHM

The early day ALPRs are mostly on hand-crafted algorithms. Image processing techniques such as edge detection [3] and coefficient correlation [4] were common, as well as ML algorithms such as k-nearest neighbour [5], sparse autoencoder method [6], support vector machine (SVM) and artificial neural network (ANN) [7]. The paradigm had shifted when a subset of ML algorithms, i.e. CNN deep learning, started performing and CNN computation became more viable. Unlike ANN, CNN can process multi-dimensional data such as images. Initial findings from [6] concluded that the accuracy would improve with more train data since CNN is a data-driven algorithm that differs from traditional hand-crafted coding. Then, [8] attempted to recognise LPs and its characters with single CNN by retraining AlexNet, which is one of the most popular CNN algorithms in 2017. They trained the CNN with custom cropped images of car LPs and achieved 95.24% of accuracy. Another similar research also uses CNN-based character classification to replace traditional OCR, proving that CNN can classify characters from blurry LP images[9]. Vehicle ALPR research in [10] massively deployed deep learning algorithms with CNN and long-short term memory (LSTM). CNN was used to extract features, then LSTM were trained to process the features to recognise the characters. They can discriminate both private and public car plates of different colours and recognise the characters. Similarly, [11] uses simpler parallel CNNs to identify the nature of the car plate, such as types, dimensions and colour, then used LSTM to recognise the car plate characters, achieving 99.8% of precision. Reference [12] combined both edge detection and CNN as hybrid processing pipelines to enhance ALPR performance.

It is realised that although CNN is superior in performance, the cost of computing is prohibitive in a real-world scenario. Image sizes are limited to the CNNs input size. Its classification performance relies on cropped image batches. However, it does not translate into the ability to localise and identify car plates in a huge image area which is the more practical use case in ALPR application. Thus, [13] tried to speed up the recognition process with regional CNN (R-CNN) but achieved a precision of 0.4 out of 1 on a single huge image, primarily due to R-CNN's limitations. Newer research from [14] also showed that a much better variation of R-CNN called masked R-CNN is capable of a complete ALPR task at comparable 98% precision and recall.

Data processing pipeline unification has been done by [15] to fully utilise CNN to detect and recognise the LP characters, bypassing any unnecessary architectures for different tasks but showing that CNN favours detection tasks but not character recognition. Another research exploited big data (about 250k images), namely Chinese City Parking Dataset (CCPD) with one pass CNN much like SSD to recognise and localise the LP. It is proven to be effective and robust for various environments (blurry, angled, tilted LP), avoiding recurrent CNN computation like R-CNN, which is the reason for the high computing cost for CNN inferencing [16].

YOLO algorithm has been the interest for ALPR in recent years. YOLO algorithm was introduced by [17] in 2015 and achieved one-pass CNN object classification and localisation. Further revisions of YOLO improve the detection capability and speed. The first use of YOLO CNN was attempted by [18] to detect LPs of vastly different plate orientations, yielding 99.5% F1-score. YOLOv2 algorithm with modified ResNet50 CNN was proposed by [19] to localise and detect the nature of multi-national LP (country, size, and languages but did not work on recognising the characters on LPs), achieving 99.57% detection precision. Reference [20] also used YOLOv2 because they claimed YOLOv3 has more layers that slow down the training, which is not entirely true depending on which CNN model to utilise in the YOLO workflow. They only compared the metric of motorcyclist LP of riders with or without helmets, in which the comparison might not be significant. Nevertheless, they also achieved 95 to 97.5% precision score with YOLOv2 algorithm. Similarly, [19] extended the dataset for multi-national and multi-language LP, reaching 99.57% of AP in LP detection. Reference [20] further enhance datasets by synthesising LPs to overcome small dataset size and train a custom CNN model ported to Fast-YOLO to perform ALPR. Reference [21] utilised YOLOv3 for both LP detection and recognition stages with 95-97% accuracy. Overall, the YOLO-based algorithm is very promising to be repurposed for LP detection. Another work from [22] is very similar to YOLO, but they used a branching method at the end of CNN to detect LPs.

Besides the YOLO algorithm, CNN could also do image segmentation with up-sampling layers, giving [23] an idea of using an entirely semantic segmentation method to detect and recognise the LP. Their work is specifically on Arabic LP, so it is hard to compare to the CCPD dataset. Reference [23] also uses the CNN segmentation method to extract features and perform complete ALPR with parallel CNNs.

Some researchers customised CNN for a particular function instead of a full ALPR task. Reference [24] used CNN to predict the originated states of the vehicle by LPs. R-CNN was also used for customised ALPR, such as detecting LP on non-motor illegal vehicles [25].

The importance of clean data in the CNN application could not be ignored when [11] combined traditional image processing techniques to filter out unnecessary noises and used CNN at the final stage of car plate recognition, achieving 99.6% accuracy. With that acknowledgement, [26] identified that rain streaks might be one of the big problems of ALPR in a real environment. Thus, they first pre-processed images of noisy rain streak with dictionary learning, then only processed the vehicle LP with CNN.

B. THE NATURE OF DATA-DRIVEN ALPR

Deep learning is one of the data-driven programming approaches. Instead of a hard-coded feature extraction algorithm, feeding as much data as possible will ''code'' the necessary feature map. Thus, the reliance on big data is one of the key factors for ML implementation. CNN model architecture plays a vital role in ML, but it heavily depends on the methods or preference of the data processing pipeline.

The ALPR approach could be classified into one-staged and two-staged processes. The two-staged process is more straightforward because the data classes are separated, i.e. LP itself and its characters in optimising the coding for each data processing pipeline. This approach is especially true from traditional image processing, where images are considered complex data. A two-staged process usually detects and crops the region of interest (ROI) around LP to eliminate any unwanted background details, then only attempt to recognise the characters on the cropped LP images with optical character recognition (OCR). OCR could be based on hard-coded algorithms such as connected component analysis, local binary pattern, temporal matching [27], or CNN-based classification. However, the image processing pipeline of ALPR had been shifting to a one-staged process with the emergence of ML, extracting both LP and its character in one pass. One pass processing is possible with some innovations in CNN model designs and post-processing techniques. In the one-staged process, the CNN model mostly only acts as a feature extractor to preserve the spatial information of the object of interest, i.e. LP and its characters in ALPR. The classification and bounding box localisation are passed to other post-processing techniques such as non-max suppression (NMS) and intersection over union (IoU) to compute the confidence level and the ROI of the object class within an image.

C. THE CHALLENGES OF MALAYSIAN LP

Several ALPR works on Malaysian's LP exist, but none are up to the global trend of ML-based ALPR. There are some unique challenges to implementing ALPR on Malaysian LP. First is the availability of the dataset because there are no known open-source LP images for Malaysian vehicles. The LP images are confidential or owned by specific authorities, which could not be accessed easily and openly. Secondly is the inconsistency of the LP format. Many on-road Malaysian LP characters could have different fonts, spacing and placements, even with non-standard stickers or labels, violating the official LP guideline. There also exist many valid LPs with unique characters such as ''XXIV'', ''SUKOM'', ''1M4U'' and Putrajaya''. Newer LPs also located characters after the numbering with increasing new on-road vehicles. Those varying LP standards render most overseas ALPR techniques inapplicable because foreign LPs have fixed character numbers and spacing, suitable for processing with the character segmentation method.

III. METHODOLOGY

Many ALPR algorithms had been proposed in previous research, but they hardly discussed the relationships and the reasoning of the related training parameters. In this work, multi-level (2-level and 3-level) factorial DOE would be utilised to study the YOLOv3 training parameters' interactions and optimise the LP detection performance for stage one of ALPR, i.e. LP detection only. Self-prepared Malaysia's vehicle LP dataset will be used since this research has to tackle ALPR problems on Malaysian vehicles specifically. Stage two of ALPR, i.e. character recognition, will not be part of the research for the time being because LP labels are geo-specific and highly dependent on dataset labelling and algorithms.

A. MALAYSIAN VEHICLE LP DATASET

The datasets are obtained by several methods. One is taking photos from the federal highway, which consists of multiple vehicles in a single 32MP image with a DSLR telephoto lens. Some photos are taken with hand-held cameras or smartphone cameras; thus, the photos have mixed sources of sensor noises and qualities. Images with clear local LP labels were also downloaded from social websites and some local car auction websites. Some of the downloaded images will have watermarks and were eliminated by manually cropping the ROI of the images. Overall, a total of 10k images were manually collected, processed and labelled. Only the LP spatial locations are labelled for (x,y,h,w), where 'x', 'y', 'w', and 'h' are the horizontal and vertical locations of bounding box centre, width, and height of the LP bounding box, respectively, as illustrated in Figure 1. The LP characters are simply recorded, and no character-wise bounding box labelling had been done. The (x,y,w,h) values are in a ratio relative to the image size so that the images can be resized without affecting the relative location of the bounding boxes. The relative point value could be converted to pixel value before feeding the label to the YOLOv3 algorithm.

The cropped images of Malaysian LP are of approximate square shape due to manual cropping, and the height and width of the cropped images are slightly inconsistent. Meanwhile, the YOLOv3 pre-processing is designed to feed images of video format aspect ratio where the image width

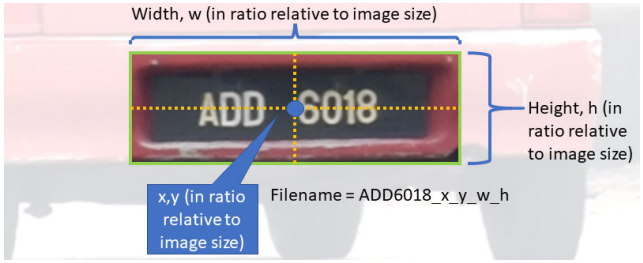

FIGURE 1. The labelling convention.

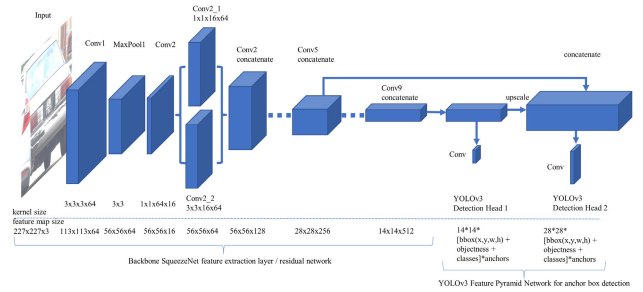

FIGURE 2. YOLOv3 structure with "SqueezeNet" backbone CNN.

is bigger than its height. The Malaysia LP images have to be resized to fit into the existing process pipeline, but that would result in feature loss. Thus, the square images were resized minimally so that the image height is slightly smaller than its width to minimise feature loss but still able to be fed to the YOLOv3 algorithm.

B. THE YOLOv3 PARAMETERS

YOLOv3 is a state-of-art real-time object recognition algorithm that could accept various CNN model designs as the backbone feature extractor (with few CNN backend requirements). The principal technique of YOLOv3 is on the backend feature pyramid network (FPN)[28] and anchor box layer as a post-processing pipeline to retrieve spatial information and class confidence from the extracted features. The FPN extracts the spatial information of the convoluted feature maps at different scales for multiscale object detection, and the output consists of anchor box features. The output will be post-processed with IoU and NMS operations to resolve object bounding boxes. The backbone CNN model is "SqueezeNet", originally proposed by [29] and remains untouched to isolate the parameters specific to the CNN model. The overall structure of the "SqueezeNet" based YOLOv3 algorithm is illustrated in Figure 2. YOLOv3 does have a few training parameters with default values to be adjusted in the MATLAB native code, as shown in Table 1. Some parameter values are limited to the original example datasets and not strictly tied to the YOLOv3 algorithm.

C. MULTI-LEVEL FACTORIAL DOE

The purpose of the DOE is to study the correlation of the training parameters and their effects on the YOLOv3 LP detection performance. The DOE was performed at 2-level and 3-level factorials, whereby the 2-levels factorial DOE is applied at the initial DOE, and the 3-level factorial design is applied on the subsequent DOEs. The choice of level number would be explained along with the experiments in Section IV. Multi-level factorial DOE design follows a function, as in

$$run = level^{factor} \tag{1}$$

One consideration of DOE on CNN is that a complete epoch of CNN training could take minutes or even hours depending on the CNN algorithms complexity and computing hardware. The number of runs increases exponentially with the number of levels and factors. There are nine possible parameters for the YOLOv3 training that contribute to nine factors, referring to (1), a 2-level full factorial DOE would require 512 runs, or 19683 runs in the case of 3-level full factorial, which is impractical. Thus, reducing either the number of levels or factors is necessary.

Several research papers performed DOE on tuning CNN model parameters, albeit not specifically on ALPR tasks. Reference [30] utilised a new class of 3-level definitive screening design (DSD) proposed by [31] to tackle many factors to identify significant main effects while estimating some of the interaction effects. Standard 2-level fractional factorial design is also used by [32] to reduce the number of runs to optimise the CNN model. However, it is unknown whether the residual data point is normally distributed given such a small 16 runs on seven factors for the analysis of variance (ANOVA) to be valid.

Fortunately, the training parameters could be partitioned into two categories, data-specific and training-specific, as shown in Table 2. Data-specific variables change with data and label size, whilst training-specific variables will manipulate the CNN training behaviours. Partitioning the parameters for DOE would reduce a considerable number of runs. Instead

TABLE 1. List of parameters For Yolov3 training.

Parameters	Default value	Description
imageAspectRatio	4:3	The ratio of image width over image height.
numberofAnchor	6	The number of preset anchors for the bounding box on the predictor map.
trainTestRatio	0.6	The number of images being split for training and testing, e.g. 0.6 = 60% for training, 40% for testing.
numberofEpochs	80	The number of complete feedforward and backpropagation training loops.
miniBatchSize	8	The number of images per batch during feedforward training.
warmupPeriod	0.416	Time to max out learn rate exponentially, e.g. 0.3 = 30% of total epochs to reach target learn rate.
penaltyThreshold	0.5	Minimum detection overlapping threshold before applying penalty to the network.
L2Regullarization	0.0005	Unitless weight update gradient regularisation factor.
learnRate	0.001	The magnitude of updating CNN's trainable parameters.

TABLE 2. Parameters categorisation.

Data-specific	Training-specific
aspectRatio	numberofEpochs
numberofAnchor	miniBatchSize
trainTestRatio	warmupPeriod
	penaltyThreshold
	L2Regullarization
	learnRate

TABLE 3. DOE outline.

DOE		I	II	III
Alias	A	imageAspectRatio	miniBatchSize	trainTestRatio
	B	numberofAnchor	warmupPeriod	L2Regularization
	C	trainTestRatio	penaltyThreshold	learnRate

of nine factors at once, it would be more practical to isolate a few factors once at a time.

The DOEs are rather an iterative process. The first experiment will be executed on data-specific parameters since it only has three variables. Then, the best performing data-specific settings would be transferred for the subsequent experiments to eliminate the data-specific factors. Even so, there are six training-specific parameters, which might result in the long run. Thus, the next DOEs were designed iteratively, i.e. examining only three parameters at a time in the hope of discovering the interaction of the parameters and optimizing a few of them at a time. The factors and aliases for each DOE are listed in Table 3. Some parameters could be sensitive to value changes and cause overfitting, failing the CNN epoch training. The parameter operating ranges is further described for each DOE in Section IV.

The DSD proposed by [31] seems to fit the experiment requirements, but the interaction of parameters is also of interest in this research. Also, the ''numberofEpochs'' might not be the interest which will be explained in Section IV part A, resulting in a total of five factors only. Thus, DSD is not used. There is another D-optimal Designs [33] technique to reduce the DOE runs, but it is not in the scope of this article.

IV. EXPERIMENTS

A. DOE I

DOE I is a 2-level full factorial experiment with three factors and two replicates, resulting in a total of 16 runs. A complete 2-level factorial experiment only requires eight runs, but the extra loop ensures the normal distribution of the data point. Only then the scores are valid for the ANOVA. Three factors were tested, i.e. image aspect ratio, number of anchors and the train-test ratio. The settings of each factor, their aliases, and the default values of the other parameters for DOE I are listed in Table 4, whilst its ANOVA and interaction plot output are shown in Figure 3.

The default number of epochs was 80, but it was reduced to 10 for all DOEs. The purpose of the DOE is to study the interaction of the parameters and their relative convergence capability. It also helps minimise the CNN training

TABLE 4. Doe I list of parameters.

Alias	Parameters	Low (-1)	High (1)
A	imageAspectRatio	4:3	1:0.98
B	numberofAnchor	6	12
C	trainTestRatio	0.6	0.7

DOE I settings		Default settings	
Factor	3	numberofEpochs	10
Replicates	2	miniBatchSize	16
Total run	16	warmupPeriod	0.41
		penaltyThreshold	0.5
		L2Regularization	0.005
		learnRate	0.001

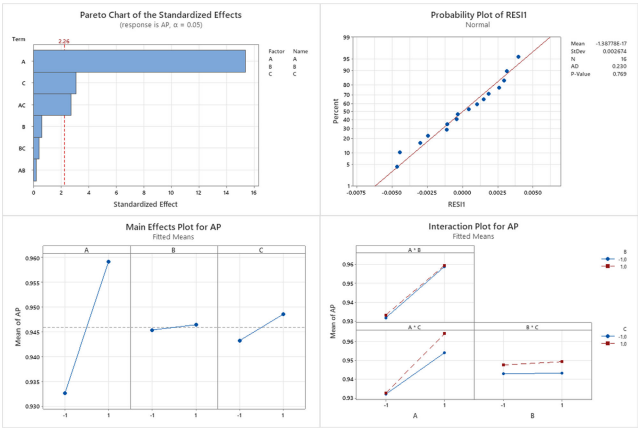

FIGURE 3. DOE I result of analysis.

time to have a faster design cycle for DOE since it takes several minutes to complete an epoch. Thus, a comprehensive CNN training epoch is not required. However, a complete 80 epochs of training would be carried out at the end of all DOEs to validate the DOE findings. ''imageAspectRatio'' is limited to near square ratio as described in Section III part A. More ''numberofAnchors'' could improve the mean intersection union of the localization, thus improving the AP for more object classes. Hence, its upper limit is set to double the lower limit. ''trainTestRatio'' is dependent on the size of the supplied dataset. A bigger dataset could allocate more data for training. A 70% training ratio is general for most ML approaches.

It is found that the image aspect ratio plays a significant role in performance outcomes. One apparent reason is that the images of 4:3 aspect ratio have fewer LP-based features after image resizing. Similarly, near-square images have relatively more pixel-level features than the one of 4:3 aspect ratio, better utilising the CNN feature map for performance convergence. A higher train-test ratio also contributes to higher performance as more images are available for training, which explains the slight interaction between the image aspect ratio and the train test ratio. Unlike the traditional CNN classification task, the YOLO algorithm does not require a validation set since the CNN only does feature extraction and does not directly classify the object. The number of anchors is statistically insignificant in the LP detection task because

TABLE 5. Doe II list of parameters.

Alias	Parameters	Low (1)	Mid (2)	High (3)
A	miniBatchSize	8	16	32
B	warmupPeriod	0.15	0.30	0.45
C	penaltyThreshold	0.5	0.6	0.7

DOE II settings		Default settings	
Factor	3	numberofEpochs	10
Replicates	1	imageAspectRatio	1:0.98
Total run	27	trainTestRatio	0.7
		numbeofAnchor	6
		L2Regularization	0.005
		learnRate	0.001

Malaysian LP only has a few possible shapes and sizes. LP is the only class of interest in the ALPR.

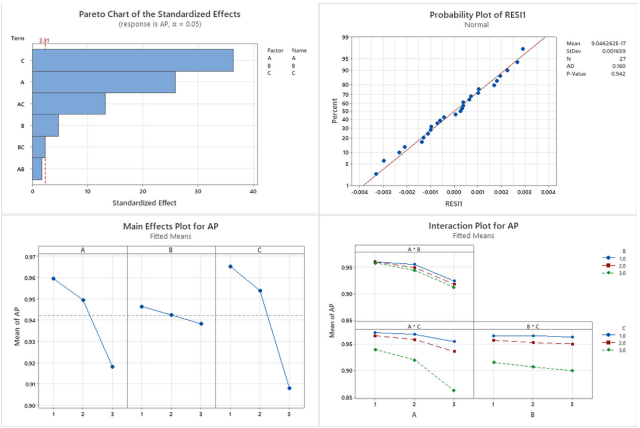

FIGURE 4. DOE II result of analysis.

B. DOE II

DOE II is a 3-level full factorial experiment with three factors and one replicates, resulting in 27 runs. It is found that 3-level factorial could provide more insight into the interactions of the factors and normally distributed data points with just one replicate. An extra 11 runs compared to the previous DOE I is a good trade-off for having additional information for the 3-level factorial interaction plots while ensuring normal data distribution for ANOVA outputs to be valid. Since the previous experiment provided an insight into the parameter's interaction, it will not be redone as a 3-level factorial DOE. The settings of each factor, their aliases, and the default values of the other parameters for DOE II are listed in Table 5, whilst its ANOVA and interaction plot output is shown in Figure 4. The best parameter settings from DOE I were utilised in this DOE II, i.e. ''imageAspectRatio'', ''numberofAnchor'' and ''trainTestRatio'' are 1:0.98, 6 and 0.7, respectively.

The ''miniBatchSize'' is set as 16 and 32 since the computing memory is the only limiting factor. Although the original value for ''warmupPeriod'' is 41.6% of the total epochs, a lower value will have a faster ramp to the target learning rate, thus having better initial convergence but might risk CNN overfitting. The ''penaltyThreshold'' is a ceiling for applying penalty function to the CNN model. A higher threshold will improve the object detection confidence score but decrease the anchor box detection overlapping tolerance, lowering the AP.

One common assumption for CNN training is that image batch size could be larger with a higher learning rate given enough hardware memory space on a computer. The change of learning rate is indirectly adjusted by the warm-up period. A lower warm-up period will result in a faster learning rate increment. According to the factorial plots, the interaction of both ''miniBatchSize'' and ''warmupPeriod'' has no significance to the overall AP. Or rather, the batch size itself has a significant effect, whereby it is the number of images to be fed forward to the CNN model in every feedforward training. After each feedforward will have a loss gradient update, in which the loss function tries to converge the CNN weight. A smaller batch size could achieve higher AP because it could have better convergence to the local minima.

Conversely, a higher number of image batch could result in excessive feature generalisation, converging to global minima only. Large-batch methods tend to converge to sharp minimizers of the training and testing functions, and sharp minima lead to poorer generalisation [34]. The ''warmupPeriod'' have some significance according to the Pareto chart in Fig. 4, but the loss curve shows that it has the risk of overfitting for a lower value. It had been pushed to the value of 0.01 but rolled back to 0.15 because overfitting occurred, i.e. the loss curve increases although it is supposed to converge to zero. It is important to generalise the newly initialized CNN feature map with a slower learning rate at the initial stage before pushing for a higher learning rate for a faster CNN model convergence. The ''penaltyThreshold'' is shown to have the most significant influence on the AP score, although it has no interaction with other parameters.

C. DOE III

DOE III is also a 3-level full factorial experiment with three factors and one replicates, resulting in 27 runs. From the DOE I, it is found that increasing in train-test ratio could have statistical significance to the performance outcome. However, it is still possible to push the ratio to 80% instead of limited to 70%. Also, it is unknown whether the train-test ratio has any correlation with the L2Regularization and learning rate. Thus, the train-test ratio was included in this DOE to extend its behaviour study to 3-level factorial. The settings of each factor, their aliases, and the default values of the other parameters for DOE III are listed in Table 6, whilst its ANOVA and interaction plot outputs are shown in Figure 5.

The ''L2Regualrization'' is the magnitude for weight update gradient, modifying the weight update rate. Whilst the ''learnRate'' is the global multiplier for updating the CNN trainable parameters. CNN training is sensitive to ''L2Regualrization'' and ''learnRate'' values, so they are only adjusted in small margins.

TABLE 6. Doe III list of parameters.

Alias	Parameters	Low (1)	Mid (2)	High (3)
A	trainTestRatio	0.6	0.7	0.8
B	L2Regularization	0.0002	0.00035	0.0005
C	learnRate	0.0008	0.001	0.0012

DOE III settings		Default settings	
Factor	3	numberofEpochs	10
Replicates	1	imageAspectRatio	1:0.98
Total run	27	numbeofAnchor	6
		miniBatchSize	8
		warmupPeriod	0.15
		penaltyThreshold	0.5

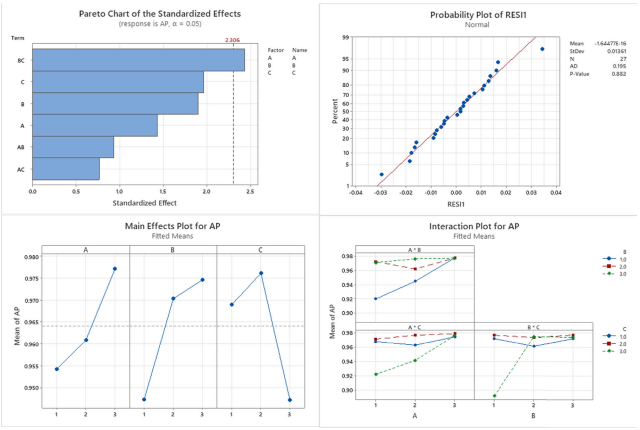

FIGURE 5. DOE III result of analysis.

TABLE 7. Final test.

Parameters	Setting				
	A	B	C	D	E
trainTestRatio	0.6	0.7	0.8	0.6	0.8
numberofEpochs	80	80	80	80	80
miniBatchSize	16	32	32	8	8
warmupPeriod	0.45	0.3	0.3	0.416	0.15
penaltyThreshold	0.7	0.6	0.5	0.5	0.5
L2Regularization	0.00035	0.0002	0.0005	0.0005	0.0005
learnRate	0.0008	0.0012	0.001	0.001	0.001
3-fold AP	0.9688	0.9812	0.9547	0.9853	0.9900

The interaction plot B∗C shows that they have strong interaction. The ''learnRate'' is favoured at the value of 0.001, and a lower value would result in a performance drop. Whilst ''L2Regularization'' is best at the value of 0.0005. Higher ''L2Regualrization'' than the value listed in Table 6 had caused random train failure because a higher value could have a bigger weight update gradient, indirectly leading to CNN overfitting. Also, higher ''learnRate'' with low ''L2Regularization'' adversely caused a significant AP reduction, which explains the strong interaction outcome. Higher ''trainTestRatio'' has been shown to contribute to higher AP even though it has almost no interaction with ''L2Regualrization'' and ''learnRate''.

D. FINAL TEST

The purpose of the final test is to validate the findings of the DOE. A full 80 epochs were executed on the YOLOv3 algorithm to validate whether the DOE effectively tunes the performance outcome. There are five tests. Setting E is of optimum settings from the DOEs, while Setting D is of original parameters before the DOEs. The rest of Setting A, B and C are non-optimum settings. Each test is repeated for 3-fold cross-validation so that the tests are less likely to be data-dependent. The average of three runs for each set was taken as the final score of each test. In the end, the results are listed in Table 7. Figure 6 shows the precision-recall curve of each setting for the first-fold run to compare the fitness of the YOLOv3 to the Malaysian LP dataset. Some output LP detection samples are shown in Figure 7.

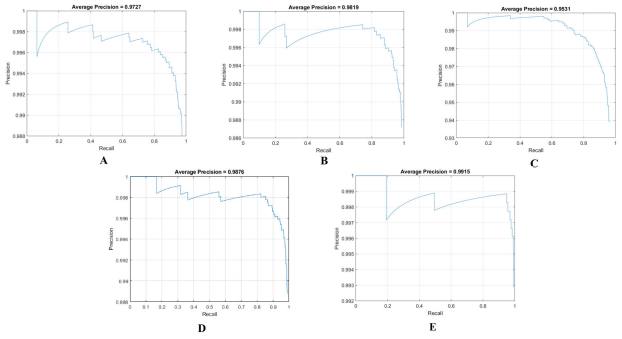

FIGURE 6. The precision-recall curve of final test result (first fold run of all settings).

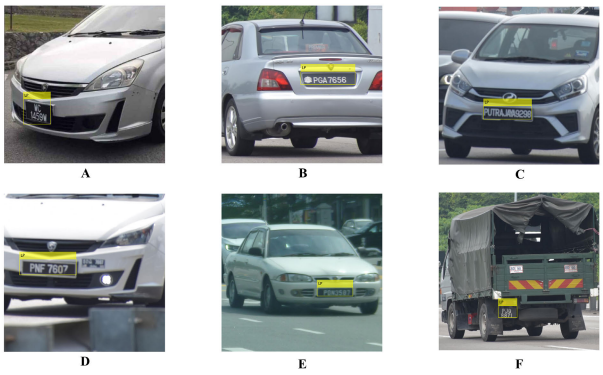

FIGURE 7. Examples of YOLOv3 Malaysian LP detection. A) Tilted LP detection. B) LP with a non-standard sticker. C) Unique legal LP initials. D) Double LPs, the smaller LP belongs to neighbour Thailand country. E) LP detection on blurry images. F) LP detection on large vehicles such as lorries.

The DOE optimised Setting E delivered the highest 99.00% AP score while the default Setting D has 98.53% AP, or 0.47% improvement only because the train test ratio and the warm-up time are the only difference. Other settings changes were relatively underperformed, ranging from 95.47% to 98.12% AP scores. The precision-recall curve also shows that the optimised setting has the best average curve fitness of all settings with minimum precision of 0.993. It shows that the DOEs tuned the performance of the YOLOv3 to detect the LP location. However, the setting could be specific

to the Malaysian LP dataset. Other open datasets like CCPD and UFPR are yet to be tested.

SSD is an alternative algorithm for ALFR, but it has a poorer performance than YOLOv3 from initial training with the default setting, only achieving 87.75% AP in addition to ten times longer training epochs. Also, SSD utilised RestNet-50 CNN model, so it is not a fair comparison. SSD has some different classes of parameters and might require different DOE strategies.

V. CONCLUSION

A series of simple DOEs had been shown to improve the ALPR performance of YOLOv3 with the CNN model untouched, specifically on the LP detection task. An AP of 99% is achieved for Malaysian vehicle LP detection by strategically tuning the YOLOv3 training parameters. A minor change relative to the stock parameters did improve the performance. Adjusting other settings in the DOEs also provide insights into the interactions of the YOLOv3 parameters. Images with more pixel areas are generally better because more features are available for CNN feature extraction. It is also found that a smaller mini-batch size has a better fitting to local minima, improving the overall AP. The warm-up period is useful in generalising the initial feature map across all image batches before increasing the global learning rate, but a longer learning rate ramp-up time will decrease the overall AP.

REFERENCES

[1] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 9905, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[3] O. Khalifa, S. Khan, R. Islam, and S. Ahmad, "Malaysia vehicle license plate recognition," *Int. Arab J. Inf. Technol.*, vol. 4, no. 4, pp. 359–364, 2007.

[4] N. Simin and F. C. C. Mei, "Automatic car-plate detection and recognition system," in *Proc. EURECA*, 2013, pp. 113–114.

[5] C. K. Soon, K. C. Lin, C. Y. Jeng, and S. A. Suandi, "Malaysian car number plate detection and recognition system," *Austral. J. Basic Appl. Sci.*, vol. 6, no. 3, pp. 49–59, 2012.

[6] R. Yang, H. Yin, and X. Chen, "License plate detection based on sparse auto-encoder," in *Proc. 8th Int. Symp. Comput. Intell. Design (ISC)*, vol. 2, 2016, pp. 465–469, doi: 10.1109/ISCID.2015.151.

[7] A. Menon and B. Omman, "Detection and recognition of multiple license plate from still images," in *Proc. Int. Conf. Circuits Syst. Digit. Enterprise Technol. (ICCSDET)*, Dec. 2018, pp. 1–5, doi: 10.1109/ICCSDET.2018.8821138.

[8] S. Lee, K. Son, H. Kim, and J. Park, "Car plate recognition based on CNN using embedded system with GPU," in *Proc. 10th Int. Conf. Hum. Syst. Interact. (HSI)*, Jul. 2017, pp. 239–241, doi: 10.1109/HSI.2017.8005037.

[9] J. Spanhel, J. Sochor, R. Juranek, A. Herout, L. Marsik, and P. Zemcik, "Holistic recognition of low quality license plates by CNN using track annotated data," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, doi: 10.1109/AVSS.2017.8078501.

[10] P. Shivakumara, D. Tang, M. Asadzadehkaljahi, T. Lu, U. Pal, and M. H. Anisi, "CNN-RNN based method for license plate recognition," *CAAI Trans. Intell. Technol.*, vol. 3, no. 3, pp. 169–175, 2018, doi: 10.1049/trit.2018.1015.

[11] W. Wang, J. Yang, M. Chen, and P. Wang, "A light CNN for end-to-end car license plates detection and recognition," *IEEE Access*, vol. 7, pp. 173875–173883, 2019, doi: 10.1109/ACCESS.2019.2956357.

[12] P. Dhar, S. Guha, T. Biswas, and M. Z. Abedin, "A system design for license plate recognition by using edge detection and convolution neural network," in *Proc. Int. Conf. Comput., Commun., Chem., Mater. Electron. Eng. (ICME)*, Feb. 2018, pp. 3–6, doi: 10.1109/IC4ME2.2018.8465630.

[13] Z.-K. Huang and L.-Y. Hou, "Chinese license plate detection based on deep neural network," in *Proc. Int. Conf. Control Robots (ICCR)*, Sep. 2018, pp. 84–88, doi: 10.1109/ICCR.2018.8534484.

[14] Z. Selmi, M. B. Halima, U. Pal, and M. A. Alimi, "DELP-DAR system for license plate detection and recognition," *Pattern Recognit. Lett.*, vol. 129, pp. 213–223, Jan. 2020, doi: 10.1016/j.patrec.2019.11.007.

[15] T.-N. Nguyen and D.-D. Nguyen, "A new convolutional architecture for Vietnamese car plate recognition," in *Proc. 10th Int. Conf. Knowl. Syst. Eng. (KSE)*, Nov. 2018, pp. 7–12, doi: 10.1109/KSE.2018.8573375.

[16] Z. Xu, W. Yang, A. Meng, N. Lu, H. Huang, C. Ying, and L. Huang, "Towards end-to-end license plate detection and recognition: A large dataset and baseline," in *Proc. Eur. Conf. Comput. Vis.*, Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 11217, 2018, pp. 261–277, doi: 10.1007/978-3-030-01261-8_16.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[18] L. Xie, T. Ahmad, L. Jin, Y. Liu, and S. Zhang, "A new CNN-based method for multi-directional car license plate detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 507–517, Feb. 2018, doi: 10.1109/TITS.2017.2784093.

[19] M. Salemdeeb and S. Erturk, "Multi-national and multi-language license plate detection using convolutional neural networks," *Eng., Technol. Appl. Sci. Res.*, vol. 10, no. 4, pp. 5979–5985, Aug. 2020, doi: 10.48084/etasr.3573.

[20] S. M. Silva and C. R. Jung, "Real-time license plate detection and recognition using deep convolutional neural networks," *J. Vis. Commun. Image Represent.*, vol. 71, Aug. 2020, Art. no. 102773, doi: 10.1016/j.jvcir.2020.102773.

[21] A. Tourani, A. Shahbahrami, S. Soroori, S. Khazaee, and C. Y. Suen, "A robust deep learning approach for automatic Iranian vehicle license plate detection and recognition for surveillance systems," *IEEE Access*, vol. 8, pp. 201317–201330, 2020, doi: 10.1109/ACCESS.2020.3035992.

[22] S. L. Chen, Q. Liu, J. W. Ma, and C. Yang, "Scale-invariant multidirectional license plate detection with the network combining indirect and direct branches," *Sensors*, vol. 21, no. 4, pp. 1–18, 2021, doi: 10.3390/s21041074.

[23] N. Omar, A. Sengur, and S. G. S. Al-Ali, "Cascaded deep learning-based efficient approach for license plate detection and recognition," *Expert Syst. Appl.*, vol. 149, Jul. 2020, Art. no. 113280, doi: 10.1016/j.eswa.2020.113280.

[24] M. Mondal, P. Mondal, N. Saha, and P. Chattopadhyay, "Automatic number plate recognition using CNN based self synthesized feature learning," in *Proc. IEEE Calcutta Conf. (CALCON)*, Dec. 2017, pp. 378–381, doi: 10.1109/CALCON.2017.8280759.

[25] J. Feng, X. Wang, and H. Lv, "Non-motor vehicle illegal behavior discrimination and license plate detection based on real-time video," *J. Phys., Conf. Ser.*, vol. 1544, no. 1, May 2020, Art. no. 012105, doi: 10.1088/1742-6596/1544/1/012105.

[26] K. B. Sathya, V. Vaidehi, and G. Kavitha, "Rendering of impaired visual effects on genesis of streak–recognizing car license plate," in *Proc. Amity Int. Conf. Artif. Intell. (AICAI)*, Feb. 2019, pp. 468–472, doi: 10.1109/AICAI.2019.8701308.

[27] S. B. Yoo and M. Han, "Temporal matching prior network for vehicle license plate detection and recognition in videos," *ETRI J.*, vol. 42, no. 3, pp. 411–419, Jun. 2020, doi: 10.4218/etrij.2019-0245.

[28] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," Dec. 2016, *arXiv:1612.03144*. Accessed: Jan. 26, 2022.

[29] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," 2016, pp. 1–13, *arXiv:1602.07360*.

[30] G. S. Baggs, P. Guerrier, A. Loeb, and J. C. Jones, "Automated copper alloy grain size evaluation using a deep-learning CNN," 2020, pp. 1–52 *arXiv:2005.09634*.

[31] B. Jones and C. J. Nachtsheim, "A class of three-level designs for definitive screening in the presence of second-order effects," *J. Qual. Technol.*, vol. 43, no. 1, pp. 1–15, Jan. 2011, doi: 10.1080/00224065.2011.11917841.
[32] W. T. Song, I.-C. Lai, and Y.-Z. Su, "A statistical robust glaucoma detection framework combining retinex, CNN, and DOE using fundus images," *IEEE Access*, vol. 9, pp. 103772–103783, 2021, doi: 10.1109/ACCESS.2021.3098032.
[33] P. F. de Aguiar, B. Bourguignon, M. S. Khots, D. L. Massart, and R. Phan-Than-Luu, "D-optimal designs," *Chemom. Intell. Lab. Syst.*, vol. 30, no. 2, pp. 199–210, 1995, doi: 10.1016/0169-7439(94)00076-X.
[34] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, "On large-batch training for deep learning: Generalization gap and sharp minima," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Sep. 2016, pp. 1–16. Accessed: Jan. 26, 2022.

YANG YANG LEE was born in Penang, Malaysia, in 1992. He received the B.Sc. (Eng.) degree in mechatronic engineering and the M.Sc. (Eng.) degree in electrical and electronic engineering from Universiti Sains Malaysia (USM), in 2016 and 2019, respectively. His research interests include artificial intelligence, data analytics, embedded systems, machine vision, and automation.

ZAINI ABDUL HALIM (Member, IEEE) was born in Penang, Malaysia, in 1973. She received the B.Sc. (Eng.) and M.Sc. (Eng.) degree in electrical and electronic engineering and the Ph.D. degree in microelectronic from Universiti Sains Malaysia (USM), in 1997, 2000, and 2008, respectively.

From 1997 to 1998, she was a Product Engineer with the Test Department of Applied Magnetic (M) Sdn Bhd. She has been a Lecturer with Universiti Sains Malaysia, since April 2004. She is the author of more than 50 articles and more than ten inventions and holds three patents. Her research interests include embedded system design and digital design.

Dr. Zaini was a recipient of the Best Session Presenter Award in ICACSIS 2012, under IEEE Indonesia Section, in 2012. She has received Laurel Wreath Award from International Federation of Inventors' Associations (IFIA) for outstanding ecological creations with high innovation values which had been shown on the 4th World Competition of Green Invention, Nuremberg, Germany, in 2014.

MOHD NADHIR AB WAHAB (Member, IEEE) received the B.Eng. (Hons.) and M.Sc. degrees in mechatronics engineering from Universiti Malaysia Perlis, in 2010 and 2012, respectively, and the Ph.D. degree in robotics and automation system from the University of Salford, U.K., in 2017. He is currently a Lecturer with the School of Computer Sciences, Universiti Sains Malaysia. His main research interests include artificial intelligence, mobile robotics, computer vision, deep learning, machine learning, optimization, navigation, and path planning.