S & M 3522

# Optimizing Algorithm Hyperparameters and Backbone of Single-shot Detector for Object Detection

Wei-Tai Huang,[1,4] Yenming J. Chen,[2,4] Jinn-Tsong Tsai,[3,4*] and Wen-Hsien Ho[4,5,6**]

[1]Department of Mechanical Engineering, National Pingtung University of Science and Technology,
Pingtung 912, Taiwan
[2]Department of Information Management, National Kaohsiung University of Science and Technology,
Kaohsiung 824, Taiwan
[3]Department of Computer Science and Artificial Intelligence, National Pingtung University,
Pingtung 900, Taiwan
[4]Department of Healthcare Administration and Medical Informatics, Kaohsiung Medical University,
Kaohsiung 807, Taiwan
[5]Department of Medical Research, Kaohsiung Medical University Hospital,
Kaohsiung 807, Taiwan
[6]College of Professional Studies, National Pingtung University of Science and Technology,
Pingtung 912, Taiwan

(Received July 31, 2023; accepted January 9, 2024)

***Keywords:*** single-shot detector, Resnet model, VGG model, Taguchi method

In this study, we explored a single-shot detector (SSD) backbone and its optimized algorithm hyperparameters for object detection, and proposed a systematic method for determining appropriate algorithm hyperparameter combinations for the SSD backbone. The VGG16 backbone for SSD has been used for object detection. The Resnet backbone won first place in the 2015 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), while the VGG16 backbone ranked second place in the 2014 ILSVRC. We selected the Resnet50 backbone for SSD for vehicle image detection research because the Resnet50 backbone has a high feature extraction capability. We proposed SSD with the Resnet50 backbone and its optimized algorithm hyperparameters, called the SSD-Resnet50 model, to replace SSD with the VGG16 backbone and its optimized algorithm hyperparameters, called the SSD-VGG16 model, to enhance the vehicle image detection feature extraction capability. The Taguchi method optimized the algorithm hyperparameters of the Resnet50 and VGG16 backbones, thus improving the detection accuracies of the SSD-Resnet50 and SSD-VGG16 models, respectively. Experimental results show that the SSD-Resnet50 model using 300 × 300 × 3 input images achieved a detection accuracy of an average average precision (AP) of 97.15% in three independent experiments, outperforming the SSD-VGG16 model using 300 × 300 × 3 input images with an average AP of 86.83% on the test set of vehicle images. As a result, the SSD-Resnet50 model has a higher accuracy of vehicle detection in images from the Caltech cars 1999 and 2001 datasets.

[*]Corresponding author: e-mail: jttsai@mail.nptu.edu.tw
[**]Corresponding author: e-mail: whho@kmu.edu.tw
https://doi.org/10.18494/SAM4651

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

## 1. Introduction

Object detection has been widely used in fault detection,(1,2) video monitoring,(3,4) medical treatment,(5) and cloud computation services.(6) Object detection techniques include object localization and image classification to locate objects of interest and determine the specific class of each object. A single-shot detector (SSD) based on deep learning proposed by Liu *et al.*(7) used VGG16 as a backbone to solve the problem of object detection accuracy and has high performance in both detection speed and accuracy.(8) SSD is made up of a couple of convolutional layers stacked together as a backbone model that functions as the feature extractor. A better feature extractor can improve the object detection performance. Therefore, Shen *et al.*(9) designed a variant of the deeply supervised DensenNets to replace the VGG16 backbone of SSD and contributed a set of design principles for designing deeply supervised objection detectors. Liu *et al.*(5) explored more feature extractors (e.g., Resnet50, VGG16, and InceptionV3) for object detection and showed that effective and efficient feature extractors can lead to improved detector performance. Zhai *et al.*(8) designed the feature extraction network DenseNet-S-32-1, in which the VGG16 backbone for SSD was replaced with DenseNet-S-32-1 to enhance the feature extraction capability. The above studies revealed that an effective strategy for improving object detection accuracy is to design a reasonable backbone. Furthermore, there are few studies on how backbone algorithm hyperparameters affect the object detection accuracy. The authors previously studied how optimizing the algorithm hyperparameters of backbones improves the image classification accuracy.(10–13) Therefore, this study is motivated by the lack of research on better feature extractor models with optimized algorithm hyperparameters of backbones to improve the object detection accuracy.

We propose a systematic approach to determine better algorithm hyperparameter combinations of the backbone for SSD for object detection. Another aim of this study is to find a better feature extractor with optimized algorithm hyperparameters for detecting vehicles in images from the Caltech cars 1999 and 2001 datasets.(15) The SSD proposed by Liu *et al.*(7) used VGG16 as a backbone to solve the problem of object detection accuracy. The VGG16 proposed by Simonyan and Zisserman(14) achieved a Top-5 error rate of 7.32% and placed second in the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), while the Resnet proposed by He *et al.*(16) achieved a Top-5 error rate of 3.57% and was the winner of the 2015 ILSVRC. Therefore, VGG16 and Resnet50 were selected as backbones for detecting vehicles in images. In the VGG16 and Resnet50 backbones, the improved classification quality can be determined by setting algorithm hyperparameter combinations before the learning process begins. In this study, we used a robust and systematic Taguchi experimental approach to search for better algorithm hyperparameter combinations for the VGG16 and Resnet50 backbones. In experimental comparisons, the SSD-Resnet50 model, which was equipped with the Resnet50 backbone and its optimized algorithm hyperparameters for SSD, had higher object localization and image classification accuracies than the SSD-VGG16 model, which had the VGG16 backbone and its optimized algorithm hyperparameters for SSD.

## 2. Problem Description

The vehicle images taken from the rear were collected from the Caltech cars 1999 and 2001 datasets. The cars 1999 dataset was taken by Markus Weber in the California Institute of Technology parking lots. There are 126 car images taken from the rear, and the resolution of each image size is 896 × 592 pixels in jpeg format. The cars 2001 dataset was taken by Paul Updike and Brad Philip on the freeways of southern California. There are 526 car images taken from the rear, and the resolution of each image size is 360 × 240 pixels in jpeg format. Because there are many repeated images, we selected some representative vehicle images as training and test data. Additionally, we generated ground truth labels for training and test images to evaluate detection accuracy. Some representative vehicle images and their ground truth labels are shown in Fig. 1. The considered problem was how to efficiently and accurately detect vehicles in images for assisting and improving autonomous driving.

## 3. Methods

We first collect and process vehicle images for object detection. Then, backbones and algorithm hyperparameters are selected and the Taguchi experimental method is used to design algorithm hyperparameter combinations for backbones. Next, detection experiments are conducted on vehicle images, and object detection performance characteristics of the SSD-VGG16 and SSD-Resnet50 models are recorded. We infer the best algorithm hyperparameter combination and finally compare the detection accuracies of the SSD-VGG16 and SSD-Resnet50 models. The details of the steps are as follows.

### 3.1 Collecting and processing vehicle images for object detection

Vehicle images taken from the rear were collected from the Caltech cars 1999 and 2001 datasets. We selected 295 vehicle images and labeled vehicles for object detection. Data augmentation was employed during training to increase model accuracy by randomly transforming the raw data. This helps to increase the diversity of the training data without

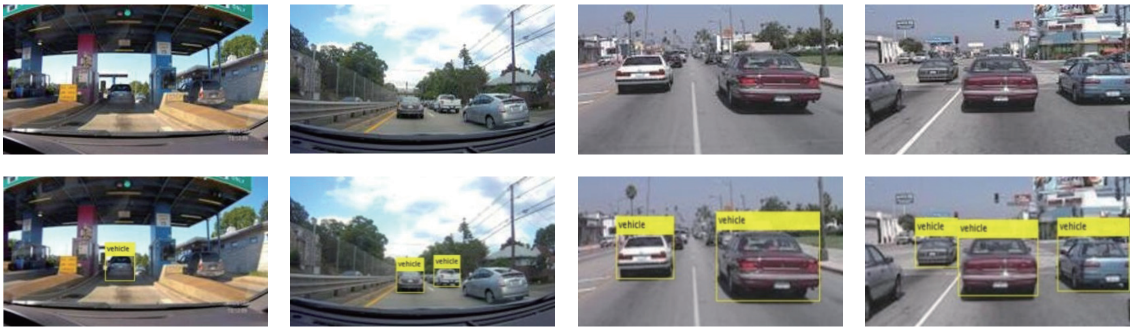


Fig. 1. (Color online) Some representative vehicle images and their ground truth labels.

requiring additional samples. Note that data augmentation techniques are not applied to the test data. The test data should ideally remain a representative of the original data and unmodified to ensure an unbiased evaluation of the model's performance.

### 3.2 Selecting backbones and algorithm hyperparameters

The VGG16 backbone for SSD proposed originally by Liu *et al.*(7) had been used for object detection. VGG16, proposed by Simonyan and Zisserman(14) of the Visual Geometry Group Lab of Oxford University, was second in image classification and won first place in object localization in the 2014 ILSVRC.(17) He *et al.*(16) proposed Resnet, which was the winner of the 2015 ILSVRC for image classification, localization, and detection, and the winner of the 2015 MS COCO for detection and segmentation. Therefore, we chose Resnet50 as the backbone for SSD in our vehicle image detection research because of its excellent feature extraction capability. To obtain high accuracy when detecting images, it is critical to select appropriate algorithm hyperparameter combinations for the VGG16 and Resnet50 backbones for SSD. Four algorithm hyperparameters (MiniBatchSize, Optimizer, LearnRateDropPeriod, and InitialLearnRate) were selected in this study for the VGG16 and Resnet50 backbones for SSD.

### 3.3 Using the Taguchi experimental method to design the algorithm hyperparameter combinations for backbones

The Taguchi method(18–20) is an experimental statistical approach used to assess and enhance product and process improvements. It focuses on reducing variation rather than eliminating it completely, aiming to improve quality while minimizing the number of experiments required to study design variables. To efficiently analyze multiple factors at once, experiments are organized in an orthogonal array (OA). The signal-to-noise ratio (SNR) and OA help identify better factor-level combinations for effective optimization. In this study, the algorithm hyperparameters for the VGG16 and ResNet50 backbones are Optimizer, MiniBatchSize, InitialLearnRate, and LearnRateDropPeriod. To efficiently explore nonlinear effects and minimize the number of experiments required, a three-level $L_9(3^4)$ OA was utilized.

### 3.4 Conducting detection experiments on vehicle images and recording object detection performance between SSD-VGG16 and SSD-Resnet50 models

Object detection results on the training and test sets include (1) the average precision ($AP$) for each experiment, (2) average $AP$ over three independent experiments, (3) the standard deviation ($SD$) of $AP$ over three independent experiments, and (4) SNR($\eta$) over three independent experiments.

$AP$ represents the area under the precision-recall curve. Precision is the positive predictive rate, while recall (sensitivity) is the true positive rate. A larger $\eta$ value indicates higher performance. To quantify $\eta$ in decibels (dB), Taguchi recommended taking the common logarithm of $\eta$ multiplied by 10. In the study, the "the-smaller-the-better" characteristic was employed, $\eta = -10 \log (\overline{y} - m)^2$, where $\overline{y} = \frac{1}{n}\sum_{t=1}^{n} y_t$ ($y_t$ represents the trained and predicted *AP* in each experiment) and $m = 1$ (i.e., the target's *AP* is 100%).

### 3.5 Inferring the best algorithm hyperparameter combination

We used the $L_9(3^4)$ OA response table and $\eta$ values to find the best algorithm hyperparameter combination. The effect of different factors is $E_{fl}$, which is the average of the sum of $\eta_i$ for factor $f$ at level $l$, where $f$ is the name of the factor, $l$ is the number of the level, and $i$ is the number of the experiment. After nine experiments of the three-level $L_9(3^4)$ OA, we used the response table to investigate $\eta$ at each factor level. The response table shows the average $\eta$ for each factor level and the maximum average $\eta$ for each factor. We used the response table to find the best factor level, which is the level with the highest $E_{fl}$ value in the experimental area.

### 3.6 Comparing detection accuracy between SSD-VGG16 and SSD-Resnet50 models

Object detection performance in terms of *AP* was compared between the SSD-VGG16 and SSD-Resnet50 models.

## 4. Results

We proposed a Resnet50 backbone for SSD and its optimized algorithm hyperparameters, called the SSD-Resnet50 model, to enhance the feature extraction capability for detecting vehicles in images. The experimental environment was a computer with the Turbo-RTX2080Ti-11G GPU and Intel i7 CPU, and we used Matlab R2022a and its toolbox developed by MathWorks.

### 4.1 Image data preparation and algorithm hyperparameter selection

The experimental data included training and test sets to test the performance of detecting vehicles in images. In the study, a total of 295 vehicle images were chosen and annotated for object detection. For each experiment, 236 images (80% of the dataset) were randomly assigned as the training set, while the remaining 59 images (20% of the dataset) were designated as the test set. Ground truth labels were generated for both the training and test sets to evaluate the accuracy of the object detection. To achieve effective object detection, each image was processed as a 300 × 300 × 3 image. Data augmentation methods involve various techniques such as the random scaling of images and their box labels, the random horizontal flipping of images and their box labels, and the application of dithering to image colors. An example of a vehicle image with data augmentation is illustrated in Fig. 2.

For the training process, we selected the VGG16 and Resnet50 backbones for SSD, and attempted to set different algorithm hyperparameter combinations before the learning process started. We selected four algorithm hyperparameters for the VGG16 and Resnet50 backbones for

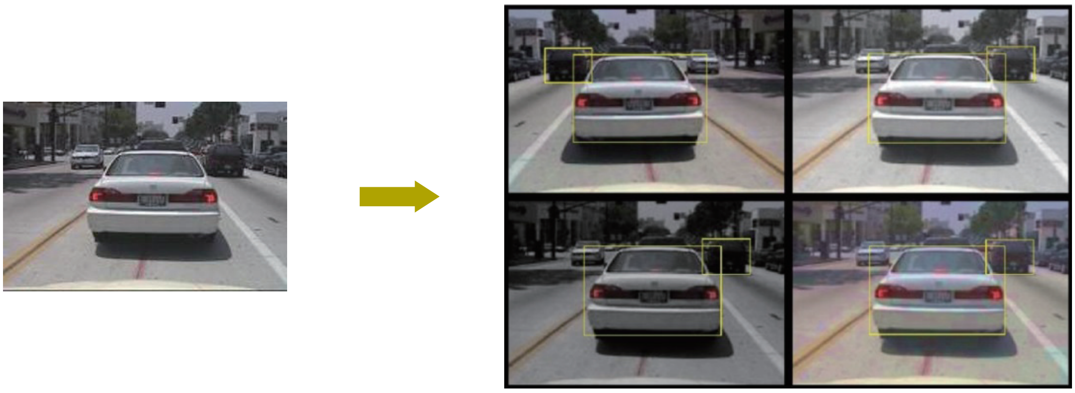


Fig. 2. (Color online) Example of vehicle image with data augmentation.

SSD: Optimizer, MiniBatchSize, InitialLearnRate, and LearnRateDropPeriod. Also, LearnRateDropFactor was configured at 0.8, MaxEpoch was set to 300, and LearnRateSchedule was designated as ‘piecewise’. The LearnRate was calculated by multiplying the learning rate of the previous period by the LearnRateDropPeriod value.

### 4.2 Designing algorithm hyperparameter combinations for VGG16 and Resnet50 backbones for SSD using the Taguchi method

The three-level OA with the smallest number of experiments for the four factors is $L_9(3^4)$. Table 1 shows the $L_9(3^4)$ OA and Table 2 shows the factors and their levels for the VGG16 and Resnet50 backbones for SSD. The three levels of Optimizer (factor A) are adaptive moment estimation (adam), stochastic gradient descent with momentum (sgdm), and adaptive moment estimation (adam). Owing to GPU memory constraints, MiniBatchSize (factor B) has three levels of 14, 16, and 18. InitialLearnRate (factor C) has three levels of $10^{-1}$, $10^{-3}$, and $10^{-4}$. LearnRateDropPeriod (factor D) has three levels of 30, 40, and 50. The $L_9(3^4)$ OA requires only nine experiments instead of $81(3^4)$ experiments. Table 3 shows the algorithm hyperparameter combinations of the values in Tables 1 and 2. The algorithm hyperparameter combinations were used in the VGG16 and Resnet50 backbones for SSD for detecting vehicles in images.

### 4.3 Conducting detection experiments on vehicle images and recording object detection performance of the SSD-VGG16 model

We used the algorithm hyperparameter combinations presented in Table 3 for independent experiments on the training and test sets for the VGG16 backbone for SSD. The performance test results for detecting vehicles in images are given in Table 4, which shows *AP* in a single run, as well as average *AP* over three independent experiments, *SD* over three independent experiments, and $\eta$ over three independent experiments.

Table 5 shows the response for each factor of the VGG16 backbone for SSD. Table 5 shows that factor levels 1, 3, 3, and 2 were selected for factors A, B, C, and D, respectively. Therefore, the best factor-level combination for the VGG16 backbone for SSD was A1: adam, B3: 18, C3: $10^{-4}$, and D2: 40.

Table 1
$L_9(3^4)$ OA.

| Experiment No. | Factors | | | |
|---|---|---|---|---|
| | A | B | C | D |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 2 | 2 | 2 |
| 3 | 1 | 3 | 3 | 3 |
| 4 | 2 | 1 | 2 | 3 |
| 5 | 2 | 2 | 3 | 1 |
| 6 | 2 | 3 | 1 | 2 |
| 7 | 3 | 1 | 3 | 2 |
| 8 | 3 | 2 | 1 | 3 |
| 9 | 3 | 3 | 2 | 1 |

Table 2
Factors and levels for VGG16 and Resnet 50 backbones for SSD.

| Factor<br>(Algorithm hyperparameter) | Levels | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| A: Optimizer | adam | sgdm | adam |
| B: MiniBatchSize | 14 | 16 | 18 |
| C: InitialLearnRate | $10^{-1}$ | $10^{-3}$ | $10^{-4}$ |
| D: LearnRateDropPeriod | 30 | 40 | 50 |

Table 3
Algorithm hyperparameter combinations for VGG16 and Resnet 50 backbones for SSD.

| Experiment No. | Algorithm hyperparameters | | | |
|---|---|---|---|---|
| | Optimizer | MiniBatchSize | InitialLearnRate | LearnRateDropPeriod |
| 1 | adam | 14 | $10^{-1}$ | 30 |
| 2 | adam | 16 | $10^{-3}$ | 40 |
| 3 | adam | 18 | $10^{-4}$ | 50 |
| 4 | sgdm | 14 | $10^{-3}$ | 50 |
| 5 | sgdm | 16 | $10^{-4}$ | 30 |
| 6 | sgdm | 18 | $10^{-1}$ | 40 |
| 7 | adam | 14 | $10^{-4}$ | 40 |
| 8 | adam | 16 | $10^{-1}$ | 50 |
| 9 | adam | 18 | $10^{-3}$ | 30 |

Table 4
*AP*, average *AP*, *SD*, and $\eta$ values achieved by VGG16 backbone for SSD in detecting vehicles in images using algorithm hyperparameter combinations given in Table 3 in three independent experiments.

| Experiments 1–9 | Dataset | *AP*-Experimental run no. | | | Average *AP* | *SD* | $\eta$ |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | | | |
| 1 | Training set | 0 | 0 | 0 | 0 | 0 | 0 |
| | Test set | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Training set | 0.6241 | 0.6334 | 0.7081 | 0.6552 | 0.04605 | 9.24865 |
| | Test set | 0.482 | 0.5843 | 0.7813 | 0.6159 | 0.15213 | 8.31036 |
| 3 | Training set | 0.894 | 0.8756 | 0.8791 | 0.8829 | 0.00977 | 18.6289 |
| | Test set | 0.911 | 0.8054 | 0.8632 | 0.8599 | 0.05288 | 17.0692 |
| 4 | Training set | 0.0001 | 0.1153 | 0 | 0.0385 | 0.06654 | 0.34071 |
| | Test set | 0 | 0.2344 | 0 | 0.0781 | 0.13533 | 0.70664 |

Table 4
(Continued) *AP*, average *AP*, *SD*, and $\eta$ values achieved by VGG16 backbone for SSD in detecting vehicles in images using algorithm hyperparameter combinations given in Table 3 in three independent experiments.

| Experiments 1–9 | Dataset | *AP*-Experimental run no. | | | Average *AP* | *SD* | $\eta$ |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | | | |
| 5 | Training set | 0.0266 | 0.0481 | 0.0105 | 0.0284 | 0.01886 | 0.25025 |
| | Test set | 0.029 | 0.0315 | 0.0081 | 0.0229 | 0.01285 | 0.20092 |
| 6 | Training set | 0 | 0 | 0 | 0 | 0 | 0 |
| | Test set | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Training set | 0.8849 | 0.8668 | 0.8829 | 0.8782 | 0.00992 | 18.2871 |
| | Test set | 0.8843 | 0.8633 | 0.8009 | 0.8495 | 0.04338 | 16.4493 |
| 8 | Training set | 0 | 0 | 0 | 0 | 0 | 0 |
| | Test set | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | Training set | 0.4617 | 0.5508 | 0.6486 | 0.5537 | 0.09348 | 7.00746 |
| | Test set | 0.3708 | 0.2792 | 0.575 | 0.4083 | 0.15143 | 4.55846 |

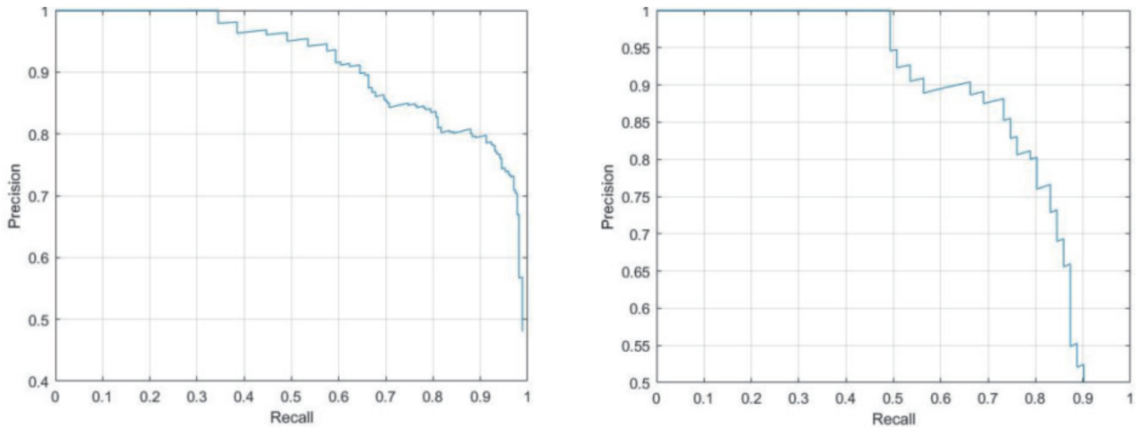
Table 5
Responses for each factor of VGG16 backbone for SSD.

| Level | Factors | | | |
|---|---|---|---|---|
| | A | B | C | D |
| 1 | 8.4598 | 5.7186 | 0.0000 | 1.5865 |
| 2 | 0.3025 | 2.8371 | 4.5252 | 8.2532 |
| 3 | 7.0026 | 7.2092 | 11.2398 | 5.9253 |
| Effect | 8.1573 | 4.3721 | 11.2398 | 6.6667 |
| Maximum | 8.4598 | 7.2092 | 11.2398 | 8.2532 |
| Best level number | 1 | 3 | 3 | 2 |
| Best level value | adam | 18 | $10^{-4}$ | 40 |

In the validation experiments, the optimized algorithm hyperparameter combination (i.e., A1: adam, B3: 18, C3: $10^{-4}$, and D2: 40) was used to detect vehicles in images in three independent experiments using the SSD-VGG16 model. Table 6 shows the *AP*, average *AP*, *SD*, and $\eta$ values achieved by the SSD-VGG16 model in three independent experiments on the training and test sets of vehicle images. The average *AP* and $\eta$ values of the SSD-VGG16 model were 0.8683 and 17.6061, respectively, which exceeded those in each experiment on the $L_9(3^4)$ OA (Table 4) performed in the test set. Figure 3 shows *AP* examples for the training and test sets of vehicle images using the SSD-VGG16 model. The optimized algorithm hyperparameter combination in the response table produced the best result, even though not all factor level combinations were tested. Therefore, the optimized algorithm hyperparameter combination obtained in validation experiments was used for the SSD-VGG16 model to detect vehicles in images.

Table 6
*AP*, average *AP*, *SD*, and $\eta$ values achieved by SSD-VGG16 model for detecting vehicles in images using optimized algorithm hyperparameter combination in three independent experiments.

| Model | Dataset | *AP*-Experimental run no. | | | Average *AP* | *SD* | $\eta$ |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | | | |
| SSD-VGG16 | Training set | 0.8736 | 0.8876 | 0.8678 | 0.8763 | 0.0102 | 18.1549 |
| | Test set | 0.8697 | 0.8397 | 0.8954 | 0.8683 | 0.0279 | 17.6061 |



(a) (b)

Fig. 3. (Color online) *AP* examples for training and test sets of vehicle images using SSD-VGG16 model. (a) *AP* for training set and (b) *AP* for test set.

## 4.4 Conducting detection experiments on vehicle images and recording object detection performance of the SSD-Resnet50 model

We used the algorithm hyperparameter combinations presented in Table 3 for the independent experiments on the training and test sets with the Resnet50 backbone for SSD. In performance tests for detecting vehicles in images, Table 7 shows *AP* in a single run, as well as average *AP* over three independent experiments, *SD* over three independent experiments, and $\eta$ over three independent experiments. Table 8 shows the responses for each factor of the Resnet50 backbone for SSD. Table 8 shows that factor levels 1, 3, 2, and 2 were selected for factors A, B, C, and D, respectively. Thus, the best factor-level combination for the Resnet50 backbone for SSD was A1: adam, B3: 18, C2: $10^{-3}$, and D2: 40.

In the validation experiment, the optimized algorithm hyperparameter combination (i.e., A1: adam, B3: 18, C2: $10^{-3}$, and D2: 40) was used to detect vehicles in images in three independent experiments with the SSD-Resnet50 model. Table 9 shows the *AP*, average *AP*, *SD*, and $\eta$ values achieved by the SSD-Resnet50 model in the three independent experiments on the training and test sets of vehicle images. The average *AP* and $\eta$ values obtained by the SSD-Resnet50 model were 0.9715 and 30.8929, respectively, which exceeded those in each experiment on the $L_9(3^4)$ OA (Table 7) performed on the test set. Figure 4 shows examples of *AP* for the training and test

Table 7
*AP*, average *AP*, *SD*, and *η* values achieved by Resnet50 backbone for SSD in detecting vehicles in images using algorithm hyperparameter combinations given in Table 3 in three independent experiments.

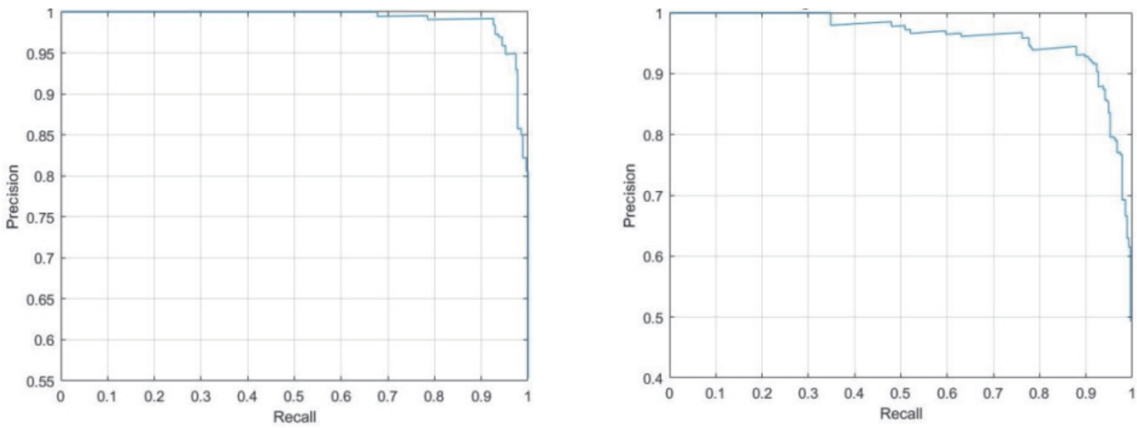
| Experiments 1–9 | Dataset | *AP*-Experimental run no. | | | Average *AP* | *SD* | *η* |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | | | |
| 1 | Training set | 0.199 | 0.0155 | 0.1281 | 0.1142 | 0.0925 | 1.0533 |
| | Test set | 0.1622 | 0.0084 | 0.1718 | 0.1141 | 0.0917 | 1.0526 |
| 2 | Training set | 0.9908 | 0.9887 | 0.9895 | 0.9897 | 0.0011 | 39.7152 |
| | Test set | 0.9884 | 0.9701 | 0.9404 | 0.9663 | 0.0242 | 29.4474 |
| 3 | Training set | 0.9948 | 0.9888 | 0.9889 | 0.9908 | 0.0034 | 40.7558 |
| | Test set | 0.936 | 0.9428 | 0.994 | 0.9576 | 0.0317 | 27.4527 |
| 4 | Training set | 0.108 | 0.2567 | 0.2082 | 0.1910 | 0.0758 | 1.8407 |
| | Test set | 0.1245 | 0.2639 | 0.2015 | 0.1966 | 0.0698 | 1.9017 |
| 5 | Training set | 0.0095 | 0.021 | 0.0212 | 0.0172 | 0.0067 | 0.1510 |
| | Test set | 0.0065 | 0.0326 | 0.0286 | 0.0226 | 0.0141 | 0.1983 |
| 6 | Training set | 0.9355 | 0.8976 | 0.942 | 0.9250 | 0.0240 | 22.5026 |
| | Test set | 0.9046 | 0.8511 | 0.8901 | 0.8819 | 0.0277 | 18.5575 |
| 7 | Training set | 0.987 | 0.984 | 0.987 | 0.9860 | 0.0017 | 37.0774 |
| | Test set | 0.9669 | 0.9376 | 0.9583 | 0.9543 | 0.0151 | 26.7953 |
| 8 | Training set | 0.0465 | 0.1261 | 0.1601 | 0.1109 | 0.0583 | 1.0210 |
| | Test set | 0.0401 | 0.0805 | 0.1618 | 0.0941 | 0.0620 | 0.8587 |
| 9 | Training set | 0.9957 | 0.9944 | 0.9927 | 0.9943 | 0.0015 | 44.8319 |
| | Test set | 0.9255 | 0.9815 | 0.952 | 0.9530 | 0.0280 | 26.5580 |

Table 8
Responses for each factor of Resnet50 backbone for SSD.

| Level | Factors | | | |
|---|---|---|---|---|
| | A | B | C | D |
| 1 | 19.3176 | 9.9166 | 6.8229 | 9.2696 |
| 2 | 6.8858 | 10.1681 | 19.3024 | 24.9334 |
| 3 | 18.0707 | 24.1894 | 18.1488 | 10.0710 |
| Effect | 12.4318 | 14.2728 | 12.4795 | 15.6638 |
| Maximum | 19.3176 | 24.1894 | 19.3024 | 24.9334 |
| Best level number | 1 | 3 | 2 | 2 |
| Best level value | adam | 18 | $10^{-3}$ | 40 |

Table 9
*AP*, average *AP*, *SD*, and *η* values achieved by SSD-Resnet50 model for detecting vehicles in images using optimized algorithm hyperparameter combination in three independent experiments.

| Model | Dataset | *AP*-Experimental run no. | | | Average *AP* | *SD* | *η* |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | | | |
| SSD-Resnet50 | Training set | 0.9915 | 0.9928 | 0.9935 | 0.9926 | 0.0010 | 42.6154 |
| | Test set | 0.9899 | 0.9538 | 0.9707 | 0.9715 | 0.0181 | 30.8929 |



(a)

(b)

Fig. 4. (Color online) Examples of *AP* for training and test sets of vehicle images using SSD-Resnet50 model. (a) *AP* for training set and (b) *AP* for test set.

sets of vehicle images using the SSD-Resnet50 model. The optimized algorithm hyperparameter combination in the response table produced the best result even though not all factor level combinations were tested. Therefore, the optimized algorithm hyperparameter combination determined from the validation experiments was used for the SSD-Resnet50 model to detect vehicles in images.

### 4.5 Comparing the detection accuracies of SSD-VGG16 and SSD-Resnet50 models

The optimized algorithm hyperparameter combination (i.e., A1: adam, B3: 18, C3: $10^{-4}$, and D2: 40) obtained by the Taguchi experimental method was used to detect vehicles in images in three independent experiments using the SSD-VGG16 model. The average *AP* and $\eta$ values obtained by the SSD-VGG16 model performed on the test set were 0.8683 and 17.6061, respectively. Additionally, the optimized algorithm hyperparameter combination (i.e., A1: adam, B3: 18, C2: $10^{-3}$, and D2: 40) obtained by the Taguchi experimental method was used to detect vehicles in images in three independent experiments using the SSD-Resnet50 model. The average *AP* and $\eta$ values of the SSD-Resnet50 model for the test set were 0.9715 and 30.8929, respectively.

The results show that, in three independent experiments on the test set of vehicle images, the SSD-Resnet50 model used on 300 × 300 × 3 input images achieved a detection accuracy with an average *AP* of 0.9715 and an $\eta$ value of 30.8929, outperforming the SSD-VGG16 model used on 300 × 300 × 3 input images where an average *AP* of 0.8683 and an $\eta$ value of 17.6061 were obtained. Therefore, the SSD-Resnet50 model had superior detection accuracy in detecting vehicles in images.

## 5. Discussion

The results of this study showed that the appropriate algorithm hyperparameter combination for the SSD-VGG16 and SSD-Resnet50 models is essential for accurately detecting vehicles in images. Table 4 shows that the average *AP* values of experiments 1, 4, 5, 6, and 8 are below 0.1 because of the poor algorithm hyperparameter combinations for the VGG16 backbone for SSD. Table 7 shows that the average *AP* values of experiments 1, 4, 5, and 8 are below 0.2 because of the poor algorithm hyperparameter combinations for the Resnet50 backbone for SSD. The results indicate that the poor algorithm hyperparameter combinations for the SSD-VGG16 and SSD-Resnet50 models prevented the accurate detection of vehicles in images. Therefore, in this study, we used the Taguchi method to determine the optimized algorithm hyperparameter combination for the feature extractor backbone for SSD for object detection.

The SSD-Resnet50 model had superior detection accuracy in detecting vehicle images. Figure 5 shows that there is only one vehicle ground truth label on the test images, but the SSD-Resnet50 model found two vehicle labels on the test images. This result shows that the SSD-Resnet50 model can effectively find the features of a vehicle and accurately locate and classify the vehicle. In future applications, a good object detection model will be able to find new objects beyond the ground truth labels in the validation dataset and can then find new objects on unlabeled test images.

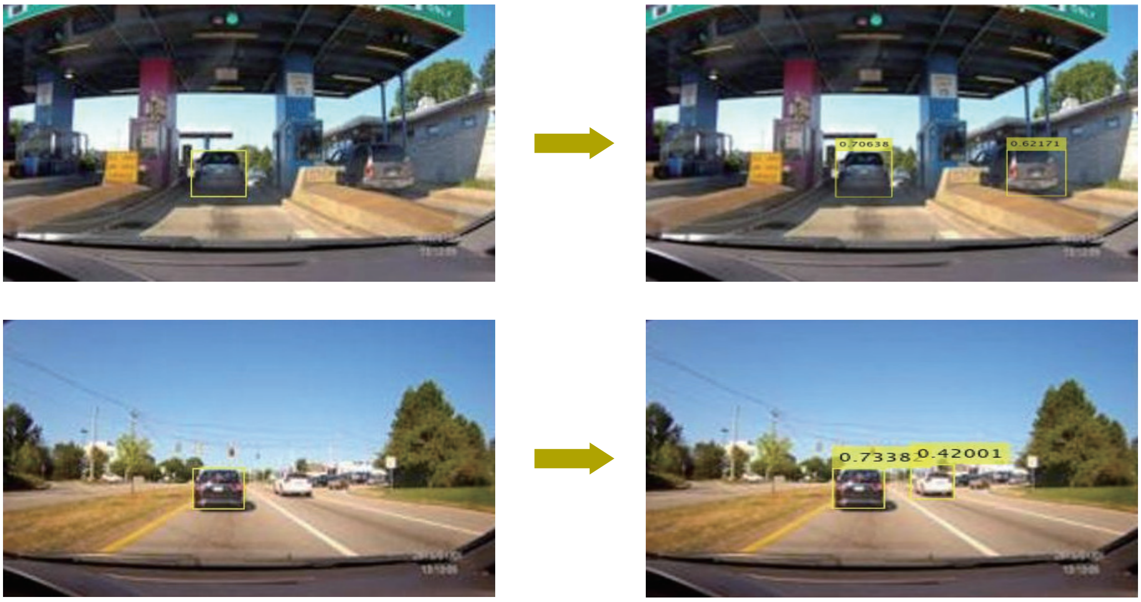


Fig. 5. (Color online) Two examples showing ground truth label of one vehicle in image (left picture) vs labels of two vehicles (right picture) obtained by SSD-Resnet50 model.

## 6. Conclusions

The proposed SSD-Resnet50 model can detect vehicles in images accurately and efficiently. The study has three contributions. The first contribution is the demonstration of the high detection accuracy obtained using the optimized algorithm hyperparameter combination of the SSD-Resnet50 and SSD-VGG16 models. The second contribution is the confirmation that the Taguchi experimental method can identify the optimal algorithm hyperparameter combination for SSD backbones. The third contribution is the finding that a good object detection model can identify new objects on the test set beyond the ground truth labels. Experimental results showed that the algorithm hyperparameters for the Resnet50 and VGG16 backbones were optimized by the Taguchi method, thereby improving the detection accuracies of the SSD-Resnet50 and SSD-VGG16 models, respectively. Additionally, in three independent experiments on the test set of vehicle images, the SSD-Resnet50 model achieved an average $AP$ of 0.9715 and an $\eta$ value of 30.8929 for 300 × 300 × 3 input images, outperforming the SSD-VGG16 model, which had an average $AP$ of 0.8683 and an $\eta$ value of 17.6061. Therefore, the SSD-Resnet50 model had superior detection accuracy in detecting vehicles in images obtained from the Caltech cars 1999 and 2001 datasets.

## Acknowledgments

This study was funded by the National Science and Technology Council, Taiwan, R.O.C., under Grants MOST 111-2221-B-037-007, NSTC 112-2218-E-992-004, NSTC 112-2221-E-153-004, and NSTC 112-2221-E-037-004-MY3. This work was also supported in part by the NPUST-KMU joint research project (NPUST-KMU-112-P009) and NKUST-KMU joint research project (#NKUSTKMU-112-KK-009).

## References

1 Y. Wu, B. Jiang, and N. Lu: IEEE Trans. Syst. Man Cybern.: Syst. **49** (2019) 2108. https://doi.org/10.1109/TSMC.2017.2757264
2 Y. Wu, B. Jiang, and Y. Wang: ISA Trans. **99** (2020) 488. https://doi.org/10.1016/j.isatra.2019.09.020
3 L. Hu and Q. Ni: IEEE Internet Things J. **5** (2018) 747. https://doi.org/10.1109/JIOT.2017.2705560
4 A. Mhalla, T. Chateau, S. Gazzah, and N. E. B. Amara: IEEE Trans. Intell. Transp. Syst. **20** (2019) 4006. https://doi.org/10.1109/TITS.2018.2876614
5 M. Liu, J. Jiang, and Z. Wang: IEEE Access **7** (2019) 75058. https://doi.org/10.1109/ACCESS.2019.2921027
6 P. T. Wang, S. Y. Lin, and J. S. Sheu: Advan. Technol. Innov. **6** (2021) 213. https://doi.org/10.46604/aiti.2021.7192
7 W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg: Comput. Vision and Pattern Recognit. (2016) 1. https://doi.org/10.48550/arXiv.1512.02325
8 S. Zhai, D. Shang, S. Wang, and S. Dong: IEEE Access **8** (2020) 24344. https://doi.org/10.1109/ACCESS.2020.2971026
9 Z. Shen, Z. Liu, J. Li, Y. G. Jiang, Y. Chen, and X. Xue: IEEE Intern. Conf. Comput. Vision (2017) 1919. https://doi.org/10.1109/ICCV.2017.212
10 Y. M. Chen, Y. J. Chen, W. H. Ho, and J. T. Tsai: BMC Bioinf. **22** (2021) 1. https://doi.org/10.1186/s12859-021-04083-x
11 Y. M. Chen, F. I. Chou, W. H. Ho, and J. T. Tsai: BMC Bioinf. **22** (2021) 1. https://doi.org/10.1186/s12859-022-04558-5

12 Y. M. Chen, Y. J. Chen, Y. K. Tsai, W. H. Ho, and J. T. Tsai: J. Intell. and Fuzzy Syst. **40** (2021) 7883. https://doi.org/10.3233/JIFS-189610

13 F. I. Chou, Y. K. Tsai, Y. M. Chen, J. T. Tsai, and C. C. Kuo: IEEE Access **7** (2019) 68316. https://doi.org/10.1109/ACCESS.2019.2918563

14 K. Simonyan and A. Zisserman: Comput. Vision and Pattern Recognit. (2014) 1. https://doi.org/10.48550/arXiv.1409.1556

15 B. Philip, P. Updike, and P. Perona: Caltech Data (2022). https://doi.org/10.22002/D1.20085

16 K. He, X. Zhang, S. Ren, and J. Sun: Comput. Vision and Pattern Recognit. (2015) 770. https://doi.org/10.48550/arXiv.1512.03385

17 O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F. F. Li: Int. J. Comput. Vision **115** (2015) 211. https://doi.org/10.1007/s11263-015-0816-y

18 H. H. Lee: Taguchi Methods: Principles and Practices of Quality Design. (Gau-Lih, Taiwan, 2011).

19 G. Taguchi, S. Chowdhury, and S. Taguchi: Robust Engineering. (McGraw-Hill, New York, 2000).

20 Y. Wu: Taguchi Methods for Robust Design. (The American Society of Mechanical Engineers, New York, 2000).