



Article

Recognizing Road Surface Traffic Signs Based on Yolo Models Considering Image Flips

Christine Dewi ^{1,2,*} , Rung-Ching Chen ^{3,*} , Yong-Cun Zhuang ³, Xiaoyi Jiang ⁴ and Hui Yu ⁵

¹ Department of Information Technology, Satya Wacana Christian University, Salatiga 50711, Indonesia

² Artificial Intelligent Research Center, Satya Wacana Christian University, Salatiga 50711, Indonesia

³ Department of Information Management, Chaoyang University of Technology, Taichung 41349, Taiwan

⁴ Department of Mathematics and Computer Science, University of Münster, D-48149 Münster, Germany

⁵ School of Creative Technologies, The University of Portsmouth, Portsmouth PO1 2UP, UK

* Correspondence: christine.dewi@uksw.edu (C.D.); crching@cyut.edu.tw (R.-C.C.)

Abstract: In recent years, there have been significant advances in deep learning and road marking recognition due to machine learning and artificial intelligence. Despite significant progress, it often relies heavily on unrepresentative datasets and limited situations. Drivers and advanced driver assistance systems rely on road markings to help them better understand their environment on the street. Road markings are signs and texts painted on the road surface, including directional arrows, pedestrian crossings, speed limit signs, zebra crossings, and other equivalent signs and texts. Pavement markings are also known as road markings. Our experiments briefly discuss convolutional neural network (CNN)-based object detection algorithms, specifically for Yolo V2, Yolo V3, Yolo V4, and Yolo V4-tiny. In our experiments, we built the Taiwan Road Marking Sign Dataset (TRMSD) and made it a public dataset so other researchers could use it. Further, we train the model to distinguish left and right objects into separate classes. Furthermore, Yolo V4 and Yolo V4-tiny results can benefit from the “No Flip” setting. In our case, we want the model to distinguish left and right objects into separate classes. The best model in the experiment is Yolo V4 (No Flip), with a test accuracy of 95.43% and an IoU of 66.12%. In this study, Yolo V4 (without flipping) outperforms state-of-the-art schemes, achieving 81.22% training accuracy and 95.34% testing accuracy on the TRMSD dataset.

Keywords: CNN; Yolo; object recognition; road sign; machine learning



Citation: Dewi, C.; Chen, R.-C.; Zhuang, Y.-C.; Jiang, X.; Yu, H. Recognizing Road Surface Traffic Signs Based on Yolo Models Considering Image Flips. *Big Data Cogn. Comput.* **2023**, *7*, 54. <https://doi.org/10.3390/bdcc7010054>

Academic Editor: Min Chen

Received: 11 February 2023

Revised: 17 March 2023

Accepted: 20 March 2023

Published: 22 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Technologies for traffic signs and road markings, which have recently risen to the top of the list of research priorities, are attracting a lot of attention. It has been widely accepted because of various research studies covering areas such as engineering, traffic safety, education, and human physical abilities [1,2]. Due to the lack of empirical studies on the understanding of traffic signs and markings in Taiwan, such a study may need to be conducted. It is necessary for people who use roads to be able to identify, comprehend, and obey traffic signs and road marking signs to lower the number of accidents that occur on such roads. To provide unambiguous information, traffic signs are designed using several fundamentally distinct design styles. In addition, the background of many different buildings and shop signs makes it hard for the system to identify the street signs automatically. Thus, it becomes difficult to locate the road signs in many environments [3]. On the other hand, earlier research has mostly concentrated on deciphering road signs and has given less consideration to drivers' acquaintance with and adherence to traffic regulations as a separate topic in the field of traffic safety investigation [4]. Accidents, such as those caused by excessive speed or inappropriate lane changes, sometimes occur when drivers ignore or fail to detect a sign ahead. In addition, there are currently only a handful of Taiwan-adapted road sign detection systems, and in many studies, researchers collected traffic signs from locations around the world for analysis.

Machine learning-based approaches proposed by Poggenhans et al. [5] include employing optical character recognition to detect road marking signs and artificial neural networks (ANNs) to categorize them. For feature extraction, they make use of a histogram of markings. It is possible to categorize text ground markings by their method, such as crosswalks, stop lines, arrow signs, and other types of surface markings [6–8].

Danescu and Nedevschi [9] constructed an autonomous road marking identification and tracking system that used a two-step segmentation technique for detecting and recognizing road markings. Depending on the scenario, the accuracy of the classifications achieved ranges from 80 to 95 percent, with the lowest accuracy being 80 percent. Qin and colleagues used a machine vision approach to investigate four different types of road markings. Images of the marking contours were generated at random using image processing techniques and then extracted from the images. The classification and detection modules received the extracted features after they were sent to them. Methods such as You Only Look Once (Yolo) [10] and Single Shot Detection (SSD) [11–13], as well as a region proposal-based method, are used to detect road markings on a map. The region-based strategy surpassed the sliding window search method in terms of the number of suggestions received and the amount of time it took to complete the investigation [14].

To detect objects, the Yolo V4 algorithm, which is based on the cross-stage partial network (CSPNet), has been presented. As part of this study, a network scaling approach is utilized to adjust the depth, width, and resolution of the network in addition to the topology of the network, which ultimately led to the development of the Scaled-Yolo V4 algorithm. Yolo V4 was developed specifically for real-time object detection using general graphics processing units (GPUs). To get the best speed or accuracy trade-off, C.Y. Wang et al. [15] redesigned Yolo V4 to Yolo V4-CSP [16,17].

The following are some of the most important contributions that this research has made: (1) Includes a condensed explanation of CNN-based object recognition methods, with a special emphasis on the Yolo V2, Yolo V3, Yolo V4, and Yolo V4-tiny models. (2) Our experimental studies examine and evaluate several state-of-the-art object detectors, including those used to detect traffic signs, among other things. Vital parameters such as the mean acquisition time (mAP), the detection time (IoU), and the number of BFLOPS are measured in performance metrics. (3) Experimentally, we distinguish between left and right objects. Flip data augmentation can be disabled by setting $flip = 0$ in the configuration file for road signs. In this study, Yolo V4 (No Flip) outperformed state-of-the-art schemes, obtaining 81.22% accuracy in training and 95.34% accuracy in testing the Taiwan Road Marking Sign Dataset (TRMSD). (4) We investigate the importance of the flip and no flip parameters in the Yolo configuration file and provide a full discussion of them.

The following is the organizational structure of this research study. Section 2 discusses the related works that have been published recently and the technique we propose, which is described in Section 3. Section 4 offers a description of the experiment as well as the results of the investigation. Section 5 discusses our conclusions as well as our future work.

2. Materials and Methods

2.1. Road Marking Recognition

Many scientists and engineers have conducted substantial studies into the automated recognition of road markings [18]. The identification of road markings and signage has been the subject of earlier research, which has used a number of image processing methodologies [19]. Two types of object detectors are commonly used: one-stage object detectors and two-stage object detectors with several stages. Using a single convolution neural network (CNN) operation, it is possible to obtain the output of a single-stage object detector. The high-score region proposals received from the first stage are typically fed into the second stage for two-stage object detectors. Neural networks have effectively recognized road markings [20]. Kheyrollahi et al. suggested a solution based on inverse perspective mapping and multi-level binarization to extract robust road marking features [21].

Ding et al. [22] described a method for detecting and identifying road markers. To recognize five road marking signs, the researchers combined the properties of the histogram of oriented gradients (HOG) with those of a support vector machine (SVM). Alternately, a neural network is used to identify road markings, significantly increasing the precision of the system. Scientists say they are using hierarchical neural networks with backpropagation techniques as a learning process [12,23].

Road sign recognition based on the Yolo architecture has also attracted considerable attention, and several papers have discussed this topic. W. Yang et al. [24] tested Yolo V3 and Yolo V4 using the CSUST (Chinese Traffic Sign Detection Benchmark) dataset, which is divided into four categories: warning, speed limit, directional, and prohibitory signs. The experimental results showed that Yolo V4 outperformed Yolo V3 in target detection, with better performance in recognizing road signs and detecting small objects. D. Mijić et al. [25] proposed a suggested solution for traffic sign detection using Yolo V3 and a custom image dataset. L. Gatelli et al. [26] proposed a vehicle classification method suitable for use in Brazil that helps management personnel address social needs related to traffic safety.

In [27], there is a proposal for a further technique that makes use of machine learning and is intended to detect and classify road markings. This illustration made use of a binarized normed gradient network, a support vector machine (SVM) classifier, and a principal component analysis to identify and classify various types of objects. For image-recognition networks to act like biological systems, convolution can improve the accuracy of the results obtained [28]. A CNN has also been used to detect and categorize traffic signs [29,30]. The latest deep learning methods, such as CNN, have effectively solved the object recognition problem. On the PASCAL VOC [31] dataset, detection frameworks such as Faster R-CNN [32] and Fast R-CNN demonstrate their superior detection performance. Faster R-CNN is one of these algorithms, which abandons classical selective search in favor of region proposal networks (RPNs) to attain superior performance.

Road marking identification requires real-time processing speed, and a faster R-CNN can solve the challenge, but it falls short in terms of inference speed. As an alternative, today's leading detection frameworks, such as SSD [12] and Yolo, may be inferred in real-time while remaining resilient for applications such as road marking detection. The Yolo V2, V3, V4, and V4-tiny methods were used to recognize road markings signs in Taiwan, which was the subject of our research. We also add the flip method to our data preprocessing to improve the performance of our proposed method.

2.2. You Only Look Once (YOLO)

A single-shot detection architecture, on the other hand, utilizes a single CNN as opposed to several CNNs to forecast numerous bounding boxes and the relevant classes concurrently. Yolo V2 is a single-shot detection framework that is at the forefront of the most recent technological developments. The ability to make real-time inferences is the most significant aspect of this detector system. The Yolo V2 detection framework surpasses the Faster R-CNN algorithm in terms of both the mean average precision and the inference speed. Yolo V3 is both an heir to and an improvement over Yolo V1 and Yolo V2 [33] in that it is both an inheritance and an improvement. In the Yolo V1 technique, input images are resampled to a predetermined size and divided into an $m \times m$ grid. According to preliminary findings, the network Darknet-53, which has a larger architecture than VGG-16, is more useful in gathering various and intricate knowledge from objects and, as a result, plays a vital role in enhancing the detection accuracy of Yolo V3. The Yolo V3 algorithm, which was proposed in 2018, was comprised entirely of CNN [34,35].

On top of that, each image is divided up, and bounding boxes and probability distributions are calculated for each grid cell [36,37]. Yolo V3 is composed of 106 layers, and it generates predictions by mixing data from a variety of scales. The output image has a size of 416 pixels by 416 pixels, and it was produced by blending three different scales together [38]. Moreover, the detection is carried out on three different layers simultaneously [39]. The width and height measurements that were given to the computer were

13×13 and 26×26 , respectively, as well as 52×52 [40]. Yolo V4 [41], the most recent version in the Yolo series, was launched in April 2020 as a new edition of the Yolo network.

As a result, we tested Yolo V4 utilizing the integrated dataset in our trials, as we believe that model enhancement can lead to breakthroughs in accuracy and efficiency. C.Y. Wang et al. [42,43] redesigned Yolo V4 to Yolo V4-tiny to get the best speed or accuracy trade-off. A cross-stage partial network (CSPNet) is designed to attribute the problem to the same gradient information within network optimization. The complexity of the network optimization can be significantly reduced while maintaining accuracy [44–46].

2.3. Compared Method

Our systems are depicted in Figure 1. Particularly for the Yolo V2, Yolo V3, Yolo V4, and Yolo V4-tiny image processing systems, this study employs a number of CNN-based object detection techniques. Furthermore, we employ data augmentation, flip or no flip. Specifically, in this experiment, we train the model to distinguish among objects on the left and right sides of the screen that are classified as independent classes [47–49].

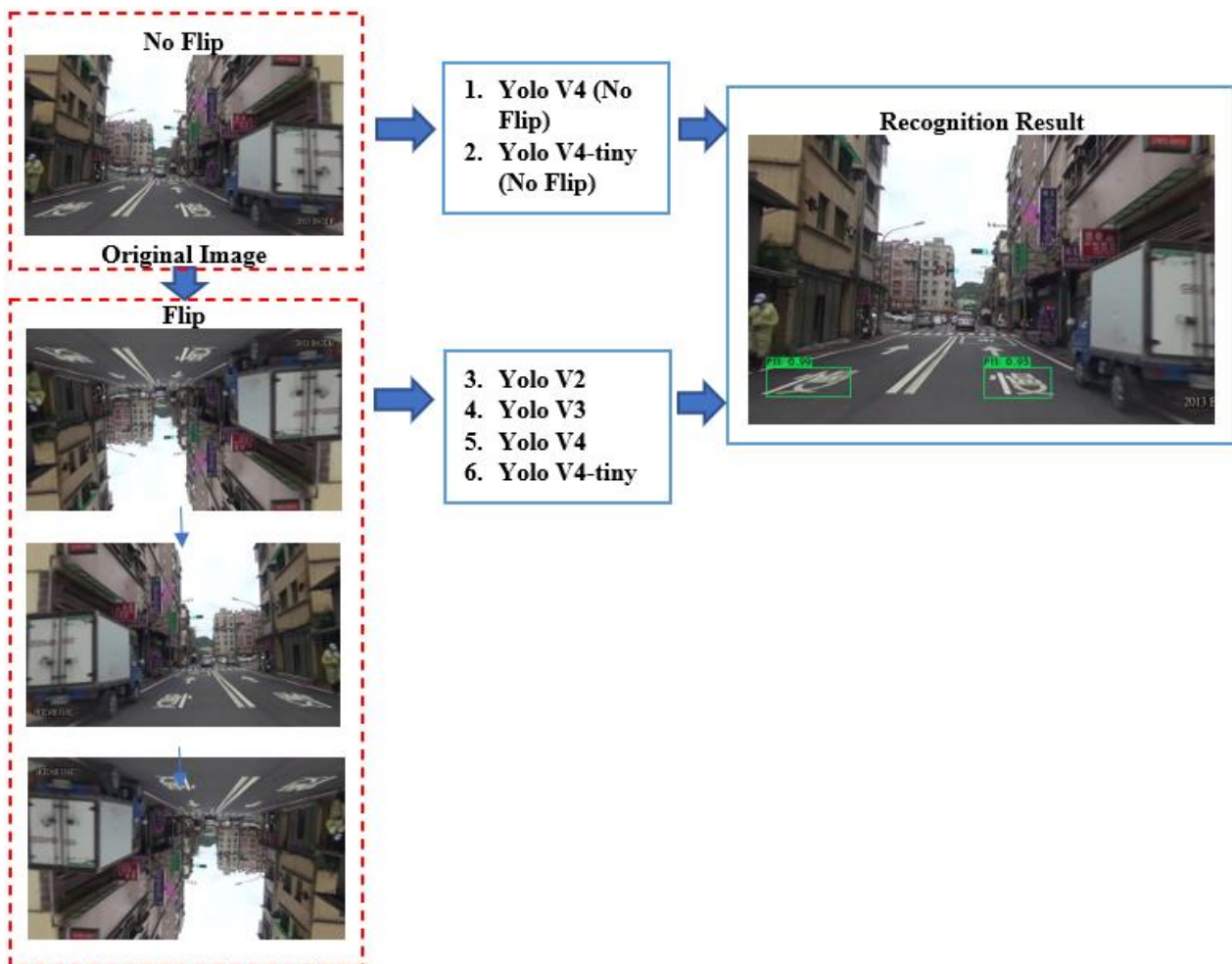


Figure 1. An overview of the system.

As a result of the fact that the program labels each category in a distinct manner, the BBox mark tool [34] was developed in order to generate a bounding box for the entire sign image. This box may have many markings. During the first stage of the experiment, each label was only compared to a single training model, and there was only one detector model used for the detection process. The majority of platforms for annotation support the Yolo labeling format, which results in a single annotation text file being produced for each image. Each object in the image is marked with a bounding-box (BBox) annotation in

each text file. Each object in the image has its own annotation. They are based on the size of the image and range from 0 to 1. Each of them is represented in the following manner: <object-class-ID> <X center> <Y center> <Box width> <Box height>. The Equations (1)–(6) serve as the foundation for the adjustment procedure [50].

$$dw = 1/W \quad (1)$$

$$x = \frac{(x_1 + x_2)}{2} \times dw \quad (2)$$

$$dh = 1/H \quad (3)$$

$$y = \frac{(y_1 + y_2)}{2} \times dh \quad (4)$$

$$w = (x_2 - x_1) \times dw \quad (5)$$

$$h = (y_2 - y_1) \times dh \quad (6)$$

H indicates for the height of the image, dh refers for the absolute height of the image, W serves for the width of the image, and dw represents for the absolute width of the picture. Algorithm 1 describes the Yolo V4 road marking detection process.

Algorithm 1. Yolo V4 road marking detection process.

1. Create grids with a size of $(n \times n)$ using the provided image data.
 2. Make a total of K bounding boxes and give an estimate of the number of anchor boxes in each box.
 3. Utilizing CNN, fully extract all object features from the image.
 4. Predict the $\mathbf{b} = [b_x, b_y, b_w, b_h, b_c]^T$ and the $\mathbf{class} = [R1, R2, R3 \dots R15]^T$.
 5. Choose the optimum confidence IoU_{pred}^{truth} of the K bounding boxes with the threshold IoU_{thres} .
 6. If IoU_{pred}^{truth} means that the bounding box includes the object. Otherwise, the bounding box does not contain the object.
 7. Select the group that has the best projection of being true when using non-maximum suppression (NMS).
 8. Shows the results of tests performed on road markings.
-

Nevertheless, NMS is structured in the following ways: First, arrange predictions according to their level of confidence in their accuracy. If we look at the predictions for the same class and find that the IoU with the existing prediction is more than 0.5, we have no choice but to start with the best rankings and ignore the prediction that is currently in place. The last stage of the process results in the production of a categorized image that bears a label indicating the class.

2.4. Experiment Setting

Table 1 explains our experiment setting. In addition, we used the default settings of Yolo V2, Yolo V3, Yolo V4, and Yolo V4-tiny. In this default setting, Yolo implements image flipping. Our experiment set flip = 0 for Yolo V4 and Yolo V4-tiny, which means not using flip when processing the image.

Table 1. Experiment Setting.

Model	Flip
Yolo V2	Yes
Yolo V3	Yes
Yolo V4	Yes
Yolo V4 (No Flip)	No
Yolo V4-tiny	Yes
Yolo V4-tiny (No Flip)	No

3. Results

3.1. Data Pre-Processing

The Nvidia RTX 3080 GPU accelerator and an Intel i7-11700 Central Processing Unit (CPU) with eight core processors were utilized as the training model environment for the purpose of detecting and recognizing road markings. The random-access memory (RAM) is equipped with 32 gigabytes of DDR4-3200 memory.

The condition of the image being flipped is shown in Figure 2. A flipped image, commonly described as a reversed image, is a fixed or motion picture that is formed by mirroring the original image across a horizontal axis. A flipped image is one that has been reflected across the vertical axis [51,52]. A flip is a feature in a photograph that enables us to rotate a picture horizontally or vertically. Furthermore, Figure 2a depicts the original image, Figure 2b depicts the flipped vertical image, Figure 2c depicts the flipped horizontal image, and Figure 2d depicts both images.

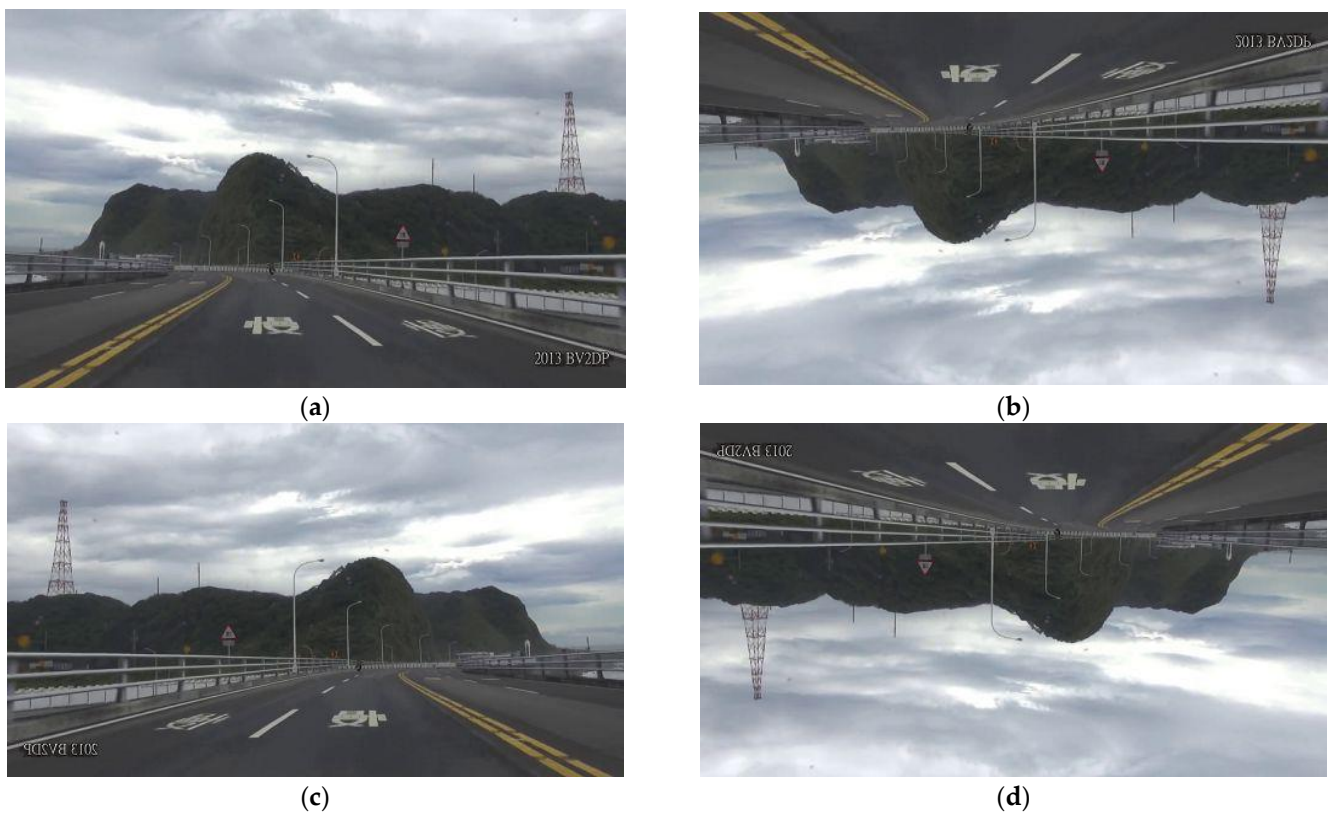


Figure 2. Flipped image condition. (a) Original image; (b) Flipped vertical image; (c) Flipped horizontal image; (d) Flipped both images.

3.2. Dataset

Furthermore, this experiment with the Taiwan road marking sign was performed using pictures that we collected from video and image sources. 80% of the dataset is used for training, while 20% is used to test the results. The video was recorded by the dashboard camera during the daytime in Taichung, Taiwan. Table 2 displays the Taiwan Road Marking Sign Dataset (TRMSD). Besides, we use images that range from 391 to 409 for each class to avoid data imbalance. Therefore, the total number of images in our dataset is 6009, and their dimensions are 512 by 288. Our study included a total of 15 classes, numbered P1 through P15. These classes were as follows: “Go Straight, Turn Left”, “Turn Right”, “Turn Right or Go Straight”, “Turn Left or Go Straight”, “Zebra Crossing”, “Slow Sign”, “Overtaking Prohibited”, “Barrier Line”, “Cross Hatch”, and “Stop Line”. Our study also included the following speed limits: “40”, “50”, “60”, and “70”. Figure 3a depicts the labels of the TRMSD datasets, which contain 15 classes. Class P7, Class P2, and Class P9 consist

of the most instances, totaling more than 400 instances. All classes in this data set have more than 300 instances, and Figure 3b illustrates the labels correlogram of the dataset.

Table 2. Taiwan Road Marking Sign Dataset (TRMSD).











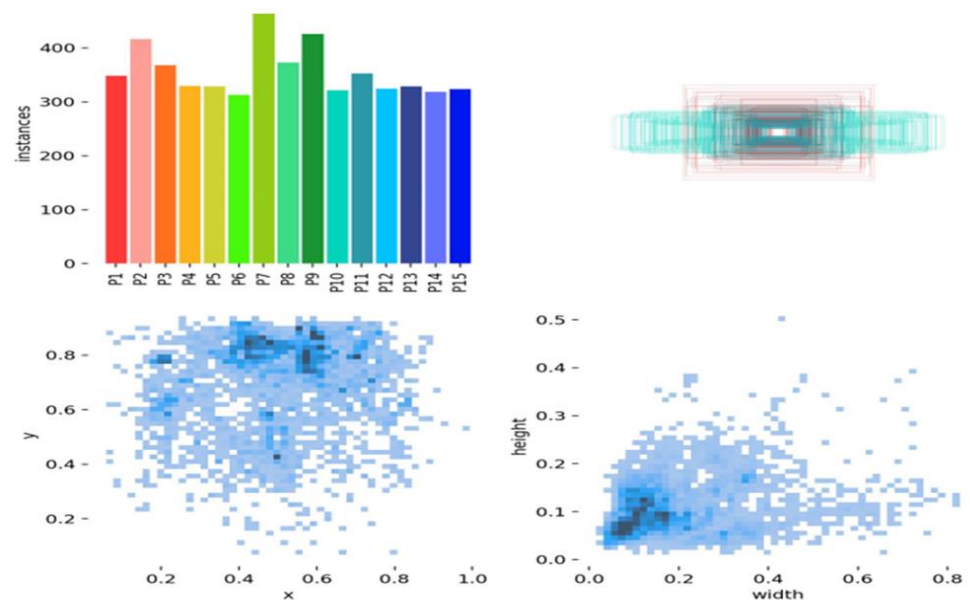
Class ID	Chinese Name	English Name	Image	Total Image
P1	右轉	Turn Right		405
P2	左轉	Turn Left		401
P3	直走	Go Straight		407
P4	直走或右轉	Turn Right or Go Straight		409
P5	直走或左轉	Turn Left or Go Straight		403
P6	速限40	Speed Limit (40)		391
P7	速限50	Speed Limit (50)		401
P8	速限60	Speed Limit (60)		400
P9	速限70	Speed Limit (70)		398
P10		Zebra Crossing (Crosswalk)		401

Table 2. Cont.

Class ID	Chinese Name	English Name	Image	Total Image
P11		Slow Sign		399
P12		Overtaking Prohibited		404
P13		Barrier Line		409
P14		Cross Hatch		398
P15		Stop Line		403



(a) Labels

Figure 3. Cont.

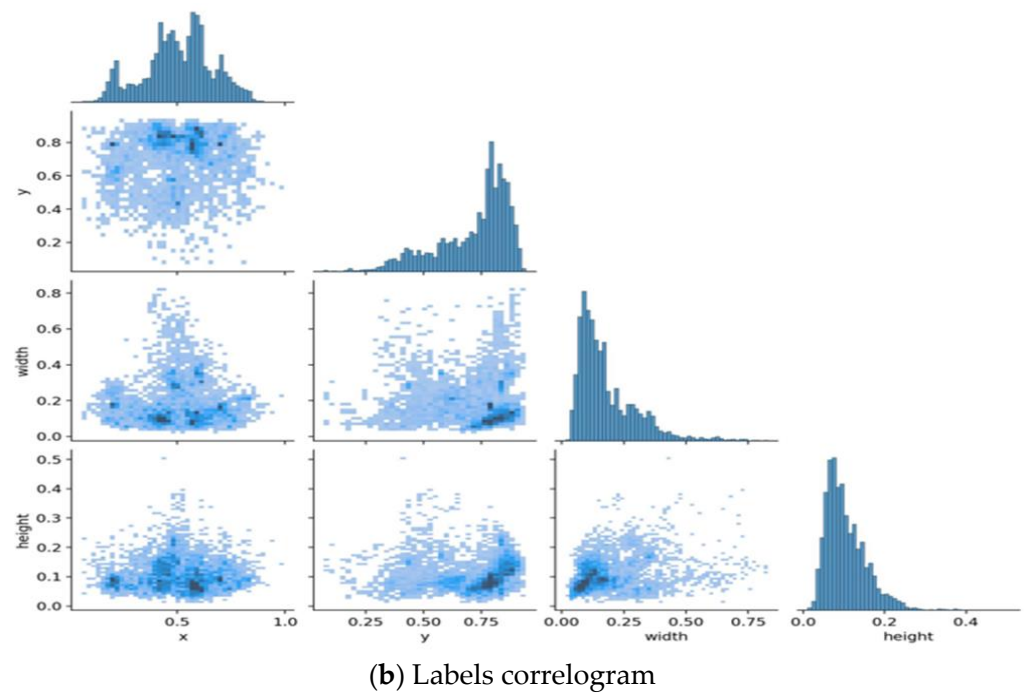


Figure 3. Taiwan Road Marking Sign Dataset (TRMSD). (a) Labels, and (b) Label correlograms.

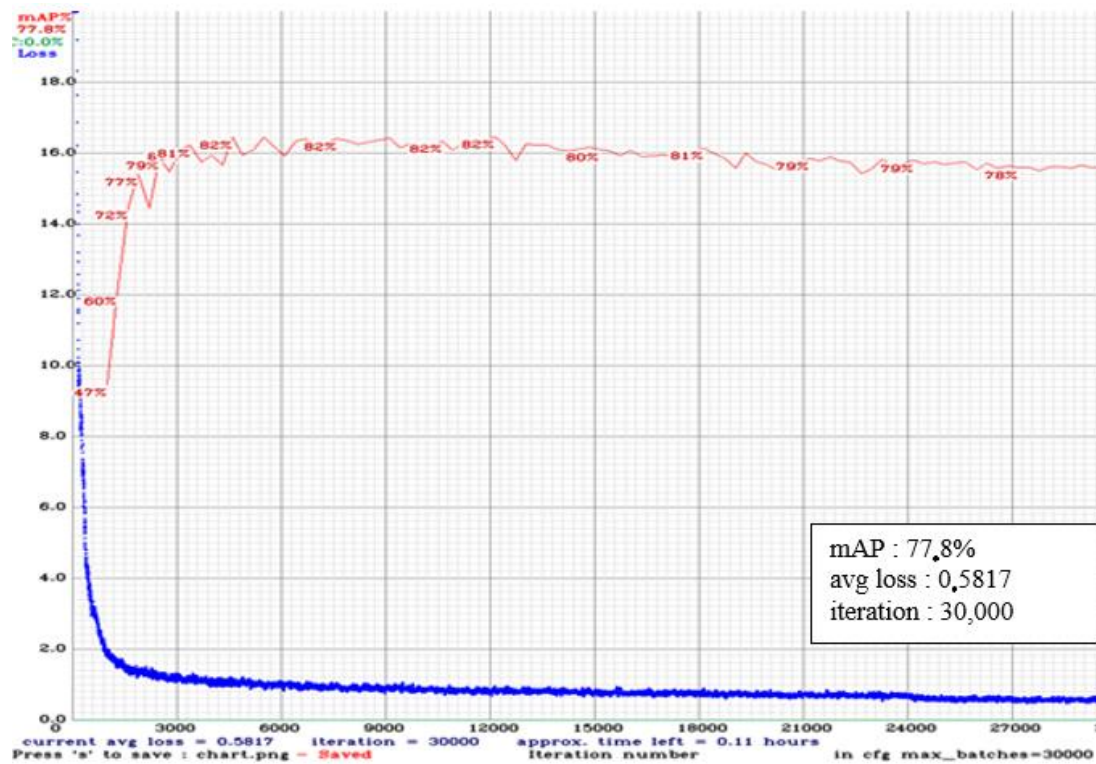
4. Discussion

4.1. Yolo Training Result

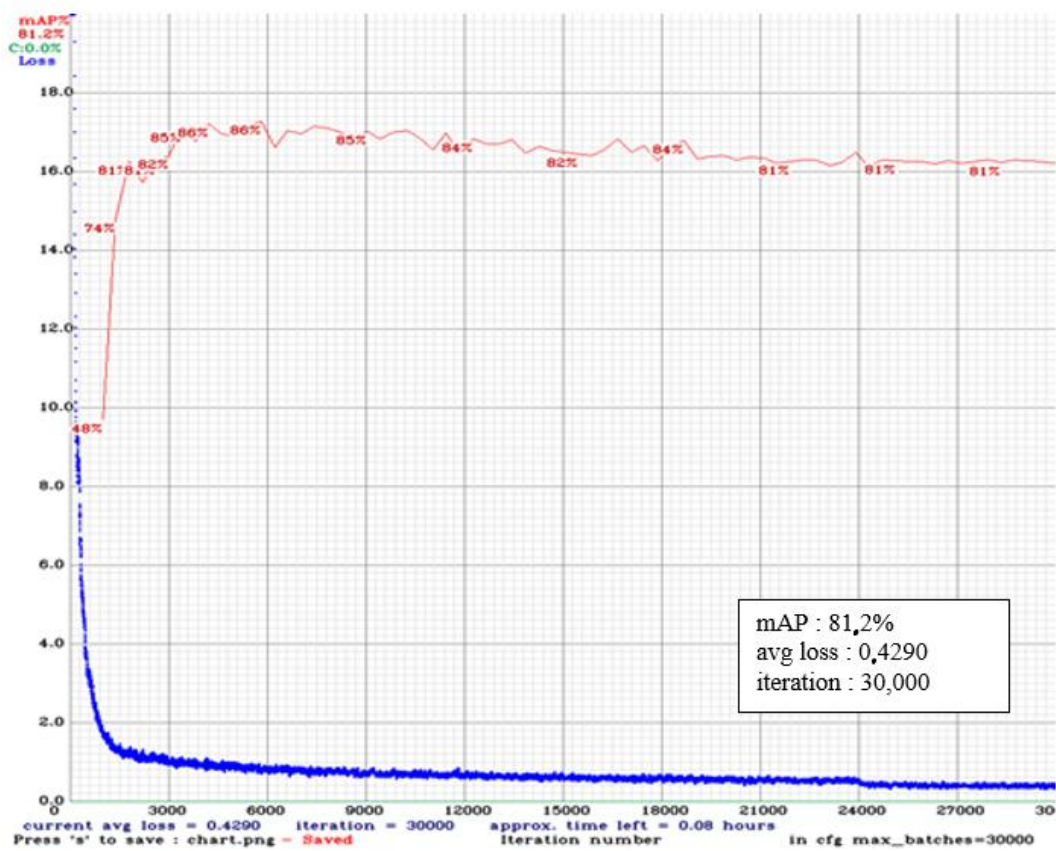
Figure 4 illustrates the training results for each model in the experiment. According to the findings of our research, utilizing a learning rate of 0.00261 for the analysis, a learning rate decay of 0.1 at each iteration, and a momentum learning rate of 0.949 for the model helps improve the Yolo model while it is being trained. To overcome the problem of over-fitting, we incorporate cross-validation and early stopping into our experiment design. A common procedure is to perform 5-fold cross-validation to obtain an out-of-sample prediction error. The rules for early stopping indicate how many times a learner can repeat an activity before becoming overly proficient. Max batches = 30,000 iterations are used in this experiment, as well as the step policy: steps = 24,000 and 27,000, scales = 0.1 and 0, momentum = 0.949, decay = 0.0005, saturation = 1.5, exposure = 1.5, and mosaic = 1. Generally, $m \times$ class object detectors require a maximum batch size of $2000 \times m$ for execution. The training process is terminated in the experiment after 30,000 iterations (2000×15 classes). Other variables are employed in the training process, including the current iteration number and the scale (0.1, 0.1). This value is adjusted on a consistent basis, and the formula for determining the current learning rate is as follows: learning rate \times scales [0] \times scales [1] = 0.00001.

The average loss value of Yolo V2 during the training stage is 0.1162, and the training loss remains stable after 3000 epochs. Further, Yolo V2 exhibits a mAP of 76.75% and an IoU of 53.61%. Yolo V4 (No Flip) achieves the highest mAP , 81.22%, while training with IoU at 65.98% and a loss value of 0.429. Followed by Yolo V4-tiny (No Flip) with mAP 80.47% and IoU 63.79%. Next, Yolo V3 got a mAP value of 78.31% and an IoU of 58.62%, as shown in Table 3. In addition, Table 4 shows the training results for each class, P1 to P15. Classes P1, P6, P7, P8, P9, P11, and P15 achieve above 90% accuracy. The intersection over union (IoU) metric is used to evaluate object detectors, as indicated in Equation (7) [53,54].

$$IoU = \frac{Area_{pred} \cap Area_{gt}}{Area_{pred} \cup Area_{gt}} \quad (7)$$



(a) Yolo V4



(b) Yolo V4 (No Flip)

Figure 4. The training results for Yolo V4 model. (a) Yolo V4, (b) Yolo V4 (No Flip).

Table 3. Training performance results.

Model	Loss Value	Precision	Recall	F1-Score	IoU (%)	mAP@0.50 (%)
Yolo V2	0.1162	0.68	0.83	0.74	53.61	76.75
Yolo V3	0.1493	0.73	0.81	0.77	58.62	78.31
Yolo V4	0.5817	0.72	0.81	0.76	58.23	77.76
Yolo V4 (No Flip)	0.429	0.81	0.86	0.84	65.98	81.22
Yolo V4-tiny	0.3289	0.66	0.81	0.73	52.83	80.55
Yolo V4-tiny (No Flip)	0.2428	0.76	0.86	0.81	60.45	84.77

Table 4. Training performance results for each class.

Class ID	Yolo V2	Yolo V3	Yolo V4	Yolo V4 (No Flip)	Yolo V4-Tiny	Yolo V4-Tiny (No Flip)
P1	68.95	74.87	76.49	95.20	81.88	95.60
P2	61.40	63.57	64.43	81.22	74.48	84.16
P3	67.80	70.27	68.13	52.51	69.04	62.98
P4	40.87	45.44	42.69	57.15	63.45	73.66
P5	31.67	35.10	27.35	53.02	50.36	72.76
P6	100.00	100.00	100.00	100.00	100.00	100.00
P7	91.15	90.10	90.81	90.33	86.32	88.89
P8	98.48	98.18	98.78	99.78	95.64	95.59
P9	99.99	99.98	99.99	99.98	100.00	100.00
P10	85.24	87.93	85.21	90.48	90.27	93.40
P11	97.45	97.08	99.59	97.69	98.36	98.96
P12	68.32	64.42	72.36	59.56	70.45	70.73
P13	69.76	70.43	63.67	63.04	61.10	95.43
P14	87.08	87.48	89.51	87.23	78.85	81.65
P15	83.04	89.75	87.36	91.14	88.01	87.74
Average	76.75	78.31	77.76	81.22	80.55	84.77

Our method computes the area of overlap between the predicted bounding box and the ground truth bounding box, and this information is placed in the numerator of the covariance matrix. Alternatively, the denominator is the area of union, which is defined as the area included by both the predicted bounding box and the ground truth bounding box in the same coordinate system. Taking the area of overlap and dividing it by the area of the union gives the result of *IoU*. There are three categories that the result examples can be placed into: (1) True positive (TP): the model predicted a label and matched it correctly as per ground truth. (2) False positive (FP): the model predicted a label, but it is not part of the ground truth. (3) True negative (TN): the model does not predict the label and is not part of the ground truth. The Equations (8) and (9) describe both the precision and the recall [55,56].

$$Precision (P) = \frac{TP}{TP + FP} \quad (8)$$

$$Recall (R) = \frac{TP}{TP + FN} \quad (9)$$

Another assessment index, F1 [57], is depicted in Equation (10).

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (10)$$

The Yolo loss function, calculated using Equation (11) is [58].

$$\begin{aligned} & \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y - \hat{y}_i)^2 \right] \\ & + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] + \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} (C_i - \hat{C}_i)^2 \\ & + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2 + \sum_{i=0}^{s^2} \mathbb{I}_i^{obj} \sum_{cc \in classes} (p_i(c) - \hat{p}_i(c))^2 \end{aligned} \quad (11)$$

where \mathbb{I}_{ij}^{obj} denotes whether the object appears in cell i , and \mathbb{I}_{ij}^{obj} denotes that the j^{th} bounding box predictor in cell i is responsible for the prediction. The next step is to utilize $(\hat{x}, \hat{y}, \hat{w}, \hat{h}, \hat{c}, \hat{p})$ to represent the center coordinates of the predicted bounding box, as well as its breadth, height, confidence, and category likelihood. The vast majority of boxes are empty of their contents. This results in a problem known as class imbalance, in which we train the model to recognize background more frequently than we train it to recognize objects. As a solution, we reduce the significance of this loss by a factor of $\lambda_{noobj} = 0.5$.

4.2. Result Discussions

Table 5 describes the test performance results of each algorithm. Yolo V4 (No Flip) showed the highest mAP of 95.43% in the experiment, with an IoU of 66.18%, precision of 83%, recall of 93%, and an F1-score of 91%. Next, Yolo V4 achieves a mAP of 93.55% with an IoU of 66.24%. Yolo V4-tiny (No Flip) showed 92.98% mAP and 64.7% IoU . Yolo V3 reached the lowest mAP value with a mAP value of 89.97% and an IoU of 61.98. augmentation as flip = 0.

Table 5. Testing performance results.

Model	Recall	Precision	F1-Score	TP	FP	IoU (%)	mAP@0.50 (%)
Yolo V2	0.94	0.73	0.82	5006	1883	56.79	90.53
Yolo V3	0.88	0.79	0.83	4694	1259	61.98	89.97
Yolo V4	0.93	0.82	0.87	4970	1096	66.24	93.55
Yolo V4 (No Flip)	0.95	0.83	0.89	5074	1003	66.12	95.43
Yolo V4-tiny	0.88	0.72	0.80	4707	1802	58.15	87.53
Yolo V4-tiny (No Flip)	0.94	0.82	0.88	4996	1065	66.86	94.42

Depending on the outcomes of the tests, it can be determined that the “No Flip” parameter has the potential to increase the performance of the Yolo V4 and Yolo V4-tiny results. In addition, the Yolo V4 mAP increased by 1.98%, from 93.55% to 95.43%. Furthermore, Yolo V4-tiny mAP rose 6.89% from 87.53% to 94.42%. In our experiment, we disabled flip data augmentation by setting flip = 0. We would like the model to be able to classify Left and Right objects as distinct classes.

Table 6 describes the performance results of each class, P1 to P15. Class P6 obtained the highest average mAP of 100%, followed by Class P9 with 99.87% mAP , and Class P11 with 99.68% mAP . In contrast, a minimum mAP of 76.79% was obtained by Class P5. Class P5 is a Turn Left or Straight sign; this class is like Class P4 (Turn Right or Straight). Furthermore, Class P4 reaches 82.19% mAP , and this value is slightly different from Class P5.

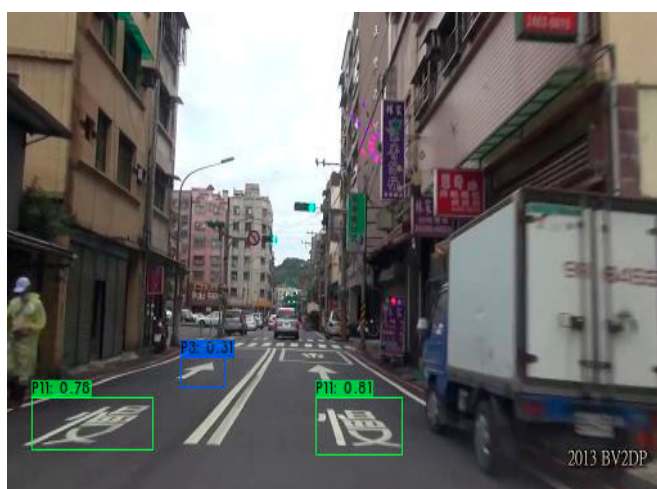
Overall, all classes in the experiment achieved a high mAP above 90%. Most of the marks for Speed Limit 60, Speed Limit 50, Speed Limit 40, and Speed Limit 70 receive higher mAP than those for other classes. This is because the model can recognize numbers well, and this class is very different from other classes.

Table 6. Testing performance results for each class.

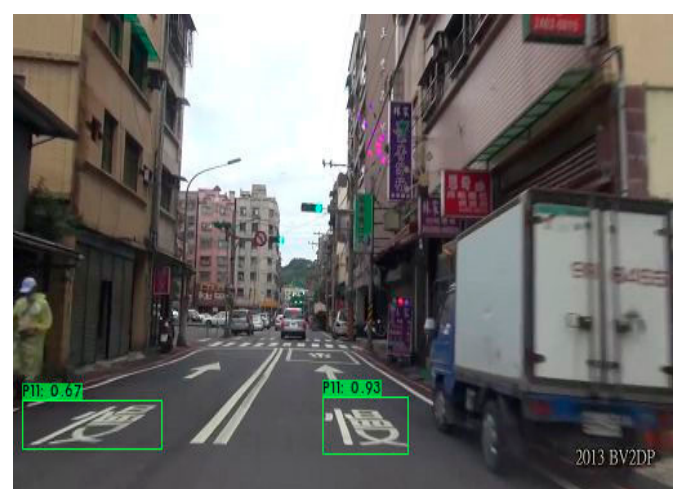
Class ID	Yolo V2	Yolo V3	Yolo V4	Yolo V4 (No Flip)	Yolo V4-Tiny	Yolo V4-Tiny (No Flip)	Average
P1	86.61	85.57	89.03	98.35	86.80	98.38	91.56
P2	80.99	82.01	86.59	96.66	84.22	96.53	88.78
P3	87.08	84.42	91.23	92.21	78.38	87.38	87.85
P4	77.84	73.63	78.58	92.60	71.14	91.58	82.19
P5	70.82	66.78	73.9.	88.20	65.44	85.88	76.79
P6	100.00	100.00	100.00	100.00	99.99	100.00	100.00
P7	93.56	96.02	99.08	98.38	97.50	97.91	95.99
P8	96.92	97.51	98.76	99.03	98.49	98.49	98.10
P9	99.90	99.83	99.95	99.86	99.86	99.87	99.87
P10	96.59	98.30	98.73	97.38	95.26	97.87	95.16
P11	99.15	99.69	99.97	99.91	99.50	99.50	99.63
P12	92.26	89.87	99.14	97.58	79.94	91.84	92.19
P13	85.85	82.96	93.67	79.28	71.62	81.76	84.00
P14	97.00	97.73	98.52	96.50	92.99	94.53	94.65
P15	93.34	95.18	96.01	95.57	91.82	94.74	92.94
Average	90.53	89.97	94.95	95.43	87.53	94.42	91.98

The road marking sign Class P11 recognition result is shown in Figure 5. All models can detect and recognize marks well, with varying accuracy. Yolo V4 (No Flip) can recognize two signs of P11 with 99% and 95% accuracy, respectively.

Nevertheless, Figure 6 illustrates the recognition results for Class P8. Yolo V4 (No Flip) can recognize three signs with an accuracy of 88%, 90%, and 65%, respectively, as shown in Figure 6d. Furthermore, Yolo V4 and Yolo V4-tiny can only detect two signs of class P8 with the same image as in Figure 6c,f. The outcomes of the incorrect recognition of the road markings are depicted in Figure 7. Some models may not recognize all the markings in the image. They only recognize 1 sign, as shown in Figure 7a–f. The Yolo V4-tiny underwent double detection, as shown in Figure 7e. Furthermore, this experiment's result can be applied to countries in Asia and other languages.

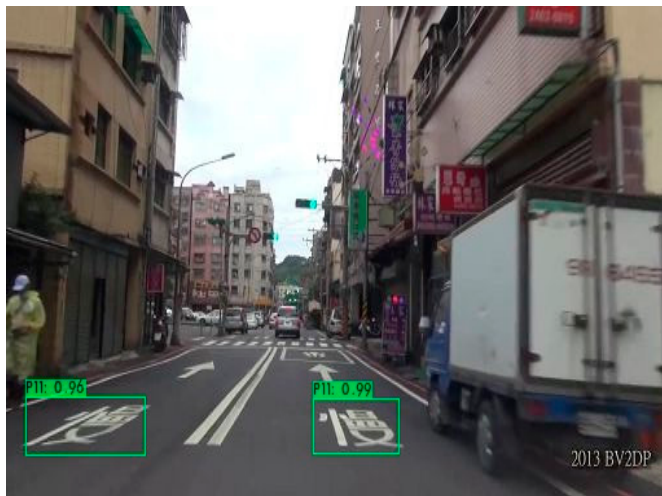


(a) Yolo V2

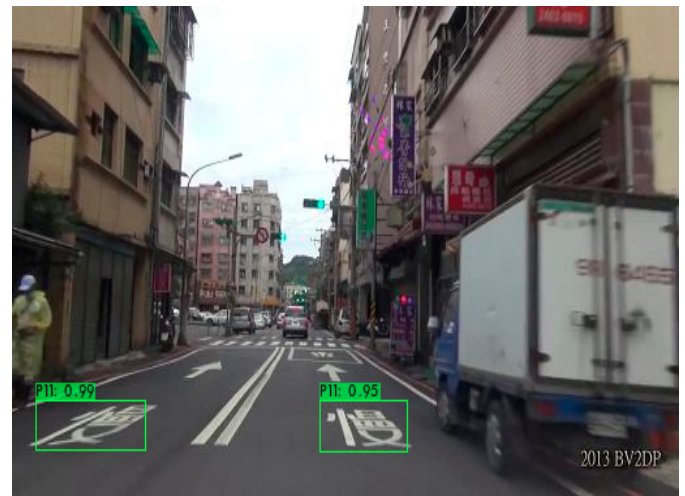


(b) Yolo V3

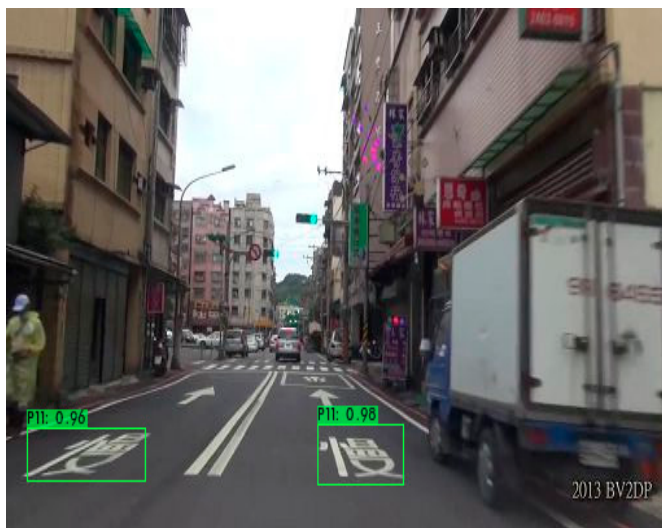
Figure 5. Cont.



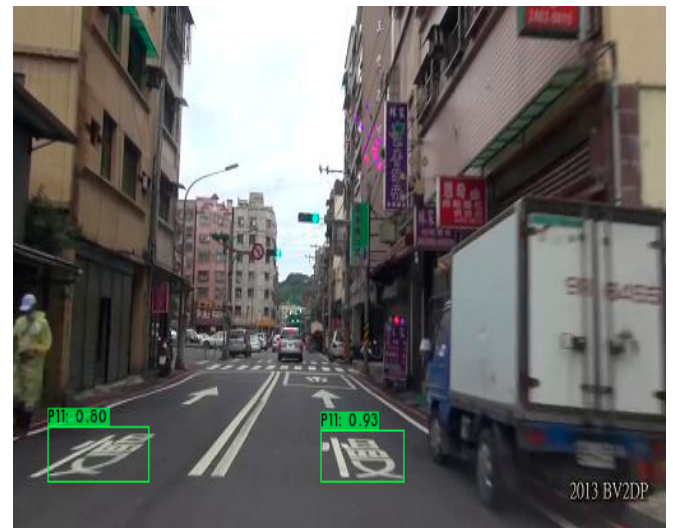
(c) Yolo V4



(d) Yolo V4 (No Flip)

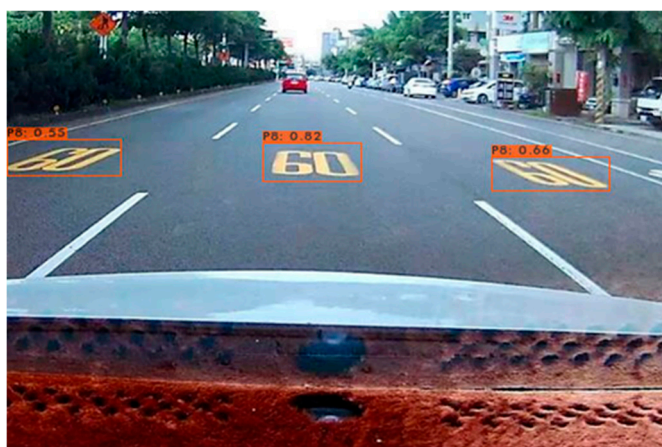


(e) Yolo V4-tiny



(f) Yolo V4-tiny (No Flip)

Figure 5. The road marking sign class P11 recognition results. (a) Yolo V2, (b) Yolo V3, (c) Yolo V4, (d) Yolo V4 (No Flip), (e) Yolo V4-tiny, (f) Yolo V4-tiny (No Flip).



(a) Yolo V2



(b) Yolo V3

Figure 6. Cont.



(c) Yolo V4



(d) Yolo V4 (No Flip)



(e) Yolo V4-tiny



(f) Yolo V4-tiny (No Flip)

Figure 6. The road marking sign class P8 recognition results. (a) Yolo V2, (b) Yolo V3, (c) Yolo V4, (d) Yolo V4 (No Flip), (e) Yolo V4-tiny, (f) Yolo V4-tiny (No Flip).



(a) Yolo V2



(b) Yolo V3

Figure 7. Cont.



(c) Yolo V4



(d) Yolo V4 (No Flip)



(e) Yolo V4-tiny



(f) Yolo V4-tiny (No Flip)

Figure 7. The road marking sign misrecognition results. (a) Yolo V2, (b) Yolo V3, (c) Yolo V4, (d) Yolo V4 (No Flip), (e) Yolo V4-tiny, (f) Yolo V4-tiny (No Flip).

5. Conclusions

The purpose of this experiment is to provide a brief review of CNN-based object identification algorithms, with a special emphasis on the Yolo V2, Yolo V3, Yolo V4, and Yolo V4-tiny algorithms, as well as the Yolo V4-tiny algorithm. Our experimental studies examine and evaluate several state-of-the-art object detectors, including those used to detect traffic signs, among other things. The evaluation criteria measure important parameters such as the mean acquisition time (mAP), the detection time (IoU), and the number of BFLOPS.

Based on the results of our investigation, we came up with the following summary: (1) Yolo V4 and Yolo V4-tiny results can take advantage of the “No Flip” setting. In our scenario, we want the model to discriminate between Left and Right objects as distinct classes. (2) The best model in the experiment is Yolo V4 (No Flip), with a testing accuracy of 95.43% and *IoU* 66.12%. (3) We build our Taiwan Road Marking Sign Dataset (TRMSD). In the future, we will combine road marking recognition with explainable artificial intelligence (XAI) and test with another dataset. Furthermore, we will be upgrading our TRMSD dataset and focusing on pothole sign recognition.

Author Contributions: Conceptualization, C.D., H.Y. and X.J.; data curation, R.-C.C. and Y.-C.Z.; formal analysis, C.D.; investigation, C.D.; methodology, C.D. and H.Y.; project administration, R.-C.C. and H.Y.; resources, Y.-C.Z.; software, Y.-C.Z.; supervision, R.-C.C.; validation, C.D. and X.J.; visualization, H.Y.; writing—original draft, C.D.; writing—review and editing, C.D., X.J. and R.-C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is supported by the Ministry of Science and Technology, Taiwan. The Nos are MOST-110-2927-I-324-50, MOST-110-2221-E-324-010 and MOST-111-2221-E-324-020, Taiwan. Additionally, this study was partially funded by the EU Horizon 2020 program RISE Project UL-TRACEPT under Grant 778062.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Acknowledgments: The author would like to thank all colleagues from the Chaoyang University of Technology and Satya Wacana Christian University, Indonesia, and all involved in this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yu, N.; Yang, R.; Huang, M. Deep Common Spatial Pattern Based Motor Imagery Classification with Improved Objective Function. *Int. J. Netw. Dyn. Intell.* **2022**, *1*, 73–84. [\[CrossRef\]](#)
2. Heidari, A.; Navimipour, N.J.; Unal, M.; Zhang, G. Machine Learning Applications in Internet-of-Drones: Systematic Review, Recent Deployments, and Open Issues. *ACM Comput. Surv.* **2022**, *55*, 1–45. [\[CrossRef\]](#)
3. Wontorczyk, A.; Gaca, S. Study on the Relationship between Drivers' Personal Characters and Non-Standard Traffic Signs Comprehensibility. *Int. J. Environ. Res. Public Health* **2021**, *18*, 1–19. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Shakiba, F.M.; Shojaee, M.; Azizi, S.M.; Zhou, M. Real-Time Sensing and Fault Diagnosis for Transmission Lines. *Int. J. Netw. Dyn. Intell.* **2022**, *1*, 36–47. [\[CrossRef\]](#)
5. Poggenhans, F.; Schreiber, M.; Stiller, C. A Universal Approach to Detect and Classify Road Surface Markings. In Proceedings of the IEEE Conference on Intelligent Transportation Systems, ITSC, Las Palmas de Gran Canaria, Spain, 15–18 September 2015; pp. 1915–1921.
6. Marcus, G.; Davis, E. Insights for AI from the Human Mind. *Commun. ACM* **2021**, *64*, 38–41. [\[CrossRef\]](#)
7. Su, Y.; Cai, H.; Huang, J. The Cooperative Output Regulation by the Distributed Observer Approach. *Int. J. Netw. Dyn. Intell.* **2022**, *1*, 20–35. [\[CrossRef\]](#)
8. Heidari, A.; Navimipour, N.J.; Unal, M. A Secure Intrusion Detection Platform Using Blockchain and Radial Basis Function Neural Networks for Internet of Drones. *IEEE Internet Things J.* **2023**, *1*. [\[CrossRef\]](#)
9. Danescu, R.; Nedevschi, S. Detection and Classification of Painted Road Objects for Intersection Assistance Applications. In Proceedings of the IEEE Conference on Intelligent Transportation Systems, ITSC, Madeira Island, Portugal, 19–22 September 2010; pp. 433–438.
10. Redmon, J.; Farhadi, A. YOLO v.3. *Tech Rep.* **2018**, 1–6. [\[CrossRef\]](#)
11. Zhai, S.; Shang, D.; Wang, S.; Dong, S. DF-SSD: An Improved SSD Object Detection Algorithm Based on DenseNet and Feature Fusion. *IEEE Access* **2020**, *8*, 24344–24357. [\[CrossRef\]](#)
12. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot Multibox Detector. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
13. Zhao, G.; Li, Y.; Xu, Q. From Emotion AI to Cognitive AI. *Int. J. Netw. Dyn. Intell.* **2022**, *1*, 65–72. [\[CrossRef\]](#)
14. Wang, X.; Sun, Y.; Ding, D. Adaptive Dynamic Programming for Networked Control Systems Under Communication Constraints: A Survey of Trends and Techniques. *Int. J. Netw. Dyn. Intell.* **2022**, *1*, 85–98. [\[CrossRef\]](#)
15. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. Scaled-Yolov4: Scaling Cross Stage Partial Network. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13024–13033.
16. Marfia, G.; Roccetti, M. TCP at Last: Reconsidering TCP's Role for Wireless Entertainment Centers at Home. *IEEE Trans. Consum. Electron.* **2010**, *56*, 2233–2240. [\[CrossRef\]](#)
17. Song, W.; Suandi, S.A. TSR-YOLO: A Chinese Traffic Sign Recognition Algorithm for Intelligent Vehicles in Complex Scenes. *Sensors* **2023**, *23*, 749. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Yahyaouy, A.; Sabri, A.; Benjelloun, F.; El Manaa, I.; Aarab, A. Autonomous Approach for Moving Object Detection and Classification in Road Applications. *Int. J. Comput. Aided Eng. Technol.* **2023**, *1*, 1. [\[CrossRef\]](#)
19. Vokhidov, H.; Hong, H.G.; Kang, J.K.; Hoang, T.M.; Park, K.R. Recognition of Damaged Arrow-Road Markings by Visible Light Camera Sensor Based on Convolutional Neural Network. *Sensors* **2016**, *16*, 2160. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Dewi, C.; Chen, R.; Liu, Y.; Yu, H. Various Generative Adversarial Networks Model for Synthetic Prohibitory Sign Image Generation. *Appl. Sci.* **2021**, *11*, 2913. [\[CrossRef\]](#)
21. Kheyrollahi, A.; Breckon, T.P. Automatic Real-Time Road Marking Recognition Using a Feature Driven Approach. *Mach. Vis. Appl.* **2012**, *23*, 123–133. [\[CrossRef\]](#)
22. Ding, D.; Yoo, J.; Jung, J.; Jin, S.; Kwon, S. Efficient Road-Sign Detection Based on Machine Learning. *Bull. Networking, Comput. Syst. Softw.* **2015**, *4*, 15–17.

23. Salome, G.F.P.; Chela, J.L.; Junior, J.C.P. Fraud Detection with Machine Learning—Model Comparison. *Int. J. Bus. Intell. Data Min.* **2023**, *1*, 1. [\[CrossRef\]](#)
24. Yang, W.; Zhang, W. Real-Time Traffic Signs Detection Based on YOLO Network Model. In Proceedings of the 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2020, Chongqing, China, 29–30 October 2020.
25. Mijic, D.; Brisinello, M.; Vranjes, M.; Grbic, R. Traffic Sign Detection Using YOLOv3. In Proceedings of the IEEE International Conference on Consumer Electronics, ICCE-Berlin, Berlin, Germany, 9–11 November 2020.
26. Gatelli, L.; Gosmann, G.; Fitarelli, F.; Huth, G.; Schwertner, A.A.; De Azambuja, R.; Brusamarello, V.J. Counting, Classifying and Tracking Vehicles Routes at Road Intersections with YOLOv4 and DeepSORT. In Proceedings of the INSCIT 2021—5th International Symposium on Instrumentation Systems, Circuits and Transducers, Virtual, 23–27 August 2021.
27. Chen, T.; Chen, Z.; Shi, Q.; Huang, X. Road Marking Detection and Classification Using Machine Learning Algorithms. In Proceedings of the IEEE Intelligent Vehicles Symposium, Seoul, Republic of Korea, 28 June–1 July 2015; pp. 617–621.
28. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
29. Dewi, C.; Chen, R.C.; Yu, H.; Jiang, X. Robust Detection Method for Improving Small Traffic Sign Recognition Based on Spatial Pyramid Pooling. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 1–18. [\[CrossRef\]](#)
30. Dewi, C.; Chen, R.C.; Yu, H. Weight Analysis for Various Prohibitory Sign Detection and Recognition Using Deep Learning. *Multimed. Tools Appl.* **2020**, *79*, 32897–32915. [\[CrossRef\]](#)
31. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [\[CrossRef\]](#)
32. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [\[CrossRef\]](#)
33. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 22–25 July 2017; pp. 6517–6525.
34. Springenberg, J.T.; Dosovitskiy, A.; Brox, T.; Riedmiller, M. Striving for Simplicity: The All Convolutional Net. *arXiv* **2014**, arXiv:1412.6806.
35. Xu, H.; Srivastava, G. Automatic Recognition Algorithm of Traffic Signs Based on Convolution Neural Network. *Multimed. Tools Appl.* **2020**, *79*, 11551–11565. [\[CrossRef\]](#)
36. Hu, Y.; Wu, X.; Zheng, G.; Liu, X. Object Detection of UAV for Anti-UAV Based on Improved YOLO V3. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8386–8390.
37. Mittal, U.; Chawla, P.; Tiwari, R. EnsembleNet: A Hybrid Approach for Vehicle Detection and Estimation of Traffic Density Based on Faster R-CNN and YOLO Models. *Neural Comput. Appl.* **2023**, *35*, 4755–4774. [\[CrossRef\]](#)
38. Wu, Y.; Li, Z.; Chen, Y.; Nai, K.; Yuan, J. Real-Time Traffic Sign Detection and Classification towards Real Traffic Scene. *Multimed. Tools Appl.* **2020**, *79*, 18201–18219. [\[CrossRef\]](#)
39. Chen, Q.; Liu, L.; Han, R.; Qian, J.; Qi, D. Image Identification Method on High Speed Railway Contact Network Based on YOLO v3 and SENet. In Proceedings of the Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8772–8777.
40. Corovic, A.; Ilic, V.; Duric, S.; Marijan, M.; Pavkovic, B. The Real-Time Detection of Traffic Participants Using YOLO Algorithm. In Proceedings of the 26th Telecommunications Forum, TELFOR 2018, Belgrade, Serbia, 20–21 November 2018.
41. Bochkovskiy, A.; Wang, C.-Y.; Mark Liao, H.-Y. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:arXiv.2004.10934. [\[CrossRef\]](#)
42. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. GoogLeNet Going Deeper with Convolutions. *arXiv* **2014**, arXiv:1409.4842. [\[CrossRef\]](#)
43. Dewi, C.; Chen, R.-C.; Jiang, X.; Yu, H. Deep Convolutional Neural Network for Enhancing Traffic Sign Recognition Developed on Yolo V4. *Multimed. Tools Appl.* **2022**, *81*, 1–25. [\[CrossRef\]](#)
44. Otgonbold, M.E.; Gochoo, M.; Alnajjar, F.; Ali, L.; Tan, T.H.; Hsieh, J.W.; Chen, P.Y. SHEL5K: An Extended Dataset and Benchmarking for Safety Helmet Detection. *Sensors* **2022**, *22*, 2315. [\[CrossRef\]](#)
45. Sun, X.M.; Zhang, Y.J.; Wang, H.; Du, Y.X. Research on Ship Detection of Optical Remote Sensing Image Based on Yolo V5. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2022; Volume 2215.
46. Wang, G.; Ding, H.; Yang, Z.; Li, B.; Wang, Y.; Bao, L. TRC-YOLO: A Real-Time Detection Method for Lightweight Targets Based on Mobile Devices. *IET Comput. Vis.* **2022**, *16*, 126–142. [\[CrossRef\]](#)
47. Chien-Yao, W.; Bochkovskiy, A.; Hong-Yuan, L.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *arXiv* **2022**, arXiv:arXiv.2207.02696. [\[CrossRef\]](#)
48. Github. Ultralytics Yolo V5. Available online: <https://github.com/ultralytics/yolov5> (accessed on 13 January 2021).
49. Chen, Y.W.; Shiu, J.M. An Implementation of YOLO-Family Algorithms in Classifying the Product Quality for the Acrylonitrile Butadiene Styrene Metallization. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 8257–8269. [\[CrossRef\]](#) [\[PubMed\]](#)
50. Gao, G.; Wang, S.; Shuai, C.; Zhang, Z.; Zhang, S.; Feng, Y. Recognition and Detection of Greenhouse Tomatoes in Complex Environment. *Trait. du Signal* **2022**, *39*, 291–298. [\[CrossRef\]](#)
51. Sui, J.-Y.; Liao, S.; Li, B.; Zhang, H.-F. High Sensitivity Multitasking Non-Reciprocity Sensor Using the Photonic Spin Hall Effect. *Opt. Lett.* **2022**, *47*, 6065. [\[CrossRef\]](#)

52. Wan, B.F.; Zhou, Z.W.; Xu, Y.; Zhang, H.F. A Theoretical Proposal for a Refractive Index and Angle Sensor Based on One-Dimensional Photonic Crystals. *IEEE Sens. J.* **2021**, *21*, 331–338. [[CrossRef](#)]
53. Arcos-García, Á.; Álvarez-García, J.A.; Soria-Morillo, L.M. Evaluation of Deep Neural Networks for Traffic Sign Detection Systems. *Neurocomputing* **2018**, *316*, 332–344. [[CrossRef](#)]
54. Li, Z.; Tian, X.; Liu, X.; Liu, Y.; Shi, X. A Two-Stage Industrial Defect Detection Framework Based on Improved-YOLOv5 and Optimized-Inception-ResnetV2 Models. *Appl. Sci.* **2022**, *12*, 834. [[CrossRef](#)]
55. Yang, H.; Chen, L.; Chen, M.; Ma, Z.; Deng, F.; Li, M.; Li, X. Tender Tea Shoots Recognition and Positioning for Picking Robot Using Improved YOLO-V3 Model. *IEEE Access* **2019**, *7*, 180998–181011. [[CrossRef](#)]
56. Yuan, Y.; Xiong, Z.; Wang, Q. An Incremental Framework for Video-Based Traffic Sign Detection, Tracking, and Recognition. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1918–1929. [[CrossRef](#)]
57. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple Detection during Different Growth Stages in Orchards Using the Improved YOLO-V3 Model. *Comput. Electron. Agric.* **2019**, *157*, 417–426. [[CrossRef](#)]
58. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.