

Article

# Deep Learning for Traffic Sign Recognition Based on Spatial Pyramid Pooling with Scale Analysis

Shao-Kuo Tai <sup>1</sup>, Christine Dewi <sup>1,2,\*</sup>, Rung-Ching Chen <sup>1,\*</sup>, Yan-Ting Liu <sup>1</sup>, Xiaoyi Jiang <sup>3</sup>  
and Hui Yu <sup>4</sup>

<sup>1</sup> Department of Information Management, Chaoyang University of Technology, Taichung City 41349, Taiwan; sgdai@cyut.edu.tw (S.-K.T.); s10414141@cyut.edu.tw (Y.-T.L.)

<sup>2</sup> Faculty of Information Technology, Satya Wacana Christian University, Salatiga 50711, Indonesia

<sup>3</sup> Department of Mathematics and Computer Science, University of Münster, D-48149 Münster, Germany; xjiang@uni-muenster.de

<sup>4</sup> School of Creative Technologies, The University of Portsmouth, Portsmouth PO1 2UP, UK; hui.yu@port.ac.uk

\* Correspondence: s10714904@cyut.edu.tw (C.D.); crching@cyut.edu.tw (R.-C.C.)

Received: 12 September 2020; Accepted: 2 October 2020; Published: 7 October 2020



**Abstract:** In the area of traffic sign detection (TSD) methods, deep learning has been implemented and achieves outstanding performance. The detection of a traffic sign, as it has a dual function in monitoring and directing the driver, is a big concern for driver support systems. A core feature of autonomous vehicle systems is the identification of the traffic sign. This article focuses on the prohibitive sign. The objective is to detect in real-time and reduce processing time considerably. In this study, we implement the spatial pyramid pooling (SPP) principle to boost Yolo V3's backbone network for the extraction of functionality. Our work uses SPP for more comprehensive learning of multiscale object features. Then, perform a comparative investigation of Yolo V3 and Yolo V3 SPP across various scales to recognize the prohibitory sign. Comparisons with Yolo V3 SPP models reveal that their mean average precision (*mAP*) is higher than Yolo V3. Furthermore, the test accuracy findings indicate that the Yolo V3 SPP model performs better than Yolo V3 for different sizes.

**Keywords:** TSD; object detection; TSR; Yolo V3; SPP; scale analysis

## 1. Introduction

Traffic sign recognition (TSR) technologies are an essential feature of numerous real-world implementations, including Automated Driver Assistance Systems (ADAS) [1,2], autonomous driving, traffic control, driver welfare, and maintenance of the road network. Many researchers are currently working on this problem with popular computer vision algorithms [3]. The emergence of recent improvements in deep learning [4] has contributed to the significant advance for target detection [5–7] and identification tasks [8–10]. Moreover, most studies centered on creating profound convolutional neural networks (CNN) to increase precision [11,12].

The reason that traffic signs are created to be different and recognizable, using basic types and standardized colors accordingly to their country-specific existence, suggests a limiting issue in their identification and recognition. A method that generalizes efficient identification is difficult to find [1]. Nonetheless, it is still a challenge to develop a stable real-time TSR. During test time, latency is critical for decision-making depending on the atmosphere and real-life factors, such as partial occlusion, multiple views, illuminations, and temperature. Every TSR needs to address these problems well. This research will concentrate on the prohibitive identification and understanding of signs in Taiwan. The inspiration is the absence of a traffic sign detection database or analysis system in Taiwan. The most excellent advanced algorithms for object detection like SSD [13,14], Faster R-CNN [15,16], R-FCN [17],

and Yolo [18,19] already used convolutional neural network (CNN) that can be used in handheld devices and consumer goods. Yolo has been a successful CNN rival in real-time object detection [20]. Research [21,22] reveals that high-speed detection efficiency has been closely tracked to identify smaller objects. Yolo's latest update, Yolo V3 [23] and Yolo V4 [24], have exhibited important development in that ability. Yolo V3 is selected as one of the main objectives in real-time recognition as the object identification system. This paper aims to eliminate the network fixed-size restriction, obtain the best features in max-pooling layers, and enhance Yolo V3 performance and the layer is established by the SPP layer [25–28].

This paper's major contributions are: (1) Follow the principle of SPP to strengthen the original structure of Yolo V3 for building features extraction and determining maximum information flow between layers in the system; (2) Use spatial pyramids to organize and aggregate local features at various levels in the same layer for more detailed training of multiple scale object functions; (3) A comparative study of Yolo V3 and Yolo V3 SPP on various scales for the identification of prohibited signs in Taiwan are carried out; (4) The result suggests that the detection time would be quicker if a big number of scales is used. Therefore, the precision declines relative to the original Yolo V3 scale. Similarly, in Yolo V3 SPP, the accuracy declines to take on various sizes.

The following parts of this paper describe the proposed model. The related work of traffic sign identification systems explains in the Section 2. The approach methodology briefly discusses in Section 3. Furthermore, Section 4 includes a summary, the preparation, and system test outcomes. In Section 5, assumptions are established, and further study is introduced.

## 2. Materials and Methods

### 2.1. Traffic Sign Recognition with You Only Look Once (Yolo) V3

Based on [29], this research work combines Adaboost, and Yolo V2 approaches for traffic sign studies. The system uses real traffic signs collected in the center of Kaohsiung, a large city in southern Taiwan. Additional research on traffic signs, particularly in Taiwan, is presented in [30]. This work tracks traffic signs from video recordings using its proposed program for obtaining the traffic signs image. CNN validates the precision of the generated dataset.

In [31], focus on Taiwan's stop sign detection and recognition. They conduct some experiments with a different setting and analyze the importance of anchor calculation using k-means and original Yolo V3 for Taiwan stop sign detection and recognition. Their experiment proved that anchor recalculation based on our dataset is very important.

Dewi et al. [28] investigates the state-of-the-art of various object detection systems including Yolo V3, Resnet 50, Densenet, and Tiny Yolo V3 combined with spatial pyramid pooling (SPP). Their research adopts the concept of SPP to improve the backbone network of Yolo V3, Resnet 50, Densenet, and Tiny YoloV3. Hence, their experiment findings show that Yolo V3 SPP strikes the best total BFLOPS (65.69), and *mAP* (98.88%). The highest average accuracy is Yolo V3 SPP at 99%, followed by Densenet SPP at 87%, Resnet 50 SPP at 70%, and Tiny Yolo V3 SPP at 50%. Hence, SPP can improve the performance of all models in the experiment.

Other research studied various weights presented by the darknet framework, including the best weight, the final weight, and the last weight [31]. They conduct and analyze the comparative experiment of Yolo V3 and Yolo V3 SPP with different weights. Experimental results show that the mean average precision (*mAP*) of Yolo V3 SPP is better than other models.

Based on the previous research work we found that nobody focused on the significant of scale parameter of Yolo in the configuration file. In our research will concentrate more on the importance of scale parameters in the Yolo V3 and Yolo V3 SPP configuration file.

Yolo V3 was introduced the first time by Redmon et al. [32,33] in 2016. A single neural network interprets the entire picture. Yolo V3 separated the image into grid cells and provides boundary boxes and possibilities for each grid cell [34]. Yolo V3 makes a prediction using multiscale fusion.

The  $416 \times 416$  input image size is integrated with three scales using up-sample and FPN fusion [35]. The three scales obtained are  $13 \times 13$ ,  $26 \times 26$ , and  $52 \times 52$ , respectively [36].

Yolo V3 consists of 53 layers with deep characteristics and was built on Darknet-53. Yolo V3 has demonstrated better than ResNet-101, ResNet-152, or Darknet-19 [33]. Figure 1 exhibits the construction of Darknet-53. The input image is divided by the Yolo V3 algorithm into  $S \times S$  grids. If the central point of the ground reality of the object decreases within the required grid, the grid can define the target. Each grid outputs B bounding prediction boxes, including bounding box location data that consist of coordinates of the middle point ( $x, y$ ), width ( $w$ ), height ( $h$ ), and confidence prediction.

	Type	Filter Size	Size	Output
	Convolutional	32	$3 \times 3$	$256 \times 256$
	Convolutional	64	$3 \times 3/2$	$128 \times 128$
1x	Convolutional	32	$1 \times 1$	$128 \times 128$
	Convolutional	64	$3 \times 3$	
	Residual			
	Convolutional	128	$3 \times 3/2$	$64 \times 64$
2x	Convolutional	64	$1 \times 1$	$64 \times 64$
	Convolutional	128	$3 \times 3$	
	Residual			
	Convolutional	256	$3 \times 3/2$	
8x	Convolutional	128	$1 \times 1$	$32 \times 32$
	Convolutional	256	$3 \times 3$	
	Residual			
	Convolutional	512	$3 \times 3/2$	$16 \times 16$
8x	Convolutional	256	$1 \times 1$	$16 \times 16$
	Convolutional	512	$3 \times 3$	
	Residual			
	Convolutional	1024	$3 \times 3/2$	$8 \times 8$
4x	Convolutional	512	$1 \times 1$	$8 \times 8$
	Convolutional	1024	$3 \times 3$	
	Residual			
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 1. Darknet-53 Structure.

The Yolo loss function of the boundary box consists of four sections [37], and the formula could be seen on Equation (1) [38–40].

$$Loss = Coord\_Err + BBox\_Err + Category\_Err + Conf\_Err \tag{1}$$

Further,  $Coord\_Err$  is the loss of predicted central coordinate, and  $BBox\_Err$  is the loss of width and height of the prediction bounding box. Next,  $Category\_Err$  is the loss of the predicted category, and  $Conf\_Err$  is the loss of the predicted confidence. The process of measurement is shown in Equations (2)–(5).

$$Coord\_Err = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B I_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \tag{2}$$

$$BBox\_Err = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B I_{ij}^{obj} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \quad (3)$$

$$Category\_Err = \sum_{i=0}^{s^2} I_{ij}^{obj} \sum_{c \in class} (p_i(c), \hat{p}_i(c))^2 \quad (4)$$

$$Conf\_Err = \sum_{i=0}^{s^2} \sum_{j=0}^B I_{ij}^{obj} (c_i - \hat{c}_i)^2 + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^B I_{ij}^{obj} (c_i - \hat{c}_i)^2 \quad (5)$$

Moreover,  $(x_i, y_i)$  is the position of the prediction bounding box.  $(\hat{x}_i, \hat{y}_i)$  is the actual position obtained from the training data.  $w_i$  and  $h_i$  are the width and height of the predicted bounding box, respectively.  $\lambda_{coord}$  is to control the prediction position loss of the prediction box.  $\lambda_{noobj}$  is to control the no target loss in a single grid.  $c_i$  is the confidence score.  $\hat{c}_i$  is the intersection part of the predicted bounding box and the actual box.

Further, Yolo V3 employs the sigmoid function as a tool for predicting the activation function. The sigmoid function solves the problem efficiently, while the equal target has two labels [39,41,42].

### 2.2. Spatial Pyramid Pooling (SPP) Network

Spatial Pyramid Pooling (SPP) [25,26] is one of computer vision’s most popular approaches. Spatial Pyramid Pooling (SPM) is commonly referred to as SPP and Bag-of-Words (BOW) model development [43]. SPP [24] belongs to an essential feature of leading and competitive classification schemes [44–46] and detection [47] before the current rise of CNN.

Some advantages of SPP are given in [27]. First, the SPP provides a fixed output despite input dimensions, whereas sliding window is not possible in preceding systems [48]. Second, SPP applies multi-level room cabinets and the pooling of sliding windows requires just one window. Since input dimensions are versatile, SPP can incorporate functionality obtained at variable dimensions. Figure 2 indicates a network configuration for an SPP network. This work placed the SPP block in the configuration file of Yolo V3. In the SPP layer, the final convolutional feature maps’ outcome is classified into spatial bins in proportional sizes. The number of bins is fixed despite the dimensions of the image.

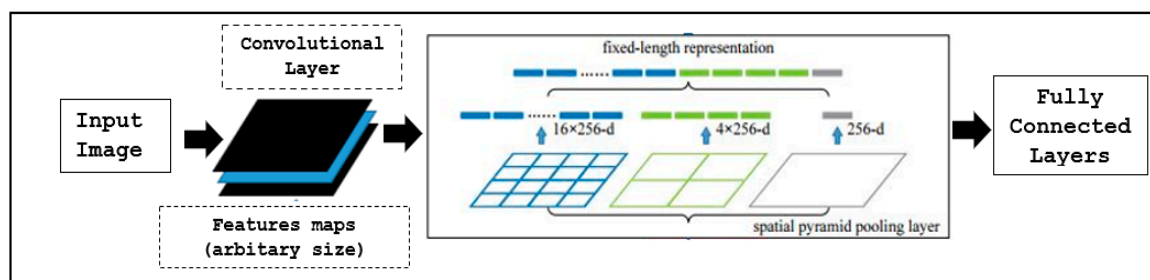


Figure 2. SPP network layers.

### 2.3. Yolo V3 SPP Architecture

This segment outlines the proposed technique for detecting and identifying road signs from Taiwan using Yolo V3 with SPP. Figure 3 describes the Yolo V3 SPP architecture. Object detection using Yolo V3 SPP proceeds as follows. The initial stage separates the image input into  $S \times S$  grids. Each grid generates  $K$  bounders according to the calculation of the anchor boxes. The framework then implements the CNN for extracting all object characteristics from the picture and forecast the  $b = [b_x, b_y, b_w, b_h, b_c]^T$  and the  $class = [class_1, class_2, \dots, class_c]^T$ . Afterward, it compares the maximum confidence  $IoU_{pred}^{truth}$  of the  $K$  bounding boxes with the threshold  $IoU_{thres}$ . If  $IoU_{pred}^{truth} > IoU_{thres}$ , meaning that the bounding

box contains the object. Otherwise, the bounding box does not contain the object. Next, the system then selects the category with the highest predicted probability as the object category. Finally, for performing a maximum local exploration, for suppressing redundant boxes, output, and displaying the results of object detection, this experiment employs Non-Maximum Suppression (NMS).

In the research, Yolo V3 SPP uses convolutional layer sampling to achieve the max-pool layers' best possible functionality. Yolo V3 SPP employs three scales of the max pool for all images using *[route]*. Various layers -2, -4 and -1, -3, -5, -6 in *conv<sub>5</sub>* were used in each *[route]*. Moreover, *conv<sub>5</sub>* is the final layer of convolution and 256 is the *conv<sub>5</sub>* layer filter number. These created feature maps, called fixed-length representations, are then collected (see Figure 2). This experiment compares the performance of Yolo V3 and Yolo V3 SPP at different scales. SoftMax classification layers and boundary box regression are initialized in the Gaussian zero-mean distributions with standard deviations of 0.01 and 0.001. The global learning rate is 0.001, momentum is 0.9, and the parameter decay is 0.0005. The learning rate parameter determines how vigorously the latest batch of data can be used for learning.

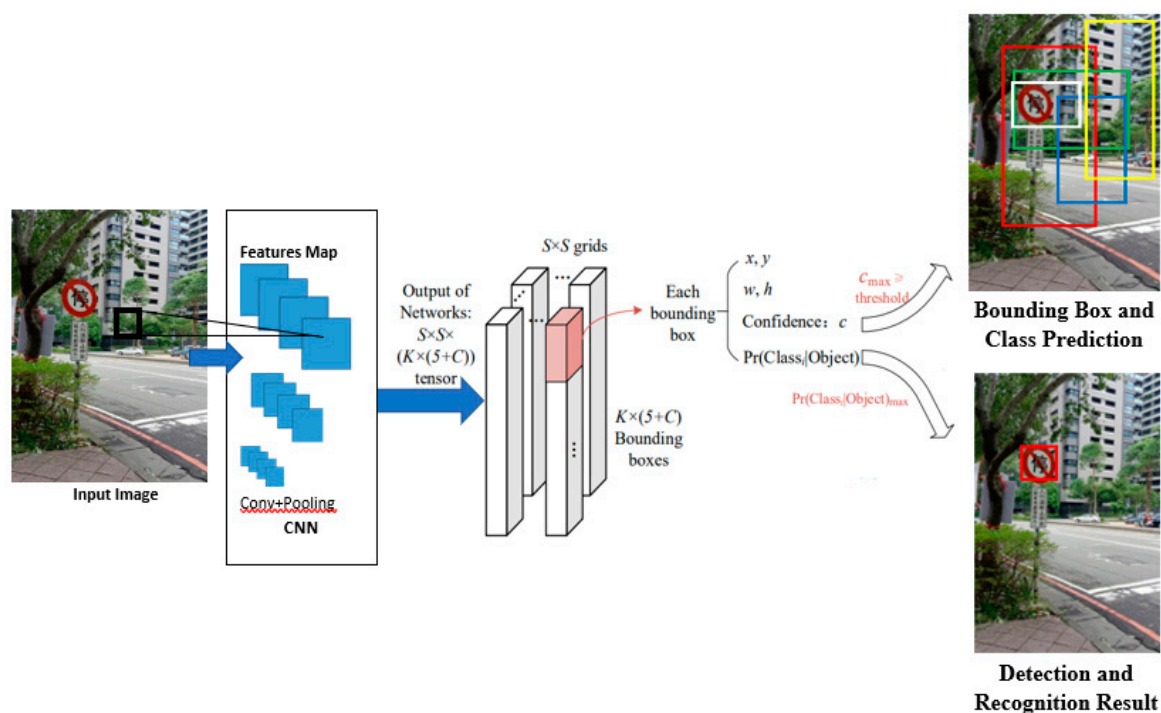


Figure 3. Yolo V3 SPP architecture.

The work arranges six models, Yolo V3 1, Yolo V3 2, Yolo V3 3, Yolo V3 SPP 1, Yolo V3 SPP 2, Yolo V3 SPP 3. This work uses different scales of (0.1, 0.1), (0.2, 0.2), and (0.3, 0.3) for each Yolo V3 and Yolo V3 SPP. An n-classes object detector should run the training for at most limited  $2000 \times n$  batches. In the experiment, four classes have 8000 iterations for maximum batches. It means that the training will be processed until 8000 iterations. For example, the scale = 0.1, 0.1 and the current iteration number are 10,000 (0.001) batches so the system can calculate the current learning rate = learning rate  $\times$  scales [0]  $\times$  scales [1] =  $0.001 \times 0.1 \times 0.1 = 0.00001$ .

#### 2.4. Prohibitory Sign and Object Detection

The Yolo V3 system is used to detect and identify for prohibitory signs in Taiwan in one step. The system starts by making a boundary box for each sign with the BBox label tool for training [49]. The method of labeling is done with four type marks. More than one bounding box can host an image. In this stage, one class detector model is used, where a symbol is a single model of training. Object coordinates in the form  $(x_1, y_1, x_2, y_2)$  are the bounding box marking tool's output value.



This output is not in the form of the Yolo object coordinates format. Yolo's input value is the central point and the width and height of the object ( $x, y, w, h$ ). Therefore, the system must transform the bounding box coordinate into the input model for Yolo. The conversion process is shown in Equations (6)–(9).

$$dw = 1/w, x = \frac{(x_1 + x_2)}{2} \times dw, dh = 1/h \quad (6)$$

$$y = \frac{(y_1 + y_2)}{2} \times dh \quad (7)$$

$$w = (x_2 - x_1) \times dw \quad (8)$$

$$h = (y_2 - y_1) \times dh \quad (9)$$





Further,  $w$  is the image width,  $dw$  is the absolute image width,  $h$  is the image height, and  $dh$  is the total image height. Float values of the image width and height ( $dw, dh$ ) can also be similar to 0.0 to 1.0.

### 3. Results

#### 3.1. Dataset

In this work, we collected and processed traffic sign images manually from CarMax dashboard camera footage while driving on a sunny day and at night around Taichung City. The camera images, from which the traffic sign images are extracted, have a resolution of  $1920 \times 1080$  pixels. We also used the Oppo F5 mobile phone camera to collect the traffic sign images with a resolution of  $1080 \times 2160$  pixels. The traffic sign images are cropped and annotated before use for training. Furthermore, the concentration is on the prohibition sign, including 235 no entry images, 250 no stopping images, 185-speed limit images, and 230 no parking images. The data collection is separated into 70 percent for training and 30 percent for testing [28]. Further, 900 images are shown in Table 1. in this work.

**Table 1.** Taiwan Prohibitory Signs.

ID	Name	Sign
P1	No entry	
P2	No stopping	
P3	No parking	
P4	Speed Limit	

#### 3.2. Training Results

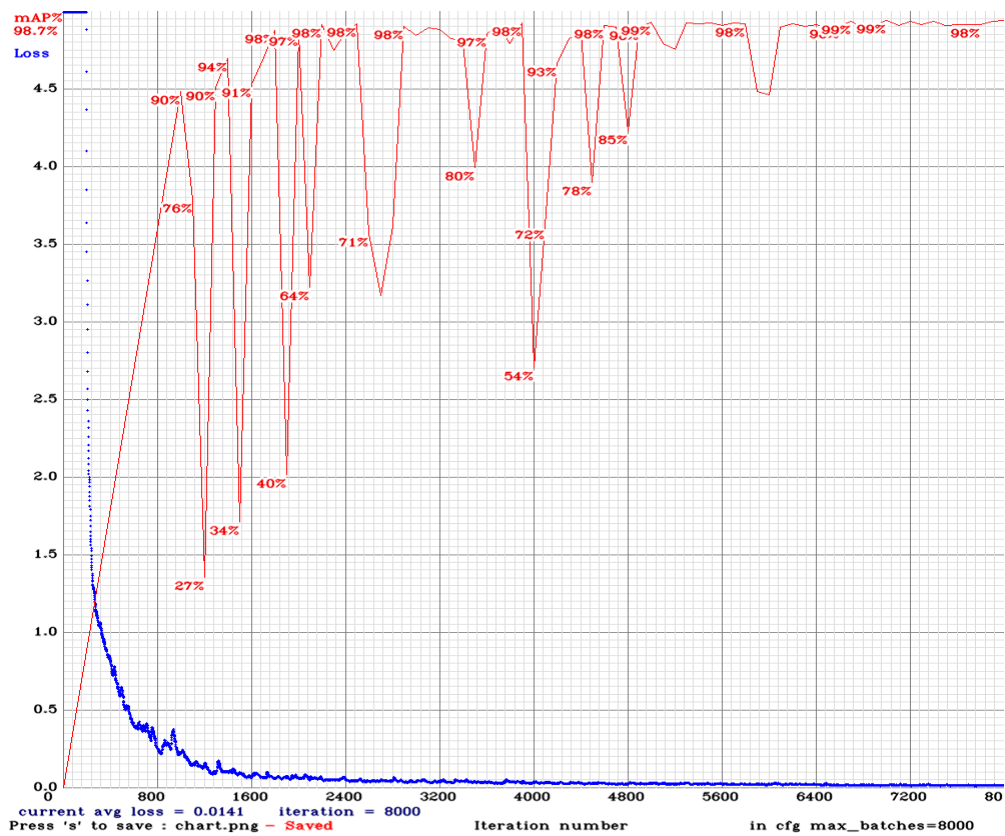
Data augmentation is a significant element of the advancement of deep learning models. While the data augmentation has been shown to enhance the classification of images significantly, object identification has not been extensively studied [50]. Additionally, data augmentation is a famous method widely employed to improve the training process of CNN. The system is applied pre-processing steps, including data augmentation in the training stage. Therefore, during data augmentation, the system performs several operations, such as rotation with a probability of 0.5 and a maximum rotation of 20 degrees for each image. Next, the zoom range was 10 percent and 0.2 for width shift and height shift range. Further, the traffic signs are identified by using a bounding box labeling tool [49] for providing a coordinate position to the object. The outcomes of the tools and the class mark are four points of the position coordinate system. Then, before training, the system will transform a label to a Yolo format label. This work applied another method, the Yolo Annotation

framework in Python programming language [51], to convert the values to a format that can be read by the Yolo V3 training algorithm. The research experiment is carried out on a computer-based on the Python environment, which applies a Nvidia RTX2080Ti GPU (11GB memory) and an i7 CPU with 16 GB DDR2 memory.

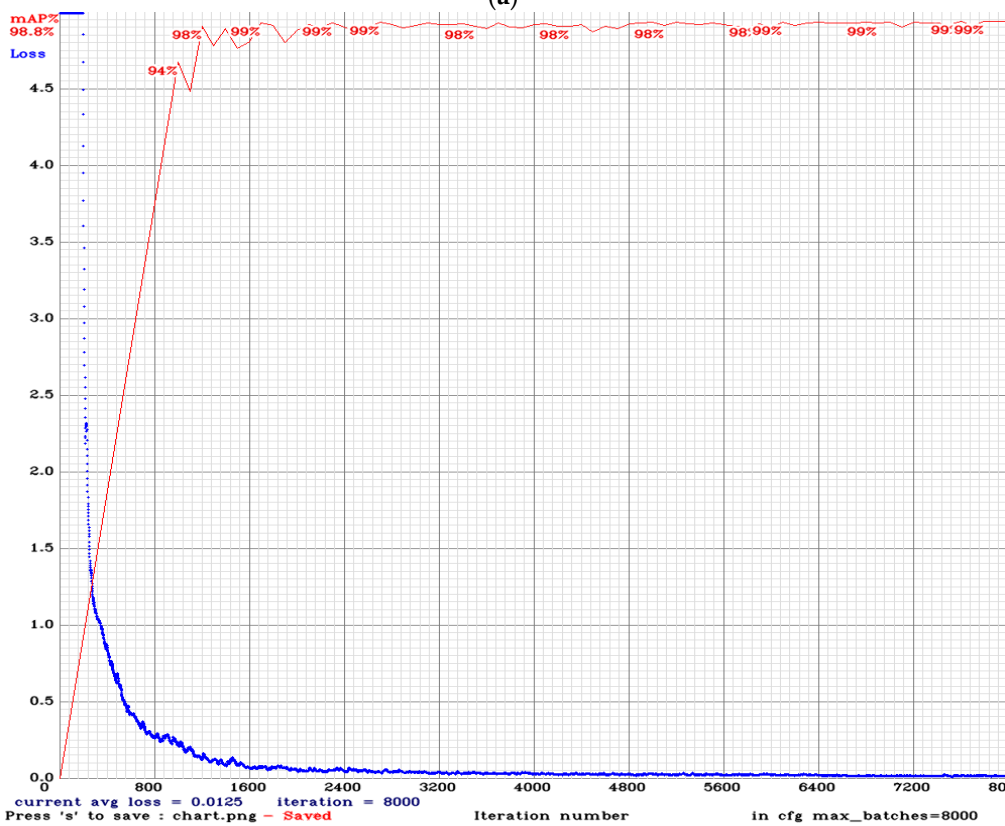
Figure 4 represents the training process's reliability using Yolo V3 1 (a) and Yolo V3 SPP 1 (b). The work uses 8000 iterations,  $policy = steps$ , and  $steps = 6400, 7200$ . Since the start has zero knowledge, the learning rate must be high at the beginning of the training phase. However, with the volume of data available in the neural network, the weights tend to adjust less vigorously. The learning rate must be lowered over time. Furthermore, this reduction in learning rates in the configuration file is made by stating first that the learning rate decreases step by step. Moreover, the learning rate begins at 0.001 and stays constant for 6400 iterations. It multiplies through percentages to get the latest standard of learning. Figure 4 shows that Yolo V3 SPP 1 is more stable than Yolo V3 1 through the training process. The detailed outcomes of the training performance are demonstrated in Table 2.

**Table 2.** Training loss value,  $mAP$ , and  $AP$  performance for all classes.

Model	Loss Value	ID	AP (%)	TP	FP	Precision	Recall	F1-score	IoU (%)	$mAP@0.50$ (%)
Yolo V3 1	0.0141	P1	97.5	77	0	0.99	0.99	0.99	82.19	98.73
		P2	98.8	83	0					
		P3	99.9	62	1					
		P4	98.74	76	2					
Yolo V3 2	0.015	P1	97.5	78	0	0.98	0.99	0.98	83.09	98.84
		P2	100	83	0					
		P3	99.85	61	3					
		P4	98.01	75	3					
Yolo V3 3	0.0129	P1	97.5	77	0	0.98	0.97	0.97	84.68	98.49
		P2	98.8	83	0					
		P3	99.92	59	0					
		P4	97.75	72	5					
Average	0.014					0.98	0.98	0.98	83.32	98.68
Yolo V3 SPP 1	0.0125	P1	97.5	78	0	0.99	0.99	0.99	90.09	98.88
		P2	98.8	83	0					
		P3	99.9	62	1					
		P4	98.94	79	3					
Yolo V3 SPP 2	0.0144	P1	97.43	78	0	0.99	0.99	0.99	89.4	99.12
		P2	100	83	0					
		P3	100	62	1					
		P4	99.05	76	2					
Yolo V3 SPP 3	0.0133	P1	97.5	78	0	0.99	0.99	0.99	88.54	98.93
		P2	100	83	0					
		P3	100	62	0					
		P4	98.23	76	2					
Average	0.0134					0.99	0.99	0.99	89.34	98.97



(a)



(b)

Figure 4. Training performance for all classes using (a) Yolo V3 1 and (b) Yolo V3 SPP 1.



Table 2 displays the loss value of training,  $mAP$ , and  $AP$  results for all classes after 8000 cycles of training. The average training validity failure is about 0.013 for both levels. Therefore, the training model has extremely reliably identified objects. After 7200 iterations, the training model converges and stays consistent for the rest of the training. The validation loss for Yolo V3 1 is 0.0141, Yolo V3 2 is 0.015, Yolo V3 3 is 0.0129, Yolo V3 SPP 1 is 0.0125, for Yolo V3 SPP 2, 0.0144, and Yolo V3 SPP 3, 0.0133. The mean average accuracy ( $mAP$ ) is averaged over the  $p(o)$  accuracy by Equation (10) [52,53].

$$mAP = \int_0^1 p(o)do \quad (10)$$

Furthermore,  $p(o)$  is the precision of the Taiwan prohibitory sign detection. Precision and Recall are illustrated by Equations (12) and (13), [40,54]:

$$Precision (P) = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \quad (11)$$

$$Recall (R) = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)} \quad (12)$$

Moreover, TP represents True positives. FP is a positive sample of the misclassified. FN stands for a negative sample of the misclassified. The value of  $IoU$  is the relationship between the result of the detection, the reality of the ground truth, and its relation [55].  $IoU$  measures the projection ratio and shown in Equation (13) [1,56,57].

$$IoU = \frac{Area_{pred} \cap Area_{gt}}{Area_{pred} \cup Area_{gt}} \quad (13)$$

In Table 2, Yolo V3 SPP (98.88%, 99.12%, 98.93%) obtains a maximum  $mAP$  better than that of Yolo V3 (98.73%, 98.84%, 98.49%). Furthermore, Yolo V3 loaded 107 layers during the  $mAP$  calculation with BFLOPS rates of 65,312, and Yolo V3 SPP loaded 114 layers with BFLOPS rates of 65,69. SPP can enhance the overall BFLOFS 0.378, making Yolo V3 SPP more stable and precise.

## 4. Discussion

### 4.1. Testing Accuracy

Table 3 demonstrates the test accuracy for the prohibition signs in Taiwan. In comparison, Class P2 displays the highest mean precision accuracy, around 96.29%, supported by Class P1 at 92.45%, Class P4 at 91.69%, and Class P3 at 90.70%. Yolo V3 SPP 3 obtained the highest accuracy, around 95.53%, of any of the models tested, followed by Yolo V3 SPP 1 at 93.59%. Furthermore, Class P2 has the highest number of training images among other classes, amounting to 250, so the accuracy result for this class is the highest.

### 4.2. Testing Results

In this section, the experiments use random twenty prohibitive sign images of varying sizes and environments for model checking. The accuracy and time measurements of the experiments are presented in Table 4.

Generally, Yolo V3 SPP demonstrates higher precision than Yolo V3. The most leading average accuracy is Yolo V3 SPP 1 at 99.1%, followed by Yolo V3 SPP 3 at 93.33%. The trend is that the accuracy of Yolo V3 SPP grows along with the detection time. This indicates that Yolo V3 SPP needs more time to detect the sign. For example, for Yolo V3 SPP 1, the average time of detection is 0.458 s, and Yolo V3 1 needs 0.448 s. Further, a different scale affects the learning rate and detection time.

**Table 3.** Testing accuracy for all classes.

Model	Scale	Accuracy				
		P1	P2	P3	P4	Average
Yolo V3 1	<b>0.1, 0.1</b>	0.934	0.914	0.900	0.935	0.921
Yolo V3 2	<b>0.2, 0.2</b>	0.888	0.968	0.897	0.882	0.909
Yolo V3 3	<b>0.3, 0.3</b>	0.924	0.987	0.911	0.913	0.934
Yolo V3 SPP 1	<b>0.1, 0.1</b>	0.949	0.943	0.909	0.941	0.935
Yolo V3 SPP 2	<b>0.2, 0.2</b>	0.878	0.976	0.904	0.886	0.911
Yolo V3 SPP 3	<b>0.3, 0.3</b>	0.971	0.987	0.918	0.942	0.955
Average		0.924	0.962	0.907	0.916	

**Table 4.** The testing performance of the experiments.

No	Yolo V3 1		Yolo V3 2		Yolo V3 3		Yolo V3 SPP 1		Yolo V3 SPP 2		Yolo V3 SPP 3	
	Acc (%)	Time (s)	Acc (%)	Time (s)	Acc (%)	Time (s)	Acc (%)	Time (s)	Acc (%)	Time (s)	Acc (%)	Time (s)
1	0.971	0.446	0.992	0.443	0.961	0.452	0.993	0.456	0.979	0.458	0.998	0.452
2	0.825	0.456	0.965	0.441	0.695	0.458	0.992	0.453	0.960	0.452	0.948	0.458
3	0.997	0.450	0.999	0.433	0.994	0.442	1.000	0.457	0.992	0.443	1.000	0.498
4	0.996	0.464	0.990	0.449	0.980	0.446	0.999	0.457	0.934	0.480	1.000	0.454
5	0.932	0.438	0.990	0.446	0.955	0.441	0.980	0.462	0.970	0.455	0.984	0.462
6	0.967	0.456	0.926	0.449	0.900	0.439	0.973	0.447	0.889	0.448	0.903	0.454
7	0.903	0.455	0.881	0.448	0.910	0.443	0.973	0.457	0.943	0.450	0.972	0.462
8	0.845	0.449	0.880	0.461	0.599	0.456	0.994	0.451	0.932	0.454	0.991	0.462
9	0.949	0.446	0.986	0.463	0.822	0.440	0.992	0.465	0.978	0.458	0.999	0.461
10	0.869	0.442	0.967	0.441	0.867	0.455	0.991	0.452	0.960	0.445	0.999	0.463
11	0.990	0.452	0.960	0.445	0.960	0.457	0.989	0.456	0.909	0.459	0.992	0.444
12	0.965	0.451	0.638	0.470	0.885	0.461	0.998	0.470	0.987	0.456	0.987	0.447
13	1.000	0.449	1.000	0.448	1.000	0.455	1.000	0.463	0.999	0.450	0.999	0.452
14	0.884	0.433	0.871	0.446	0.573	0.449	0.988	0.472	0.933	0.461	0.933	0.447
15	0.907	0.444	0.980	0.438	0.787	0.455	0.983	0.468	0.908	0.477	0.908	0.445
16	0.907	0.441	0.901	0.438	0.981	0.436	0.994	0.439	0.748	0.444	0.748	0.451
17	0.991	0.439	0.992	0.433	0.534	0.438	0.997	0.446	0.989	0.441	0.989	0.442
18	0.953	0.456	0.867	0.450	0.905	0.451	0.998	0.457	0.952	0.476	0.952	0.461
19	0.891	0.446	0.554	0.437	0.814	0.440	0.994	0.463	0.594	0.459	0.594	0.453
20	0.898	0.456	0.806	0.451	0.857	0.414	0.987	0.464	0.751	0.452	0.751	0.436
Average	0.932	0.448	0.907	0.447	0.849	0.446	<b>0.991</b>	<b>0.458</b>	0.915	0.456	0.933	0.455

The average detection time for Yolo V3 SPP 1 using scale = 0.1, 0.1 is 0.0458, falling 0.002 in Yolo V3 SPP 2 (scale = 0.2, 0.2) to 0.0456, while for Yolo V3 SPP 3 (scale = 0.3,0.3) detection time is 0.0455 s. These results indicate that if the system uses a large number for scale, the detection time will be faster. Hence, the accuracy decreases compare to the original scale in Yolo V3. Similar to this, the accuracy decreases in Yolo V3 SPP adopting a different scale. The experiment results thus show that Yolo V3 SPP is more robust than Yolo V3. In this work, we use three different scales and provided a deep analysis for Yolo V3 and Yolo V3 SPP. Based on this experiment result, we can summarize as follows. (1) If the system wants the highest accuracy, we can use the original scale = 0.1, 0.1. (2) The system will use scale = 0.3, 0.3 if we want to increase the detection time more quickly.

The previous research [28,31,52] only focus on using the basic configuration of Yolo V3. They use the best weight provided from darknet with the scale = 0.1, 0.1. They optimize for accuracy but not for detection time. SPP contains more layers than the original method, which is why SPP takes more time for processing time. In our research, we provide a way to reduce detection time by increasing the scale parameter in the Yolo V3 and Yolo SPP configuration files. Our research proves that by using a scale = 0.3, 0.3 the detection time is faster than using a scale = 0.1, 0.1.

Sub-sampling and max-pooling have significant benefits. Convolution subsampling can be stronger reversed in subsequent sampling layers. Max pooling works slightly for deleting certain maximum frequency noise from the target image by choosing only maximum values from adjacent areas. By merging them, SPP seems to utilize both benefits to improve Yolo V3’s backbone network.

Furthermore, Figure 5a–c gives the test results for the Yolo V3 model with an average accuracy of around 95.92% and a detection time of 0.4415 s. Moreover, Figure 5d–f shows the test results for Yolo V3 SPP using the similar image. The average accuracy is 97.27%, and the detection time is 0.4548 s. The system can identify the prohibitory sign class P3 well. In Figure 6a–c, Yolo V3 failed to detect all class P1 signs in the image, detecting only a single sign. However, Yolo V3 SPP 1 can detect three signs well in Figure 6d, and Yolo V3 SPP 2 can detect two signs in Figure 6e,f.



**Figure 5.** Recognition test for prohibitory sign class P3 by (a) Yolo V3 1, (b) Yolo V3 2, (c) Yolo V3 3, (d) Yolo V3 SPP 1, (e) Yolo V3 SPP 2, and (f) Yolo V3 SPP 3.





**Figure 6.** Recognition test for prohibitory sign class P1 by (a) Yolo V3 1, (b) Yolo V3 2, (c) Yolo V3 3, (d) Yolo V3 SPP 1, (e) Yolo V3 SPP 2, and (f) Yolo V3 SPP 3.

## 5. Conclusions

This article refers to the SPP and transforms the network structure of Yolo V3. The research uses SPP to select local regions on a different scale of the same convolutional layer to learn multiscale

system characteristics. The experimental findings indicate that SPP will increase the performance of Taiwan's prohibitory signals identification and recognition. Furthermore, the accuracy decreases compare to the original scale in Yolo V3. However, the accuracy increases in Yolo V3 SPP adopting different scales. Moreover, comparison *mAP* of all models revealing Yolo V3 SPP outperforms Yolo V3. Nevertheless, *mAP* findings reveal that the Yolo V3 SPP model performs better over various scales than Yolo V3. Further, the scale will affect the learning rate and detection time. If we use a significant number for scale, the detection time will decrease, however, the accuracy will fall. We can conclude from the experiment result: (1) the system can be applied the original scale = 0.1, 0.1 if we want the best precision. (2) If we're going to increase the detection time more quickly, scale = 0.3. 0.3 can be used.

In future studies, we will enlarge the dataset focus from Taiwan prohibitory signs to all Taiwan traffic signs with the different condition including occlusion, multiple view, illumination, color variation, multiple weather conditions including heavy rain and snow. Further, future studies can extend the data set over the generative adversarial network (GAN) to create a synthetic image and obtain better results. Furthermore, we will test different scales and learning rates in the Yolo V3 SPP configuration file and the newest Yolo V4.

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

**Funding:** This research was funded by the Ministry of Science and Technology, Taiwan. The Nos are MOST-107-2221-E-324 -018 -MY2 and MOST-106-2218-E-324 -002, Taiwan. This research is also partially sponsored by Chaoyang University of Technology (CYUT) and the Higher Education Sprout Project, Ministry of Education (MOE), Taiwan, under the project name: "The R&D and the cultivation of talent for health-enhancement products."

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

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

## References

1. 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](#)]
2. Nagpal, R.; Paturu, C.K.; Ragavan, V.R.; Navinprashath, R.; Bhat, R.; Ghosh, D. Real-time traffic sign recognition using deep network for embedded platforms. *Electron Imaging* **2019**, *2019*, 33-1–33-8. [[CrossRef](#)]
3. Kaplan, A.M.; Haenlein, M. Users of the world, unite! The challenges and opportunities of Social Media. *Bus. Horiz.* **2010**, *53*, 59–68. [[CrossRef](#)]
4. Qian, X.; Feng, H.; Zhao, G.; Mei, T. Personalized Recommendation Combining User Interest and Social Circle. *IEEE Trans. Knowl. Data Eng.* **2013**, *26*, 1763–1777. [[CrossRef](#)]
5. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
6. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014; Volume 1, pp. 1–16.
7. Tao, X.; Gong, Y.; Shi, W.; Cheng, D. Object detection with class aware region proposal network and focused attention objective. *Pattern Recognit. Lett.* **2020**, *130*, 353–361. [[CrossRef](#)]
8. Gong, Y.; Wang, L.; Guo, R.; Lazebnik, S. Multi-scale Orderless Pooling of Deep Convolutional Activation Features. In *Proceedings of the Lecture Notes in Computer Science*; Springer Science and Business Media LLC: Berlin, Germany, 2014; Volume 8695, pp. 392–407.
9. Møgelmoose, A.; Trivedi, M.M.; Moeslund, T.B. Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 1484–1497. [[CrossRef](#)]



10. Shahud, M.; Bajracharya, J.; Praneetpolgrang, P.; Petcharee, S. Thai Traffic Sign Detection and Recognition Using Convolutional Neural Networks. In Proceedings of the 2018 22nd International Computer Science and Engineering Conference (ICSEC), Chiang Mai, Thailand, 21–24 November 2018; pp. 1–5.
11. Hu, Y.; Wang, F.; Liu, X. A CPSS Approach for Emergency Evacuation in Building Fires. *IEEE Intell. Syst.* **2014**, *29*, 48–52. [[CrossRef](#)]
12. Dewi, C.; Chen, R.-C. Random Forest and Support Vector Machine on Features Selection for Regression Analysis. *Int. J. Innov. Comput. Inf. Control.* **2019**, *15*, 2027–2038.
13. Jin, Y.; Fu, Y.; Wang, W.-Q.; Guo, J.; Ren, C.; Xiang, X. Multi-Feature Fusion and Enhancement Single Shot Detector for Traffic Sign Recognition. *IEEE Access* **2020**, *8*, 38931–38940. [[CrossRef](#)]
14. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In *Proceedings of the Lecture Notes in Computer Science Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*; Springer: Berlin, Germany, 2016; pp. 21–37.
15. 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](#)]
16. Zhang, J.; Xie, Z.; Sun, J.; Zou, X.; Wang, J. A Cascaded R-CNN With Multiscale Attention and Imbalanced Samples for Traffic Sign Detection. *IEEE Access* **2020**, *8*, 29742–29754. [[CrossRef](#)]
17. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object detection via region-based fully convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 379–387.
18. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
19. Dewi, C.; Chen, R.-C. Integrating Real-Time Weather Forecasts Data Using OpenWeatherMap and Twitter. *Int. J. Inf. Technol. Bus.* **2019**, *1*, 48–52.
20. Dewi, C.; Chen, R.C.; Hendry; Hung, H. Te Comparative Analysis of Restricted Boltzmann Machine Models for Image Classification. In *Proceedings of the Lecture Notes in Computer Science Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*; Springer: Berlin, Germany, 2020; Volume 12034 LNAI, pp. 285–296.
21. Jensen, M.B.; Nasrollahi, K.; Moeslund, T.B. Evaluating State-of-the-Art Object Detector on Challenging Traffic Light Data. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 882–888.
22. Zhang, J.; Huang, M.; Jin, X.; Li, X. A Real-Time Chinese Traffic Sign Detection Algorithm Based on Modified YOLOv2. *Algorithms* **2017**, *10*, 127. [[CrossRef](#)]
23. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
24. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934, 1–17.
25. Grauman, K.; Darrell, T. The pyramid match kernel: Discriminative classification with sets of image features. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, Beijing, China, 17–21 October 2005; Volume 2, p. 1458.
26. Lazebnik, S.; Schmid, C.; Ponce, J. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 1 (CVPR'06), New York, NY, USA, 17–22 June 2006; pp. 1–8.
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, 1–14. [[CrossRef](#)]
28. Dewi, C.; Chen, R.-C.; Tai, S.-K. Evaluation of Robust Spatial Pyramid Pooling Based on Convolutional Neural Network for Traffic Sign Recognition System. *Electronics* **2020**, *9*, 889. [[CrossRef](#)]
29. Yanliang, C.; Yongran, L.; Baisen, W.; Hanyin, Z.; Zhihong, W. Applying deep learning technology to image recognition of traffic signs. In *Proceedings of the 2019 International Conference on Technologies and Applications of Artificial Intelligence*; Springer: Berlin, Germany, 2019; pp. 1–4.
30. Yen-Zhang, H. Building a traffic signs open dataset in Taiwan and verify it by convolutional neural network. Ph.D. Thesis, National Taichung University of Science and Technology, Taichung, Taiwan, 2018.
31. Dewi, C.; Chen, R.-C.; Liu, Y.-T.; Liu, Y.-S.; Jiang, L.-Q. Taiwan Stop Sign Recognition with Customize Anchor. In Proceedings of the 12th International Conference on Computer Modeling and Simulation; Association for Computing Machinery (ACM), Brisbane, Australia, 26–28 February 2020.



32. Redmon, J.; Divvala, S.; Girshick, R.; Ali, F. (YOLO) You Only Look Once. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1–10.
33. Springenberg, J.T.; Dosovitskiy, A.; Brox, T.; Riedmiller, M. Striving for simplicity: The all convolutional net. *arXiv* **2014**, arXiv:1412.6806.
34. 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.
35. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
36. 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 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8772–8777.
37. Kamal, U.; Tonmoy, T.I.; Das, S.; Hasan, K. Automatic Traffic Sign Detection and Recognition Using SegU-Net and a Modified Tversky Loss Function With L1-Constraint. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1467–1479. [[CrossRef](#)]
38. Mao, Q.-C.; Liu, X.-Y.; Liu, Y.-B.; Jia, R.-S. Mini-YOLOv3: Real-Time Object Detector for Embedded Applications. *IEEE Access* **2019**, *7*, 133529–133538. [[CrossRef](#)]
39. Wu, F.; Jin, G.; Gao, M.; He, Z.; Yang, Y. Helmet Detection Based on Improved YOLO V3 Deep Model. In Proceedings of the 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC), Banff, Canada, 9–11 May 2019; pp. 363–368.
40. Xu, Q.; Lin, R.; Yue, H.; Huang, H.; Yang, Y.; Yao, Z. Research on Small Target Detection in Driving Scenarios Based on Improved Yolo Network. *IEEE Access* **2020**, *8*, 27574–27583. [[CrossRef](#)]
41. Zhang, Z.; Zhang, X.; Lin, X.; Dong, L.; Zhang, S.; Zhang, X.; Sun, D.; Yuan, K. Ultrasonic Diagnosis of Breast Nodules Using Modified Faster R-CNN. *Ultrason. Imaging* **2019**, *41*, 353–367. [[CrossRef](#)] [[PubMed](#)]
42. Dewi, C.; Chen, R.-C.; Hendry; Liu, Y.-T. Similar Music Instrument Detection via Deep Convolution YOLO-Generative Adversarial Network. In Proceedings of the 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST), Morioka, Japan, 23–25 October 2019; pp. 1–6.
43. Sivic, J.; Zisserman, A. Video Google: A text retrieval approach to object matching in videos. In Proceedings of the Proceedings Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; Volume 2, pp. 1470–1477.
44. Yang, J.; Yu, K.; Gong, Y.; Huang, T. Linear spatial pyramid matching using sparse coding for image classification. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. *CVPR Workshops* **2009**, 1794–1801.
45. Wang, J.; Yang, J.; Yu, K.; Lv, F.; Huang, T.; Gong, Y. Locality-constrained Linear Coding for image classification. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3360–3367.
46. Perronnin, F.; Sánchez, J.; Mensink, T. Improving the Fisher Kernel for Large-Scale Image Classification. In *Proceedings of the Lecture Notes in Computer Science*; Springer Science and Business Media LLC: Berlin, Germany, 2010; Volume 6314, pp. 143–156.
47. Van De Sande, K.E.A.; Uijlings, J.R.R.; Gevers, T.; Smeulders, A.W.M. Segmentation as selective search for object recognition. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 1879–1886.
48. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
49. Bbox label tool. Available online: <https://github.com/puzzledqs/BBox-Label-Tool> (accessed on 13 March 2020).
50. Zoph, B.; Cubuk, E.D.; Ghiasi, G.; Lin, T.-Y.; Shlens, J.; Le, Q.V. Learning Data Augmentation Strategies for Object Detection. *arXiv* **2019**, 1–13.
51. Murugavel, M. YOLO Annotation Tool. Available online: <https://github.com/ManivannanMurugavel/YOLO-Annotation-Tool> (accessed on 20 March 2020).

52. Dewi, C.; Chen, R.-C.; Yu, H. Weight analysis for various prohibitory sign detection and recognition using deep learning. *Multimed. Tools Appl.* **2020**, *1–19*. [[CrossRef](#)]
53. Chen, R.-C.; Dewi, C.; Huang, S.-W.; Caraka, R.E. Selecting critical features for data classification based on machine learning methods. *J. Big Data* **2020**, *7*, 1–26. [[CrossRef](#)]
54. Dewi, C.; Chen, R.-C. Human Activity Recognition Based on Evolution of Features Selection and Random Forest. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019; pp. 2496–2501.
55. Huang, Y.-Q.; Zheng, J.-C.; Sun, S.-D.; Yang, C.-F.; Liu, J. Optimized YOLOv3 Algorithm and Its Application in Traffic Flow Detections. *Appl. Sci.* **2020**, *10*, 3079. [[CrossRef](#)]
56. Dewi, C.; Chen, R.-C. Decision Making Based on IoT Data Collection for Precision Agriculture. In *Intelligent Tools for Building a Scientific Information Platform*; Springer Science and Business Media LLC: Berlin, Germany, 2019; pp. 31–42.
57. Dewi, C.; Chen, R. Intelligent information and database systems: Recent developments. In *Intelligent Information and Database Systems: Recent Developments*; Springer: Berlin, Germany, 2019; ISBN 978-3-030-14131-8.



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).