Hindawi
Scientific Programming
Volume 2020, Article ID 8403262, 10 pages
https://doi.org/10.1155/2020/8403262

Research Article

Object Detection through Modified YOLO Neural Network

Tanvir Ahmad,[1] **Yinglong Ma**,[1] **Muhammad Yahya**,[2] **Belal Ahmad**,[3] **Shah Nazir**,[4] **and Amin ul Haq**[5]

[1]*School of Control and Computer Engineering, North China Electric Power University, Beijing, China*
[2]*Department of Electrical and Electronics, Universiti Kuala Lumpur, Kuala Lumpur, Malaysia*
[3]*Embedded and Pervasive Computing (EPIC) Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China*
[4]*Department of Computer Science, University of Swabi, Ambar, Khyber Pakhtunkhwa, Pakistan*
[5]*School of Computer Science and Engineering, University of Electronic Science and Technology, Chengdu, China*

Correspondence should be addressed to Tanvir Ahmad; tanvirahmad@ncepu.edu.cn

Received 7 December 2019; Revised 3 February 2020; Accepted 22 February 2020; Published 6 June 2020

Guest Editor: Rahman Ali

In the field of object detection, recently, tremendous success is achieved, but still it is a very challenging task to detect and identify objects accurately with fast speed. Human beings can detect and recognize multiple objects in images or videos with ease regardless of the object's appearance, but for computers it is challenging to identify and distinguish between things. In this paper, a modified YOLOv1 based neural network is proposed for object detection. The new neural network model has been improved in the following ways. Firstly, modification is made to the loss function of the YOLOv1 network. The improved model replaces the margin style with proportion style. Compared to the old loss function, the new is more flexible and more reasonable in optimizing the network error. Secondly, a spatial pyramid pooling layer is added; thirdly, an inception model with a convolution kernel of $1 * 1$ is added, which reduced the number of weight parameters of the layers. Extensive experiments on Pascal VOC datasets 2007/2012 showed that the proposed method achieved better performance.

1. Introduction

Human beings can easily detect and identify objects in their surroundings, without consideration of their circumstances, no matter what position they are in and whether they are upside down, different in color or texture, partly occluded, etc. Therefore, humans make object detection look trivial. The same object detection and recognition with a computer require a lot of processing to extract some information on the shapes and objects in a picture.

In computer vision, object detection refers to finding and identifying an object in an image or video. The main steps involved in object detection include feature extraction [1], feature processing [2–4], and object classification [5]. Object detection achieved excellent performance with many traditional methods that can be described from the following four aspects: bottom feature extraction, feature coding, feature aggregation, and classification. The feature extraction plays an essential role in the object detection and recognition process [6]. There will be more redundant information which can be modeled to achieve better performance than previous point-of-interest detection. Previously used scale-invariant feature transformations (SIFT) [7] and histogram of oriented gradients (HOG) [8] belong to this category.

The object detection is critical in different applications, such as surveillance, cancer detection, vehicle detection, and underwater object detection. Various techniques have been used to detect the object accurately and efficiently for different applications. However, these proposed methods still have problems with a lack of accuracy and efficiency. To tackle these problems of the object detection, machine learning and deep neural network methods are more effective in correcting object detection.

Thus, in this study, a modified new network is proposed based on the YOLOv1 [9] network model. The performance

of the modified YOLOv1 is improved through the following points:

(i) The loss function of the YOLOv1 network is optimized.

(ii) The inception model structure is added.

(iii) A spatial pyramid pooling layer is used.

(iv) The proposed model effectively extracts features from images, performing much better in object detection.

The remaining of this paper is organized as follows. Section 2 describes related work. Section 3 presents the methodology, which describes network architecture in detail. Section 4 presents the analysis of the improved network from various aspects. In Section 5, the experiment setup, results, and comparison with other networks are discussed. The paper conclusion and future work are given in Section 6.

2. Related Work

Detecting and identifying multiple objects in an image is hard for machines to recognize and classify. However, a noteworthy effort has been carried out in the past years in the detection of objects using convolutional neural networks (CNNs). In the object detection and recognition field, neural networks are in use for a decade but became prominent due to the improvement of hardware new techniques for training these networks on large datasets [10, 11]. In object detection and recognition, researchers have used deep learning for learning features directly from the image pixels, which are more effective than the manual features [4, 12]. Recently deep learning-based algorithms remove the manual features extraction methods and directly use features extracting methods [13] from the original images. This methodology has been successfully proven in feature pyramid network (FPN) [14], single shot detector (SSD) [15], and deconvolutional single shot detector (DSSD) [16]. Deep learning is a prevailing direction in the field of machine learning [17]. In [18, 19], researchers showed that the CNNs inherit the advantages of deep learning, which makes their results in the field of object detection and recognition greatly improved compared with the traditional methods. Researchers had made many efforts to use stochastic gradient descent and backpropagation to train deep networks for object detection [20]. Those networks were able to learn but were too slow in practice to be useful in real-time applications; the technique in [12] showed that stochastic gradient descent by backpropagation was effective in training CNNs. CNNs became in use but fell out of fashion due to the support vector machine as in [21] and other simpler methods like linear classifiers as in [22]. New techniques that have been developed recently [23, 24] show higher image classification accuracy in ImageNet large scale visual recognition [25]. These techniques have brought much more easiness to train large and deeper networks and shown enhanced performance. Newly, approaches have been established to identify vehicles and other objects from videos or static images using deep convolutional neural networks (DCNN) [26–30]. For example, faster R-CNN [19] proposes candidate regions and uses CNN to confirm candidates as valid objects. YOLO uses end-to-end unified, fully convolutional network structure that predicts the objectless assurance and the bounding boxes concurrently over the whole image. SSD [31] outperforms YOLO by discretizing the production space of bounding boxes into a set of avoidance boxes over different feature ratios and scales per feature map location. YOLO-2 [32] achieves state-of-the-art performance in object detection by improving various aspects of its earlier version. A fully convolutional network is utilized for object detection from three-dimensional (3D) range scan data with LIDAR. A 2D-DBN design is proposed, which uses second-order planes instead of first-order vectors as inputs and uses bilinear projection for retaining discriminative information to develop the recognition rate [33]. Although DCNN based approaches accomplish the state-of-the-art accuracy of detection or classification, these approaches often require intensive calculation and a considerable amount of labeled training data. Through the past few years, to use deep neural networks economically in real-time applications, a substantial amount of work has been done to report these two problems [34, 35]. In this study, a different modified architecture for object detection is addressed, which is capable of providing high accuracy and speed.

3. Methodology

In this section, the proposed model is described in detail. Firstly, the improvement based on loss function is presented. Secondly, the improvement based on inception structure model is described. And lastly, the improvement based on the spatial pyramid pooling layer is portrayed. The symbolic representations are described in Table 1.

3.1. Improvement in Network Design. The following improvements to the YOLO network model are made while maintaining the original model dominant idea.

3.1.1. Improvement Based on Loss Function. The loss function of the original YOLOv1 network takes the same error for the large and small objects, which makes the model's prediction for neighboring objects unsatisfactory. If two objects appear in the same grid, only one object can be detected, and there will be a problem in detecting small objects. Compared with the old loss function, the new loss function is more flexible and optimized. In the new loss function, the original difference is replaced by the proportionality. Equation (1) shows the original loss function of YOLOv1; YOLOv1 uses one single loss function for both bounding boxes and the classification of the object. Loss function can be described in five parts: the first and second are focusing on the loss of the bounding box coordinates, while the third and fourth are responsible for the difference in the confidence of having an object in the grid, and part five is responsible for the difference in class probability. The λ_{coord} and λ_{noobj} are scalars to weight each loss function,

Table 1: The mathematical symbols.

Symbol/ notation	Description
λ_{coord}	Hyperparameter is set to ensure a "fair" contribution of the bounding box location
λ_{noobj}	Hyperparameter is set for bounding box score prediction
C	Categories
$P(C)$	Probability of detected class categories
N	Training samples
X	Training input
Y	Output label
t	Input label

λ_{coord} is set to 5, and λ_{noobj} is set to 0.5 by the original author of YOLOv1.

$$\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{obj}} \left(x_i - x_i'\right)^2 + \left(y_i - y_i'\right)^2 \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{obj}} \left(\sqrt{w_i} - \sqrt{w_i'}\right)^2 + \left(\sqrt{h_i} - \sqrt{h_i'}\right)^2 \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij} \left(c_i - c_i'\right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{noobj}} \left(c_i - c_i'\right)^2 \\
&+ \sum_{i=0}^{S^2} I_{ij} \sum_{c \in \text{class}} \left(p_i(c) - p_i'(c)\right)^2.
\end{aligned} \tag{1}$$

In convolutional neural networks, variance function is often used as the loss function [36] of the network. For example, for a variety of problems, the total number of categories is C and training samples is N. The algorithm which is used for multiclassification first needs to find those weights and biases that make the output of the neural network close to $y(x)$ (which is labeled category) for all training inputs x; to quantify how close the output of all training inputs x is to $y(x)$, the loss function is defined as

$$E^N = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{C} \left(t_k^n - y_k^n\right)^2. \tag{2}$$

Here, t represents the label of the input object, and y represents the actual output value of the input object to the network. The function of choosing the variance form is the loss function to facilitate subsequent optimization. On the other hand, the current training level can be predicted by observing the severity of the fluctuation of the loss value in practice.

In the YOLOv1 network loss function design, the variance function is used as part of the entire loss function, the normalization idea of contrast is used to improve it, and the improved model replaces margin style with proportion style, so here the size of the object in the picture is considered. The specific modified loss function is shown in

$$\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{obj}} \left(x_i - x_i'\right)^2 + \left(y_i - y_i'\right)^2 \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{obj}} \left(\frac{w_i - w_i'}{w_i'}\right)^2 + \left(\frac{h_i - h_i'}{h_i'}\right)^2 + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \left(c_i - c_i'\right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{noobj}} \left(c_i - c_i'\right)^2 + \sum_{i=0}^{S^2} I_{ij}^{\text{obj}} \sum_{c \in \text{class}} \left(p_i(c) - p_i'(c)\right)^2.
\end{aligned} \tag{3}$$

Here, I_{ij}^{obj} indicates that the target object is assumed to be present in the i^{th} position of the area. x and y represent the current position of the image; w and h represent the width and height of the image. c is the total number of objects to be identified, and $p\ (c)$ is the probability that the object belongs to a specific class c. Here, it should be noted that the loss function guides the optimization of the class to which the object belongs and optimizes the position of the boundary box for detecting the object.

3.1.2. Improvement of Inception Structure Model. The third and fourth layers of the original network are replaced with new inception models. The inception model itself has the ability to deepen and widen the network and enhance the network; a $64 \times 1 \times 1$ layer is added between the first and second layers of the original network, which reduces the network parameters. Figure 1 shows the structure part of the YOLOv1 network after adding the inception model. Inception architecture is used to find out how an optimal local sparse structure in a convolutional neural network can be approximated and covered by readily available dense components.

The inception model can deepen and widen the network, and the convolutional kernel of different scales is connected in parallel. Thus, the multiscale feature can be more effective, and the hidden information in the image can be used more efficiently.

3.1.3. Improvement of SPP Structure Model. Figure 2 shows the addition of spatial pyramid pooling (SPP) layer, and below are the advantages of using it.

(i) It can output a fixed-size image for any size input or any ratio of the input image.

(ii) It can extract pool features at varying scales.

A classifier (SVM/Softmax), as well as fully connected layers, requires a fixed-length vector, which can be generated through Bag-of-Words (BoW) [35, 37, 38], the spatial pyramid downsampling boosts the BoW because it preserves spatial information by pooling the spatial bins. These spatial bins have sizes proportional to the image size, so the number of bins is fixed regardless of the image size, which makes the SPP [39, 40] not only improve network performance but also dramatically reduce the required calculation time by avoiding repeatedly computing the convolutional features.

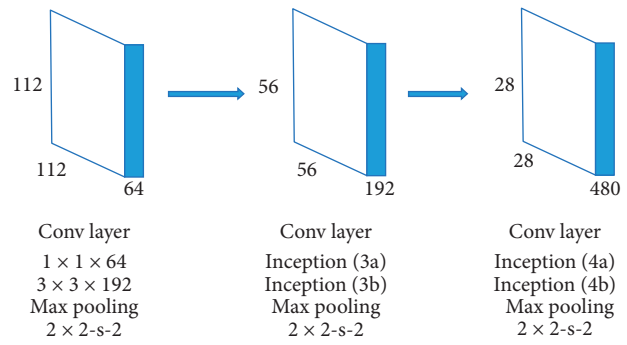


Figure 1: Partial structure of the new network after adding inception.

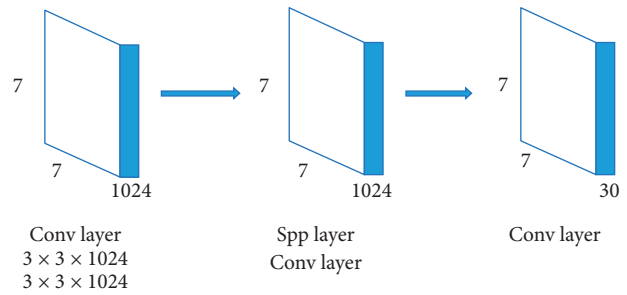


Figure 2: Partial structure of the new network after joining the SPP layer.

By using the SPP layer, more feature-rich image information is obtained, and also great improvements in the network's time efficiency are observed. Hence, this technique shows remarkable detection accuracy.

4. Analysis of the Network

Following is the comprehensive analysis of our proposed network and improved YOLO model based on the results of the experimental tests.

(i) By the analysis of the confusion matrix, we observed what kind of sample detection performance is better for the new network, what kind of sample detection performance is not good, and how to distinguish the easily confused categories and understand the advantages and disadvantages of the network.

(ii) We examined the network architecture of the new network model, such as the comparison of the number of network parameters, and assessed its performance.

4.1. Confusion Matrix. Through the confusion matrix, the test results are analyzed. A confusion matrix is a list of data classes; in each class, the actual data is classified so that we can observe which categories of samples are easily confused in the modified network. In the confusion matrix, the rows represent the true categories of the test images. The columns show the classes of the test images divided by the network in the actual test.

In the original Pascal VOC dataset, there are 20 categories of objects; here some representative categories, which easily cause misidentification, are selected.

Table 2 is the confusion matrix of the modified network model on the Pascal VOC 2007 dataset. It can be noticed from Table 2 that the airplane is mistakenly recognized as a bird, and the original samples belonging to birds are identified as airplanes. The reason is that the overall shape is too similar: the airplane has two wings, and so does the bird; the airplane's body shape is very similar to that of a bird; therefore, the results show that 22% of the airplanes are mistakenly identified as birds, and 36% of the birds were incorrectly identified as airplanes. In addition, the chair and sofa are also relatively easy to cause misidentification, because in real life it is very easy to differentiate between chairs and sofas, but in picture chairs and sofas are very easy to appear the same, which can cause miss identification very easily. And the same applies for sheep, horses, dogs, and cats.

From Table 2, it can be seen that the overall average misrecognition rate is not too high, indicating that the overall ability of the network to extract features and detect target objects in the image is relatively reliable.

4.2. Network Architecture. Here, the proposed network architecture is described. Before going into detail, please note that the first and second layers are the same: both are convolutional layers plus the downsampling layer structure; the third and fourth layers are the same: both are inception + pool structures; the fifth and sixth layers are the same: both are convolutional cascade structures; the seventh layer is spatial pyramid pooling layer; and the eighth and ninth layers are the fully connected layers.

For the first layer, it is assumed that the input is an image, r is the number of rows of the image, c is the number of columns of the image to a network of the first layer input, and the sliding step is s_1; the computational cost of obtaining a feature map is shown in the equation:

$$(x_1 y_1)\frac{r - x_1 + s_1}{s_1}\frac{c - y_1 + s_1}{s_1}. \tag{4}$$

Computing area is the size of the convolution kernel area, so the result of (4) is obtained, and then we assume that the first layer has n_1 feature maps, so the calculation of the first layer is

$$\frac{r - x_1 + s_1}{s_1}\frac{c - y_1 + s_1}{s_1}(x_1 y_1 n_1), \tag{5}$$

and the size of the feature map after convolution will become

$$\begin{aligned} r_1 &= \left\lceil \frac{r - x_1 + s_1}{s_1} \right\rceil, \\ c_1 &= \left\lceil \frac{c - y_1 + s_1}{s_1} \right\rceil. \end{aligned} \tag{6}$$

Next is the maximum downsampling layer; since the downsampling layer does not change the number of feature maps, the numbern_2 of the feature maps is equal to the

Table 2: Confusion matrix for the new network.

(%)	Bird	Chair	Sofa	Aero	Horse	Sheep	Dog	Cat
Bird	64			36				
Chair		43	57					
Sofa		41	59					
Aero	22			78				
Horse					78	22		
Sheep					28	72		
Dog							82	18
Cat							14	86

number n_1 of the previous feature maps. Assuming the size of the downsampling window, the size of the feature map obtained after downsampling is

$$x_2 = \frac{r_1}{r_2}, \quad y_2 = \frac{c_1}{c_2}. \tag{7}$$

The calculation of the total number of n_2 feature maps will become

$$n_2 r_2 c_2 x_2 y_2. \tag{8}$$

The following is the convolution second layer, assuming that the number of features

$$r_3 = \left\lceil \frac{r_2 - x_3 + s_3}{s_3} \right\rceil, \quad c_3 = \left\lceil \frac{c_2 - y_3 + s_3}{s_3} \right\rceil. \tag{9}$$

The calculation with the upper layer of the feature map for convolution operation will be as follows.

$$(x_3 y_3 n_3) \left\lceil \frac{r_2 - x_3 + s_3}{s_3} \right\rceil \left\lceil \frac{c_2 - y_3 + s_3}{s_3} \right\rceil. \tag{10}$$

Assuming that the output of the maximum downsampling layer in the second layer is characterized by the size of the downsampling window and with the step size s_4, calculation of the total amount of the layer can be obtained by the same way.

$$x_4 = \frac{r_3}{r_4}, \quad y_4 = \frac{c_3}{c_4}. \tag{11}$$

From the above, it can be seen that the output feature size of MaxPool2 is n_4. In the inception structure, the step size is 1, and the calculation is from left to right. The third layer's inception structure model is shown in Figure 3 and mathematically shown in

$$\begin{aligned} &n_4 \times 1 \times 1 \times 64, \\ &n_4 \times 1 \times 1 \times 96 + 96 \times 3 \times 3 \times 128, \\ &n_4 \times 1 \times 1 \times 16 + 16 \times 5 \times 5 \times 32, \\ &n_4 \times r_4 \times c_4 + n_4 \times 1 \times 1 \times 32. \end{aligned} \tag{12}$$

Thus, the whole calculation of inception four layers can be done in the above way. Next is the fifth layer of the convolution, and the total calculation is

$$\begin{aligned} &1 \times 1 \times r_5 \times c_5 \times 512 + 2 \times 3 \times 3 \times (r_5 - 3 + 1) \\ &\times (c_5 - 3 + 1) \times 1024 \\ &+ 3 \times 3 \times \left(\frac{r_5 - 3 + 2}{2}\right)\left(\frac{c_5 - 3 + 2}{2}\right) \times 1024. \end{aligned} \tag{13}$$

Since the sixth layer and the fifth layer have the same structure, the calculation is the same as (13).

The seventh layer is the pyramid layer, denoted by L, where $n = 1, 2, \ldots, L$. The calculation amount of the pyramid layer is

$$\sum_{n=1}^{L} \left(n_5 r_{6,n} c_{6,n} \left\lceil \frac{r_5}{r_{6,n}} \right\rceil \left\lceil \frac{c_5}{c_{6,n}} \right\rceil \right). \tag{14}$$

The eighth layer is fully connected. Assume that the number of input features is n_6, and the number of output features isn_7. Because the input of the layer is the former layer, it will be processed after all the features of the map are gathered as a vector, so $n_6$is

$$n_6 = \sum_{n=1}^{L} n_5 r_{6,n} c_{6,n}. \tag{15}$$

Because the full-connection layer is derived from the original neural network, the calculation method is the same as that of the neural network, so the computational cost of the layer is

$$n_7 \left(2 \sum_{n=1}^{L} n_5 r_{6,n} c_{6,n} + 1 \right). \tag{16}$$

From the above description of network architecture analysis, it is observed that the network's overall calculation, input layer image size, convolution kernel size, and the number of convolutional layers, shows that network depth and width are having big impact.

5. Experiment

Pascal VOC is divided into two datasets: Pascal VOC 2007 and Pascal VOC 2012 dataset. The newly designed network was tested on both datasets [41]. The Pascal VOC dataset consist of 20 categories: person, bird, cat, cow, horse, sheep, airplane, bike, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, and TV monitor. Figures 4 and 5 show the sample images.

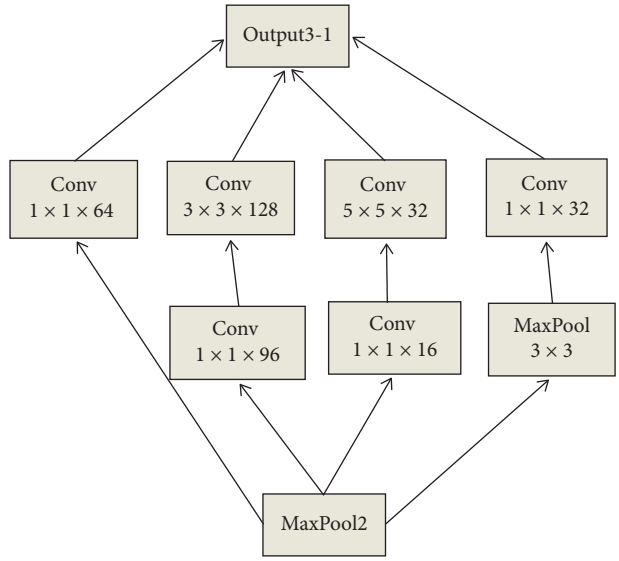


Figure 3: Inception model architecture.

(a) (b) (c)

(d) (e) (f)

Figure 4: Pascal VOC 2007 dataset images. (a) 000006. (b) 000008. (c) 000014. (d) 000015. (e) 000017. (f) 000018.

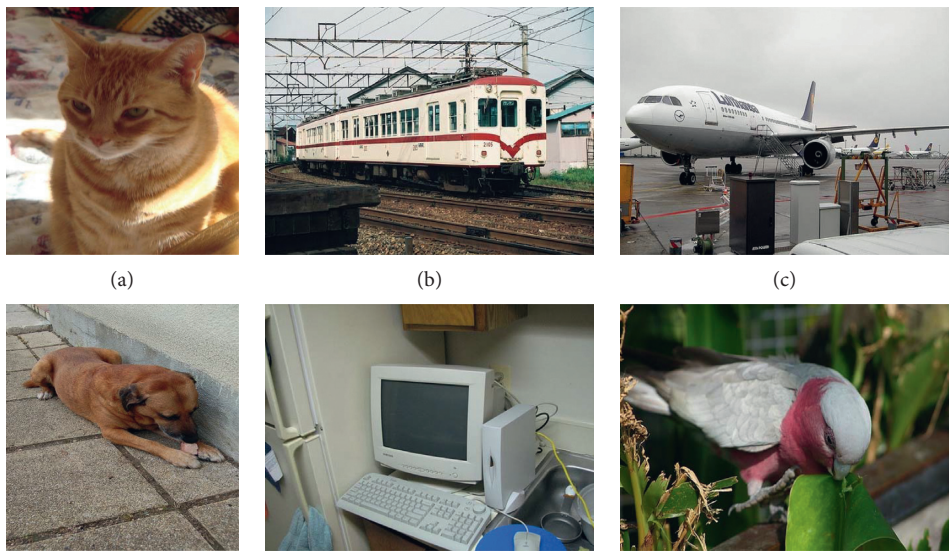
The whole experiment process is conducted on NVIDIA GeForce GTX 1060 GPU using the Ubuntu operating system. The number of iterations was 40000.

5.1. Results and Discussion. The results are discussed and the network performance is checked using t-SNE visualization tool, showing the extent to which the new network is able to extract rich features from images.

Next, the visualization of a large number of sample features in 2D is observed by using the t-SNE visualization tool, which maps high-dimensional to low-dimensional data [42].

Figure 6 shows ten selected categories from the Pascal VOC dataset (bird, chair, sofa, bike, airplane, horse, sheep, dog, cat, cow) using the t-SNE visualization tool; in the figure, different colors represent different types; if the two types are fused, this means that these types are easily getting confused with one another.

There are about seven categories which are not compatible with each other, indicating that the characteristics of these seven types of differences are relatively large and relatively easy to identify; in addition to several types of partial integration, the characteristics of several types have a certain degree of similarity, which is easy to cause misidentification. However, overall, the use of the new network to extract the characteristics is very effective and robust, but it is also inadequate and needs to be further improved. The improved network was tested on Pascal VOC 2007 and



(a) (b) (c)

(d) (e) (f)

Figure 5: Pascal VOC 2012 dataset images. (a) 000028. (b) 000031. (c) 000033. (d) 000036. (e) 000039. (f) 000040.

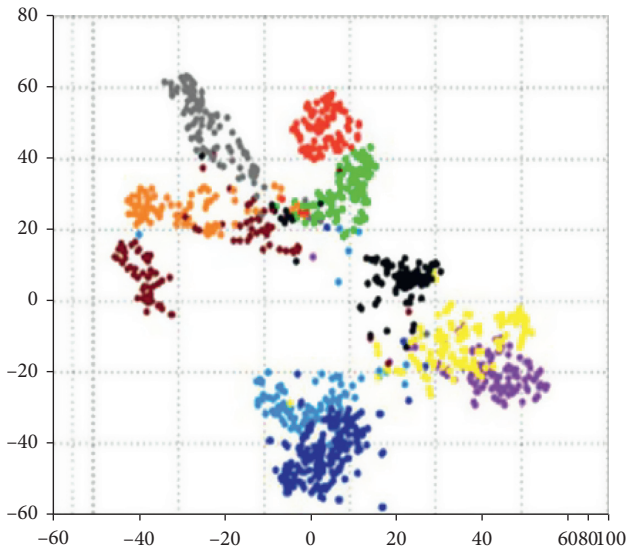


Figure 6: Two-dimensional visualization of ten samples.

Pascal VOC 2012, respectively. The results are shown in Tables 3 and 4.

The data in Tables 3 and 4 is expressed in percentage. In the above results, to make the comparison results more consistent, the training dataset used in the above algorithm is the train/val dataset of Pascal VOC 2007 and Pascal VOC 2012. The data presented in Tables 3 and 4 are test results for each class of 20 objects. Our modified network average detection rate is 65.6% and 58.7% on the Pascal VOC 2007 and 2012 dataset. To check the performance, we compared the results of our modified network with those of R-CNN and YOLOv1, as depicted in Tables 5 and 6 for Pascal VOC 2007 and 2012, respectively. Table 5 shows the Pascal VOC 2007 comparison test results, and in Table 6 Pascal VOC 2012 comparative test results are presented.

Table 3: Pascal VOC 2007 test results.

VOC 2007	The modified YOLOv1	VOC 2007	The modified YOLOv1
Aero	77.9	Table	51.2
Bike	77.6	Dog	81.9
Bird	63.7	Horse	77.5
Boat	47.6	M-bike	78.7
Bottle	44.8	Person	68.6
Bus	70.7	Plant	37.1
Car	68.9	Sheep	71.8
Cat	85.3	Sofa	58.4
Chair	42.2	Train	71.0
Cow	71.9	Tv	64.6
Average recognition rate			65.6

Table 4: Pascal VOC 2012 test results.

VOC 2012	The modified YOLOv1	VOC 2012	The modified YOLOv1
Aero	76.1	Table	49.1
Bike	67.8	Dog	80.3
Bird	58.0	Horse	72.7
Boat	39.9	M-bike	71.9
Bottle	24.2	Person	64.2
Bus	68.9	Plant	29.0
Car	57.6	Sheep	54.5
Cat	82.5	Sofa	55.2
Chair	36.3	Train	73.9
Cow	61.1	Tv	51.7
Average recognition rate			58.7

It can be seen from the tables that our modified model has improved recognition over the YOLOv1 and R-CNN model in almost every type. Table 7 depicts the processing time of an image of three different networks, R-CNN, YOLOv1, and our improved YOLO, for testing the same image. The time taken by the R-CNN network is 6.9 seconds,

Table 5: Pascal VOC 2007 comparison test results.

VOC 2007	R-CNN	YOLOv1	The modified YOLOv1
Aero	63.5	78	77.9
Bike	66	74.2	77.6
Bird	47.9	61.3	63.7
Boat	37.7	45.7	47.6
Bottle	29.9	42.7	44.8
Bus	62.5	68.2	70.7
Car	70.2	66.8	68.9
Cat	60.2	80.2	85.3
Chair	32	40.6	42.2
Cow	57.9	70	71.9
Table	47	49.8	51.2
Dog	53.5	79	81.9
Horse	60.1	74.5	77.5
M-bike	64.2	77.9	78.7
Person	52.2	64	68.6
Plant	31.3	35.3	37.1
Sheep	55	67.9	71.8
Sofa	50	55.7	58.4
Train	57.7	68.7	71
TV	63	62.6	64.6
Average recognition rate	53.1	63.4	65.6

Table 6: Pascal VOC 2012 comparative test results.

VOC 2012	R-CNN	YOLOv1	The modified YOLOv1
Aero	68.1	77	76.1
Bike	63.8	64.2	67.8
Bird	46.1	57.7	58
Boat	29.4	38.3	39.9
Bottle	27.9	22.7	24.2
Bus	56.6	68.3	68.9
Car	57	55.9	57.6
Cat	65.9	81.4	82.5
Chair	26.5	36.2	36.3
Cow	48.7	60.8	61.1
Table	39.5	48.5	49.1
Dog	66.2	77.2	80.3
Horse	57.3	72.3	72.7
M-bike	65.4	71.3	71.9
Person	53.2	63.5	64.2
Plant	26.2	28.9	29
Sheep	54.5	52.2	54.5
Sofa	38.1	54.8	55.2
Train	50.6	73.9	73.9
TV	51.6	50.8	51.7
Average recognition rate	49.6	57.9	58.7

Table 7: Comparison of test results for time performance.

Device	R-CNN (s)	YOLOv1 (s)	The modified YOLOv1 (s)
GPU time/image	6.9	0.14	0.11

the YOLO network takes 0.14 seconds, and our model takes 0.11 seconds. Figures 7 and 8 show the testing results on Pascal VOC 2007 and Pascal VOC 2012 dataset images [41].

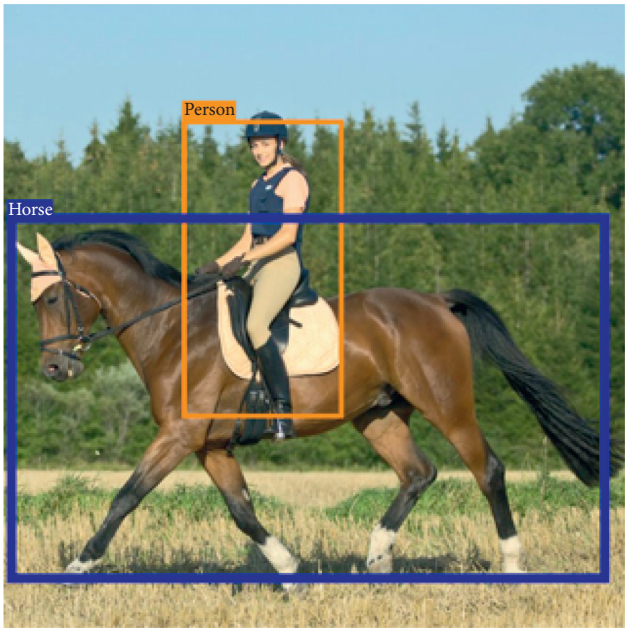


Figure 7: Testing results of our model on Pascal VOC 2007.

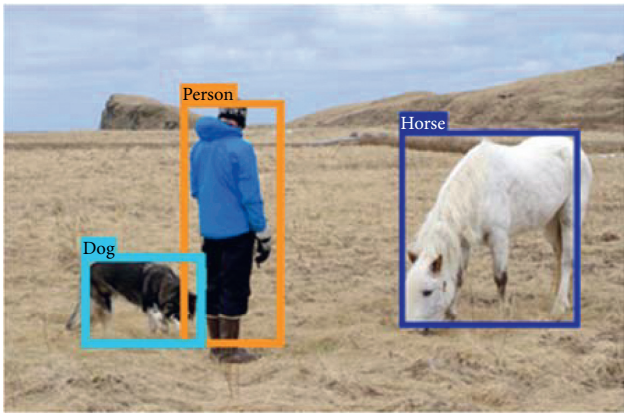


Figure 8: Testing results of our model on Pascal VOC 2012.

From the testing results, the robustness of the improved network is noticed; it classifies each class accurately and detects the desired class.

6. Conclusion

In this paper, we proposed YOLOv1 neural network based object detection by modifying loss function and adding spatial pyramid pooling layer and inception module with convolution kernels of $1 * 1$. The new network is trained on an end-to-end method, and the extensive experiment on a challenging Pascal VOC dataset, 2007/2012, shows the effectiveness of the improved new network, with the detection results being 65.6% and 58.7%, respectively. The results of the proposed network have been compared with those of R-CNN and YOLOv1, from which the effectiveness of the proposed method is demonstrated.

In the future, we expect to extend our work further to make our own benchmark dataset and a hybrid detector for small object detection.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding this paper.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2018YFC0831404 and the State Grid Corp of China Science and Technology Project "Research on Key Technologies of Knowledge Discovery Based ICT System Fault Analysis and Assisted Decision".

References

[1] A. Tiwari, A. Kumar, and G. M. Saraswat, "Feature extraction for object recognition and image classification," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, pp. 2278–0181, 2013.

[2] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2497–2504, New York, NY, USA, 2014.

[3] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100,000 object classes on a single machine," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1814–1821, New York, NY, USA, 2013.

[4] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[5] C.-J. Du, H.-J. He, and D.-W. Sun, "Object classification methods," in *Computer Vision Technology for Food Quality Evaluation*, pp. 87–110, Elsevier, Berlin, Germany, 2016.

[6] K. W. Eric, Li Yueping, N. Zhe, Y Juntao, L. Zuodong, and Z. Xun, "Deep fusion feature based object detection method for high resolution optical remote sensing images," *Applied Science*, vol. 34, 2019.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedigs of the International Conference on Computer Vision & Pattern Recognition (CVPR'05)*, pp. 886–893, Berlin, Germany, 2005.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, Las Vegan, NV, USA, 2016.

[10] Y. Zheng, C. Zhu, K. Luu, C. Bhagavatula, T. H. N. Le, and M. Savvides, "Towards a deep learning framework for unconstrained face detection," in *Proceedings of the 2016 IEEE 8th International Conference on BiometricsTheory, Applications and Systems (BTAS)*, IEEE, New York, NY, USA, pp. 1–8, 2016.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, New York, NY, USA, 2014.

[12] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Berlin, Germany, 2015.

[13] W. Liu, D. Anguelov, D. Erhan et al., "Single shot multibox detector," *European Conference on Computer Vision*, vol. 45, pp. 21–37, 2016.

[14] T.-Yi Lin, P. Dollár, R. B. Girshick et al., "Feature pyramid networks for object detection," *IEEE CVPR*, vol. 43, pp. 936–944, 2017.

[15] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," *Computer Vision-ECCV 2016*, vol. 43, pp. 21–37, 2016.

[16] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, C. Alexander, and C. Berg, "DSSD: Deconvolutional Single Shot Detector," *CoRR*, vol. 45, 2017.

[17] M. Lin, Q. Chen, and S. Yan, "Network in-network," 2013.

[18] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 61, pp. 91–99, 2015.

[20] Z. Zeng, J. Zhang, and X. Wang, "Place recognition an overview of vision perspective," *Applied Science*, vol. 15, 2018.

[21] X. Wan, "A comparative study of cross-lingual sentiment classification," in *Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 1, pp. 24–31, Macau, China, 2012.

[22] J. Gu and C. Lan, "Joint pedestrian and body part detection via semantic relationship learning app," *Journal of Machine Learning Research*, vol. 9, 2019.

[23] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[24] A. Humayun, F. Li, and J. M. R. I. G. O. R. Rehg, "Reusing inference in graph cuts for generating object regions," in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 336–343, Columbus, OJ, USA, June 2014.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep CNNs," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1097–1105, Lake Tahoe, NV, USA, December 2012.

[26] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," 2016.

[27] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2247–2256, 2015.

[28] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 10, pp. 1797–1801, 2014.

[29] X. Chen, "Vehicle detection in satellite images by parallel deep convolutional neural networks," in *Proceedings of the 2013 2nd IAPR Asian Conference on Pattern Recognition (ACPR)*, vol. 45, pp. 181–185, New York, NY, USA, 2013.

[30] Y.-K. Park, J.-K. Park, H.-I. On, and D.-J. Kang, "Convolutional neural network-based system for vehicle front-side detection," *Journal of Institute of Control, Robotics and Systems*, vol. 21, no. 11, pp. 1008–1016, 2015.

[31] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," 2016.
[32] H. Wang, Y. Cai, and L. Chen, "A vehicle detection algorithm based on deep belief network," *The Scientific World Journal*, vol. 2014, 2014.
[33] K. Kim, S. Lee, J.-Y. Kim, M. Kim, and H.-J. Yoo, "A configurable heterogeneous multicore architecture with cellular neural network for realtimeobject recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 11, pp. 1612–1622, 2009.
[34] N. Sudha, A. R. Mohan, and P. K. Meher, "A self-configurable systolic architecture for face recognition system based on principal component neural network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 8, pp. 1071–1084, 2011.
[35] K. He, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 1904–1916, 2014.
[36] T.-Yi Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE ICCV*, vol. 43, pp. 2999–3007, 2017.
[37] J Liu, M. Shah, B. Kuipers, and S. Savarese, "Cross-view action recognition via view knowledge transfer," in *Proceedings of the CVPR 2011*, pp. 3209–3216, Colorado Springs, CO, USA, June 2011.
[38] L. Wu, S. C. Hoi, and N. Yu, "Semantics-preserving bag-of-words models and applications," *IEEE Transaction Image Processing*, vol. 19, pp. 1908–1920, 2010.
[39] R. Girshick, "Fasr r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Boston, MA, USA, June 2015.
[40] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," *CVPR*, vol. 45, 2006.
[41] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
[42] L. Maaten and G. Hinton, *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.