# Single Shot MultiBox Detector on WIDER FACE

**Yifan Shao**

Nanjing University, Nanjing, Jiangsu Province, China 210046

**Keywords:** Convolutional Neural Network, Face Detection, Single Shot Detection, Real-time Object Detection

**Abstract:** We try SSD, which is a method for detecting objects in images using a single deep neural network, on the data set called WIDER FACE. The approach, named SSD, discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes. SSD is simple relative to methods that require object proposals because it completely eliminates proposal generation and subsequent pixel or feature resampling stages and encapsulates all computation in a single network. This makes SSD easy to train and straightforward to integrate into systems that require a detection component. Former writers has tried experiments on the PASCAL VOC, COCO, and ILSVRC datasets confirm that SSD has competitive accuracy to methods that utilize an additional object proposal step and is much faster, while providing a unified framework for both training and inference. This article has established on WIDER FICE data set, which has 61 collections, each of them is split into train, validation, and test sets.

## 1. Introduction

Currently, the most advanced target detection system is a variation of the following methods: Suppose a bounding box, resample pixels or features for each box, and apply a high-quality classifier. Since the selective search, this pipeline has played a leading role in the detection benchmark, and now it is in Pascal VOC, coco and Ilsvrc detection is based on faster r-cnn, although it has deeper features. Although these methods are accurate, but for embedded systems, the amount of calculation is too large, even for high-end hardware, for real-time applications is too slow

We choose the model which is trained 8000 times and 10000 times to test our model. In the result picture, we can clearly see that the almost all faces is detected and signed. We got the greatest result for testing 15 faces in 1 picture by using the model which is trained 10000 times. The result is 11 of the faces is signed.

We summarize our contributions as follows:

1) We trained SSD model with a more sufficient data set, and record the loss during each 100 epochs.

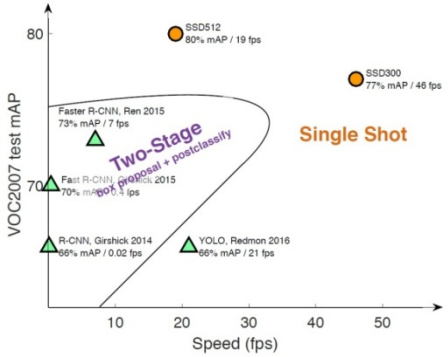
2) We used our trained model to test our real-life picture, to ensure our trained model is helpful on detecting pictures out of the data set

## 2. Related Work

In recent years, target detection has made great progress. The main algorithms are divided into two types: (1) two-stage method, such as r-cnn algorithm. The main idea is to use heuristic method first Search) or CNN network (RPN) produces a series of sparse candidate frames, and then classifies and regresses these candidate frames. The advantage of two stage method is high accuracy; (2) one stage method, such as Yolo and SSD, its main idea is to carry out intensive sampling uniformly at different positions of the picture. Different scales and aspect ratio can be used for sampling, and then CNN can be used to extract features The whole process of direct classification and regression

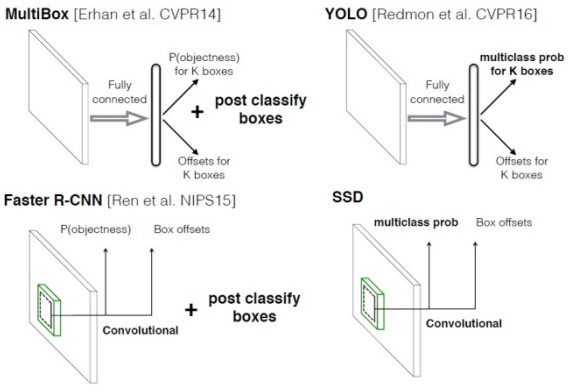
  DOI: 10.25236/mmmce.2020.014

only needs one step, so it has the advantage of high speed, but an important disadvantage of uniform dense sampling is that it is difficult to train. This is mainly because the positive sample and the negative sample (background) are extremely unbalanced (see focal loss), resulting in a slightly lower accuracy of the model SSD algorithm belongs to one stage method. Figure (a) shows the comparison of different detection algorithms.




(a) Performance comparison of different algorithms

Multibox indicates that SSD is a multiframe prediction. Figure (b) shows the basic framework of different algorithms. For faster r-cnn, the candidate frames are first obtained through CNN, and then classified and regressed. Yolo and SSD can complete the detection in one step. Compared with Yolo, SSD uses CNN to detect directly, instead of Yolo's detection after the full connection layer. In fact, using convolution to detect directly is just one of the differences between SSD and Yolo. In addition, there are two important changes. One is that SSD extracts feature maps of different feature maps (feature maps at the front) can be used to detect small objects, while small scale feature maps (feature maps at the back) can be used to detect large objects. The other is that SSD uses different scales Prior boxes, default boxes, called anchors in fast r-cnn. Yolo algorithm is difficult to detect small targets, and the location is not accurate, but these important improvements make SSD overcome these shortcomings to some extent.

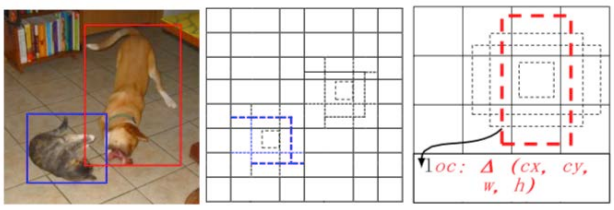



(b) Basic frame diagrams of different algorithms

## 3. Architecture

### 3.1 Multiscale feature map for detection

The so-called multi-scale feature map with different sizes is adopted. In general, the feature map in front of CNN network is relatively large, and the convolution or pool with stripe = 2 will be gradually used later to reduce the size of the feature map. As shown in Figure 3, a relatively large feature map and a relatively small feature map are used for detection. The advantage of this method is that the larger feature map is used to detect the smaller target, while the smaller feature map is used to detect the larger target. As shown in Figure 4, the 8X8 feature map can be divided into more units, but the prior frame scale of each unit is smaller.




(1) Image with GT-box (2)8*8 feature map (3) 4*4feature map

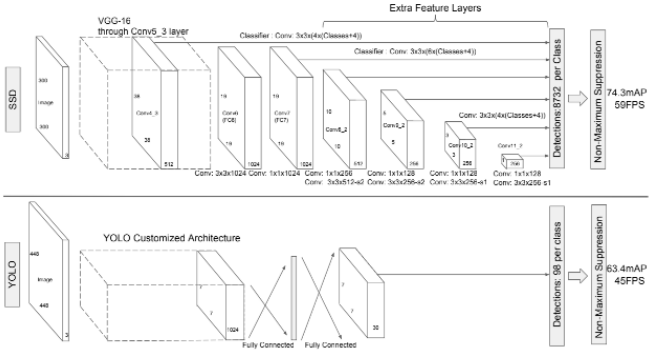
### 3.2 Set a priori box

SSD uses the concept of anchor in faster r-cnn for reference. Each unit sets a priori box with different scale or aspect ratio. The predicted bounding boxes are based on these priori boxes, which can reduce the training difficulty to a certain extent. In general, each unit will be set with multiple priori boxes, with different dimensions and aspect ratio. As shown in Figure 5, it can be seen that each unit uses four different priori boxes. In the picture, the cat and dog use the priori box that is most suitable for their shape to train.

For each prior box of each cell, it outputs a set of independent detection values, corresponding to a bounding box, which is mainly divided into two parts. The first part is the confidence degree or score of each category. It is worth noting that SSD regards the background as a special category. If there are c categories in total, SSD actually needs to predict c+1 confidence values. The first confidence degree refers to the score that does not contain the target or belongs to the background. Later, when we talk about the confidence of c categories, please keep in mind the special category that contains the background, that is, the real detection category only has c-1. For a feature map with a size m*n, there are mn cells in total, and the number of prior boxes set for each cell is recorded as k, then each cell needs (c+4)*k prediction values in total, and all cells need (c+4)*kmn prediction values in total. Because *SSD* uses convolution for detection, it needs (c+4)*k convolution kernels to complete the detection process of this feature map.

### 3.3 Model

SSD uses vgg16 as the basic model, and then adds a convolution layer on the basis of vgg16 to obtain more feature maps for detection. The network structure of SSD is shown in Figure (a). Above is SSD model, and below is Yolo model. It can be seen that SSD uses multi-scale feature map for detection.




(a) SSD network structure

Vgg16 is used as the basic model. First, vgg16 is pre trained in ilsvrc cls-loc data set, The full connection layer FC6 and fc7 of vgg16 are converted into convolution layer conv6 and convolution layer conv7 respectively. At the same time, pool5 of pooling layer is changed from the original 2x2 of strip = 2 to 3x3 of strip = 1. In order to match this change, Atrius Algorithm is adopted, in fact, is conv6, which uses the expanded convolution or the convolution with holes. It expands the convolution field exponentially without increasing the parameters and the complexity of the model. It uses the expansion rate parameter to express the expansion size.

Among them, the conv4-3 layer in vgg16 will be used as the first feature map for detection. The size of the feature map of conv4-3 layer is 38x38, but this layer is relatively ahead, and its norm is

large, so an L2 normalization layer is added behind it to ensure that the difference between the detection layer and the later layer is not very big. This is different from the batch normalization layer, which only normalizes each pixel in the channel dimension, while batch The normalization layer is based on three dimensions of [batch size, width, height].

From the convolution layer added later, conv7, conv8, conv9, conv10 and conv11 are extracted as the feature images for detection. In addition, conv4 and conv3 layers are added, 6 feature images are extracted, and their sizes are (38,38),(19,19),(10,10),(5,5),(3,3),(1,1) respectively , but the number of priori boxes set in different feature graphs is different The setting of prior box includes two aspects: Scale (or size) and aspect ratio. For the scale of the prior frame, it follows a linear increasing rule: with the decrease of the size of the feature map, the scale of the prior frame increases linearly.

$$s_k + = + s_{min} + + + \frac{s_{max} + - + s_{min}}{m - 1}(k - 1), + k \in [1, m] \quad (1)$$

Where m refers to the number of characteristic graphs, $s_k$ represents the scale of the prior box size relative to the picture, while $s_{min}$ and $s_{max}$ represent the minimum and maximum values of the scale.

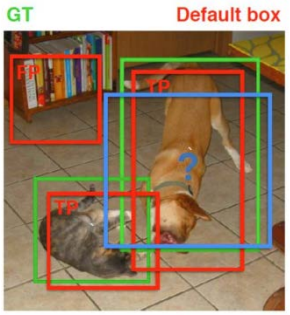
## 4. Train

In the process of training, first of all, it is necessary to determine which priori box the ground truth (real target) in the training picture matches with, and the boundary box corresponding to the matched priori box will be responsible for predicting it. There are two main matching principles between the prior frame of SSD and the ground truth.

First of all, for each ground truth in the picture, find the prior box with the largest IOU, which matches it, so that each ground truth must match a prior box. In general, the prior box matching the ground truth is called a positive sample. On the contrary, if a prior box does not match any ground truth, then the prior box can only match the background, which is a negative sample. There are very few ground truth in a picture, but there are many prior boxes. If you only match according to the first principle, many prior boxes will be negative samples, and the positive and negative samples are extremely unbalanced, so the second principle is needed.

The second principle is: for the remaining unmatched priori box, if the IOU of a certain ground truth is greater than a certain threshold (generally 0.5), then the priori box also matches the ground truth. This means that a certain ground truth may match multiple prior boxes, which is OK. But it can't be reversed, because a priori box can only match one ground truth. If multiple ground truths and a priori box IOU are greater than the threshold value, then the priori box only matches the ground truth with the largest IOU. The second principle must be carried out after the first principle. If the maximum IOU corresponding to a certain ground truth is less than the threshold value, and the matched prior box is greater than the threshold value with the IOU of another ground truth, the prior box should match the former.

First, make sure that a certain ground Truth must have a prior box to match it. But there are many priori boxes, the maximum IOU of a certain ground truth must be greater than the threshold value, so it is possible to only implement the second principle. The following figure is a matching diagram, in which the green GT is the ground truth, the red is the prior box, FP is the negative sample, and TP is the positive sample.




(a)prior box

Although a ground truth can match multiple priori boxes, there are too few ground truth priori boxes, so there will be a lot of negative samples relative to positive samples. In order to ensure the balance of positive and negative samples as much as possible, SSD uses hard negative mining to sample negative samples. During sampling, it arranges them in descending order according to the confidence error (the smaller the confidence of the prediction background is, the larger the error is). Top-k with larger error is selected as the negative sample for training to ensure that the proportion of positive and negative samples is close to 1:3.

Loss function:

$$L(x,+c,+l,+g)+=+\frac{1}{N}(L_{conf}(x,c)+++\alpha+L_{loc}(x,l,g)) \quad \text{(i)}$$

Where N is the number of positive samples in the prior box. Here $x_{ij}^{p} \in +\{+1,0+\}$ is an indication parameter. When $x_{ij}^{p}=+1$, it means that the apriori box of i matches the ground truth of j , and the category of ground truth is p. c is the predicted value of category confidence. l is the predicted value of the position of the corresponding bounding box of the prior box, and g is the position parameter of the ground truth. For position error, smooth L1 loss is adopted, which is defined as follows:

$$L_{loc}(x,l,g)=\sum_{i\in Pos}^{N}\sum_{m\in\{cx,cy,w,h\}} x_{ij}^{k}\text{smooth}_{\text{L1}}(l_i^m-\hat{g}_j^m)$$

$$\hat{g}_j^{cx}=(g_j^{cx}-d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy}=(g_j^{cy}-d_i^{cy})/d_i^h$$

$$\hat{g}_j^{w}=\log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^{h}=\log\left(\frac{g_j^h}{d_i^h}\right)$$

$$\text{smooth}_{L_1}(x)=\begin{cases}0.5x^2 & \text{if } |x|<1\\ |x|-0.5 & \text{otherwise,}\end{cases} \quad \text{(ii)}$$

Performance evaluation: First, take a look at the performance of SSDs on voc2007, voc2012 and coco datasets as a whole, as shown in Table 1. In contrast, the performance of ssd512 will be better. The * representation uses the image expansion data authentication technique to improve the detection effect of SSDs on small targets, so the performance will be improved.

| Method | VOC2007 test | | VOC2012 test | | COCO test-dev2015 trainval35k | | |
|---|---|---|---|---|---|---|---|
| | 07+12 | 07+12+COCO | 07++12 | 07++12+COCO | | | |
| | 0.5 | 0.5 | 0.5 | 0.5 | 0.5:0.95 | 0.5 | 0.75 |
| SSD300 | 74.3 | 79.6 | 72.4 | 77.5 | 23.2 | 41.2 | 23.4 |
| SSD512 | 76.8 | 81.6 | 74.9 | 80.0 | 26.8 | 46.5 | 27.8 |
| SSD300* | 77.2 | 81.2 | 75.8 | 79.3 | 25.1 | 43.1 | 25.8 |
| SSD512* | **79.8** | **83.2** | **78.5** | **82.2** | **28.8** | **48.5** | **30.3** |

(1) performance of SSDs

As we can see in Table 2, We compare the parameters of different SSD networks, the size of network structure space, and the important parameter FLOPs that reflect the computing power required by the network model, and we can see that the model of ssd512 is the most complex and re-

quires the largest computing power.

| model | Input size | param memory | Feature memory | flops |
|---|---|---|---|---|
| vgg-s | 224*224 | 393MB | 12MB | 3 GLOPs |
| ssd-vggvd-512 | 512*512 | 104MB | 337MB | 91 GFLOPs |
| ssd-vggvd-300 | 300*300 | 100MB | 116MB | 31 GFLOPs |

(2) param size and complexity of different network

The following table shows the impact of different combinations of checks on SSD performance. From the table, we can draw the following conclusions:

1) Data amplification technology is very important for the improvement of map;

2) The prior frame with different aspect ratio can get better results;

| | SSD300 | | | | | |
|---|---|---|---|---|---|---|
| more data augmentation? | | ✔ | ✔ | ✔ | ✔ | ✔ |
| use conv4_3? | ✔ | | ✔ | ✔ | ✔ | ✔ |
| include $\{\frac{1}{2}, 2\}$ box? | ✔ | ✔ | | ✔ | ✔ | ✔ |
| include $\{\frac{1}{3}, 3\}$ box? | ✔ | ✔ | | | ✔ | ✔ |
| use atrous? | ✔ | ✔ | ✔ | ✔ | | ✔ |
| VOC2007 `test` mAP | 65.4 | 68.1 | 69.2 | 71.2 | 71.4 | 72.1 |

(3) Impact of different combinations

## 5. Experiment

Data Set: Face detection target is very simple: for an image, tell whether there is a face, if there is a face, return the position of all faces in the image; face scale, occlusion, light, expression change, make-up, posture change and other situations, face detection task is still blocked and long;

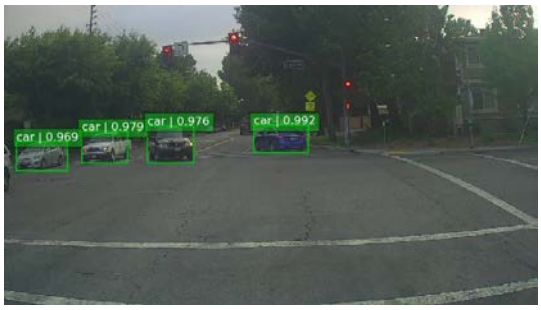
The face detection algorithm library and the detection algorithm itself are mutually promoting. With a larger and closer algorithm library to the real scene, we can be inspired to design better algorithms and solve all kinds of challenges in face detection. However, at present, the existing face detection algorithm library is generally too small in data scale and too single in scene to cover the situation in the real scene;

The data set of wider face contains 32203 face images and 393703 faces. The data volume is 10 times higher than that of the current database, and various scenes are very complex. In order to analyse all kinds of false detections in depth, we also label each face bbox with multiple attributes: occlusion, posture, events, etc., which can evaluate the performance of the algorithm from all directions and angles;

Training Process: For each prediction box, first determine its category (the one with the highest confidence) and confidence value according to the category confidence, and filter out the prediction box belonging to the background. Then according to the confidence threshold, filter out the prediction boxes with lower thresholds. For decoding the left prediction frame, the real position parameters of the prediction frame are obtained according to the prior frame clip To prevent the position of the prediction box from exceeding the picture). After decoding, it is generally required to arrange in descending order according to the confidence, and then only keep top-k (e.g Four hundred) Forecast boxes. Finally, it's going on NMS The algorithm filters out those prediction boxes with large overlap. Finally, the remaining prediction box is the detection result.

SSD has an open source implementation in many frameworks. Here, we implement SSD's influence process based on the pytorch version .We set $s_{min} = 0.15, +s_{max} = 0.9$ which is different from that in paper.

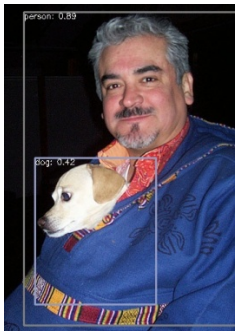
Firstly, we use VOC2017 to train our model, and here is the basic result.




(a) Train result of VOC2017

Then we use the data set wider face to recognize and detect the face. We use the pre trained vgg19 as the pre-training model, and import the data set to experiment. The results are as follows

(b) Train result of WIDER FACE



(c) Train result of WIDER FACE

For the larger face, our detection effect is very good, we can basically identify all the faces in the picture, but for some small faces, limited by the performance of the backbone network, the recognition effect is not very ideal. Here is the result:

(d) 8000 iterations Vs. 10000 iterations

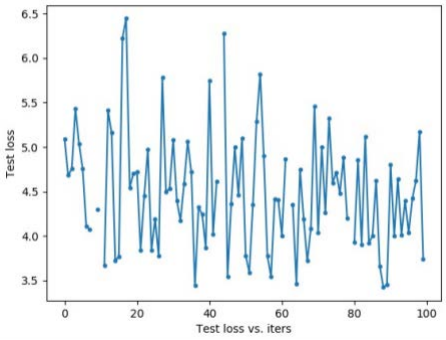
It can be seen from the above figure that the effect of different iterations is different, but from 8000, increasing the number of iterations cannot significantly improve the recognition effect, and the same face will have different recognition results in different iterations of the model.

## 6. Data Analysis

We found that for the different model with different trained times, the test result seemed not sta-

ble. For some faces in the 8000 times iterations experiment, it can be detected by the model, but cannot be detected by the model which is trained 10000 times. If we only see the result by the loss function, the 10000-time-trained model is clearly better than then 8000 one.

Another thing is, the test result is still not satisfying, though it uses different functions to reinforce the model and to decline the influence of the noise. We assume that the back-bone net is not powerful and deep enough to train. So changing the back-bone net for VGG-16 to Res-net50 will be a good decision.




(a) Test loss

## 7. Conclusions

This paper introduces SSD, a kind of fast single object detector for many kinds. One of the key features of our model is the output of multi-scale convolutional bounding box with multiple feature maps attached to the top of the network. This representation allows us to effectively simulate possible box shaped spaces. Experiments show that a large number of carefully selected default bounding boxes can improve performance given appropriate training strategies. Compared with the existing methods, our SSD model has at least one order of magnitude box prediction sampling location, scale and aspect ratio [5,7]. We prove that under the same vgg-16 infrastructure, SSD is superior to its most advanced object detector counterpart in terms of accuracy and speed. We use wider face data set to train our model, which contains 32203 face images and 393703 faces.

## References

[1] Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. IJCV (2013)

[2] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS. (2015)

[3] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)

[4] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: ICLR. (2014)

[5] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. (2016)

[6] Girshick, R.: Fast R-CNN. In: ICCV. (2015)

[7] Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: CVPR. (2014)

[8] Szegedy, C., Reed, S., Erhan, D., Anguelov, D.: Scalable, high-quality object detection. arXiv preprint arXiv: 1412.1441 v3 (2015)