# REAL-TIME OBJECT DETECTION BY A MULTI-FEATURE FULLY CONVOLUTIONAL NETWORK

*Yajing Guo*[1], *Xiaoqiang Guo*[2], *Zhuqing Jiang*[1], *Aidong Men*[1], *Yun Zhou*[2]

[1]Beijing University of Posts and Telecommunications, Beijing, China,100876
{gyj,jiangzhuqing,menad}@bupt.edu.cn
[2]Academy of Broadcasting science, Beijing, China, 100866
{guoxiaoqiang,zhouyun}@abs.ac.cn

## ABSTRACT

Prior work on object detection depends on region proposals to guide the search for object instances. Generally, several thousand proposals must be processed, thus hurting the detection efficiency. In this paper, we propose a new model free from region proposals for object detection which treats detection task as a regression problem. To improve small-size object detection and localization, we employ the deep hierarchical features extracted from convolutional neural networks (CNNs). The hierarchical architecture combines appearance information from a shallow layer with semantic information from a deep layer. Our approach can predict bounding boxes and class probabilities simultaneously from a full input image. We transfer a classification network called Darknet into fully convolutional network and fine-tune it for the detection task. Experiments on PASCAL VOC dataset demonstrate that our approach outperforms other detection models.

***Index Terms***— Real-time object detection, multi-feature, fully convolutional network

## 1. INTRODUCTION

Current state-of-the-art object detection approaches are moving from dense sliding window based methods like DPM [1] to sparse region proposals based methods like Selective Search [2] and EdgeBoxes [3]. Object proposals generation methods can effectively reduce the number of candidate bounding boxes and improve the detection accuracy by enabling more complicated learning mechanism than sliding window based methods. Since Krizhevsky et al. [4] won the ILSVRC2012, convolutional neural network (CNNs) have been widely applied to object recognition. Girshick et al. [5] applies CNNs to bottom-up region proposals generated by Selective Search, namely R-CNN, remarkably improving mean average precision (mAP) relative to the previous best result. Fast versions [6] [7] [8] with higher accuracy and speed are also developed. The series of R-CNN methods replace the hand-crafted features such as SIFT [9], HOG [10] and LBP [11] with high level semantic representation produced by CNNs, achieving state-of-the-art performance. Although these methods use several hundred or thousand region proposals to reduce searching space for an image, they still can't achieve real-time detection.

Recently, a unified real-time object detection model called YOLO [12] is proposed which treats detection task as a single regression problem. YOLO is extremely fast with a simple pipeline. Using global context information, YOLO can make less background errors than Fast R-CNN [7], but struggles to localize objects correctly. Thus it still lags behind state-of-the-art detection systems in accuracy. Liu et al. [13] integrate the anchor mechanism of Faster R-CNN into a single regression network. This method boosts speed by eliminating bounding box proposals generation and improves accuracy by a series improving methods including offsets in bounding box locations, using different filters for different aspect ratio detections and so on.

Fully Convolutional Networks (FCN) [14] is demonstrated that convolutional neural networks can obtain impressive performance on semantic segmentation task. In [14], the authors combine fine appearance information from a shallow layer with coarse semantic information from a deep layer. Kong et.al propose HyperNet [15] which combine deep, coarse information with shallow, fine information to make features more abundant. However this method is based on Region Proposal Network (RPN) [8] so it cannot make detection real-time.

Our work is motivated by the following problems. First, object region proposals based detection systems still cannot achieve real-time detection because proposals generation consumes much running time. Second, recent real-time object detection systems have low detection accuracy. In this paper, we present a multi-feature fully convolutional network (MFCN) for real-time detection which can predict bounding boxes and class probabilities simultaneously from a full input image. To improve small-size object detection and localization,

The study described in this paper was funded by National Natural Science Foundation of China(No.61671077 and No.61671264).

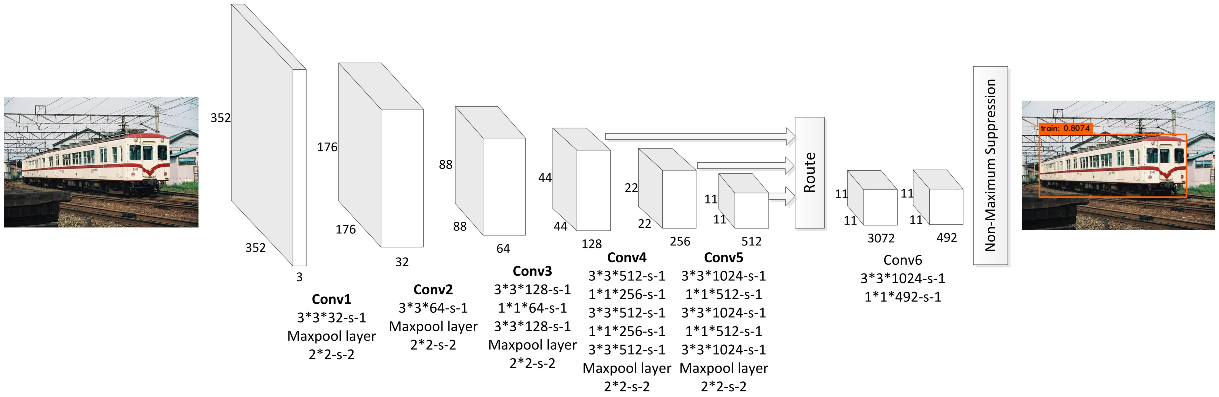

**Fig. 1**. **Our object detection framework.** The model is multi-feature fully convolutional network. The convolutional layers before route layer (Conv1, Conv2, Conv3, Conv4 and Conv5)extract features while the following convolutional layers predict the output coordinates and probabilities. Each set of convolutions comprises 1-5 convolutional layers. The route layer concatenates fine shallow features and coarse deep features.

the multi-feature architecture combines appearance information from a shallow layer with semantic information from a deep layer to make features more abundant.

The rest of this paper is organized as follows. Section 2 introduces the network architecture of detection model and our loss function. Evaluation of the proposed method is shown in Section 3, and Section 4 gives conclusions of our work.

## 2. THE PROPOSED METHOD

In this section we first briefly describe the Darknet [16] framework. Next, we elaborate our network design for object detection. Finally, training approaches and the loss functions are introduced.

### 2.1. Darknet

Most detection models adopt VGG-16 [17] as the feature extractor which is a powerful classification framework with needless complexity. The author of YOLO propose a custom network called Darknet which is faster than VGG-16. The Darknet is based on the Googlenet architecture [18]. Although Darknet is slightly worse than VGG-16 in accuracy, Darknet is much faster than VGG-16. Similar to the VGG networks, the Darknet uses mostly 3×3 filters and double the number of channels after each max pooling step. Also, following the work on Network in Network [19], the Darknet adopts 1×1 filters to compress the feature representation between 3×3 convolutions.

### 2.2. Our detection network

Our detection framework is illustrated in Fig.1. We combine separate components of object detection into a single neural network. Our network takes the entire image as input and predicts all bounding boxes across all classes simultaneously. First, the input image is resized to $352 \times 352$. Then five groups of convolutional layers are used to extract features, generating different scale feature maps. The route layer is used to integrate multi-scale feature maps into a whole. Finally, the following convolutional layers produce a fix-sized collections of bounding boxes and confidence scores for the object instances, followed by a non-maximum suppression step to generate the final detections.

#### *2.2.1. The model design*

The model divides the input image into $S \times S$ grids. The grid has the responsibility for predicting the category of object if the center of that object falls in a grid cell. Each grid cell predicts $B$ bounding boxes with confidence scores. The confidence score is defined as $P(object) * IoU_{pred}^{truth}$, reflecting the confidence that the bounding box contains an object and also the accuracy that the predictor overlaps ground truth. Each bounding box contains $4+C$ values: $x$, $y$, $w$, $h$ and confidence for all $C$ catogories. $P(object)$ is 1 if the grid cell contains any object and is 0 if no object exits in the grid cell. Thus the confidence value represents the IoU(Intersection-over-Union) between ground truth and predicted bounding box.

Each grid cell also predicts $C$ conditional probabilities $Pr(class_i|object)$, representing the probabilities that an object contained in the grid cell belongs to class $i$. Therefore, we encoded these predictions as an $S \times S \times (B * (4 + C) + C)$ tensor. In practice, we use $S = 11$ to make less error for small object detection, compared with $S = 7$ in YOLO. For evaluating on the PASCAL VOC dataset [20], the classes number $C = 20$.

### 2.2.2. *Multi-feature concatenation*

We add multi-feature structure to the network to improve the performance because both shallow and deep features are important for object detection. Most of state-of-the-art detectors adopt the final feature map. However different feature maps generated by different layers usually represent various information. For example, feature maps generated by shallow layers have better localization but with a lower recall while those generated by deep layers have a higher recall but struggle with better localization performance. Our model fuses the output by Conv3, Conv4 and Conv5. To solve the different resolution problem, we down-sample feature maps from Conv3 and Conv4.

### 2.2.3. *Anchor boxes and default aspect ratios*

For each grid cell, we assign a set of default anchor boxes. Our anchor boxes are similar to that used in Faster R-CNN [8]. However, we assign one of anchor boxes as ground truth box which is different from Faster R-CNN. Because the anchor boxes tile the feature map in a convolutional manner, the position of each box instance relative to its corresponding cell is fixed. For each grid we set 6 anchor boxes with 2 scales and 3 aspect ratios. The aspect ratios are set as $\{1{:}1, 1{:}2, 2{:}1\}$ and the scales are 1 and 2 of corresponding grid cell (for multi-feature maps, containing $32^2$, $64^2$, $128^2$ and $256^2$). By applying anchor boxes to different feature maps, we can efficiently discretize the space of possible output box shapes.

We implement our model as a fully convolutional neural network, removing the fully connected layers of Darknet framework. The convolutional layers before route layers extract features from the input image, while the route layer concatenates fine shallow features and coarse deep features. Then the following convolutional layers are used to predict the output coordinates and probabilities. The final output of our model is a $11\times11\times492$ tensor.

## 2.3. Training approaches

We construct our model based on the Darknet which is trained on the ImageNet 1000-class competition dataset [21]. We convert the classification model to perform detection. Thus we fine-tune the convolutional layers before route layer and randomly initialize weights of following convolutional layers. To improve detection performance, we need to increase the resolution of the input image from 224×224 to 352×352, but the resolution is still less than $448\times448$ for YOLO.

Our loss function is a weighted sum of the confidence loss (con) and localization loss (loc), defined as:

$$L(c, c^\star, t, t^\star) = L_{con}(c, c^\star) + \lambda L_{loc}(t, t^\star) \qquad (1)$$

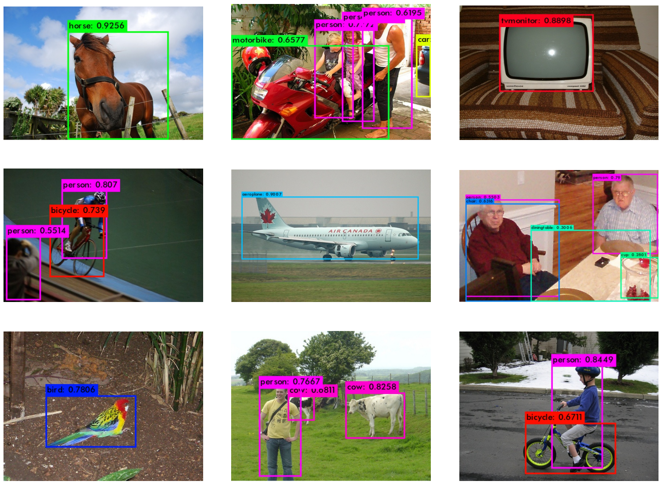

**Fig. 2**. Detection examples on PASCAL VOC2012 test set with our model. **Best viewed in color**.

For confidence loss,

$$L_{con}(c, c^\star) = \sum_{i=0}^{S^2} \sum_{j=0}^{B} P_{ij}(c_i - c_i^\star)^2 \qquad (2)$$

Where $P_{ij}$ denotes that the $j$th bounding box predictor in cell $i$ is responsible for that prediction. The loss function only penalizes classification error if an object is present in that grid cell.

For localization loss, we use the smooth $L_1$ loss between the predicted box ($t$) and the ground truth box ($t^\star$) parameters.

$$L_{loc}(t, t^\star) = \sum_{i \in x,y,w,h} smoothL_1(t_i - t^\star) \qquad (3)$$

in which

$$smoothL_1(x) = \begin{cases} 0.5x^2 & if|x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \qquad (4)$$

We regress to offsets for the center of the bounding box and for its width and height. The network predicts 5 values for each bounding box, $t_x$, $t_y$, $t_w$, $t_h$ and $t_c$. The predictions correspond to:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_w e^{t_h} \\ P(object) * IoU_b^{truth} &= \sigma(t_c) \end{aligned} \qquad (5)$$

The cell is offset from the top left corner of the image by $(c_x, c_y)$. $p_w$ and $p_h$ are bounding box priors for width and height respectively. The weight term $\lambda$ is set to 1 by cross validation. By this way, the parameters are easier to learn and the network can be trained stably.

**Table 1**. **PASCAL VOC2012 test detection results.** Each model was trained on PASCAL VOC2012 trainval and VOC2007 trainval and test set. Fast and Faster R-CNN use images with minimum dimension 600, while the image size for YOLO and SSD300 is 448 × 448 and 300 × 300 respectively.

| Method | mAP | areo | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fast R-CNN [7] | 68.4 | 82.3 | 78.4 | 70.8 | 52.3 | 38.7 | 77.8 | 71.6 | 89.3 | 44.2 | 73.0 | 55.0 | 87.5 | 80.5 | 80.8 | 72.0 | 35.1 | 68.3 | 65.7 | 80.4 | 64.2 |
| Faster R-CNN [8] | 70.4 | 84.9 | 79.8 | 74.3 | 53.9 | 49.8 | 77.5 | 75.9 | 88.5 | 45.6 | 77.1 | 55.3 | 86.9 | 81.7 | 80.9 | 79.6 | 40.1 | 72.6 | 60.9 | 81.2 | 61.5 |
| YOLO [12] | 57.9 | 77.0 | 67.2 | 57.7 | 38.3 | 22.7 | 68.3 | 55.9 | 81.4 | 36.2 | 60.8 | 48.5 | 77.2 | 72.3 | 71.3 | 63.5 | 28.9 | 52.2 | 54.8 | 73.9 | 50.8 |
| SSD300 [13] | 72.4 | 85.6 | 80.1 | 70.5 | 57.6 | 46.2 | 79.4 | 76.1 | 89.2 | **53.0** | 77.0 | **60.8** | 87.0 | **83.1** | 82.3 | 79.4 | 45.9 | 75.9 | **69.5** | 81.9 | 67.5 |
| our MFCN | **73.2** | **86.1** | **82.0** | **74.4** | **59.2** | **50.8** | **79.6** | **76.2** | **90.2** | 52.1 | **78.2** | 58.1 | **89.0** | 82.5 | **83.4** | **81.1** | **48.5** | **77.1** | 62.4 | **83.6** | **68.2** |

# 3. EXPERIMENTS

## 3.1. Experimental settings

We conduct our experiments on the PASCAL VOC2007 and 2012 dataset [20] and compared with state-of-the-art methods. The dataset contains tens of thousands of images from 20 categories with bounding box annotation for each object. The VOC2012 trainval and VOC2007 trainval and test (21503 images) are used for training, while VOC2012 test (10991 images) is used for testing.

We construct our model by fine-tuning the Darknet. The training process uses stochastic gradient descent (SGD) to minimize the loss for tuning the model. During fine-tuning, each SGD mini-batch is sampled from 1 image and the mini-batch size is 64. We use a momentum of 0.9 and a decay of 0.0005.

## 3.2. Experimental results

We compare our model with other state-of-the-art detection methods on PASCAL VOC2012 dataset from two aspects: accuracy and speed.

### *3.2.1. Detection accuracy*

In Table 1, we report the detection accuracy of our proposed method and state-of-the-art methods. Our model achieves 73.2% mAP, which is 4.8 points higher than Fast R-CNN and 2.8 points higher than Faster R-CNN. Moreover, our detection performance slightly exceeds that of SSD300 (72.4%). Compared to YOLO, our MFCN is significantly better, likely due to the use of multiple feature maps and default anchor boxes. Anchor boxes mechanism helps to discretize the space of output box shapes, thus we can achieve outsanding performance for small objects (“bird”, “bottle”, “plant”).

### *3.2.2. Detection speed*

Table 2 compares the detection speed of our MFCN with other methods. Fast R-CNN and Faster R-CNN are region proposals based networks while YOLO, SSD and our MFCN are free from region proposals. Because region proposals generation consumes lot of running time, Fast R-CNN and Faster R-CNN are much slower than our method. Our model outperforms YOLO in both detection accuracy and speed. The speed improvement is mainly due to the size of the input image, 352 for ours and 484 for YOLO respectively. With similar resolution of input images, our method is slightly exceeds SSD300 because we adopt Darknet whereas SSD uses VGG-16. Although the accuracy of MFCN is slightly lower than SSD500, it is 4× faster.

**Table 2**. Detection performance for speed on PASCAL VOC2012 test set. Our MFCN is faster and more accurate than prior detection methods.

| Method | mAP (%) | Time (ms) | FPS |
|---|---|---|---|
| Fast R-CNN [7] | 68.4 | 1830 | 0.5 |
| Faster R-CNN [8] | 70.4 | 142 | 7 |
| YOLO [12] | 57.9 | 22 | 45 |
| SSD300 [13] | 72.4 | 21 | 46 |
| SSD500 [13] | **74.9** | 52 | 19 |
| our MFCN | 73.2 | **13** | **75** |

# 4. CONCLUSIONS

In this paper, we propose an effective model called MFCN (Multi-feature Fully Convolutional Network). The network mainly relies on three key factors: 1)framing object detection as a regression problem can simplify detection pipeline and improve the detection speed, 2) multi-feature concatenation can efficiently fuse shallow and deep information and increase the detection confidence, and 3) anchor boxes mechanism helps to discretize the space of output box shapes. Our approach is free from region proposals and can predict bounding boxes and class probabilities simultaneously from full images. Experiments on PASCAL VOC2012 dataset show good performance of the proposed method.

# 5. REFERENCES

[1] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *TPAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[2] J.R. Uijlings, K.E. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *IJCV*, vol. 104, no. 2, pp. 154–171, 2013.

[3] C. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*, 2014, pp. 391–405.

[4] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *ECCV*. Springer, 2014, pp. 346–361.

[7] R. Girshick, "Fast r-cnn," in *ICCV*, 2015, pp. 1440–1448.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015, pp. 91–99.

[9] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*. IEEE, 2005, vol. 1, pp. 886–893.

[11] X. Wang, T.X Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *CVPR*. IEEE, 2009, pp. 32–39.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *arXiv preprint arXiv:1506.02640*, 2015.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and Reed S.E., "Ssd: Single shot multibox detector," *CoRR, abs/1512.02325*, 2015.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CVPR*, 2015.

[15] T. Kong, A. Yao, Y. Chen, and F Sun, "Hypernet: towards accurate region proposal generation and joint object detection," *CVPR*, 2016.

[16] J. Redmon, "Darknet: Open source neural networks in c," http://pjreddie.com/darknet/, 2013-2016.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, Reed S., D Anguelov, Erhan.D, Vanhoucke.V, and Rabinovich.A, "Going deeper with convolutions," *CoRR, abs/1409.4842*, 2014.

[19] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[20] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*. IEEE, 2009, pp. 248–255.