Rethinking Classification and Localization in R-CNN

Yue Wu¹, Yinpeng Chen², Lu Yuan², Zicheng Liu², Lijuan Wang², Hongzhi Li² and Yun Fu¹
¹Northeastern University, ²Microsoft

Abstract

Modern R-CNN based detectors share the RoI feature extractor head for both classification and localization tasks, based upon the correlation between the two tasks. In contrast, we found that different head structures (i.e. fully connected head and convolution head) have opposite preferences towards these two tasks. Specifically, the fully connected head is more suitable for the classification task, while the convolution head is more suitable for the localization task.

Therefore, we propose a double-head method to separate these two tasks into different heads (i.e. a fully connected head for classification and a convolution head for box regression). Without bells and whistles, our method gains +3.4 and +2.7 points mAP on MS COCO dataset from Feature Pyramid Network baselines with ResNet-50 and ResNet-101 backbones, respectively.

1. Introduction

A majority of two-stage object detectors [6, 7, 22, 3, 16] share the similar structure: learning two tasks (classification and bounding box regression) by sharing the feature extraction head on proposals, as these tasks are highly correlated. Two different head structures are widely used: convolution head (conv5) on single feature map (conv4) in Faster R-CNN [22] and fully connected head (2-fc) on multiple level feature maps in FPN [16]. However, there is a lack of understanding of the correlation between the classification and localization tasks on these two head structures.

Related to this problem, recent COCO Detection 18 Challenge winner (Megvii¹) proposed to combine bounding box regression and segmentation in a convolution head, and leave classification alone in 2-fc head. This motivates us to rethink the classification and localization with respect to different head structures. Intuitively, spatial information is crucial for object classification to determine if a complete object (not just part of the object) is covered by the region proposal. The fully connected head fits well for this task, as it is spatial sensitive. In contrast, the regression task

¹http://cocodataset.org/#detection-leaderboard

Figure 1. Overview of the double-head detector. (a) the original FPN with a fully connected (2-fc) head, (b) modified FPN with a convolution head (used for toy experiment), and (c) our proposed double-head FPN, which splits classification and localization into two heads. The fully connected head is used for classification during inference, with the localization as an auxiliary task during training. The convolution head is used for bounding box regression during inference, with the classification as an auxiliary task during training.
In this paper, we propose a double-head detector, which includes a fully connected head (FC-Head) for classification and a convolution head (Conv-Head) for box regression (see Figure 1-(c)), to leverage the advantage of both heads. Firstly, we found that our double-head design is better than using either head (FC-Head or Conv-Head) alone for both classification and localization tasks. It also outperforms two single head detectors (Figure 1-(a), (b)). This demonstrates that the fully connected head prefers the classification task while the convolution head prefers the localization task. Secondly, we found that our double-head model can be further improved by using the other task for supervision, i.e., adding localization supervision on FC-Head and classification supervision on Conv-Head.

Our double-head detector outperforms FPN baseline by a non-negligible margin. Experimental results on MS COCO dataset demonstrates that our approach gains 3.4 and 2.7 of mAP over FPN baselines with ResNet-50 and ResNet-101 backbones, respectively.

2. Related Work

One-stage object detectors: One-stage methods attract more attention recently, mostly due to the computational efficiency. OverFeat [23] detects objects by sliding multi-scale windows on the shared convolutional feature maps. Recently, SSD [19, 5] and YOLO [20, 21] have been tuned for speed by predicting object classes and locations directly. RetinaNet [17] alleviates the extreme foreground-background class imbalance problem by introducing the focal loss.

Two-stage object detectors: RCNN [8] applies a deep neural network to extract features for proposals generated by selective search [24] and feed them into SVM classifiers. SPPNet [10] speeds up RCNN significantly by introducing a spatial pyramid pooling layer to reuse features computed over feature maps generated at different scales. Fast RCNN [6] utilizes a differentiable RoI Pooling operation to fine-tune all layers end-to-end, and further improves the speed and performance over SPPNet. Later, faster RCNN [22] introduces Region Proposal Network (RPN) into the network. R-FCN [3] employs the position sensitive RoI pooling to address the translation-variance problem in object detection. Feature Pyramid Network (FPN) [16] builds a top-down architecture with lateral connections to utilize high-level semantic feature maps at all scales, which benefits the small object detection more as finer feature maps are utilized. Deformable ConvNet [4] proposes deformable convolution and deformable RoI pooling to augment the spatial sampling locations. Cascade RCNN [2] constructs a sequence of detectors trained with increasing intersection over union (IoU) threshold, which improves the object detection progressively. IoU-Net [13] introduces another standalone branch to predict the IoU between each detected bounding box and the matched ground-truth, which generates localization confidence to replace the classification confidence for non-maximum suppression (NMS).

Next, we further compare backbone networks and detection heads for two-state detectors.

Backbone Networks: Fast RCNN [6] and Faster RCNN [22] extract features in stage conv4, while FPN [16] utilizes features from multiple layers (conv2 to conv5). Deformable-v1 [4] applies deformable convolution at the last few convolution layers and Deformable-v2 [26] adds more deformable convolution at all 3 × 3 convolution layers in stages conv3, conv4, and conv5. Trident Network [14] generates scale-aware feature maps with multi-branch architecture.

Detection Heads: Light-Head RCNN [15] utilizes thin feature maps and a cheap subnet in detection heads to reduce the computational cost in detection heads. Cascade RCNN [2] builds multiple detection heads in a cascade manner. Mask RCNN [9] introduces an extra head for object segmentation. IoU-Net [13] proposes an extra head to predict the IoU score of each proposal. Similar to IoU Net, Mask scoring RCNN [12] presents an extra head to predict the MaskIoU score of each generated segmentation mask. In contrast to the existing detection heads which share the same RoI feature extractor for both classification and bounding box regression, we propose to split these two tasks into different heads to leverage the power of both the fully connected head and convolution head.

3. Hypothesis on Detection Heads

Motivation: We are motivated by the method that COCO Detection 18 Challenge winner (Megvii) proposed for instance segmentation. It integrates the regression and segmentation tasks in a shared convolution head, and leave the classification task in another fully connected head. This motivates us to rethink the classification and box regression tasks for object detection with respect to different head structures (e.g. convolution or fully connected).

Intuitively, we believe that the convolution head is more suitable for bounding box regression, even without the help from the segmentation. This is because the convolution head is able to capture the context for the whole object, which is crucial to refine the bounding box from the proposal. In contrast, the fully connected head, which is spatial sensitive, fits well for the classification task in object detection. Different from image classification, object classification need to consider if the object (if there is) is complete within the proposed bounding box. This requires both context and spatial information.

Hypothesis: Given the motivation discussed above, we have a hypothesis as follows:

_The fully connected head is more suitable for the_
classification task, while the convolution head is more suitable for the localization task.

Validation: To validate this hypothesis, we design a new detector with double heads: a fully connected head and a convolution head (see Figure 1-(c)), on a shared backbone network. Both heads are supervised by both classification and box regression, and are trained jointly. Thus, we have two classification outputs and two box regression outputs (from two different heads). As a result, we have four different combinations of classification and box regression outputs, whose corresponding detection performances on MS COCO minival are shown in Table 1.

The best combination is to use the fully connected head (FC-Head) for classification and use the convolution head (Conv-Head) for localization. It outperforms other combinations by 1.4+ mAP. It also outperforms the detector with a single fully connected head (Figure 1-(a)) and the detector with a single convolution head (Figure 1-(b)). If we reverse the task selection by using FC-Head for localization and using Conv-Head for classification, the performance drops substantially (mAP drops 3.4 points). This validates our hypothesis that FC-Head is more suitable for classification, while Conv-Head is more suitable for localization.

Furthermore, we remove the regression supervision from FC-Head and the classification supervision from Conv-Head. Thus, FC-Head is trained for classification alone and Conv-Head is trained for localization alone. The performance is further improved from 38.5 to 39.4 mAP (see Table 2). This motivates us to systematically investigate the double-head design for two-stage detectors. In the next section, we will discuss our double-head method in details.

4. Our Method: Double-Head Detector

Our double-head method (see Figure 1-(c)) has a fully connected head (referred to FC-Head) for the classification task and a convolution head (referred to Conv-Head) for the localization task. Although each head has its own target task during inference, both heads could be supervised by the ground truth of both classification and bounding box during training. We will introduce the network architecture, multi-task joint training, instantiantis and double-head inference in this section.

4.1. Network Architecture

We use FPN [16] backbone to generate region proposals and extract object features from multiple levels using RoI align [9]. Each proposal has a feature map with size $7 \times 7 \times 256$ and is sent to both FC-Head and Conv-Head. Both heads transform the feature map ($7 \times 7 \times 256$) into a feature vector (1024 dimensions), which is used for classification and bounding box regression (see Figure 1-(c)).

Fully Connected Head (FC-Head) has two fully connected layers (see Figure 1-(c)), following the design in FPN [16] (Figure 1-(a)). The output dimension is 1024.

Convolution Head (Conv-Head) first increases the number of channels from 256 to 1024 by using a residual block (shown in Figure 2). Then it passes $K$ residual bottleneck blocks [11] (shown in Figure 3). At the end, average pooling is used to generate the feature vector with dimension 1024, which is shared for both classification and bounding box regression.

We also introduce a variation of the convolution head by inserting a non-local block before each residual block to enhance foreground. We refer this to Conv-NL-Head. Figure 4 shows the diagram of a non-local block. Note that the number of parameters for the residual block and the non-local block are close (1.06M vs 1.00M).

4.2. Multi-Task Joint Training

Both the fully connected head and the convolution head are jointly trained with region proposal network (RPN) end to end. The overall loss is computed as follows:

$$
\mathcal{L} = \mathcal{L}^{fc} + \mathcal{L}^{conv} + \mathcal{L}^{rpn},
$$

where $\mathcal{L}^{fc}$, $\mathcal{L}^{conv}$, $\mathcal{L}^{rpn}$ are the losses for FC-head, Conv-head and region proposal network (RPN), respectively. The loss for FC-head ($\mathcal{L}^{fc}$) is computed as follows:

$$
\mathcal{L}^{fc} = \lambda^{fc} L_{cls}^{fc} + (1 - \lambda^{fc}) L_{reg}^{fc},
$$
Table 1. Comparison among four combinations of task selection (classification or box regression) from two detection heads (FC or Conv). The Evaluations are performed on COCO minival. The backbone network is FPN on ResNet-50. Using the fully connected (FC) head for classification and the convolution (Conv) head for box regression outperforms other combinations.

| FC Head | Conv Head | AP | AP$_{0.5}$ | AP$_{0.75}$ | AP$_s$ | AP$_m$ | AP$_l$ |
|---------|-----------|----|-----------|-----------|--------|--------|--------|
| ✓       | ✓         | 36.7 | 55.6 | 39.8 | 19.9 | 40.5 | 50.2 |
| ✓       | ✓         | 37.1 | 58.8 | 39.9 | 20.8 | 40.6 | 48.8 |
| ✓       | ✓         | 35.1 | 55.5 | 37.9 | 19.4 | 38.9 | 47.8 |
| ✓       | ✓         | 38.5 | 58.9 | 41.7 | 21.6 | 41.7 | 50.8 |
| single Conv Head | | 35.9 | 54.0 | 39.6 | 19.1 | 39.4 | 50.2 |
| single FC Head | | 36.8 | 58.7 | 40.4 | 21.2 | 40.1 | 48.8 |

Table 2. Comparison between multi-task supervision per head and single task supervision per head. ✓ indicates that the corresponding head is supervised by the corresponding task. ✓ indicates that the corresponding output is used for the detection result. The evaluations are performed on COCO minival, using FPN on ResNet-50 as backbone.

| FC Head | Conv Head | AP | AP$_{0.5}$ | AP$_{0.75}$ | AP$_s$ | AP$_m$ | AP$_l$ |
|---------|-----------|----|-----------|-----------|--------|--------|--------|
| ✓ ✓ ✓ ✓ | ✓ ✓ ✓ ✓ | 38.5 | 58.9 | 41.7 | 21.6 | 41.7 | 50.8 |
| ✓ ✓ ✓ ✓ | ✓ ✓ ✓ ✓ | 39.4 | 59.7 | 42.8 | 22.5 | 42.3 | 52.3 |

4.3. Instantiations

Next, we describe three cases with different weights ($\lambda^{fc}$ and $\lambda^{conv}$).

Balanced ($\lambda^{fc} = \lambda^{conv} = 0.5$): The classification and localization are equally important during the training for both heads. The first experiment in the hypothesis validation (the first row in Table 2) is a balanced case.

Extreme ($\lambda^{fc} = 1$, $\lambda^{conv} = 0$): Both heads are supervised by their target task alone (i.e. FC-Head is supervised by classification alone, and Conv-Head is supervised by box regression alone), without the help from the other task. The second experiment in the hypothesis validation (the second row in Table 2) is belong to this case.

Optimal: $\lambda^{fc}$ and $\lambda^{conv}$ are obtained by grid search (with minimum mAP). The ablation study for the above three cases, which will be shown in section 5.3, is helpful to analyze our double-head design.

4.4. Double-Head Inference

The inference uses the classification result from the fully connected head and uses the regression result from the convolution head. We also propose a complementary fusion of the classification scores from both heads as follows:

$$s = s^{fc} + s^{conv}(1 - s^{fc})$$

(4)

where $s^{fc}$ and $s^{conv}$ are classification scores from FC-Head and Conv-Head, respectively. We believe that as the two heads learn the object classification in different ways,
they may capture complementary information for classification. Therefore, we design a score increment as a product of the second score and the reverse of the first score (e.g. \( s^{conv}(1 - s^{fc}) \)). This is different from [2] which combining all cascade classifiers by average. Note that the fusion of classification scores is only applicable when \( \lambda^{fc} \neq 0 \) and \( \lambda^{conv} \neq 0 \).

5. Experimental Results

The proposed double-head detector (referred as double-head) was evaluated on MS COCO dataset[18]. We compare our method to the original FPN [16] with single 2-fc head (referred as FPN baseline). We also perform the ablation study to analyze different components of our approach.

5.1. Datasets

All experiments are done on MS COCO dataset [18], which has 80 object categories. Following [1, 16], the union of COCO 2014 train with 80K images and COCO 2014 val35k with 35K images is used as the training set. The evaluation is on COCO 2014 minival with 5K images. Object detection accuracy is measured by the standard COCO-style Average Precision (AP) with different IoU thresholds from 0.5 to 0.95 with an interval of 0.05.

5.2. Implementation Details

Our implementation is based on the Mask R-CNN benchmark in Pytorch 1.0.\(^2\) The implementation details are as follows:

**Architecture:** Our double-head design is evaluated on two FPN [16] backbones (ResNet-50 and ResNet-101 [11]). The standard RoI pooling is replaced by RoIAlign [9]. FC-Head is the same with the FPN [16] baseline (2-fc) with 1024 dimensions. Conv-Head has one residual block (see Figure 2) to increase the number of channels from 256 to 1024 and \( K \) convolution blocks (see Figure 3). The variation of Conv-Head with non-local block [25] (i.e. Conv-NL-Head) has 512 inter-channels within each non-local block. For both heads, the class-specific bounding box regression is used.

**Hyper-parameters:** All models are trained in 4 GPUs, with a mini-batch size of 2 images per GPU. The weight decay is 0.0001 and momentum is 0.9.

**Learning rate scheduling:** All models were fine-tuned with 180k iterations. The learning rate is initialized with 0.01 and reduced to 0.001 after 120K iterations and 0.0001 after 160K iterations.

---

We use no data augmentation for testing, and only horizontal flipping augmentation for training. Images are resized such that the shortest side is 800 pixels.

5.3. Ablation Study

We run a number of ablations on MS COCO dataset [18] to analyze our double-head detectors. ResNet-50 backbone is used for all ablation experiments. By default, Conv-Head has \( K = 4 \) convolution blocks, while its variation Conv-NL-Head has 2 convolution blocks and 2 non-local blocks, which has slightly less parameters than 4 convolution blocks.

**Balanced vs Extreme vs Optimal:** Three double-head instantiations on two convolution head variations (Conv-Head and Conv-NL-Head) are shown in Table 3. Both head variations share the same trend among these instantiations. The extreme is better than the balanced (about 1.0 AP), and is close to the optimal. The optimal instantiations demonstrate that (a) the classification supervision on the convolution head is not necessary (small \( \lambda^{conv} \)), and (b) the localization supervision on FC-Head is helpful, but not significant. When using Conv-Head, the optimal weights are \( \lambda^{fc} = 0.8 \) and \( \lambda^{conv} = 0.1 \). When using Conv-NL-Head, the optimal weights are \( \lambda^{fc} = 0.7 \) and \( \lambda^{conv} = 0.0 \).

**Classification Weight on FC-Head \( \lambda^{fc} \):** Table 4 shows double-head with various classification weights on FC-Head (\( \lambda^{fc} \)). Here we use Conv-NL-Head for the convolution head, with optimal weights \( \lambda^{conv} = 0.0 \). The performances are stable (39.3 to 39.9 AP), and the peak is at \( \lambda^{fc} = 0.7 \), demonstrating that box regression on FC-Head introduces incremental improvements.

**Classification Weight on the Convolution Head \( \lambda^{conv} \):** Table 5 shows double-head with various classification weights on the convolution head (\( \lambda^{conv} \)). Here we use Conv-NL-Head for the convolution head, with optimal weights \( \lambda^{conv} = 0.7 \). Clearly, the performance declines as \( \lambda^{conv} \) increases. This suggests that classification supervision is not necessary for the convolution head.

**Structure of the Convolution Head:** We study the choice of the number of blocks in the convolution head. Both variations (Conv-Head and Conv-NL-Head) are studied. Conv-NL-Head has \( K/2 \) residual blocks (see Figure 3) and \( K/2 \) non-local blocks (see Figure 4). The evaluations are shown in Table 6. As the number of blocks increases, the performance improves gradually with decreasing rate. Considering the trade-off between accuracy and speed and the fair comparison between Conv-Head and Conv-NL-Head, we choose \( K = 4 \) for the rest of the paper.

**Fusion of Classification Scores from Both Heads:** We study four different ways to fuse the classification scores from both fully connected head (\( s^{fc} \)) and convolution head (\( s^{conv} \)): (a) using \( s^{fc} \) alone, (b) average, (c) maximum, and (d) complementary fusion using Eq. 4. We also study two

---

\(^2\)https://github.com/facebookresearch/maskrcnn-benchmark
Table 3. Comparison among balanced, extreme and optimal instantiations. When using Conv-Head, the optimal weights for the classification task are $\lambda^{fc} = 0.8$ and $\lambda^{conv} = 0.1$. When using Conv-NL-Head, the optimal weights are $\lambda^{fc} = 0.7$ and $\lambda^{conv} = 0.0$. Using attention (Conv-NL-Head) brings a small improvement. For both Conv-Head and Conv-NL-Head, the extreme instantiation is better than the balanced and is close to the optimal.

Table 4. The classification weight on FC-Head ($\lambda^{fc}$). Conv-NL-Head is used with 2 residual blocks and 2 non-local blocks. The classification weight on Conv-NL-Head is set $\lambda^{conv} = 0$.

Table 5. The classification weight on the convolution head ($\lambda^{conv}$). Conv-NL-Head is used with 2 residual blocks and 2 non-local blocks. The classification weight on FC-Head is set $\lambda^{fc} = 0.7$.

Table 6. The number of blocks (see Figure 3 and 4) in the convolution head. The first group is corresponding for Conv-Head, the second group is for Conv-NL-Head, which has $K/2$ residual blocks and $K/2$ non-local blocks.

The balanced and is close to the optimal.

5.4. Comparison with FPN Baselines

In this section, we compare our double-head detectors with FPN [16] and Faster RCNN [22]. The comparison is conducted on two backbones (ResNet-50 and ResNet-101). Our double-head detectors includes two variations of the convolution head (Conv-Head and Conv-NL-Head). Conv-Head has four residual blocks, while Conv-NL-Head has two residual blocks and two non-local blocks. When using Conv-Head, the classification weights for both heads are set as $\lambda^{fc} = 0.8$ and $\lambda^{conv} = 0.1$. When using Conv-NL-Head, the classification weights for both heads are set as $\lambda^{fc} = 0.7$ and $\lambda^{conv} = 0.2$. During inference, the complementary fusion (Eq. 4) is used.

Table 8 shows the evaluations. Our method outperforms both FPN baseline and Faster RCNN on all evaluation metrics. Compared to FPN baselines, our method with Conv-NL-Head gains 3.4 and 2.7 of AP on ResNet-50 and ResNet-101 backbones, respectively. Our method gains 3.4+ AP on the higher IoU threshold (0.75) and 1.4+ AP on the lower IoU threshold (0.5). This demonstrates that our double-head method has better capability than the single head methods.

We also observe that when using ResNet-101 backbone, Faster R-CNN has better AP on medium and large objects than FPN. Even comparing with the best performance of both FPN and Faster R-CNN across different scales, our method gains 1.9 AP on small objects, 2.1 AP on medium objects and 2.6 AP on large objects. This demonstrates the
successor detects these objects too. FC-Head our double-head method uses proposals, allowing them to be successfully detected. Since FC-Head In contrast, moved by the post-processing, resulting in miss detection. Since the probabilities from Conv-Head shown in the top row (the regions in the boxes are zoomed in). The classification probabilities of these small objects from both heads are lower than Conv-Head alone (in the green box) and our double-head, are missed when using Conv-Head alone (in the red box). This is due to the weak classification from Conv-Head. The classification probabilities of these small objects from both heads are shown in the top row (the regions in the boxes are zoomed in). Since the probabilities from Conv-Head are lower than the threshold in post-processing, these proposals are removed by the post-processing, resulting in miss detection. In contrast, FC-Head produces high probabilities for these proposals, allowing them to be successful detected. Since our double-head method uses FC-Head for classification, it successfully detects these objects too.

Figure 6 demonstrates the advantage of Conv-Head has a single proper detection for the baseball bat, FC-Head has duplicate detections. This is due to the weak regression from FC-Head. Both heads share the same proposals (yellow boxes). For the biggest proposal, both heads have inaccurate regression results (the biggest box for each head). The biggest box from FC-Head has lower Intersection over Union (IoU) with other boxes around the baseball bat, resulting in a duplicate detection. Meanwhile, the biggest box from Conv-Head, which is more accurate than the one from FC-Head, is removed by the post-processing since it has higher IoU with other boxes. Leveraging the advantage of Conv-Head for bounding box regression, our double-head method does not have duplication either.

### 6. Conclusions

In this paper, we validated our hypothesis in object detection that the fully connected head is more suitable for the classification task, while the convolution head is more suitable for the localization task. Based upon this finding, we proposed to separate object classification and bounding box regression into different heads, i.e., a fully connected head for classification and a convolution head for regression. Our method gains 3.4 and 2.7 of mAP over FPN with ResNet-50 and ResNet-101 backbones respectively on MS COCO dataset. We believe that it can be helpful for the future research in object detection.

---

**Table 7.** Fusion of classification scores from both heads. Complementary fusion (Eq. 4) outperforms others for both Conv-Head and Conv-NL-Head.

| Fusion Method | Score | Conv Head | AP  | AP0.5 | AP0.75 | APS | AM | AP
|---------------|-------|-----------|-----|-------|--------|-----|----|----
| FC alone      | sfc   | Conv      | 39.5 | 59.5  | 43.3   | 22.5 | 42.4 | 52.0 |
| Average       | (sfc + scon)v/2 | Conv      | 39.2 | 59.1  | 42.9   | 22.4 | 42.4 | 51.4 |
| Max           | max(sfc, scon)   | Conv      | 39.5 | 59.4  | 43.2   | 22.5 | 42.5 | 52.1 |
| Complementary | sfc + scon(1 − sfc) | Conv      | 39.6 | 59.7  | 43.3   | 22.7 | 42.8 | 52.0 |
| FC alone      | sfc   | Conv-NL   | 39.7 | 59.6  | 43.5   | 22.3 | 42.9 | 52.5 |
| Average       | (sfc + scon)v/2 | Conv-NL   | 39.8 | 59.7  | 43.9   | 22.3 | 42.8 | 53.0 |
| Max           | max(sfc, scon)   | Conv-NL   | 39.9 | 59.7  | 44.0   | 22.6 | 42.9 | 52.9 |
| Complementary | sfc + scon(1 − sfc) | Conv-NL   | 40.2 | 60.2  | 44.2   | 22.9 | 43.1 | 53.3 |

**Table 8.** Comparisons with baseline detectors (FPN and Faster RCNN) on MS COCO val2014. Our double-head method outperforms both FPN baseline and Faster RCNN on two backbones (ResNet-50 and ResNet-101).

| Method          | Backbone         | AP   | AP0.5 | AP0.75 | APS   | AM   | AP   |
|-----------------|------------------|------|-------|--------|-------|------|------|
| Faster R-CNN [22] | ResNet-50-C4    | 34.8 | 55.8  | 37.0   | 19.1  | 38.8 | 48.2 |
| FPN baseline [16]   | ResNet-50       | 36.8 | 58.7  | 40.4   | 21.2  | 40.1 | 48.8 |
| Double-head (Conv) | ResNet-50       | 39.6 | 59.3  | 43.3   | 22.7  | 42.8 | 52.0 |
| Double-head (Conv-NL)| ResNet-50   | 40.2 | 60.2  | 44.2   | 22.9  | 43.1 | 53.3 |
| Faster R-CNN [22] | ResNet-101-C4  | 38.5 | 59.4  | 41.4   | 19.7  | 43.1 | 53.3 |
| FPN baseline [16]   | ResNet-101      | 39.1 | 61.0  | 42.4   | 22.2  | 42.5 | 51.0 |
| Double-head (Conv) | ResNet-101      | 41.2 | 61.9  | 44.9   | 23.7  | 44.7 | 54.8 |
| Double-head (Conv-NL)| ResNet-101  | 41.8 | 62.4  | 45.8   | 24.1  | 45.2 | 55.9 |

superiority of our double-head method, which leverages the advantage of the fully connected head on classification and the advantage of the convolution head on localization.

### 5.5. Qualitative Analysis

In this section, we compare FC-Head with Conv-Head qualitatively. We apply a well-trained double-head model (Figure 1-(c)) and compare the detection results of (a) using Conv-Head alone, (b) using FC-Head alone, and (c) using both heads.

Figure 5 illustrates a case that FC-Head has better classification ability than Conv-Head. It shows that several small objects, which are successfully detected by using FC-Head alone (in the green box) and our double-head, are missed when using Conv-Head alone (in the red box). This is due to the weak classification from Conv-Head. The classification probabilities of these small objects from both heads are shown in the top row (the regions in the boxes are zoomed in). Since the probabilities from Conv-Head are lower than the threshold in post-processing, these proposals are removed by the post-processing, resulting in miss detection. In contrast, FC-Head produces high probabilities for these proposals, allowing them to be successfully detected. Since our double-head method uses FC-Head for classification, it successfully detects these objects too.

Figure 6 demonstrates the advantage of Conv-Head in bounding box regression. Compared with Conv-Head has a
Figure 5. **FC-Head** is more suitable for classification than **Conv-Head**. Bottom row (from left to right): ground truth (GT), detection results using **Conv-Head** alone, detection results using **FC-Head** alone, detection results using our double-head method. Top row (from left to right): classification probabilities from **Conv-Head** for proposals in the red box, classification probabilities from **FC-Head** for proposals in the green box. The proposals are drawn as yellow boxes. The missing small objects (in the red box) are due to the low classification probabilities from **Conv-Head**. These objects are successfully detected (in the green box) by **FC-Head** with high classification probabilities.

**Conv-Head** has better regression for the inaccurate proposal.

Figure 6. **Conv-Head** is more suitable for regression than **FC-Head**. Bottom row (from left to right): ground truth (GT), detection results using **Conv-Head** alone, detection results using **FC-Head** alone, detection results using our double-head method. Top row (from left to right): proposals (yellow boxes) for the baseball bat, regressed boxes from **Conv-Head**, regressed boxes from **FC-Head**. **FC-Head** has a duplicate detection for the baseball bat (in red box), which is generated from an inaccurate proposal. As this detection has low Intersection over Union (IoU) with other more accurate detection boxes, it can not be removed by the non maximum suppression (NMS) in post-processing. However, for the same inaccurate proposal, **Conv-Head** generates a better regressed box, which has higher IoU with other detection boxes, and is successfully removed by the NMS. As a result, **Conv-Head** does not have duplication around the baseball bat.
References

[1] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2874–2883, 2016. 5

[2] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018. 2, 5

[3] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In Advances in Neural Information Processing Systems 29, pages 379–387. 2016. 1, 2

[4] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, pages 764–773, 2017. 2

[5] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659, 2017. 2

[6] R. Girshick. Fast R-CNN. In Proceedings of the International Conference on Computer Vision (ICCV), 2015. 1, 2

[7] 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 (CVPR), 2014. 1

[8] 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, pages 580–587, 2014. 2

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In Computer Vision (ICCV), 2017 IEEE International Conference on, pages 2980–2988. IEEE, 2017. 2, 3, 5

[10] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In European conference on computer vision, pages 346–361. Springer, 2014. 2

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 3, 5

[12] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang. Mask Scoring R-CNN. In CVPR, 2019. 2

[13] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang. Acquisition of localization confidence for accurate object detection. In Proceedings of the European Conference on Computer Vision (ECCV), pages 784–799, 2018. 2

[14] Y. Li, Y. Chen, N. Wang, and Z. Zhang. Scale-aware trident networks for object detection. arXiv preprint arXiv:1901.01892, 2019. 2

[15] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Light-head r-cnn: In defense of two-stage object detector. arXiv preprint arXiv:1711.07264, 2017. 2

[16] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 936–944, July 2017. 1, 2, 3, 5, 6, 7

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. IEEE transactions on pattern analysis and machine intelligence, 2018. 2

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014. 5

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016. 2

[20] 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, pages 779–788, 2016. 2

[21] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. arXiv preprint, 2017. 2

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015. 1, 2, 6, 7

[23] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229, 2013. 2

[24] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. International journal of computer vision, 104(2):154–171, 2013. 2

[25] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7794–7803, 2018. 5

[26] X. Zhu, H. Hu, S. Lin, and J. Dai. Deformable convnets v2: More deformable, better results. arXiv preprint arXiv:1811.11168, 2018. 2
A. More Visualization Results

In this appendix, we compare the fully connected head (FC-Head) with the convolution head (Conv-Head) qualitatively. We apply a well trained double-head model (Figure 1-(c)) and compare the detection results of (a) using Conv-Head alone, (b) using FC-Head alone, and (c) using both heads.

In Figure 7, these results show that FC-Head is good at handling small objects, which Conv-Head misses the detections. For example, in the first row, some people are missed by Conv-Head on the right side of the seaside, but are recovered by the FC-Head (third column) successfully. Our final results (last column) also recover these objects. This also happens for the small objects between the two persons in the second row, several small persons in the third row and the object between the person and the clock in the last row. As FC-Head is more suitable for classification, especially for small objects, it is able to detect these small objects well. By leveraging the classification power of FC-Head, our double-head detector also has the ability to detect these small objects successfully. Similar findings can be observed in other rows. Some corresponding intermediate results are shown in Figure 9 to compare the probabilities of Conv-Head and FC-Head.

Figure 8 demonstrates the advantage of Conv-Head. We can see FC-Head has duplicated detections (baseball bat in the first row and the surfboard in the second row). Intermediate results of the surfboard case are analyzed in Figure 10.

In Figure 11, we show some failure cases. In the first row, FC-Head makes a lot predictions on small carrots, which results that our method also predict a lot small carrots. Meanwhile, the convolution head focuses on the whole plate, which aligns with the ground truth annotations. Moreover, Conv-Head finds a box on the top left corner, which is missed by FC-Head and is not in the annotations as well. In the second row, the FC-Head predicts some false alarms below the standing human, which causes that our method also raises similar false alarms. But given the regression results from Conv-Head, false alarms from our method are trying to locate the chairs, compared with some nonsense results from FC-Head. Although given these failure cases, it demonstrates that our method inherit the merits from both Conv-Head and FC-Head.
Figure 7. Comparison between detection results with ground truth, results from Conv-Head, results from the FC-Head and results from our method: First column: ground truth. Second column: results from the Conv-Head. Third column: results from FC-Head. Last column: results from our method with Conv-Head from classification and FC-Head for regression. These results shows that the FC-Head is good at handling small objects which Conv-Head misses some objects.

Figure 8. Comparison between detection results with ground truth, results from Conv-Head, results from FC-Head and results from our method: First column: ground truth. Second column: results from Conv-Head. Third column: results from FC-Head. Last column: results from our method with Conv-Head from classification and FC-Head for regression. These results shows that the Conv-Head is good at handling large objects meanwhile FC-Head makes duplicate predictions.
Figure 9. **FC-Head** head is more suitable for classification than **Conv-Head**. Bottom row (from left to right): ground truth (GT), detection results using **Conv-Head** alone, detection results using **FC-Head** alone, detection results using our double-head method. Top row (from left to right): classification probabilities from **Conv-Head** for proposals in the red box, classification probabilities from **FC-Head** for proposals in the green box. The proposals are drawn as yellow boxes. The missing small objects (in the red box) are due to the low classification probabilities from CONV head. These objects are successfully detected (in the green box) by **FC-Head** with high classification probabilities.
**Conv-Head** has better regression for the inaccurate proposal.

Figure 10. **FC-Head** is more suitable for classification than **Conv-Head**. Bottom row (from left to right): ground truth (GT), detection results using **Conv-Head** alone, detection results using **FC-Head** alone, detection results using our double-head method. Top row (from left to right): classification probabilities from **Conv-Head** for proposals in the red box, classification probabilities from **FC-Head** for proposals in the green box. The proposals are drawn as yellow boxes. The missing small objects (in the red box) are due to the low classification probabilities from **Conv-Head**. These objects are successfully detected (in the green box) by **FC-Head** with high classification probabilities.

Figure 11. Comparison between detection results with ground truth, results from **Conv-Head**, results from the **FC-Head** and results from our method: First column: ground truth. Second column: results from **Conv-Head**. Third column: results from **FC-Head**. Last column: results from our method with **Conv-Head** from classification and **FC-Head** for regression. These results show some failure cases that our method inherits the classification ability from **FC-Head** and sometimes generates false alarms.