G-HEAD: GATING HEAD FOR MULTI-TASK LEARNING IN ONE-STAGE OBJECT DETECTION

He Jiang[1,2], *Qingyi Gu*[1,*]

[1]Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
{jianghe2019, qingyi.gu}@ia.ac.cn

ABSTRACT

Object detection is commonly formulated as a multi-task learning problem in deep learning methods. Due to the divergence between classification and regression tasks, modern one-stage detectors typically utilize two parallel branches as the detection head, which might be sub-optimal. In this paper, we propose a new Gating Head (G-Head) to enhance the interaction between different tasks and promote the multi-task learning process. By introducing Multi-Scale Aggregation (MSA), Multi-Aspect Learning (MAL), and Gating Selector (GS), our method can significantly boost the performance of existing one-stage frameworks with fewer parameters and computational costs. To validate the efficiency, effectiveness, and generalization of our G-Head, extensive experiments are conducted on the challenging MS COCO dataset. Without bells and whistles, we achieve a new state-of-the-art 48.7 AP under single-model and single-scale test.

Index Terms— Object detection, multi-task learning, detection head, gating mechanism

1. INTRODUCTION

Object detection is one of the fundamental problems in computer vision and serves as the basis for downstream tasks such as instance segmentation and panoptic segmentation. Generally, object detection is formulated as a multi-task learning problem [1–10] and its target is to tell what and where objects are for a given image. The classification task aims to focus on the salient parts of objects while the regression task is more sensitive to the outlines. Due to the misalignment between these two tasks, current one-stage detectors [4, 5, 7, 10] typically use a detection head with two parallel branches to optimize the multi-task learning problem. Recent works [8, 9, 11] try to further boost the performance by introducing extra auxiliary tasks. For instance, FCOS [8] and ATSS [9] define centerness to enhance the predictions made by the center of an object. IoU-aware RetinaNet [11] estimates the IoUs between regression results and ground truths to improve localization accuracy. Since the parameters and the computational costs of a parallel head can grow linearly as the number of branches increases, extra predictions are performed on the existing branches as illustrated in Fig 1. Although the aforementioned methods have achieved substantial success, we argue that the parallel head structure might limit the upper bound of detection performance for the following two reasons:

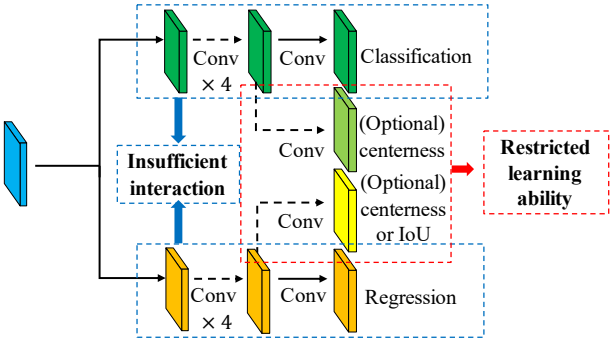


Fig. 1. Illustration of the parallel head structure. It suffers from two major limitations: (1) **Insufficient interaction between classification and regression tasks.** (2) **Restricted learning ability of independent branch.**

Insufficient interaction between classification and regression tasks. Utilizing separate branches to perform multi-task learning is a natural solution to deal with the divergence between different tasks. This heuristic method encourages each branch to pay attention to the specific details for its corresponding task. However, as shown in Fig 1, the two individual branches fail to interact with each other in the learning process, which could lead to inferior performance. In object detection, classification and regression tasks are closely related and they also share common knowledge. For example, knowing what kind of category an object is can provide prior knowledge for inferring its shape and boundaries. Conversely, given the location and outlines of an object, the detector can classify it more easily. Therefore, the predictions for classification and regression tasks should be made by combining both shared and specific information.

Restricted learning ability of independent branch. Making predictions for each task in object detection should consider various aspects jointly, for example, the salient parts and

* Corresponding Author

outlines of objects and the contextual information. Using an independent branch might fail to learn all of these aspects due to its restricted learning ability. As shown in Fig 1, recent detectors [8,9,11] further utilize existing branches to learn extra tasks. Despite its simplicity and efficiency, we argue that forcing an independent branch to make predictions for multiple tasks is challenging. It could further increase the burden and difficulty during the learning process. More importantly, there is no valid proof that predicting centerness [8,9] or IoUs [11] shares the same information with the regression task.

To address the issues mentioned above, we propose a new head structure called Gating Head (G-Head), which aims to solve the multi-task learning problem in a more effective and elegant way. The design philosophy of our G-Head is completely different from that of the parallel head. To be specific, our G-Head is composed of three parts. First, Multi-Scale Aggregation (MSA) module is leveraged to enhance the interaction between different tasks by collecting shared information from multi-scale features. Then, Multi-Aspect Learning (MAL) module is utilized to decompose the multi-task learning problem into more fine-grained aspects. Finally, Gating Selector (GS) is used to adaptively collect information from multiple aspects for each task. In conclusion, our contributions can be summarized into three folds:

(1) We analyze the structure of the conventional parallel head and point out why it limits the detection performance.

(2) A new G-Head is proposed to replace the parallel head. With fewer parameters and FLOPs, our G-Head consistently boosts the performance of various one-stage detectors.

(3) Extensive experiments are conducted on MS COCO [12] dataset to verify the effectiveness and generalization of our method. Under single-model and single-scale test, we achieve 48.7 AP without bells and whistles.

2. RELATED WORKS

2.1. One-stage Detectors

With the rapid development of deep learning technology, CNN-based models [1–11] have been leading the area of object detection. Two-stage detectors [1–3] are first proposed to lay the foundation for subsequent methods. Then one-stage detectors [4, 5, 7–9] gradually draw wider attention due to their simplicity and high efficiency. For example, SSD [4] directly makes predictions upon the in-network feature hierarchy and achieves real-time speed. RetinaNet [7] leverages FPN [6] to build a top-down augmentation path and proposes focal loss to address the extreme imbalance problem of classification in dense object detection. YOLOv3 [5] proposes a new backbone called DarkNet-53 and concatenates cross-scale features to perform predictions. To further reduce hand-designed hyper-parameters and save human labor, anchor-free detectors [8, 10] are proposed. With fewer computational costs, their performance surpasses anchor-based models [4, 5, 7] by a large margin. Recently, ATSS [9] systematically analyzes how anchor-free detectors outperform anchor-based methods and introduces a dynamic label assignment strategy to bridge the performance gap between them. The pipeline of the aforementioned one-stage detectors generally follows the backbone-neck-head paradigm. In this paper, we focus on the head structure and aim to provide a new view of the multi-task learning problem in object detection.

2.2. Head Structures

Object detection is typically formulated as a multi-task learning problem. To tackle the difference between multiple tasks, mainstream one-stage detectors [4, 5, 7, 10] generally utilize a parallel head structure to perform multi-task learning, which proves to be useful. Recent works [8, 9, 11] find that introducing extra auxiliary tasks can further promote detection performance. Considering the memory usage and computational costs, they directly attach new tasks to the existing branches. For example, IoU-aware RetinaNet [7] predicts IoUs based on the regression features and multiplies the classification scores by the predicted IoUs. Using this combined criterion to implement NMS algorithm makes two branches work more collaboratively. FCOS [8] and ATSS [9] assume that the central region of an object can produce more accurate predictions. Thus, they introduce the concept of centerness and utilize existing branches to forecast the centerness scores. These previous works [7–11] seldom dive deeply into the head structure and adopt the same parallel design for convenience, which might be sub-optimal for the object detection task. In this work, we propose a new head structure to enhance the interaction between different tasks and introduce a gating mechanism to perform multi-task learning.

3. PROPOSED METHOD

To address the problems mentioned in Section 1, we propose Gating Head (G-Head), which is a substitute for the parallel head in modern one-stage detectors. The overall framework of our G-Head is illustrated in Fig 2. It is mainly composed of three modules, *i.e.*, Multi-Scale Aggregation (MSA), Multi-Aspect Learning (MAL), and Gating Selector (GS). In the following subsections, we will describe them in detail.

3.1. Multi-Scale Aggregation

To enhance the interaction between different tasks, we propose the MSA module to learn shared information by aggregating multi-scale features. As shown in Fig 2 part (a), given the input $\boldsymbol{X}$ from one level of FPN [6], we first leverage N consecutive convolutions to extract features $\boldsymbol{X}_i$ $(i \in 1, 2, \ldots, N)$ with various receptive fields. Then, all of them are stacked together to obtain $\boldsymbol{X}^{stack}$. MSA aims at adaptively aggregating multi-scale features for each level of FPN.

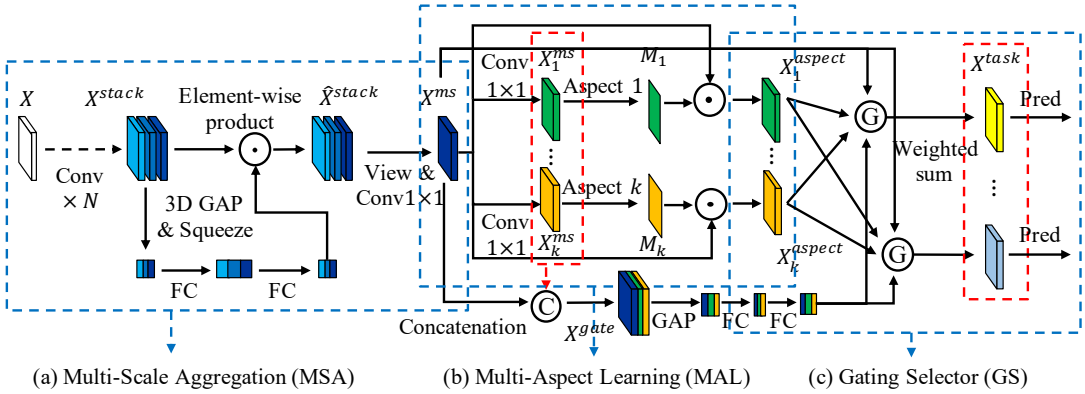


Fig. 2. The overall framework of Gating Head (G-Head).

The procedure can be expressed as

$$\boldsymbol{X}^{ms} = Conv(\boldsymbol{w}^{stack} \cdot \boldsymbol{X}^{stack}) \tag{1}$$

where $\boldsymbol{w}^{stack}$ represents weights w_i^{stack} for each X_i and $Conv$ indicates a 1×1 convolution for channel reduction. We introduce an efficient method to learn the importance of each feature with negligible overheads and it is formulated as

$$\boldsymbol{w}^{stack} = \sigma(fc_2(\delta(fc_1(\frac{1}{HWC}\sum_{H,W,C}\boldsymbol{X}^{stack})))) \tag{2}$$

where δ and σ stand for ReLU and Sigmoid respectively. First, a 3D global average pooling is used to collect global information. Then, we expand the input feature by 4 times using fc_1. Finally, fc_2 is leveraged to generate $\boldsymbol{w}^{stack}$.

3.2. Multi-Aspect Learning

To ease the difficulty of multi-task learning, MAL is proposed to decompose the object detection task into more fragmented aspects. Each branch in MAL is supposed to be aware of a specific aspect such as the salient parts, object boundaries, or contextual information, etc. The details of the MAL module are illustrated in Fig 2 part (b). Given the multi-scale feature $\boldsymbol{X}^{ms}$ from MSA, the purpose of the ith branch is to produce an aspect-wise feature $\boldsymbol{X}_i^{aspect}$, which is denoted as

$$\boldsymbol{X}_i^{aspect} = \boldsymbol{M}_i \odot \boldsymbol{X}^{ms} \tag{3}$$

where $\boldsymbol{M}_i$ represents a specific attention mask. To obtain $\boldsymbol{M}_i$ cheaply, we first utilize a 1×1 convolution to reduce the feature channels. The reduced feature for ith branch is denoted as $\boldsymbol{X}_i^{ms}$. Inspired by CBAM [13], we perform average pooling and max pooling along the channel dimension of $\boldsymbol{X}_i^{ms}$. Then, we concatenate them together and apply a 3×3 deformable convolution [14] to produce the aspect-wise mask $\boldsymbol{M}_i$. Mathematically, the procedure can be written as

$$\boldsymbol{X}_i^{ms} = Conv(\boldsymbol{X}^{ms}) \tag{4}$$

$$\boldsymbol{M}_i = DConv(Cat(AP(\boldsymbol{X}_i^{ms}), MP(\boldsymbol{X}_i^{ms}))) \tag{5}$$

where AP and MP represent average pooling and max pooling respectively, and $DConv$ means deformable convolution.

3.3. Gating Selector

Given features $\boldsymbol{X}_i^{aspect}$ learned from different aspects, we introduce Gating Selector (GS) to automatically combine the required features for each task. As shown in Fig 2 part (c), we concatenate the feature $\boldsymbol{X}^{ms}$ from MSA and the reduced features $\boldsymbol{X}_i^{ms}$ in MAL. In this way, we can obtain $\boldsymbol{X}^{gate}$ containing both shared and specific information. Then, we utilize a SE-like operation [15] to compute the normalized weights for $\boldsymbol{X}^{ms}$ and $\boldsymbol{X}_i^{aspect}$, which is formulated as

$$\boldsymbol{w}^{aspect} = \mathcal{F}(fc_2(\delta(fc_1(\boldsymbol{X}^{gate})))) \tag{6}$$

where δ represents ReLU while $\mathcal{F}$ indicates the softmax operation. For each task, there exists a individual GS to compute unique $\boldsymbol{w}^{aspect}$. As a result, the task-specific feature $\boldsymbol{X}^{task}$ for final prediction is calculated as

$$\boldsymbol{X}^{task} = w_0^{aspect}\boldsymbol{X}^{ms} + \sum_{i=1}^{k} w_i^{aspect}\boldsymbol{X}_i^{aspect} \tag{7}$$

Since $\boldsymbol{X}_i^{aspect}$ is obtained by masked $\boldsymbol{X}^{ms}$, Eq 7 can be further simplified as

$$\boldsymbol{X}^{task} = \sum_{i=0}^{k} w_i \boldsymbol{M}_i \odot \boldsymbol{X}^{ms} \tag{8}$$

where $\boldsymbol{M}_0$ is an identity mask. According to Eq 8, we can calculate the weighted sum of $\boldsymbol{M}_i$ first and multiply $\boldsymbol{X}^{ms}$ by it to save memory and computational costs.

3.4. Difference with spatial/channel attention

The original CBAM module [13] utilizes spatial attention to focus on the semantics parts of objects so the model could classify them more easily. By contrast, our MAL module is designed for object detection task and it produces attention masks from various aspects. In addition, the SE-like operation in GS is responsible for generating weights for different masks, while the SE module [15] in backbone networks is used to calibrate the input feature along channel dimension.

Table 1. Ablation study on the efficiency and effectiveness of MSA with different number of consecutive convolutions (N).

N	Params(M)	GFLOPs	AP	AP_{50}	AP_{75}
Baseline [7]	37.74	250.34	35.7	55.0	38.5
4	35.70	198.50	37.1	58.0	39.6
6	37.01	224.84	38.0	58.4	40.8
8	38.32	251.18	**38.4**	**58.5**	**41.3**

Table 2. Ablation study on the efficiency and effectiveness of MAL with different number of aspects (k).

k	Params(M)	GFLOPs	AP	AP_{50}	AP_{75}
Baseline [7]	37.74	250.34	35.7	55.0	38.5
2	36.99	224.45	37.9	58.3	40.8
4	37.01	224.84	**38.0**	**58.4**	**40.8**
6	37.04	225.23	37.9	58.5	40.6
8	37.06	225.63	37.8	58.1	40.7

4. EXPERIMENTS

We conduct all of our experiments on the MS COCO dataset [12] which contains 80 categories. The MS COCO dataset is split into *train*2017, *val*2017, and *test-dev*2017 with 118K, 5K, 20K images respectively. We follow the standard COCO-style mAP to evaluate our method and all models are trained on the *train*2017 without any extra data. For ablation studies, we report all results on the *val*2017. When comparing with other state-of-the-art methods, we submit predictions on the *test-dev*2017 which has no public labels to the official online server, and use the returned results.

4.1. Implementation Details

We implement all the experiments on the same codebase using MMDetection and PyTorch. All models are trained on 4 RTX A5000 with 4 images per GPU in a mini-batch. The $1\times$ and $2\times$ training schedules follow the standard settings in MMDetection and other training hyper-parameters are kept unchanged. Unless otherwise specified, we set $N = 6$ in MSA and $k = 2m$ in MAL respectively, where m indicates the number of tasks. The reduced channel number for each branch in MAL is set as 32 by default.

4.2. Ablation Studies

For ablation studies, we take RetinaNet [7] with ResNet-50 [16] as the baseline and adopt $1\times$ training schedule. Following the common practices [7–10], we set the input resolution as 1333×800 pixels. The computational costs (FLOPs) are also measured on the input size of 1333×800.

Ablation study on MSA. We validate the efficiency and effectiveness of MSA by varying the number of consecutive convolutions, which is denoted as N. As shown in Table 1, our G-Head consistently outperforms the baseline under different settings. As N grows from 4 to 8, MSA aggregates more multi-scale information and enhances the interaction between different tasks better. Thus, G-Head can continually benefit from the increase of N. Considering the efficiency and accuracy tradeoff, we choose $N = 6$ as the default setting. In this case, G-Head can achieve significant improvement over the baseline while requiring fewer parameters and computational costs (FLOPs).

Table 3. Ablation study on the effectiveness of shared information in GS.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Baseline [7]	35.7	55.0	38.5	18.9	38.9	46.3
w/ shared	**38.0**	**58.4**	**40.8**	**23.2**	**41.3**	**49.0**
w/o shared	37.7	58.1	40.7	22.5	41.6	48.2

Table 4. Generalization on the existing one-stage detectors.

Method	Params(M)	GFLOPs	AP	AP_{50}	AP_{75}
RetinaNet [7]	37.74	250.34	35.7	55.0	38.5
+ G-Head	37.01	224.84	**38.0**	**58.4**	**40.8**
FoveaBox [10]	36.19	215.80	36.4	56.2	38.7
+ G-Head	35.46	190.30	**37.7**	**58.2**	**40.2**
FCOS [8]	32.02	209.85	38.6	57.4	41.4
+ G-Head	31.32	184.65	**39.4**	**57.9**	**42.9**

Ablation study on MAL. We evaluate the efficiency and effectiveness of MAL in Table 2. By decomposing the multi-task learning problem into more aspects, we find that the increase of parameters and FLOPs is almost negligible. However, the detection performance gradually drops as k grows from 4 to 8, which demonstrates that excessively fragmented aspects might hurt the learning process. Suppose that there are m tasks to opimize in total, we aim to make the MAL module adapt to the varying learning difficulty which depends on m. Analyzing how many aspects that a specific task corresponds to is beyond the scope of this work. Thus, we experimentally set $k = 2m$ aspects for different situations.

Ablation study on GS. In Eq 6, we concatenate the multi-scale feature $\boldsymbol{X}^{ms}$ which carries shared information and the aspect-wise features $\boldsymbol{X}_i^{ms}$ to predict weights $\boldsymbol{w}^{aspect}$ jointly. Then, we also incorporate $\boldsymbol{X}^{ms}$ to constitute the task-specific feature $\boldsymbol{X}^{task}$. To investigate the effect of using shared information, we conduct an ablative study on GS and the results are shown in Table 3. Compared with utilizing aspect-wise features alone, introducing shared information demonstrates higher performance, which meets our expectation. The reason is that $\boldsymbol{X}^{ms}$ contains more comprehensive knowledge, making GS produce more rational features for different tasks.

Generalization on existing frameworks. G-Head serves as a plug-and-play module and it can be easily integrated into existing one-stage detectors, such as RetinaNet [7], FoveaBox [10], and FCOS [8]. As shown in Table 4, G-Head consistently surpasses the conventional parallel head by 0.8 AP to 2.3 AP. In the meanwhile, G-Head also allocates fewer parameters and FLOPs. The experimental results in Table 4 indicate that our method is compatible with a wide variety of frameworks (both anchor-based and anchor-free models), which demonstrates its strong generalization ability.

Table 5. Comparisons with other state-of-the-art methods on the MS COCO *test-dev*2017 split.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
state-of-the-art							
RetinaNet [7]	ResNet-101	39.1	59.1	42.3	21.8	42.7	50.2
FoveaBox-align [10]	ResNet-101	42.1	62.7	45.5	25.2	46.6	54.5
FSAF [17]	ResNeXt-64x4d-101	44.6	65.2	48.6	29.7	47.1	54.6
FreeAnchor [18]	ResNeXt-64x4d-101	46.0	65.6	49.8	27.8	49.5	57.7
FCOS w/ imprv [19]	ResNeXt-32x8d-101-DCN	46.6	65.9	50.8	28.6	49.1	58.6
SAPD [20]	ResNeXt-64x4d-101-DCN	47.4	67.4	51.1	28.1	50.3	61.5
BorderDet [21]	ResNeXt-64x4d-101-DCN	48.0	67.1	52.1	29.4	50.7	60.5
GFL [22]	ResNeXt-32x4d-101-DCN	48.2	67.4	52.6	29.2	51.7	60.2
baseline							
ATSS [9]	ResNet-101	43.6	62.1	47.4	26.1	47.0	53.6
ATSS [9]	ResNeXt-32x8d-101	45.1	63.9	49.1	27.9	48.2	54.6
ATSS [9]	ResNeXt-64x4d-101	45.6	64.6	49.7	28.5	48.9	55.6
ATSS [9]	ResNet-101-DCN	46.3	64.7	50.4	27.7	49.8	58.4
ATSS [9]	ResNeXt-32x8d-101-DCN	47.7	66.6	52.1	29.3	50.8	59.7
ATSS [9]	ResNeXt-64x4d-101-DCN	47.7	66.5	51.9	29.7	50.8	59.4
ours							
G-Head	ResNet-50	43.2	61.3	47.2	27.0	46.3	52.6
G-Head	ResNet-101	45.1	63.3	49.3	28.0	48.5	55.2
G-Head	ResNeXt-32x4d-101	46.2	64.6	50.6	29.5	49.4	56.4
G-Head	ResNet-101-DCN	47.6	66.3	52.1	29.2	51.1	59.5
G-Head	ResNeXt-32x4d-101-DCN	48.7	67.7	53.2	30.6	52.0	60.5

4.3. Main Results

For comparisons with other state-of-the-art methods, we take ATSS [9] as our baseline and plug our G-Head into the framework. We train our models with $2\times$ schedule and scale jitter. In the test time, we do not incorporate any augmentations such as model ensemble or multi-scale test and report the results with single model and single scale only.

Compatible with different backbones. We apply our G-Head to different backbones such as ResNet-50 [16], ResNet-101, and ResNeXt-101 [23] to verify its compability. As shown in Table 5, our method consistently outperforms the baseline by a large margin. Under the similar configuration, G-Head can boost the performance by 1.5 AP with ResNet-101. Taking a lighter ResNeXt-32x4d-101 as backbone, our G-Head surpasses the baseline with ResNeXt-64x4d-101 by 0.6 AP, which is considerable for the MS COCO dataset.

Compared with state-of-the-art detectors. We comprehensively compare our method with other state-of-the-art detectors [7, 10, 17–22] and the results are shown in Table 5. Without bells and whistles, our best model with backbone ResNeXt-32x4d-101-DCN [14] achieves competitive 48.7 AP, which outperforms FCOS [19], SAPD [20] and BorderDet [21] with heavier backbones and surpasses the best detector GFL [22] by 0.5 AP.

4.4. Visualization

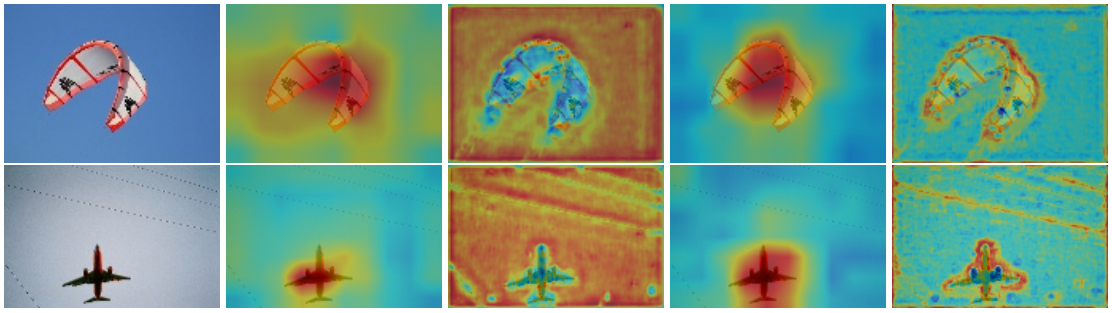
To illustrate the effectiveness of our method intuitively, we visualize the task-specific features $\boldsymbol{X}^{task}$ for both parallel head (P-Head) and G-Head. Compared with the parallel head, our G-Head learns more discriminative features as shown in Fig 3. On the one hand, G-Head captures more comprehensive and more informative semantics than the parallel head. Thus, it can focus on the objects more accurately as demonstrated in Fig 3 (d). On the other hand, G-Head can exactly describe the boundaries of an object as shown in Fig 3 (e). In contrast, the conventional parallel head might be distracted by background noises due to its distributed responses as shown in Fig 3 (c).

5. CONCLUSION

In this paper, we present a novel head structure called G-Head to enhance the interaction between different tasks in object detection. By decomposing the multi-task learning problem into multiple fragmented aspects, G-Head can ease the training process greatly. Our G-Head is a substitute for the conventional parallel head and it can significantly boost the performance while allocating fewer parameters and FLOPs. Under single-model and single-scale test, our method achieves 48.7 AP, which is very competitive against other state-of-the-art models on the challenging MS COCO dataset.

6. ACKNOWLEDGMENT

This work was supported by the Scientific Instrument Developing Project of the Chinese Academy of Sciences under Grant YJKYYQ20200045.



(a) Input (b) P-Head cls. (c) P-Head reg. (d) G-Head cls. (e) G-Head reg.

Fig. 3. Visualization of the features for different tasks. P-Head stands for the parallel head and the abbreviations cls. and reg. represent classification and regression, respectively.

7. REFERENCES

[1] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.

[2] Ross B. Girshick, "Fast R-CNN," in *ICCV*, 2015, pp. 1440–1448.

[3] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *NIPS*, 2015, pp. 91–99.

[4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg, "SSD: single shot multibox detector," in *ECCV*, 2016, pp. 21–37.

[5] Joseph Redmon and Ali Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.

[6] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 936–944.

[7] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," in *ICCV*, 2017, pp. 2999–3007.

[8] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He, "FCOS: fully convolutional one-stage object detection," in *ICCV*, 2019, pp. 9626–9635.

[9] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *CVPR*, 2020, pp. 9756–9765.

[10] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, Lei Li, and Jianbo Shi, "Foveabox: Beyound anchor-based object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 7389–7398, 2020.

[11] Shengkai Wu, Xiaoping Li, and Xinggang Wang, "Iou-aware single-stage object detector for accurate localization," *Image Vis. Comput.*, vol. 97, pp. 103911, 2020.

[12] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick, "Microsoft COCO: common objects in context," in *ECCV*, 2014, pp. 740–755.

[13] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon, "CBAM: convolutional block attention module," in *ECCV*, 2018, pp. 3–19.

[14] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai, "Deformable convnets V2: more deformable, better results," in *CVPR*, 2019, pp. 9308–9316.

[15] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018, pp. 7132–7141.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[17] Chenchen Zhu, Yihui He, and Marios Savvides, "Feature selective anchor-free module for single-shot object detection," in *CVPR*, 2019, pp. 840–849.

[18] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye, "Freeanchor: Learning to match anchors for visual object detection," in *NIPS*, 2019, pp. 147–155.

[19] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He, "FCOS: A simple and strong anchor-free object detector," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 1922–1933, 2022.

[20] Chenchen Zhu, Fangyi Chen, Zhiqiang Shen, and Marios Savvides, "Soft anchor-point object detection," in *ECCV*, 2020, pp. 91–107.

[21] Han Qiu, Yuchen Ma, Zeming Li, Songtao Liu, and Jian Sun, "Borderdet: Border feature for dense object detection," in *ECCV*, 2020, pp. 549–564.

[22] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *NIPS*, 2020.

[23] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, "Aggregated residual transformations for deep neural networks," in *CVPR*, 2017, pp. 5987–5995.