# CMFPN: Context Modeling Meets Feature Pyramid Network

**Faroq AL-Tam, Muhammad AL-Qurishi, Thariq Khalid Kadavil, Riad Souissi**
ELM Company, Riyadh - Saudi Arabia.

## Abstract

Feature fusion is a powerful technique that enables predictors to access a semantically rich representation of an image. Feature Pyramid Networks (FPNs) are the most widely used models for fusing features. However, the context within the FPN layers is inconsistent, leading to false predictions. This article addresses the context inconsistency in FPN and proposes CMFPN, a new design that improves feature fusion by decoupling feature aggregation from context modeling. Experimental results, based on the COCO dataset, show that CMFPN effectively resolves the context issues and enhances the Average Precision (AP) results for both object detection and instance segmentation by $2.30\%$ and $1.7\%$, respectively.

## 1 Introduction and Related Work

In computer vision, a backbone network is used to extract multi-scale feature maps from an image He et al. (2016); Liu et al. (2021), which can be fused using a Feature Pyramid Network (FPN) Lin et al. (2017) to build a rich representation of the image. Besides feature fusion, FPN enhances the gradient flow Jin et al. (2022) and simplifies the prediction task Chen et al. (2021). However, it has some weaknesses. FPN brings limited benefits to large objects prediction Jin et al. (2022), and the long path from the higher to the lower layers in FPN can affect dense predictions Liu et al. (2018). Other limitations are also addressed in Guo et al. (2020); Zhao et al. (2017); Xie et al. (2023).

FPN performs Feature Aggregation (FA) and Context Modeling (CM) simultaneously through lateral connections (from the backbone) and top-down path (Figure 1). Therefore, the context at the top layers of the pyramid is inconsistent with the context at the lower ones. Causing out-of-context predictions in many cases (more details in Appendix A.1). To resolve the context inconsistency, this work presents CMFPN. A new design that decouples FA from CM by introducing a new global context path (depicted in green in Figure 1).

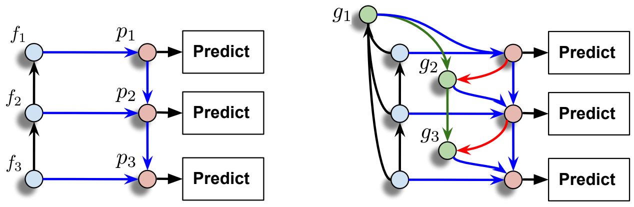

Figure 1: Feature pyramid networks of three layers. Left: FPN and right: CMFPN.

## 2 Method

### 2.1 Revising FPN

Given an input image $I \in \mathbb{R}^{C\times N^2}$, where $C$ and $N$ are the channel and spatial dimensions of $I$, respectively. A set of feature maps $\mathcal{F} = \{f_1, \cdots, f_K\}, \forall f_k \in \mathbb{R}^{C_k\times N_k^2}$ can be extracted from $I$ using a backbone and used to build a pyramid of feature maps $\mathcal{P} = \{p_1, \cdots, p_K\}, \forall p_k \in \mathbb{R}^{C_\mathcal{P}\times N_k^2}$. Assume $\mathcal{F}$ and $\mathcal{P}$ are ordered by size such that $p_{k+i}$ is $2^i$ times larger than $p_k$. Then, $p_k$ can be obtained as:

$$p_k = \begin{cases} V_k(W_k f_k) & \text{if } k = 1, \\ V_k\left(W_k f_k + p_{k-1}\right) & \text{otherwise,} \end{cases} \tag{1}$$

where $W_k$ is a mapping matrix (a $\text{Conv}_{1\times1}$ layer) to compress $f_k$ to a lateral feature map with a predefined number of channels $C_{\mathcal{P}}$, and $V_k$ is a transformation matrix (a $\text{Conv}_{3\times3}$ layer). In (1), $p_{k-1}$ is assumed to be properly scaled before adding it to $W_k f_k$.

### 2.2 CMFPN

To integrate the global context in the FPN design, the formulation in (1) can be rewritten as:

$$p_k = \begin{cases} V_k\left(W_k f_k + g_k\right) & \text{if } k = 1, \\ V_k\left(W_k f_k + g_k + p_{k-1}\right) & \text{otherwise,} \end{cases} \tag{2}$$

where $g_k$ is the global context obtained initially from $\mathcal{F}$ and updated at every level $k$. To calculate $g$, two main steps are applied: backbone feature maps re-calibration and context updating, which are described in details in Appendix A.2.

## 3 Main Results

The object detection and instance segmentation results are listed in Tables 1 and 2, respectively. They clearly show that, CMFPN has consistently boosted the prediction results regardless of object size, backbone, and task . Further details and an ablation study are presented in Appendix A.3.

| **Model** | **Backbone** | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|
| Faster R-CNN | R-50 + FPN | 36.90 | 58.40 | 39.70 | 21.70 | 40.50 | 48.10 |
| Faster R-CNN | R-50 + CFPN Xie et al. (2023) | 37.20 | - | - | 21.70 | 41.40 | 48.60 |
| YOLOF | R-50 Chen et al. (2021) | 37.70 | 56.90 | 40.60 | 19.10 | 42.5 | 53.20 |
| Faster R-CNN | R-50 + CMFPN | $39.00_{(+2.1)}$ | 60.50 | 42.30 | 22.90 | 42.20 | 51.60 |
| Mask R-CNN | R-50 + FPN | 37.40 | 58.50 | 40.10 | 21.70 | 40.70 | 48.60 |
| Mask R-CNN | R-50 + CMFPN | $39.60_{(+2.2)}$ | 60.90 | 42.90 | 23.80 | 43.00 | 52.40 |
| Cascade Mask R-CNN | R-50 + FPN | 40.70 | 59.10 | 44.30 | 22.50 | 44.30 | 54.00 |
| Cascade Mask R-CNN | R-50 + CFPN Xie et al. (2023) | 41.50 | - | - | 24.10 | 45.70 | 54.00 |
| Cascade Mask R-CNN | R-50 + CMFPN | $42.90_{(+2.2)}$ | 62.00 | 46.40 | 25.40 | 46.60 | 57.10 |
| Mask R-CNN | Swin-T + FPN | 42.40 | 65.10 | 46.10 | 25.80 | 45.60 | 56.10 |
| Mask R-CNN | Swin-T + CMFPN | $45.10_{(+2.7)}$ | 67.00 | 48.90 | 27.30 | 48.80 | 60.40 |

Table 1: The object detection results on the coco *val* 2017.

| **Model** | **Backbone** | $AP^{Seg}$ | $AP^{Seg}_{50}$ | $AP^{Seg}_{75}$ | $AP^{Seg}_{S}$ | $AP^{Seg}_{M}$ | $AP^{Seg}_{L}$ |
|---|---|---|---|---|---|---|---|
| Mask R-CNN | R-50 + FPN | 33.90 | 55.10 | 36.00 | 16.00 | 36.50 | 49.80 |
| Mask R-CNN | R-50 + CMFPN | $35.60_{(+1.7)}$ | 57.50 | 37.70 | 17.50 | 38.20 | 51.90 |
| Cascade Mask R-CNN | R-50 + FPN | 35.30 | 56.00 | 37.80 | 16.20 | 38.00 | 51.80 |
| Cascade Mask R-CNN | R-50 + CMFPN | $37.10_{(+1.8)}$ | 58.50 | 39.70 | 18.40 | 39.80 | 54.30 |
| Mask R-CNN | Swin-T + FPN | 39.10 | 62.10 | 42.10 | 19.60 | 41.80 | 57.50 |
| Mask R-CNN | Swin-T + CMFPN | $40.70_{(+1.6)}$ | 64.20 | 43.70 | 21.00 | 43.80 | 60.00 |

Table 2: The instance segmentation results on the coco *val* 2017.

## 4 Conclusion

This work presents CMFPN, a new design to fix the context issues in FPN by decoupling feature aggregation from context modeling. CMFPN provides consistent context for all feature pyramid maps. This is reflected in a significant improvement in both object detection and instance segmentation regardless of object size, task, or feature extractor. In the future, CMFPN will be evaluated under different datasets and tasks. New context modeling layers will also be explored.

### URM Statement

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

## References

Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Mmdetection: Open mmlab detection toolbox and benchmark, 2019.

Qiang Chen, Yingming Wang, Tong Yang, Xiangyu Zhang, Jian Cheng, and Jian Sun. You only look one-level feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13039–13048, June 2021.

C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan. Augfpn: Improving multi-scale feature learning for object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12592–12601, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society. doi: 10.1109/CVPR42600.2020.01261. URL `https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.01261`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Zhenchao Jin, Dongdong Yu, Luchuan Song, Zehuan Yuan, and Lequan Yu. You should look at all objects. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 332–349, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20077-9.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022, October 2021.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Jin Xie, Yanwei Pang, Jing Pan, Jing Nie, Jiale Cao, and Jungong Han. Complementary feature pyramid network for object detection. *ACM Trans. Multimedia Comput. Commun. Appl.*, 19(6), may 2023. ISSN 1551-6857. doi: 10.1145/3584362. URL https://doi.org/10.1145/3584362.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

# A Appendix

## A.1 Context issues in FPN

FPN suffers from inconsistent context among the pyramid feature maps. As shown in Figure 1, each feature map in the top-down path ($p$ path) is mainly constructed from a single lateral feature map and the context is accumulated from top to bottom which cause two main issues: $i$) the context is added to the pyramid feature maps without proper calibration; $ii$) the objects are evaluated under inconsistent context at each level.

All types of wrong predictions can be caused by these limitations, but false positives are the most common issues. Taking Figure 2 as an example, it is clear that a robust FPN-based model (Cascade Masked RCNN + ResNet + FPN in this case) fails to correctly predict objects due to several reasons. However, they can be related to context issues. In the first example, weak features cause a wrong prediction of tree leaves. In the second example, the resemblance between the car-door handle and the toilette was not validated by the context. The same issue appears in the third example. In the last example, the similarity between snow and the white ground causes false snowboard prediction. On the other hand, when the context was modeled properly by the proposed approach, all false predictions were avoided.

## A.2 Calculating the global context

The calculation and update of the global context is shown in Figure 3 and is detailed in the next steps.

### A.2.1 Transforming Backbone Feature Maps

In this step, the feature maps in $\mathcal{F}$ are recalibrated using a squeeze-and-excitation (SE) layer Hu et al. (2018) and scaled to the same size. Let $\bar{k} = \lceil \frac{K}{2} \rceil$ be the position of the mid-sized feature map in $\mathcal{F}$, then $f_k$ can be transformed as:

$$\tilde{f}_k = \text{Scale}_{2(\bar{k}-k)}(\text{SE}(f_k)), \quad k = 1, \cdots, K, \tag{3}$$

where $\text{Scale}_{2(\bar{k}-k)}$ is the scaling operator with the subscript being the scale ratio. The transformed set of backbone feature maps is denoted by $\tilde{\mathcal{F}}$.

### A.2.2 Updating the Context

To maintain a global context $g$ for use during the generation of $\mathcal{P}$, $g$ is updated at each level of the pyramid as follows:

$$g_k = \begin{cases} V_k^g \text{Concat}_{C_\mathcal{P}}(\tilde{\mathcal{F}}) & \text{if } k = 1, \\ V_k^g \left( W_k \tilde{f}_{k-1} + \text{CCM}(g_{k-1}, p_{k-1}) + g_{k-1} \right) & \text{otherwise,} \end{cases} \tag{4}$$

where $\text{Concat}_{C_\mathcal{P}}$ is the concatenation operator applied on the channel dimension $C_\mathcal{P}$. The Cross-Context Modeling (CCM) layer calculates a simplified cross attention between the latest global context $g_{k-1}$ and the feature map $p_{k-1}$.

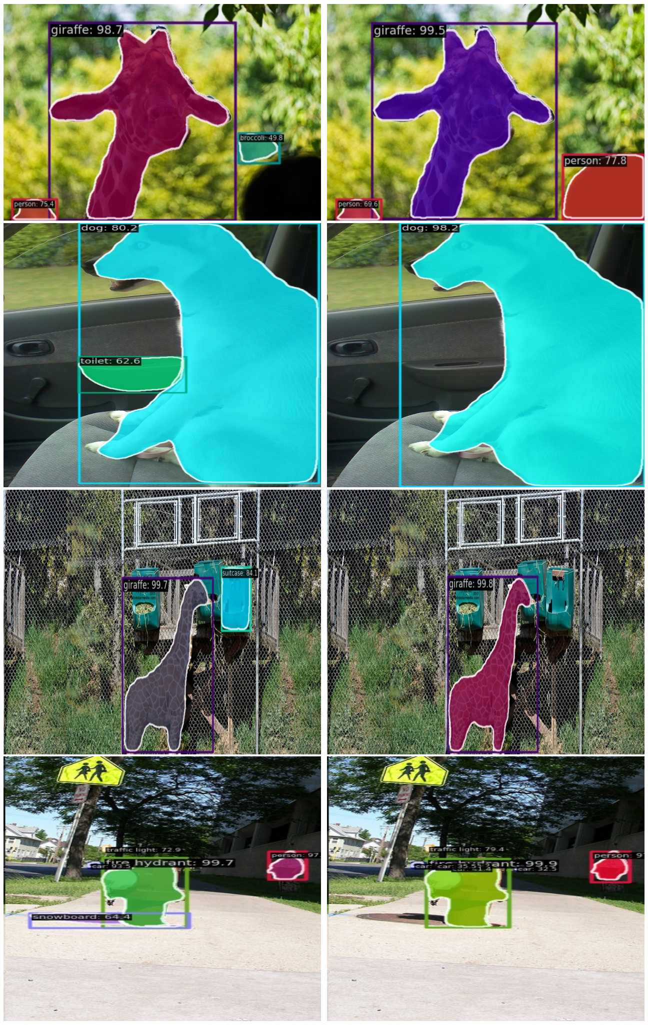

Figure 2: Context issues in FPN predictions. Left: FPN, and right: the proposed CMFPN. The model being used is the Cascade Masked RCNN with ResNet-50.

### A.2.3 Cross Context Modeling (CCM)

Building on the concepts in Wang et al. (2018) and Cao et al. (2019), a single context can be calculated for any query, leading to a substantial reduction in the computation of the attention mechanism. Therefore, the CCM layer can be calculated as:

$$\text{CCM}(g_k, p_k) = V_k^{\text{CM}} \text{CM}\big(\text{Concat}_{C_{\mathcal{P}}}\left(g_k, \text{Scale}_{2^{(\bar{k}-k)}}(p_k)\right)\big), \tag{5}$$

where CM is the Context Modeling block Cao et al. (2019):

## A.3 Experimental Results

### A.3.1 Dataset

The MS-COCO 2017 dateset Lin et al. (2014) was used to evaluate the proposed model. All models used in this work are trained on the training set *train-2017*, which contains 118k samples, and evaluated on the the validation set, which contains 5k samples.

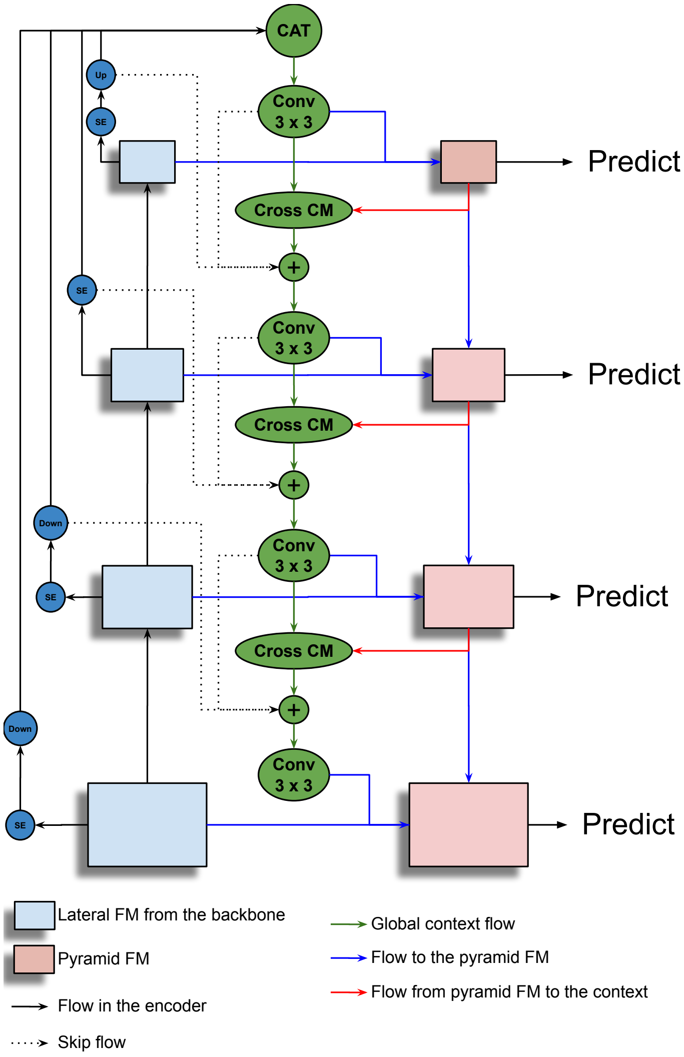

Figure 3: CMFPN

### A.3.2 Implementation details

The new module, CMFPN, has been implemented in PyTorch in MMDetection Chen et al. (2019) framework and evaluated with the same configurations widely adopted in the literature. The backbones used are initialized with pre-trained weights on ImageNet while the new components are initialized randomly. The input images are resized to $800 \times 1333$. The loss functions being adopted is the cross-entropy for both class and masks predictions, and $L_1$ (smoothed $L_1$ for the Swin models) for bounding boxes regression. The optimizer used to train the models is the SGD (AdamW with betas=$\{0.9, 0.999\}$ for the Swin models), with the adopted hyper-parameters as shown in Table 3.

During training, *RandomFlip* was the only augmentation applied. During inference, the input is resized to the same scale as in the training and no test-time augmentation was applied. A filtering score of 0.05 was used to suppress background bounding boxes where 1000 predictions were reported at each pyramid level. NMS with IoU threshold of 0.5 per class was used to keep only 100 confident predictions per image.

### A.3.3 Results Analysis

For object detection (Table 1), CMFPN brings $\approx 2.30\%$ improvement in average with minimum and maximum of $2.10\%$ and $2.7\%$, respectively. CMFPN has consistent performance when tested with

| **Parameter** | **Value** |
|---|---|
| momentum | 0.9 |
| weight decay | 1e-4 |
| initial RL | 0.02 |
| schedules | 1x schedule (12 epochs) |
| scheduler | decay of 0.1 at at the 9th and 11th epochs. |
| CMFPN layers | 2 |
| $C_{\mathcal{P}}$ | 256 |
| GPU | $4 \times$ A100 |
| batchsize | 32 (8 per GPU) |

Table 3: The parameters being adopted for training.

different heads and backbones. Moreover, all objects benefit from CMFPN regardless of their sizes. However, large objects are the most beneficial, where it outweighs FPN by $\approx 4.30\%$, when using the Mask R-CNN-Swin-T model, and $\approx 3.67\%$ in average for all models. This large improvement is aligned with the recent research regarding FPN inefficiency for large objects prediction Jin et al. (2022). Moreover, CMFPN is $\approx -1.6\%$ lower than YOLOF when using Faster R-CNN + CMFPN. However, CMFPN is more consistent and is better than YOLOX in small objects detection with $\approx 3.8\%$ and in average CMFPN outweighs YOLOX by $\approx 1.3\%$. This demonstrates the effectiveness of the proposed context path in the CMFPN design in boosting the performance and resolving the context limitation in FPN.

Regarding the instance segmentation results (Table 2), CMFPN improves the results by $\approx 1.7\%$ in average with minimum of $1.6\%$ and maximum of $1.8\%$. CMFPN has improved the results of all objects regardless of their sizes, and it has performed consistently when used in different architectures. For small objects, the results have improved by $1.7\%$ in average with maximum and minimum of $2.2\%$ (for Cascade Mask R-CNN + ResNet-50) and $1.4\%$ (for Mask R-CNN + Swin-T), respectively. Large objects masks have also improved by $2.36\%$ in average with maximum of $2.5\%$( for Cascade Mask R-CNN + ResNet-50 and Mask R-CNN + Swin-T) and minimum of $2.1\%$ (Mask R-CNN + ResNet-50).

### A.3.4 Ablation study

To perform ablation, the Mask R-CNN + Swin-T + CMFPN model is selected since it is the most powerful model among the evaluated choices. The first component is the complexity of the context $g$ ($C_{\mathcal{P}}$ in Section 2.1), which can be small (128 channels) or large (256 channels). Another component is the residual connections from the lateral maps to the context path (dashed line in Figure 3).

The results for object detection and instance segmentation are shown in Tables 4 and 5, respectively. Increasing $C_{\mathcal{P}}$ from 128 to 256 channels improves the object detection by $\approx 0.8\%$ and instance segmentation by $\approx 1.4\%$, while adding the residual connections improves the detection results by $\approx 0.3\%$ and instance segmentation by $\approx 0.1\%$.

| **Model** | AP | $\text{AP}_{50}$ | $\text{AP}_{75}$ | $\text{AP}_{\text{S}}$ | $\text{AP}_{\text{M}}$ | $\text{AP}_{\text{L}}$ |
|---|---|---|---|---|---|---|
| Mask R-CNN + Swin-T + CMFPN ($C_{\mathcal{P}}$ = 128) | 44.00 | 66.00 | 47.90 | 26.40 | 47.80 | 58.90 |
| + $C_{\mathcal{P}}$ = 256 | 44.80 | 66.90 | 48.60 | 27.00 | 48.70 | 59.60 |
| + Residual Connections | 45.10 | 67.00 | 48.90 | 27.30 | 48.80 | 60.40 |

Table 4: Ablation study results of the object detection on the coco *val* 2017.

| | | | | | | |
|---|---|---|---|---|---|---|
| Mask R-CNN + Swin-T + CMFPN ($C_{\mathcal{P}}$ = 128) | 39.20 | 62.30 | 41.90 | 19.50 | 42.30 | 57.50 |
| + $C_{\mathcal{P}}$ = 256 | 40.60 | 63.80 | 43.50 | 20.60 | 43.80 | 59.30 |
| + Residual Connections | 40.70 | 64.20 | 43.70 | 21.00 | 43.80 | 60.00 |

Table 5: Ablation study results of the instance segmentation on the coco *val* 2017.