ROBUST AND ACCURATE OBJECT DETECTION VIA SELF-KNOWLEDGE DISTILLATION

Weipeng Xu\textsuperscript{1} Pengzhi Chu\textsuperscript{1} Renhao Xie\textsuperscript{1} Xiongziyan Xiao\textsuperscript{1} Hongcheng Huang\textsuperscript{1,*}

\textsuperscript{1} Shanghai Jiao Tong University, China

ABSTRACT

Object detection has achieved promising performance on clean datasets, but how to achieve better tradeoff between the adversarial robustness and clean precision is still under-explored. Adversarial training is the mainstream method to improve robustness, but most of the works will sacrifice clean precision to gain robustness than standard training. In this paper, we propose Unified Decoupled Feature Alignment (UDFA), a novel fine-tuning paradigm which achieves better performance than existing methods, by fully exploring the combination between self-knowledge distillation and adversarial training for object detection. With extensive experiments on the PASCAL-VOC and MS-COCO benchmarks, the evaluation results show that UDFA can surpass the standard training and state-of-the-art adversarial training methods for object detection. For example, compared with teacher detector, our approach on GFLV2 with ResNet-50 improves clean precision by 2.2 AP on PASCAL-VOC; compared with SOTA adversarial training methods, our approach improves clean precision by 1.6 AP, while improving adversarial robustness by 0.5 AP. Our code is available at https://github.com/grispeut/udfa.

Index Terms— deep learning, object detection, adversarial robustness, knowledge distillation

1. INTRODUCTION

Deep learning has made a significant breakthrough in many computer vision tasks such as classification [1], object detection [2, 3, 4] and instance segmentation [5, 6]. However, adversarial samples [7, 8, 9, 10] are still viewed as a troublesome threat to deep learning models. Adversarial samples are crafted for leading deep learning models to make wrong predictions by adding imperceptible perturbations to clean images. Generally, when the knowledge of structures and parameters of a given model is available, various methods can successfully fool the model with so-called white-box attack [8, 11].

In order to improve the model robustness, many efforts have been made to defend adversarial attack [8, 11, 12]. Among them, adversarial training [8, 11] has been demonstrated to be the most effective method so far. Based on adversarial training, a number of approaches have been proposed to enhance its performance further [13, 14]. Although the above approaches have verified their effectiveness on classification task, they do not extend their work to the object detection task. Compared with classification task, detection requires predicting the locations of semantic objects, which is complicated but also important.

To improve the robustness of object detection models, Zhang et al. [15] first present a method to combine adversarial training with detection tasks by using task oriented domain to generate adversarial samples in adversarial training (denoted as TOD-AT). Chen et al. [16] further propose class-wise adversarial training especially for the situations where multiple objects of different classes exist in single image. The above two works can indeed enhance adversarial robustness of detectors than standard training. However, performance on clean datasets drops significantly (about a drop of 20% mAP on PASCAL-VOC [17]), which is not conducive to real-world applications. In order to balance clean precision and robustness, Feature Alignment [18] proposes to guide the outputs of middle layers for better adversarial training (denoted as Vanilla-FA), which builds on adversarial training, knowledge distillation, and self-supervised learning. Although Vanilla-FA can mitigate the drop of clean precision and boost robustness of detectors, the clean precision is still lower than standard training. For robust and accurate object detection, Det-AdvProp [19] combines TOD-AT with AdvProp [20], making clean precision and robustness outperform standard training. To the best of our knowledge, Det-AdvProp is the state-of-the-art (SOTA) result in adversarial-training-based algorithms for object detection, which does not sacrifice clean accuracy to gain robustness.

This paper aims at building more accurate and robust detectors. Motivated by Vanilla-FA, we propose a Unified Decoupled Feature Alignment (UDFA) framework, which can largely improve the performance than Vanilla-FA. Besides, UDFA outperforms standard training and Det-AdvProp in terms of clean precision and robustness. Vanilla-FA uses additional Knowledge-Distilled Feature Alignment (K DFA) branch and Self-Supervised Feature Alignment (SSFA) branch to promote adversarial training. K DFA constructs contrastive loss between student and teacher; SSFA constructs contrastive loss between adversarial images and clean images from student. However, the interactions and de-
The overall pipeline of UDFA. The SSFA module constructs feature loss from student detector, and the KDFA module constructs feature loss between teacher and student.

dependencies of two branches have not been explored. They directly push student to imitate different feature representations from knowledge distillation and self-supervised learning, which will result in conflicts or competitions. We conjecture that the inner compromise leads to the sub-optimal results. In light of the lessons above, we propose to unify the two independent branches by knowledge distillation. We find that original KDFA branch can essentially be viewed as a self-supervised learning branch and a new knowledge distillation branch. Therefore, our UDFA framework is composed of an original KDFA branch and two decoupled sub-branches, which can address the conflict in theory. This paper aims at exploring a novel fine-tuning paradigm by self-knowledge distillation, so we use pretrained detector as teacher. In addition, we adopt DeFeat [21] to fit it into the feature alignment framework. We show that decoupled features can improve the performance than global pooling feature used in Vanilla-FA. With extensive experiments on PASCAL-VOCS [17] and MS-COCO [22] benchmarks, the results demonstrate that our UDFA can effectively improve the clean accuracy and robustness than SOTA algorithms.

The main contributions of our work can be summarized as follows: i) our approach is the first distillation-based approach which does not sacrifice clean precision to gain robustness in adversarial training; ii) we explain the self-knowledge distillation in adversarial training from a new perspective by viewing self-supervised learning as a sub-branch of original self-knowledge distillation branch.

2. METHODS

In this section, we first analyze the potential drawbacks in Vanilla-FA [18], and then introduce our UDFA framework.

2.1. Potential Conflict in Vanilla-FA

Vanilla-FA proposes Knowledge-Distilled Feature Alignment (KDFA) and Self-Supervised Feature Alignment (SSFA) to guide the output of intermediate feature layer. Without loss of generality, the norm $\| \cdot \|$ is used to measure the distance between two feature vectors, which can be formulated as follows:

$$L_{fe1} = \|S(x_{adv}) - T(x_{clean})\|_p,$$  
$$L_{fe2} = \|S(x_{adv}) - S(x_{clean})\|_p,$$  

where $x_{clean}, x_{adv}, S(\cdot), T(\cdot)$ are clean images, adversarial images, feature extractor of student and teacher. And $p$ specifies a particular norm. $L_{fe1}$ denotes KDFA branch, which guides $S(x_{adv})$ to imitate $T(x_{clean})$. $L_{fe2}$ denotes SSFA branch between $S(x_{adv})$ and $S(x_{clean})$, which are generated from student detector. $S(x_{clean})$ and $T(x_{clean})$ are served as soft labels to guide the learning of $S(x_{adv})$. Nevertheless, KDFA and SSFA are two independent branches and $S(x_{clean})$ is mutable during the training process. Therefore we can not ensure the credibility of $S(x_{clean})$ in the guiding process. Moreover, inconsistent soft labels might hurt the distillation performance, leading to further potential conflicts and sub-optimal results.

We propose decoupled knowledge-distilled feature alignment branches to avoid potential conflict between vanilla KDFA and SSFA branch. Following the triangle inequality, we can disentangle vanilla KDFA branch into two sub-branches, which can be expressed as follows:

$$L_{fe3} = \|S(x_{clean}) - T(x_{clean})\|_p,$$  
$$L_{fe1} \leq L_{fe2} + L_{fe3},$$  

where $L_{fe3}$ denotes a new KDFA branch between $S(x_{clean})$ and $T(x_{clean})$. When we use the teacher detector to initialize student detector, $S(x_{clean})$ and $T(x_{clean})$ are completely coincident in feature space. Theoretically, minimizing $L_{fe1}$ and minimizing $L_{fe2} + L_{fe3}$ have the same optimal solutions, namely $S(x_{adv}), S(x_{clean})$ and $T(x_{clean})$ coinciding in feature space. However, the parameters of student model are different from teacher during the training process. Therefore the optimal solutions are almost impossible to obtain under total
loss function in detection task. When \( L_{\text{fea}2} + L_{\text{fea}3} \) or \( L_{\text{fea}1} \) converges in a balanced state, \( S(x_{\text{clean}}) \) and \( S(x_{\text{adv}}) \) will be constrained to the manifold of \( T(x_{\text{clean}}) \) in feature space.

2.2. UDFA Framework

Based on the above analysis, UDFA is designed by adding a new KDFA branch between \( S(x_{\text{clean}}) \) and \( T(x_{\text{clean}}) \) based on Vanilla-FA. The overall architecture of UDFA is illustrated in Figure 1. The teacher detector is trained on clean datasets with standard training setting, and served as pretrained model to initialize student in adversarial training. For fair comparisons, we do not use more powerful teacher in our UDFA framework. Therefore, UDFA is essentially a self-distillation process, which aims at exploring more information from pretrained detector. Unlike traditional knowledge distillation, our framework also involves self-supervised learning branch to improve robustness, and aims at addressing the conflict between original knowledge distillation task and self-supervised learning task. In order to improve the effectiveness of distillation, we adopt decoupled fore/back-ground features [21] instead of global pooling feature used in Vanilla-FA. Compare with global pooling feature, decoupled features contains location information by distinguishing foreground and background region, more friendly to object detection task. The overall training targets can be summarized as:

\[
L_{\text{total}} = \alpha \ast (L_{\text{cls}}(x_{\text{clean}}) + L_{\text{loc}}(x_{\text{clean}})) + (1 - \alpha) \ast (L_{\text{cls}}(x_{\text{adv}}) + L_{\text{loc}}(x_{\text{adv}})) + \beta \ast L_{\text{fea}}(x_{\text{clean}}, x_{\text{adv}}),
\]

(5)

where \( L_{\text{loc}}(\cdot) \) is the location loss of detector (e.g., GIoU loss [23]) and \( L_{\text{cls}}(\cdot) \) is the classification loss (e.g., focal loss [24]). \( \alpha \) is the weight to balance the detection loss of clean images and adversarial images. \( L_{\text{fea}}(\cdot) \) is the feature alignment loss in UDFA. \( \beta \) controls the trade-off between detection loss and feature alignment loss.

Feature alignment loss \( L_{\text{fea}} \) is composed of (1) KDFA loss \( L_{\text{fea}1} \) between \( S(x_{\text{adv}}) \) and \( T(x_{\text{clean}}) \); (2) SSFA loss \( L_{\text{fea}2} \) between \( S(x_{\text{adv}}) \) and \( S(x_{\text{clean}}) \); (3) KDFA loss \( L_{\text{fea}3} \) between \( S(x_{\text{clean}}) \) and \( T(x_{\text{clean}}) \). Unlike Vanilla-FA only using the output of backbone, UDFA uses multi-level features from feature pyramid network to transfer dark knowledge, which can provide more powerful supervised information to guide feature alignment. Motivated by [21], \( L_{\text{fea}i} \) is defined as follows:

\[
L_{\text{fea}i}(S, T) = \frac{\gamma_{bg}}{N_{bg}} \sum_{h=1}^{H} \sum_{w=1}^{W} (1 - M_{h,w}) (S_{h,w} - T_{h,w})^2
\]

\[
+ \frac{\gamma_{fg}}{N_{fg}} \sum_{h=1}^{H} \sum_{w=1}^{W} M_{h,w} (S_{h,w} - T_{h,w})^2, \quad i = 1, 2, 3
\]

(6)

where \( S \) and \( T \) denote feature map from student or teacher. \( N_{fg} \) and \( N_{bg} \) are the number of elements in foreground regions and background regions. \( \gamma_{fg} \) and \( \gamma_{bg} \) are trade-off parameters in fore/back-ground features. \( M \) is the binary mask of feature map corresponding to input image. \( h \) and \( w \) denote the size of feature map.

3. EXPERIMENT

3.1. Datasets and Implementation Details

For datasets, we use PASCAL-VOC and MS-COCO to verify our approach. We conduct experiments across one-stage detectors (e.g. GFLV2 [4]) and two-stage detectors (e.g. Faster-RCNN [2]) with ResNet [1] backbone of different scales. All the pretrained detectors (also served as teacher detectors in our work) are finetuned from ImageNet [25] pretrained classifiers. For adversarial attack, we use typical PGD attack [11] following Vanilla-FA [18].

3.2. Comparison with Existing Methods

We first verify the effectiveness of our proposed UDFA on typical one-stage detector GFLV2. Comparison of the results on PASCAL-VOC benchmark are shown in Table 1. For adversarial robustness, we use different PGD steps to generate adversarial samples with a standard perturbation budget setting (8/255) for test. Without loss of generality, we also report the average adversarial AP under different steps (1,2,4), which is denoted as adv AP. We also use standard settings (without adversarial training) to finetune pretrained detector for a fair comparison, and denote it as STD. It is observed that STD obtains lower clean AP and higher adv AP than pretrained detector. According to our previous experience, we will attribute the decline in clean precision to the overfitting phenomenon. However, the improvement on robustness demonstrates that there might be a tradeoff between clean accuracy and robustness even without adversarial training settings. For adversarial training algorithms, similar to the previous observations, Vanilla-AT, TOD-AT and Vanilla-FA

| method      | backbone | clean AP | adv AP |
|-------------|----------|----------|--------|
|             |          | AP | AP50 | AP75 | AP | AP50 | AP75 |
| PRE         | Res-18   | 51.1 | 74.6 | 55.3 | 5.9 | 17.1 | 3.2 |
| STD         | Res-18   | 50.1 | 72.5 | 54.3 | 7.2 | 20.4 | 3.9 |
| Vanilla-AT  | Res-18   | 48.7 | 70.4 | 52.3 | 10.4 | 25.9 | 6.7 |
| TOD-AT      | Res-18   | 48.7 | 70.3 | 52.4 | 10.7 | 26.6 | 7.2 |
| Vanilla-FA  | Res-18   | 49.4 | 71.4 | 53.1 | 11.2 | 28.9 | 6.8 |
| Det-AdvProp | Res-18   | 51.4 | 73.8 | 55.5 | 8.1 | 22.7 | 4.6 |
| UDFA (ours) | Res-18   | 52.2 | 74.8 | 56.3 | 12.4 | 31.2 | 7.6 |

Table 1. Main results on PASCAL-VOC. PRE is pretrained detector. STD denotes standard training.
Table 2. Verify effectiveness of UDFA with AdvProp on one-stage detector (GFLV2).

| method               | backbone | clean AP | Adv AP |
|----------------------|----------|----------|--------|
|                      |          | AP | AP50 | AP75 | AP | AP50 | AP75 |
| PRE                  | Res-18   | 51.1 | 74.6 | 55.3 | 5.9 | 17.1 | 3.2  |
| STD                  | Res-18   | 50.1 | 72.5 | 54.3 | 7.2 | 20.4 | 3.9  |
| Det-AdvProp          | Res-18   | 51.4 | 73.8 | 55.5 | 8.1 | 22.7 | 4.6  |
| UDFA (ours)          | Res-18   | 52.9 | 75.9 | 57.6 | 9.1 | 24.6 | 5.1  |
| PRE                  | Res-34   | 54.9 | 77.6 | 59.7 | 7.2 | 20.2 | 3.9  |
| STD                  | Res-34   | 53.9 | 75.7 | 58.5 | 8.5 | 23.9 | 4.7  |
| Det-AdvProp          | Res-34   | 55.8 | 77.3 | 61.4 | 10.3| 27.5 | 6.2  |
| UDFA (ours)          | Res-34   | 57.3 | 79.0 | 62.6 | 10.6| 28.4 | 6.2  |

Table 3. Verify effectiveness of UDFA with AdvProp on two-stage detector (Faster-RCNN).

| method               | backbone | clean AP | Adv AP |
|----------------------|----------|----------|--------|
|                      |          | AP | AP50 | AP75 | AP | AP50 | AP75 |
| PRE                  | Res-18   | 48.5 | 77.0 | 52.2 | 1.8 | 6.0  | 0.9  |
| STD                  | Res-18   | 47.4 | 74.0 | 51.9 | 2.0 | 6.8  | 0.7  |
| Det-AdvProp          | Res-18   | 48.5 | 75.3 | 52.8 | 3.4 | 10.1 | 1.8  |
| UDFA (ours)          | Res-18   | 50.8 | 77.3 | 55.7 | 3.9 | 11.0 | 2.2  |
| PRE                  | Res-34   | 52.9 | 78.5 | 58.3 | 3.2 | 9.5  | 1.5  |
| STD                  | Res-34   | 51.8 | 76.8 | 57.4 | 3.4 | 10.0 | 1.3  |
| Det-AdvProp          | Res-34   | 53.6 | 77.8 | 59.4 | 5.8 | 15.0 | 3.8  |
| UDFA (ours)          | Res-34   | 55.7 | 79.9 | 62.1 | 5.9 | 15.0 | 4.1  |

Table 4. Clean precision on coco val2017 and test-dev2017.

Table 5. Evaluation results under different attack strengths.

| method               | backbone | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  |
|----------------------|----------|----|----|----|----|----|----|----|----|
| PRE                  | Res-18   | 25.8| 6.6| 5.8| 5.4| 5.2| 5.2| 5.2| 5.2|
| STD                  | Res-18   | 28.5| 6.5| 5.8| 5.6| 5.5| 5.5| 5.6| 5.7|
| UDFA (ours)          | Res-18   | 31.7| 8.8| 7.4| 7.0| 6.8| 6.8| 6.9| 6.9|
| PRE                  | Res-34   | 31.6| 8.3| 7.2| 6.8| 6.7| 6.7| 6.8| 6.8|
| STD                  | Res-34   | 32.2| 9.2| 8.2| 7.9| 7.9| 7.9| 8.0| 8.1|
| UDFA (ours)          | Res-34   | 34.6| 11.1| 9.6| 9.1| 8.9| 8.9| 9.0| 9.0|
| PRE                  | Res-50   | 31.2| 8.2| 7.1| 7.0| 7.1| 7.1| 7.2| 7.2|
| STD                  | Res-50   | 31.9| 9.6| 8.6| 8.6| 8.6| 8.6| 8.9| 8.9|
| UDFA (ours)          | Res-50   | 34.4| 11.0| 9.9| 9.6| 9.5| 9.6| 9.7| 9.8|

3.4. Results on MS-COCO

We further conduct experiments on MS-COCO to verify our approach. The results of GFLV2-Res34 trained on train2017 are summarized in Table 4. It is observed that UDFA obtains consistent improvement on val2017 and test-dev2017. Fore ablation studies, we conduct more experiments on GFLV2 trained with minitrain2017 [26]. In Table 5, we report the performance on val2017 under different attack strengths. Although there are still large gaps between clean precision and adversarial robustness, our approach achieves better clean precision and robustness than exiting methods.

4. CONCLUSIONS

In this paper, we propose a new adversarial training framework for building more accurate and robust detector. The proposed UDFA is specifically designed for the fine-tuning process, and benefits from pretrained detector by self-knowledge distillation. From the view of knowledge distillation, we decouple knowledge distillation branch into two sub-branches, which unify knowledge distillation task and self-supervised learning task to avoid potential conflict. Additionally, we adopt decoupled fore/back-ground features for effective distillation and feature alignment. With extensive experiments on PASCAL-VOC and MS-COCO benchmarks, the evaluation results demonstrate that our UDFA can obtain higher clean precision and robustness than standard training, and can obtain higher clean precision and comparable or higher robustness than SOTA adversarial-training-based algorithms.
5. REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in CVPR, 2016.

[2] 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.

[3] 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.

[4] Xiang Li, Wenhai Wang, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang, “Generalized focal loss V2: learning reliable localization quality estimation for dense object detection,” in CVPR, 2021.

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick, “Mask R-CNN,” in ICCV, 2017.

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in ECCV, 2018.

[7] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus, “Intriguing properties of neural networks,” in ICLR, 2014.

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy, “Explaining and harnessing adversarial examples,” in ICLR, 2015.

[9] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan L. Yuille, “Adversarial examples for semantic segmentation and object detection,” in ICCV, 2017.

[10] Francesco Croce and Matthias Hein, “Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks,” in ICML, 2020.

[11] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, “Towards deep learning models resistant to adversarial attacks,” in ICLR, 2018.

[12] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi N. R. Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey, “Characterizing adversarial subspaces using local intrinsic dimensionality,” in ICLR, 2018.

[13] Dongxian Wu, Shu-Tao Xia, and Yisen Wang, “Adversarial weight perturbation helps robust generalization,” in NIPS, 2020.

[14] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu, “Improving adversarial robustness requires revisiting misclassified examples,” in ICLR, 2020.

[15] Haichao Zhang and Jianyu Wang, “Towards adversarially robust object detection,” in ICCV, 2019.

[16] Pin-Chun Chen, Bo-Han Kung, and Jun-Cheng Chen, “Class-aware robust adversarial training for object detection,” in CVPR, 2021.

[17] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman, “The pascal visual object classes (VOC) challenge,” Int. J. Comput. Vis., 2010.

[18] Weipeng Xu, Hongcheng Huang, and Shaoyou Pan, “Using feature alignment can improve clean average precision and adversarial robustness in object detection,” in ICIP, 2021.

[19] Xiangning Chen, Cihang Xie, Mingxing Tan, Li Zhang, Cho-Jui Hsieh, and Boqing Gong, “Robust and accurate object detection via adversarial learning,” in CVPR, 2021.

[20] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le, “Adversarial examples improve image recognition,” in CVPR, 2020.

[21] Jianyuan Guo, Kai Han, Yunhe Wang, Han Wu, Xinghao Chen, Chunjing Xu, and Chang Xu, “Distilling object detectors via decoupled features,” in CVPR, 2021.

[22] 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.

[23] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” in CVPR, 2019.

[24] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár, “Focal loss for dense object detection,” in ICCV, 2017.

[25] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in CVPR, 2009.

[26] Nermin Samet, Samet Hicsonmez, and Emre Akbas, “Houghnet: Integrating near and long-range evidence for bottom-up object detection,” in ECCV, 2020.