2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Symmetric Fine-Tuning for Improving Few-Shot Object Detection

Emmanouil Mpampis, Nikolaos Passalis, and Anastasios Tefas
Computational Intelligence and Deep Learning Group, AIIA Lab.
Department of Informatics, Aristotle University of Thessaloniki
Thessaloniki, Greece
{empampis, passalis, tefas}@csd.auth.gr

***Abstract*—Object detection plays a crucial role in automated image analysis by identifying and localizing objects within an image. One-stage Deep Learning (DL)-based object detectors have achieved impressive results, primarily due to large-scale datasets available for training them. However, these approaches rely heavily on abundant labeled data, posing challenges when only a few samples per class are available. To this end, few-shot object detection approaches have been proposed. Among them, fine-tuning the final detection head while keeping the feature extractor/backbone frozen is a commonly used approach for few-shot object detection. This approach effectively utilizes pre-existing knowledge encoded in the backbone, using a small number of samples to learn new object categories. However, in this paper, we argue that fine-tuning only the last layers may limit accuracy and lead to overfitting if the initial layers of the detection head are not adapted for the new task. The data processing inequality, which states that information lost in early network layers cannot be recovered in subsequent ones, supports this argument. To address this issue, we propose a symmetric fine-tuning method that involves both the first and last layers of the detection head, aiming to maintain a fixed trainable parameter budget while strategically selecting parameters for fine-tuning. Experimental results demonstrate the effectiveness and efficiency of this approach and open up several interesting future research directions.**

***Index Terms*—Object detection, Few-shot, Fine-Tuning, Symmetric Fine-Tuning**

I. Introduction

Object detection combines the tasks of identifying and localizing objects within an image, enabling the automation of determining "what" and "where" certain objects are depicted. In recent years, the emergence of one-stage Deep Learning (DL)-based object detectors like YOLO [1], SSD [2], and CenterNet [3], along with the availability of large-scale datasets such as MS COCO [4] and VOC [5], has yielded impressive results. However, these approaches heavily rely on substantial amounts of labeled data for training and are not effective when only a few samples per class are available. This limitation arises in various real-world use cases, including logo detection [6], media monitoring [7], robotics [8], and others. To overcome this constraint, recent research has focused on developing few-shot object detection methods that can learn from a small number of labeled examples while retaining the advantages of one-stage object detection pipelines.

Few-shot object detection is a challenging task that involves training a model to recognize objects with minimal supervision [9]. The objective is to develop algorithms that can effectively generalize to new object domains or unseen instances with only a few annotated samples. While existing few-shot object detection methods have shown promising results [10], several challenges remain, with overfitting being among the most important ones. Overfitting occurs when the model fails to generalize well to unseen data due to the limited number of training examples, resulting in poor performance and reduced generalization ability. Additionally, many current few-shot learning methods rely on complex architectures that require training multiple models simultaneously, which increases computational complexity and implementation difficulty. Two-stage models have become a popular approach in few-shot object detection [11], [12], [13], [14]. These models employ separate region-of-interest (ROI) extraction and classifier modules, such as Faster RCNN [15]. As a result, these models can be more easily manipulated to mitigate overfitting in either the localization or classification tasks. However, such approaches are typically slower than recent one-stage approaches, making them less suitable for large scale deployment, especially in streaming applications where energy and speed restrictions exist [7].

Fine-tuning the final detection head while keeping the feature extractor/backbone frozen is a commonly used few-shot learning approach for few-shot object detection with one-stage DL detectors [1]–[3]. This approach leverages the pre-existing knowledge encoded in the backbone, using only a small number of samples to learn new categories. It is generally effective because the detection head, responsible for classification decisions, contains relatively fewer parameters compared to the rest of the network. To mitigate overfitting, fine-tuning is typically applied only to the last few layers. Although widely used, in this paper we argue that this approach may limit accuracy and potentially exacerbate overfitting. To understand this phenomenon, we need to consider the data processing inequality [16], which states that information lost in early layers cannot be recovered in subsequent layers. This result has been validated in several studies, highlighting the importance of maintaining good information flow throughout the network [17]. Consequently, fine-tuning only the last layers can lead to overfitting, since if the early layers discard useful information, the later layers may rely on irrelevant features to fit the limited available data. On the other hand, fine-tuning the entire detection head can potentially address this issue, but it also increases the risk of overfitting due to the network's larger capacity compared to the limited training samples. Therefore, the main research question addressed in this paper is whether a more structured approach to fine-tuning can be employed, enabling the introduction of

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

additional information to the detection head while reducing the risk of overfitting.

The main contribution of this paper is the introduction of a symmetric fine-tuning approach designed to overcome the challenges mentioned above. The proposed method focuses on maintaining a fixed "trainable parameter budget", which refers to the number of parameters that will be trained, while strategically selecting the most appropriate parameters for fine-tuning. To achieve this, we advocate for a symmetric fine-tuning approach that involves both the first and last layers of the detection head in the fine-tuning process. The experimental results validate the effectiveness and efficiency of the proposed approach, highlighting its potential to advance the field of few-shot object detection and open doors for more advanced techniques.

The rest of this paper is structured as follows. Section II presents the proposed symmetric fine-tuning approach. The results of the conducted experimental study are presented and discussed in Section III. Finally, Section IV concludes this paper.

II. Proposed method

Let $\mathcal{X} = \{(\mathbf{x}_1, \mathbf{t}_1), (\mathbf{x}_2, \mathbf{t}_2), \ldots, (\mathbf{x}_N, \mathbf{t}_N)\}$ represent a training set of N samples used for few-shot fine-tuning of an object detector. Here, $\mathbf{x}_i$ represents an image, and $\mathbf{t}_i$ represents the corresponding bounding box annotations in a format suitable for the employed object detector. We define $\mathbf{y} = f(\mathbf{x})$ as a single-stage detector, consisting of a feature extraction backbone denoted by $\tilde{\mathbf{y}} = f_b(\mathbf{x})$, and a detection head $\mathbf{y} = f_d(\tilde{\mathbf{y}}, \mathbf{W})$, where $\mathbf{W}$ represents the parameters of the detection head. In this work, we assume that the feature extraction backbone is already trained, so we do not explicitly define the training parameters for $f_b(\cdot)$. Additionally, we use M to represent the number of layers involved in $f_d(\cdot, \mathbf{W})$, where $\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_M]$, and $\mathbf{W}_i$ denotes the parameters of the i-th layer. The specific choice of the object detection algorithm, such as YOLO [1], and SSD [2], determines the loss function used for training, denoted as $\mathcal{L}(\mathbf{y}, \mathbf{t})$. It is important to note that the proposed method is not affected by the selection of the loss function, feature extraction backbone, or detection head architecture. It can be applied without restrictions to any single-stage object detection approach, provided that a multi-layer detection head is employed.

In traditional few-shot fine-tuning, we typically choose the $K > 0$ last layers and optimize the detection head as follows:

$$\mathbf{W}_T^* = \arg\min_{\mathbf{W}_T} \sum_{i=1}^{N} \mathcal{L}(f_d(f_b(\mathbf{x}_i), \mathbf{W}), \mathbf{t}_i), \tag{1}$$

where $\mathbf{W}_T = [\mathbf{W}_{M-K+1}, \ldots, \mathbf{W}_M]$. This means we focus on training only the last M layers of the detection head, while keeping the remaining parameters fixed.

In contrast with the aforementioned approach, the proposed method introduces a parameter budget B to limit the number of parameters eligible for optimization. To simplify the method, we allocate the budget based on layers rather than the actual count of parameters. Hence, B represents the maximum number of layers that can be optimized. Our symmetric fine-tuning approach involves optimizing the first B layers along with the last B layers. This allows the detection head to uncover information from the backbone that may have been suppressed during the initial training of the object detector, such as task-irrelevant details. Consequently, we formulate the optimization problem as follows:

$$\mathbf{W}_P^* = \arg\min_{\mathbf{W}_P} \sum_{i=1}^{N} \mathcal{L}(f_d(f_b(\mathbf{x}_i), \mathbf{W}), \mathbf{t}_i), \tag{2}$$

where

$$\mathbf{W}_P = [\mathbf{W}_1, \ldots, \mathbf{W}_B, \mathbf{W}_{M-B+1}, \ldots, \mathbf{W}_{M-1}, \mathbf{W}_M]. \tag{3}$$

An important question arises concerning the unoptimized intermediate layers and their suitability for the task at hand. However, both our experimental evaluation and recent literature [18] suggest that these layers can indeed learn generic feature combinations in a meaningful way, enabling effective optimization. It should be noted that the proposed method can also be viewed as a mean of enforcing a prior in the detection process. By keeping the intermediate layers frozen, we constrain the solution space and mitigate possible overfitting phenomena. Furthermore, for fair comparisons in our experimental evaluation, we specifically consider cases where $K = 2B$, ensuring an equal number of optimized layers in both the traditional fine-tuning approach and our proposed method.

III. Experimental Evaluation

In this section we provide the experimental evaluation. First, we describe the employed evaluation setup. Then, we provide the experimental results and discuss the obtained results.

A. Experimental Setup

For all the conducted experiments we use the Single Shot MultiBox Detector (SSD) [2] using the implementation provided by Tensorflow[1]. We also employ a ResNet50 backbone as a feature extractor and an object detection head that consist of four convolutional layers before feeding the final predictor layer for classification and bounding box regression. The model was trained on the entire COCO dataset and the input images were resized to a resolution of 640×640 pixels, ensuring consistency across in-domain and out-of-domain experiments.

To perform few-shot learning we fine-tuned the model in the OpenImages dataset [19] with four classes that are not included originally in MS COCO (out-of-domain experiments), i.e., *"helmet"*, *"fish"*, *"tire"*, and *"flower"*, as well as two classes that are already included in the MS COCO dataset (in-domain experiments), i.e., *"bird"* and *"coffee cup"*. For each class, we selected a set of 10 random images from the OpenImages dataset and resized them to 640×640 pixels. These images were chosen to not include occluded and truncated examples, as well as instances where objects belonged to a group. We provide the training setup, along with code for reproducing the conducted experiments at REMOVED-FOR-PEER-REVIEW. To assess the model's performance, for each class we used a separate set of 100

[1]https://github.com/tensorflow/models

TABLE I: Evaluating different fine-tuning approaches. mAP@0.5-0.99 is reported for each class separately, as well as the average for all classes ("avg"), in-domain ('avg (in)'), and out-of-domain ('avg (out)')). Bold values indicate the best results among the same set of experiments (baseline vs. proposed). Underlined numbers indicate the best overall results.

Method	coffee cup	bird	helmet	tire	fish	flower	avg	avg (in)	avg (out)
Baseline (Last Layer)	0.759	0.455	0.017	0.03	0.24	0.17	0.278	0.607	0.114
Baseline (L2)	0.740	0.470	0.069	0.270	0.302	0.217	0.344	0.605	0.214
Proposed (L2)	**<u>0.782</u>**	**<u>0.478</u>**	**0.11**	**0.403**	**0.33**	**0.253**	**0.393**	**<u>0.630</u>**	**0.274**
Baseline (L3)	0.748	0.458	0.109	0.361	0.335	0.262	0.379	0.603	0.267
Proposed (L3)	**0.766**	**0.477**	**0.15**	**0.445**	**0.337**	**0.271**	**0.408**	**0.621**	**0.301**
Baseline (L4)	0.736	0.457	0.146	0.426	0.331	0.275	0.395	0.596	0.294
Proposed (L4)	**0.756**	**0.474**	**<u>0.157</u>**	**<u>0.466</u>**	**<u>0.390</u>**	**<u>0.280</u>**	**<u>0.420</u>**	**0.615**	**<u>0.323</u>**

randomly selected images from the OpenImages dataset, ensuring that maintained the same characteristics as the training set. We utilized the standard COCO evaluation methodology to compute our metrics for each class.

To augment the training data and compensate for the limited size of the training set, we applied various augmentations such as rotation, flip, scale, stretch, color, and positive extraction [20]. The approach of selectively extracting only positive instances was also employed to guide our model towards effectively reducing false positives during training.

We conducted experiments using the following seven setups:

- **Baseline - Last Layer**, where only the last layer (classifier and box-predictor are trained). This setup is abbreviated as "Baseline".
- **Baseline - Two Layers**, where both the predictor layer and the last convolutional layer are trained. This setup is abbreviated as "Baseline (L2)".
- **Proposed - Two Layers**, where the predictor layer and the first convolutional layer from the head are trained. This setup is abbreviated as "Proposed (L2)".
- **Baseline - Three Layers**, where the predictor layer and the two last convolutional layers are trained. This setup is abbreviated as "Baseline (L3)".
- **Proposed - Three Layers**, where the predictors and the two first convolutional layers from the head are trained. This setup is abbreviated as "Proposed (L3)".
- **Baseline - Four Layers**, where the predictors and the three last convolutional layers are trained. This setup is abbreviated as "Baseline (L4)".
- **Proposed - Four Layers**, where the predictors, the last and the first two convolutional layers from the head are trained. This setup is abbreviated as "Proposed (L4)".

For each of these experiments we use the same base model with the same hyperparameters but varying the number of layers trained.

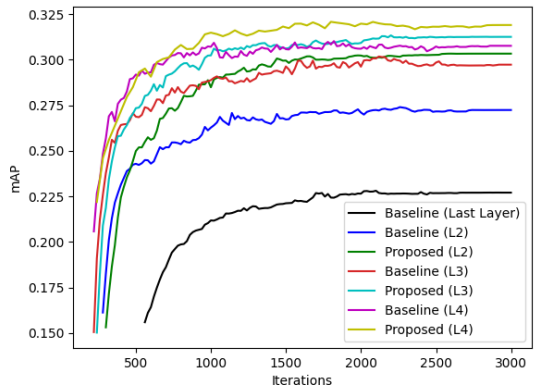
B. Experimental Evaluation

In Table I we report the mean Average Precision (mAP) for Intersection over Union (IoU) values of 0.5 to 0.99. First, we note that in the average case optimizing more layers leads to increased precision for both the standard and proposed fine-tuning approaches. Furthermore, note that for the two in-domain classes (i.e., "coffee cup" and "bird") overfitting phenomena arise more easily when optimizing more layers. For example, optimizing the last layers reduces the mAP from 0.759 to 0.740. On the other hand, using the proposed method in these cases allows for mitigating this effect. For example, in the same case, when the first and the last layers are optimized ("Proposed (L2)"), mAP rises to 0.782. Furthermore, we observe that in all evaluated cases using the proposed symmetric fine-tuning approaches leads to better results compared to fine-tuning the same number of layers at the end of the network. Also, we observe that the proposed method can achieve significantly better utilization of the layer budget since using just two layers can reach (or even exceed in the same cases), the baseline which optimizes four layers. This observation hints towards confirming our initial hypothesis that the first layers act as a bottleneck, reducing the fine-tuning accuracy.

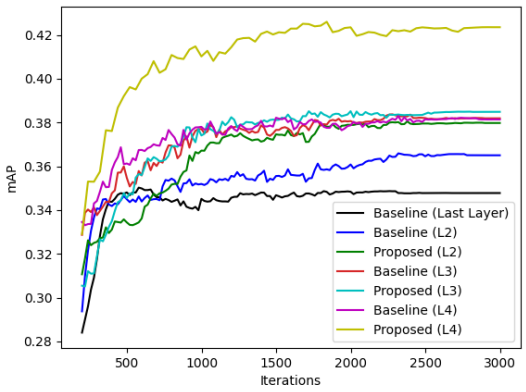
Furthermore, in Fig. 1 we provide the mAP over the whole fine-tuning process for the different approaches. First, we note the slower convergence for out-of-domain classes, compared to in-domain classes. Furthermore, again we confirm that the proposed symmetric fine-tuning approach leads to the overall best results, often outperforming other methods that use a significantly higher optimization budget. The difference between the proposed and the baseline approaches grows as the difficulty of the classes to be learned grows, e.g., for the 'helmet' class.

IV. Conclusions

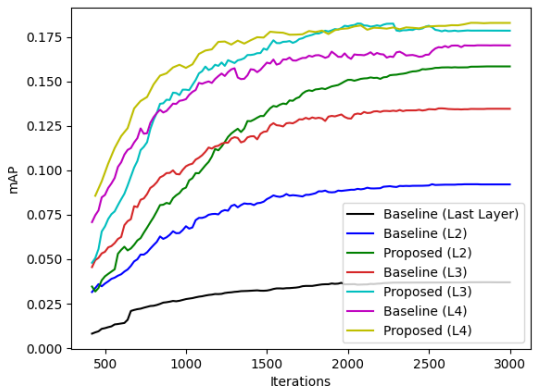
In this paper, we proposed a symmetric fine-tuning method for few-shot object detection that involves both the first and last layers of the detection head. This approach aims to maintain a fixed trainable parameter budget while strategically selecting parameters for fine-tuning. The proposed method follows findings that are supported by the data processing inequality, which states that information lost in early network layers cannot be recovered in subsequent layers. The experimental results demonstrate the effectiveness and efficiency of the proposed symmetric fine-tuning approach. By incorporating both the initial and final layers of the network in the fine-tuning process, we achieve improved accuracy and mitigate the risk of overfitting. This method provides a more comprehensive adaptation of the detection head for the few-shot object detection task, leveraging the benefits of



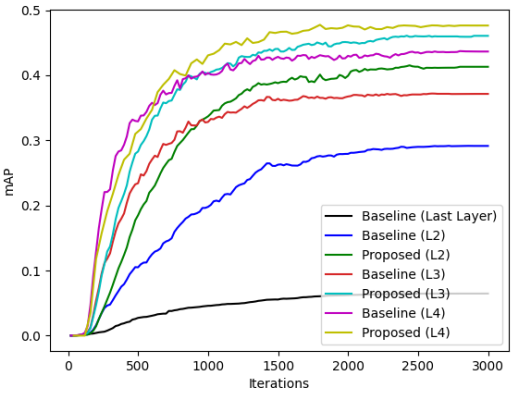
(a) mAP during fine-tuning for the flowers class



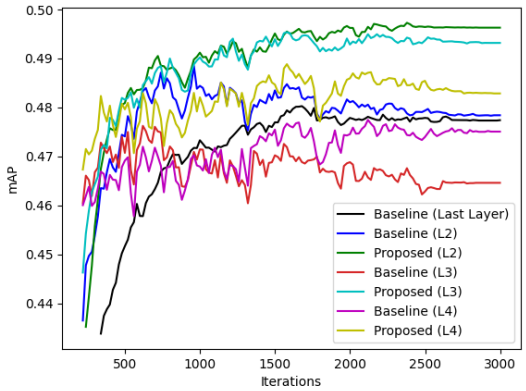
(b) mAP during fine-tuning for the fish class



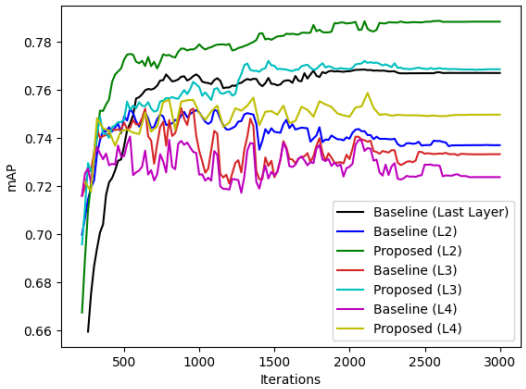
(c) mAP during fine-tuning for the helmets class



(d) mAP during fine-tuning for the tire class



(e) mAP during fine-tuning for the bird class



(f) mAP during fine-tuning for the coffee cup class

Fig. 1: mAP during training for out-of-domain (subfigures 1a, 1b, 1c and 1d), as well as for in-domain (subfigures 1e and 1f) classes.

pre-existing knowledge while allowing for effective learning of new object categories.

The findings of this study open up several interesting future research directions. First, future research can explore more sophisticated methods for determining the optimal trainable parameter budget. This could involve incorporating factors such as model complexity, dataset characteristics, or task-specific requirements to dynamically allocate resources and improve the overall performance of the fine-tuning process. Furthermore, investigating dynamic ways to select the most appropriate parameters for optimization based on information flow within the network holds significant potential. This approach would consider the relevance and impact of each layer or module on the overall performance and adaptively prioritize their fine-tuning. Such dynamic parameter selection methods could improve the efficiency and effectiveness of the fine-tuning process. Finally, exploring how the proposed findings and techniques in few-shot object detection can be extended and applied to generic optimization methods for deep neural networks is another intriguing research direction. This exploration could lead to advancements in optimizing deep neural networks across various domains and tasks.

ACKNOWLEDGMENT

This research was funded by the project "SEMANTIC ANNOTATION AND METADATA ENRICHMENT OF OPEN VIDEO STREAMS USING DEEP LEARNING" (Project code: KMP6-0079092) that was implemented under the framework of the Action "Investment Plans of Innovation" of the Operational Program "Central Macedonia 2014 2020", that is co-funded by the European Regional Development Fund and Greece.

REFERENCES

[1] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, 2016.

[3] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6569–6578, 2019.

[4] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," 2015.

[5] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2009.

[6] H. Su, X. Zhu, and S. Gong, "Open logo detection challenge," *arXiv preprint arXiv:1807.01964*, 2018.

[7] N. Passalis, M. Tzelepi, P. Charitidis, S. Doropoulos, S. Vologiannidis, and A. Tefas, "Deep video stream information analysis and retrieval: Challenges and opportunities," in *Proceedings of the IEEE International Conference on Multimedia Information Processing and Retrieval*, pp. 336–341, 2022.

[8] N. Passalis, S. Pedrazzi, R. Babuska, W. Burgard, D. Dias, F. Ferro, M. Gabbouj, O. Green, A. Iosifidis, E. Kayacan, *et al.*, "Opendr: An open toolkit for enabling high performance, low footprint deep learning for robotics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12479–12484, 2022.

[9] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–34, 2020.

[10] M. Köhler, M. Eisenbach, and H.-M. Gross, "Few-shot object detection: A comprehensive survey," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2023.

[11] K. Guirguis, A. Hendawy, G. Eskandar, M. Abdelsamad, M. Kayser, and J. Beyerer, "Cfa: Constraint-based finetuning approach for generalized few-shot object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4039–4049, June 2022.

[12] G. Han, J. Ma, S. Huang, L. Chen, and S.-F. Chang, "Few-shot object detection with fully cross-transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5321–5330, June 2022.

[13] H. Lee, M. Lee, and N. Kwak, "Few-shot object detection by attending to per-sample-prototype," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2445–2454, January 2022.

[14] X. Yan, Z. Chen, A. Xu, X. Wang, X. Liang, and L. Lin, "Meta r-cnn: Towards general solver for instance-level low-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[16] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proceedings of the IEEE Information Theory Workshop*, pp. 1–5, 2015.

[17] N. Passalis, M. Tzelepi, and A. Tefas, "Heterogeneous knowledge distillation using information flow modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[18] K. Lu, A. Grover, P. Abbeel, and I. Mordatch, "Pretrained transformers as universal computation engines," *arXiv preprint arXiv:2103.05247*, vol. 1, 2021.

[19] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, *et al.*, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1956–1981, 2020.

[20] H. Xu, X. Wang, F. Shao, B. Duan, and P. Zhang, "Few-shot object detection via sample processing," *IEEE Access*, vol. 9, pp. 29207–29221, 2021.