Tweaking EfficientDet for frugal training

Georgios Orfanidis
g.orfanidis@iti.gr
Information Technologies Institute - Centre of Research & Technology - Hellas
Thessaloniki, Greece

Konstantinos Ioannidis
Information Technologies Institute - Centre of Research & Technology - Hellas
Thessaloniki, Greece
kioannid@iti.gr

Anastasios Tefas
Department of School of Informatics - Aristotle University of Thessaloniki
Thessaloniki, Greece
tefas@csd.auth.gr

Stefanos Vrochidis
Information Technologies Institute - Centre of Research & Technology - Hellas
Thessaloniki, Greece
stefanos@iti.gr

Ioannis Kompatsiaris
Information Technologies Institute - Centre of Research & Technology - Hellas
Thessaloniki, Greece
ikom@iti.gr

ABSTRACT

Object detection appears to be omnipresent nowadays with detectors being available for every problem available, covering solutions from extra-light to ultra resource demanding models. Yet, the vast majority of these approaches are based on large datasets to provide the required feature diversity. This work focuses on object detection solutions which do not rely heavily on abundant training datasets but rather on medium-sized data collections. It uses Efficientdet object detector as base for the application of novel modifications which achieve better performance both in efficiency as well in effectiveness. The focus on medium-sized datasets aim at representing more commonplace datasets which can be accumulated and compiled with relative ease.

CCS CONCEPTS

• **Computer systems organization** → **Object Detection**; *Deep Learning*; • **Efficiency** → Lightweight models.

KEYWORDS

object detection, datasets, deep neural networks

ACM Reference Format:
Georgios Orfanidis, Konstantinos Ioannidis, Anastasios Tefas, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2023. Tweaking EfficientDet for frugal training. In *ICMR2023, June 12–15, 2023, Thessaloniki, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3591106.3592255

1 INTRODUCTION

Object detection is a fundamental task of computer vision which has demonstrated impressive performances after the introduction of deep learning techniques. The applicability of most works depends on the availability of relevant datasets. Contrary to this approach, this work focuses, mainly, on solutions which rely on medium-sized and easily compiled data collections, which are dominated by different principles than larger datasets. Additionally, our solution utilises modest hardware equipment to better demonstrate common place object detection use.

In order to support our case, we utilize state-of-the-art Efficientdet [26] model as base for our experiments. It includes a family of 7 detectors and uses as backbone the relevant models of EfficientNet [25]. Models of EfficientDet are named D0 to D7 with D0 representing the lightest version. The reported results on Efficientdet [26] focus solely on COCO dataset [16], a benchmark dataset nowadays, ignoring the behavior on smaller datasets. In our case, experiments were conducted on diverse datasets to showcase the robustness and applicability of the proposed modifications.

The modification introduced focus on various aspects of the detector. More specifically, they include a) a soft-anchor version for better localization of the detected boxes, b) a lighter unit for replacing part of the detector, and specifically, the Separable Convolutional Neural Unit, introduced in [8], c) a different approach for training procedure which attempts to exploit the diverse size of training instances and finally d) a more dynamic concatenation of the final layer in the detector head.

The rest of the paper is organised as follows: section 2 presents relevant previous works, the 3^{rd} section presents in more details the proposed method and its specific adaptations, section 4 contains the information regarding the datasets being used and the experimental results, while the paper is concluded with section 5 which summarizes the presented work.

2 RELATED WORK

Typically, object detectors are categorized by the utilization of a Region Proposal Network (RPN) [23] or not. Models using RPN are considered two-stage detectors like Fast RCNN [6], Faster RCNN [23], Cascade R-CNN [4] and Mask RCNN [7] while models not using RPN comprise one-stage detectors like Yolo [20], its successors [1, 21, 22] and SSD [17]. Research as in [9] has shown that two-stage detectors are often more accurate yet less efficient due to the introduction of the extra RPN submodel. One-stage detectors

ICMR2023, June 03–05, 2018, Thessaloniki, Greece

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/3591106.3592255

on the other hand are the most popular choice recently, due to their higher efficiency and simplicity. The majority of those also utilize predefined anchor boxes.

Nowadays, the community has special interest for efficient implementations, often being one-stage detectors like the aforementioned Yolo family which include smaller efficient models, anchor-free detectors as [14] and [28] or even model compression as [18] and special models for CPU computations as [10]. Several works have been proposed as improvements to the training procedure. One such is Freezout [3] which attempts to speed up training time by consecutively freezing layers or DSOD [24] uses densely connected layers instead of transfer learning. Finally, soft-anchor approach [29] focuses on the localization of the derived detected boxes using a weighted version of anchors.

3 PROPOSED METHOD

The architecture of Efficientdet is shown in Figure 1 with some of the proposed modifications also visible: Gradual training (section 3.3) and the weighted Concatenation approach (section 3.4). The remaining modifications are weighted Anchors method and Shufflenet unit insertion.

3.1 Weighted anchors

EfficientDet uses focal [15] and Huber [11] loss for classification and regression respectively. Additionally, a fundamental notion related to the latter loss calculation is the use of predefined boxes called anchors. Those boxes aim at providing the most representative box molds for actual annotated ground truth boxes. During training each ground truth box is compared against the predefined anchors to determine if it constitutes a positive or negative sample.

Two Intersection over Union (IoU) thresholds, one negative and one positive, are being used for this classification, with the negative one being the threshold for background and the positive one used for determining positive samples. Instead of the typical approach to simply utilize every positive sample as of equal importance we introduce a weight coefficient w to weigh each sample depending on its overlap with the ground truth box and use this weighted version on the loss function. We suspect that larger overlap produce samples of better quality and should be more extensively represented in the loss function.

3.2 Shufflenet units

Substitution of units with more efficient ones often help without sacrificing any value on the effectiveness level. Following such approach, we substituted Separable Convolutional Networks with Shuffle units [27] (Figure 1a). The actual Shuffle units used were taken from Thundernet model [19]. It should also be noted that the unit being replaced was depthwise Separable Convolutional Network [8] which was also introduced as an efficient alternative to the typical Convolutional Neural Networks.

3.3 Gradual training

A different approach has been adopted for the training of our models which separates training into 2 phases: training with smaller and larger objects. Features learned at each phase appear to has high-degree of independence, to activate different layers (Figure 1b) and thus, adopting such approach improves overall performance.

Table 1: Datasets statistic analysis

	Pascal Voc dataset			Nexet dataset		
set	aopi	# imgs	size	aopi	# imgs	size
train	2.85	16551	19.08%	2.73	39188	2.12%
train small	4.09	9005	11.01%	2.89	35547	1.64%
train large	1.38	7546	47.64%	1.13	3641	14.13%
aopi: average object per image, imgs: images, size: object to image size						

3.4 Weighted Concatenation

Class and Box prediction subnets typically include a concatenation step. We introduced a weighted version of such step (Figure 1c) which further improves performance.

4 EXPERIMENTAL RESULTS

4.1 Datasets

Pascal VOC [5] is a standard dataset for evaluating object detection with 20 classes of generic objects on traditional point of view. Nexet dataset [13] is a diverse dataset for autonomous driving with 5 classes which can be used for object detection. The dataset was randomly split into 2 sets: 80% (39188 samples) - 20% (9941 samples) for training and evaluation, respectively.

Table 1 presents an analysis of the two dataset where it can be seen that Nexet contains smaller objects in comparison to Pascal Voc. Furthermore, the separation into imagesets containing large and small objects respectively produces highly unbalanced imagesets in Nexet case and, thus, hinders the Gradual Training approach results.

4.2 Results

This section presents the results of the conducted experiments: Quantitative results for Pascal Voc and Nexet in Tables 2 and some qualitative results in Figure 2. Pascal Voc results seem to benefit for shallower model implementations, and thus, reducing the depth of the DBiFPN from 3 to 1 increased mAP by more than 2%. Applying the same decrease in Nexet did not contributed much when the smaller image size was used but on larger images the difference was annihilated.

For the training phase we opt for Adam optimizer [12] in place of SGD [2] in order to closely simulate the conditions this work is expected to be used. This work experimented with utilizing a frozen backbone (trained on ImageNet) for the initial steps of the training which proved beneficiary for the effectiveness of the model (Table 3). The initial learning rate was set to 10^{-3} for the entire frozen phase, while it was reduced by a factor of 10 at three steps during the last phase. Batch size was set to 32 for the frozen backbone phase(s) and 4 for the rest. Training and evaluation were performed on a GeForce RTX 3080 graphics card.

We utilize the two lightest model of EfficientDet family, $\phi 0$ and $\phi 1$, as milestone comparison models. There are 4+1 models presented in this paper with each one of these 4 incorporating a new

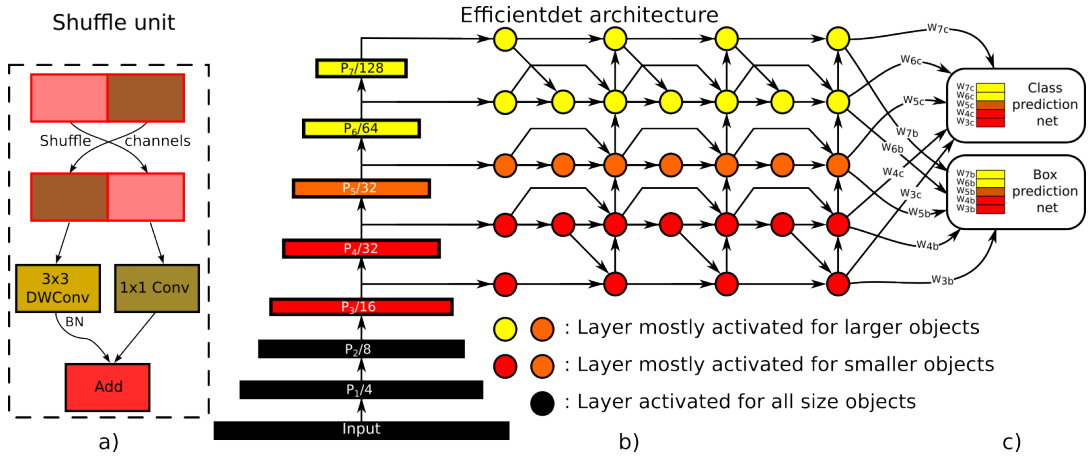

Figure 1: a) Shuffle unit b) Efficientdet architecture c) Weighted concatenation

Table 2: Object detection results on Pascal Voc and Nexet

Pascal Voc dataset Average precision results										
model	backbone ϕ	input size ϕ	DBiFPN	GT	wA	SU	wCon	mAP	Flops	fps
EfficientDet	0	0	3					75.24%	2.156B	65.0
EfficientDet lite	0	0	1					77.64%	2.093B	60.8
EfficientDet lite	0	0	1	✓				79.30%	2.093B	60.8
wEfficientDet lite	0	0	1	✓	✓			79.48%	2.093B	60.8
wShuffleEfficientDet lite	0	0	1	✓	✓	✓		79.72%	2.089B	57.0
wwShuffleEfficientDet lite	0	0	1	✓	✓	✓	✓	**80.35%**	2.089B	56.0
EfficientDet lite	0	1	1					77.93%	3.269B	59.3
EfficientDet lite	0	1	1	✓				79.93%	3.269B	59.3
wEfficientDet lite	0	1	1	✓	✓			79.95%	3.269B	59.3
wShuffleEfficientDet lite	0	1	1	✓	✓	✓		80.98%	3.262B	54.8
w2ShuffleefficientDet lite	0	1	1	✓	✓	✓	✓	**81.31%**	3.263B	53.3
EfficientDet	1	1	4					79.36%	5.242B	43.4
Nexet dataset Average precision results										
model	backbone ϕ	input size ϕ	DBiFPN	GT	wA	SU	wCon	mAP	Flops	fps
EfficientDet	0	0	3					53.38%	2.114B	61.3
EfficientDet	0	0	3					53.38%	2.114B	61.3
EfficientDet	0	0	3		✓			53.70%	2.114B	61.3
wShuffleefficientDet	0	0	3		✓	✓		53.92%	2.106B	57.6
wwShuffleEfficientDet	0	0	3		✓	✓	✓	**53.94%**	2.106B	56.5
EfficientDet	0	1	3					60.04%	3.300B	59.7
wEfficientDet	0	1	3		✓			60.52%	3.300B	59.7
wShuffleefficientDet	0	1	3		✓	✓		60.45%	3.289B	55.3
w2ShuffleefficientDet	0	1	3		✓	✓	✓	**60.72%**	3.289B	53.8
EfficientDet	1	1	3					61.58%	5.152B	43.8
DBiFPN: Depth Bidirectional FPN, GT: Gradual Training, wA: weighted Anchors SU: Shuffle Unit, wCon: weighted Concatenation, mAP: mean Average Precision										

improvement while the +1 model refers to a lighter version of Efficientdet with less depth. More specifically, *EfficientDet lite* reduces DBiFPN to 1, the next model introduces also Gradual Training during the initial phase of training, *wEfficientDet lite* introduces additionally Weighted Anchors, *wShuffleefficientDet* replaces Separable Convolutional layers with Shuffle Units (Figure 1), and finally, *w2ShuffleefficientDet* also adds weighted Concatenation prior to the Class and Box heads. The best performance is observed when all 4 improvements are included and it achieves 80.35% mAP for Pascal Voc with lesser Flops than the original Efficientdet ϕ0.

For Nexet the same set of improvement is applied to the baseline model, but, due to different dataset characteristics the inclusion

Table 3: Ablation study results on Pascal Voc and Nexet

dataset	model	mAP	model	wfunction	mAP	dataset order	mAP
Pascal Voc	entire model training	**75.24%**	EfficientDet lite	IoU	**79.48%**	small-large	**79.30%**
Pascal Voc	training on phases model	69.74%	EfficientDet lite	Centerness	79.04%	large-small	77.49%
dataset	model	GT	mAP	model	DBiFPN	input size ϕ	mAP
Nexet	EfficientDet	✓	53.06%	w2ShufflefficientDet	3	0	**53.92%**
Nexet	EfficientDet	–	**53.38%**	w2ShufflefficientDet	1	0	53.43%
Nexet	w2ShufflefficientDet	✓	53.43%	EfficientDet	3	1	**60.04%**
Nexet	w2ShufflefficientDet	–	**53.92%**	EfficientDet	1	1	60.01%

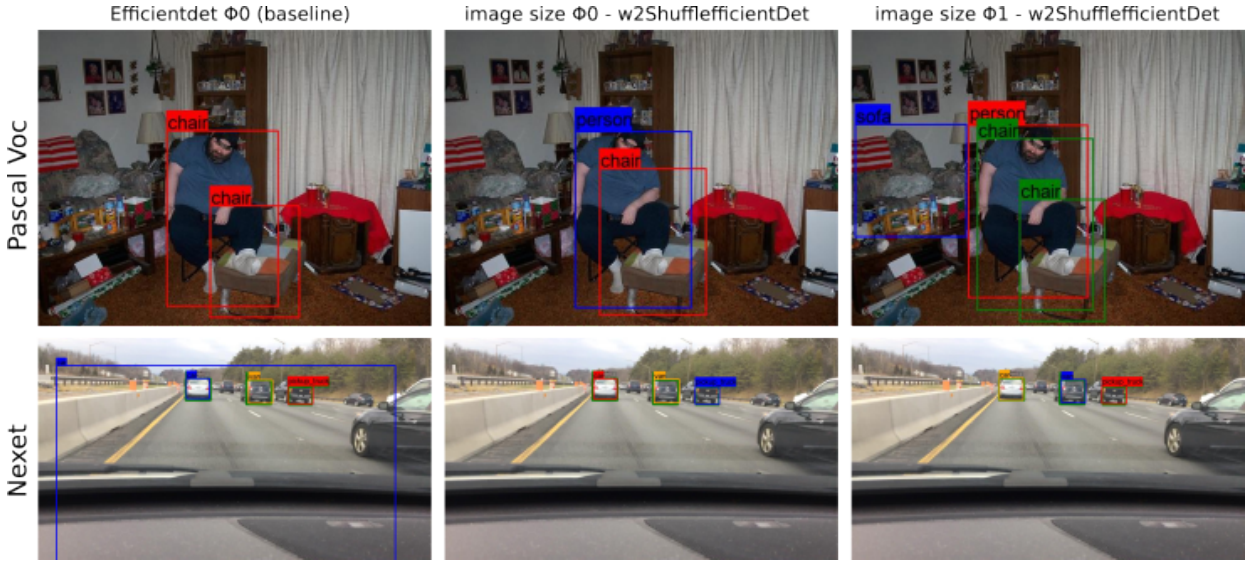

Figure 2: Pascal Voc and Nexet qualitative results

of certain of these modifications produce varying behavior. This was the case with the BiFPN depth reduction and the Gradual Training paradigms. Neither inclusion of those two innovations proved beneficiary for the model's performance. Regarding the impact of various improvement introduced, the two datasets exhibit different behavior. Pascal Voc takes advantage mostly from reduced model depth (+2.4%) as well as weighted Concatenation (+0.63%) while nexet mostly by using a bigger input image (+6.66%), possibly implying the initial object instances are too small (Table 1).

The inspiration for our proposed method of weighted anchors derived from [29] but we used a different weighting function. Our function is IoU based while the later relies on a generalized version of centerness function. Utilizing specific order for the partial datasets has also been investigated (Table 3). Also, for Nexet dataset we conducted a series of experiments to check if narrower models perform better as in Pascal Voc case. Although full model performed slightly better the actual difference was negligible when input image was larger (Table 3).

5 CONCLUSIONS

This work focuses on the special conditions training on medium-sized datasets imposes. More specifically, we examine a series of modifications which can be adapted in order to improve the performance of the trained model while maintaining the efficiency. Those improvements include a gradual training approach, an improved weighted boxes regression loss, the inclusion of more efficient units, like shuffle unit, and finally a dynamic concatenation approach for the detection head. We argue that the size of the dataset not only impacts the performance of the final model but also alters the conditions for the training phase. Thus, the approach suggested in this work focuses on exploiting those condition in our favor in order to obtain improved model performance. The base model used was Efficientdet while experiments were conducted on Pascal Voc and Nexet datasets.

ACKNOWLEDGMENTS

ISOLA and NESTOR projects have received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 883302 and 101021851, respectively. Content reflects only the authors' view and European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).

[2] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*. Springer, 177–186.

[3] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. 2017. Freeze-out: Accelerate training by progressively freezing layers. *arXiv preprint arXiv:1706.04983* (2017).

[4] Zhaowei Cai and Nuno Vasconcelos. 2018. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6154–6162.

[5] Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2015. The pascal visual object classes challenge: A retrospective. *International journal of computer vision* 111, 1 (2015), 98–136.

[6] Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 1440–1448.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[8] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[9] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. 2017. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7310–7311.

[10] Rachel Huang, Jonathan Pedoeem, and Cuixian Chen. 2018. YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2503–2510.

[11] Peter J Huber. 1992. Robust estimation of a location parameter. In *Breakthroughs in statistics*. Springer, 492–518.

[12] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[13] Itay Klein. 2017. *NEXET — The Largest and Most Diverse Road Dataset in the World*. Retrieved Jan, 2023 from https://www.kaggle.com/datasets/solesensei/nexet-original

[14] Hei Law and Jia Deng. 2018. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*. 734–750.

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[17] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*. Springer, 21–37.

[18] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. 2018. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270* (2018).

[19] Zheng Qin, Zeming Li , Zhaoning Zhang, Yiping Bao , Gang Yu, Yuxing Peng, and Jian Sun. 2019. ThunderNet: Towards real-time generic object detection on mobile devices. In *Proceedings of the IEEE/CVF international conference on computer vision*. IEEE/CVF, 6718–6727.

[20] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.

[21] Joseph Redmon and Ali Farhadi. 2017. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7263–7271.

[22] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).

[24] Zhiqiang Shen, Zhuang Liu, Jianguo Li, Yu-Gang Jiang, Yurong Chen, and Xiangyang Xue. 2017. Dsod: Learning deeply supervised object detectors from scratch. In *Proceedings of the IEEE international conference on computer vision*. 1919–1927.

[25] Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *International conference on machine learning*. PMLR, 6105–6114.

[26] Mingxing Tan, Ruoming Pang, and Quoc V Le. 2020. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10781–10790.

[27] Xiangyu Zhang, Xinyu Zhou, Mengxiao, Lin, and Jian Sun. 2018. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In *IEEE conference on computer vision and pattern recognition*.

[28] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. 2019. Objects as points. *arXiv preprint arXiv:1904.07850* (2019).

[29] Chenchen Zhu, Fangyi Chen, Zhiqiang Shen, and Marios Savvides. 2020. Soft anchor-point object detection. In *European conference on computer vision*. Springer, 91–107.