Human-machine collaboration on data annotation of images by semi-automatic labeling

Tom Haider
Florian Michahelles

ABSTRACT

Deployment of deep neural network architectures in computer vision applications requires labeled images which human workers create in a manual, cumbersome process of drawing bounding boxes and segmentation masks. In this work, we propose an image labeling companion that supports human workers to label images faster and more efficiently. Our data-pipeline utilizes One-Shot, Few-Shot and pre-trained object detection models to provide bounding box suggestions, thereby reducing the required user interactions during labeling to corrective adjustments. The resulting labels are then used to continuously update the underlying suggestion models. Optionally, we apply a refinement step, where an available bounding box is converted into a finer segmentation mask. We evaluate our approach with a group of participants who label images using our tool - both manually and with the system. In all our experiments, the achieved quality is consistently comparable with manually created labels at factor 2 to 6 faster execution times.

KEYWORDS

human-machine collaboration, data labeling, annotation

ACM Reference Format:
Tom Haider and Florian Michahelles. 2021. Human-machine collaboration on data annotation of images by semi-automatic labeling. In *Mensch und Computer 2021 (MuC '21), September 5–8, 2021, Ingolstadt, Germany*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3473856.3473993

1 INTRODUCTION

State of the art Object Classification or Detection algorithms [33, 37] using Deep Neural Networks require enormous amounts of labeled data to be trained. The availability of such datasets is however one of the biggest bottlenecks when deploying such algorithms in practice. Applicable public datasets are typically not available for very specific kinds of use cases (e.g. in the industry) and thus, datasets often need to be generated entirely from scratch. Traditionally, this requires human workers, that annotate images manually - an effort that is both expensive and time consuming. In this work we propose a data pipeline that supports human workers during annotation, by recognizing the relevant labeling targets automatically and suggesting where to place bounding boxes in correspondence to object classes. If target objects are novel, i.e. there is no large dataset at hand, which contains labeled instances of this object class, we apply a One-Shot or Few-Shot detection model, utilizing user created bounding boxes of previously labeled images. Optionally, we apply a refinement step where the bounding box is converted into a segmentation mask. We evaluate our system in a short user study, which shows the potential of accelerating labeling tasks.

Our contribution in this paper is to present a socio-technical system that enables faster human-machine collaborative labeling process than either machine or human could do separately on their own. We present data-labeling as a leading example for digital companions [22].

2 RELATED WORK

Arguably one of the most prominent use-cases where extensively annotated images are required is fully supervised object detection, which aims at simultaneous localization and classification of objects in a scene. Architectures like [13, 14, 25, 30–33, 37] are among the most well-known approaches to this problem.

Semantic segmentation and instance segmentation naturally extend this setting to the pixel level and require even more sophisticated image labels. The deeplab family, e.g. [9–12] is a popular series of models for semantic segmentation. Mask R-CNN [15], an extension of Faster R-CNN [33] is one of the most popular models for instance segmentation. All these models require a an enormous amount of training data. Therefore a variety of less data-demanding methods has been proposed.

[17] and [29] tackle the problem of One-Shot object detection. In this scenario, abundant instances of some base classes are available for training, though only a single instance for novel classes. A slightly relaxed version of this problem is addressed in [20, 21, 38, 39] with Few-Shot object detection. The goal in this scenario is reaching higher precision levels compared to the One-Shot setting.

[27] attempts One-Shot instance segmentation. To the best of our knowledge, this is the only work approaching this problem and it strongly suffers from a high number of false positives.

In Interactive Image Segmentation, coarse inputs provided by an annotator are used to generate an annotation mask around objects. A human user starts by providing some input (e.g. key points/regions), the model produces a suggestion (a first segmentation mask) and the user can then provide additional inputs based on this suggestion, [6, 8, 18, 24, 26, 35] are prominent architectures in this field.

Finally, tools as LabelMe [34] allow for the manual creation of both rectangle and polygon annotations. Similarly, LabelImg [2] is a popular GUI for annotating images in the common PASCAL VOC [4] or YOLO [5] format. Both tools, however, require an annotator to label the images in a complete manual fashion. CVAT [3] already incorporates automatic functionalities for the creation of annotations using pre-trained detection or segmentation models. COCO

MuC '21, September 5–8, 2021, Ingolstadt, Germany
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8645-6/21/09…$15.00
https://doi.org/10.1145/3473856.3473993

Annotator [1] is a web-based image annotation tool that also allows for the use of pertained Models to automatically create annotations and furthermore also utilizes DEXTR [26] for interactive image segmentation. Google fluid [7] aims at full image annotation in a single pass instead of using a series of micro-tasks such as indicating object presence in an image, clicking on instances of a specific class, or drawing polygons/boxes around instances. Instead, a pre-trained Mask R-CNN is used to provide an initial segmentation mask for each object in the scene. The user can then add or delete segments by left/right clicking on the corresponding areas.

We conclude that sophisticated object recognition models perform exceedingly well follwing a supervised learning paradigm. Additionally, a variety of feature-rich tools has been developed for manual image labeling. Some of these tools even allow to apply pre-trained detection models to automatically label images. Our approach shares similarities with these tools. However, the integration of One-Shot/Few-Shot detection models into a dedicated labeling application has not yet been explored. In this work we want to overcome the current weaknesses of One-Shot/Few-Shot detection models with a few, very targeted user inputs. By retraining these models as more and more labeled images become available, we tackle the human-in-the-loop scenario, not yet addressed in previously proposed labeling tools.

3 THE DATA LABELING COMPANION

In many datasets, objects to be labeled are very similar. Thus, the today's process of human annotators repetitively labeling the same or similar objects could be accelerated and assisted by a system making suggestions based on previous user inputs. Then, the user would rather choose between accepting the suggestion and making an adjustment alleviating some of the repetitiveness of the task. The question therefore is, whether object detection/segmentation models can serve this purpose and provide such assistance, whilst also being able to improve their level of suggestions with more user input becoming available.

In section 2 we reviewed approaches of object detection in the fully supervised setting, the one-/Few-Shot setting, the weakly supervised and the unsupervised learning setting. We found that even with small amounts of labeled data, models like [20, 38, 39] are already quite accurate in predicting the rough position of an object, but struggle to infer the accurate outline of objects (i.e., accurate bounding box dimensions). One-Shot/Few-Shot object detection models could therefore be used to provide an initial bounding box suggestion that only needs to be adjusted by the user. On the other hand, models trained in a fully supervised fashion can also accurately predict the spacial dimensions of objects classes they were trained on. We therefore combine the strengths of these two worlds, using pretrained models for object classes where labeled images already exist (e.g. from a public dataset) and One-Shot/Few-Shot object detection models where labeled data is absent, i.e. the objects are completely unknown/novel. This allows us to derive the following general annotation pipeline (see Figure 1)

(0) **Provide Information on classes in Dataset.** The human annotator starts providing a list of all the classes they expect to label in the dataset. Each class can thus be classified as known or unknow. Then, for each image in the dataset, the process is as follows:

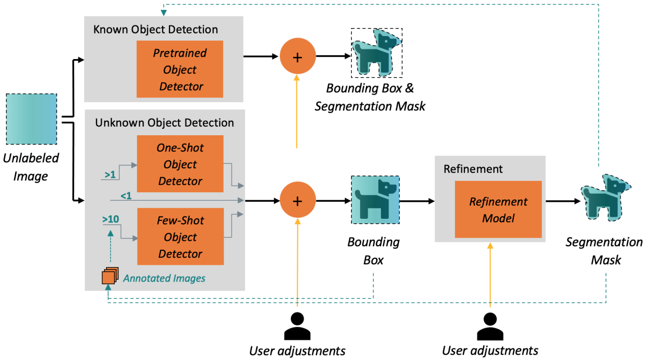

Figure 1: Annotation pipeline - process of labeling an image

(a) **Known object detection:** If the current object at interest is among the list of known objects, the supervised object detector module is applied. That is, we use existing or pretrained detection/segmentation models which localize and detect the objects (either pixel or object level) and provide detected class labels.

(b) **User Adjustments:** The user/annotator is provided with a suggestion on a bounding box or a segmentation mask, which they can accept, decline or adjust. The output of this phase is the final annotation of the current image for the current class.

(c) **Unknown object detection:** If the current class is not recognized as a known object we apply our unknown-object detector module, which consists of two major components: a One-Shot object detector and a Few-Shot object detector.

(d) **One-Shot Object Detector:** The One-Shot detector takes a single reference image of an object and looks for similar features in the query image. If no reference image of the object of interest exists, the user has to provide an initial bounding box of that object. This bounding box then serves as a reference for future objects of that class. Once there are one or more reference images of an object class available, the One-Shot detector can provide the user with a suggestion of a bounding box.

(e) **Few-Shot Object Detector:** For the Few-Shot object detector, 10 or more reference images of an object are used to train a network for detecting objects of the given classes. That is, as soon as there are enough reference images of an object class available, a Few-Shot object detection model can be trained. This model should ideally be capable of giving more accurate suggestions on bounding boxes than the One-Shot module.

(f) **User Adjustment:** The output of either the One-Shot detector or the Few-Shot detector is provided to the user as a bounding box suggestion, which they can accept decline or adjust. The output of this phase is the final bounding box label of the current image for the current object class.

(g) **Refinement:** After defining the bounding box for an object, it can be further processed into a segmentation mask, using the refinement module. The refinement module uses some additional user input to regress from a coarse bounding box of an object to a finer polygon i.e. segmentation mask. This the final label of the current image for the current class.

(h) **Learning:** The final segmentation masks or bounding boxes of an annotated image are then used as reference images for future query images – either to train the few shot detector or provide a ground truth for the One-Shot detector.

(i) **Model Migration:** Once a sufficient amount of unknown object instances has been labeled in the described manner, the provided image labels may be used to train an object detection/segmentation model in a fully supervised setting.

4 IMPLEMENTATION

The front-end of the proposed image labeling companion is built on-top of the web-based COCO Annotator [1] and is hosted on a web-server. The backend modules (Known Object detection, One-Shot Object Detection, Few-Shot Object Detection and Refinement) are each encapsulated in an independent web-service container and are hosted on a GPU supported server. The Known object detection module is implemented as Mask R-CNN [14] with a ResNet-101-FPN [16] as backbone, pre-trained on the 80 classes of the MS COCO [23] dataset. As One-Shot Object Detector for novel objects, a Siamese Mask R-CNN [28] network with a ResNet-50 as backbone network is used. Once several reference images of an object (e.g. 10 or more) are annotated by the user (either fully manually or by support of the One-Shot model), the now available labels for unknown classes are used to fine-tune a Few-Shot detection model, following [19]. Finally, to refine from bounding boxes to segmentation masks, we follow the interactive segmentation approach from [36], running inference only on a region of interest centered around the previously generated bounding box. All models can be continuously updated by adding the newly annotated images to the datasets used for training/fine-tuning.

5 EVALUATION

5.1 Experiment Setup

To quantify the effectiveness of our system, we carried out a within-subjects user study with 15 participants creating annotations in different scenarios. We measured annotation time and quality of the created annotations using Intersection over Union (IoU) with existing groundtruth labels.

We compared between manual labeling and semi-automatic labeling with the help of our suggestion backend. To ablate the effect of each of the deployed models in the suggestion backend, we differentiated between *known* and *unknown* object classes during the evaluation. Since we considered both bounding boxes and segmentation masks, this resulted in a total of six different annotation tasks, each carried out on a series of images containing multiple object classes: (1) Manually draw a bounding box around known and unknown objects; (2) manually draw a polygon around known and unknown objects; (3) generate a bounding-box around known objects, based on a proposal by the many-shot detector (pre-trained detection model); (4) generate a bounding-box around unknown objects, based on a proposal by the One-Shot detector; (5) generate a bounding box around unknown objects, based on a proposal by the Few-Shot detector; (6) generate a bounding box around both known and unknown objects, and then refine each bounding box to a segmentation mask using the interactive segmentation tool.

We led each participant through the study using screen sharing and the remote-control feature of Microsoft Teams. The participants also had the chance to familiarize themselves with our tool on a set of examples prior to each of the annotation tasks.

5.2 Dataset

The dataset used in this study consists of a total of four classes, two of which are *known* and two are *unknown*. For known object categories, we selected two classes contained in the MS COCO 2017 dataset: *airplane* and *dog*. As for *unknown* classes, we directly wanted to test the functionality of our tool in an industrial context. Therefore we combined two datasets that were deployed in previous projects of our research group: The *valve* datasets and the *tools* dataset. The valve dataset comprises only of one class: *valves*. The *tool* dataset contains pictures of a variety of tools. For this user study, however, we choose to only consider the *flashlight* class and selected only images from this datasets that contain such. All other objects contained in the selected images are treated as background.

As reference/training images for the One-Shot and the Few-Shot detection model, we chose images distinct from the ones selected for evaluation. In the One-Shot setting, a single reference image per class were used. In the Few-Shot setting, we trained the model with 10 reference images for each novel class.

For a better comparison, the labels for the One-Shot and the Few-Shot model were created prior to the study by the author, and the Few-Shot model was pre-trained with these labels. The different annotation tasks represent the independent variables of our user study. We select two classes for both known (airplane and dog form MS COCO 2017 dataset) and unknown object classes (valve and flashlight) from a proprietary dataset.

5.3 Participants

In total, 15 participants (5 Female, 10 Male, between the age of 21 and 38) took part in the study. Only two participants had no prior experience with labeled images, five indicated sparse experience, six had some prior experience and two participants frequent interaction with labeled images. Most of the participants never labeled images themselves before, one has labeled images frequently, two have done so sometimes, and another two very rarely.

6 RESULTS

6.1 Bounding Box Annotations

Figure 2 shows on the top line the time and annotation quality for creating bounding boxes. As less user interactions are required on average, when bounding boxes are created based on a relatively accurate proposal,the inner-class variance of the annotation time is greatly reduced for bounding-box generation by the many-shot or Few-Shot model. The quality of the generated bounding boxes by the One-Shot model is not as high, and therefore this effect is weaker in that case. In terms of mean IoU, our proposed method is on par with manual annotations. Annotations supported with proposals by Mask-RCNN, Siamese Mask-RCNN or FSRW lead to a mean IoU of 94%, 92% and 93%, which is slightly higher than the 91% obtained with purely manual labeling. Most notably, the total time per annotation is dependent on the quality of the initial bounding box proposal. With Mask R-CNN as proposal generator,

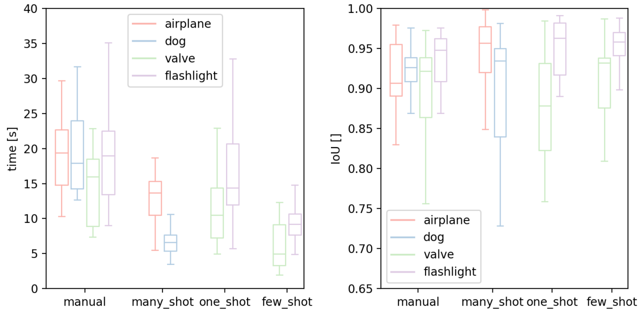

Figure 2: Time & quality of bounding-box annotations.

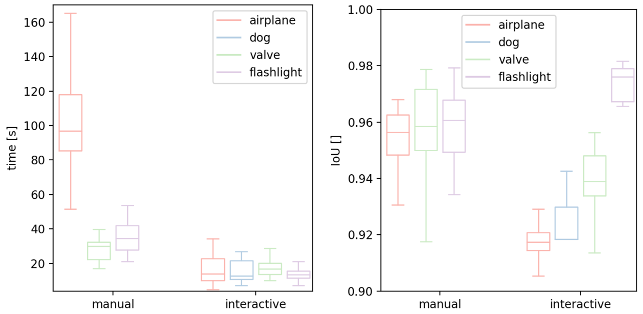

Figure 3: Time & quality of segmentation mask annotations.

annotation time is almost halved compared to manual labeling (~18.4s vs. ~10.0s mean annotation time per label). With FSRW, annotation time is reduced by factor of 2.4 (~7.7s mean ann. time p. label). Annotation takes less time with FSRW than Mask R-CNN because the imagery of the used valves contains more than one object instance. This impacts the total annotation time and therefore a direct comparison of Mask R-CNN and FSRW is not appropriate. We can however directly compare Siamese Mask R-CNN with FSRW, as they are tested on the same object classes. The latter leads to significantly shorter annotation times, while providing the same level of annotation quality. Hence, we can conclude, the better the proposal by the model, the less interactions are required by the annotator and the less time it takes to reach the final bounding box annotation. We can therefore support our hypothesis, that many-shot/One-Shot/Few-Shot object detection models can be deployed to speed up the process of annotation images with bounding boxes, while maintaining an annotation quality comparable to complete manual annotation.

6.2 Segmentation Mask Annotations

Compared to manual annotation our proposed method for generating segmentation masks shows significant annotation time improvements over all measured classes. The corresponding results are depicted in Figure 3 at the bottom. For the manual case the results are dependent on the object classes due to the individual shape of the objects, e.g. airplanes are large with many details to be considered when fitting polygons. In contrast, valves or flashlights can be encompassed by polygons with fewer points. Clearly, the annotation time per instance is linearly dependent on the number of polygon-points and thus the more complex objects are, the longer the annotation time when manually drawing segmentation masks. This dependence cannot be observed with our proposed system as with f-brs, polygons no longer need to be drawn edge by edge. Instead, entire regions are added or removed with positive and negative clicks respectively.

Our proposed method yields a slightly reduced IoU with ground truth for the classes airplane and valve and a slightly increased IoU for the class flashlight. These differences are however not significant. Our proposed method leads to a comparable annotation quality while requiring significantly less annotation time, which is depicted in figure 3. For fully manual labeling, participants needed on average ~55.5s per annotation. With the proposed systems this time was reduced to ~16.3s on average. This supports our hypothesis that many-shot/Few-Shot object detection models in combination with interactive segmentation models can be deployed to speed up the process of annotating images with segmentation masks, while maintaining an annotation quality comparable to complete manual annotation.

7 CONCLUSIONS AND FUTURE WORK

We presented the digital labeling companion, an intelligent labeling tool that combines the strengths of human annotators and powerful object detection models. Especially for novel object classes with individual instances that are fairly similar to each other, our method consistently leads to significant time improvement over manual labeling. In very difficult scenarios, where individual instances of the same object class are vastly different from each other or where only very few training examples exist, the deployed detection models struggle to make accurate predictions. This directly affects the performance of our system. In such cases extensive human labor is still inevitable. In many cases however, our system is already able to provide suggestions that are accurate enough to substantially speed up the process of labeling images, while maintaining a high level of annotation quality. We see data labeling as a successful example of leveraging both human and machine capabilities for a more efficient outcome. We want to motivate human-machine interaction researchers to dig further into what is it the machine can do, what is it the human can do, and how can both be more successful together.

We want to acknowledge the limited generalization capabilities of the conducted user study. To formulate the presented results in a more generic way, a more versatile dataset would be required, as well as more participants. It is therefore difficult to directly compare our results to previous work. As for now, we only compare our solution to the manual annotation process and leave an in depth-comparison to existing tools such as [3] or [34] to future work. We also want to point out the limitations of the deployed models. The presented architecture is not a unified end-to-end architecture but rather a novel and modular combination of existing models from the presented related fields. Thus, the final performance of our system is heavily reliant on the models used in each of the components.

REFERENCES

[1] [n.d.]. Cocoannotator, Github. https://github.com/jsbroks/coco-annotator, last accessed on 05/04/2021.

[2] [n.d.]. labelImg. https://github.com/tzutalin/labelImg, last accessed on 05/02/2021.

[3] [n.d.]. Opencv / cvat. https://github.com/opencv/cvat/, last accessed on 05/04/2021.

[4] [n.d.]. The PASCAL Visual Object Classes Homepage. http://host.robots.ox.ac.uk/pascal/VOC/, last accessed on 05/05/2021.

[5] [n.d.]. YOLO: Real Time Object Detection, Github. https://github.com/pjreddie/darknet/wiki/YOLO:-Real-Time-Object-Detection, last accessed on 05/04/2021.

[6] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. 2018. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 859–868.

[7] Mykhaylo Andriluka, Jasper RR Uijlings, and Vittorio Ferrari. 2018. Fluid annotation: a human-machine collaboration interface for full image annotation. In *Proceedings of the 26th ACM international conference on Multimedia*. 1957–1966.

[8] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. 2017. Annotating object instances with a polygon-rnn. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5230–5238.

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062* (2014).

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* 40, 4 (2017), 834–848.

[11] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* (2017).

[12] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*. 801–818.

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

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 580–587.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. *CoRR* abs/1703.06870 (2017). arXiv:1703.06870 http://arxiv.org/abs/1703.06870

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). arXiv:1512.03385 http://arxiv.org/abs/1512.03385

[17] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. 2019. One-shot object detection with co-attention and co-excitation. In *Advances in Neural Information Processing Systems*. 2725–2734.

[18] Won-Dong Jang and Chang-Su Kim. 2019. Interactive image segmentation via backpropagating refinement scheme. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5297–5306.

[19] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. 2018. Few-shot Object Detection via Feature Reweighting. *CoRR* abs/1812.01866 (2018). arXiv:1812.01866 http://arxiv.org/abs/1812.01866

[20] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. 2019. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE International Conference on Computer Vision*. 8420–8429.

[21] Leonid Karlinsky, Joseph Shtok, Sivan Harary, Eli Schwartz, Amit Aides, Rogerio Feris, Raja Giryes, and Alex M Bronstein. 2019. Repmet: Representative-based metric learning for classification and few-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5197–5206.

[22] Mareike Kritzler, Jack Hodges, Dan Yu, Kimberly Garcia, Hemant Shukla, and Florian Michahelles. 2019. Digital Companion for Industry. In *Companion Proceedings of The 2019 World Wide Web Conference* (San Francisco, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 663–667. https://doi.org/10.1145/3308560.3316510

[23] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. *CoRR* abs/1405.0312 (2014). arXiv:1405.0312 http://arxiv.org/abs/1405.0312

[24] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. 2019. Fast interactive object annotation with curve-gcn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5257–5266.

[25] 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.

[26] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. 2017. Deep Extreme Cut: From Extreme Points to Object Segmentation. *CoRR* abs/1711.09081 (2017). arXiv:1711.09081 http://arxiv.org/abs/1711.09081

[27] Claudio Michaelis, Ivan Ustyuzhaninov, Matthias Bethge, and Alexander S Ecker. 2018. One-shot instance segmentation. *arXiv preprint arXiv:1811.11507* (2018).

[28] Claudio Michaelis, Ivan Ustyuzhaninov, Matthias Bethge, and Alexander S. Ecker. 2018. One-Shot Instance Segmentation. *CoRR* abs/1811.11507 (2018). arXiv:1811.11507 http://arxiv.org/abs/1811.11507

[29] Anton Osokin, Denis Sumin, and Vasily Lomakin. 2020. OS2D: One-Stage One-Shot Object Detection by Matching Anchor Features. *arXiv preprint arXiv:2003.06800* (2020).

[30] 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.

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

[32] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *arXiv* (2018).

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.

[34] Bryan C. Russell, A. Torralba, K. Murphy, and W. Freeman. 2007. LabelMe: A Database and Web-Based Tool for Image Annotation. *International Journal of Computer Vision* 77 (2007), 157–173.

[35] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. 2020. f-BRS: Rethinking Backpropagating Refinement for Interactive Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8623–8632.

[36] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. 2020. f-BRS: Rethinking Backpropagating Refinement for Interactive Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8623–8632.

[37] 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.

[38] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. 2020. Frustratingly Simple Few-Shot Object Detection. *arXiv preprint arXiv:2003.06957* (2020).

[39] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. 2019. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 9577–9586.