Coral Reef annotation, localisation and pixel-wise classification using Mask R-CNN and Bag of Tricks

Lukáš Picek[1,5], Antonín Říha[2], and Aleš Zita[3,4]

[1] Dept. of Cybernetics, Faculty of Applied Sciences, University of West Bohemia
[2] Faculty of Information Technology, Czech Technical University
[3] The Czech Academy of Sciences, Institute of Information Theory and Automation
[4] Faculty of Mathematics and Physics, Charles University
[5] PiVa AI

Abstract. This article describes an automatic system for detection, classification and segmentation of individual coral substrates in underwater images. The proposed system achieved the best performances in both tasks of the second edition of the ImageCLEFcoral competition. Specifically, mean average precision with Intersection over Union (IoU) greater then 0.5 (mAP@0.5) of 0.582 in case of Coral reef image annotation and localisation, and mAP@0.5 of 0.678 in Coral reef image pixel-wise parsing. The system is based on Mask R-CNN object detection and instance segmentation framework boosted by advanced training strategies, pseudo-labeling, test-time augmentations, and Accumulated Gradient Normalisation. To support future research, code has been made available at: https://github.com/picekl/ImageCLEF2020-DrawnUI.

Keywords: Deep Learning, Computer Vision, Instance Segmentation, Convolutional Neural Networks, Machine Learning, Object Detection, Corals, Biodiversity, Conservation

1 Introduction

The ImageCLEFcoral [4] challenge was organized in conjunction with the ImageCLEF 2020 evaluation campaign [12] at the Conference and Labs of the Evaluation Forum (CLEF[1]). The main goal for this competition was to create such an algorithm or system that can automatically detect and annotate a variety of benthic substrate types over image collections taken from multiple coral reefs as part of a coral reef monitoring project with the Marine Technology Research Unit at the University of Essex.

[1] http://www.clef-initiative.eu/

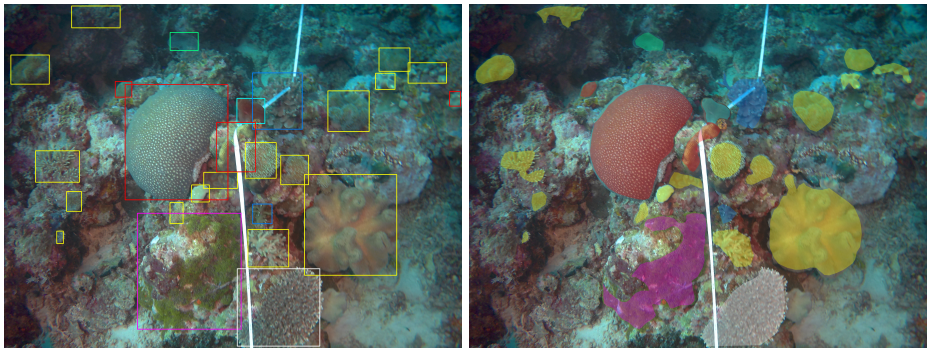

Fig. 1. Example training images showing different types of annotations - Bounding Boxes and Segmentation Masks. Every colour represents one substrate type, e.g. yellow represents *Soft Coral* and red belongs to *Hard Coral Boulder*.

1.1 Motivation

Live corals are an important biological class that has a massive contribution to the ocean ecosystem biodiversity. Corals are key habitat for thousands of marine species [5] and provide an essential source of nutrition and yield for people in the developing countries [3,2]. Therefore, automatic monitoring of coral reefs condition plays a crucial part in understanding future threats and prioritizing conservation efforts.

1.2 Datasets

This section will briefly describe the provided data and their subsets: an annotated dataset that contains 440 images, and a testing dataset with 400 images without annotations. Additionally, we introduce an precisely engineered training/validation split of the annotated dataset for the training purposes.

Annotated dataset - The annotated dataset is a combination of 440 images containing 12,082 individual coral objects. Each coral was annotated with expert level knowledge, including segmentation mask, bounding box, and class that represents 1 out of 13 substrate types. The dataset is heavily unbalanced (refer to Table 1), having almost 50% of objects from a single class (Soft Coral) and approximately 8% for the eight least frequent classes. Moreover, images have different colour variations, are heavily blurred, and came from different locations and geographical regions. Furthermore, coral substrates belonging to the same class can be observed in different morphology, colour variations, or patterns. Finally, some images contain a measurement tape that partially covers objects of interest.

For the network training process evaluation, the annotated dataset needed to be divided into two parts. One used for network optimization and the second for

network performance validation. To create these subsets, every tenth image was designated for validation set, the rest was used for training. As the validation set class distribution did not match the training one, particular images from the validation set needed to be replaced by carefully cherry-picked images from the training set. This resulted in an almost perfect split with similar distributions for both, the training and the validation set. This similarity ensured a representative validation process.

Testing dataset - The testing dataset contains 400 images from four different locations. Namely, the same location as is in the training set, similar location to the training set, geographically similar location to the training set, and geographically distinct location from the training set.

Table 1. Dataset class distribution including training and validation split description. 396 images were used for training; 44 for validation.

Dataset distribution			**Train. / Val. split**	
Substrate type	# Bboxes	Fraction [%]	Train. Boxes	Val. Boxes
Soft Coral	5,663	46.87	5,035	628
Sponge	1,691	13.99	1,472	219
Hard Coral – Boulder	1,642	13.59	1,513	129
Hard Coral – Branching	1,181	9.774	1,084	97
Hard Coral – Encrusting	946	7.829	831	115
Hard Coral – Mushroom	223	1.845	199	24
Hard Coral – Submassive	198	1.845	162	36
Hard Coral – Foliose	177	1.464	144	33
Sponge – Barrel	139	1.150	124	15
Algae - Macro or Leaves.	92	0.761	81	11
Soft Coral – Gorgonian	90	0.745	70	20
Hard Coral – Table	21	0.175	17	4
Fire Coral – Millepora	19	0.157	15	4

1.3 The System

The proposed object detection and instance segmentation system extends recent state-of-the-art Convolutional Neural Network (CNN) object detection framework (Mask R-CNN [8]) with additional **Bag of Tricks** that considerably increased the performance. The TensorFlow Object Detection API[2] [11] was used as a deep learning framework for fine-tuning the publicly available checkpoints. All bells and whistles are further described in Section 2. Additionally, approaches that did not contribute positively but could have some potential for future editions of the ImageCLEFcoral competition are discussed.

[2] https://github.com/tensorflow/models/blob/master/research/object_detection

2 Methodology

This section describes all approaches and techniques used in the benthic substrate detection, annotation and segmentation tasks. The modern object detection and instance segmentation methods are summarized, followed by the description of the chosen system and its configuration. Furthermore, all the used bells and whistles (Bag of Tricks) are introduced and described.

2.1 Object Detection

Although conventional digital image processing methods are capable of detecting particular local features, modern object detectors based on Deep Convolutional Neural Networks (DCNN) achieve superior performance in object detection and instance segmentation tasks. Several network architectures were pre-selected based on study published by Huang et al. [11], namely the Faster R-CNN [18], SSD [15] and Mask R-CNN [8]. The initial performance experiment was to train these detection frameworks with default or recommended configurations. This experiment revealed the most suitable framework for both the tasks within the ImageCLEFcoral competition - the Mask R-CNN.

2.2 Network parameters

Experiments on the validation set, reveled the best optimizer settings for the framework. These settings were shared between all of our experiments, unless stated otherwise. For detailed description refer to Table 2.

Table 2. Training and network parameters shared among all experiments.

Parameter	Value	Parameter	Value
Optimizer	RMSprop	Gradient Clipping	12.5
Momentum	0.9	Input size	1000×1000
Initial and min LR	0.032 - 0.00004	Feature extractor stride	8
LR decay type	Exponential	Pretrained Checkpoints	COCO
LR decay factor	0.975	Num epochs	50
Batch size	1	Gradient accumulation	16

2.3 Bag of Tricks

Augmentations - The provided dataset contains 440 images. Considering that 44 were used for validation, 396 images is too few for robust network optimization. To alleviate this issue, multiple data augmentation techniques were utilized. The following methods were included in the final training pipeline:

Colour Distortions - Brightness variations with max delta of 0.2, contrast and saturation variations scale each by random value in range of 0.8 - 1.25, hue variations offsets by random value of up to 0.02, and random RGB to grayscale conversion with 10% probability.

Image Flips - Random horizontal and vertical flip, and 90 degree rotations. Each with 50% chance.

Random Jitter - Every bounding box corner can be randomly shifted by amount corresponding up to 2% of the bounding box width and height in x and y coordinates, respectively.

Cut Out [6] - Random black square patches are added into the image. More precisely, add up to 10 patches with 50% occurrence probability and each with side length corresponding to 10% of the image height or width, whichever is smaller.

By utilizing techniques mentioned above, we have increased the model mAP@0.5 performance by **0.0392** as measured on the validation set.

Input Resolution - In the task of object detection, primarily where a small object occurs, input resolution plays a crucial role. Theoretically, the higher the resolution is, the more objects will be detected. Unfortunately, the detection of high resolution images is GPU memory-limited. Hence, it always is a trade-off between performance and hardware requirements.

Backbone - To find the best backbone architecture for Mask R-CNN framework. We performed an experiment over 3 different backbone models including ResNet-50 [9], ResNet-101 [9], and Inception-ResNet-V2 [20]. Detailed performance comparison is included in Table 3.

Table 3. Effect of input resolution and backbone architecture on model performance.

Backbone	Input Resolution	mAP@0.5	mAP@0.75
ResNet-50	600×600	0.1826	0.0956
ResNet-50	800×800	0.2077	0.1017
ResNet-50	1000×1000	0.2227	0.1260
ResNet-50	1200×1200	**0.2380**	**0.1579**
ResNet-101	800×800	**0.2381**	**0.1453**
Inception-ResNet-V2	800×800	0.2362	0.1361

Pseudo Labels - Performance of DCNN's heavily depends on the size of the training set. To facilitate this issue, we have developed a naive pseudo-labelling approach inspired by [1]. In short, already trained network is used to label the unlabelled testing data with so-called weak labels. Only the overconfident detections were used; the rest of the image was blurred out. Even though there is a high chance of overfitting to incorrect pseudo-labels due to the confirmation bias, pseudo-labels can significantly improve the performance of the CNN if pseudo-labelled images are added sensitively.

Transfer Learning - Big-transfer [13] or transfer learning is a fine-tuning technique commonly used in deep learning. Rather then initialize the weights of neural network randomly, pretrained weights are used. Furthermore, final model could benefit from similar domain weights. To evaluate a potential of such approach for the purposes of this competition, we experimented with fine-tuning of the publicly available checkpoints, including ImageNet[3], iNaturalist[3], COCO [14], PlantCLEF2018 [19] and PlanCLEF2019 [17]. The idea was that fine-tuning checkpoints trained on nature-oriented datasets would outperform the non-nature oriented ones. One could assume, that this is caused by significant difference when compared to other domains. Based on that it has been decided to use the COCO pretrained checkpoint which includes both the backbone and region proposed weights.

Table 4. Transfer Learning experiment - Effect of pretrained weights on model performance. For this experiment, the Mask R-CNN with ResNet-50 backbone and input size of 800×800 was used.

Pretrained weights	mAP@0.5	mAP@0.75
ImageNet (only backbone)	0.1826	0.0956
COCO (All Mask R-CNN weights)	0.2077	0.1017
iNaturalist (only backbone)	0.2091	0.0854
PlantCLEF2018 (only backbone)	0.1991	0.0914
PlantCLEF2019 (only backbone)	0.1895	0.0932

Test Time Augmentations - Test time augmentation is a method of applying transformations on a given image to generate its several slightly different variations that are used to create predictions that, when combined, can improve final prediction. Our submissions utilized augmentations consisting of simple horizontal and vertical flips of the image. Their combinations produced four sets of detections for each image. These sets were then joined using voting strategy described in [16] by Moshkov et al..

Ensembles - Ensemble methods combine predictions from multiple models to obtain final output [21]. These methods can be used to improve accuracy in machine learning tasks. In our work, we utilize a simple method for combining outputs from multiple detection networks based on voting [16]. Detections describing one object are grouped together by size of the overlap region belonging to the same class. Instances, where majority of the detectors agree on class label and position are replaced by single detection with the highest score.

Accumulated Gradient Normalization - In order to achieve the best performance possible, we aimed to maximize the resolution of input data. Therefore,

[3] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf1_detection_zoo.md

we have decided to train the network on mini-batches of size 1. To overcome disadvantages that comes with using minimal mini-batch size [7], the Accumulated Gradient Normalization [10] technique was utilized. This approach resulted in a considerable performance gain.

3 Submissions

For evaluation of the participants submissions, the AICrowd platform[4] was used. Each participating team was allowed to submit up to 10 submission files following specific requirements for both tasks. We have used allowed maximum for both tasks. Because we have utilized single architecture for both the detection and segmentation tasks, multiple submissions were produced using the same network. Therefore in the following part, we denoted annotation and localisation task submissions by **D** and pixel-wise parsing task submissions by **S**. Finally, thresholding was used to discard predictions with low confidence.

Baseline configuration - As a baseline for all our experiments we used Mask R-CNN with ResNet-50 as a backbone. For training we used parameters and augmentations described in Table 2.2 and Section 2.3, respectively. Input resolution was 1000×1000 pixels.

Submission 1D/1S - Baseline experiment using a confidence threshold that corresponded to the best F1 score on our validation dataset (0.58).

Submission 2D - Submission 1D with a fixed programming bug that resulted in few detections being incorrectly generated.

Submission 3D - Submission 2D with confidence threshold set to 0.95.

Submission 4D/2S - Baseline configuration that used Pseudo-labels as described in Section 2.3. The confidence threshold was set to 0.95.

Submission 5D/3S - Baseline configuration that utilized test time augmentations as described in Section 2.3 with confidence threshold of 0.9.

Submission 6D/4S - Submission 5D/3S with confidence threshold of 0.999.

Submission 7D/5S - Ensemble of two checkpoints of baseline configuration model. Taken after 40 epochs and 50 epochs. Confidence threshold of 0.9.

Submission 8D/6S - Submission 7D/5S with confidence threshold of 0.999.

Submission 9D/8S - Submission 7D/5S with test time augmentations and with confidence threshold of 0.999.

Submission 10D/10S - Submission 7D/5S with confidence threshold of 0.95.

Submission 7S - Submission 9D/8S with confidence threshold of 0.9.

Submission 9S - Submission 9D/8S with modified voting ensemble. Only one detection is sufficient as opposed to majority voting.

[4] https://www.aicrowd.com

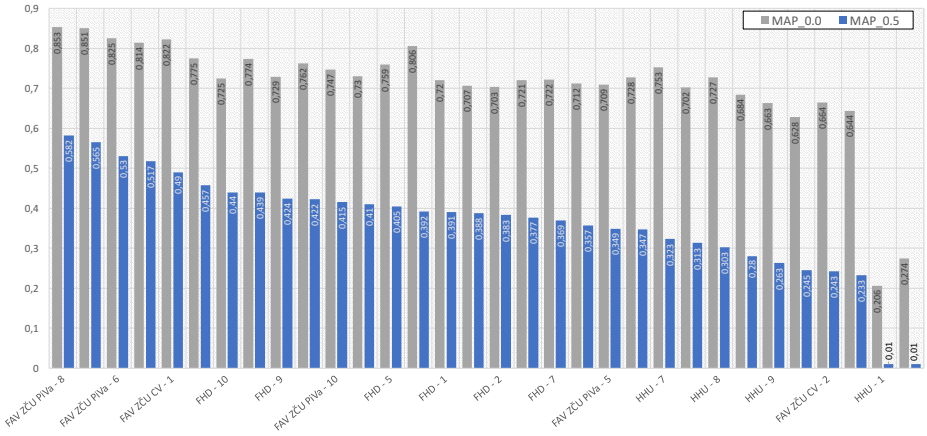

Fig. 2. Results for all runs submitted in annotation and localisation task by the competition participants, including mAP@0.0 and mAP@0.5 metrics.

4 Competition Results

The official competition results are shown in Figure 2 for annotation and localisation task, and in Figure 3 for pixel-wise parsing. Our System achieved the best performances in both tasks of the second edition of the ImageCLEFcoral competition. Specifically, mAP@0.5 of **0.582** in case of Coral reef image annotation and localisation (Run ID 68143), and mAP@0.5 of **0.678** in Coral reef image pixel-wise parsing (Run ID 67864). Results of all our submissions are listed in Table 5. Table 6 illustrates the performance over different subsets of the test dataset. The system performed comparably over the Same Location (SL), Similar Location (SiL) and Geographically Similar Location (GS) subsets. The performance significantly drops in Geographically Distinct Location (GD). This is probably caused by a lack of diverse training data.

The best scoring submission for pixel-wise parsing task was a single Mask R-CNN with ResNet-50 backbone architecture and input resolution of 1000×1000. The system was trained for 50 epochs while using heavy augmentations as described in Section 2.3. Additionally, the pseudo-labeling (refer to Section 2.3) was used to increase the training dataset size with overconfident detections from the test set. Finally, the predictions were filtered with confidence threshold of 0.95 to maximize the official mAP metric while still having decent recall score.

The best scoring submission for annotation and localisation task was an ensemble of two checkpoints of the same Mask R-CNN model with ResNet-50 backbone architecture and input resolution of 1000×1000, one taken after 40 and other one after 50 epochs. The system was trained using heavy augmentations. Furthermore, the predictions were filtered with confidence threshold of 0.999 to maximize the official metric of mAP.

Table 5. Submission scores achieved over test set. Official competition metrics.

Annotation and localisation task submissions										
Submission	1D	2D	3D	4D	5D	6D	7D	**8D**	9D	10D
mAP@0.5	0.347	0.357	0.439	0.565	0.349	0.530	0.377	**0.582**	0.517	0.415
mAP@0.0	0.728	0.712	0.774	0.851	0.709	0.825	0.721	**0.853**	0.814	0.747
Run ID	67857	67858	67862	67863	68093	68094	68138	68143	68145	68146

Pixel-wise parsing task submissions										
Submission	1S	**2S**	3S	4S	5S	6S	7S	8S	9S	10S
mAP@0.5	0.441	**0.678**	0.434	0.629	0.470	0.664	0.407	0.624	0.617	0.507
mAP@0.0	0.694	**0.845**	0.689	0.817	0.701	0.842	0.675	0.813	0.807	0.727
Run ID	67856	67864	68092	68095	68137	68139	68140	68142	68144	68147

Table 6. Submission results achieved over 4 subsets of the testing set: Same Location (SL), Similar Location (SiL), Geographically Similar Location (GS), Geographically Distinct Location (GD).

Annotation and localisation task submissions										
Submission	1D	2D	3D	4D	5D	6D	7D	8D	9D	10D
SL mAP@0.5	0.401	0.417	0.489	0.614	0.410	0.566	0.434	**0.648**	0.547	0.475
SiL mAP@0.5	0.234	0.247	0.322	**0.440**	0.230	0.431	0.254	0.343	0.438	0.258
GS mAP@0.5	0.470	0.446	0.508	0.562	0.453	0.516	0.516	**0.627**	0.533	0.527
GD mAP@0.5	0.225	0.230	0.280	0.292	0.231	**0.346**	0.210	0.329	0.344	0.242
Run ID	67857	67858	67862	67863	68093	68094	68138	68143	68145	68146

Pixel-wise parsing task submissions										
Submission	1S	2S	3S	4S	5S	6S	7S	8S	9S	10S
SL mAP@0.5	0.527	**0.744**	0.513	0.670	0.545	0.742	0.480	0.663	0.656	0.583
SiL mAP@0.5	0.312	0.516	0.309	**0.553**	0.335	0.448	0.284	0.529	0.546	0.34
GS mAP@0.5	0.476	**0.588**	0.493	0.537	0.553	0.627	0.493	0.586	0.546	0.573
GD mAP@0.5	0.276	0.403	0.283	0.439	0.266	0.386	0.267	**0.446**	0.418	0.291
Run ID	67856	67864	68092	68095	68137	68139	68140	68142	68144	68147

5 Conclusion and Discussion

The proposed system designed for automatic pixel-wise detection of 13 coral substrates achieved impressive mAP@0.5 of **0.582** in localization task and **0.678**, for instance segmentation task of the ImageCLEFcoral competitions. The system is wrapped up around the Mask R-CNN, the state-of-the-art instance segmentation framework, and additional known as well as some unique techniques, e.g., detection ensemble, test time data augmentations, accumulated gradient normalization, and pseudo-labelling. Surprisingly, results for pixel-wise parsing are considerably better. This is unexpected mainly because the test set is the same for both tasks, and our submissions used the same set of detections. Therefore, more similar scores were expected. This led us to believe that annotations for both tasks are not the same.

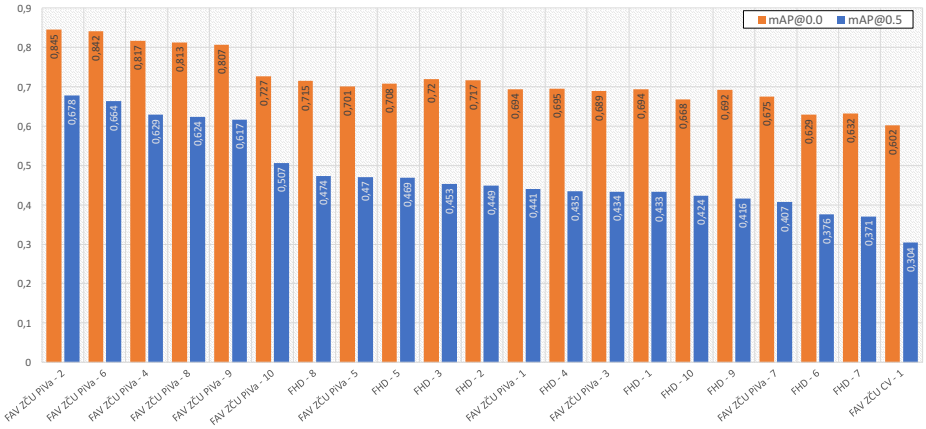

Fig. 3. Results for all runs submitted in pixel-wise parsing task by the competition participants, including mAP@0.0 and mAP@0.5 metrics.

More in-depth performance examination of our submissions revealed a small regularisation capability related to geographical regions and specific locations. This is indication that the network could be over-fitted on the training dataset location, which have specific distribution of coral species. The system could achieve better performance with class priors corresponding to desired location. If the location transfer is essential, location generalisation should be main goal for the future challenges.

While comparing the model performance with the top results from the previous edition of this challenge (mAP@0.5 of 0.2427 and 0.0419), our model achieved superior performance. Even though the test datasets are not identical, such difference shows the increasing trend of machine learning model performance. This increase is probably related to a higher number of training images.

Lastly, due to our GPU memory constraints we were limited to an input image resolution of 1000×1000 combined with ResNet-50 backbone. Conducted experiments showed that input resolution of 1200×1200 and ResNet-101 would yield better results, therefore usage of GPUs with more memory would lead to a considerable increase of the system's performance.

Acknowledgements

Lukáš Picek was supported by the Ministry of Education, Youth and Sports of the Czech Republic project No. LO1506, and by the grant of the UWB project No. SGS-2019-027.

References

1. Arazo, E., Ortego, D., Albert, P., O'Connor, N.E., McGuinness, K.: Pseudo-labeling and confirmation bias in deep semi-supervised learning. arXiv preprint arXiv:1908.02983 (2019)
2. Birkeland, C.: Global status of coral reefs: In combination, disturbances and stressors become ratchets pp. 35–56 (2019)
3. Brander, L.M., Rehdanz, K., Tol, R.S., Van Beukering, P.J.: The economic impact of ocean acidification on coral reefs. Climate Change Economics **3**(01), 1250002 (2012)
4. Chamberlain, J., Campello, A., Wright, J.P., Clift, L.G., Clark, A., García Seco de Herrera, A.: Overview of the ImageCLEFcoral 2020 task: Automated coral reef image annotation. In: CLEF2020 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org <http://ceur-ws.org> (2020)
5. Coker, D.J., Wilson, S.K., Pratchett, M.S.: Importance of live coral habitat for reef fishes. Reviews in Fish Biology and Fisheries **24**(1), 89–126 (2014)
6. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
7. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K.: Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv preprint arXiv:1706.02677 (2017)
8. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
10. Hermans, J., Spanakis, G., Möckel, R.: Accumulated gradient normalization. arXiv preprint arXiv:1710.02368 (2017)
11. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7310–7311 (2017)
12. Ionescu, B., Müller, H., Péteri, R., Abacha, A.B., Datla, V., Hasan, S.A., Demner-Fushman, D., Kozlovski, S., Liauchuk, V., Cid, Y.D., Kovalev, V., Pelka, O., Friedrich, C.M., de Herrera, A.G.S., Ninh, V.T., Le, T.K., Zhou, L., Piras, L., Riegler, M., l Halvorsen, P., Tran, M.T., Lux, M., Gurrin, C., Dang-Nguyen, D.T., Chamberlain, J., Clark, A., Campello, A., Fichou, D., Berari, R., Brie, P., Dogariu, M., Ştefan, L.D., Constantin, M.G.: Overview of the ImageCLEF 2020: Multimedia retrieval in medical, lifelogging, nature, and internet applications. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 11th International Conference of the CLEF Association (CLEF 2020), vol. 12260. LNCS Lecture Notes in Computer Science, Springer, Thessaloniki, Greece (September 22-25 2020)
13. Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., Houlsby, N.: Big transfer (bit): General visual representation learning (2019)
14. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
15. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)

16. Moshkov, N., Mathe, B., Kertesz-Farkas, A., Hollandi, R., Horvath, P.: Test-time augmentation for deep learning-based cell segmentation on microscopy images. Scientific reports **10**(1), 1–7 (2020)
17. Picek, L., Sulc, M., Matas, J.: Recognition of the amazonian flora by inception networks with test-time class prior estimation. In: Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum (2019)
18. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 91–99. Curran Associates, Inc. (2015)
19. Sulc, M., Picek, L., Matas, J.: Plant recognition by inception networks with test-time class prior estimation. In: Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum (2018)
20. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI conference on artificial intelligence (2017)
21. Zhang, C., Ma, Y.: Ensemble machine learning: methods and applications. Springer (2012)