Differences in Detection: Explainability Where it Matters

Johannes Theodoridis
University of Tübingen
Institute for Applied AI
Jo.Theodoridis@googlemail.com

Johannes Maucher
Hochschule der Medien Stuttgart
Institute for Applied AI
maucher@hdm-stuttgart.de

Andreas Schilling
University of Tübingen
schilling@uni-tuebingen.de

Abstract

We propose Differences in Detection (DnD), an intuitive method to compare two object detection models. Based on the same matching algorithm, it complements the standard metrics of mean Average Precision (mAP) and TIDE error analysis with the ability to compare two models directly. More specifically, we calculate the intersection of ground truth labels that are recognized by both models, followed by the corresponding difference sets and the complement set of ground truth labels that are missed by both models. The resulting comparison is more direct and intuitive than a comparison of independent summary statistics. It reveals individual and shared mistakes and becomes particularly interesting when combined with error types. In this case, the differences in detection errors can be analyzed naturally in a standard confusion matrix. While valuable in itself, we believe that one of the best applications of DnD is to guide explainability methods such as ODAM towards metric-relevant examples, grounded in structured subsets. The code for our method is available here:

https://github.com/JohannesTheo/differences-in-detection

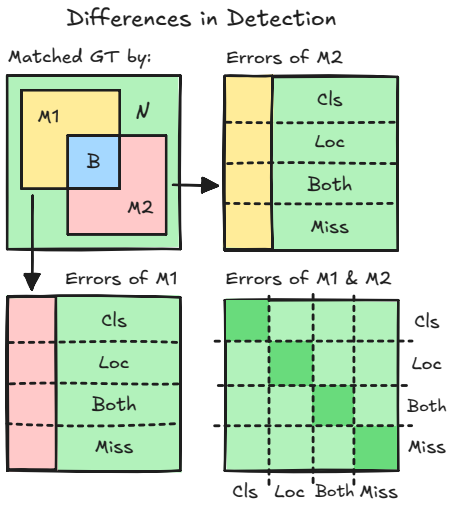

Figure 1. For the predictions of two models we calculate the intersection and difference of their matched ground truth labels. Unmatched instances can be labeled by the matching TIDE error. Together, this enables a structured comparison of predictions at the instance level which is not possible with standard mAP or TIDE.

1. Introduction

Analyzing object detection models can be easy and difficult at the same time. Standardized summary metrics such as mean Average Precision (mAP) enable a quick and concise comparison of overall performance and come with easily accessible implementations, e.g. `pycocotools` [10] or `faster-coco-eval` [12]. At the same time, they can be difficult to interpret because localization requires us to select a specific Intersection over Union (IoU) threshold in order to calculate the precision-recall curve for detections. Since predictions must also be ranked based on their score, the resulting matching algorithm, but more importantly the final value of the IoU- and class-averaged metric, is not particularly intuitive. If a model gains $+0.5$ mAP on a specific benchmark, for instance on MS-COCO [10], what does that mean? With a little effort, we can also compare the mean Average Recall (mAR) or the class-wise AP values of two models, but more insight becomes challenging. For analyzing the sources of errors, Bolya et al. [1] introduced the TIDE toolbox. It defines six types of error and measures the expected gain in mAP when errors are fixed *individually*. Previous methods calculated errors *progressively* [2, 10], conveniently adding up to 100 mAP, but greatly overestimating the impact of the last corrected error type. Together, summary statistics and TIDE errors can provide a good overview, but still have a common limitation. In both cases, models are evaluated in isolation, and we do not know whether two models make similar or different errors, or whether they even recognize the same ground truth instances. At the other end of the spectrum, visual explainability methods for object detec-

tion [3, 13, 14, 18–21] enable true in-depth analysis at the prediction level. Methods like Object Detector Activation Maps (ODAM) [21] produce heatmaps that can help us interpret model decisions. A key benefit of our method is that it provides natural and intuitive subsets to guide these methods. Consider the following example. The `val2017` split of MS-COCO has 5000 images and $37k$ ground truth annotations. The re-annotated variants Sama-COCO [23] and COCO-ReM [15] even have $41k$ and $47k$ annotations respectively. For a single image, a common evaluation protocol is to enforce a strict maximum of 100 detections. Even though the theoretical limit of $500k$ is usually not reached, object detection models often exceed the $100k$ mark in practice, which is still far too many predictions to search and analyze manually.

In the following, we present a simple solution to both problems. It complements the strengths of mAP and TIDE errors with the ability to compare two models directly and provides structured subsets of relevant predictions that can be used to apply explainability methods where it matters.

2. Differences in Detection

We propose a direct comparison of object detection models by analyzing their differences in detection on a complete dataset. More precisely, we are interested in the similarities and differences between the matching and missing ground truth labels. Assuming two models $M1$ and $M2$, we first apply the same matching algorithm as mAP to the respective prediction sets $D1$ and $D2$. Similarly to mAP, predictions can be anything that supports an IoU-based definition of true positives such as bounding boxes, instance masks, or object boundaries [4]. However, instead of calculating summary metrics, we extract the *matching pairs* of predictions and ground truth labels (dt_i, gt_m) and (dt_j, gt_m) with $dt_i \in D1$ and $dt_j \in D2$. Since both models are linked through the shared set of *ground truth labels* GT, we can now calculate more interesting subsets as displayed visually in Figure 1. For simplicity, we will reuse the model notation and refer to these subsets as follows:

	Matched GT by:	
$B = D1 \cap D2$	both	(1)
$M1 = D1 - B$	only $D1$	(2)
$M2 = D2 - B$	only $D2$	(3)
$N = GT - (D1 \cup D2)$	neither	(4)

	Unmatched GT by:	
$E1 = N \cup M2$	$D1$	(5)
$E2 = N \cup M1$	$D2$	(6)

In contrast to summary statistics like mAP, our direct comparison provides intuitive and natural subsets which can be helpful to select examples for in-depth analysis. As an alternative, we could also match $D1$ and $D2$ directly, but this would not guarantee that the resulting pairs would be relevant to the metric we are ultimately interested in. Within our approach, the error sets $E1$ and $E2$ are particularly interesting since they include the positive predictions of the other model $M2$ and $M1$, respectively. By definition, the latter are equivalent to the corresponding difference sets that contain the individual errors. Together, they allow us to analyze paired examples of failure and success, which is intriguing.

	Individual Errors of:	
$Ex1 = E1 - N = M2$	$M1$	(7)
$Ex2 = E2 - N = M1$	$M2$	(8)

So far, unmatched gt instances are defined as *missed* and have only one dt relation for individual errors. Since we use the same matching algorithm as mAP, we can reuse the error correction of TIDE and increase the depth of our analysis. For every unmatched gt_u, we query the error oracle as shown in Tab. 1, and extract the respective matching triplet (e, dt_i, gt_u) and (e, dt_j, gt_u) that would fix it.

Table 1. TIDE Error Analysis: Candidate dt are sorted by score before matching. By default, the background and foreground IoU thresholds t_b and t_f that define false and true positives are set to 0.1 and 0.5. Please see Bolya et al. [1] for a complete explanation.

Error	Class	IoU	Oracle
`Cls`	✗	$> t_f$	**match *gt***
`Loc`	✓	$< t_f$	**match *gt***
`Both`	✗	$< t_f$	ignore dt
`Dupe`	✓	$> t_f$	ignore dt
`Bkg`	✳	$< t_b$	ignore dt
`Miss`	–	–	**ignore *gt***

Importantly, we only consider false negatives of type `Cls`, `Loc` and `Miss`, which can be related to specific gt instances. The remaining error types are false positives and are fixed by ignoring the respective dt. To gain a little more insight, we deviate from TIDE and implement a matching oracle for `Both` errors as well. With everything in place, we can now compare the distributions of individual errors $Ex1$ and $Ex2$ separately, and the shared errors N in a confusion matrix. In both cases, we treat the assigned error types as categories.

In the following, we present two example applications of our proposed method in the context of MS-COCO. Specifically, we show how it complements a standard comparison with mAP and TIDE. In general, our tool can be used with any dataset that complies with the `COCO API` format [10].

2.1. Performance comparison:

In our first experiment, we compare Mask R-CNN [7] with two backbone architectures on the popular MS-COCO benchmark. We choose ConvNext-v2-B [17] and a plain VisionTransformer (ViT) [5] in the ViTDet-B [9] configuration as our models. Both are initialized from pretrained MS-COCO weights, trained with Large Scale Jitter (LSJ) [6] data augmentation, and both are evaluated on the `val2017` split. In Figure 2, we display the summary metrics mAP, mAR and a standard TIDE error analysis for bounding box predictions. As can be seen, ConvNext-v2 has a slightly higher mAP but a much better mAR compared to ViTDet. This is also reflected in the TIDE error analysis, where ViTDet would benefit more from reducing false negatives, while ConvNext-v2 would benefit more from reducing false positive predictions. More insight can be gained from a per category comparison, but neither method allows us to analyze predictions at the instance level. For instance, do both models actually recognize the same objects, and, if not, do they make similar or different mistakes?

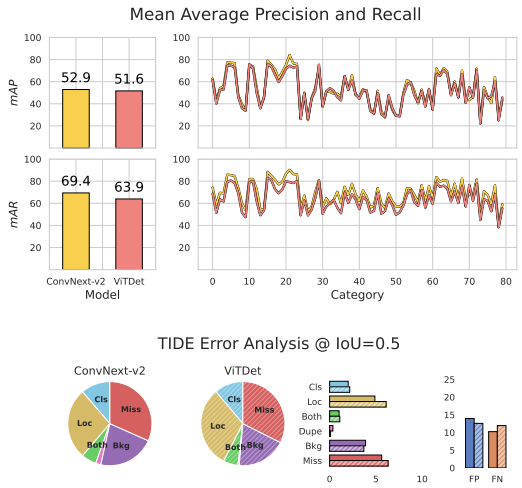

Figure 2. Comparison of summary metrics and TIDE errors for Mask R-CNN with ConvNext-v2-B and ViTDet-B on MS-COCO.

As shown in Figure 3, our proposed DnD method can easily answer these questions. Furthermore, it enables new and additional in-depth comparisons, which are not possible with mAP and TIDE alone. For example, the shared sets of matched (B) and unmatched (N) instances can be used to uncover easy and hard examples in a detection dataset or to track previously matched instances throughout the training process. However, the core benefit of DnD is that it enables a structured and direct comparison of models at the instance level. In our case, it turns out that ViTDet actually detects instances $M2$ that ConvNext-v2 does not, despite its generally better performance and a much larger set of exclusive matches $M1$. The union of shared and individual

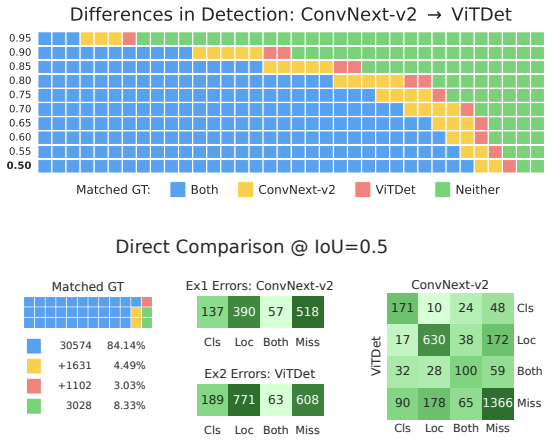

Figure 3. Differences in Detection for two Mask R-CNN models with the backbones ConvNext-v2-B and ViTDet-B on MS-COCO.

detections, $B \cup M1$ and $B \cup M2$ is equivalent to the class-agnostic recall. Although interesting, we believe that DnD is used best as a starting point for explainability methods or for hypothesis testing. In the former case, the related sets of individual errors $Ex1$ and $Ex2$ provide paired examples for visualization as shown in Fig. 4. In the latter case, the confusion matrix of shared errors N not only mirrors TIDE, but also holds references to ground truth annotations which can be used to precisely assess the effect of ablation studies.

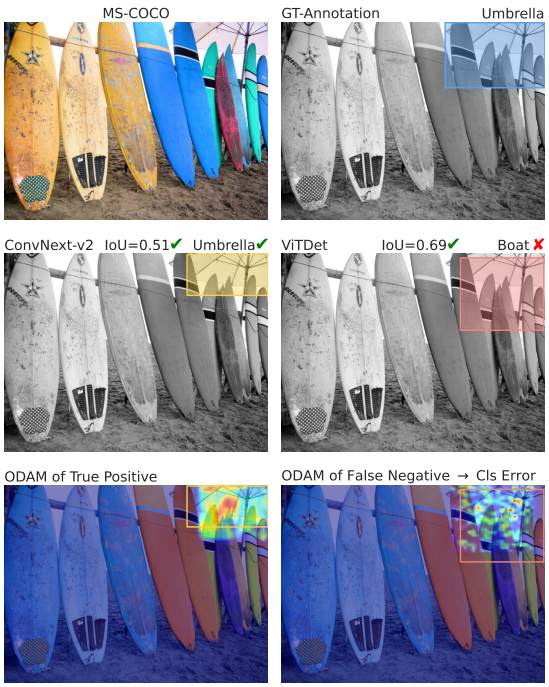

Figure 4. Example from the set of individual errors of ViTDet. ODAM visualizations display activations within proposal regions.

2.2. Robustness analysis:

In our second experiment, we show an alternative use case of DnD and compare one model on two datasets. As our model, we select Mask R-CNN with a hybrid FAN [22] backbone that was developed to improve robustness. In this case, $D1$ now represents the predictions made on MS-COCO and $D2$ the predictions made on COCO-C [11], a challenging robustness benchmark. In general, other datasets with the same GT annotations such as (Object-Centric) Stylized COCO [11, 16] can be used as well. For our experiment, we choose the common corruption `Gaussian Noise` as proposed by Hendrycks and Dietterich [8] at medium severity level 3. In Fig. 5 we compare summary metrics and TIDE errors on MS-COCO and COCO-C and show the corresponding DnD in Fig. 6.

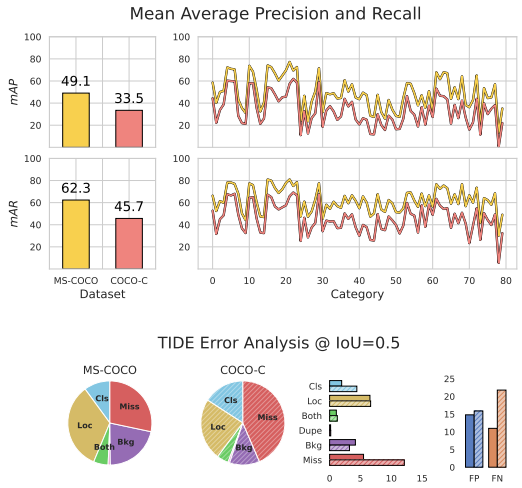

Figure 5. Comparison of summary metrics and TIDE errors for Mask R-CNN with FAN-S (hybrid) on MS-COCO and COCO-C.

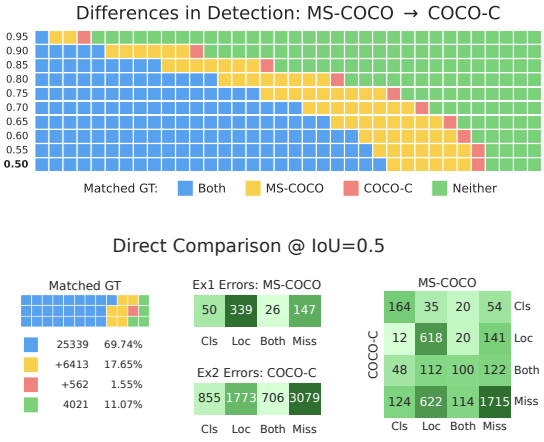

Figure 6. Differences in Detection for Mask R-CNN with FAN-S (hybrid) backbone evaluated on MS-COCO and COCO-C.

Figure 7. Example from the set of individual errors on COCO-C. ODAM visualizations display activations within proposal regions.

As can be seen, Mask R-CNN with FAN-S is still very much affected by the added gaussian noise in COCO-C and we visualize one such example from the $Ex2$ set in Fig. 7. Interestingly, some previously unmatched instances are detected after adding noise, which is not intuitive or expected.

3. Limitations

For more than two models, DnD must be applied in a chain of comparisons in its current form. However, extending the set definitions to the multi model case could be interesting, for instance to analyze the contributions of individual models in ensemble approaches. A second limitation is that DnD is defined from the GT perspective. In consequence, it represents the class-agnostic recall and only considers positively matched predictions or TIDE errors that can fix a false negative gt_u example. Extending DnD to false positive predictions as well would enable a direct comparison that is fully related to the mAP metric, but requires to match the detections of two models directly, which is not well-defined and left as future work.

4. Conclusion

In this work, we propose Differences in Detection (DnD), an intuitive method to compare the predictions and errors of two object detection models directly, complementing the strength of mAP and TIDE error analysis. Derived from the same matching algorithm, DnD subsets are well-defined and structured which is interesting in general but particularly useful for selecting metric-relevant examples for explainability methods in situations where it matters.

References

[1] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. TIDE: A General Toolbox for Identifying Object Detection Errors. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III*, pages 558–573, Berlin, Heidelberg, 2020. Springer-Verlag. 1, 2

[2] Ali Borji and Seyed Mehdi Iranmanesh. Empirical Upper Bound in Object Detection and More, 2019. arXiv:1911.12451 [cs]. 1

[3] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847, 2018. 2

[4] Bowen Cheng, Ross Girshick, Piotr Dollar, Alexander C. Berg, and Alexander Kirillov. Boundary IoU: Improving Object-Centric Image Segmentation Evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15334–15342, 2021. 2

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. 3

[6] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 2918–2928, 2021. 3

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 3

[8] Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*, 2019. 4

[9] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring Plain Vision Transformer Backbones for Object Detection. In *Computer Vision – ECCV 2022*, pages 280–296, Cham, 2022. Springer Nature Switzerland. 3

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 1, 2

[11] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. In *Machine learning for autonomous driving workshop, NeurIPS 2019*, 2019. 4

[12] MiXaiLL76. Faster-COCO-Eval: Faster interpretation of the original COCOEval, 2024. 1

[13] Vitali Petsiuk, Rajiv Jain, Varun Manjunatha, Vlad I. Morariu, Ashutosh Mehra, Vicente Ordonez, and Kate Saenko. Black-box Explanation of Object Detectors via Saliency Maps. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11438–11447, 2021. 2

[14] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. 2

[15] Shweta Singh, Aayan Yadav, Jitesh Jain, Humphrey Shi, Justin Johnson, and Karan Desai. Benchmarking Object Detectors with COCO: A New Path Forward. In *Computer Vision – ECCV 2024*, pages 279–295, Cham, 2025. Springer Nature Switzerland. 2

[16] Johannes Theodoridis, Jessica Hofmann, Johannes Maucher, and Andreas Schilling. Trapped in Texture Bias? A Large Scale Comparison of Deep Instance Segmentation. In *Computer Vision – ECCV 2022*, pages 609–627, Cham, 2022. Springer Nature Switzerland. 4

[17] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. ConvNeXt V2: Co-Designing and Scaling ConvNets With Masked Autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16133–16142, 2023. 3

[18] Shujun Xia, Chenyang Zhao, and Antoni Chan. Explaining Object Detection Through Difference Map. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 713–722, 2025. 2

[19] Toshinori Yamauchi. Spatial Sensitive Grad-CAM++: Improved Visual Explanation for Object Detectors via Weighted Combination of Gradient Map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 8164–8168, 2024.

[20] Toshinori Yamauchi and Masayoshi Ishikawa. Spatial Sensitive GRAD-CAM: Visual Explanations for Object Detection by Incorporating Spatial Sensitivity. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 256–260, 2022.

[21] Chenyang ZHAO and Antoni B. Chan. ODAM: Gradient-based Instance-Specific Visual Explanations for Object Detection. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[22] Daquan Zhou, Zhiding Yu, Enze Xie, Chaowei Xiao, Animashree Anandkumar, Jiashi Feng, and Jose M. Alvarez. Understanding The Robustness in Vision Transformers. In *Proceedings of the 39th International Conference on Machine Learning*, pages 27378–27394. PMLR, 2022. 4

[23] Eric Zimmermann, Justin Szeto, Jerome Pasquero, and Frederic Ratle. Benchmarking a Benchmark: How Reliable is MS-COCO? In *ICCV 2023 DataComp Workshop*, 2023. _eprint: 2311.02709. 2