Appendix

In this appendix, we provide supplemental information relevant to the paper. First, in appendix A we provide additional quantitative experiments where we investigate related loss functions and furthermore test out scenarios that could be encountered in real-world problems. Next, in appendix B we provide an analysis of error types made with our method. In appendix C we furthermore provide an analysis of qualitative results of our detection method. We finish off with derivations of statistical properties in appendix D.

A Additional Quantitative Experiments

A.1 Comparison of MS-SSIM Loss Functions

Multiscale SSIM (MS-SSIM) [4] is an adaptation of SSIM in which SSIM is calculated using varying window sizes F on the same feture map. MS-SSIM and combinations of MS-SSIM with ℓ_1 have proven successful in deep learning applications [5] in the image quality assessment domain. In this set of experiments we evaluate the performance of these various loss functions. Following [5], we test out smooth-ℓ_1, $\ell_{\text{MS-SSIM}}$, and a combination of $0.15 \cdot \ell_1 + 0.85 \cdot \ell_{\text{MS-SSIM}}$. The results are presented in table 1. It can be observed that: (i) smooth ℓ_1 boosts performance by 1.8 on its own. (ii) Adopting SSIM and the variations MSSIM and $\ell_{\ell_1+MS-SSIM}$ result in AP improvements of 3.5, 3.6, very similar to ℓ_{SSIM}. This demonstrates that adopting any form of SSIM is more beneficial than the pointwise ℓ_p norms.

Backbone	$\mathcal{L}_\varepsilon$	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Teacher ResNet-101		41.0	60.3	44.0	24.1	45.3	53.8
Vanilla ResNet-50	-	*36.4*	*55.6*	*38.7*	*21.1*	*40.3*	*46.6*
ResNet-50	$\ell_{1,\text{smooth}}$	38.2 (+1.8)	57.2	40.7	21.5	41.9	49.9
ResNet-50	$\ell_{\text{MS-SSIM}}$	39.9 (+3.5)	59.1	42.7	22.6	44.0	**53.7**
ResNet-50	ℓ_1 + $\ell_{\text{MS-SSIM}}$	40.0 (+3.6)	**59.2**	**43.2**	22.6	44.0	53.0
ResNet-50	$\boldsymbol{\ell}_{\text{SSIM}}$	**40.1 (+3.7)**	**59.2**	43.1	**23.1**	**44.6**	53.2

Table 1: Comparison of distillation functions using RetinaNet [2] on MSCOCO [1]

A.2 Real World Applications

Unlabeled Data In order to simulate a scenario in which data annotations are not available we furthermore investigate performance on MSCOCO [1] without GT annotations. This is achieved through *hard output distillation*, i.e. we use the outputs of the teacher model with confidence $p > 0.3$ as labels for the student. The results are shown in Table 2. Our ℓ_{ssim} method achieves a +1.9 AP improvement over the vanilla model, compared to +1.1 AP when using ℓ_2. This furthermore demonstrates the advantage ℓ_{ssim} distillation can bring when dealing with a scenario in which annotations are not available.

Backbone	$\mathcal{L}_\varepsilon$	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Teacher ResNet-101		41.0	60.3	44.0	24.1	45.3	53.8
Vanilla ResNet-50		*34.6*	*53.8*	*36.7*	*20.3*	*38.4*	*43.7*
ResNet-50	ℓ_2	35.7 (+1.1)	54.9	38.1	20.5	39.5	44.8
ResNet-50	$\boldsymbol{\ell}_{\text{SSIM}}$	**36.5 (+1.9)**	**55.8**	**38.9**	**21.6**	**40.4**	**46.1**

Table 2: RetinaNet [2] experiments on MSCOCO [1] w/o annotations.

Robustness For use cases such as autonomous driving, it is of major importance that the detector functions regardless of image distortions or weather conditions. The Robust Detection Benchmark [3] introduces a new way to evaluate detectors in which the performance of the algorithm is tested over 15 different types of distortions such as blur, noise, snow and fog conditions. Additionally, five different severity levels are introduced for each distortion, for a total of 75 different scenarios. Two metrics are introduced: mPC (mean Performance under Corruption) measures the average AP over each of the distortions, while rPC (relative Performance under Corruption) measures the performance on distortions relative to clean data. It can be observed in table 3 that our distillation method not only is more robust (+2.1 mPC), but also improves the relative robustness (+0.7 rPC). Our distillation method therefore not only demonstrates an absolute increase in performance, but also has improved generalization ability to scenarios in which the visual conditions are not as optimal as in a prepared dataset.

Backbone	$\mathcal{L}_\varepsilon$	AP	mPC	rPC
Vanilla ResNet-50		*36.5*	*18.0*	*49.4*
ResNet-50	ℓ_{ssim}	40.1 (+3.6)	20.1 (+2.1)	50.1 (+0.7)

Table 3: RetinaNet [2] robustness experiments on MSCOCO [1]

B Error Analysis

Takeaway *The types of error that our distilled detector makes are relatively similar to that of a vanilla model. The similar pattern is that the types of errors made are fairly well distributed, with slightly more class and background confusions. Our method is furthermore particularly effective in improving performance in more strict localization metrics, and in the detection of large objects.*

In order to gain insight in the overall strengths and weaknesses of our distillation method, we conduct an investigation of the types of errors made on the MSCOCO [1] validation dataset. We compare a vanilla RetinaNet-50 [2] trained without distillation which we refer to as the baseline to our SSIM distillation method.

Figure 1 shows a curve averaged over all class categories for different types of errors for the baseline and for our method. Each plot consists of a series of precision-recall curves with each curve denoting a slightly more permissive evaluation setting. Overall AP_{75} is .431, 11.4% better than the baseline, and for a more permissive AP_{50} we arrive at .591, a 6.3% improvement. If we furthermore assume perfect localization, the AP increases from .633 to .665, a 5.1% improvement. It can be observed that as we increase the permissiveness of the localization of our detector, the performance improvement is relatively less. Therefore we can conclude that our method is mainly effective in improving detection scenarios that require more precise localization.

If we furthermore move on to loosening the classification requirements, we can see that when equalizing similar categories the AP reaches 0.697, 3.9% better than the baseline. Removing all class confusions pushes AP to 0.776, 2.6% better than vanilla detection, and removing background FPs results in .878 AP, 1.3% better. Overall it can be observed that the types of errors made are quite diverse, but lean slightly to class confusions of other classes and background confusions.

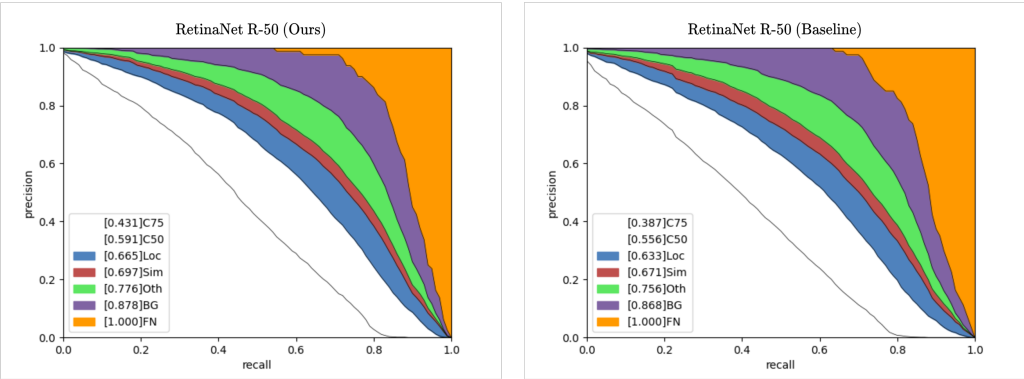


Figure 1: Distribution of error types on MSCOCO [1]. Area under Curve is provided in brackets in the legend: **C75** - at box IoU .75 (AP_{75}); **C50** - at at box IoU .50 (AP_{50}); **Loc** - at IoU .10 (localization ignored, no duplicates); **Sim** - after removal of supercategory FPs; **Oth** - after removal of all class confusions; **BG** - after removal of all background FPs; **FN** - False Negative predictions ($AP = 1.00$).

Furthermore, in fig. 2 we again (ref. to fig. 7 in main paper) illustrate the types of error sorted by box size, where the comparison is split up in evaluation settings with increasing permissiveness. It can be noticed that the most substantial improvement in distillation performance is achieved in the large detection category. Furthermore, the most substantial improvements particularly in the medium and large category are achieved in the stricter evaluation settings.

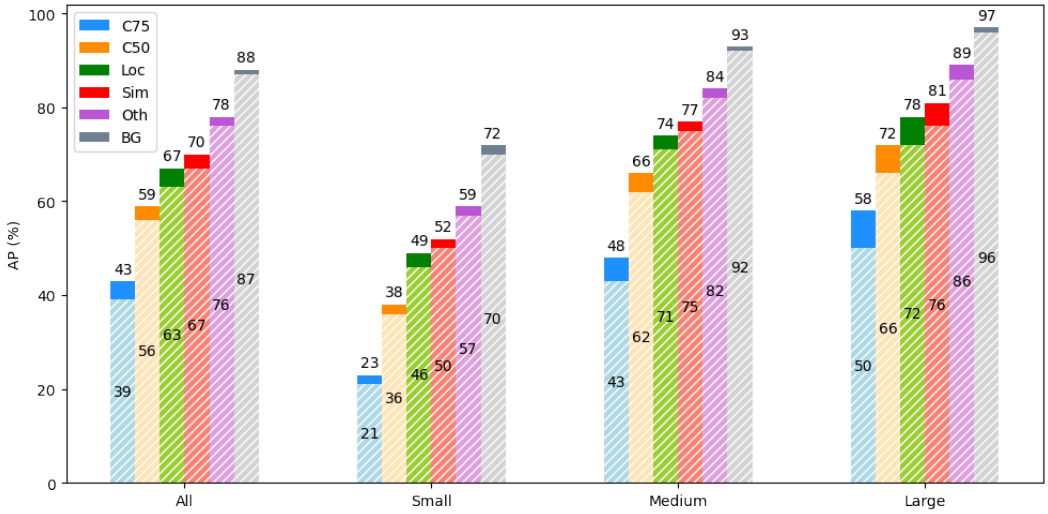


Figure 2: AP score for varying box sizes. Hatched areas represent the baseline, solid areas represent the performance increase obtained through distillation.

C Qualitative Analysis

Takeaway *We demonstrate several examples where the quantitative results are supported by the qualitative results. Particularly the detection of large objects is significantly improved, and in cases even surpasses the performance of the teacher. Additionally, both in confusing detection cases and more straightforward cases the knowledge transfer from teacher to student is manifested, both positively and negatively.*

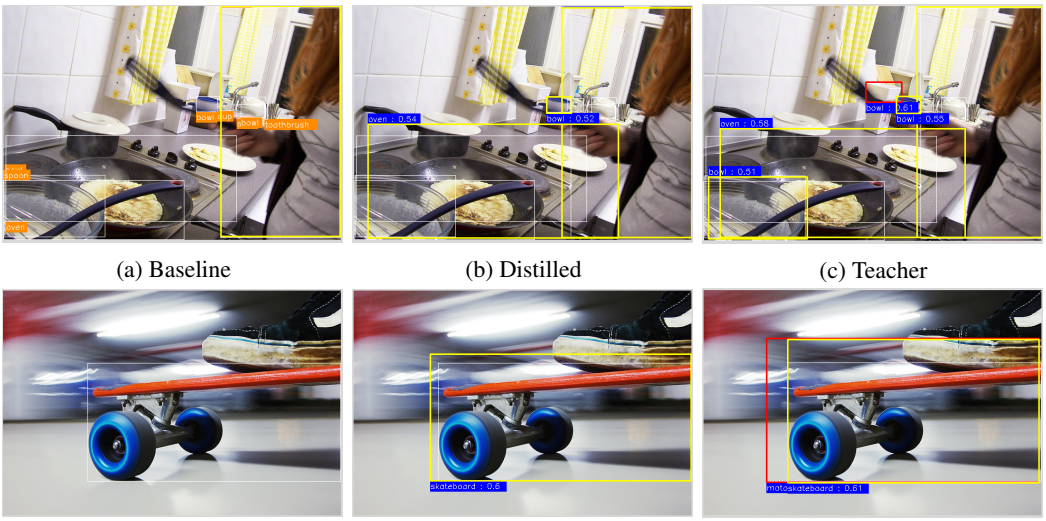
In order to verify the effectiveness of our method we analyze qualitative results in the form of several examples of detections, where we compare three models: (i) a vanilla RetinaNet-50 [2] trained without distillation which we refer to as *baseline*, (ii) a RetinaNet-50 [2]trained with our SSIM distillation method, which we refer to as *distilled*, and (iii) additionally we include the results produced by a teacher RetinaNet-101 [2], which we simply refer to as *teacher*. Throughout this section, yellow boxes denote correct predictions, red boxes denote incorrect predictions or localizations with false class predictions, and white boxes are ground truth bounding boxes.

First of all in fig. 3 we provide an example of a straightforward detection scenario, in order to obtain an indication of the overall performance. As can be expected, both the classification and localization across the board are very good. However, in this case the confidence with which our method predicts the classes is significantly higher than the baseline.

(a) Baseline (b) Distilled (c) Teacher

Figure 3: Straightforward detection scenario

Next, we provide some examples in order to verify the quantitative results which indicate that our method particularly excels in the AP_L category, which is a reflection of the performance on large objects. Figure 4a - 4c presents an example of a relatively complex scene containing multiple large objects. It can be observed that our method is able to detect additional large objects not detected by the baseline. The detections are still not as plentiful as the teacher, but the models does also not make a false positive detection. This phenomenon can also be observed in fig. 4d - 4f, where a detection is made on a close-up of a single object. The distilled model is able to detect the object, and furthermore does not make the false positive prediction made by the teacher.

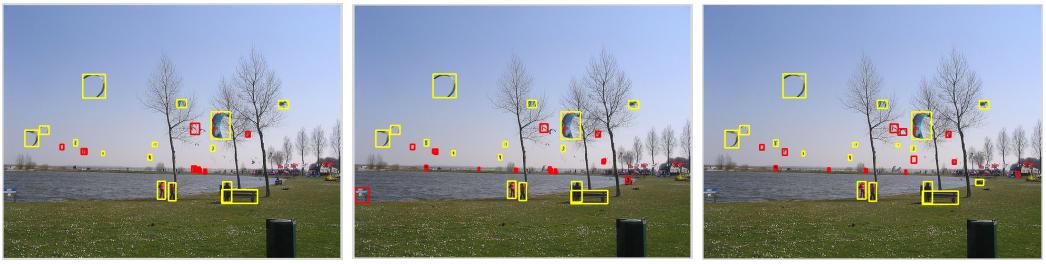


(a) Baseline (b) Distilled (c) Teacher

(d) Baseline (e) Distilled (f) Teacher

Figure 4: Detection scenarios with large objects

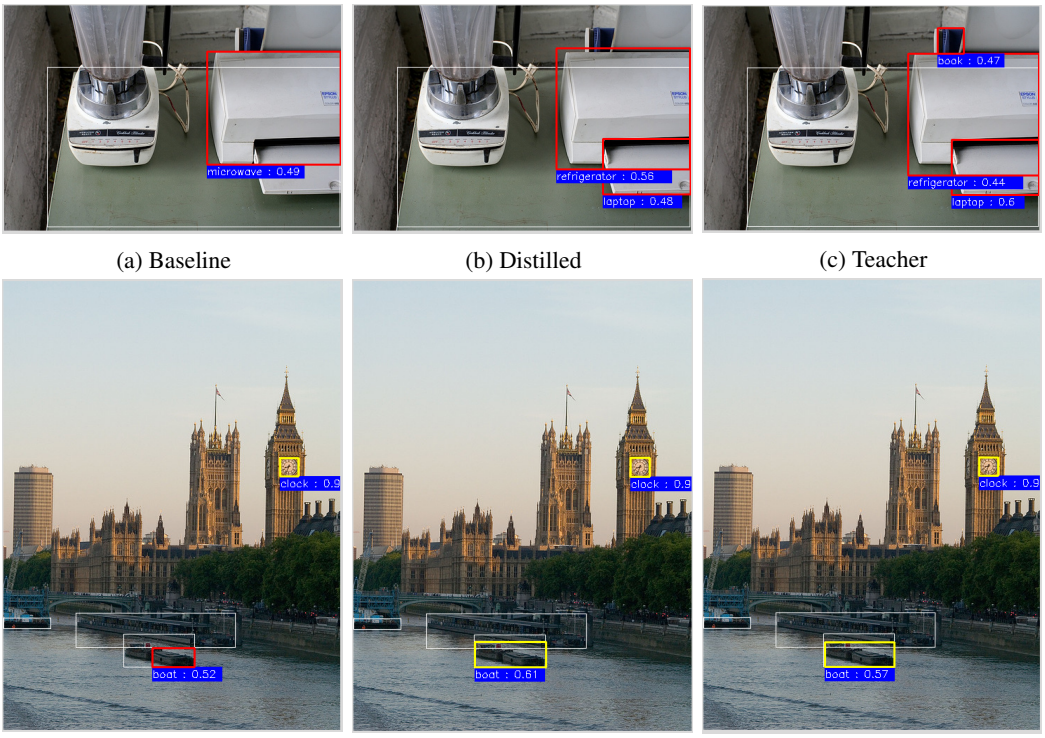
Next, we look at an example in which our method improves performance on detection of small objects. Although not as substantial as in large objects, the AP improvement over the baseline is still 2-3 AP across various evaluation settings, refer to fig. 2. Figure 5 illustrates an example of the distilled model's ability to detect objects that are tiny because of their large distance. Note that the ground truth annotations are not always perfectly accurate, in this case some clearly correct detections of persons in a distance are reported as incorrect.



(a) Baseline (b) Distilled (c) Teacher

Figure 5: Detection scenario containing many small objects

Finally, we analyze examples in which the qualitative results indicate that knowledge transferred from the teacher had impact. Figure 6a - 6c illustrate an example of incorrect predictions by each model. In contrast to the baseline, the distilled model mimics the teacher in making the same (incorrect) class prediction and an additional incorrect localization prediction. Furthermore in fig. 6d - 6f the distilled model produces improved localization compared to the baseline, where it can be observed that the teacher is mimicked.



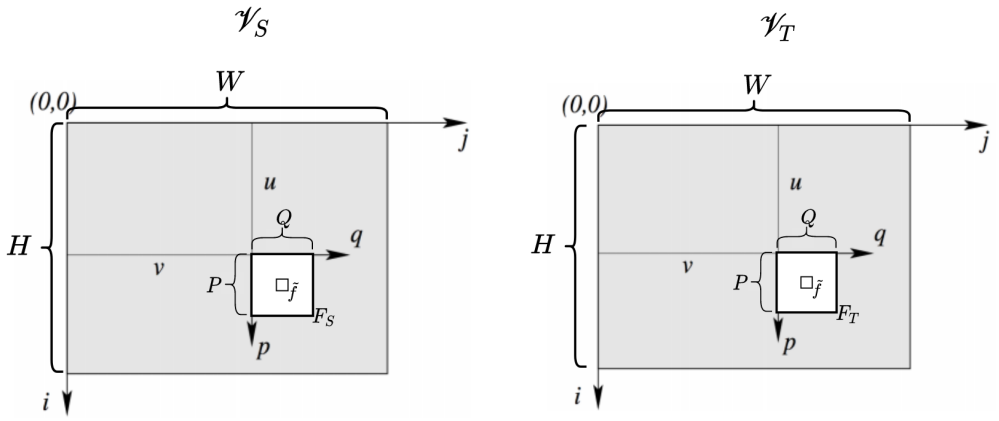
(a) Baseline (b) Distilled (c) Teacher

(d) Baseline (e) Distilled (f) Teacher

Figure 6: Detection scenarios with information transfer. **a-c** example of classification transfer. **e-f** example of localization transfer.

D Derivations

In this section we demonstrate how to calculate each statistical property used in our KD method.



(a) Student (b) Teacher

Figure 7: Geometric illustration of intermediate feature maps. u and v are the location of the top left feature of patches F. The patches are subsequentely defined as follows: $F_{\mathcal{S}} = \mathcal{V}_{\mathcal{S}}([u, u+1, ..., u+P], [v, v+1, ..., v+Q])$ and $F_{\mathcal{T}} = \mathcal{V}_{\mathcal{T}}([u, u+1, ..., u+P], [v, v+1, ..., v+Q])$ with central feature $\tilde{f}$. Finally P and Q indicate the size of the patch.

Mean

$$\mu_{\mathcal{S}}(F_{\mathcal{S}}) = \frac{1}{PQ}\sum_{p=1}^{P}\sum_{q=1}^{Q} F_{\mathcal{S}}(p,q) \qquad (1a) \qquad \mu_{\mathcal{T}}(F_{\mathcal{T}}) = \frac{1}{PQ}\sum_{p=1}^{P}\sum_{q=1}^{Q} F_{\mathcal{T}}(p,q) \qquad (1b)$$

Variance

$$\sigma^2_{\mathcal{S}}(F_{\mathcal{S}}) = \frac{1}{PQ-1}\sum_{p=1}^{P}\sum_{q=1}^{Q}\left(F_{\mathcal{S}}(p,q) - \underbrace{\frac{1}{PQ}\sum_{p=1}^{P}\sum_{q=1}^{Q} F_{\mathcal{S}}(p,q)}_{\mu_{\mathcal{S}}(F_{\mathcal{S}})}\right)^2 \qquad (2)$$

$$\sigma^2_{\mathcal{T}}(F_{\mathcal{T}}) = \frac{1}{PQ-1}\sum_{p=1}^{P}\sum_{q=1}^{Q}\left(F_{\mathcal{T}}(p,q) - \underbrace{\frac{1}{PQ}\sum_{p=1}^{P}\sum_{q=1}^{Q} F_{\mathcal{T}}(p,q)}_{\mu_{\mathcal{T}}(F_{\mathcal{T}})}\right)^2 \qquad (3)$$

Covariance

$$\sigma_{\mathcal{ST}}(F_{\mathcal{S}}, F_{\mathcal{T}}) = \frac{1}{PQ-1}\sum_{p=1}^{P}\sum_{q=1}^{Q}\left[\left(F_{\mathcal{S}}(p,q) - \underbrace{\frac{1}{PQ}\sum_{p=1}^{P}\sum_{q=1}^{Q} F_{\mathcal{S}}(p,q)}_{\mu_{\mathcal{S}}(F_{\mathcal{S}})}\right)\cdot \left(F_{\mathcal{T}}(p,q) - \underbrace{\frac{1}{PQ}\sum_{p=1}^{P}\sum_{q=1}^{Q} F_{\mathcal{T}}(p,q)}_{\mu_{\mathcal{T}}(F_{\mathcal{T}})}\right)\right] \qquad (4)$$

References

[1] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. *ECCV*, 2014.

[2] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *CVPR*, 2017.

[3] 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. *arXiv preprint arXiv:1907.07484*, 2019.

[4] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems \& Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[5] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss Functions for Image Restoration With Neural Networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, 2016. ISSN 2573-0436. doi: 10.1109/tci.2016.2644865.