Supplementary Material

Trapped in texture bias? A large scale comparison of deep instance segmentation

Johannes Theodoridis[1,2], Jessica Hofmann[1], Johannes Maucher[1], and
Andreas Schilling[2]

[1] Institute for Applied AI - Hochschule der Medien Stuttgart, Germany
{theodoridis,jh275,maucher}@hdm-stuttgart.de
[2] University of Tübingen, Germany andreas.schilling@uni-tuebingen.de

Table of Contents

1 Stylized COCO in Depth

In this section we provide more examples for Stylized COCO and its object-centric variants. Figure 1 shows the effect of masking and controlling the style strength. Figure 2 displays the difference between the *extreme* points $\alpha = 0$ (no style, only image corruption) and $\alpha = 1$ (full style). Images are chosen intentionally to show the effect of *disappearing* objects. Figure 3 provides a representative sample of images in Stylized COCO. Finally we porivde a complete example for one image in Figure 4. Note that the blending sequences are created for every dataset version.

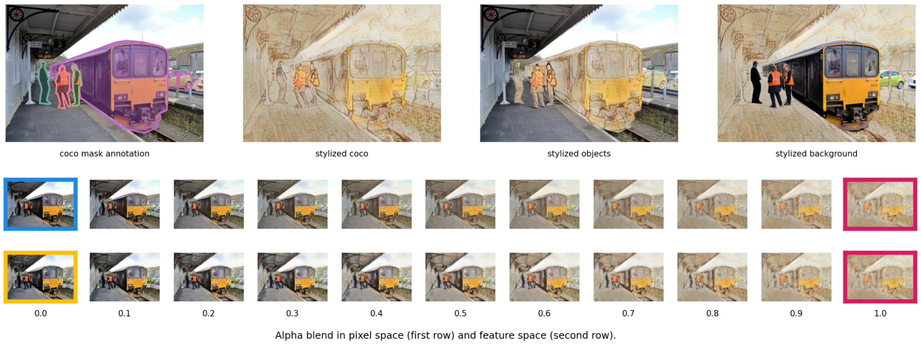


Fig. 1: Comparison of Stylized COCO, Stylized Objects and Background. Bottom rows show the pixel and feature space blending sequences for stylized COCO

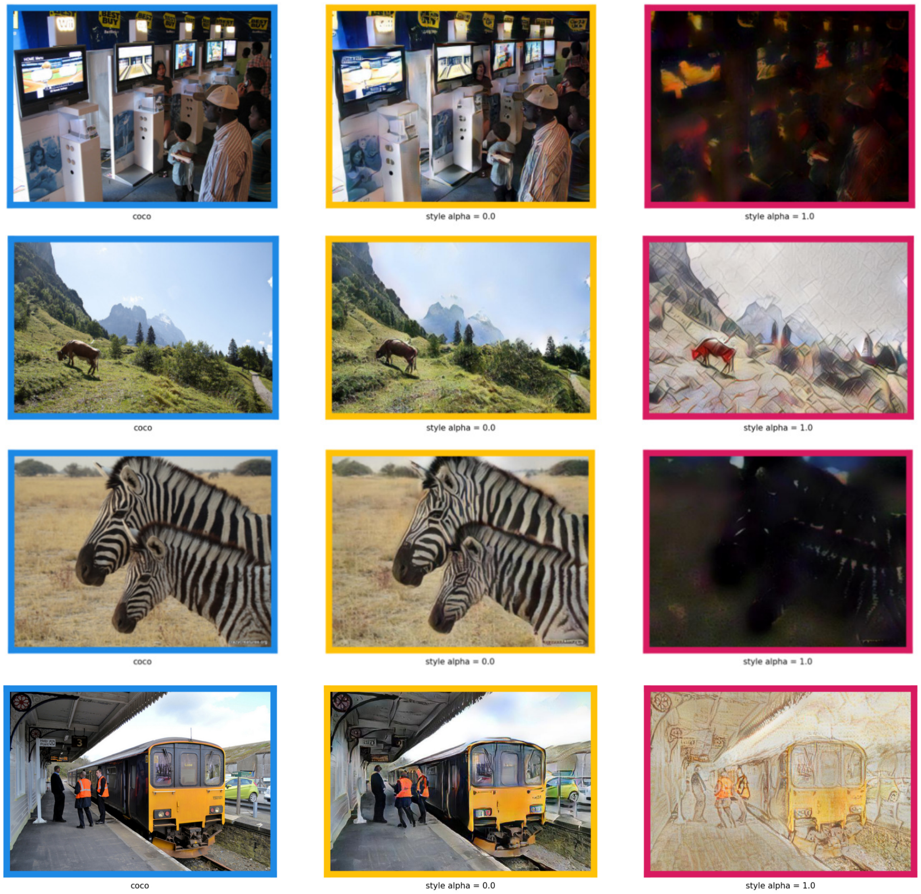


Fig. 2: Comparison of AdaIn style transfer strength. Depending on the style image, an alpha value of 1 (pink) can produce rather extreme versions where objects are almost eradicated from the scene. An alpha value of 0 (yellow) corresponds to a style transfer of the content image with itself. As can bee seen in the middle column, this variant already introduces subtle image corruptions to the shape and texture of objects

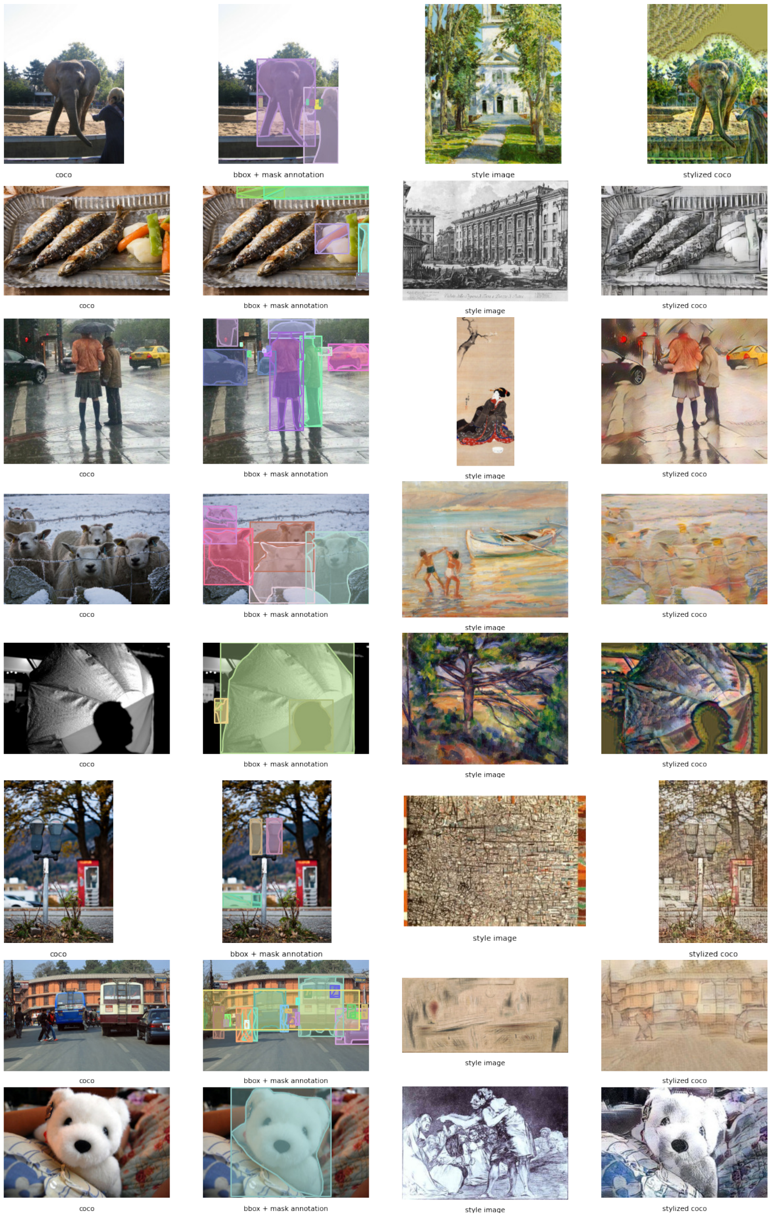
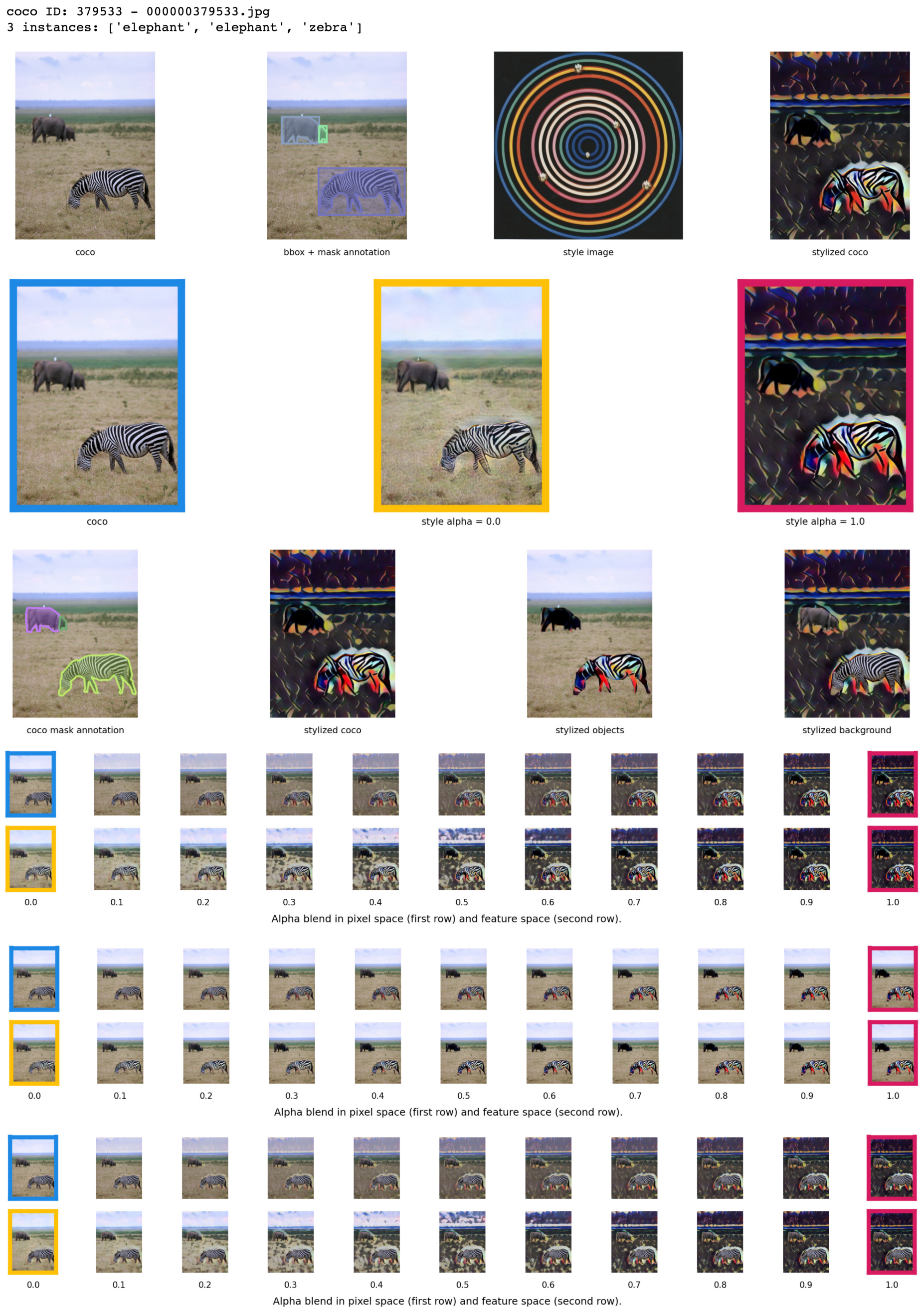


Fig. 3: Creation process of Stylized COCO. We plot the mask annotations to locate ground truth instance in the stylized images. The mask annotations are used to create the Stylized Objects and Background variants



```
coco ID: 379533 - 000000379533.jpg
3 instances: ['elephant', 'elephant', 'zebra']
```

Fig. 4: Full example for one image. *Top row:* Creation Process of Stylized-COCO. *Second row:* Comparison of style strength. *Third row:* Comparison of the dataset versions at $\alpha = 1$. *Last three rows:* Blending Sequences for Stylized COCO, Objects and Background. We create these for every image which results in 60 copies of the COCO `val2017` subset

2 Code and Model Weights

The code to reproduce Stylized COCO, our results and the resulting detection and evaluation data can be found here:

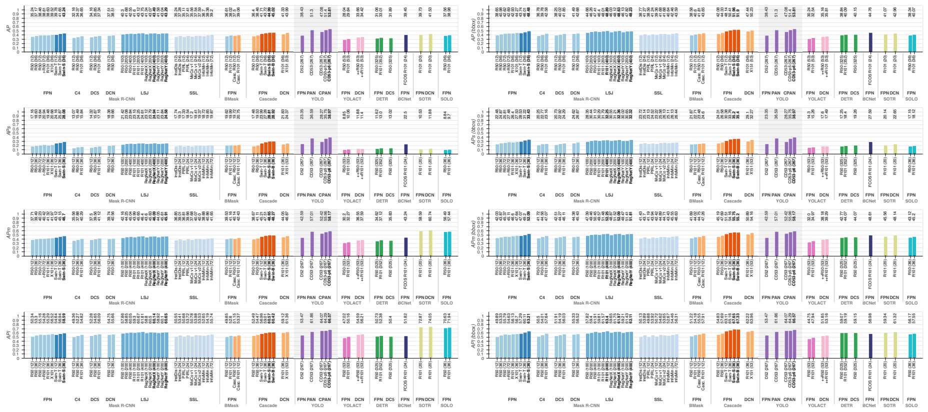
- `https://github.com/JohannesTheo/trapped-in-texture-bias`

We use code and pre-trained models from the following popular projects. Note that unfortunately, CPMask [1] was not fully released at the time of this publication but will be included in our code release in the future. Before we test a model on Stylized COCO and its variants, we reproduce the reported score on COCO `val2017`. Models that do not reported metrics on `val2017` have been validated on `test-dev2017` before testing.

- Detectron 2: `https://github.com/facebookresearch/detectron2/`
- BMask R-CNN: `https://github.com/hustvl/BMaskR-CNN`
- PyContrast (SSL): `https://github.com/HobbitLong/PyContrast/`
- Swin: `https://github.com/SwinTransformer/Swin-Transformer-Object-Detection`
- YOLO: `https://github.com/AlexeyAB/darknet`
- YOLACT(++): `https://github.com/dbolya/yolact`
- DETR: `https://github.com/facebookresearch/detr`
- BCNet: `https://github.com/lkeab/BCNet`
- CPMask: `https://github.com/fanq15/FewX`
- SOTR: `https://github.com/easton-cau/SOTR`
- SOLOv2: `https://github.com/aim-uofa/AdelaiDet/tree/master/configs/SOLOv2`

3 Complete Results (including bounding box scores)

In this section we provide a complete overview of our results. In particular, this includes all common COCO scores, the absolute model performances and the corresponding metrics for IoU type bounding box. Figure 5 compares all models on the original COCO `val2017` subset. The data analysis of our main study is guided by the distance matrices displayed in Figure 6. A key observation from this comparison is that models perform more similar to models within the same framework than to models from other frameworks, see Table 1. Figure 7 shows the large scale comparison from the main paper. In addition to the relative performance we display the corresponding plot with absolute performance and IoU type bounding box. Similarly, Figure 8, Figure 9 and Figure 10 provide the bounding box results for the controlled comparisons of framework, backbone and neck architecture.

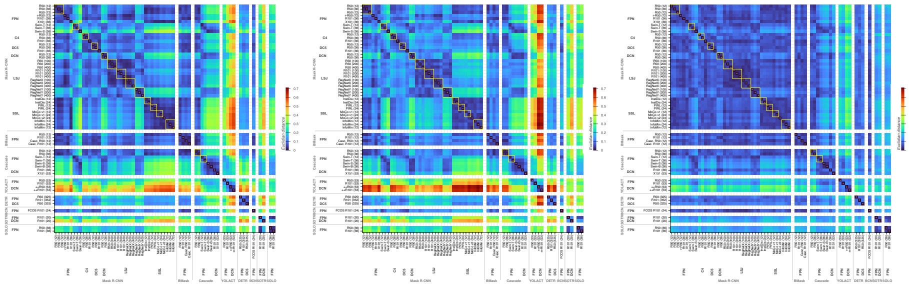


(a) Segmentation scores (YOLO is bbox)
(b) Bounding Box scores

Fig. 5: Comparison of absolute model performance on COCO `val2017`. Methods that do not report scores for `val2017` have been validated on `test-dev2017`

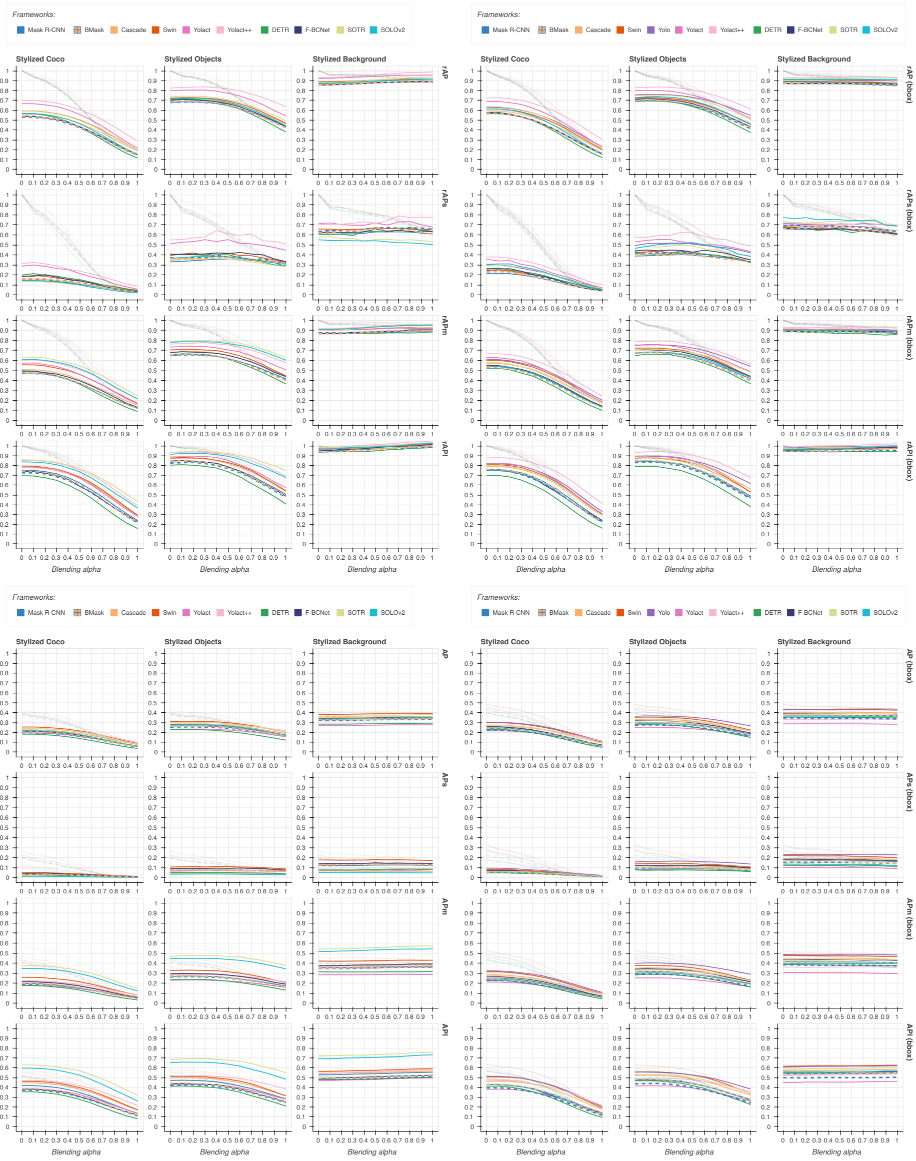
Table 1: Performance similarity of models within and outside a framework group.

Average model distance (L2)	to models from the same framework	to models from other frameworks
Stylized-COCO	0.09 ± 0.06	0.23 ± 0.08
Stylized-Objects	0.09 ± 0.05	0.26 ± 0.11
Stylized-Background	0.06 ± 0.03	0.14 ± 0.05



(a) Stylized COCO
(b) Stylized Objects
(c) Stylized Background

Fig. 6: Euclidian distance between the relative performance of models over the full alpha range. Zoom in for better visibility. We average the L2 distance over AP, APs, APm and APl. Yellow squares highlight the same model with different learning schedule



(a) Segmentation scores

(b) Bounding Box scores

Fig. 7: Large scale comparison of model robustness. *Top row* shows relative performance as in the main paper. *Bottom row* displays absolute performance for comparison. Note that comparing relative or absolute performance results in a different ranking of frameworks due to the varying base performance

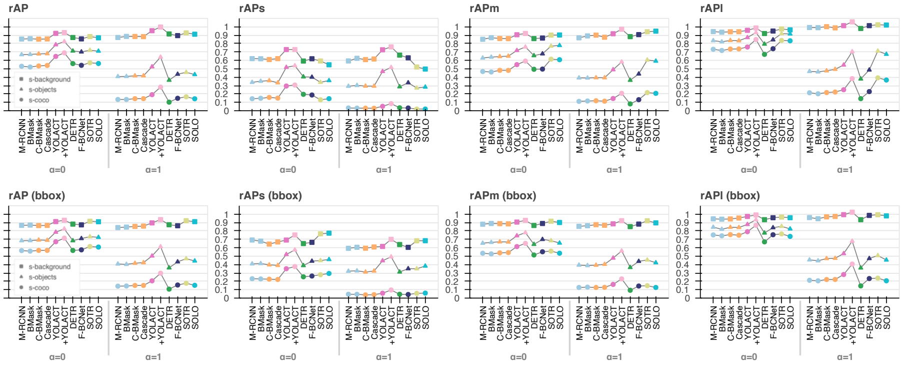
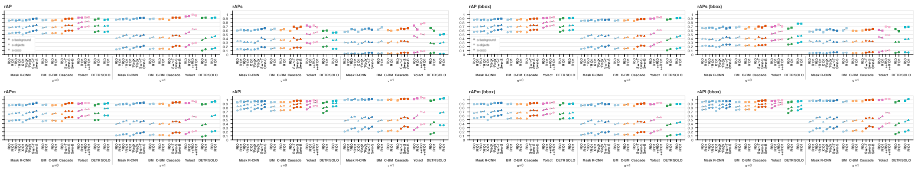


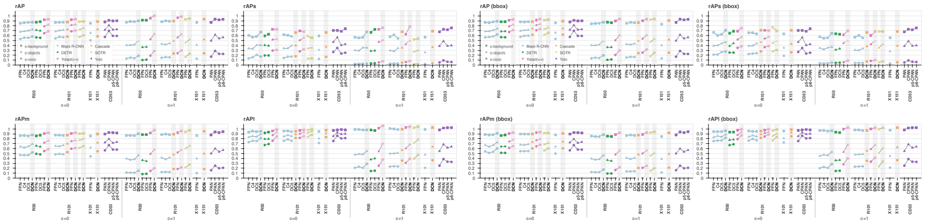
Fig. 8: Controlled comparison of robustness by framework (fixed backbone and neck: R50 + FPN). We compromise on R101 for SOTR and F-BCNet. *Top row:* segmentation scores. *Bottom row:* bounding box scores



(a) Segmentation scores

(b) Bounding Box scores

Fig. 9: Controlled comparison of robustness by backbone architecture. Models marked with * are trained with LSJ



(a) Segmentation scores (YOLO is bbox)

(b) Bounding Box scores

Fig. 10: Controlled comparison of robustness by neck type. Models with dynamic components are highlighted

4 Ablation Study: Salt and Pepper Noise

In this section we conduct an ablation study to answer two questions which arose during our experiments:

1. How does style transfer compare to common corruption types?
2. Is the better performance on masked objects a result of object contour or a side effect of less corruption in the image?

To answer these questions, we create am additional version of COCO `val2017` with Salt-and-pepper noise (impulse noise). A comparison to Stylized COCO and the masked variants are shown in Figure 11. We use `skimage.util.random_noise` with `mode="s&p"` and `amount=0.2` so that 20% of the pixels are corrupted. The key difference that can be observed is that impulse noise corruptions follow an independence assumption whereas style transfer corruptions are strongly correlated with the shape features of the content image and the texture features of the style image. We append more images at the end.

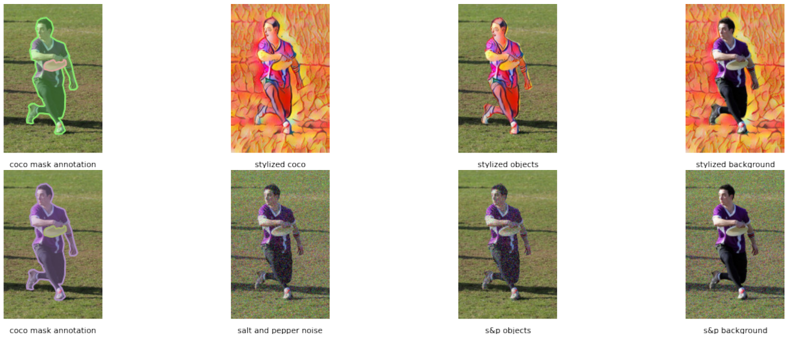

coco mask annotation
stylized coco
stylized objects
stylized background
coco mask annotation
salt and pepper noise
s&p objects
s&p background

Fig. 11: Comparison of Stylized COCO and Salt-and-Pepper COCO

To quantify the difference, we compare the structural similarity and color distance of Salt-and-Pepper COCO in Figure 12 (colored dots). We can observe

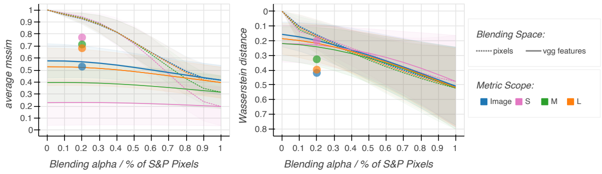


Fig. 12: Left: Average structural similarity between image gradients in relation to COCO (a score of 1 means that there is no difference between images). Right: Wasserstein distance between RGB histograms (reversed y-axis)

that the structural similarity of Salt-and-Pepper COCO is closer to the original data whereas the color distance is comparable to Stylized COCO at around $\alpha = 0.7$. A qualitative comparison to the *extreme* points of Stylized COCO ($\alpha = 0$ and $\alpha = 1$) is shown in Figure 13. Arguably, all three are clearly different types of image corruption to the human observer.

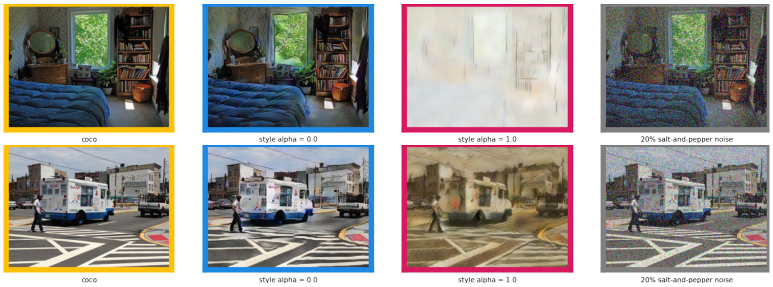


Fig. 13: Style transfer and impulse noise are different corruption types

4.1 Results

We evaluate a representative subset of models from the main paper on Salt-and-Pepper COCO. The results are displayed in Figure 14 in comparison to Stylized COCO. The first finding we like to highlight is that S&P noise and style $\alpha = 1$ have a strikingly similar effect on model performance. The image corruptions at style $\alpha = 0$ are clearly less severe in contrast. The second finding concerns the difference between the full (●) and object masked (▲) dataset versions. More precisely, models benefit notably more from masked style transfer at $\alpha = 1$ than from masked S&P noise.

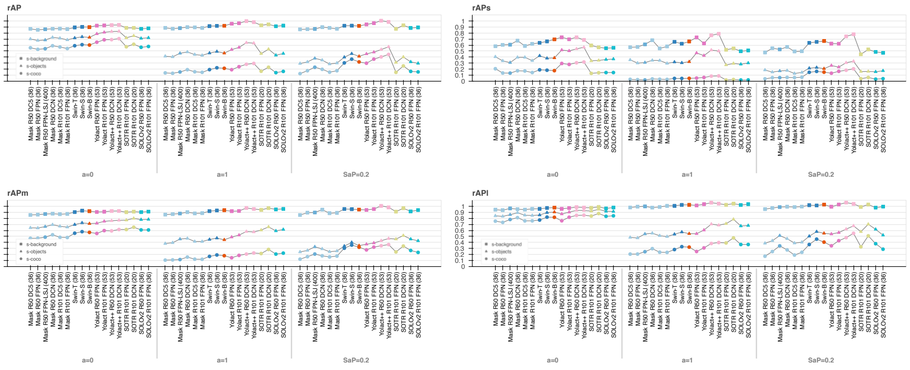
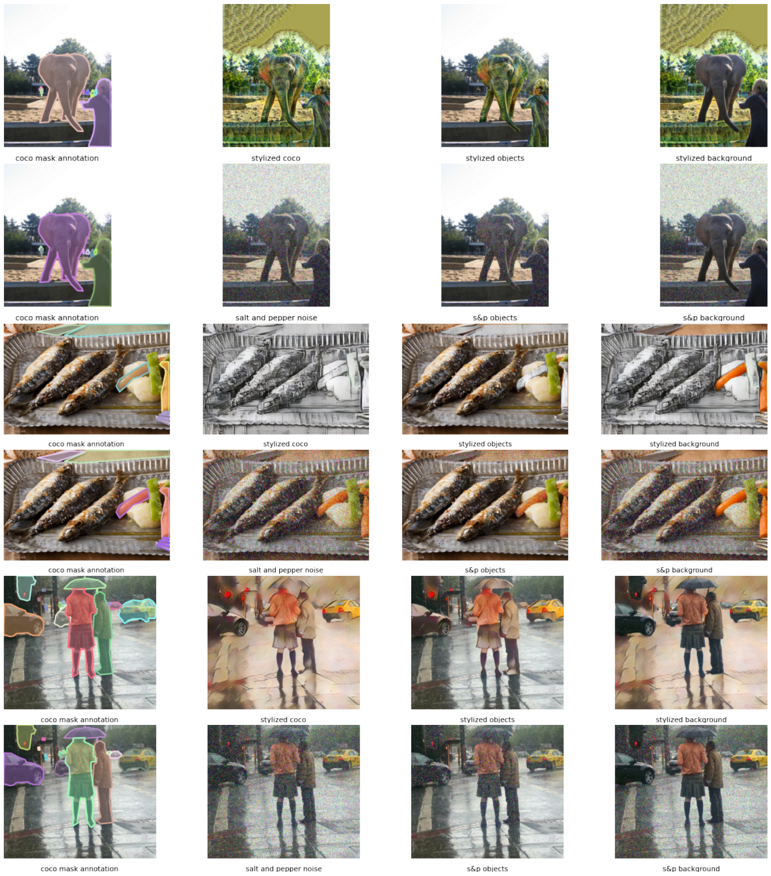


Fig. 14: Results on S&P COCO in comparison to Stylized COCO

4.2 Conclusion

Regarding our first question, we have shown quantitatively and qualitatively that style transfer is a unique *corruption type* and not comparable to the common impulse noise corruption. In general, model robustness against both types appears to be correlated. At closer inspection however, a slightly different ranking can be observed. Regarding our second question we conclude that object contour is indeed an important and exploitable feature for segmentation models. We derive this claim from the fact that models do benefit from masked style $\alpha = 1$ but can not exploit masked Salt-and-Pepper noise to a similar extend. This validates our causally motivated approach of object-centric texture masking and shows that masking corruption types with an independence assumption can not provide similar insights about semantic model robustness.


coco mask annotation, stylized coco, stylized objects, stylized background

coco mask annotation, salt and pepper noise, s&p objects, s&p background

coco mask annotation, stylized coco, stylized objects, stylized background

coco mask annotation, salt and pepper noise, s&p objects, s&p background

coco mask annotation, stylized coco, stylized objects, stylized background

coco mask annotation, salt and pepper noise, s&p objects, s&p background

Fig. 15: Stylized COCO and Salt-and-pepper COCO in comparison

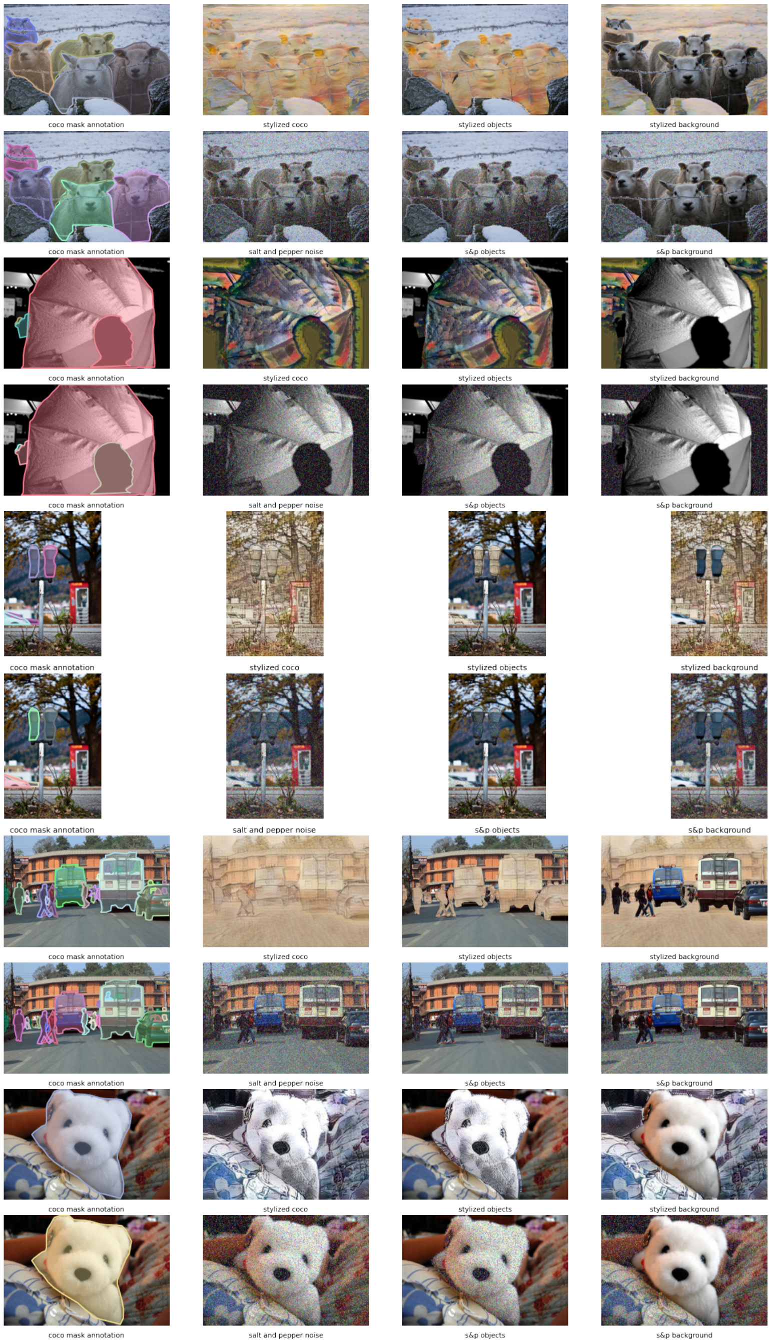


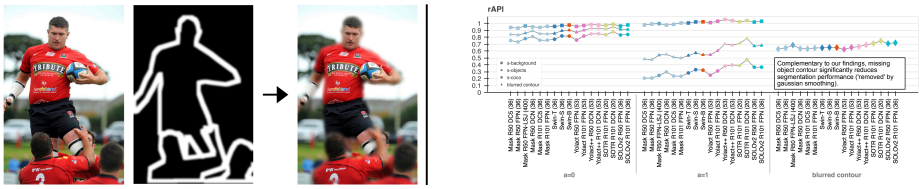
Fig. 16: Stylized COCO and Salt-and-pepper COCO in comparison

5 Ablation Study: Removing Object Contour

As a result of section 4, we saw that models for instance segmentation can exploit object contour independently from, and despite of, out-of-distribution texture. In this ablation, we perform the complementary experiment to answer the question:

1. Which feature is more important? Object texture or object contour?

The data set creation process is displayed in Figure 17 left. We use the ground truth annotations (mask polygons) to create a slightly thicker soft mask around the object contour. This area is then blurred by gaussian smoothing to *remove* the actual object contour. Object texture is not changed in this experiment. Note that we only consider large instances. The problem with small and medium sized objects is that their mask annotations often contain small but essential parts, e.g. the legs of a horse etc. We experimented with different parameters for the soft mask but eventually decided against these settings since contour removal resulted in undesired texture or part removal too frequently.



Complementary to our findings, missing object contour significantly reduces segmentation performance ('removed' by gaussian smoothing).

Fig. 17: *Left:* Creation process of the contourless COCO dataset. We use a soft mask around the ground truth mask annotations and gaussian smoothing to remove the contour of large instances. *Right:* Results on contourless COCO in comparison to Stylized COCO and its variants.

The results on contourless, large instances are shown in Figure 17 right (blurred contour). Two observations can be made. The first is that rAPl performance drops significantly for all models which confirms object contour as an important feature. The second observation allows us to answer our initial question. By comparing the results on blurred contour to the result on Stylized Objects at $\alpha = 1$, we can see that models within the Mask R-CNN framework as well as YOLACT models are more affected by out-of-distribution texture than contour removal (worse performance on s-objects, $\alpha = 1$ in comparison to blurred contour). In contrast, YOLACT++ as well as SOTR and SOLOv2 models appear to be more *balanced* in this regard. In line with the key findings of our main study, we conclude that the latter frameworks are more robust to novel object texture and we like to point out that all three, contain dynamic convolution operations of some sort.

References

1. Fan, Q., Ke, L., Pei, W., Tang, C.K., Tai, Y.W.: Commonality-Parsing Network Across Shape and Appearance for Partially Supervised Instance Segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020, vol. 12353, pp. 379–396. Springer International Publishing, Cham (2020)