AN OBJECT-CENTRIC SENSITIVITY ANALYSIS OF DEEP LEARNING BASED INSTANCE SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In this study we establish a comprehensive baseline regarding the object-centric robustness of deep learning models for instance segmentation. Our approach is motivated by the work of Geirhos et al. (2019) on texture bias in CNNs. However, we do not compare against human performance but instead incorporate ideas from object-centric representation learning. In addition, we analyze and control the effect of strong stylization that can lead to disappearing objects. The result is a stylized and object-centric version of MS COCO on which we perform an extensive sensitivity analysis regarding visual feature corruptions. We evaluate a broad range of frameworks including Cascade and Mask R-CNN, Swin Transformer, YOLACT(++), DETR, SOTR and SOLOv2. We find that framework choice, data augmentation and dynamic architectures improve robustness whereas supervised and self supervised pre-training does surprisingly not. In summary we evaluate 63 models on 61 versions of COCO for a total of 3843 evaluations.

1 INTRODUCTION

In this study, we want to learn how deep learning models for instance segmentation perform under object-centric corruptions. By object-centric, we mean corruptions that change the appearance of visual objects at an instance level. More precisely, we investigate the impact of **color**, **texture**, **shape** and **contour** which we consider to be the most defining feature types of visual objects.

- Our first goal is to understand the importance of each feature type, *finding potential biases of the learned representations*.
- Our second goal is to understand the robustness to feature corruptions, *finding potential abstractions such as texture invariance*.

Together we aim to gain a better understanding if, and to what extend we can expect *out of distribution* generalization regarding these feature types in an object-centric manner.

Our method is motivated by the work of Baker et al. (2018) and Geirhos et al. (2019) on texture bias in classification models. More precisely, we adopt the idea of Stylized-ImageNet (SIN) by Geirhos et al. (2019). To test deep learning models for instance segmentation, we use a stylized version of MS COCO (Lin et al., 2014) as shown Figure 1 left. In addition, we create two object-centric versions of the stylized COCO dataset as displayed in Figure 1 right. This change is motivated by work on object-centric representation learning (Greff et al., 2020). While technically straightforward, it allows us to distinguish between the corruption of actual objects and context information, i.e. the background. We will motivate this approach in more detail and show the empirical difference between holistic and object-centric stylization. Finally, we address a problem that can occur with strong stylization by controlling the strength of the style transfer. In summary, we present an object-centric sensitivity analysis of deep learning models for instance segmentation.

1.1 MOTIVATION AND BACKGROUND

For image classification, Baker et al. (2018) and Geirhos et al. (2019) found that convolutional neural networks (CNNs) trained on ImageNet (IN) are biased towards textures and mostly ignore object shapes, i.e. ignore the defining structure of an object. Brendel & Bethge (2019) have shown

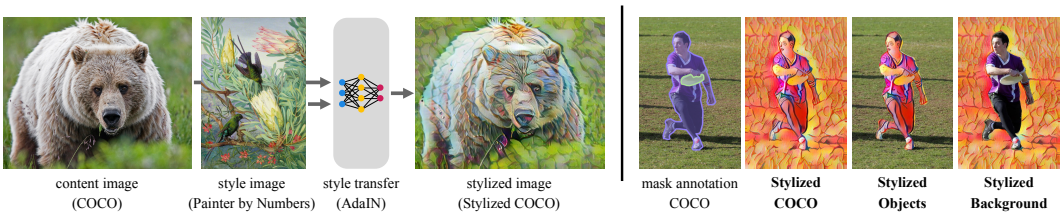


Figure 1: Left: Simplified creation process of the Stylized COCO dataset. Style images a randomly chosen from Kaggles Painter by Numbers dataset. Right: We use mask annotations to create additional, object-centric versions of Stylized COCO.

that CNNs can robustly classify objects in texturized images where the global appearance of objects is fully mixed up. Based on these findings, Geirhos et al. (2019) hypothesize that "local textures [...] provide sufficient information about object classes [and] ImageNet object recognition could, in principle, be achieved through texture recognition alone".

- **Texture hypothesis**: "Object textures are more important than global object shapes for CNN object recognition." - (Geirhos et al., 2019)

As pointed out by the same authors, the texture hypothesis stands in contrast to the "widely accepted intuition [...] that CNNs combine low-level features (e.g. edges) to increasingly complex shapes (such as wheels, car windows) until the object (e.g. car) can be readily classified".

- **Shape hypothesis**: Intermediate CNN layers recognize "parts of familiar objects, and subsequent layers [...] detect objects as combinations of these parts" - (LeCun et al., 2015)

To resolve this contradiction, Geirhos et al. (2019) introduced the Stylized-ImageNet (SIN) dataset. In SIN, global object shapes are mostly preserved while local texture cues are replaced by the AdaIn style transfer (Huang & Belongie, 2017). This way, texture cues are no longer predictive for a specific class and the CNN is *forced* to use shape information instead. Testing a ResNet-50 on IN and SIN, the authors could show that representations trained on IN do not generalize to SIN (16.4% top-5) while SIN representations generalize fairly well to IN (82.6% top-5)[1]. This indicates that, "a ResNet-50 architecture can indeed learn to recognize objects based on object shape, revealing that the texture bias in current CNNs is not by design but induced by ImageNet training data (Geirhos et al., 2019)". As a possible explanation, the authors further hypothesize "that standard ImageNet-trained models may be taking a 'shortcut' by focusing on local textures, which could be seen as a version of Occam's razor: If textures are sufficient, why should a CNN learn much else?".

We agree and argue that a similar argument can be made for the task of object detection, i.e. it is not inevitably necessary to represent objects in a holistic way to regress a bounding box. By holistic we mean that the representation includes all features that belong to a ground truth object and not only those of a specific but highly predictive part. Instance segmentation on the other hand fulfills this requirement. In consequence, we are interested in the robustness of instance segmentation models against gradually corrupted object features such as texture, shape or contour.

From the perspective of object-centric representation learning, instance segmentation can be seen as the first stage of a larger symbol binding process (Sun, 1992; Greff et al., 2020). More precisely, it solves *the segregation problem* that is, "the process of structuring raw sensory information into meaningful entities (Greff et al., 2020)". In this branch of research, segregation is typically embedded within unsupervised, disentangled representation learning (Higgins et al., 2018) since supervised models tend to struggle with the enormous variability in object appearances (Yuille & Liu, 2018; Recht et al., 2019). Disentangled representations and its object-centric extension have shown to improve efficiency and generalization on down-stream tasks such as *relational reasoning* and *reinforcement learning* (Higgins et al., 2017; van Steenkiste et al., 2019; Dittadi et al., 2021). They are also considered to be crucial towards *causal representation learning* (Schölkopf et al., 2021). On the downside, models from this group only perform well on synthetic data such as CLEVR (Johnson et al., 2017) and scaling them up is a major open challenge. With this in mind,

[1] For comparison: IN $\rightarrow$ IN: 92.9% top-5, SIN $\rightarrow$ SIN: 79.0% top-5 (Table 1 in Geirhos et al. (2019).

we take the opposite approach and investigate high performing supervised models and their object-centric robustness on natural images.

2 METHODS

In this section we describe the datasets, frameworks and models that are used in this study. The code to reproduce the results as well as $\approx$ 1TB of detection and evaluation data can be found here: `link-to-project-page-when-published`.

2.1 AN OBJECT-CENTRIC VERSION OF STYLIZED COCO

Stylized COCO as shown in Figure 1 left is an adaptation of Stylized-ImageNet by Geirhos et al. (2019). It was first used by Michaelis et al. (2019) as data augmentation technique during training to improve robustness of detection models against common corruption types such as gaussian noise or motion blur. We instead use a stylized version of the `val2017` subset to test instance segmentation models on this data directly. By manual inspection of Stylized COCO, we found that strong stylization can sometimes lead to images where the object contour starts to vanish up to the point where objects and their boundaries dissolve completely. This effect depends on the style image used in the creation process but affects objects of all scales alike. As shown in Figure 2, we resolve this issue by using the ground truth mask annotations to limit the style transfer to the actual objects or the background. This not only ensures that object contours are preserved but also controls for global stylization as a confounding variable. By assuming an object-centric causal model, Stylized COCO allows us to ask interventional questions regarding the original COCO dataset, e.g. *"What happens if we change the texture of images?"*. By masking the style transfer to objects or background, we can also ask counterfactual questions such as *"Was it actually the object that caused the change in performance?"*, *"What if we change the background instead?"*. We will refer to the different dataset versions as Stylized COCO (●), Stylized Objects (▲) and Stylized Background (■).

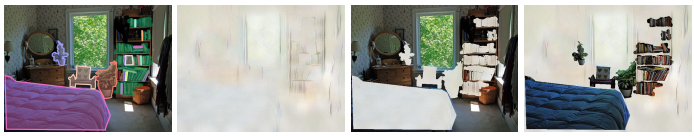


Figure 2: Depending on the style image, object boundaries can vanish due to strong stylization. The Stylized Objects and Background versions of Stylized COCO resolve this issue.

A related, second problem that remains after masking is that shape information within the object can also be lost due to strong stylization. We address this issue by controlling the strength of the AdaIN style transfer. This can be done with an α parameter that acts as a mixing coefficient between the content and style image. More precisely, AdaIN employs a pre-trained VGG encoder f on both images, performs an interpolation step between the resulting feature maps and produces the final output with a learned decoder network g. In summary, a stylized image t is produced by $T(c, s, \alpha) = g((1-\alpha)f(c) + \alpha\text{AdaIN}(f(c), f(s)))$ where c and s are the content and style images respectively. We will refer to this method as blending in *feature space*. The top row of Figure 3 shows two examples of the extreme points $\alpha = 0$ and $\alpha = 1$. Note that at $\alpha = 0$, the image colors are mostly preserved but the algorithm has already introduced artifacts in the form of subtle texture and shape changes. In response we create a control group where we perform alpha blending between the pixel values of the original content image c and the fully stylized image $t_{\alpha=1}$: $P(c, t_{\alpha=1}, \alpha) = ((1-\alpha) * c + \alpha * t_{\alpha=1})$. We will refer to this method as blending in *pixel space*. In contrast to the feature space sequence, the control group should preserve textures and object shape over a longer range. The idea is to compare models on both sequences in order to attribute performance changes to either color and texture or shape, depending on the objects size. In contrast to Geirhos et al. (2019) who used a fixed style strength to test IN features ($\alpha = 1$), we produce the full alpha-range $\alpha \in (0.0, 0.1, 0.2, ..., 1.0)$ for both blending spaces. Note that every alpha value depicts a separate and complete version of the accordingly styled COCO `val2017` subset. The qualitative differences can be inspected in Figure 3 bottom left (zoom in for better visibility).

We also went for quantitive measures to validate our subjective impression of Stylized COCO. Figure 3 bottom right shows a comparison of image gradients and RGB histograms at the *extreme* points. Compared to the original image we can see the subtle shape changes in the gradient map of

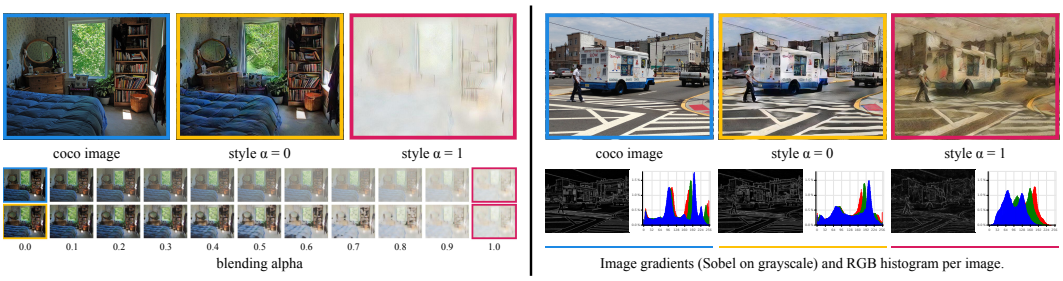


Figure 3: Top row: Comparison of COCO and Stylized COCO at different alphas. The AdaIN method introduces subtle artifacts even at $\alpha = 0$ (no style). Bottom left: We control the style strength in feature space (yellow to pink) and pixel space (blue to pink). Note that every alpha value in these sequences depicts a separate and complete version of the accordingly styled COCO `val2017` subset. Bottom right: Comparison of image gradients and color histograms at different alphas.

$\alpha = 0$ and a significantly different color histogram at $\alpha = 1$. To describe this effect over the full alpha range, we compute the structural similarity index (SSIM) (Wang et al., 2004) between gradient images of corresponding image pairs. Between RGB histograms we compute the Wasserstein distance alike. We always compare against the original COCO data and report the mean distance averaged over the full dataset at a specific alpha. In addition to the image-to-image scores we also include an instance level comparison for the COCO scales S,M and L. Instances have been cropped based on bbox information. This addition was added after we observed that small objects appear to be more affected by the AdaIN artifacts compared to medium and large instances. Figure 4 displays the results and confirms our assumption. Structural similarity depends on object size and is in fact, almost constant over the full feature space range for small objects. Furthermore, the control group preserves structural similarity over a longer range as intended. Color distance in contrast converges at around $\alpha = 0.3$. Based on these insights, we feel confident to better attribute potential performance dips and subsequently, determine the relative importance of each feature type.

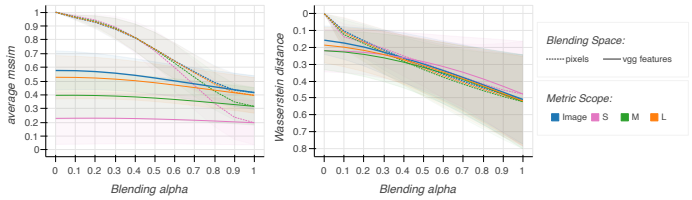


Figure 4: Left: Average structural similarity between image gradients in relation to COCO (a score of 1 means that there is no difference between images). Right: Wasserstein distance between RGB histograms (reversed y-axis).

2.2 Model Selection

To establish a comprehensive overview of model robustness, we have opted for a broad comparison of popular frameworks and architectures. With this approach, we hope to contribute a solid baseline on Stylized COCO that enables informed decisions on future research directions. The dimensions we consider to be impactful are **framework**, **architecture** and **pre-training**. The selected combinations of framework and architecture can be found in Table 1.

Frameworks for instance segmentation can be categorized in different ways. A first distinction can be made between methods that solve the detection problem as a refinement process of box proposals (multi stage) and methods that predict bounding boxes directly (one stage). We include the popular multi-stage frameworks Mask R-CNN (He et al., 2017) and Cascade Mask R-CNN (Cai & Vasconcelos, 2018) that uses multiple refinement stages instead of one. A second distinction concerns the use of predefined anchor boxes. Anchor based methods predict relative transformations on these priors whereas anchor free methods predict absolute bounding boxes. We include YOLACT(++) (Bolya et al., 2019; 2020) as a one-stage, anchor based framework. YOLACT is a real-time method that solves instance segmentation without explicit localization (feature pooling). Instead, it generates prototype masks over the entire image which are combined with per-instance mask coefficients to form the final output. The (++) version improves by adding a mask re-scoring branch (Huang et al., 2019a) and deformable convolutions (v2) (Dai et al., 2017; Zhu et al., 2019). We include DETR (Carion et al., 2020) as a one-stage, anchor free framework that formulates object

detection as a set prediction problem over image features. Note that it was not primarily designed for instance segmentation but offers a corresponding extension that we use in our study. Finally, we distinguish between top down frameworks where detection precedes segmentation and bottom up methods where bounding boxes are derived from mask predictions. We include the bottom-up methods SOLOv2 (Wang et al., 2020c) and SOTR (Guo et al., 2021). SOLO (Wang et al., 2020b) divides the input into a fixed grid and predicts a semantic category and corresponding instance mask at each location. The final segmentation is obtained with non-maximum-suppression on the gathered grid results to resolve similar predictions of adjacent grid cells. SOLOv2 improves by introducing dynamic convolutions to the mask prediction branch, i.e. an additional input dependent branch that dynamically predicts the convolution kernel weights. A similar idea was used by Tian et al. (2020b). SOTR uses a twin attention mechanism (Huang et al., 2019b) to model global and semantic dependencies between encoded image patches. The final result is obtained by patch wise classification and a multi-level upsampling module with dynamic convolution kernels for mask predictions, similar to SOLOv2. For completeness, we also include YOLO(v3,4 and scaled v4) to our comparison since detection is a vital sub-task of top down frameworks (Redmon & Farhadi, 2018; Bochkovskiy et al., 2020; Wang et al., 2021).

Architectures used in instance segmentation can be divided into backbone, neck and functional heads. The latter output the final results and are framework specific in most cases. Backbones and necks however are typically chosen from a pool of established models which allows for a controlled comparison. The role of backbone networks is to extract meaningful feature representations from the input, i.e. to encode the input. The neck modules define which representations are available to the functional heads, i.e. define the information flow. We include the CNN backbones ResNet (He et al., 2016), ResNext (Xie et al., 2017) and RegNet (Radosavovic et al., 2020), a network found with meta architecture search that outperforms EfficientNet (Tan & Le, 2019). Note that DETR and SOTR are hybrid frameworks that use transformer architectures to process the encoded backbone features. With Swin Transformer (Liu et al., 2021) we also include a convolution free backbone alternative based on the Vision Transformer approach (ViT) (Dosovitskiy et al., 2021). The most popular neck choice is the Feature Pyramid Network (FPN) (Lin et al., 2017). It builds a hierarchical feature representation from intermediate layers to improve performance at different scales, e.g. small objects. For comparison we also include a ResNet conv4 neck (C4) as used in Ren et al. (2015) and a ResNet conv5 neck with dilated convolution (DC5) as used by Dai et al. (2017). Finally we abbreviate FPN models that use deformable convolutions as DCN (Dai et al., 2017; Zhu et al., 2019). Similar to dynamic convolutions which predict kernel weights, DCNs learn to dynamically transform the sampling location of the otherwise fixed convolution filters.

Pre-training of backbone networks is commonly done as supervised learning on ImageNet (IN). Due to the recent success of self supervised learning (SSL) in classification, we are interested in how these representations perform in terms of object-centric robustness. In particular we are interested in the contrastive learning framework that seeks to learn "representations with enough invariance to be robust to inconsequential variations (Tian et al., 2020a)". Based on availability we include the methods InstDis (Wu et al., 2018), MoCo (He et al., 2020; Chen et al., 2020), PIRL (Misra & van der Maaten, 2020) and InfoMin (Tian et al., 2020a). Note that pre-trained backbones were only used as initialization for a supervised training on COCO. As a final comparison we include models that have been trained with random initialization and Large Scale Jittering (LSJ) (Ghiasi et al., 2021) data augmentation as an alternative to pre-training.

The complete list of models can be inspected in Figure 5. From the overview in Table 1 we can see that our selection allows for a fair comparison of frameworks as long as we fix the backbone and neck architecture to ResNet+FPN. Vice versa we can compare backbone and neck combinations within a fixed framework, in particular Mask R-CNN. Note that we also compared different learning schedules but did not include this dimension more prominent in the categorization after finding it to be the least significant factor in our evaluation.

3 Related Work

We are primarily motivated by the work of Geirhos et al. (2019) on texture bias in CNNs. We adapt the idea of Stylized Imagenet and extend it to the task of instance segmentation. More precisely we

Table 1: Overview of selected frameworks and architectures. Swin Transformer use hierarchical representations similar to FPN necks in CNNs. RegNetY additionally implements the Squeeze-and-Excitation operation (Hu et al., 2018). Yolo consists of darknet (D), spatial pyramid pooling (SSP) (He et al., 2014) and a Path Aggregation Network (PAN) (Liu et al., 2018) in varying combinations with CSPNet (C) (Wang et al., 2020a).

Architecture	**Framework**						
CNN	multi stage		one stage				
Hybrid	anchor based				anchor free		
ViT	top down (bbox→segm)					bottom up (segm→bbox)	
	Mask R-CNN	Cascade	YOLO(v3,4+)	YOLACT(++)	DETR	SOTR	SOLOv2
R50	FPN, C4, DC5, DCN	FPN	-	FPN, DCN	FPN, DC5	-	FPN
R101	FPN, C4, DC5,	DCN	-	FPN, DCN	FPN	FPN, DCN	FPN
X101	FPN	-	-	-	-	-	-
X151	-	FPN, DCN	-	-	-	-	-
RegNetX	FPN	-	-	-	-	-	-
RegNetY	FPN	-	-	-	-	-	-
Swin-T	FPN*	FPN*	-	-	-	-	-
Swin-S	FPN*	FPN*	-	-	-	-	-
Swin-B	-	FPN*	-	-	-	-	-
D53	-	-	FPN	-	-	-	-
CD53	-	-	(C)PAN, SPP	-	-	-	-

use a stylized version of MS COCO for our experiments as introduced by Michaelis et al. (2019)[2] for object detection. Classical robustness research is usually more focused on common corruption types like gaussian noise or other image holistic distortions (Hendrycks & Dietterich, 2019; Michaelis et al., 2019; Kamann & Rother, 2021; Mummadi et al., 2021). A popular alternative approach is to compare model performance and failure cases against humans (Geirhos et al., 2018; 2019; 2020; 2021; Shankar et al., 2020; Tuli et al., 2021). Geirhos et al. (2021) and Tuli et al. (2021) for instance find that Transformer models perform more consistent to humans than CNNs. Madan et al. (2021) on the other hand report that both model types are prone to small in-distribution changes in 3D perspective and lighting. We contribute to this matter by including both types in our comparison. More generally we see our approach as a third and complementary option to model robustness in the spirit of object centric-learning Greff et al. (2020); Dittadi et al. (2021). As an alternative to our sensitivity method, Islam et al. (2021) analyze feature importance in latent representations and Cao & Johnson (2021) use feature visualization to understand object detectors. In terms of alternative datasets we want to highlight the 3DB framework (Leclerc et al., 2021), a rendering engine that enables artifact free texture transfer on synthetic objects and SI-Score (Yung et al., 2021), a dataset for analyzsing robustness to rotation, location and size.

4 RESULTS

In this section we present the evaluation results on Stylized COCO, Objects and Background. Each dataset version contains 20 copies of the accordingly styled COCO `val2017` subset. We first provide an overview of the full blending sequence and then focus on the *extreme* points $\alpha = 0$ and $\alpha = 1$. In total we tested 63 pre-trained models [3] on 61 datasets which sums up to 3843 evaluations. We primarily focus on the Average Precision (AP) for segmentation and the scale dependent COCO metrics APs, APm and APl, due to our object-centric perspective. More metrics and the corresponding bbox scores can be found in the Appendix, together with a complete distance matrix that we used to decide on model grouping and the selection of detailed comparisons. We display the reproduced[4], absolute performance on COCO `val2017` in Figure 5 for all models. As can be seen, training schedule, data augmentation and architecture choice have the biggest impact within a framework with RegNet and Swin Transformer performing best. Note that SOTR and SOLO have worse APs but better APl compared to other frameworks (see Appendix).

We now report the main results of our study. To quantify object-centric robustness, we report the relative performance in comparison to the uncorrupted data. For every dataset version, blending

[2] Stylized Datasets Code: `https://github.com/bethgelab/stylize-datasets`

[3] We use model weights from the following projects:
Detectron 2: `https://github.com/facebookresearch/detectron2/`
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`
SOTR: `https://github.com/easton-cau/SOTR`
SOLOv2: `https://github.com/aim-uofa/AdelaiDet/tree/master/configs/SOLOv2`

[4] Methods that do not report scores for `val2017` have been validated on `test-dev2017`.

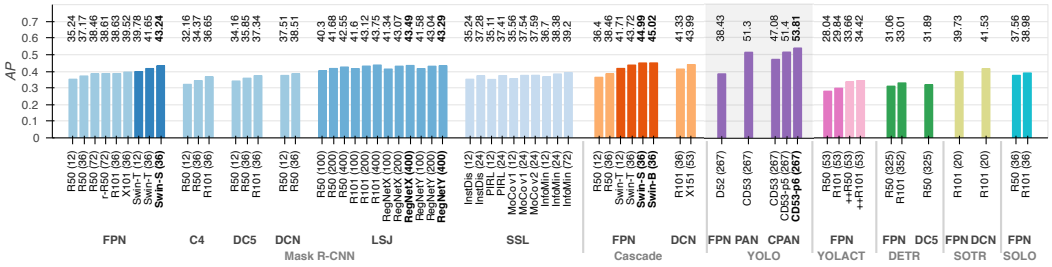


Figure 5: Absolute performances on COCO `val2017`. Training schedules in epochs have been appended to model names. Note that Yolo is bbox AP which is not directly comparable but included for model completeness.

space and alpha value we calculate: $rP_\alpha = P_\alpha / P_{coco}; P \in \{AP, APs, APm, APl\}$. Figure 6 compares all models on the stylized COCO versions, averaged by framework. For better visibility we display the pixel space control group slightly faded (always starting from 100%). From the first row we can tell that all models follow a similar trajectory. Furthermore, there appears to exist a clear ordering between frameworks which we investigate in more detail in Figure 7. In terms of object-centric performance we can see that models perform significantly better on Stylized Objects compared to the fully Stylized COCO. We conclude that object contour is in fact taken into account by instance segmentation models. An alternative explanation might be that models have learned to ignore background patches. By inspecting the performance on Stylized Background however we can tell that corrupted background features, e.g. context information has little impact compared to object-centric features. Note that this is not the case for small objects where models appear to depend much more on context information compared to medium and large objects. From the gap between the *pixel space* control group and the *feature space* blending sequence we can tell that subtle shape and texture corruptions are more severe than color changes. Performance on small objects is in fact almost fully dominated by the artifacts of the AdaIN method. For medium and large objects we observe a different behavior. Small corruptions to the original shapes and textures ($\alpha = 0$) are less

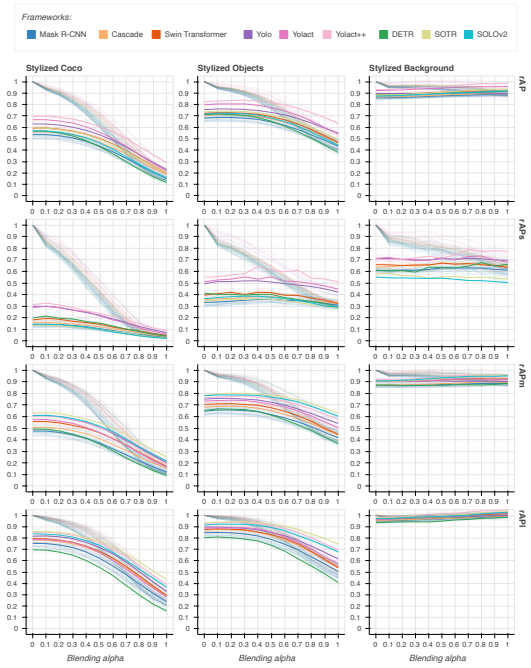


Figure 6: General overview of model robustness. Masking the style transfer effect to ground truth objects results in improved relative performance without changing the overall trend. Structural similarity as displayed in Figure 4 seems highly predictive for the low alpha range but can not explain the further decreasing robustness values. We also like to point out the small performance dip of the control group at $\alpha = 0.1$ that seems to be induced by color shift. Surprisingly, increasing the style strength on background features appears to have a positive effect for large objects (rAPl). Swin Transformer are display as a separate group due to their convolution free backbone architecture. Note that absolute metrics result in different ordering due to varying base performance. We append this version together with the corresponding bbox variants in the Appendix.

severe and the remaining performance is fairly robust until actual *out of distribution texture* from the style image is introduced (starting around $\alpha = 0.4$ in Stylized Objects). In reverse we conclude that texture is more important to instance segmentation models than (corrupted) object shape.

For a fair **comparison between frameworks** we fix the architecture to ResNet-50 FPN. Figure 7 displays the results from the feature blending sequence at $\alpha = 0$ and $\alpha = 1$. We can again observe that object-centric stylization is important but does not change the general ranking. Similarly, small objects are more directly affected by shape and texture artifacts whereas medium and large objects degrade due to texture transfer. Between Cascade and Mask R-CNN we find only minor differences, YOLACT(++) performs consistently better across all scales and SOLO and SOTR perform on par or even better but struggle with small objects. DETR shows the opposite behavior.

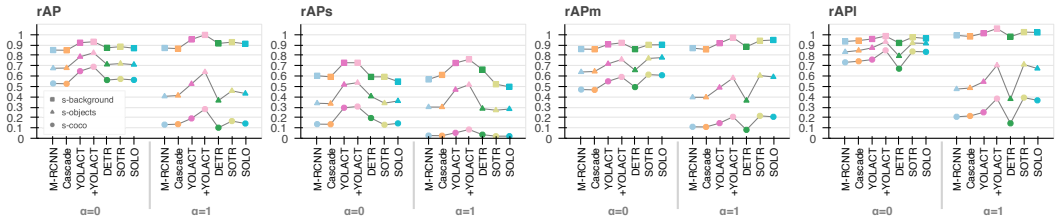


Figure 7: Object-centric robustness by framework. Note that we compromised on R-101 for SOTR, see Table 1.

To investigate **the impact of backbone architectures** we fix the neck method (FPN) and compare model pairs within a given framework. As can be seen from the rAP metric in Figure 8, deeper models perform generally slightly better. The rAPm and rAPl metrics follow this trend. We therefore append the results for rAPs were the behavior is less clear and surprisingly, often reversed. Note that Swin-T and S are comparable to ResNet-50 and 101 in model complexity and perform best within the Cascade and Mask R-CNN framework respectively.

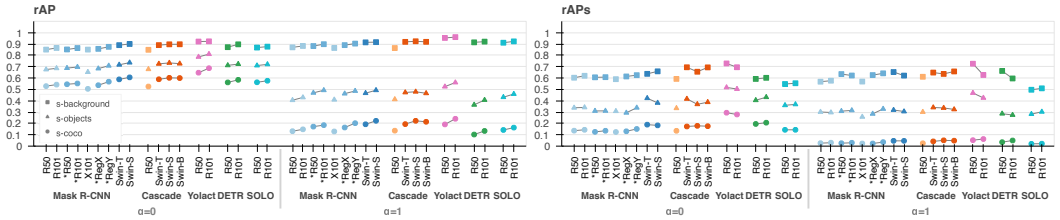


Figure 8: Object-centric robustness by backbone architecture. Models marked with * are trained with LSJ.

To investigate **the impact of neck architecture** we do the reverse and compare models by fixing backbones and framework. As can be seen from Figure 9, FPN necks are typically the least robust option. However, FPN in combination with deformable convolutions (DCN) performs consistently best. Note that SOTR also implements dynamic convolution kernels. For small objects, DC5 necks improve the robustness on Stylized Objects (triangle). We also compare Yolo against itself and find that surprisingly, the simple v4-csp model (CPAN) is the most robust option.

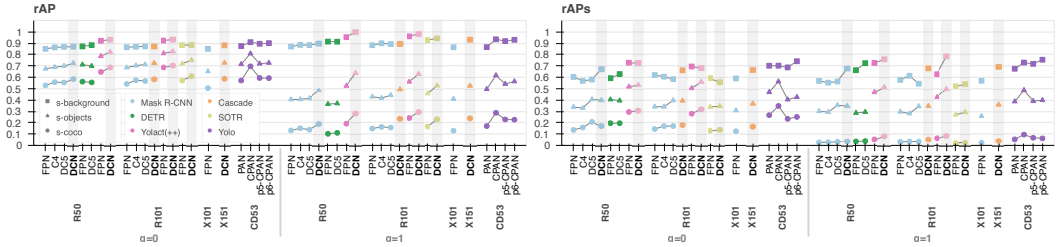


Figure 9: Object-centric robustness by neck architecture. Models with dynamic components are highlighted. Note that Yolo scores are bbox based and not comparable to other frameworks.

Finally we investigate **the impact of pre-training** and strong data augmentation. As can be seen from Figure 10, different types of initialization perform almost similar within a few percent. This is surprising since we expected representations to perform more differently based on the objective used in pre-training, in particular from the contrastive learning framework. On the other hand we can report that supervised and unsupervised representations perform almost on par. Similarly, longer training schedules have little impact on model robustness but do not lead to strong overgeneralization either. In comparison we find that random initialization with LSJ data augmentation performs best.

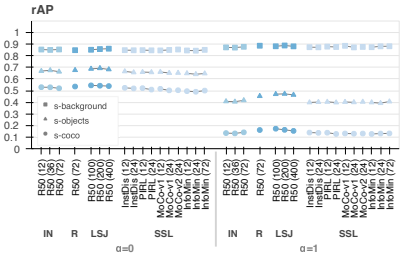


Figure 10: Object-centric robustness by pre-training and data augmentation.

5 Discussion

As pointed out in our motivation, robust instance segmentation is not only desirable for computer vision as an isolated task. Models that generalize in an object-centric way would largely contribute to a larger binding process that potentially enables more symbol like processing in deep neural networks (Greff et al., 2020). In our study we focus on the inherently complex problem of feature variability in natural images. We approach this task by controlling feature corruptions in an object centric way. As a result, we find that object contour plays a major role and that object specific corruptions are more severe than corrupted context information. This indicates that models do indeed learn some form of object-centric representation. However, we do not find signs of feature specific abstraction that would enable a robust segregation process. In line with (Geirhos et al., 2019), we find that deep learning models appear to focus on textures in general. Specifically, we show that models are fairly robust to shape and texture corruptions after an initial performance dip. The remaining loss can be attributed to actual texture transfer from an unknown domain. This effect appears to scale with instance size and we hypothesize that the difference between shape and texture simply collapses for (very) small objects in MS COCO. In contrast to Geirhos et al. (2019) who argue that this problem is induced by the ImageNet training data, we find no such correlation in terms of pre-training. We conclude that either a similar bias is induced by the COCO dataset or that architectures are simply not flexible enough to cope with feature variability, independent of training regime. The latter hypothesis was formulated by Greff et al. (2020) as **the binding problem** and concerns "the inability [of neural networks] to dynamically and flexibly combine (bind) information [... which] limits their ability to [...] accommodate different patterns of generalization". In our main results we find that framework choice has the largest impact on robustness and that interestingly, dynamic modules such as deformable and dynamic convolutions lead to improved robustness which we consider a promising direction for future research.

6 Conclusion

In this study we contribute a comprehensive baseline on the object-centric robustness of instance segmentation models. We first introduce a simple but well motivated object-centric version of Stylized COCO. We carefully control for and analyze different aspects of feature corruptions to enable an extensive sensitivity analysis. Based on a broad selection of frameworks and architectures, we test 63 models on 61 dataset versions for a total of 3843 evaluations. After a general overview we compare model robustness in more detail by framework, architecture and pre-training. Finally we discuss dynamic modules as a promising future research direction towards better, object-centric feature robustness.

Acknowledgments

We want to thank the authors of Detectron 2, PyContrast, Swin, YOLO, YOLACT, DETR, SOTR and SOLOv2 for their contribution of publicly available model weights. Without their commitment to open research, this study would not have been possible. We follow their example and publish all code and data used in this study to ensure reproducibility and more research moving forward.

REPRODUCIBILITY STATEMENT

The code to reproduce our results on Stylized COCO, Stylized Objects and Stylized Background can be found here: `link-to-project-page-when-published`. We also provide $\approx$ 1TB of detection and evaluation data. We include the latter to minimize GPU availability as a limiting factor for future research. To improve actual reproducibility, we provide automated environment setups for the dataset creation and each instance segmentation framework.

REFERENCES

Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J. Kellman. Deep convolutional networks do not classify based on global object shape. *PLOS Computational Biology*, 14(12): e1006613, December 2018. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1006613.

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934, 2020.

Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT++: Better real-time instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020. doi: 10.1109/TPAMI.2020.3014297.

Wieland Brendel and Matthias Bethge. Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet. In *International Conference on Learning Representations*, 2019.

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Ang Cao and Justin Johnson. Inverting and understanding object detectors. *CoRR*, abs/2106.13933, 2021.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, volume 12346, pp. 213–229. Springer International Publishing, Cham, 2020. ISBN 978-3-030-58451-1 978-3-030-58452-8. doi: 10.1007/978-3-030-58452-8_13.

Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2017.

Andrea Dittadi, Samuele Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello. Generalization and robustness implications in object-centric learning. *CoRR*, abs/2107.00637, 2021.

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.

Robert Geirhos, Carlos R. M. Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. Generalisation in humans and deep neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

Robert Geirhos, Kantharaju Narayanappa, Benjamin Mitzkus, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. On the surprising similarities between supervised and self-supervised models. 2020.

Robert Geirhos, Kantharaju Narayanappa, Benjamin Mitzkus, Tizian Thieringer, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Partial success in closing the gap between human and machine vision. *CoRR*, abs/2106.07411, 2021.

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)*, pp. 2918–2928, June 2021.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *CoRR*, abs/2012.05208, 2020.

Ruohao Guo, Dantong Niu, Liao Qu, and Zhenbo Li. SOTR: Segmenting objects with transformers. *CoRR*, abs/2108.06747, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, volume 8691, pp. 346–361. Springer International Publishing, Cham, 2014. ISBN 978-3-319-10577-2 978-3-319-10578-9. doi: 10.1007/978-3-319-10578-9_23.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. DARLA: Improving zero-shot transfer in reinforcement learning. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1480–1490, International Convention Centre, Sydney, Australia, August 2017. PMLR.

Irina Higgins, David Amos, David Pfau, Sébastien Racanière, Loïc Matthey, Danilo J. Rezende, and Alexander Lerchner. Towards a Definition of Disentangled Representations. *CoRR*, abs/1812.02230, 2018.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2017.

Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019a.

Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. CCNet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019b.

Md Amirul Islam, Matthew Kowal, Patrick Esser, Sen Jia, Björn Ommer, Konstantinos G. Derpanis, and Neil Bruce. Shape or texture: Understanding discriminative features in {CNN}s. In *International Conference on Learning Representations*, 2021.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1988–1997, Honolulu, HI, July 2017. IEEE. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.215.

Christoph Kamann and Carsten Rother. Benchmarking the Robustness of Semantic Segmentation Models with Respect to Common Corruptions. *International Journal of Computer Vision*, 129 (2):462–483, February 2021. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-020-01383-2.

Guillaume Leclerc, Hadi Salman, Andrew Ilyas, Sai Vemprala, Logan Engstrom, Vibhav Vineet, Kai Yuanqing Xiao, Pengchuan Zhang, Shibani Santurkar, Greg Yang, Ashish Kapoor, and Aleksander Madry. 3DB: A framework for debugging computer vision models. *CoRR*, abs/2106.03805, 2021.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature14539.

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 David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, volume 8693, pp. 740–755. Springer International Publishing, Cham, 2014. ISBN 978-3-319-10601-4 978-3-319-10602-1. doi: 10.1007/978-3-319-10602-1_48.

Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

Spandan Madan, Tomotake Sasaki, Tzu-Mao Li, Xavier Boix, and Hanspeter Pfister. Small in-distribution changes in 3D perspective and lighting fool both CNNs and Transformers. *CoRR*, abs/2106.16198, 2021.

Alexander Mathis, Thomas Biasi, Steffen Schneider, Mert Yüksekgönül, Byron Rogers, Matthias Bethge, and Mackenzie W. Mathis. Pretraining boosts out-of-domain robustness for pose estimation. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*, pp. 1858–1867. IEEE, 2021. doi: 10.1109/WACV48630.2021.00190.

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*, volume 190707484, July 2019.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Chaithanya Kumar Mummadi, Ranjitha Subramaniam, Robin Hutmacher, Julien Vitay, Volker Fischer, and Jan Hendrik Metzen. Does enhanced shape bias improve neural network robustness to common corruptions? In *International Conference on Learning Representations*, 2021.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollar. Designing network design spaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? volume 97 of *Proceedings of Machine Learning Research*, pp. 5389–5400, Long Beach, California, USA, June 2019. PMLR.

Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Evgenia Rusak, Steffen Schneider, Peter V. Gehler, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Adapting ImageNet-scale models to complex distribution shifts with self-learning. *CoRR*, abs/2104.12928, 2021.

B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Toward causal representation learning. *Proceedings of the IEEE - Advances in Machine Learning and Deep Neural Networks*, 109(5):612–634, 2021. doi: 10.1109/JPROC.2021.3058954.

Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. Evaluating machine accuracy on ImageNet. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8634–8644. PMLR, July 2020.

Ron Sun. On Variable Binding in Connectionist Networks. *Connection Science*, 4(2):93–124, January 1992. ISSN 0954-0091, 1360-0494. doi: 10.1080/09540099208946607.

Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114. PMLR, June 2019.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6827–6839. Curran Associates, Inc., 2020a.

Zhi Tian, Chunhua Shen, and Hao Chen. Conditional Convolutions for Instance Segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, volume 12346, pp. 282–298. Springer International Publishing, Cham, 2020b. ISBN 978-3-030-58451-1 978-3-030-58452-8. doi: 10.1007/978-3-030-58452-8_17.

Shikhar Tuli, Ishita Dasgupta, Erin Grant, and Thomas L. Griffiths. Are convolutional neural networks or transformers more like human vision? *CoRR*, abs/2105.07197, 2021.

Sjoerd van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are Disentangled Representations Helpful for Abstract Visual Reasoning? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 14222–14235. Curran Associates, Inc., 2019.

Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1571–1580, Seattle, WA, USA, June 2020a. IEEE. ISBN 978-1-72819-360-1. doi: 10.1109/CVPRW50498.2020.00203.

Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13029–13038, June 2021.

Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. SOLO: Segmenting Objects by Locations. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, volume 12363, pp. 649–665. Springer International Publishing, Cham, 2020b. ISBN 978-3-030-58522-8 978-3-030-58523-5. doi: 10.1007/978-3-030-58523-5_38.

Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. SOLOv2: Dynamic and fast instance segmentation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17721–17732. Curran Associates, Inc., 2020c.

Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. ISSN 1057-7149. doi: 10.1109/TIP.2003.819861.

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

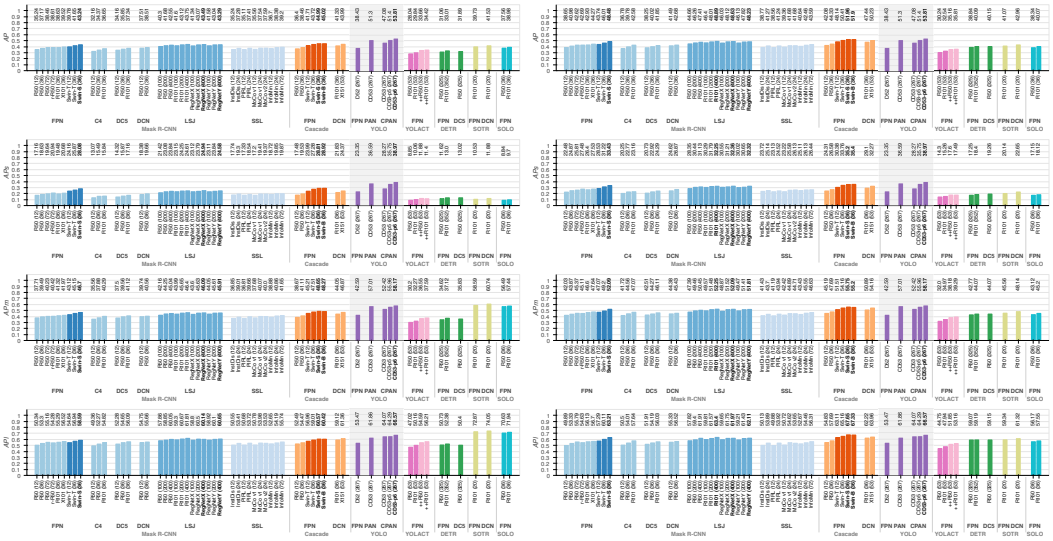
Alan Yuille and Chenxi Liu. Deep Nets: What have they ever done for Vision? Technical Report 88, Center for Brains, Minds and Machines (CBMM), MIT CSAIL, May 2018.

Jessica Yung, Rob Romijnders, Alexander Kolesnikov, Lucas Beyer, Josip Djolonga, Neil Houlsby, Sylvain Gelly, Mario Lucic, and Xiaohua Zhai. SI-Score: An image dataset for fine-grained analysis of robustness to object location, rotation and size. *CoRR*, abs/2104.04191, 2021.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable ConvNets V2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

APPENDIX

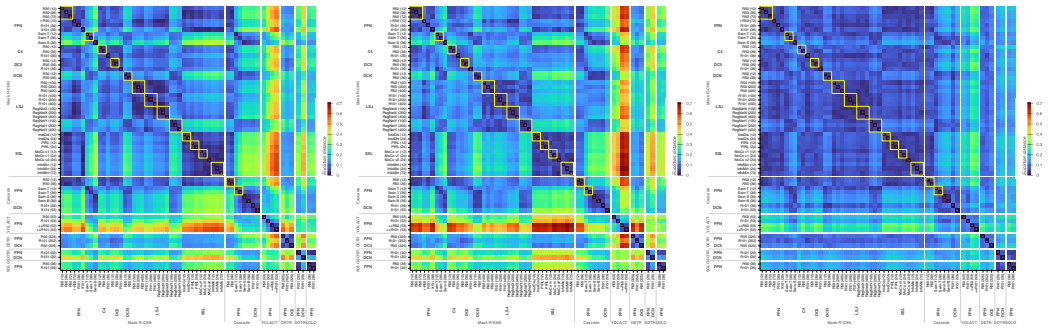
We include additional results for comparison, in particular the equivalent bounding box versions of our main paper figures. Figure 11 compares the absolute performance on COCO `val2017` between all models. The model grouping and detailed results we present in the main paper are based on the distance matrices in Figure 12. Figure 13 provides the bbox equivalent to the general overview on model robustness. We also include the absolute performance versions of those figures. Figure 14 compares the relative framework performance by bbox score. Figure 15 and Figure 16 display the comparisons of backbone and neck architectures respectively.



(a) Instance segmentation scores (YOLO is bbox)

(b) Bounding Box scores

Figure 11: Comparison of absolute model performance on COCO `val2017`. Methods that do not report scores for `val2017` have been validated on `test-dev2017`. Note that SOTR and SOLO perform worse on small objects but exhibit significantly improved segmentation scores for medium and large objects



(a) Stylized COCO

(b) Stylized Objects

(c) Stylized Background

Figure 12: Euclidian distance between the relative model performances (over the full alpha range). We average over AP, APs, APm and APl. Yellow squares group the same model with different learning schedule.

(a) Segmentation scores

(b) Bounding Box scores

Figure 13: General overview 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 order of frameworks due to varying base performance.

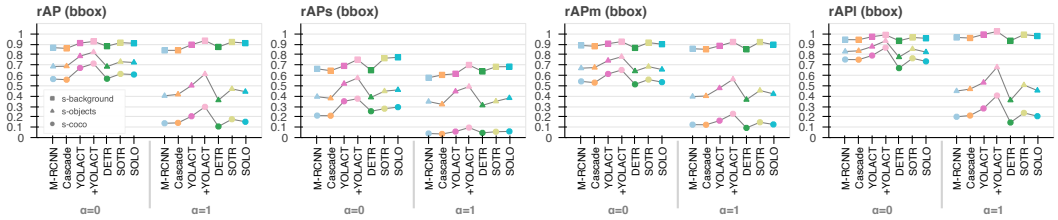
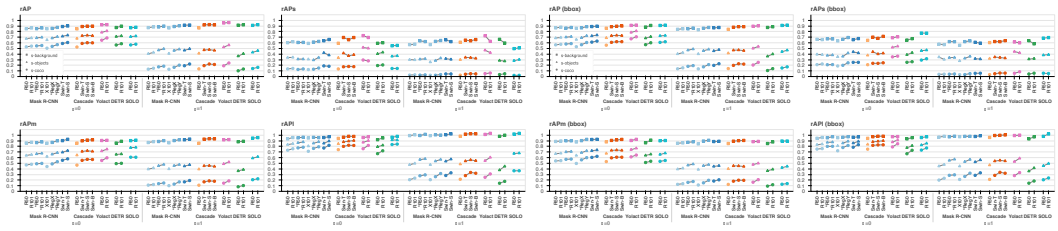
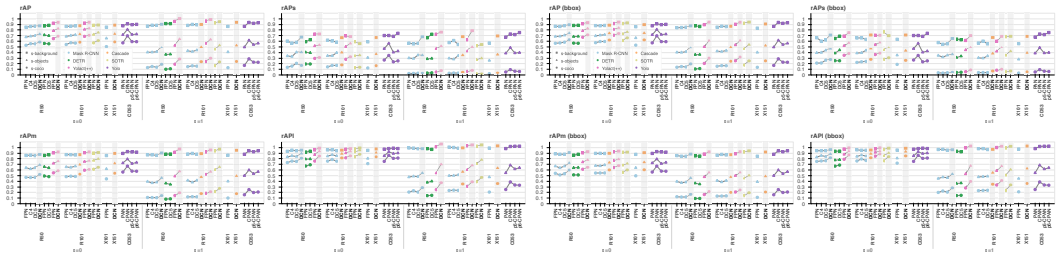


Figure 14: Object-centric robustness by framework (bbox). Note that we compromised on R-101 for SOTR.



(a) Segmentation scores (b) Bounding Box scores

Figure 15: Full comparison of object-centric robustness by backbone architecture. Models marked with * are trained with LSJ.



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

Figure 16: Full comparison of object-centric robustness by neck architecture. Models with dynamic components are highlighted.