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

Abstract. Do deep learning models for instance segmentation generalize to novel objects in a systematic way? For classification, such behavior has been questioned. In this study, we aim to understand if certain design decisions such as *framework*, *architecture* or *pre-training* contribute to the semantic understanding of instance segmentation. To answer this question, we consider a special case of robustness and compare pre-trained models on a challenging benchmark for object-centric, out-of-distribution texture. We do not introduce another method in this work. Instead, we take a step back and evaluate a broad range of existing literature. This includes Cascade and Mask R-CNN, Swin Transformer, BMask, YOLACT(++), DETR, BCNet, SOTR and SOLOv2. We find that YOLACT++, SOTR and SOLOv2 are significantly more robust to out-of-distribution texture than other frameworks. In addition, we show that deeper and dynamic architectures improve robustness whereas training schedules, data augmentation and pre-training have only a minor impact. In summary we evaluate 68 models on 61 versions of MS COCO for a total of 4148 evaluations.

Keywords: robust vision, instance segmentation, deep learning, object-centric, out-of-distribution, texture robustness

1 Introduction

In this study, we investigate a special case of robustness for deep learning based instance segmentation. More precisely, we want to learn how pre-trained models compare in the case of *out-of-distribution texture*, i.e. when learned objects contain textures that do not appear in the training data. In particular, we aim to understand if different *frameworks*, *architectures* and *pre-training schemes* contribute to model robustness in a systematic way. Despite their remarkable success in computer vision, deep neural networks still struggle in many challenging real-world scenarios [73,50,55,48]. One specific example are naturally adversarial objects [41]. Consider for instance a pedestrian with an unconventionally textured dress or a rare horse statue made out of bronze. The model might have seen many pedestrians or natural horses during training but still

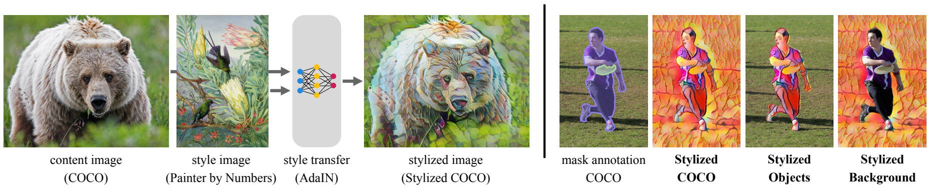

Fig. 1. Left: Simplified creation process of the Stylized COCO dataset. Style images are randomly chosen from Kaggles Painter by Numbers dataset. Right: We use mask annotations to create counterfactual, object-centric versions of Stylized COCO. We include more examples of the creation process in the supplementary material

fails to detect these rare or unseen examples, often with high confidence. Generalizing to such instances is typically described as *out-of-distribution robustness* [29,41]. For classification, [29] suggest that improvements in this direction are more likely to come from computer vision architectures than from existing data augmentation or additional public datasets. Motivated by these findings, we take a step back and perform an extensive comparison of existing literature. Since instance segmentation methods are quite complex, our goal is to unveil the impact of different components and design decision. As a result, our systematic baseline enables more informed design decisions regarding segmentation robustness in the future.

The problem space we consider is inspired by work from [1] and [19] on texture bias in convolutional neural networks (CNNs). Both groups found that when compared to humans, CNNs for classification ignore object shape in favor of local texture cues. In fact, [5] have further shown that CNNs can robustly classify objects in texturized images where the global appearance of objects is fully mixed up. Since human vision is fairly robust to novel texture and sensitive to object shape, we hypothesize that segmentation models with a similar bias will generalize in more systematic ways as well. As a first step in this direction, we want to learn if existing methods may contain components and design decisions that promote such behavior. In consequence, we opted to evaluate an extensive range of pre-trained models on a challenging but easy to understand edge case.

As shown in Figure 1 left, we utilize the AdaIN method [33] to create a stylized version of MS COCO [45]. The resulting dataset can be understood as a simulation of familiar objects with guaranteed novel texture, i.e. out-of-distribution texture. Crucially, it ensures that potentially confounding biases from the original data, such as class imbalance or specific view points, are preserved. The simulation is not perfect however. It introduces processing artifacts which we analyze in depth in our methods section. In addition, we control the strength of the style transfer and report results on the full range. This step is essential to distinguish between the effect of image corruption and actually novel object texture. In this sense, style transfer as a whole can be understood as a special type of image corruption. The statistical difference to classical corruption types, such as gaussian noise, is that the latter assumes the corruption to be independent from the signal. In style transfer, the *corruption* is by design not random and highly correlated with the shape and texture features of the content and style image respectively. Alternatives to our simulation approach are discussed in the related work part.

The complete benchmark setting is displayed in Figure 1 right. As can be seen, we utilize the segmentation labels to create two additional object-centric versions of Stylized COCO. The motivation for this step is twofold. First, the masking ensures that object contour is recovered in cases where strong stylization results in a camouflage setting. Second and more importantly, it is more plausible to limit the texture simulation to actual object instances instead of the full image. In addition to this causal justification, we are interested to see if pre-trained models are able to exploit this *implicit encoding* of ground truth information which is trivial to spot for the human observer. The Stylized Background dataset can be seen as a control group that allows us to measure the importance of context information. In general, we expect all pre-trained models to degrade with increasing out-of-distribution texture. The approach we take is therefore a negative test, i.e. if some models degrade significantly less than others we consider them more robust.

2 Methods

In this section we describe the datasets, frameworks and models that are used in this study. The code to reproduce our results and the resulting detection and evaluation data can be found here: `https://github.com/JohannesTheo/trapped-in-texture-bias`.

2.1 An object-centric version of Stylized COCO

Stylized COCO as shown in Figure 1 left is an adaptation of Stylized-ImageNet by [19]. It was first used by [50][3] as data augmentation during training to improve the robustness of detection models against common corruption types, e.g. gaussian noise or motion blur. We instead use a stylized version of the `val2017` subset to test pre-trained instance segmentation models directly on this data. 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 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

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

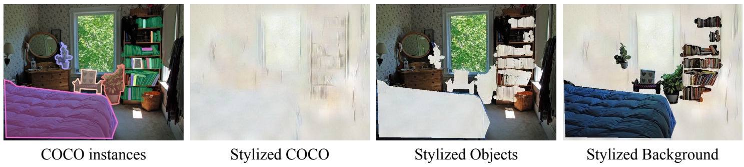

Fig. 2. Depending on the style image, object boundaries can vanish due to strong stylization. The Stylized Objects and Background versions of Stylized COCO resolve this issue

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 (■).

A second problem that remains 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 method. 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

$$T(c, s, \alpha) = g((1-\alpha)f(c) + \alpha \mathrm{AdaIN}(f(c), f(s))) \tag{1}$$

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 $\alpha = 0$ (no style) and $\alpha = 1$ (full style). 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 stylized image t at a specific alpha value:

$$P(c, t_\alpha, \alpha) = (1-\alpha) * c + \alpha * t_\alpha \tag{2}$$

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 to either image corruption or actual out-of-distribution texture. In contrast to [19] who used a fixed style strength to modify ImageNet 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 copy of the accordingly styled COCO `val2017` subset. The qualitative differences can be inspected in Figure 3 bottom left (zoom in for better visibility).

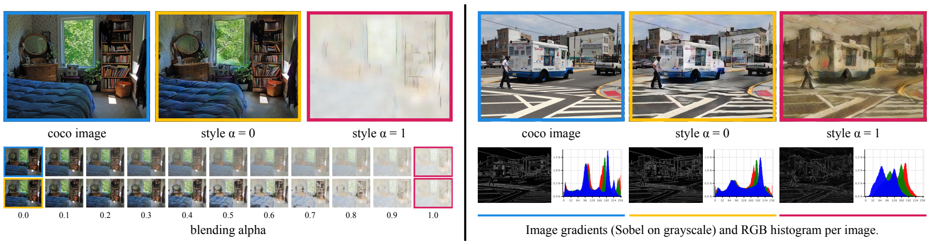

Fig. 3. Top row: Comparison of COCO and Stylized COCO at different alphas. The AdaIN method introduces subtle artifacts even at $\alpha = 0$ (no style). Bottom left: We control the style strength in feature space (yellow to pink) and pixel space (blue to pink). Every alpha depicts a complete version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

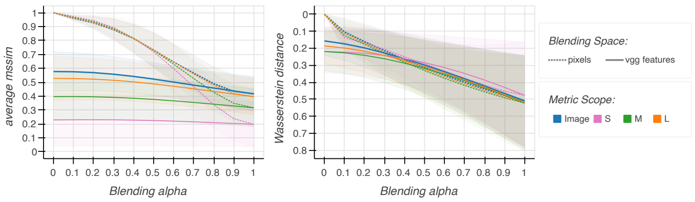

Fig. 4. Left: Average structural similarity between image gradients in relation to COCO (a score of 1 means that there is no difference between images). Right: Wasserstein distance between RGB histograms (reversed y-axis)

Quantitative measures have been calculated 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 observe the subtle shape changes in the gradient map of $\alpha = 0$ 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) [70] 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 step 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 to either image corruption or out-of-distribution texture and subsequently, determine the relative importance of each feature type.

2.2 Model Selection

To contribute a comprehensive overview on model robustness, we opted for a broad comparison of popular frameworks and architectures. The dimensions we consider to be impactful are *framework*, *architecture* and *pre-training*. The finally selected models 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 [25] and Cascade Mask R-CNN [6] that uses multiple refinement stages instead of one. Both frameworks formulate instance segmentation as a pixel-wise classification problem. Since this rather naive extension to Faster R-CNN [57] can ignore object boundaries and shapes, we include the boundary-preserving mask head alternative (Cascade-) BMask [13] for comparison. A remaining challenge to boundary detection are overlapping objects that occlude the ground truth contour of other instances. We therefore include the Bilayer Convolutional Network (BCNet) [38] as another mask head alternative. In BCNet, the occluded and occluding objects are detected separately and modeled explicitly in a layered representation. The mask head can then "consider the interaction between [the decoupled boundaries] during mask regression [38]." A second distinction between frameworks 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(++) [3,4] 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 [34] and deformable convolutions (v2) [14,76]. We include DETR [8] 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. Based on model availability we include BCNet in the FCOS [62] variant (F-BCNet). FCOS is a fully convolutional, one-stage, anchor-free alternative to Faster R-CNN that "solves object detection in a per-pixel prediction fashion, analogue to semantic segmentation [62]." 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 [69] and SOTR [23]. SOLO [68] 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 [61]. SOTR uses a twin attention mechanism [35] 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 [56,2,64].

Architectures used in instance segmentation can be divided into backbone, neck and functional heads. The latter output the final results and are framework specific. 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 [27], ResNext [72] and RegNet [54], a network found with meta architecture search that outperforms EfficientNet [59]. Note that BCNet utilizes a Graph Convolutional Network (GCN) [39] within its mask heads to model long-range dependencies between pixels (to evade local occlusion). Furthermore, DETR and SOTR are hybrid frameworks that use transformer architectures to process the encoded backbone features. With Swin Transformer [47] we also include a convolution free backbone alternative based on the Vision Transformer approach (ViT) [15]. The most popular neck choice is the Feature Pyramid Network (FPN) [44]. 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 [57] and a ResNet conv5 neck with dilated convolution (DC5) as used by [14]. Finally we abbreviate FPN models that use deformable convolutions as DCN [14,76]. 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 [60]". Based on availability we include the methods InstDis [71], MoCo [24,10], PIRL [51] and InfoMin [60]. 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) [21] data augmentation as

Table 1. Overview of frameworks, backbones and neck methods. (*) Swin Transformer use hierarchical representations similar to FPN necks in CNNs. RegNetY is similar to RegNetX but implements the Squeeze-and-Excitation operation [31]. Yolo consists of darknet (D), spatial pyramid pooling (SSP) [26] and a Path Aggregation Network (PAN) [46] in varying combinations with CSPNet (C) [65].

Backbone	**Framework**								
CNN	multi stage			one stage					
GCN	anchor based					anchor free			
Hybrid	top down (bbox→segm)							bottom up (segm→bbox)	
ViT	Mask R-CNN	BMask	Cascade	YOLO(v3,4,s4)	YOLACT(++)	DETR	FCOS BCNet	SOTR	SOLOv2
R50	FPN, C4, DC5, DCN	FPN	FPN	-	FPN, DCN	FPN, DC5	-	-	FPN
R101	FPN, C4, DC5,	FPN	DCN	-	FPN, DCN	FPN	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	-	-	-	-	-

an alternative to pre-training.

From the overview in Table 1 we can now derive dimensions that allow for a fair evaluation of models. Specifically, we can fix the backbone and neck architecture (e.g. ResNet + FPN) for a controlled comparison between *frameworks*. Vice versa, we can investigate the impact of a specific backbone and neck combination within a fixed framework (e.g. Mask R-CNN). The complete list of models is displayed in Figure 5. Note that we did not include the dimension of pre-training in the above overview for readability reasons. In our experiments however, we do compare training setups and learning schedules for a fixed model type (e.g. Mask R-CNN + ResNet + FPN).

3 Related Work

Robust vision can be approached from different perspectives. The classical view stems from signal processing and concerns image corruptions that are independent from the signal, e.g. salt and pepper noise [28,50,37,52]. A popular alternative is to compare model performance and failure cases against humans [20,19,17,18,58,63]. Since human vision is fairly robust, the hope is that vision models with a similar bias will generalize in more systematic ways as well. For instance, [18] and [63] show that transformer models perform closer to human behavior than CNNs. In support of the hypothesis, [49] find that transformer architectures are more robust against adversarial attacks. [48] on the other hand report that both model types are prone to small in-distribution changes in 3D perspective and lighting. The approach we take is inspired by these works but more direct. Instead of comparing to humans, it measures texture robustness in a challenging zero-shot setting. A third perspective originates from the long-tail distribution of real-world data. In such settings, robustness can be understood as the ability to adapt to uncommon or novel objects with efficient transfer learning [32,16], re-sampling [67,9] or regularization strategies [53,30,66]. Particular relevant are methods that handle object occlusion [11]. Since objects can be occluded in almost infinite ways, a common strategy is to represent object properties more explicitly, e.g. to decoupled shape and appearance for instance [13,38,16]. We expect these methods to be strong contenders in our comparison. As an alternative to our sensitivity benchmark, [36] analyze feature importance in latent representations and [7] use feature visualization to understand object detectors. Both leverage style transfer to simulate novel object appearances. The closest real-life alternative is the Natural Adversarial Objects (NAO) dataset [41]. It depicts a more realistic out-of-distribution setting but does not allow to control for pose and perspective, i.e. to observe the exact same objects with varying textures for instance. Other alternatives are the 3DB framework [42], a rendering engine that enables artifact free texture transfer on synthetic objects and SI-Score [74], a dataset for analyzing robustness to rotation, location and size. The key difference to work on reducing texture bias such as [19] and [43], is that these works do not control style strength, object contour and global stylization as potential confounders. As shown in subsection 2.1, this severely limits any analysis of the problem space. Our second contribution is the so far, most comprehensive selection of frameworks, architectures and pre-training schemes. This is crucial to allow for a fair comparison and rigorous analysis of model components which can not be easily derived from existing literature otherwise.

4 Results

In this section we present the zero-shot evaluation on Stylized COCO, Objects and Background. Each dataset version contains 20 copies of the accordingly styled COCO `val2017` subset. As a reference point, we *reproduce the evaluation* on the original `val2017` subset and report the absolute Average Precision (AP) for all models in Figure 5. As can be seen, training schedule, data augmentation and architecture choice have the biggest impact within a framework. Overall, RegNets trained with LSJ and Swin Transformer models perform best. Note that SOTR and SOLO have worse APs but significantly better APm and APl compared to other frameworks (see supplementary material for all scores).

We now present the results of our sensitivity analysis. In total, we tested 68 pre-trained models [4] on 61 `val2017` replicas which sums up to 4148 subset evaluations. Since it is not expedient to report this amount of data in the form of tables, we communicate mainly with figures in the main paper. However, the exact numerical values will be released for inspection together with the code. To quantify out-of-distribution robustness, we calculate the relative zero-shot performance in comparison to the performance on uncorrupted data. For every dataset version, blending space and alpha step we calculate:

$$rP_\alpha = P_\alpha / P_{coco}; P \in \{AP, APs, APm, APl\} \tag{3}$$

[4] See supplementary material for the list of code projects and weight sources.

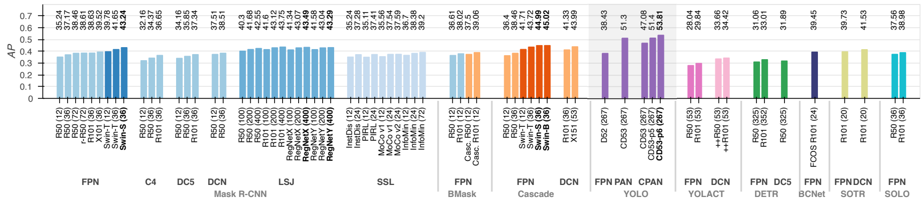

Fig. 5. Absolute performances on COCO `val2017`. Training schedules in epochs have been appended to model names. Note that Yolo is bounding box AP which is not comparable but included for model completeness. Methods that did not report scores for `val2017` have been validated on `test-dev2017` first

Note that we focus on IoU type segmentation and the scale dependent APs, m, l metrics since COCO has an unbalanced distribution of 41% small, 34% medium and 24% large instances. We include more metrics, absolute scores and the corresponding figures for bounding box IoU in the supplementary material.

A large scale comparison of all models can be found in Figure 6. Relative AP scores are displayed from left to right, datasets from top to bottom. The subfigures can be read as follows. From left to right ($\alpha : 0 \rightarrow 1$), how much of the original performance is lost with increasing out-of-distribution texture? Recall that at $\alpha = 0$, no out-of-distribution texture is used. For the feature space sequence (dark colors), this means that a loss in performance can be, at this point, attributed to image corruptions from the style transfer method. For the pixel space control sequence (light colors), this point depicts the original `val2017` score which is always 100% in our relative metric. For better visibility, we display an averaged model group per framework. This decision is not arbitrary however. Between all models, we calculated the average L2 distance over the full alpha range (see supplementary material for the resulting distance matrices). As a result of this analysis, we find models from the same framework to perform more similar to each other ($\mu_{L2} = 0.08 \pm 0.05$) than to models from other frameworks ($\mu_{L2} = 0.21 \pm 0.08$). Note that Swin Transformer is treated as a custom *framework* in this comparison even though it implements the (Cascade) Mask R-CNN strategies. In the following, we highlight a few key observations.

First subfigure (top left): The average framework AP on Stylized COCO starts from only $58 \pm 5\%$ due to the impact of image corruption. Revisit Figure 3 to see how images look like at this point ($\alpha = 0$). After this initial loss, performance remains fairly constant up to $\alpha = 0.3$ from where it drops to $18 \pm 5\%$. Apparently, frameworks seem to maintain a consistent ranking over the full alpha range which we investigate in more detail in our controlled comparison.

First row (Stylized COCO): On small objects (rAPs), model performance is severely affected by image corruptions from the beginning: $19 \pm 6\% \rightarrow 4 \pm 2\%$. With increasing objects size, this effect seems to gradually vanish. On large

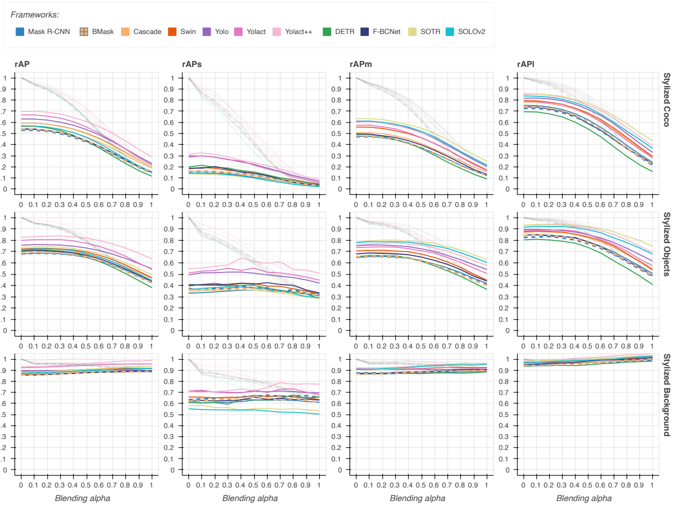

Fig. 6. General overview of model robustness against out-of-distribution texture (zoom in for better visibility). Yolo is displayed for completeness but not directly comparable.

objects (rAPl), the relative performance is then more affected by actual out-of-distribution texture than by corruption artifacts: $78 \pm 5\% \rightarrow 28 \pm 8\%$.

Second row (Stylized Objects): The average framework model is more robust to this type of data, e.g. rAP is now: $72 \pm 5\% \rightarrow 47 \pm 7\%$. In addition, models exhibit an extended range of robustness after the initial dip (now up to $\alpha = 0.5$). The apparent ranking of frameworks remains the same. Note the increasing variance towards the end of the blending sequence, e.g. rAPm: $70 \pm 5\% \rightarrow 48 \pm 9\%$, rAPl: $87 \pm 4\% \rightarrow 56 \pm 10\%$. We conclude that object contour is in fact an important property and that some models can exploit this feature more effectively than others. However, no model is able to exploit the implicit encoding of ground truth masks. Doing so would imply a high level of abstraction which we do not find to happen in existing models.

Third row (Stylized Background): In this dataset version, only background features are corrupted. This allows us to measure the importance of context information. As displayed in the second subfigure (rAPs), models do indeed rely on additional context information to segment small objects: $63 \pm 5\% \rightarrow 63 \pm 7\%$. With increasing object size, models depend less on background information. Interestingly, we observe a subtle performance gain on large objects with increasing stylization (rAPl): $96 \pm 2\% \rightarrow 101 \pm 2\%$. As before however, models do not exploit the implicit mask encoding in a more systematic way which is arguably even easier in this setting.

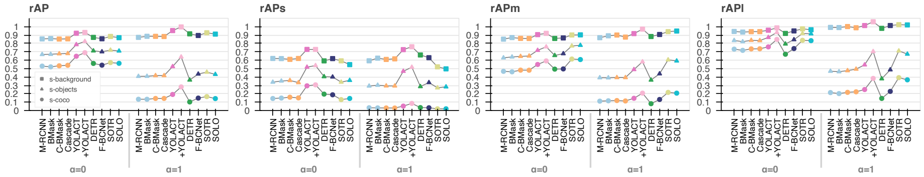

Fig. 7. A controlled comparison of framework robustness. Note that we compromise on ResNet-101 for SOTR and F-BCNet, see Table 1 for available backbones

A controlled comparison of frameworks, architectures and pre-training is presented in this subsection. Each comparison contributes to the main goal of our study which asks: *Do existing segmentation methods contain components or design decisions that promote systematic generalization regarding object-centric features?* Note that we focus on the extreme points $\alpha = 0$ and $\alpha = 1$.

Frameworks are compared first. For statistical fairness, we use the same backbone and neck method per framework (ResNet-50 + FPN). Figure 7 displays the results on the feature space sequence. As hypothesized earlier, we observe a consistent ranking between frameworks from $\alpha : 0 \rightarrow 1$ and across dataset versions. A noticeable difference is the amplified difference between frameworks and the increased distance between Stylized COCO and Stylized Objects at $\alpha = 1$. We conclude that robustness against semantic image corruption is correlated with robustness against out-of-distribution texture. However, some frameworks seem to exploit object contour more effectively than others. The most robust framework overall is YOLACT++. Its predecessor YOLACT is compatible on small objects. SOTR and SOLO(v2) perform best on medium and large objects. DETR appears as the least robust framework but we like to point out that it was not primarily designed for instance segmentation. Surprisingly, BMask and BCNet are not better than Cascade and Mask R-CNN.

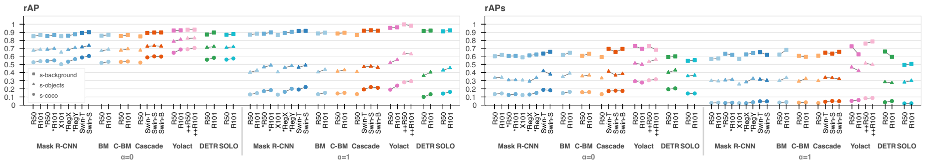

Fig. 8. Model robustness for different backbones. Scores for medium and large objects follow the trend of rAP and can be found in the supplementary material. Models marked with * are trained with LSJ

Backbones are compared next. In this analysis, we only use models with FPN neck and compare within the same framework. When available, we include model pairs with different data augmentation. The results are displayed in Figure 8. As can be seen from the left subfigure (rAP), a clear trend is visible. When every other factor is controlled, deeper backbones improve robustness. Swin-T and S are comparable to ResNet-50 and 101 in model complexity but generally more robust than their CNN counterparts. For medium and large objects, a similar trend can be reported. For small objects, the behavior is less clear and surprisingly, often reversed.

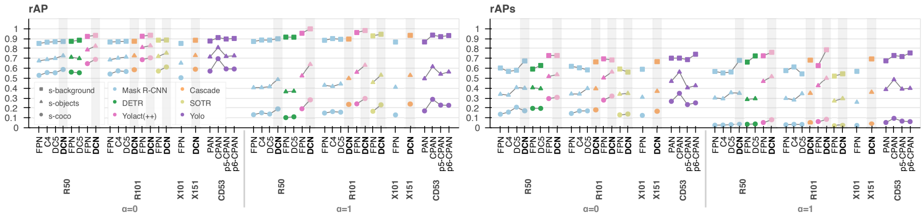

Fig. 9. Robustness by neck architecture. Deformable convolutions are highlighted for better visibility. Note that Yolo is only comparable to itself due to bounding box score

Neck methods are compared in Figure 9. We only use models with a similar training schedule and group them by backbone and framework. Surprisingly, the popular FPN method is consistently the least robust option. In contrast, deformable convolutions (DCN) consistently improve robustness as highlighted for better visibility. For small objects, DC5 necks improve the robustness on Stylized Objects (triangle). Note that Yolo is only comparable against itself with the simple v4-csp model (CPAN) performing best.

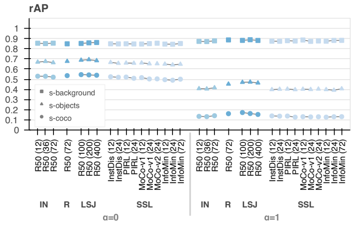

Fig. 10. Model robustness by pre-training and data augmentation. We distinguish between ImageNet (IN), random (R) and self supervised learning (SSL) initialization

Pre-training, training schedules and data augmentation have surprisingly, little to no impact on texture robustness. The result of this comparison is displayed Figure 10. On the other hand we can report that supervised and unsupervised representations perform almost on par. We suspect that the supervised fine tuning on the segmentation task overshadows potential differences in the initial backbone representations. In comparison we find that random initialization with LSJ data augmentation is the most robust combination.

5 Discussion

As expected, we found deep learning models for instance segmentation to be vulnerable to image corruption and not particularly robust against novel object texture. Based on our in depth analysis, we now understand this problem space much better. In particular that the type of robustness depends heavily on object size. Small objects are predominantly affected by image corruption which confirms related findings from work on scale aware augmentation [12]. In consequence, we hypothesize that the difference between shape and texture simply collapses for (very) small objects in MS COCO. Furthermore, we find that segmentation models can exploit object contour independently from object texture (better performance on Stylized Objects). However, the question arises if this is not simply a byproduct of the limited area of effect in this dataset. To answer this question, we conduct an ablation study with salt and pepper noise which rejects this hypothesis. A second ablation with blurred object contour confirms the importance of this feature. The results can be found in the supplementary material. Note that more generally, the importance of contour for segmentation is acknowledged in the literature, e.g. by work on mask refinement [40,75]. It would be interesting to see how these methods adapt to unseen textures. For medium and large objects, we found models to be biased towards learned texture, i.e. out-of-distribution texture results in a significant performance loss after controlling for general image corruptions. In contrast to [19] who argue that this problem is induced by the ImageNet training data, we find no such correlation in terms of pre-training for instance segmentation. We conclude that either a similar bias is induced by the COCO dataset or that framework and architecture choice is simply more important as suggested by [29]. Our key findings support the latter hypothesis. In particular the results of YOLACT(++), SOTR and SOLOv2 unveil a new and promising starting point for more in depth research, e.g. on prototype masks and implicit localization. In the same spirit, [22] formulate *the binding problem* which is “the inability [of neural networks] to dynamically and flexibly combine (bind) information [... which] limits their ability to [...] accommodate different patterns of generalization”. The consistently better results for dynamic architectures and components confirm this assumption.

6 Conclusion

In this study we contribute a comprehensive baseline on the texture robustness of deep learning based instance segmentation. As a result of our study, we find a noticeable texture bias in most existing methods. However, models do also exploit other features such as object contour. Based on this insight we feel optimistic that, with the right design decisions, vision models are not trapped in texture bias. The key finding of our study is that the frameworks YOLACT++, SOTR and SOLOv2 as well as deeper and dynamic architectures improve texture robustness. We hope that our rigorous analysis enables more in depth research and contributes to facilitate a systematic design approach to robust vision.

References

1. Baker, N., Lu, H., Erlikhman, G., Kellman, P.J.: Deep convolutional networks do not classify based on global object shape. PLOS Computational Biology **14**(12), e1006613 (Dec 2018)
2. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: Optimal speed and accuracy of object detection. CoRR **abs/2004.10934** (2020)
3. Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: YOLACT: Real-time instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (Oct 2019)
4. Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: YOLACT++: Better real-time instance segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2020)
5. Brendel, W., Bethge, M.: Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet. In: International Conference on Learning Representations (2019)
6. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2018)
7. Cao, A., Johnson, J.: Inverting and understanding object detectors. CoRR **abs/2106.13933** (2021)
8. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-End Object Detection with Transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020, vol. 12346, pp. 213–229. Springer International Publishing, Cham (2020)
9. Chang, N., Yu, Z., Wang, Y.X., Anandkumar, A., Fidler, S., Alvarez, J.M.: Image-level or object-level? A tale of two resampling strategies for long-tailed detection. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 1463–1472. PMLR (Jul 2021)
10. Chen, X., Fan, H., Girshick, R.B., He, K.: Improved baselines with momentum contrastive learning. CoRR **abs/2003.04297** (2020)
11. Chen, Y.T., Liu, X., Yang, M.H.: Multi-instance object segmentation with occlusion handling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2015)
12. Chen, Y., Li, Y., Kong, T., Qi, L., Chu, R., Li, L., Jia, J.: Scale-aware automatic augmentation for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9563–9572 (Jun 2021)
13. Cheng, T., Wang, X., Huang, L., Liu, W.: Boundary-Preserving Mask R-CNN. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020, vol. 12359, pp. 660–676. Springer International Publishing, Cham (2020)
14. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
16. 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)

17. Geirhos, R., Narayanappa, K., Mitzkus, B., Bethge, M., Wichmann, F.A., Brendel, W.: On the surprising similarities between supervised and self-supervised models (2020)
18. Geirhos, R., Narayanappa, K., Mitzkus, B., Thieringer, T., Bethge, M., Wichmann, F.A., Brendel, W.: Partial success in closing the gap between human and machine vision. CoRR **abs/2106.07411** (2021)
19. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: International Conference on Learning Representations (2019)
20. Geirhos, R., Temme, C.R.M., Rauber, J., Schütt, H.H., Bethge, M., Wichmann, F.A.: Generalisation in humans and deep neural networks. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018)
21. Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.Y., Cubuk, E.D., Le, Q.V., Zoph, B.: 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 (Jun 2021)
22. Greff, K., van Steenkiste, S., Schmidhuber, J.: On the binding problem in artificial neural networks. CoRR **abs/2012.05208** (2020)
23. Guo, R., Niu, D., Qu, L., Li, Z.: SOTR: Segmenting objects with transformers. CoRR **abs/2108.06747** (2021)
24. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2020)
25. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
26. He, K., Zhang, X., Ren, S., Sun, J.: Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014, vol. 8691, pp. 346–361. Springer International Publishing, Cham (2014)
27. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2016)
28. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: International Conference on Learning Representations (2019)
29. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15262–15271 (Jun 2021)
30. Hsieh, T.I., Robb, E., Chen, H.T., Huang, J.B.: DropLoss for long-tail instance segmentation. In: Proceedings of the Workshop on Artificial Intelligence Safety 2021 (SafeAI 2021) Co-Located with the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2021), Virtual, February 8, 2021 (2021)
31. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2018)

32. Hu, R., Dollár, P., He, K., Darrell, T., Girshick, R.: Learning to segment every thing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2018)
33. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
34. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring r-cnn. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2019)
35. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: CCNet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (Oct 2019)
36. Islam, M.A., Kowal, M., Esser, P., Jia, S., Ommer, B., Derpanis, K.G., Bruce, N.: Shape or texture: Understanding discriminative features in {CNN}s. In: International Conference on Learning Representations (2021)
37. Kamann, C., Rother, C.: Benchmarking the Robustness of Semantic Segmentation Models with Respect to Common Corruptions. International Journal of Computer Vision **129**(2), 462–483 (Feb 2021)
38. Ke, L., Tai, Y.W., Tang, C.K.: Deep occlusion-aware instance segmentation with overlapping BiLayers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4019–4028 (Jun 2021)
39. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR) (2017)
40. Kirillov, A., Wu, Y., He, K., Girshick, R.: PointRend: Image segmentation as rendering. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2020)
41. Lau, F., Subramani, N., Harrison, S., Kim, A., Branson, E., Liu, R.: Natural adversarial objects. arXiv preprint arXiv:2111.04204 (2021)
42. Leclerc, G., Salman, H., Ilyas, A., Vemprala, S., Engstrom, L., Vineet, V., Xiao, K.Y., Zhang, P., Santurkar, S., Yang, G., Kapoor, A., Madry, A.: 3DB: A framework for debugging computer vision models. CoRR **abs/2106.03805** (2021)
43. Li, Y., Yu, Q., Tan, M., Mei, J., Tang, P., Shen, W., Yuille, A., xie, c.: Shape-texture debiased neural network training. In: International Conference on Learning Representations (2021)
44. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jul 2017)
45. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014, vol. 8693, pp. 740–755. Springer International Publishing, Cham (2014)
46. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2018)
47. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
48. Madan, S., Sasaki, T., Li, T.M., Boix, X., Pfister, H.: Small in-distribution changes in 3D perspective and lighting fool both CNNs and Transformers. CoRR **abs/2106.16198** (2021)

49. Mahmood, K., Mahmood, R., van Dijk, M.: On the robustness of vision transformers to adversarial examples. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7838–7847 (Oct 2021)
50. Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A.S., Bethge, M., Brendel, W.: Benchmarking robustness in object detection: Autonomous driving when winter is coming. In: Machine Learning for Autonomous Driving Workshop, NeurIPS 2019. vol. 190707484 (Jul 2019)
51. Misra, I., van der Maaten, L.: Self-supervised learning of pretext-invariant representations. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2020)
52. Mummadi, C.K., Subramaniam, R., Hutmacher, R., Vitay, J., Fischer, V., Metzen, J.H.: Does enhanced shape bias improve neural network robustness to common corruptions? In: International Conference on Learning Representations (2021)
53. Pan, T.Y., Zhang, C., LI, Y., Hu, H., Xuan, D., Changpinyo, S., Gong, B., Chao, W.L.: On model calibration for long-tailed object detection and instance segmentation. In: Thirty-Fifth Conference on Neural Information Processing Systems (2021)
54. Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollar, P.: Designing network design spaces. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2020)
55. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do ImageNet classifiers generalize to ImageNet? Proceedings of Machine Learning Research, vol. 97, pp. 5389–5400. PMLR, Long Beach, California, USA (Jun 2019)
56. Redmon, J., Farhadi, A.: YOLOv3: An incremental improvement. CoRR **abs/1804.02767** (2018)
57. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 28. Curran Associates, Inc. (2015)
58. Shankar, V., Roelofs, R., Mania, H., Fang, A., Recht, B., Schmidt, L.: Evaluating machine accuracy on ImageNet. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 8634–8644. PMLR (Jul 2020)
59. Tan, M., Le, Q.: EfficientNet: Rethinking model scaling for convolutional neural networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 6105–6114. PMLR (Jun 2019)
60. Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P.: What makes for good views for contrastive learning? In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 6827–6839. Curran Associates, Inc. (2020)
61. Tian, Z., Shen, C., Chen, H.: Conditional Convolutions for Instance Segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020, vol. 12346, pp. 282–298. Springer International Publishing, Cham (2020)
62. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (Oct 2019)
63. Tuli, S., Dasgupta, I., Grant, E., Griffiths, T.L.: Are convolutional neural networks or transformers more like human vision? CoRR **abs/2105.07197** (2021)
64. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: Scaled-yolov4: Scaling cross stage partial network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13029–13038 (Jun 2021)

65. Wang, C.Y., Mark Liao, H.Y., Wu, Y.H., Chen, P.Y., Hsieh, J.W., Yeh, I.H.: 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. IEEE, Seattle, WA, USA (Jun 2020)
66. Wang, J., Zhang, W., Zang, Y., Cao, Y., Pang, J., Gong, T., Chen, K., Liu, Z., Loy, C.C., Lin, D.: Seesaw loss for long-tailed instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9695–9704 (Jun 2021)
67. Wang, T., Li, Y., Kang, B., Li, J., Liew, J., Tang, S., Hoi, S., Feng, J.: The Devil Is in Classification: A Simple Framework for Long-Tail Instance Segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020, vol. 12359, pp. 728–744. Springer International Publishing, Cham (2020)
68. Wang, X., Kong, T., Shen, C., Jiang, Y., Li, L.: SOLO: Segmenting Objects by Locations. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020, vol. 12363, pp. 649–665. Springer International Publishing, Cham (2020)
69. Wang, X., Zhang, R., Kong, T., Li, L., Shen, C.: SOLOv2: Dynamic and fast instance segmentation. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 17721–17732. Curran Associates, Inc. (2020)
70. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (Apr 2004)
71. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2018)
72. Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jul 2017)
73. Yuille, A., Liu, C.: Deep Nets: What have they ever done for Vision? Technical Report 88, Center for Brains, Minds and Machines (CBMM), MIT CSAIL (May 2018)
74. Yung, J., Romijnders, R., Kolesnikov, A., Beyer, L., Djolonga, J., Houlsby, N., Gelly, S., Lucic, M., Zhai, X.: SI-Score: An image dataset for fine-grained analysis of robustness to object location, rotation and size. CoRR **abs/2104.04191** (2021)
75. Zhu, C., Zhang, X., Li, Y., Qiu, L., Han, K., Han, X.: SharpContour: A contour-based boundary refinement approach for efficient and accurate instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4392–4401 (Jun 2022)
76. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable ConvNets V2: More deformable, better results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2019)