Proceedings of the 7th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/2027.39456

Generative Semantic Segmentation of Seven COCO Object Classes and Its Robustness to Image Corruptions

Zixuan Xu

Department of Computer Sciences and Engineering, University of Technology of Compiègne, Compiègne, France
zixuan.xu@etu.utc.fr

Abstract. Current image generation models show substantial image understanding capabilities and can use a visual prompt to transform an input image into a color-coded perception output. However, their output quality is sensitive to prompt design, and rigorous comparisons have yet to establish whether they are more robust than dedicated visual perception models when image quality degrades. This paper uses an image generation model as a seven-class semantic segmenter, establishes a fixed prompt and decoding pipeline, and compares it with conventional segmentation models under several common image corruptions. On an independent set of 120 images and across six prespecified primary corruption conditions, the generative pipeline achieved a mean intersection over union (mIoU) of 0.5756, compared with 0.4727 for YOLO26s-seg and 0.4926 for Mask R-CNN. The image-paired 95% confidence intervals for both differences in mean corrupted-image performance were entirely above zero, and the generative pipeline showed a smaller performance drop from clean to corrupted images. Within the seven-class COCO semantic segmentation task and corruption conditions examined here, these results indicate that the pipeline combining a fixed prompt and decoding rules has greater pipeline-level robustness than the two conventional baselines. They provide empirical evidence for evaluating the robustness of generative visual perception within a clearly defined experimental scope.

Keywords: generative visual perception, prompt sensitivity, semantic segmentation, image corruption robustness, paired evaluation

1. Introduction

Semantic segmentation is a fundamental task in visual perception. Fully Convolutional Networks (FCNs) [1], You Only Look Once (YOLO) [2], and Mask Region-based Convolutional Neural Networks (Mask R-CNN) [3] have established approaches to semantic segmentation, object detection, and instance segmentation, respectively. Specific implementations such as YOLO26s [4] and Mask R-CNN [5] also provide segmentation models pretrained on the Common Objects in Context (COCO) dataset. In recent years, image generation models have increasingly demonstrated visual understanding capabilities. Studies including InstructCV [6], Vision Banana [7], and ProbeGen [8] suggest that generative models have the potential to perform visual perception tasks such as semantic segmentation and depth estimation.

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2027.39456

Generative visual perception nevertheless faces two challenges: its results are sensitive to prompt design and output format, and few studies directly compare its performance with dedicated vision models using the same images, ground truth, and post-processing conditions. Previous work has shown that image corruptions such as blur, noise, and compression affect the performance of conventional models [9, 10], but sufficient evidence about the stability of generative segmentation under these conditions is still lacking.

To investigate these issues, this paper builds a generative semantic segmentation pipeline using the OpenAI image generation tool [11] and examines the effects of prompt design and common image corruptions on segmentation results using the COCO 2017 dataset [12]. The generative pipeline and conventional segmentation models are compared using the same inputs, ground truth, and evaluation protocol to analyze their performance on clean and corrupted images.

This study aims to test the stability of a fixed generative pipeline under common image corruptions and to clarify its advantages, scope, and limitations relative to conventional segmentation models. It does not seek to prove that generative models are universally superior to conventional models. Instead, it aims to establish a reproducible comparison method and provide experimental evidence for evaluating robustness in generative visual perception.

2. Methodology

2.1. Model selection

This paper uses YOLO26s-seg and Mask R-CNN as conventional segmentation baselines. YOLO26s-seg uses the official COCO-pretrained weights [4], an input size of 640, a confidence threshold of 0.25, and a non-maximum suppression intersection-over-union threshold of 0.7. Mask R-CNN uses TorchVision's maskrcnn_resnet50_fpn_v2 with the official COCO_V1 weights [3, 5], a score threshold of 0.25, and a mask probability threshold of 0.5. Neither model is trained on the experimental images; their weights, thresholds, and rules for composing semantic predictions are fixed before inference.

Generative segmentation is performed using the OpenAI image generation tool built into Codex. The generative model is neither trained nor fine-tuned for this experiment. Instead, a fixed prompt is used to transform each input image into a color-coded semantic mask.

2.2. Dataset and image corruptions

The experiments use the COCO 2017 val2017 split [12]. COCO provides public instance segmentation annotations, allowing instance masks to be combined deterministically into semantic ground truth. The same ground truth can then be used for blur, noise, and compression corruptions that leave object coordinates unchanged. YOLO26s-seg and Mask R-CNN both have COCO-pretrained weights with compatible class labels [4, 5], allowing conventional baselines to be established directly without requiring cross-dataset class mapping or additional pretraining to obtain standard recognition performance.

Seven target classes were specified in advance as a controlled task space, as shown in Table 1. This allows the prompt to use an explicit discrete color code and helps ensure that the limited sample covers different classes and object sizes. COCO instance annotations are converted into seven-class semantic ground truth: instances of the same class are merged, and annotations of different classes are drawn in descending order of annotated area so that smaller instances determine the class of overlapping pixels. Regions outside the seven classes are treated as background,

DOI: 10.54254/2755-2721/2027.39456

although incorrectly predicting a target class in those regions still counts as a false positive. Predictions from the conventional models are also converted into the same seven-class semantic space for evaluation.

Table 1. Color codes for the seven target classes and background

Class	Color	Hex code
person	Red	#E41A1C
car	Blue	#377EB8
chair	Green	#4DAF4A
couch	Purple	#984EA3
cat	Orange	#FF7F00
dog	Yellow	#FFFF33
bottle	Brown	#A65628
Background (non-target classes)	Black	#000000

The robustness experiment uses three types of image corruption (Gaussian blur, Gaussian noise, and JPEG compression), each at severity levels 1, 3, and 5. In that order, the standard deviations for Gaussian blur are 1, 3, and 6 pixels; those for Gaussian noise are 0.08, 0.18, and 0.38, with image values normalized to the range 0-1; and the JPEG quality settings are 25, 15, and 7. All transformations preserve image dimensions and spatial coordinates, so the semantic ground truth for the clean image is reused. Severity levels 1 and 3 constitute the primary analysis conditions, while severity level 5 serves as a stress test.

2.3. Experimental design and test conditions

The main experiment consists of two sequential stages specified in advance, as outlined in Table 2.

Table 2. Experimental stages, sample sizes, and input conditions

Stage	Number of images	Purpose
Prompt screening	50	Compare four prompts and select the fixed prompt for subsequent experiments
Independent robustness confirmation	120	Compare generative pipeline, YOLO26s-seg, and Mask R-CNN

The two stages use non-overlapping images from the COCO val2017 split. For the prompt-screening set, seven images in which the target occupies at least 2% of the image are selected for each of the seven classes, followed by one additional image containing a target class, yielding 50 images in total. The independent confirmation set excludes previously used images and contains 120 images selected by stratifying on class and object size, without the 2% area threshold. During prompt screening, four prompts are compared on clean images, and the prompt for the robustness experiment is selected according to the prespecified rule of highest mIoU. Before examining results from the independent confirmation set, the prompt, color palette, decoder, and single-generation rule

DOI: 10.54254/2755-2721/2027.39456

are fixed. In the robustness confirmation stage, one output is generated for the clean version and each of the nine corrupted versions of every image.

The prompt instructs the generator to output a flat semantic mask containing only eight colors: seven colors for the target classes and black for the background. It also requires preservation of the original object contours, spatial layout, viewpoint, and aspect ratio, without added text, legends, shadows, or gradients.

This paper defines the complete processing chain from an input image to a scorable semantic prediction as the "generative pipeline." The pipeline provides an RGB image and a fixed prompt to the OpenAI image generation tool built into Codex, which produces a color-coded semantic mask. Because the generated image may differ from the requested dimensions or pixel colors, a fixed decoder then resizes it to the input image dimensions using nearest-neighbor interpolation. Each pixel is mapped to the nearest of the eight palette colors by RGB distance, producing a semantic segmentation prediction comprising the seven target classes and background.

2.4. Evaluation metrics

The primary segmentation metric is the mean intersection over union across the seven target classes (mIoU):

$$\mathrm{mIoU}=\frac{1}{7}\sum_{k=1}^{7}\frac{\sum_i |P_{ik}\cap G_{ik}|}{\sum_i |P_{ik}\cup G_{ik}|} \tag{1}$$

Here, P_{ik} and G_{ik} denote the predicted and ground-truth regions, respectively, for class k in image i.

Robustness is evaluated using both absolute mIoU on corrupted images and the performance drop relative to clean images, so that model quality is not judged from the size of the drop alone. To further account for differences in clean-image performance, this paper calculates the retention rate of jointly correct foreground pixels: the proportion of foreground pixels predicted correctly by both pipelines on the clean image that remain correct after corruption. This metric is used only as supporting evidence.

Differences among prompt variants and between model pipelines are assessed using an image-paired bootstrap to estimate 95% confidence intervals [13]. The resampling unit is the complete image; each comparison uses 10,000 resamples and a fixed random seed. Raw-output format compliance, generation failures, and workflow time are also recorded separately to assess output reliability and practical efficiency.

3. Results

3.1. Prompt-screening

As P0 did not produce an output for one image, the four prompts are compared on the same 49 clean images for which all four outputs are available. P0 is the baseline prompt. P1 removes P0's statement of purpose and some repeated constraints while retaining the core requirements for the seven-class color mapping, preservation of object positions and contours, and output of a flat-color mask. P2 presents the task, output format, color mapping, and segmentation rules in separate sections; it explicitly requires every pixel to use one of the eight specified colors and requests a match to the original image dimensions when supported by the tool. P3 emphasizes the

DOI: 10.54254/2755-2721/2027.39456

identification of small and partially occluded objects, annotation of visible pixels only, and preservation of object position, scale, and contours without completing occluded regions.

Table 3. Seven-class semantic segmentation results for the four prompts

Prompt	Seven-class mIoU	Precision	Recall	Mean proportion of pixels exactly matching the palette
P0 Baseline prompt	0.4975	0.5658	0.8271	0.3884
P1 Condensed core requirements	0.4729	0.5198	0.8183	0.3711
P2 Structured output specification	0.6001	0.7020	0.8662	0.4670
P3 Object coverage and visible regions	0.6151	0.7234	0.8353	0.4861

Table 4. Paired mIoU differences between prompts and 95% confidence intervals

Comparison pairs	mIoU difference	Image-paired 95% CI
P1 − P0	−0.0246	[−0.0823,+0.0209]
P2 − P0	+0.1026	[+0.0473,+0.1486]
P3 − P0	+0.1176	[+0.0622,+0.1637]
P3 − P2	+0.0150	[−0.0151,+0.0523]
P2 − P1	+0.1273	[+0.0633,+0.1922]
P3 − P1	+0.1422	[+0.0765,+0.2106]

As Tables 3 and 4 show, both P2 and P3 improve significantly on P1, indicating that structured output instructions and object-coverage constraints affect generative segmentation results. P3 has the highest point estimate and is therefore selected for independent robustness confirmation under the prespecified rule. However, the interval for the difference between P3 and P2 includes zero, so these results do not establish that P3 consistently outperforms P2.

3.2. Robustness confirmation

Using 120 images that do not overlap with the prompt-screening set, this experiment compares the image-corruption robustness of the generative pipeline, YOLO26s-seg, and Mask R-CNN. All three pipelines process clean images and images corrupted by Gaussian blur, Gaussian noise, or JPEG compression at severity levels 1, 3, and 5. Their outputs are scored against the same ground truth.

DOI: 10.54254/2755-2721/2027.39456

Table 5. mIoU of the three pipelines on clean images and under nine different corruption conditions

Condition	Generative pipeline	YOLO26s-seg	Mask R-CNN
Original	0.5878	0.7224	0.7006
Gaussian blur 1	0.6349	0.7198	0.6407
Gaussian blur 3	0.5787	0.6500	0.5362
Gaussian blur 5	0.5140	0.4479	0.2845
Gaussian noise 1	0.6156	0.5448	0.5687
Gaussian noise 3	0.5019	0.2831	0.3939
Gaussian noise 5	0.3742	0.0402	0.1674
JPEG 1	0.6079	0.4408	0.5136
JPEG 3	0.5145	0.1977	0.3024
JPEG 5	0.4817	0.0370	0.0640

Table 5 shows that YOLO26s-seg and Mask R-CNN both achieve higher mIoU than the generative pipeline on clean images, while YOLO26s-seg leads at Gaussian blur severity levels 1 and 3. The generative pipeline has higher mIoU point estimates than both conventional baselines at all three Gaussian noise levels, all three JPEG compression levels, and Gaussian blur level 5.

Table 6. Mean mIoU and robustness differences across the six primary corruption conditions

Comparison	Generative pipeline mean mIoU	Baseline mean mIoU	Absolute difference	Advantage in performance drop (95% CI)
YOLO26s-seg	0.5756	0.4727	+0.1029 [+0.0219,+0.1708]	+0.2375 [+0.1536,+0.3123]
Mask R-CNN	0.5756	0.4926	+0.0830 [+0.0164,+0.1424]	+0.1957 [+0.1299,+0.2552]

The decreases in mIoU from clean images to the mean across the six primary corruption conditions are 0.0122, 0.2497, and 0.2080 for the generative pipeline, YOLO26s-seg, and Mask R-CNN, respectively (Tables 5 and 6). For both baselines, the confidence intervals for the absolute performance difference and the difference in performance drop lie entirely above zero. Thus, the generative pipeline achieves both a higher mean score on corrupted images and less degradation relative to clean images.

Figure 1 shows that all three pipelines have relatively high foreground micro IoU on the clean image: 0.847 for the generative pipeline, 0.960 for YOLO26s-seg, and 0.915 for Mask R-CNN. With Gaussian noise at severity level 3, their scores are 0.794, 0.002, and 0.386, respectively. The generative pipeline still preserves the main regions of the dog and bottle, whereas the two conventional baselines miss objects or misclassify them. The figure illustrates a failure pattern consistent with the aggregate results across 120 images.

DOI: 10.54254/2755-2721/2027.39456

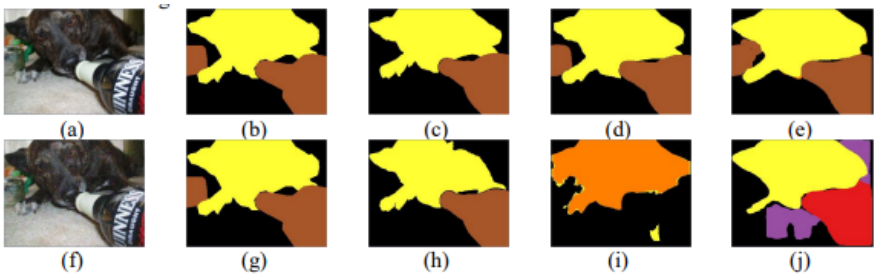

Figure 1. Comparison of the same image under clean input and Gaussian noise 3, (a) Clean input image, (b) COCO truth (clean), (c) Generative pipeline (clean), (d) YOLO26s (clean), (e) Mask R-CNN (clean), (f) Noisy input image, (g) COCO truth (noisy), (h) Generative pipeline (noisy), (i) YOLO26s (noisy), (j) Mask R-CNN (noisy)

Table 7. Mean retention of jointly correct foreground pixels across the six primary corruption conditions

Baseline	Jointly correct pixels as a share of valid foreground	Generative pipeline retention	Baseline retention	Difference (95% CI)
YOLO26 s-seg	77.50%	94.43%	78.42%	+16.01% [+11.72,+20.52] %
Mask R-CNN	79.51%	93.62%	83.74%	+9.89 % [+6.54,+13.52] %

In Table 7, the jointly correct foreground-pixel analysis provides further support for the generative pipeline's ability to retain correct predictions on corrupted images.

At some fine boundaries and in gaps between objects, COCO instance annotations do not fully represent the visible structure in the original image. This may affect mIoU calculated against those annotations.

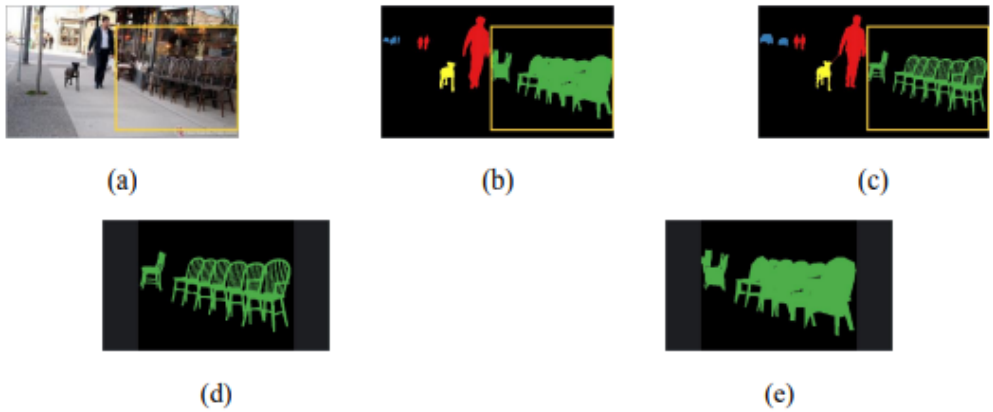

Figure 2. Example of a mismatch in boundary granularity between the generated result and the COCO annotation, (a) Original RGB image, (b) COCO truth, (c) Generative pipeline, (d) COCO truth: chair-region detail, (e) Generative pipeline: chair-region detail

DOI: 10.54254/2755-2721/2027.39456

As Figure 2 shows, the generative pipeline produces finer segmentation structures around visible gaps between the chairs. This suggests a possible advantage in representing local boundaries. That possibility still requires verification using boundary F1, boundary distance, and manually refined annotations.

3.3. Output reliability and runtime evaluation

All 1,200 prespecified tasks in the confirmation experiment produced final outputs. Other incomplete log records arose mainly from local checks and scheduling interruptions and do not represent model generation failures. Because generation calls were concurrent, active-phase elapsed time and steady-state throughput time per image are used to compare practical workflow efficiency.

Table 8. Workflow runtime and throughput

Pipeline	Active-phase elapsed time	Steady-state time per image	Approximate throughput
Generative model, three active phases	13.75 h	41.26 s	0.024 images/s
YOLO26s-seg, local CPU	149.7 s	0.125 s	8.00 images/s
Mask R-CNN, local CPU	3595.0 s	3.001 s	0.333 images/s

Based on the active-phase throughput reported in Table 8, the generative pipeline is approximately 330 times slower than YOLO26s-seg and 13.7 times slower than Mask R-CNN. The remote end-to-end time for the generative pipeline includes queuing, generation, transmission, and saving, whereas the conventional baselines' times exclude initial model loading. These ratios therefore describe workflow efficiency.

4. Discussion

Overall, within the prespecified seven-class COCO task and six primary corruption conditions, the generative pipeline achieves higher mean mIoU on corrupted images and a smaller performance drop, demonstrating stronger pipeline-level robustness. Inspection of individual images also suggests that, in some cases, the generative pipeline depicts fine boundaries and gaps between objects in a way that more closely matches the original image. If COCO annotations omit these details, mIoU computed against them may not capture the pipeline's performance at fine boundaries. More detailed ground-truth annotations are needed.

One possible explanation for the pipeline's robustness is that the generative model produces segmentation results by drawing on global semantics, object shapes, and scene context. When noise or compression mainly damages local texture while overall object contours and scene relationships remain recognizable, the model's semantic prior may help preserve target classes and their main regions. Conventional segmentation models, by comparison, extract features directly from corrupted pixels and may miss detections or produce degraded masks when the input distribution shifts. The fixed prompt and nearest-color decoding may also reduce variation in the output. The observed robustness therefore belongs to the complete pipeline formed by the generation tool, prompt, and fixed decoder.

At the same time, semantic priors may produce hallucinations, excessive smoothing, or boundary shifts. These possibilities may help explain why the generative pipeline does not lead on clean

DOI: 10.54254/2755-2721/2027.39456

images or under every blur condition. The proposed mechanisms are inferences from the experimental observations and require further testing through decoder ablations, repeated generations, and intermediate-feature analysis.

Image generation accesses an OpenAI service through a tool built into Codex, whose interface does not provide a verifiable underlying model version or server-side update history. In addition, the generative pipeline relies on fixed decoding and content moderation, and its remote end-to-end time and the baselines' local processing times are measured on different bases. The active-phase throughput results show that the generative workflow processed images more slowly than both conventional baselines in this experiment; computational resource consumption was not measured on a comparable basis.

5. Conclusion

This paper first compares four prompt variants. Relative to P1, the structured prompts P2 and P3 improve mIoU by approximately 0.13–0.14, showing that prompt design substantially affects the generative pipeline's results. P3, which emphasizes object coverage, visible regions, and preservation of spatial structure, is selected according to the prespecified rule. After the prompt and other conditions are fixed, robustness is confirmed using an independent set of 120 images.

Across the six prespecified primary corruption conditions, the generative pipeline achieves a mean corrupted-image mIoU of 0.5756, exceeding 0.4727 for YOLO26s-seg and 0.4926 for Mask R-CNN. Its performance drop from clean to corrupted images is also substantially smaller. The jointly correct foreground-pixel analysis further supports this pattern. The experiment therefore supports stronger pipeline-level robustness for the generative pipeline under the specific seven-class COCO task and image corruptions studied here. However, the conventional baselines perform better on clean images and under some blur conditions, while the generative pipeline depends on its output format and runs more slowly.

Future work can proceed in three directions. First, the evaluation can be extended to more datasets, target classes, and corruption types, with robust segmentation baselines trained using corruption augmentation and more detailed boundary ground truth to test the pipeline's potential advantage in local boundary handling. Second, repeated generations for each image can be used to estimate variance arising from generation randomness and service changes. Finally, methods for reducing the pipeline's runtime and resource use can be explored. When complete usage information becomes available from the tool, the number of calls, token consumption, and processing time should also be reported to compare the trade-offs among accuracy, robustness, and cost.

References

[1] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA. pp. 3431–3440. https://openaccess.thecvf.com/content_cvpr_2015/html/Long_Fully_Convolutional_Networks_2015_CVPR_paper.html

[2] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA. pp. 779–788. https://openaccess.thecvf.com/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html

[3] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy. pp. 2961–2969. https://openaccess.thecvf.com/content_iccv_2017/html/He_Mask_R-CNN_ICCV_2017_paper.html

[4] Ultralytics. (2026). Instance segmentation with Ultralytics YOLO. https://docs.ultralytics.com/tasks/segment/

DOI: 10.54254/2755-2721/2027.39456

[5] PyTorch. (2026). maskrcnn_resnet50_fpn_v2. https://docs.pytorch.org/vision/main/models/generated/torchvision.models.detection.maskrcnn_resnet50_fpn_v2.html

[6] Gan, Y., Park, S., Schubert, A., Philippakis, A., & Alaa, A. M. (2023). InstructCV: Instruction-tuned text-to-image diffusion models as vision generalists. arXiv:2310.00390, v3. https://arxiv.org/html/2310.00390v3

[7] Gabeur, V., Long, S., Peng, S., et al. (2026). Image generators are generalist vision learners. arXiv:2604.20329, v3. https://arxiv.org/html/2604.20329v3

[8] Di, S., Wang, Z., & Xie, W. (2026). Are image generators zero-shot perceivers? A rigorous evaluation. arXiv:2609.07884, v1. https://arxiv.org/html/2609.07884v1

[9] Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A. S., Bethge, M., & Brendel, W. (2019). Benchmarking robustness in object detection: Autonomous driving when winter is coming. arXiv:1907.07484, v2. https://arxiv.org/html/1907.07484v2

[10] Kamann, C., & Rother, C. (2020). Benchmarking the robustness of semantic segmentation models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA (held virtually). pp. 8828–8838. https://openaccess.thecvf.com/content_CVPR_2020/html/Kamann_Benchmarking_the_Robustness_of_Semantic_Segmentation_Models_CVPR_2020_paper.html

[11] OpenAI. (2026). Image generation. *OpenAI*. https://developers.openai.com/api/docs/guides/tools-image-generation

[12] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*. Zurich, Switzerland. pp. 740–755. https://arxiv.org/abs/1405.0312

[13] Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Chapman & Hall. New York.