Text and Click inputs for unambiguous open vocabulary instance segmentation

Nikolai Warner*[1]
nwarner30@gatech.edu
Meera Hahn[2]
meerahahn@google.com
Jonathan Huang[2]
jonathanhuang@google.com
Irfan Essa[1,2]
irfan@gatech.edu
Vighnesh Birodkar[2]
vighneshb@google.com

[1] Machine Learning Center
Georgia Institute of Technology
Atlanta, GA

[2] Google, Inc.
Mountain View, CA

* Work performed partially while at an internship at Google.

Abstract

Segmentation localizes objects in an image on a fine-grained per-pixel scale. Segmentation benefits by humans-in-the-loop to provide additional input of objects to segment using a combination of foreground or background clicks. Tasks include photo-editing or novel dataset annotation, where human annotators leverage an existing segmentation model instead of drawing raw pixel level annotations. We propose a new segmentation process, Text + Click segmentation, where a model takes as input an image, a text phrase describing a class to segment, and a single foreground click specifying the instance to segment. Compared to previous approaches, we leverage open-vocabulary image-text models to support a wide-range of text prompts. Conditioning segmentations on text prompts improves the accuracy of segmentations on novel or unseen classes. We demonstrate that the combination of a single user-specified foreground click and a text prompt allows a model to better disambiguate overlapping or co-occurring semantic categories, such as "tie", "suit", and "person". We study these results across common segmentation datasets such as refCOCO, COCO, VOC, and OpenImages.

1 Introduction

Instance segmentation is the problem of labelling every single pixel that belongs to a known set of categories. Deep-learning based methods have shown tremendous progress in recent years with early works such as Mask R-CNN [18] and more recently with Cascade R-CNN [2], SOLOv2 [33] and MaskFormer [9]. Although broadly applicable when we have a lot of labeled data, fully supervised instance segmentation methods are limited to the set of categories they are trained on. In this paper, we explore a model that can be more useful by taking inputs from the user about what objects to segment. We ask for 2 inputs: (i) a single click on the object to be segmented and (ii) a text description of the same object.

(a) Input and click(•) (b) No-text baseline (c) Text = "Tie" (d) Text = "Person"

Figure 1: The benefit of text input for instance segmentation. The model in 1b struggles to guess the correct object based on only the point input from 1a. Our approach, which takes both text and click as input is successfully able to segment 1c and 1d. Both models are trained on OpenImages with 64 seen classes.

In isolation, each of these modalities is insufficient to unambiguously designate a single instance to be segmented. For example, consider the click in Figure 1a. It is unclear what the user wants to segment based on this one input. The user could mean they want to select the whole person or just the tie or shirt. This lack of specificity is also reflected in a model trained on single-click data, as seen in Figure 1b. Similarly, text input alone can also be ambiguous — for example, using "car" as text input would be insufficient to describe a single instance if there are multiple cars in an image. Though there are ways to address this ambiguity through the use of referring expressions [7, 19, 25, 38], these approaches place a heavy burden on the user to carefully construct perfectly unambiguous text phrases. Together however, a Click + Text input mechanism is a simple low-effort way to unambiguously designate an instance in an image to be segmented.

A similar framework was first delineated by the PhraseClick [12] paper, which proposed an architecture that takes text as input using a bi-directional LSTM. Although PhraseClick addresses the ambiguity problem, it does so in a class specific manner. Their approach only learns to model the classes in their training dataset, and has no way to generalize beyond the set of words that it sees during training.

Our model uses the same set of inputs as PhraseClick (Click + Text), but goes beyond the fixed set of words it observes during training. To do so, we leverage the generalization abilities of image-text models such as CLIP [28], which have demonstrated zero-shot generalization abilities by learning from web-scale image/text pairs. Specifically, our method relies on saliency maps extracted from CLIP style models (e.g. using recent approaches such as MaskCLIP [40] or Transformer Explainability by Chefer et al [3]). These "text saliency" methods allow us to gauge the relevance of each pixel in an image to a given text-query. Because models like CLIP [28] are trained on large, open-vocabulary datasets, approaches like MaskCLIP [40] gives us a coarse, semantic-level understanding of a wide variety of concepts (see heatmap examples in Appendix). And combined with a click from the user, this gives us precise information about which instance they want to segment.

The benefit of using text and click can be seen in Figure 1c and 1d. Our model can successfully use the input text to predict 2 different objects given the same input point, and it is able to do so for text inputs beyond the categories it has seen during training time.

Our main contributions are as follows:

1. We propose to condition segmentation models on text by leveraging pre-trained CLIP models using MaskCLIP to generate a per-pixel saliency that is used as input to our model and show our approach to be effective for novel category generalization.
2. We show that our approach matches or exceeds the performance of the PhraseClick method [12] while generalizing to many more categories.

3. We compare with the recent Segment Anything (SAM) [20] model and show that we outperform it on the task of segmenting instances based on single click and text as input, while training on a much smaller dataset.

We also experiment with truly open-vocabulary setting on queries far out of distribution from academic datasets. As evident in Figure 2, our model performs well on classes that were outside of seen or unseen sets within the training data, on images completely distinct from our training or validation data.

(a) Our prediction for 'Model globe' and 'Basket'

(b) Our prediction for 'Kayak paddle' and 'Helmet'

(c) Our prediction for 'Microscope' and 'Hairnet'

Figure 2: Open vocabulary queries demonstrated on images from the web. These categories include 'Kayak Paddle', 'Basket', and 'Microscope' which are never seen by the model.

2 Related Work

Semantic / Instance Segmentation. Semantic segmentation is the problem of assigning a semantic label to each pixel in an image [27]. Because it requires a large dataset of dense annotations however, it can be time-consuming and expensive to crowd-source. Training segmentation models in new or niche domains therefore is constrained by data annotation availability and cost. State of the art semantic segmentation techniques employ a fully convolutional architecture that combine low level and high level feature maps for accurate segmentation masks [17]. Deeplab V3 [4] uses atrous convolutions to capture objects and features at multiple scales spanning large and small and its successor DeeplabV3+ [5] remains a strong SOTA segmentation architecture, adding a decoder module to Deeplab V3 to improve segmentation quality along object boundaries. Another class of state of the art segmentation models are based on Vision Transformers (or ViT) [14], and extend it to segmentation by decoding image patch embeddings from ViT to obtain class labels (e.g., [32]) This family includes SegViT [39] that proposes to better use the attention mechanisms of ViT to gener-

ate mask proposals, as well as ViT-Adapter-L [8] that attempts to correct weak priors in ViT using a pre-training-free adapter.

Interactive Object Segmentation. Interactive object segmentation seeks to utilize additional human inputs such as clicks or bounding boxes at inference time to guide/refine a segmentation. Deep interactive object detection [36] use a novel strategy to select foreground and background points from an image, which are transformed via Euclidean distance maps in to channels that can be used as inputs into a convolutional network. PhraseClick[12] explores how to produce interactive segmentation masks using text phrases in a fully supervised manner as an additional modality of input. They demonstrate that adding phrase information reduces the number of interactions required to achieve a given segmentation performance, as measured by mIoU.

Sofiiuk et al. [30] highlights the issue with other inference-time optimization procedures in related works and proposes an iterative training procedure with a simple feedforward model. Focal click[6] highlights how existing interactive segmentation models can perform poorly on mask refinement when they destroy the correct parts; and proposes a new method that refines masks in localized areas. SimpleClick [24] explores ViT in the context of interactive segmentation, adding only a patch embedding layer to encode user clicks without extensively modifying the ViT backbone.

Zero Shot Segmentation. ZS3Net[1] performs zero shot semantic segmentation by correlating visual and text features using word2vec [26]. They also introduce a self-training procedure using pseudo-labels for pixels of unseen classes. CAGNet [16] adds a contextual module that takes as input the segmentation backbone output and predicts a pixel-wise feature and contextual latent code per pixel. Their aim is to use more pixel-level information with their feature generator whereas ZS3Net contains a feature generator that uses only semantic word embeddings.

While traditional end-to-end segmentation features are grouped implicitly in convolutional networks, GroupVIT [35] seeks to explicitly semantically group similar image regions into larger segments to perform zero-shot segmentation. It achieves 52.3% mIoU for zero shot accuracy on PASCAL VOC 2012. LSeg [22] trains an image encoder to maximize similarity between the text embedding for a given query and the image embedding of the ground truth pixel classes. SPNet [34] performs inference on unseen classes by utilizing semantic word embeddings trained on a free text corpus such as word2vec or fast-text.

Zegformer [13] achieves impressive results on zero-shot segmentation by "decoupling" the segmentation task into two stages: grouping pixels into likely segments in a class-agnostic manner, and assigning classes to grouped pixels. MaskCLIP [40] achieved SOTA transductive zero-shot semantic segmentation by utilizing a pre-trained CLIP[28] model. They also showed that they can generate psuedo-labels of unseen categories and use it to train a semantic segmentation model. Although this approach can generalize to many classes, it necessitates training a new model for each set of new classes which is costly.

3 Method

Our main objective is to create a model capable of open vocabulary segmentation on novel classes. Figure 3 summarizes our approach to this problem. We take as input to our segmentation model an RGB image, a single foreground click, and a text prompt and produces

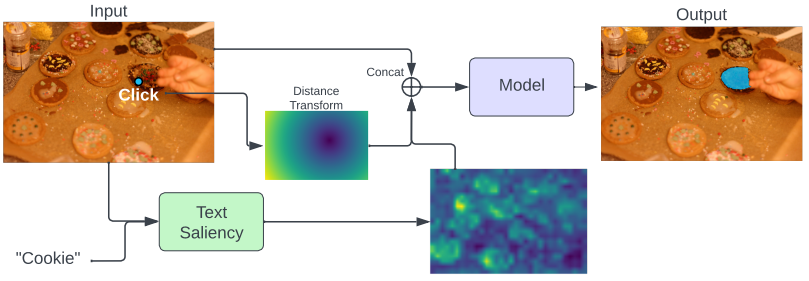
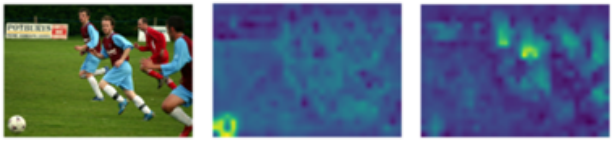


Figure 3: Our model architecture: we take as input a guided foreground click, the RGB image, and a text category. Then, the image-text saliency model (MaskCLIP here) produces a text-weighted feature map helping to localize the instance of interest. Finally, the original RGB image, clickmap, and saliency map are concatenated and fed into a modified fully convectional segmenation model, that accepts as input a 5 channel array.



(a) Input Image (b) "Ball" heatmap (c) "Shirt" heatmap

Figure 4: Example of text saliency heatmaps produced by MaskCLIP [40]. The heatmaps give us a rough estimate of where the input text is localized, while supporting the large vocabulary learnt by CLIP [28].

a class agnostic segmentation mask as output. While there are many possible ways to incorporate click and text cues into such a model, we take a simple but effective approach of encoding both side inputs as additional channels to be concatenated with the original input image, then fed to a standard segmentation network (e.g., DeepLabV3+, which we use in our experiments). Specifically, our foreground click is passed through a Euclidean distance transform to create a map with a continuous range of values normalized to [0, 1]. This is a standard technique in the interactive segmentation literature [36].

In order to convert a text prompt to a single channel image, we passed the text prompt through a text-saliency model to produce a spatially sensitive guess (i.e., a saliency heatmap) of what pixels are similar to a given text query. In our experiments, we use the MaskCLIP text-saliency model model [40] which allows us to effectively incorporate a textually-sensitive, spatial saliency map that takes as input any open vocabulary text prompt. We note that MaskCLIP builds on the CLIP vision-language model that learns to align similar images and text queries via its massive web-scale dataset of image-caption pairs and contrastive learning scheme. In Figure 4 we visualize the output of this method.

In our experiments, we have informally tried several saliency methods such as GradCAM [29], Generic Transformer Interpretability [3], and MaskCLIP [40]. From qualitative experiments, we observed the best results from MaskCLIP, and it also represents a strong baseline that is easy to implement with a few changes to the encoder layer of CLIP.

Our choice of converting a text prompt to a single channel image is nonstandard; however, we argue that it has a number of benefits. In using a powerful text-saliency model, we significantly lessen the burden on our own segmentation network since its task can now be viewed as that of refining a (admittedly) rough initial segmentation into a clean segmentation given the image and click. Moreover, since this saliency heatmap representation is itself class agnostic, our network should conceptually generalize well to classes that it did not get

to see at training time (and we show that this is indeed the case in our experiments).

As a contrast, the PhraseClick paper [12] embeds text inputs with Word2Vec and uses a bidirectional LSTM to model contextual relations between words in a phrase. However their image and text vector representations are not explicitly aligned; the image embedding vector is simply produced from a global pooling operation. Moreover, their model is not open-vocabulary, it is limited to a fixed set of prompts introduced during training.

4 Experiments

To measure our model's ability to generalize to novel classes, we train our model on a subset of all classes in the dataset (which we call "seen classes"), but at test time evaluate the trained model on the remaining classes (called "unseen classes") as well as all classes present in the dataset. Where available (with the exception of OpenImages) we follow the standard zero-shot segmentation literature splits of "seen" and "unseen" classes in our experiments.

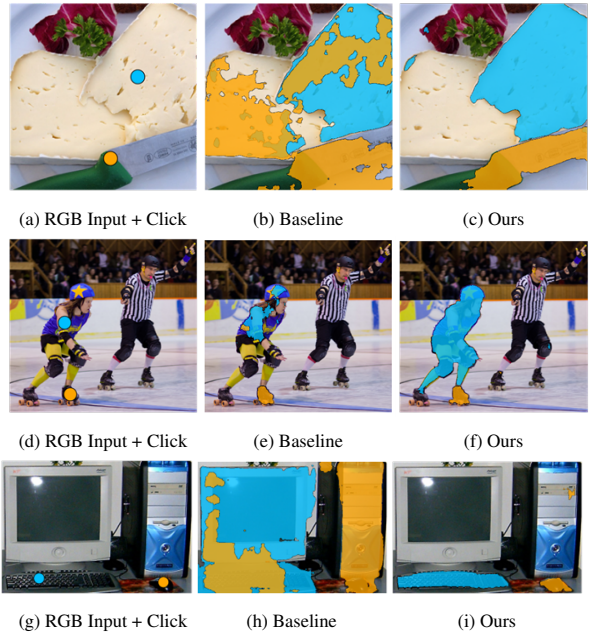
In our experiments, we modify the first layer of a DeepLabv3+ model (with ResNet backbone) to accept a 5 channel image as input, and train all layers from scratch. We modify the number of output classes in the mask prediction module to 2 (to delineate foreground/background) as we perform inference on each individual instance, and not all instances in a given image. We use standard hyperparameters (based on the MMSeg implementation [10]) for DeeplabV3+ and train on 2 Nvidia A40 GPUs with a batch size of 32. Our heatmaps and clickmaps are normalized per instance, to scale values between $[-1, 1]$.

To generate clicks for training, we sample a random point within the ground-truth segmentation mask boundary. Building off of standard interactive segmentation literature [36], positive points are selected to be at least some minimum distance from the object border, and a minimum distance from other positive points. Negative points are sampled using a variety of strategies: first, from points near the border of the object mask boundary; second, from points in other object instances in the same image that we are not trying to segment.

We train separate models for the Pascal VOC [15], COCO [23], refCOCO [38], and OpenImages datasets [21]. We train models in two configurations: zero-shot segmentation, and fully-supervised segmentation. In the former, the model has access only to instances in the limited set of seen-classes and RGB images that contain instances of those seen-class sets. For VOC, we use the 5 seen-class set defined in the ZS3 [1] out of 20 total classes. For refCOCO and COCO, we use the standard 20/60 split of segmentation classes proposed in prior zero-shot segmentation literature. For our OpenImages experiments, we found no prior standard split for zero-shot segmentation, and there are 350 total segmentation classes. Thus, we use the intersection of the COCO classes and OpenImages segmentation classes as our seen set, resulting in 64 seen classes for training ($\sim 20\%$ of total classes). All results are reported at 90K iterations unless otherwise stated.

4.1 Novel class generalization

In Table 1 we show that across all 4 datasets studied, conditioning on text-saliency improves overall mIoU across the board; and that this improvement mostly comes from larger improvements on the set of unseen classes. For example, on COCO, our heatmap-based model achieves 1.72 mIoU greater than baseline on seen classes, but 6.98 mIoU greater than baseline on unseen classes. In other words, the model is able to use the heatmaps to noticeably improve the quality of unseen class segmentations.



(a) RGB Input + Click (b) Baseline (c) Ours

(d) RGB Input + Click (e) Baseline (f) Ours

(g) RGB Input + Click (h) Baseline (i) Ours

Figure 5: Inference examples on unseen classes for baseline versus our model for (a) “Cheese” and “Knife”, (d) “Roller Skates” and “Woman”, and (g) "Keyboard" and "Mouse". Conditioning on text saliency improves novel class segmentation and removes ambiguity. Model trained on OpenImages with 64 classes set as seen, compared to the click-only baseline without heatmaps.

Dataset	Text Input	mIoU		
		Overall	Seen	Unseen
refCOCO	✓	66.02 (+3.03)	70.30 (+1.86)	56.35 (+5.68)
refCOCO		62.99	68.44	50.67
VOC	✓	57.76 (+4.52)	59.31 (+3.2)	50.73 (+10.45)
VOC		53.24	56.11	40.28
COCO	✓	38.42 (+3.89)	42.06 (+1.72)	33.45 (+6.98)
COCO		34.53	40.34	26.47
OpenImages	✓	57.05 (+4.40)	67.03 (+3.35)	53.92 (+4.74)
OpenImages		52.65	63.68	49.18

Table 1: Results for Text+Click model on seen and unseen classes. We used one click for all models and trained using only seen classes. For OpenImages we use 64 seen classes. We convert text input to a heatmap using Maskclip.

Moreover, the smaller the seen class set, the greater the benefit of conditioning the segmentation network on text saliency. We study this effect in Table 2, where we vary the fraction of classes designated as “seen” in the OpenImages dataset. Here we see that the improvement increases as number of seen classes decreases; this intuitively makes sense as our technique of converting to a saliency map places the main burden of novel class generalization on the pretrained CLIP model rather than the segmentation network itself.

Seen Classes	Text Input	mIoU		
		Overall	Seen	Unseen
64	✓	57.05 (+4.4)	67.03 (+3.35)	53.92 (+4.74)
64		52.65	63.68	49.18
34	✓	55.10 (+5.82)	62.03 (+5.19)	52.95 (+6.12)
34		49.28	56.84	46.83
23	✓	53.65 (+7.89)	61.64 (+8.38)	51.14 (+7.62)
23		45.86	53.26	43.53

Table 2: Difference in performance as the number of seen classes in OpenImages changes. Note that gap between our approach (Text+Click) and the click-only baseline increases with a smaller set of seen classes. We convert text input to a heatmap using Maskclip.

4.2 Qualitative examples

In Figure 5 we provide several qualitative examples of our inference results. In all of the examples, we click on unseen classes (e.g., "cheese", "knife", "roller skates", etc). Here we use a model trained on OpenImages with 64 classes set as seen, and compare to a simplified click-only baseline (same architecture) but without text saliency heatmaps as input. In the cheese and knife image for example, the baseline aims to separate object instances by features, but is confused by the overlapping textures from the cheese and knife instances. However, our model conditioned on text is able to clearly distinguish the separate cheese instances and separate them from the knife.

4.3 Comparison with SAM

Dataset	SAM		1-Click	Ours
	CLIP	Conf.		
COCO	36.43	39.31	36.82	**47.17**
refCOCO	47.07	52.48	66.16	**68.07**

Table 3: Comparing mIOU of our model with SAM[20]. SAM outputs 3 predictions and we choose one using SAM's confience (Conf.) or CLIP score(CLIP). Our models trained on all classes in COCO and refCOCO outperform SAM.

Dataset	Model	mIoU		
		Overall	Seen	Unseen
COCO	Ours	38.42	**42.06**	33.45
COCO	SAM	**39.31**	41.73	**37.59**
refCOCO	Ours	**66.02**	**70.30**	**56.35**
refCOCO	SAM	52.48	61.18	48.64
OI	Ours	57.05	**63.68**	53.92
OI	SAM	**63.88**	63.60	**64.47**

Table 4: Comparison with SAM while our model only trains on a subset of classes. Note that we outperform SAM on refCOCO. We use SAM's confience score to rank proposals in this experiment because it showed better results in Table 3. OI=OpenImages.

The Segment Anything Model (SAM) [20] is a model that was trained with 1.1 billion masks from the SA-1B dataset. SAM can work with a combination of positive/negative clicks and text prompts and showed impressive segmentation results with user-input. In Table 3 we compare with SAM while taking as input a single class name and a click. In spite of our smaller capacity and limited data, we out-perform SAM when training on all examples from COCO, refCOCO and OpenImages.

Note that a perfect apples-to-apples comparison is difficult here since SA-1B masks are not class-annotated so we are not able to separate seen from unseen masks, given the

(a) RGB Input + Click (b) SAM (c) Ours

(d) RGB Input + Click (e) SAM (f) Ours

Figure 6: Example comparisons to SAM for (a) "Mobile phone", and (b) "Chest of drawers". Model trained on OpenImages with 64 classes set as seen, compared to SAM baseline using highest confidence prediction.

SAM mode an unfair advantage in some ways. SAM outputs 3 predictions which we rank either by CLIP scores or SAM's own confidence scores. For CLIP scores we used the `ViT-L/14@336px` model, which SAM used for open-vocabulary training.[1]

In Table 4 we compare our approach with SAM while only training on a subset of classes. Note that even when we further limit our training set and evaluate our model on a set of classes that our model is *guaranteed* to have not seen, we still outperform SAM on refCOCO. It is important to note that because of SAM's compute requirement, we could not re-train SAM and only evaluated the pre-trained model trained on SA-1B.

5 Conclusion

We set out to explore improved instance segmentation through the use of a single click and a text prompt. A single click is insufficient to specify what part of an instance to segment; a single text prompt can still be ambiguous unless carefully crafted. We have demonstrated that a single click combined with a text prompt outperforms a click-only baseline across a variety of datasets. We also show that a model conditioned on text-saliency can generalize much better to novel categories. We use saliency maps from MaskCLIP to produce rough localizations for any category. A separate segmentation model is trained on the concatenated input, and segments in a class-agnostic manner, while still retaining class-specific information from the MaskCLIP module. The recent SAM model is class-agnostic and struggles to disambiguate user intent on the overall part vs subpart from a single click. Open vocabulary interactive segmentation is a novel task that has numerous applications, from reducing dense image annotation costs to improving background object removal in photo editing. We hope that the new text and click segmentation task will improve the accuracy of segmentations that require user interaction, while constraining the amount of interaction required. Future research directions could include automatically detecting the best category present around a user's foreground click, to remove the necessity of an additional text input. Our work also intersects with research on how to produce refined segmentation masks from a rough or low quality input (bounding box, point, low quality mask).

[1] The text based open-vocabulary model was not made publicly available.

References

[1] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *CoRR*, abs/1906.00817, 2019. URL http://arxiv.org/abs/1906.00817.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2017.

[3] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers, 2021.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. URL http://arxiv.org/abs/1706.05587.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018.

[6] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. Focalclick: towards practical interactive image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1300–1309, 2022.

[7] Yi-Wen Chen, Yi-Hsuan Tsai, Tiantian Wang, Yen-Yu Lin, and Ming-Hsuan Yang. Referring expression object segmentation with caption-aware consistency, 2019.

[8] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022.

[9] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. 2021.

[10] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020.

[11] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. VLT: Vision-language transformer and query generation for referring segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7900–7916, jun 2023. doi: 10.1109/tpami.2022.3217852. URL https://doi.org/10.1109%2Ftpami.2022.3217852.

[12] Henghui et al. Ding. Phraseclick: Toward achieving flexible interactive segmentation by phrase and click. In *ECCV*, 2020.

[13] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. *CoRR*, abs/2112.07910, 2021. URL https://arxiv.org/abs/2112.07910.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

[16] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. Context-aware feature generation for zero-shot semantic segmentation. *CoRR*, abs/2008.06893, 2020. URL https://arxiv.org/abs/2008.06893.

[17] Shijie Hao, Yuan Zhou, and Yanrong Guo. A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406:302–321, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2019.11.118. URL https://www.sciencedirect.com/science/article/pii/S0925231220305476.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. doi: 10.1109/ICCV.2017.322.

[19] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[20] Alexander et al. Kirillov. Segment anything. *arXiv:2304.02643*, 2023.

[21] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

[22] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022.

[23] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[24] Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. Simpleclick: Interactive image segmentation with simple vision transformers. *arXiv preprint arXiv:2210.11006*, 2022.

[25] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[26] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[27] Yujian Mo, Yan Wu, Xinneng Yang, Feilin Liu, and Yujun Liao. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing*, 493:626–646, 2022. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2022.01.005. URL https://www.sciencedirect.com/science/article/pii/S0925231222000054.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[29] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

[30] Konstantin Sofiiuk, Ilya A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3141–3145. IEEE, 2022.

[31] Konstantin et al. Sofiiuk. Reviving iterative training with mask guidance for interactive segmentation. In *ICIP*. IEEE, 2022.

[32] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7262–7272, 2021.

[33] 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, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17721–17732. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/cd3afef9b8b89558cd56638c3631868a-Paper.pdf.

[34] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8256–8265, 2019.

[35] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18134–18144, 2022.

[36] Ning Xu, Brian L. Price, Scott Cohen, Jimei Yang, and Thomas S. Huang. Deep interactive object selection. *CoRR*, abs/1603.04042, 2016. URL http://arxiv.org/abs/1603.04042.

[37] Zhao et al. Yang. Lavt: Language-aware vision transformer for referring image segmentation. In *CVPR*, 2022.

[38] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016.

[39] Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, et al. Segvit: Semantic segmentation with plain vision transformers. *Advances in Neural Information Processing Systems*, 35:4971–4982, 2022.

[40] Chong Zhou, Chen Change Loy, and Bo Dai. Denseclip: Extract free dense labels from CLIP. *CoRR*, abs/2112.01071, 2021. URL https://arxiv.org/abs/2112.01071.