Scientific Research Publishing

Journal of Computer and Communications, 2025, 13(1), 59-71
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

An Analysis of OpenSeeD for Video Semantic Labeling

Jenny Zhu

NYU Center for Data Science GSTEM Program, New York, NY, USA
Email: jz6616@nyu.edu

How to cite this paper: Zhu, J. (2025) An Analysis of OpenSeeD for Video Semantic Labeling. *Journal of Computer and Communications*, **13**, 59-71.
https://doi.org/10.4236/jcc.2025.131005

Received: October 8, 2024
Accepted: January 27, 2025
Published: January 30, 2025

Abstract

Semantic segmentation is a core task in computer vision that allows AI models to interact and understand their surrounding environment. Similarly to how humans subconsciously segment scenes, this ability is crucial for scene understanding. However, a challenge many semantic learning models face is the lack of data. Existing video datasets are limited to short, low-resolution videos that are not representative of real-world examples. Thus, one of our key contributions is a customized semantic segmentation version of the Walking Tours Dataset that features hour-long, high-resolution, real-world data from tours of different cities. Additionally, we evaluate the performance of open-vocabulary, semantic model OpenSeeD on our own custom dataset and discuss future implications.

Keywords

Semantic Segmentation, Detection, Labeling, OpenSeeD, Open-Vocabulary, Walking Tours Dataset, Videos

1. Introduction

Segmentation is a computer vision task that identifies which class each pixel belongs to. In other words, this entails building a model that can not only classify an object but also understand its boundaries (*i.e.* when one object ends and the other starts). This makes segmentation a difficult task but one essential for scene understanding. Applications of semantic segmentation include autonomous vehicles and medical imaging. For example, in healthcare, this automated and efficient process can be used for early disease detection. See **Figure 1** for an example. The most common method used is supervised learning with Convolutional Neural Networks (CNN) [1]. Recent work using CNNs for semantic segmentation achieves impressive results [2] [3]. However, a major challenge with

this approach is with data, specifically type of data and quantity of data.

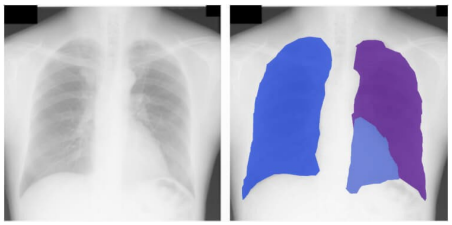

Figure 1. Example of semantic segmentation and a possible application to medical imaging. Different portions of the lung are segmented for analysis.

1) Type of data. Most imaging datasets include images that are pre-processed and don't reflect real-world conditions. For example, the image in **Figure 3** is "cleaned" of any imperfections; however, most real-world data is messy. For example, see **Figure 2** taken from a video of Tesla's self driving car. Since most datasets don't include this type of "uncleaned" data, building a model that can accurately train on real-world data increases the difficulty of the task.

Figure 2. Image data collected by self driving cars. As seen in the figure, the image includes various backgrounds objects such as small street lights or blurry background objects.

Figure 3. Example image from the ImageNet dataset. The ambulance is the center of the image, and extra background details have been removed. These images are hand-selected for only the "clean" images. Thus, such images are not reflective of real-world examples and are not the type we analyze our model with.

2) Data annotation. In supervised machine learning problems, models rely on a ground truth label or the "correct" prediction to calculate the loss and adjust the

weights accordingly. In the case of semantic segmentation problems, however, the ground truth requires each pixel to be manually annotated. This process—creating bounding boxes and instance maps for each image—is tedious and time-consuming. As a result, this limitation on the available data hinders model performance since more data is proven to increase model accuracy [4]. More importantly, semantic segmentation models require learning from large scale datasets to improve a model's ability to generalize across a wide range of images. Because of this, most semantic tasks are trained on a limited, specific dataset and are not generalizable for other tasks.

Thus, our paper presents work to address the two issues above. Firstly, we present an addition to a previous dataset, the Walking Tours Dataset, to alleviate the first problem of data variation. This dataset contains unfiltered images of a person walking through different cities. See section III for more information about the dataset. We also apply a semantic segmentation model called OpenSeeD to the Walking Tours Dataset to segment the images in the video [5]. The model we use is called OpenSeeD [5]. The results of this model can be used as the ground truth labels for future models. Thus, removing the need to manually annotate the labels.

Our research goal is to analyze the effectiveness of OpenSeeD on our novel dataset and provide suggestions for future steps with this model and its possible applications.

2. Method

A) *Semantic Segmentation Models*

Convolutional Neural Networks (CNNs) [6] have been used in semantic segmentation, achieving good performance. These models generally use traditional CNN architecture with an encoder and decoder layer. The image is passed into the models and down-sampled to lower-resolution feature mappings to extract the most important features. Afterwards, the decoder up samples these feature representations into a full image for the model to learn where objects are. More recently, Transformers [7] have been used in semantic segmentation tasks, showing even more promising results. DETR is the first to use a Transformer encoder and decoder architecture and bipartite matching algorithms [8]. There work shows comparable performance on object detection tasks with earlier fine tuned CNN models. Mask2Former applies Transformer on image segmentation tasks and outperforms the best specialized architecture on four popular datasets [9]. Mask DINO, which improves DETR by using Improved De-noising Anchor Boxes, significantly outperforms existing specialized segmentation methods [10]. The model we use in this paper is called OpenSeeD, a Transformer based model that further improves Mask DINO.

B) *OpenSeeD*

In our work, we analyze the segmentation model OpenSeeD, an open vocabulary segmentation and detection model. An open-vocabulary model is one that can classify object labels outside of those used in training. For instance, if the

model only learned the label "toy" in training, it should still be able to classify objects with the label "toy elephant" [11] outside of training. OpenSeeD is different than previous open-vocabulary models [12] [13] because OpenSeeD combines segmentation and detection. This allows OpenSeeD to achieve better results against baseline models for panoptic segmentation, instance segmentation, and semantic segmentation. For our paper, we focus on semantic segmentation.

The model OpenSeeD is unique because it explores the connection between segmentation and detection. The model is built on the framework of Mask DINO. Mask DINO extends DINO by adding a mask prediction branch. This supports segmentation task in addition to existing object detection task. OpenSeeD further extends Mask DINO by introducing conditioned mask decoding and dividing object queries into foreground and background queries. By unifying detection and segmentation in one learning process, OpenSeeD can leverage more supervision than other Open-Vocabulary models and achieve better performance on most open datasets.

During OpenSeeD's training, 300 latent queries, 9 decoder layers and 100 panoptic queries were used. Pre-trained Swin-T/L [14] is used as the visual backbone and UniCL [15] for the language backbone. It does not use other image-text pairs or grounding data for pre-training. During pre-training, for segmentation purpose, the mini-batch size is set to 32; for detection purpose, it is later set to 64. The image resolution is 1024 by 1024 for both segmentation and detection. In fine-tuning, the image sizes of 512 by 512 from Cityscapes [16] and 640 by 640 from ADE20k [17] are used. The model is pre-trained with the AdamW [18] optimizer and a learning rate of 0.0001, decaying at 0.9 and 0.95 fractions of total number steps by 10. The dataset used in OpenSeeD training are COCO [19] for segmentation annotations and Object365 [20] for detection.

3. Walking Tours Dataset

The Walking Tours Dataset includes 10 videos of a person walking through 10 different cities: Amsterdam, Chiang Mai, Kuala Lampur, Stockholm, Wildlife, Bangkok, Istanbul, Singapore, Venice, Zurich. These videos are from the YouTube channel "PopTravel", a channel that includes a collection of walking tours around various cities. They are downloaded (with a Creative Commons License) from previous work that was inspired by Wiles *et al.* [21]. They are captured in 4K resolution at 60 frames per second. See **Figure 4** and **Figure 5** for example images.

One of our contributions is the addition of three tours: Rome, Torun, and Poznan downloaded with 4K resolution [17]. The tour of Poznan is different than previous tours in Walking Tours because it is filmed at night (the others were filmed during the day). We chose to add this tour to see if the model could predict images with different lighting equally well. See **Figure 6** for an example Poznan image. Moreover, even though our other contribution tour Torun is of a city, it is different than previous city tours because it includes city scenes with almost no people and many frames where there isn't a single person. Lastly, the tour of Rome

includes footage similar to the other tours to validate the results from previous work. These contributions add to the variety to the dataset as it includes examples with different lighting and edge cases (city scenes without people) to ensure the quality of OpenSeeD's prediction.

Figure 4. Example image of Venice tour from the walking tours dataset.

Figure 5. Out of the 10 walking tour videos, the wildlife tour is unique as it includes nature scenery rather than city life.

Figure 6. One of our contribution videos. This image is of the Poznan tour filmed at night.

See **Table 1** for a detailed description of the Walking Tours Dataset and **Table 2** for our contribution videos.

A) Comparison with Other Datasets

Previous datasets such as ADE20K [17] or Pascal [22] differ from Walking Tours because their datasets lack images with background information. Their images zoom in on one object while ours is a panorama view of the entire city. An

example can be seen in Figure 7. Additionally, the average video length for our Walking Tours Dataset is 1 hour 22 minutes in comparison to the 10-second-long videos by ImageNet [23]. Our dataset also contains higher-resolution images, allowing the model to predict more specific objects in the image.

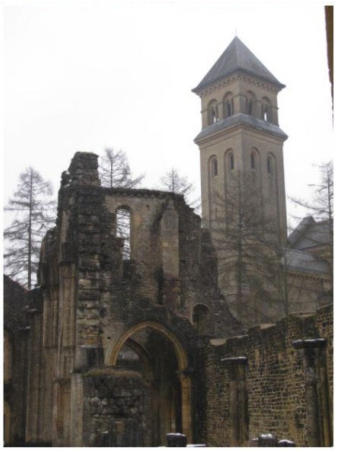

Figure 7. Image from the ADE20k dataset. The only object in the image is the church.

Table 1. Video information for each tour in the walking tours dataset. Frame interval is the number of frames we skip per video to ensure that every tour has a similar number of images. The combined total number of frames is 10,728.

Tours	Total Frames	Frame Interval	Total Images
Venice	197,800	150	1319
Bangkok	314,684	300	1049
Singapore	174,011	170	1024
Amsterdam	147,377	140	1053
Wildlife	108,115	100	1082
Chiang Mai	122,257	120	1019
Zurich	116,989	110	1064
Kuala Lampur	131,100	130	1009
Stockholm	119,686	110	1089
Istanbul	122,400	120	1020

Table 2. Video information for each tour in our contribution dataset. The combined total is 2061.

Tours	Total Frames	Frame Interval	Total Images
Rome	301,487	450	670
Poznan	71,927	100	720
Torun	147,450	220	671

In summary, our Walking Tours Dataset and contribution dataset have a few advantages compared to previous datasets.

1) Realistic representation of real-world data: Previous imaging datasets such as ImageNet include data that is cleaned or zoomed in on a specific object. Since the Walking Tours Dataset has "uncleaned" data, it better represents real-world examples, allowing future models to better predict on real-world examples.

2) Various objects per frame: Each frame in Walking Tours contains various objects and their performing actions

3) Transitions: Since the videos are captured by someone walking, the transitions from one frame to another are smooth. It captures the slow transition from different scenes (city to shops to markets), the transition of people from one place to another, and the transition of different lighting.

4. Experiments

For our main experiment, we fed the Walking Tour Dataset to OpenSeeD and analyzed the results. To do this, we passed the model configurations, weights, and path to our dataset as the inputs. We also passed in the Meta Data, specifying the classes our model should predict and the color of each class. For our "stuff classes", we used the ADE20k classes as a lot of those classes appear in Walking Tours as well. Our "stuff colors" was randomly generated with a NumPy random seed of 14. For each frame, we saved the output image, the semantic segmentation map, the number of pixels each object took up, and how frequently each class appeared in a frame. At the end, we saved the total number of frames each object appeared in and total number of pixels it occupied.

5. Discussion

We analyze our results both qualitatively and quantitatively. Our qualitative analysis allows us to summarize trends we observe while the quantitative results analyzes those specific trends in detail.

A) Qualitative Analysis

1) Accomplishments: We found that OpenSeeD did a good job segmenting specific details in images. We can see in **Figure 8** that the model segments small objects like light, signboards, and plates. We saw this trend of segmenting small images happen frequently for our results.

The model also did well on images with different lighting and different settings. For the Poznan tour (**Figure 9**) that was filmed at night, the model consistently segmented the lights, buildings, and people. Moreover, the model could detect objects from different settings such those in a crowded city road (**Figure 8**) as well those near a river side (**Figure 10**). One possible reason why the model did well segmenting these objects is that the training data frequently contained these objects. For instance, the COCO dataset is of specific objects hence the name "Common Objects in Context". Similarly, the Objects365 dataset contains detailed examples of objects such as "cups", "chair", or "bag".

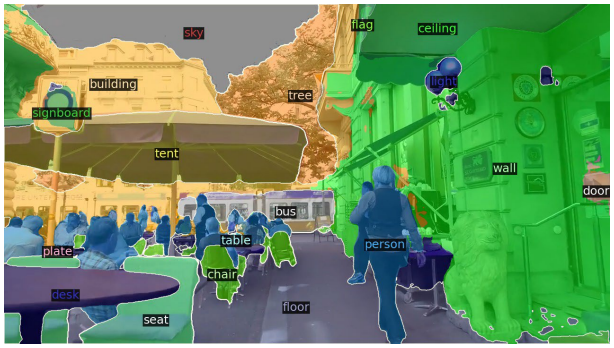

Figure 8. The model's prediction on a prediction in the Zurich dataset in a crowded cityscape. (walking, driving, biking, etc.)

Figure 9. OpenSeeD's prediction on the Poznan tour filmed at night. See **Figure 6** for the original image.

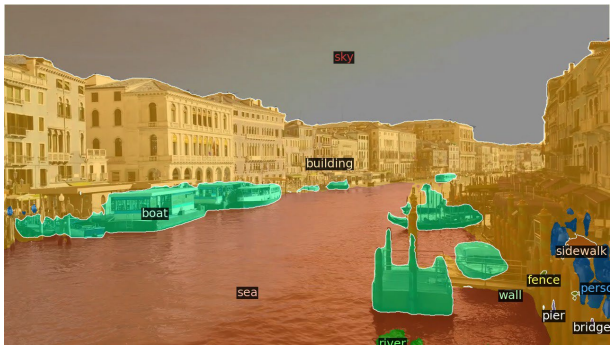

Figure 10. OpenSeeD's prediction on the Venice tour by the riverside. See **Figure 4** for the original image.

2) *Mistakes*: Although OpenSeeD yields very promising results on our datasets with large cities, it does not do as well on the Wildlife dataset. For example, in **Figure 11**, the model misidentifies the elephant as a "vase" and labels the dirt as "sidewalk". One possible reason for these results is the type of data OpenSeeD trained with. For instance, even though the COCO dataset does have the label "animals", it only has "animals" in relation to people or to urban landscapes, not in wildlife. Moreover, the Objects365 dataset rarely contains any animals, focusing purely on people and objects. This explanation is supported by our qualitative observations as in the Wildlife dataset, the model makes mistakes such as labeling

the ground as "sidewalk", which is often found in large cities, rather than "dirt". This also makes sense as the model not only mislabels the elephant but also outlines it poorly, which implies that the model was most likely not trained on images of wild animals. Another possibility that could have added to the poor results is that the stuff classes we passed into the model had a lot of specifics for city life such as "awning" or "ashbin" while the only labels it had for wildlife were "animal", "grass", and "dirt".

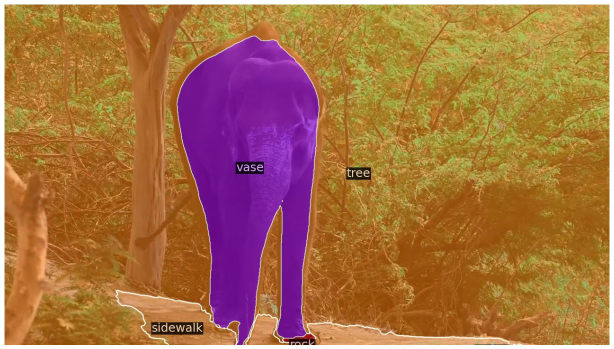

Figure 11. The model's prediction of an elephant image in the Wildlife dataset. It incorrectly predicts it as a "vase".

B) Quantitative Analysis

For our quantitative analysis, we analyze the percentage of total frames each object occupied. We include two graphs: see **Figure 12** for the Venice tour which is similar to the other city tours, and **Figure 13** of the Wildlife dataset as a contrast. For our classes on the graph, we only included objects that appeared over 100 times.

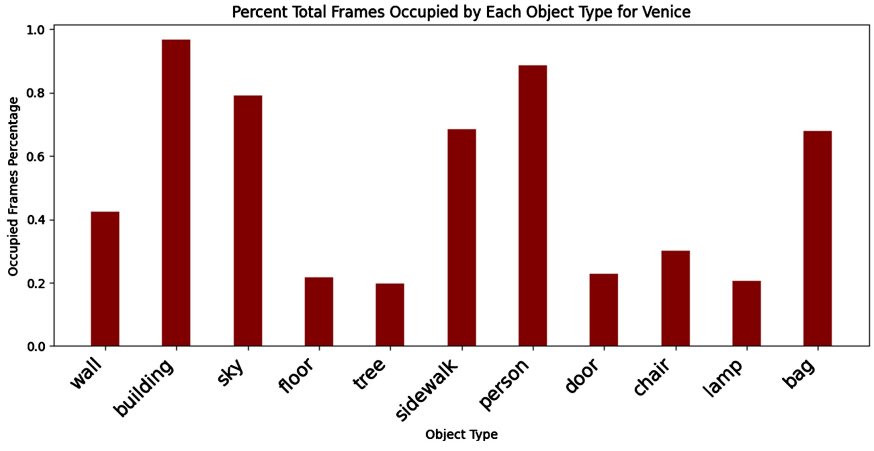

Figure 12. The graph shows the percentage of frames each object occupies for the Venice tour.

As seen from the graph of Amsterdam, almost all the video tours contained large amounts of "building", "sky", "sidewalk", or "person". And it makes sense that our model does a good job predicting those since they are large objects and a single one can take up 80% of a frame.

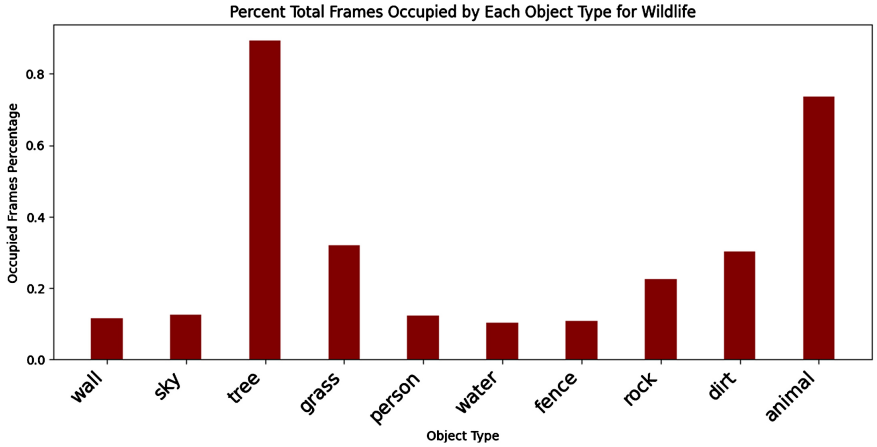

Figure 13. The graph shows the percentage of frames each object occupies for the wildlife tour.

On the other hand, this contrasts with the graph of the Wildlife dataset. In that dataset, the object that occurred the most frequently was "tree" then "animal" then "grass". Although these results superficially make sense, while visually looking through the dataset, every single image had an animal while our model predicted that less than 80% of images had

one. This meant that the model repeatedly misidentified or completely missed the animal in the dataset. Moreover, the model also predicted "person" around 20% of the time, even though our Wildlife dataset did not contain any people at all.

C) Future Research

More generally, although the model does a good job predicting the small details, the outline of these details can be improved. Although in **Figure 8** some objects are segmented well (the "char" and the "bus" e.g.), some still require improvement. For instance, the "signboard" on the top left of the image is incorrectly segmented with parts of the building and the "door" label is incorrectly segmented with parts of the wall. We can also see in the bottom corner there are small splotches of colors with unclear segmentations.

The next step of this research could be to OpenSeeD's results as the ground truth for our own semantic models. This solves the problem of having to manually label ground truth values to allow future models to be trained much more efficiently. This change, combined with the addition of new data, will allow future models to be trained on larger and more diverse datasets. Moreover, OpenSeeD's architecture simplifies the training process and its state-of-the-art performance allows it to be used as a foundational model for future semantic segmentation models.

More directly for practical purposes, OpenSeeD can be applied to tasks like self-driving cars to solve real-world problems. It can be used to detect and more accurately outline details like stoplights, signs, and objects in the distance—expanding the possibilities for self-driving cars.

6. Conclusion

In this paper, we address the issue of the limitation of semantic segmentation datasets as they are hard to collect, small, and not realistic examples. To do this, we contribute our own Walking Tours Dataset that includes hour-long, uncurated videos of realistic scenes. This dataset adds to previous work as it includes more variety with different lighting and scenes. Afterwards, we use OpenSeeD, with its strong semantic segmentation model and open vocabulary, to label the Walking Tours videos. We analyze OpenSeeD's result qualitatively and quantitatively and conclude that it is competitive against large semantic segmentation models built by Meta or Google. In the future, we would like to use our new semantic segmentation dataset to evaluate pre-trained vision models and the results of OpenSeeD as the ground truth.

Acknowledgement

Firstly, I would like to thank Christopher Hoang for his mentorship and guidance throughout my research project. Thank you. I would also like the thank Mengye Ren for accepting me as an intern at his lab. Also, thank you to the Winston Foundation for making this experience possible for me and Catherine Tissot, Matthew Leingang, Priyanshi Singh, and other course assistants for making this experience so amazing.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Long, J., Shelhamer, E. and Darrell, T. (2015) Fully Convolutional Networks for Semantic Segmentation. 2015 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Boston, 7-12 June 2015, 3431-3440. https://doi.org/10.1109/cvpr.2015.7298965

[2] Li, S., Ke, L., Danelljan, M., Piccinelli, L., Segu, M., Gool, L.V., *et al.* (2024) Matching Anything by Segmenting Anything. 2024 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Seattle, 16-22 June 2024, 18963-18973. https://doi.org/10.1109/cvpr52733.2024.01794

[3] Wang, X.D., Yang, J.F. and Darrell, T. (2024) Segment Anything without Supervision. arXiv: 2406.20081.

[4] Halevy, A., Norvig, P. and Pereira, F. (2009) The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems*, **24**, 8-12. https://doi.org/10.1109/mis.2009.36

[5] Zhang, H., Li, F., Zou, X., Liu, S., Li, C., Yang, J., *et al.* (2023) A Simple Framework for Open-Vocabulary Segmentation and Detection. 2023 *IEEE/CVF International Conference on Computer Vision* (*ICCV*), Paris, 1-6 October 2023, 1020-1031. https://doi.org/10.1109/iccv51070.2023.00100

[6] Ronneberger, O., Fischer, P. and Brox, T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W. and Frangi, A., Eds., *Medical Image Computing and Computer-Assisted Intervention—MICCAI* 2015, Springer International Publishing, 234-241.

https://doi.org/10.1007/978-3-319-24574-4_28

[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017) Attention Is All You Need. 31*st Conference on Neural Information Processing Systems* (*NIPS* 2017), Long Beach, 4-9 December 2017, 5998-6008.

[8] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. and Zagoruyko, S. (2020) End-to-end Object Detection with Transformers. In: Vedaldi, A., Bischof, H., Brox, T. and Frahm, J.M., Eds., *Computer Vision—ECCV* 2020, Springer International Publishing, 213-229. https://doi.org/10.1007/978-3-030-58452-8_13

[9] Cheng, B., Misra, I., Schwing, A.G., Kirillov, A. and Girdhar, R. (2022) Masked-attention Mask Transformer for Universal Image Segmentation. 2022 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), New Orleans, 18-24 June 2022, 1280-1289. https://doi.org/10.1109/cvpr52688.2022.00135

[10] Li, F., Zhang, H., Liu, S.L., Zhang, L., Ni, L.M., Shum, H.Y., *et al.* (2022) Mask Dino: Towards a Unified Transformer—Based Framework for Object Detection and Segmentation. arXiv: 2206.02777.

[11] Gu, X.Y., Lin, T.Y., Kuo, W.C. and Cui, Y. (2021) Open-Vocabulary Object Detection via Vision and Language Knowledge Distillation. arXiv: 2104.13921.

[12] Jia, C., Yang, Y.F., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z. and Duerig, T. (2021) Scaling up Visual and Vision-Language Representation Learning with Noisy Text Supervision. *Proceedings of the* 38*th International Conference on Machine Learning*, 18-24 July 2021, 4904-4916.

[13] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., *et al.* (2021) Learning Transferable Visual Models from Natural Language Supervision. 2021 *International Conference on Machine Learning*, 18-24 July 2021, 8748-8763.

[14] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., *et al.* (2021) Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. 2021 *IEEE/CVF International Conference on Computer Vision* (*ICCV*), Montreal, 10-17 October 2021, 9992-10002. https://doi.org/10.1109/iccv48922.2021.00986

[15] Yang, J., Li, C., Zhang, P., Xiao, B., Liu, C., Yuan, L., *et al.* (2022. Unified Contrastive Learning in Image-Text-Label Space. 2022 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), New Orleans, 18-24 June 2022, 19141-19151. https://doi.org/10.1109/cvpr52688.2022.01857

[16] Cordts, M., Omran, M., Ramos, S., Scharwachter, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. and Schiele, B. (2016) The Cityscapes Dataset. CVPR Workshop on the Future.

[17] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A. and Torralba, A. (2017) Scene Parsing through ADE20K Dataset. 2017 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, 21-26 July 2017, 5122-5130. https://doi.org/10.1109/cvpr.2017.544

[18] Loshchilov, I. and Hutter, F. (2017) Decoupled Weight Decay Regularization. arXiv: 1711.05101.

[19] Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., *et al.* (2014) Microsoft COCO: Common Objects in Context. In: Fleet, D., Pajdla, T., Schiele, B. AND Tuytelaars, T., Eds., *Computer Vision—ECCV* 2014, Springer International Publishing, 740-755. https://doi.org/10.1007/978-3-319-10602-1_48

[20] Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., *et al.* (2019) Objects365: A Large-Scale, High-Quality Dataset for Object Detection. 2019 *IEEE/CVF International Conference on Computer Vision* (*ICCV*), Seoul, 27 October-2 November 2019,

8429-8438. https://doi.org/10.1109/iccv.2019.00852

[21] Wiles, O., Carreira, J., Barr, I., Zisserman, A. and Malinowski, M. (2023) Compressed Vision for Efficient Video Understanding. In: Wang, L., Gall, J., Chin, T.J., Sato, I. and Chellappa, R., Eds., *Computer Vision—ACCV* 2022, Springer, 679-695. https://doi.org/10.1007/978-3-031-26293-7_40

[22] Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J. and Zisserman, A. (2014) The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, **111**, 98-136. https://doi.org/10.1007/s11263-014-0733-5

[23] Venkataramanan, S., Rizve, M.N., Carreira, J., Asano, Y.M. and Avrithis, Y. (2023) Is ImageNet Worth 1 Video? Learning Strong Image Encoders from 1 Long Unlabelled Video. arXiv: 2310.08584.