Semester Project: **Unsupervised Segmentation**

This project based on paperes Melas et al. [8] and Li et al. [7].

Student: Mekhron Bobokhonov

Supervisor: Baran Ozaydin

January 6, 2024

Abstract

Semantic segmentation is a longstanding challenge in computer vision, involving the breakdown of an image into meaningful segments. This task become especially interesting in an unsupervised setting due to the difficulty and cost of obtaining dense image annotations. Unsupervised semantic segmentation can be solved through a two-step process. Firstly, we can independently segment each image in the dataset by performing clustering of pixels in each image. This yields segmented images without any labels, and we refer to this first step as image-level clustering. In the second step, to assign labels to these segments, we can cluster these segments across the entire dataset. We refer to this second step as dataset-level clustering. In this project, we explore different methods for image-level clustering, including Adaptive Concept Generator and Deep Spectral Methods, as well as dataset-level clusterings, including Attention-Based Clustering, Optimal Transport Clustering, and KMeans.

1 Introduction

Semantic segmentation stands as a fundamental task in computer vision, finding applications in diverse domains such as autonomous driving and medical imaging. The development of deep learning, coupled with the increasing volume of data, has significantly boosted the performance of semantic segmentation by fine-tuning deep neural networks with pixel-level annotations [5]. Despite these successes, acquiring large-scale pixel-level annotations remains costly and time-consuming.

To address this challenge, various forms of weak supervision have been explored to enhance label efficiency [13], including image-level [1, 16], scribble-level [3], and box-level supervision [15]. Beyond these approaches, certain methods have emerged that achieve semantic segmentation without relying on any labels [6, 14], constituting unsupervised semantic segmentation (USS).

Unsupervised semantic segmentation can be solved through a two-step process. Initially, we perform image-level clustering, grouping pixels in each image into meaningful segments that represent distinct 'concepts.' This yields segmented images without any labels. In the second step, we conduct dataset-level clustering, where these segments are clustered across the entire dataset to assign labels. Recently, the self-supervised ViT [10] provides a new paradigm for the first step of the USS, image-level clustering, due to its property of containing semantic information in pixel-level representations. Figure 1 visually illustrates the essence of ViT, revealing that the pixel-level representations it produces contain underlying clusters in the representation space of an image. When projected onto the image, these clusters manifest as semantically consistent groups of pixels or regions, representing intuitive 'concepts.' In the subsequent steps, by employing clustering methods or unsupervised classification, we can assign labels to these concepts, culminating in the attainment of semantic segmentation for images within the dataset.

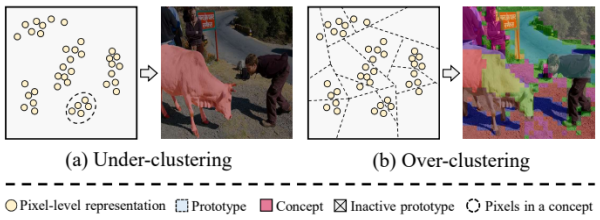

Figure 1: Li et al. [7] In the representation space of an image, the pixel-level representations produced by the self-supervised ViT contain underlying clusters.

The most interesting works to this project in the field of Unsupervised Semantic Segmentation (USS) use the Adaptive Concept Generator [7] or Spectral Methods [8] for image-level clustering and KMeans or an unsupervised classifier for data-level clustering. In this project, our focus is to implement and compare various combinations of these clustering algorithms, along with exploring new types of clustering algorithms such as Attention-Based Clustering and Optimal Transportation Clustering. We observed that attention-based clustering for data-level clustering improves the overall USS results. For evaluation, we utilized the commonly used semantic segmentation dataset PASCAL VOC 2012 [4].

2 Related Work

Vison Transformer. The Transformer, a model primarily based on the self-attention mechanism, has found extensive applications in natural language processing. The Vision Transformer (ViT) [10] is the first pure visual transformer model designed for processing images. Recently, Caron et al. [11] proposed self-distillation with no labels (DINO) to train ViT and discovered a property: its features contain explicit information about the segmentation of an image. Building on DINO, some prior studies have successfully demonstrated the extension of this property to unsupervised dense prediction tasks.

Unsupervised Semantic Segmentation. Self-supervised Vision Transformers (ViTs) trained with DINO have recently been explored for unsupervised dense prediction tasks due to their ability to represent pixel-level semantic relationships. In the context of semantic segmentation, Hamilton et al. [9] train a segmentation head by distilling feature correspondences, encouraging pixel features to form compact clusters and learn improved pixel-level representations. The approaches we are interested in used DINO representations to segment images by region. Melas et al. [8] employ spectral decomposition on the affinity graph to discover meaningful parts in an image (DSM) and implement semantic segmentation. Li et al. [7] explicitly encode segments into learnable prototypes and design the Adaptive Concept Generator (ACG), which adaptively maps these prototypes to informative segments for each image using pixel-level representations. Our focus is to implement and compare various combinations of these clustering algorithms.

3 Method

In this section, we provide a detailed description of all the Unsupervised Semantic Segmentation (USS) methods implemented during this project.

3.1 Image-wise Clustering

In the image-level clustering part USS, our aim is to independently cluster pixels for each image in the dataset, thereby segmenting every image into meaningful segments. Throughout the project, we explored the Deep Spectral Method used by Melas et al. [8] and implemented the Adaptive Concept Generator method from scratch, as utilized by Li et al. [7]. Now, let's delve into the details of each of these methods.

3.1.1 Adaptive Concept Generator (ACG)

Figure 2 illustrates the overall structure of ACSeg. Beginning with an image, the authors first apply a self-supervised Vision Transformer (ViT) to generate pixel-level representations. These representations encapsulate underlying concepts, signifying meaningful groups or regions of pixels. The Adaptive Concept

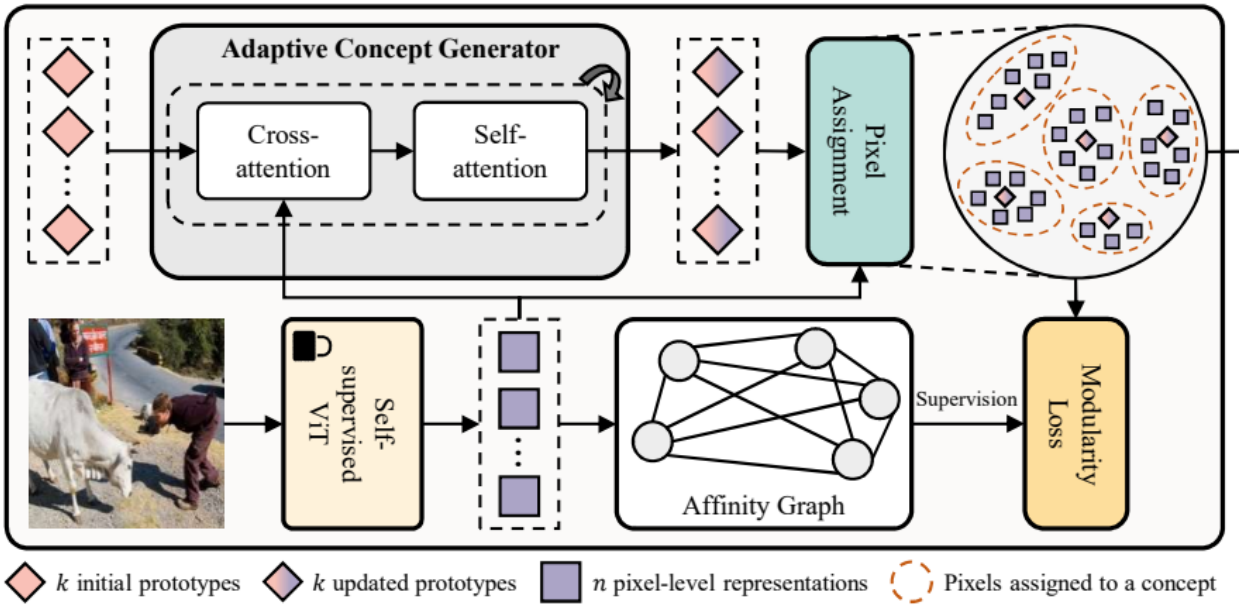

Figure 2: **ACSeg overview: Li et al.** [7]. In their approach, the authors start by employing a self-supervised Vision Transformer (ViT) to extract pixel-level representations, reflecting semantic relationships among pixels. The Adaptive Concept Generator (ACG) dynamically updates the initial prototypes to correspond to underlying concepts in the representation space using scaled dot-product attention. The assignment of pixels is determined by the cosine similarity between pixel-level representations and the concepts. To optimize the ACG, the modularity loss is applied.

Generator (ACG) is designed to explicitly output these concepts. Specifically, the ACG takes a series of learnable prototypes as input and iteratively updates them by interacting with the pixel-level representations, resulting in adaptive concept representations for each image. Finally, the concepts are explicitly represented by pixel groups, obtained by assigning each pixel to the nearest concept in the representation space.

For optimization, the authors proposed a novel loss function called modularity loss to train the ACG without any annotations. The modularity loss operates on pixel pairs intuitively, constructing an affinity graph with pixel-level representations as vertices and their cosine similarity as edges. It calculates the intensity of two pixels belonging to the same concept using the metric defined in modularity [12], thus adjusting the concept representations.

3.1.2 Deep Spectral Methods (DSM)

Figure 3 illustrates the overall structure of DSM. This method first utilizes a ViT-Small [10] trained with DINO [11] as the model to extract pixel-level representations for each image. Then, for each image, a Laplacian matrix L is constructed containing the pairwise affinities of all pixels. The authors perform a three-step process in which they: (1) break each image into segments, (2) compute a feature vector for each segment, and (3) cluster these segments across a collection of images.

For step (1), the authors discretize the first m eigenvectors $y_1, \ldots, y_m$ of L by clustering them across the eigenvector dimension using K-means clustering (for every image separately). For step (2), they take a crop around each segment and compute its feature vector f_s using their self-supervised transformer. For

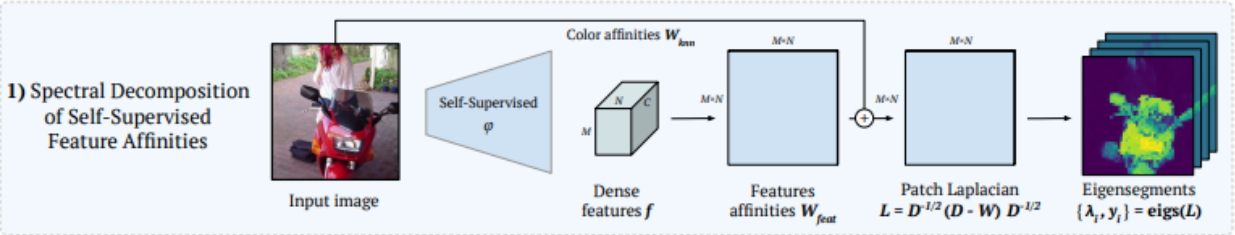

Figure 3: **DSM Overview: Melas et al.** [8] In their approach, the authors combined the advantages of modern self-supervised learning with those of traditional spectral methods for unsupervised segmentation. Given an image, they first extract dense features from a network ϕ and use these to construct a semantic affinity matrix, which is then fused with low-level color information. They subsequently decompose the image into soft segments by computing the eigenvectors of the Laplacian of this matrix.

step (3), they cluster the set of all feature vectors f_s across all images using K-means clustering with k clusters. The second clustering step assigns a label to each segment of each image; adjacent regions with the same label are merged together, effectively reducing the number of segments found per image as needed. At the end of this process, the authors arrive at a set of semantic image segmentations consistent across the entire dataset.

3.2 Dataset-level Clustering

After obtaining segmented images through image-level clustering, the subsequent step involves labeling these segments across the dataset. This implies that two segments from different images capturing objects of the same class should share the same label. Typically, during this step, we take a crop around each segment, compute its feature vector f_s using self-supervised transformer, and then cluster the set of all segments feature vectors f_s across the dataset. In this project, for dataset-level clustering, we explored Attention-Based Clustering, Optimal Transport Clustering, and KMeans clustering algorithms.

3.2.1 Attention-Based Clustering

Figure 4 illustrates the overall structure of Attention-Based Clustering (ABC). The ABC takes a matrix of all feature vectors of segments, denoted as $X = \{f_s\}$, across all images in the dataset as input. It iteratively updates these feature vectors by interacting with a learnable matrix of cluster centers, resulting in the representation of segments in the cluster center space. Finally, segment labels are obtained by assigning each segment to the nearest cluster center in the representation space.

For optimization, we propose a custom loss function to train ABC without any annotations. The loss consists of two parts: coverage and disjointness. The coverage loss ensures that each segment is covered by at least one cluster, controlling cluster sizes to prevent them from being too small or too large. The disjointness part ensures that clusters do not intersect.

Technical Details

The ABC is a cross-attention mechanism that has as parameters a learnable matrix M of cluster centers. As input, it takes a matrix of all feature vectors of segments, denoted as $X = \{f_s\}$, across all images in the dataset and applies cross-attention, taking image features X as *query* and cluster centers M as *key* and *value*. The output of the ABC is a new representation of the images in the cluster center space. The ABC consists of N update steps, and each update step is made up of cross-attention and Feed Forward. With the attention mechanism, the ABC can learn to adjust cluster centers across all segments in the dataset. For implementation, we adopt multi-head attention, layer normalization, and residual connection after the attention operation and the FFN, following the transformer architecture.

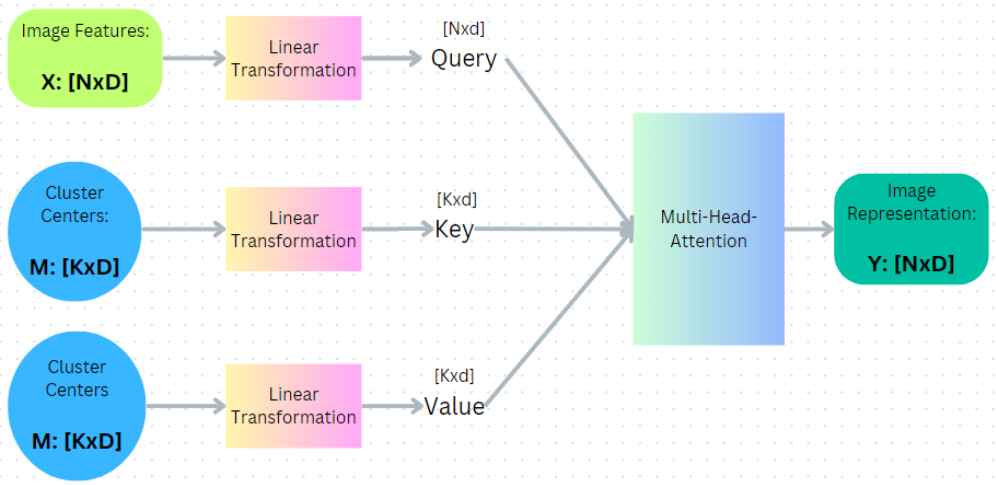

Figure 4: **Attention-Based Clustering.**

- N - dataset size
- K - number of clusters
- D - embedding dimension
- X - matrix of image features $(N \times D)$
- M - trainable matrix of cluster centers $(K \times D)$
- Y - output of the model $(N \times D)$ - new representations of images.

Loss Function For training the ABC, we design a loss function based on the idea of coverage and disjointness of clusters. We will use variables from Figure 4.

We obtain a soft assignment for each segment by calculating the cosine similarity between the output of the ABC and the cluster centers, followed by the application of the ReLU.

$$\text{output} = \text{ReLU}(\cos(Y, M)) \tag{1}$$

Next, we calculate Coverage. In Coverage, we ensure that each segment has the power to attend to at least one cluster, and we also put upper and lower limits on the size of the clusters. With a lot of experiments, we found that 3 the best lower and upper bounds are $\frac{N}{k^2}$ and $\frac{N}{\sqrt{k}}$ respectively.

Coverage: $\forall(m, n) : 1 \leq m \leq N, 1 \leq n \leq K$

$$\sum_j \text{output}_{mj} \geq 1 \;\&\; \frac{N}{k^2} \leq \sum_i \text{output}_{in} \leq \frac{N}{\sqrt{k}}. \tag{2}$$

Finally, we calculate Disjointness. In Disjointness, we assure that all clusters do not intersect.

Disjointness: off-diagonal of $\mathbf{output}^\top \mathbf{output}$ should be zero.

Generally, the calculation of loss has the following steps:

$$\textbf{output} = \mathrm{ReLU}(\cos(Y, M))$$

$$\textbf{coverage} = \mathrm{mean}\left(\mathrm{ReLU}(1 - \textbf{output}.\mathrm{sum}(\mathrm{dim}{=}1))\right) + \mathrm{mean}\left(\mathrm{ReLU}\left(\frac{N}{k^2} - \textbf{output}.\mathrm{sum}(\mathrm{dim}{=}0)\right)\right) + \mathrm{mean}\left(\mathrm{ReLU}\left(\textbf{output}.\mathrm{sum}(\mathrm{dim}{=}0) - \frac{N}{\sqrt{k}}\right)\right)$$

$$\textbf{disjointness} = \text{sum of non-diagonal elements of: } \textbf{output}^\top \textbf{output}$$

$$\textbf{Loss} = \textbf{coverage} + \textbf{disjointness}$$

During the inference phase, each segment is assigned to a specific cluster by determining the argmax of the soft assignment, denoted as **output**.

3.2.2 Optimal Transport clustering

In Optimal Transport clustering, a combination of the KMeans algorithm and an optimal transport solver is used. This is an iterative method designed to partition a dataset into $'k'$ distinct clusters. The process initiates with the random selection of $'k'$ centroids, serving as the initial cluster centers. During the assignment step, a soft assignment to the cluster centers is computed for each data point. This involves creating a matrix of cosine similarities between data points and cluster centers. Sinkhorn's Algorithm [2] is then applied to this matrix to solve an optimal transport problem. The output is a matrix W of soft assignments for data points to cluster centers. Subsequently, the centroids are updated by recalculating their positions as the weighted average of all data points, using corresponding weights from the matrix W. This assignment-update iteration continues until a predefined number of iterations.

4 Experiments

4.1 Image-wise Clustering

We compared DSM [8] and ACG [7] methods as the image-level clustering components for USS on the PASCAL VOC 2012 dataset while using the KMeans algorithm for dataset-level clustering in combination with these methods (**Results** Table 1).

4.1.1 Implementation Details

For the Adaptive Concept Generator (ACG), we used ViT-Small [10] trained with DINO [11] as the model to extract pixel-level representations. *Key* feature tokens, excluding the cls token, from the last layer of ViT serve as corresponding pixel-level representations. For both training and inference of ACG, we resized the width and height of images to 256 and did not employ additional data augmentation.

The ACG is optimized by AdamW with a learning rate of 0.0003 and weight decay of 0.01. Training the ACG was conducted for 400 epochs using a batch size of 1024. The number of update steps in ACG was set to 6, and the number of prototypes was set to 6.

For dataset-level clustering, we explored two approaches for extracting feature representations of segments. The first approach is similar to the procedure used in DSM, where we took a crop around each segment and computed its feature vector, denoted as f_s, using a self-supervised transformer. The second approach involves using masked attention for DINO CLS. In this case, we also took a crop around each segment and computed its feature vector using masked attention for the last attention layer. Finally, we applied K-means clustering to the set of all feature vectors f_s across all images, with the number of clusters set to $k = 21$.

4.1.2 Final Results

See Tables 1 and 5. In our implementation, the ACG method used as an image-level clustering component for the USS showed lower performance compared to the DSM.

4.2 Dataset-wise Clustering

We compared Attention-Based Clustering, Optimal Transport Clustering, and KMeans as the dataset-level clustering components for USS on the PASCAL VOC 2012 dataset while using DSM as the image-level clustering method in combination with all these algorithms.

4.2.1 Implementation Details

Attention-Based Clustering (ABC)

The ABC is optimized by AdamW with a learning rate of 0.00003. We train the ACG for 500 epochs using a batch size of 4096. We set the number of clusters in ABC to 21, and we found that using a single head of attention followed by MLP gives the best results (see Table 2). We also determined that the best lower and upper bounds for thresholding are $\frac{N}{k^2}$ and $\frac{N}{\sqrt{k}}$, respectively (see Table 3).

Optimal Transport clustering We ran OTC for 200 update iterations, and in each iteration, we executed the Sinkhorn algorithm with parameters $temp = 0.02$ and $n_{iter} = 10$. We also experemented with normalization of the input before the runing the algorithm and different similary functions for KMean algorithms (see Table 5).

4.2.2 Final Results

See Tables 4 and 5. In our experiments, Attention-Based Clustering showed the best results as the dataset-level clustering for USS.

5 Conclusion

In this project, we implemented and compared various methods for addressing the Unsupervised Semantic Segmentation problem. For image-level clustering, our investigation included the evaluation of Adaptive Concept Generator and Deep Spectral Methods. For dataset-level clustering, we explored methods such as Attention-Based Clustering, Optimal Transport Clustering, and KMeans. Notably, our results highlight the effectiveness of Attention-Based Clustering as a promising algorithm for clustering.

References

[1] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation., 2018. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4981–4990. 1

Method	mIoU
DSM	**25.95**
ACG (Our implementation)	21.7
DSM: Masked attention for DINO CLS	23.66

Table 1: DSM VS ACG on PASCAL VOC 2012

Architecture	mIoU
Attn + MLP	**30.42**
2xAttn + MLP	25.64
Identical	13.47
MLP	26.38

Table 2: Experiments with ABC architecture

Min Cluster Size	Max Cluster Size	mIoU
$\frac{N}{k^{1.5}}$	None	25.1895
$\frac{N}{k^2}$	None	26.72
$\frac{N}{k^2}$	$\frac{N}{k^{0.7}}$	27.11
$\frac{N}{k^2}$	$\frac{N}{k^{0.5}}$	**30.42**

Table 3: Experiments with the upper and lower cluster size thresholds.

Method	mIoU
KMeans	25.56
OTC	25.25
ABC	**30.42**

Table 4: Dataset-level Clustering Results on PASCAL VOC 2012 in USS

Method	mIoU
KMeans with L2 distance	**25.56**
KMeans with cos similarity	25.48
KMeans norm/std input and and L2 distance	24.09
KMeans norm/std input and cos similarity	24.22
OTC	25.25
OTC norm/std	23.69

Table 5: OTC and KMean experements

[2] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems, 26, 2013. 6

[3] Jiaya Jia Kaiming He Di Lin, Jifeng Dai and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation., 2016. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3159–3167, 2016. 1

[4] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. Pattern Anal. Stat. Model. Comput. Learn., Tech. Rep, 2007:1–45, 2012. 2

[5] Evan Shelhamer Jonathan Long and Trevor Darrell. Fully convolutional networks for semantic segmentation., 2015. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3431–3440. 1

[6] Jianbo Shi Maxwell D Collins Tien-Ju Yang Xiao Zhang Jyh-Jing Hwang, Stella X Yu and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 7334–7344, 2019. 1, 2. 1

[7] Zesen Cheng Runyi Yu Yian Zhao Guoli Song-Chang Liu Li Yuan Jie Chen. Kehan Li, Zhennan Wang. Acseg: Adaptive conceptualization for unsupervised semantic segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 1, 2, 3, 6

[8] Iro Laina Luke Melas-Kyriazi, Christian Rupprecht and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8364– 8375, 2022. 1, 2, 4, 6

[9] Bharath Hariharan Noah Snavely Mark Hamilton, Zhoutong Zhang and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In International Conference on Learning Representations, 2021. 2

[10] Ishan Misra Herve J ´ egou ´ Julien Mairal Piotr Bojanowski Mathilde Caron, Hugo Touvron and Armand Joulin. Emerging properties in self-supervised vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9650–9660, 2021. 1, 2, 3, 6

[11] Ishan Misra Herve J ´ egou ´ Julien Mairal Piotr Bojanowski Mathilde Caron, Hugo Touvron and Armand Joulin. Emerging properties in self-supervised vision transformers. Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9650–9660, 2021. 2, 3, 6

[12] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. Physical Review E, 69(2):026113, 2004. 3

[13] Xuehui Wang Huayu Wang Jiazhong Cen Dongsheng Jiang Lingxi Xie Xiaokang Yang Wei Shen, Zelin Peng and Qi Tian. A survey on label-efficient deep segmentation: Bridging the gap between weak supervision and dense prediction., 2022. arXiv preprint arXiv:2207.01223. 1

[14] Joao F Henriques Xu Ji and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9865–9874, 2019. 1

[15] Beomjun Kim Youngmin Oh and Bumsub Ham. Background-aware pooling and noise-aware loss for weakly-supervised semantic segmentation., 2021. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6913–6922. 1

[16] Meina Kan Shiguang Shan Yude Wang, Jie Zhang and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation., 2020. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12275–12284,. 1

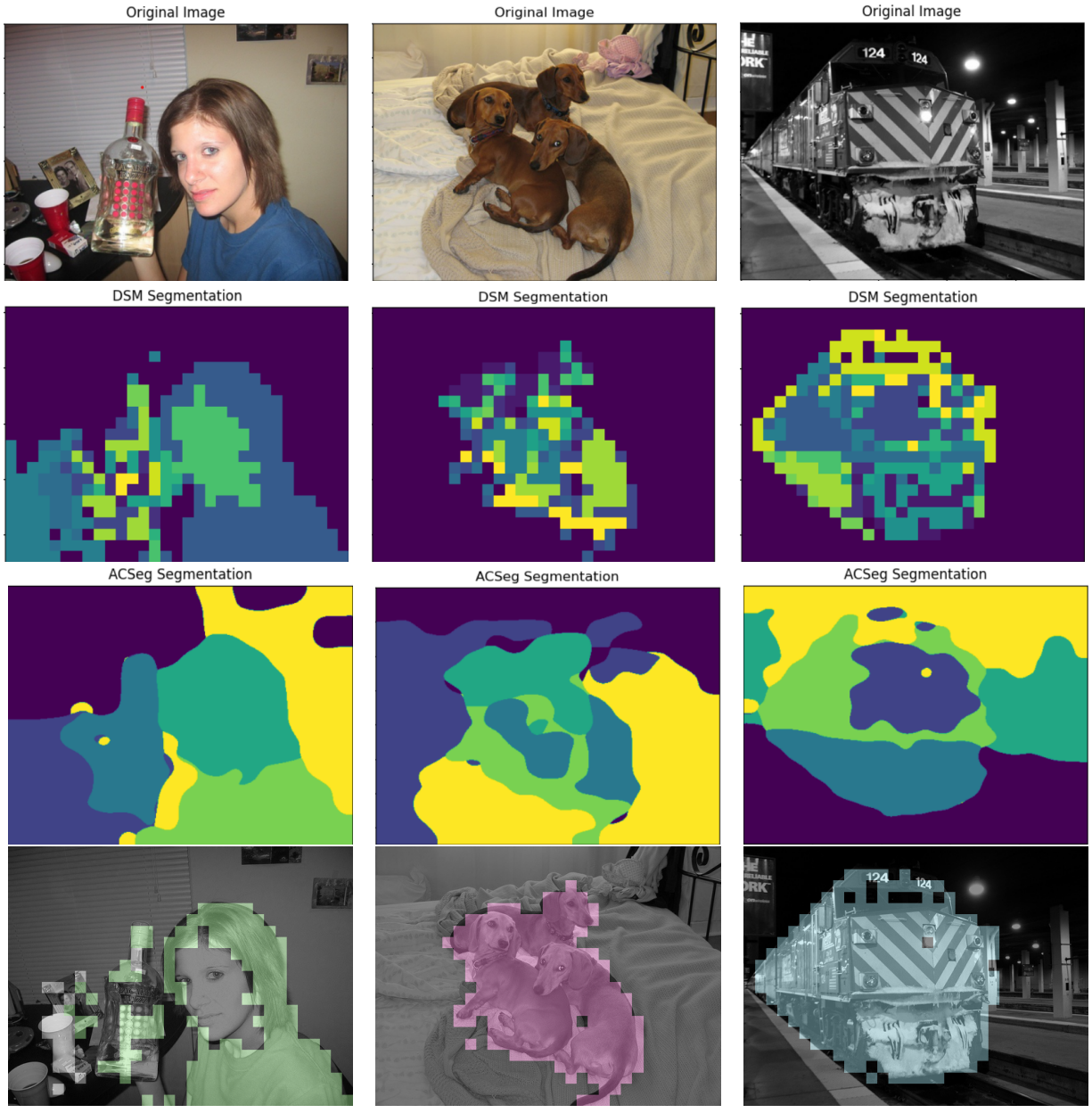

Figure 5: Visual examples of DSM, ACG and ABC results.