Zero-Shot Semantic Segmentation Using Pretrained DINO and Clustering

Abstract

This report describes a zero-shot semantic segmentation pipeline that leverages a pretrained DINO vision transformer, PCA-based dimensionality reduction, and fixed K-means clustering to generate pixel-level segmentation masks. The method requires no scene-specific training and produces visualizations of the segmentation results. In addition, the report discusses the advantages over traditional segmentation methods, current limitations of the pipeline, and provides visual comparisons.

1 Introduction

Semantic segmentation aims to assign a class label to every pixel in an image. Recent advances in self-supervised models, such as DINO [1], enable the extraction of semantically rich features without additional training. This report details a pipeline that utilizes these pretrained features, reduces their dimensionality using PCA, and clusters them using K-means to obtain segmentation masks.

2 Methodology

2.1 Preprocessing and Feature Extraction

The input image is first resized to 224×224 pixels and normalized using ImageNet statistics. A pretrained DINO model is then used to extract patch embeddings from the image. The CLS token is excluded, and the remaining patch features are assumed to form a square grid.

2.2 Dimensionality Reduction via PCA

The extracted features are high-dimensional. To facilitate clustering, PCA is applied to reduce the feature dimension to 3 components. The reduced features are then reshaped into a spatial grid corresponding to the original patch layout.

2.3 K-means Clustering

K-means clustering is applied to the flattened PCA-reduced features with a fixed number of clusters (default 5) and 10 initializations (`n_init=10`). The resulting cluster labels are reshaped back into the spatial grid.

2.4 Mask Generation and Visualization

The clustered label map is upsampled to the original image size (224×224) using nearest-neighbor interpolation to preserve discrete boundaries. The final segmentation mask is saved and visualized alongside the original image.

```
# Feature Extraction (excluding CLS token)
features = model.get_intermediate_layers(image_tensor, n=1)[0]
features = features[0, 1:, :].cpu().numpy()

# PCA Dimensionality Reduction
pca = PCA(n_components=3)
reduced_features = pca.fit_transform(features)
reduced_features = reduced_features.reshape(h, w, -1)

# K-means Clustering
flattened_features = reduced_features.reshape(-1, 3)
kmeans = KMeans(n_clusters=5, n_init=10, random_state=0).fit(flattened_features)
labels = kmeans.labels_.reshape(h, w)

# Mask Upsampling
labels_tensor = torch.tensor(labels).unsqueeze(0).unsqueeze(0).float()
mask = F.interpolate(labels_tensor, size=(224, 224), mode='nearest').squeeze().
    numpy()
```

Listing 1: Key Segmentation Pipeline Code

3 Advantages Over Traditional Methods

Traditional semantic segmentation approaches often rely on:

- **Thresholding/Edge Detection:** These methods are sensitive to variations in illumination and texture, and struggle with complex or overlapping objects.
- **Supervised CNN-Based Segmentation:** Methods like Mask R-CNN [2] require large annotated datasets and extensive training, and may not generalize well to unseen scenes.
- **Basic Clustering in Color Space:** Clustering solely on RGB or HSV values lacks semantic understanding, leading to poor performance on complex images.

Our zero-shot pipeline leverages the semantically rich features extracted by the pretrained DINO model [1]. PCA reduces the feature dimensionality to make clustering more efficient and robust. This method avoids the need for scene-specific training and extensive annotated data, making it an attractive alternative for rapid segmentation tasks.

4 Current Limitations of the Pipeline

Despite its advantages, the presented pipeline has some limitations:

- **Fixed Number of Clusters:** The number of clusters is predefined (default 5) and may not adapt optimally to all image contents.
- **No Spatial Regularization:** Clustering is performed on features independently, which can lead to spatially inconsistent segmentation in some cases.

- **Limited Dimensionality Reduction:** PCA is a linear method and might not capture all non-linear relationships in the feature space.
- **Model Dependency:** The segmentation quality relies heavily on the quality and generality of the pretrained DINO model.

5 Results and Visual Comparisons

The segmentation pipeline outputs two key artifacts:

- A segmentation mask saved as a color-coded image.
- A side-by-side visualization comparing the original image with the segmentation mask (often combined into one figure).

These outputs are stored in the `masks` and `visualizations` directories, respectively.

5.1 Comparison Figures

Figure 1 shows six combined images. Each image (1.PNG, 2.PNG, 3.PNG, 4.PNG, 5.PNG, 6.PNG) already includes both the original scene on the left and its corresponding segmentation mask on the right.

6 Conclusion

The presented pipeline demonstrates a simple, zero-shot approach to semantic segmentation using a pretrained DINO model. By combining PCA for dimensionality reduction with fixed K-means clustering, the method efficiently produces segmentation masks without requiring additional training. This approach overcomes several limitations of traditional methods, such as the dependency on large annotated datasets and lack of semantic context in basic clustering methods.

7 Future Work

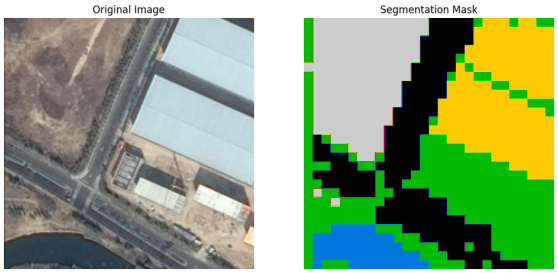
Future improvements could include:

- Developing adaptive methods for selecting the optimal number of clusters.
- Integrating spatial regularization to enhance the consistency of segmentation masks.
- Exploring nonlinear dimensionality reduction techniques, such as t-SNE or UMAP, to better capture complex feature relationships.

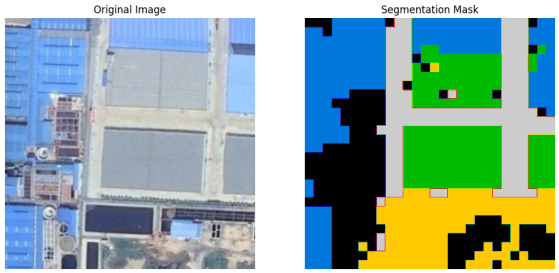
References

[1] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging Properties in Self-Supervised Vision Transformers. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

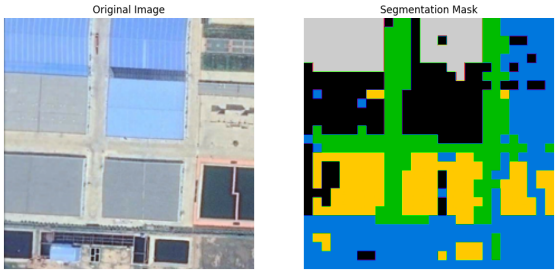
[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.



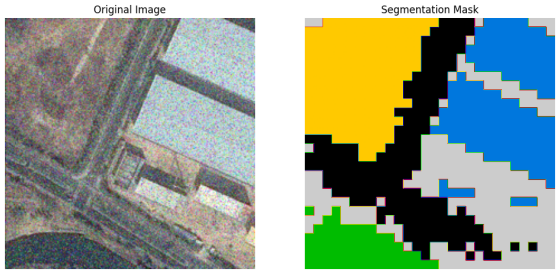
(a) Comparison 1



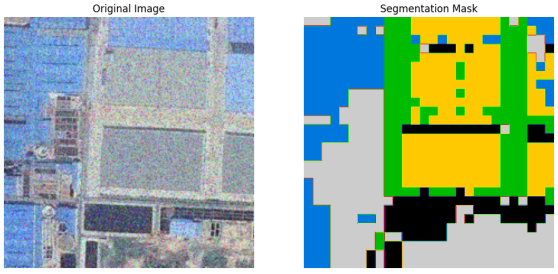
(b) Comparison 2



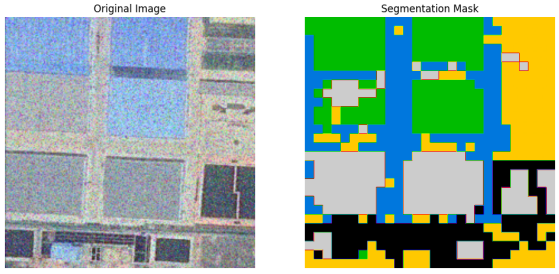
(c) Comparison 3



(d) Comparison 4



(e) Comparison 5



(f) Comparison 6

Figure 1: Six examples of combined images where each one shows the original image (left) and the corresponding segmentation mask (right).