Semantic Segmentation using CLIP Model

Ahmad Rammal
Vladislav Tominin
Iaroslav Tominin

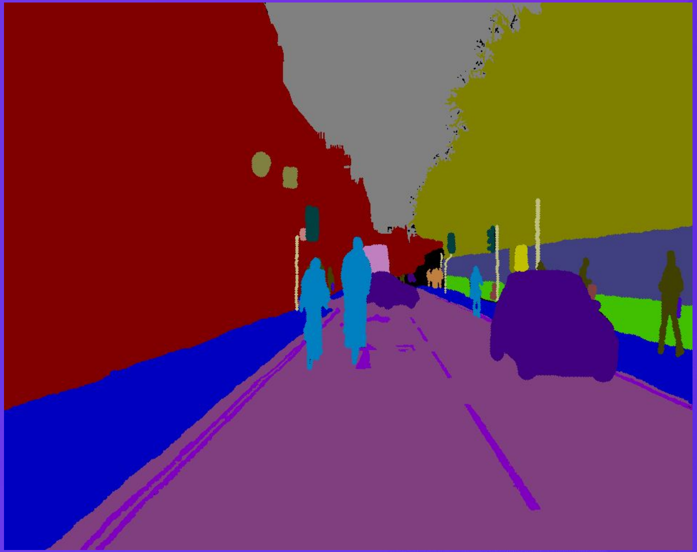
Table of Content

Table of Content

Semantic Segmentation Setting

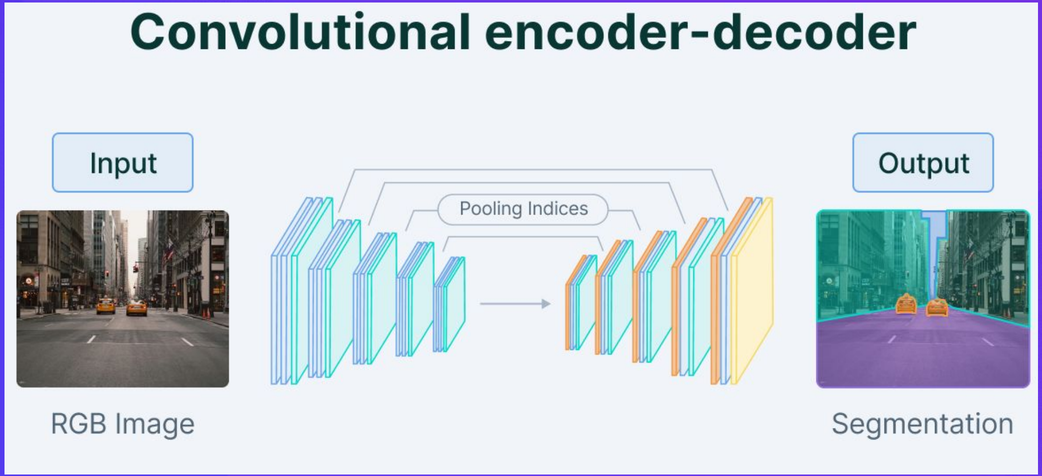
What is Semantic Segmentation?

Goal: Divide an image into segments, and assigning a label to each segment



Common Approach

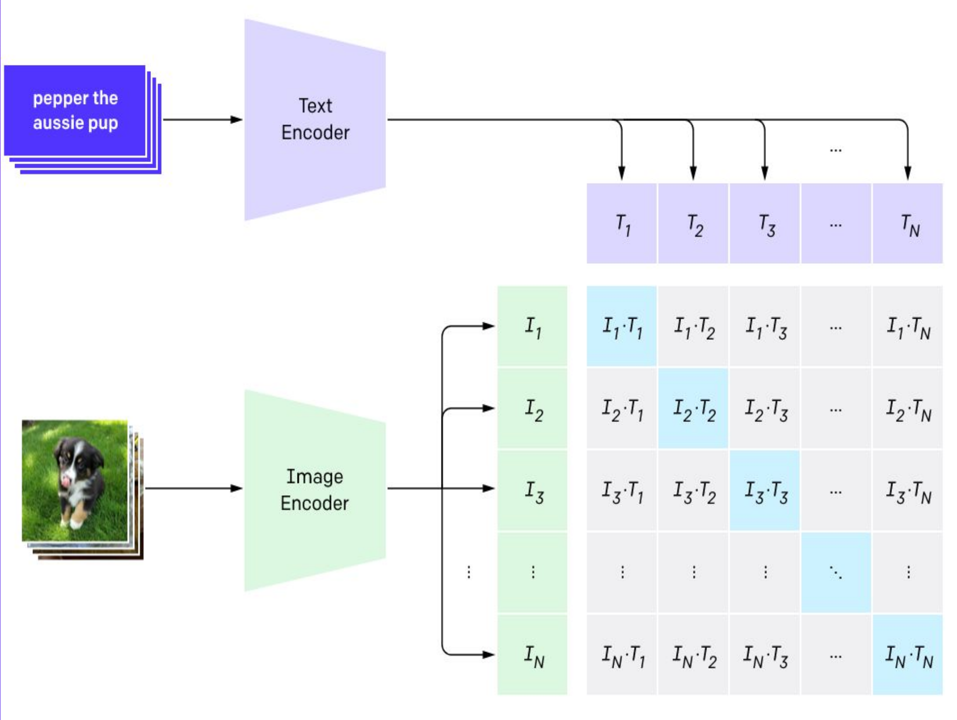
Semantic segmentation is typically achieved labeling each pixel in an image with a semantic class. A CNN is trained on a large dataset of labeled images.



CLIP Usage

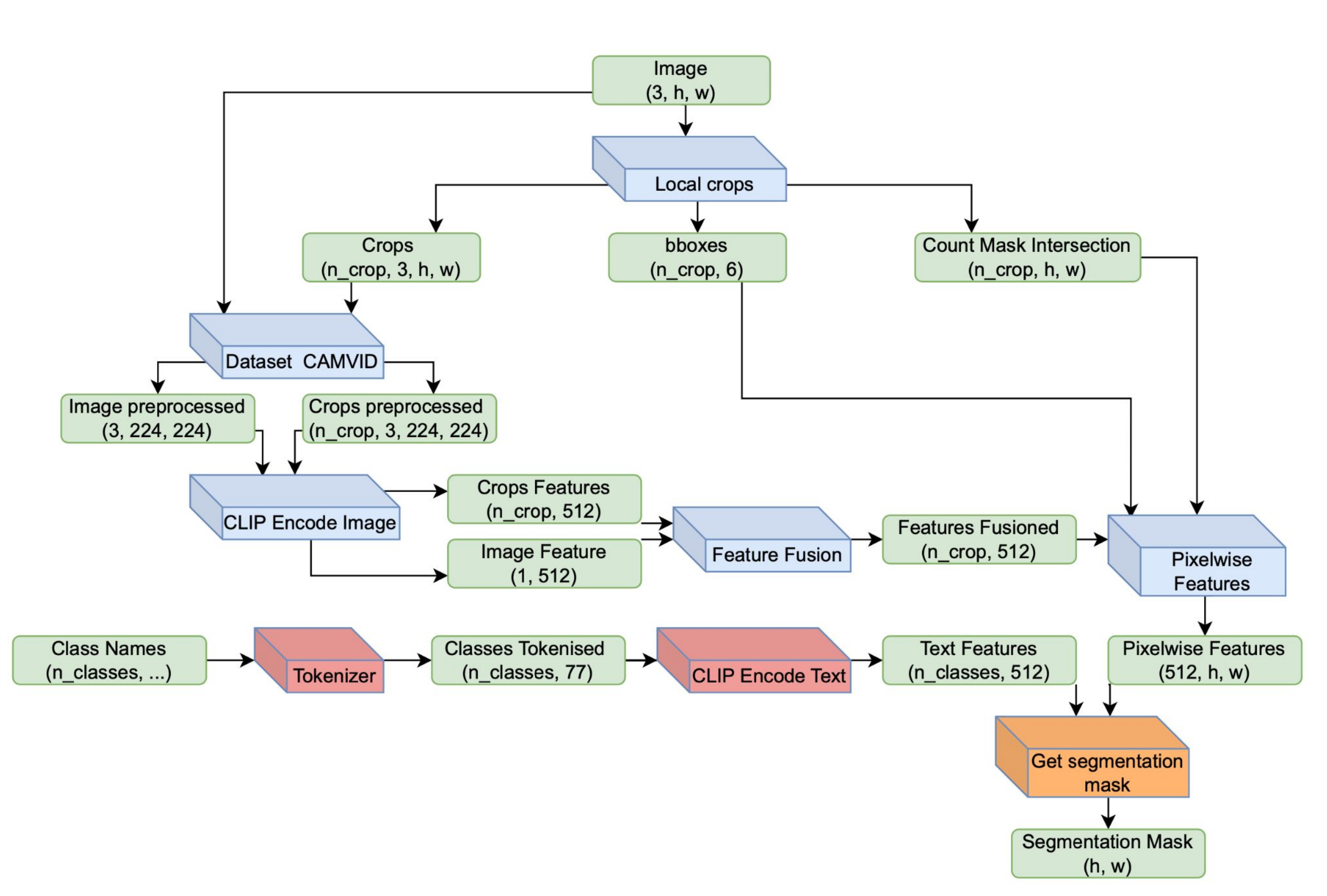
CLIP can be used as a feature extractor that can encode both the image and text inputs into a common embedding space

Further Fine-tuning is required



Semantic Segmentation Without Fine-Tuning

Model Architecture

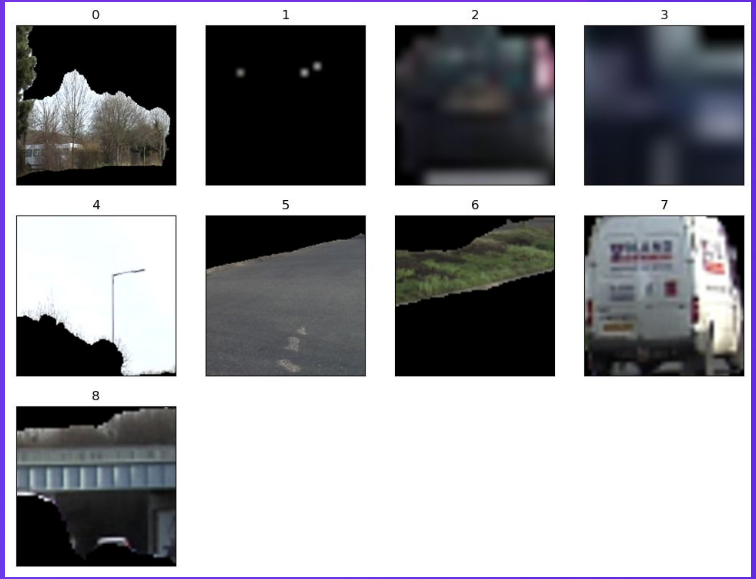


Generating Crops: Instance Segmentation

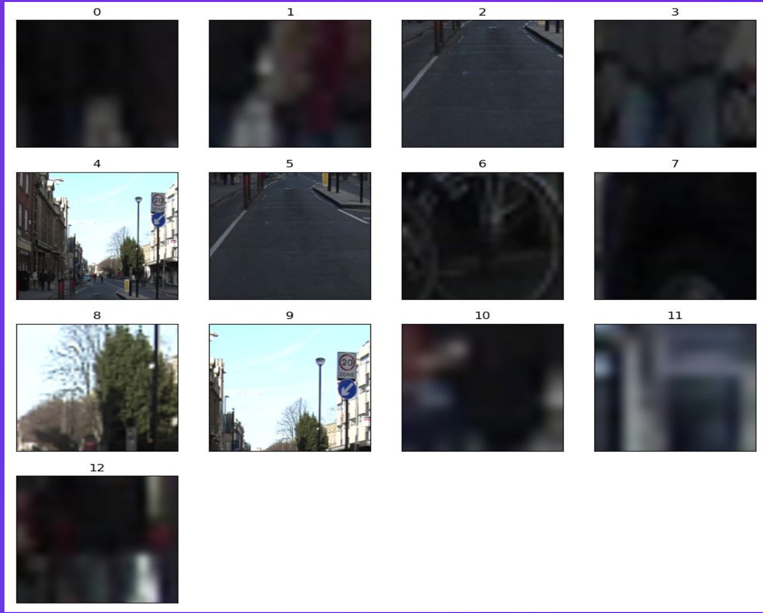
Mask R-CNN vs Mask2former

Mask2former has outperform Mask R-CNN:

Better Quality + Mask2former crops cover all the cropped photo.

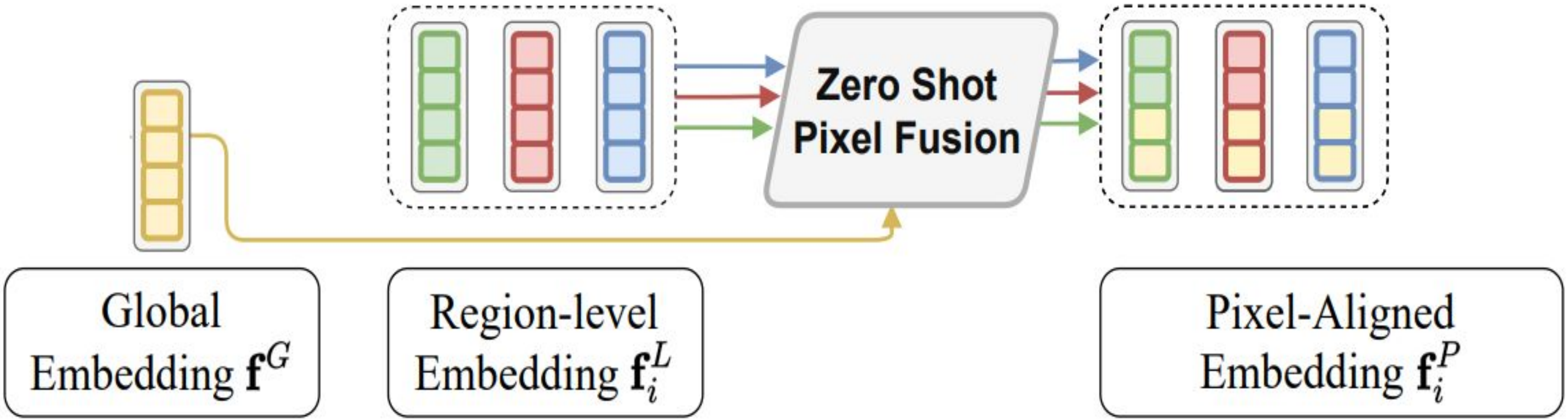


Unmasked Crops



Feature Fusion: Concept-Fusion Paper

Pixel Fusion Overview



Local and global embeddings

The bounding boxes are fed through the model $\mathbb{F}$ to obtain local embeddings $f_i^L = \mathbb{F}(b_i)$

The global embedding $f^G = \mathbb{F}(X)$ is simply the embedding of the entire image.

We compute the cosine similarity between the local features and the global feature:

$$\phi_i = \frac{\mathbf{f}^{L}{}_{i} \cdot \mathbf{f}^{G}}{\|\mathbf{f}^{L}{}_{i}\| \, \|\mathbf{f}^{G}\| + \epsilon}$$

We compute the matrix of cosine similarities between all pairs of local embeddings:

$$\varphi ij = \langle f_i^L, f_j^L \rangle : \forall i, j$$

For each local embedding f_i^L, we compute its average similarity to all other local embeddings:

$$\overline{\varphi}i = \frac{1}{R} \sum_{j=1, j \neq i}^{R} \varphi_{ij}$$

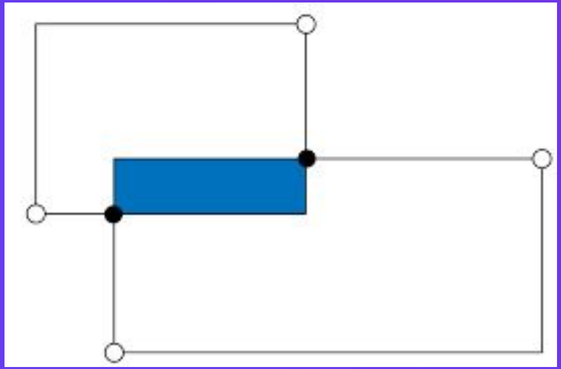
We combine the two similarities above to compute the mixing weight, with $\tau = 1$:

$$w_i = \frac{\exp\left(\frac{\phi_i + \overline{\varphi}i}{\tau}\right)}{\sum_{i=1}^{R} \exp\left(\frac{\phi_i + \overline{\varphi}_i}{\tau}\right)}$$

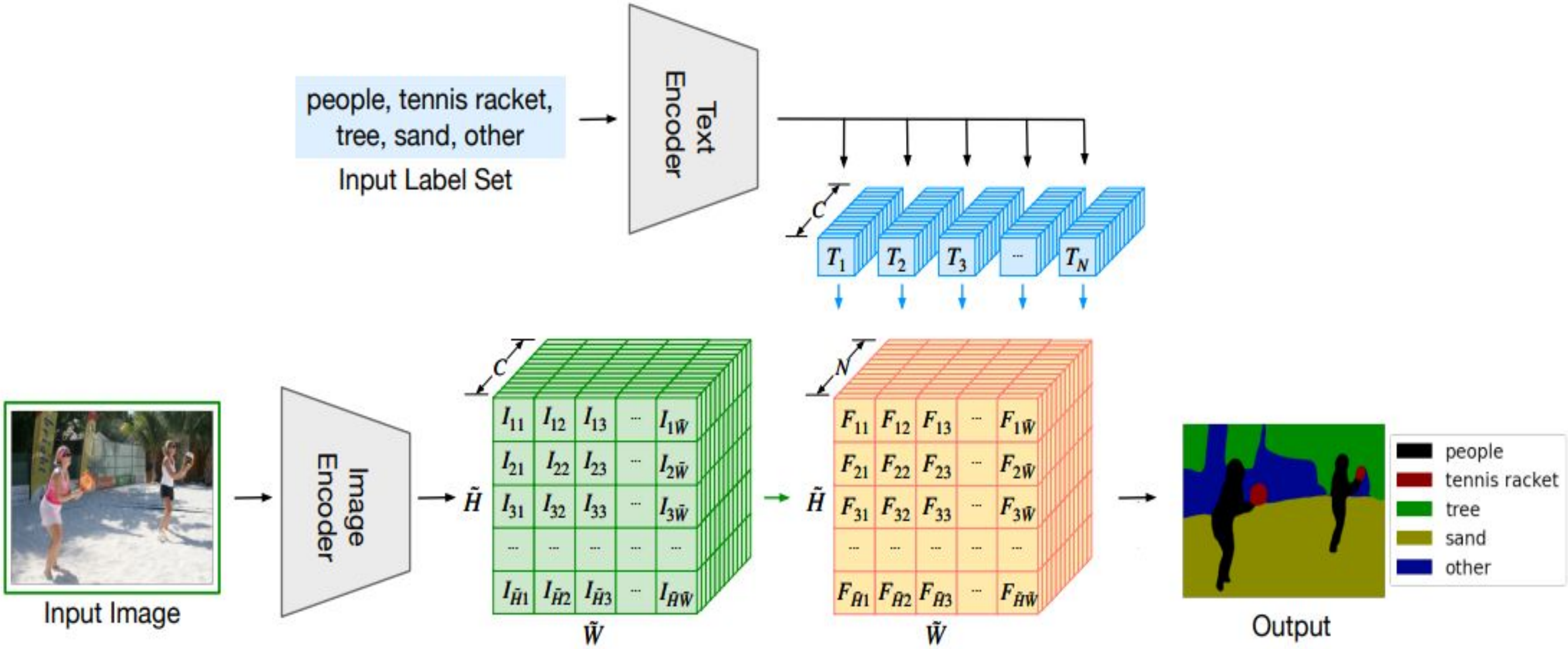
Finally, the pixel-aligned feature for each crop i is:

$$f_i^P = w_i f^G + (1 - w_i) f_i^L$$

The crop’s embedding is assigned to each pixel. If a pixel is in multiple crops we simply take the average embedding.



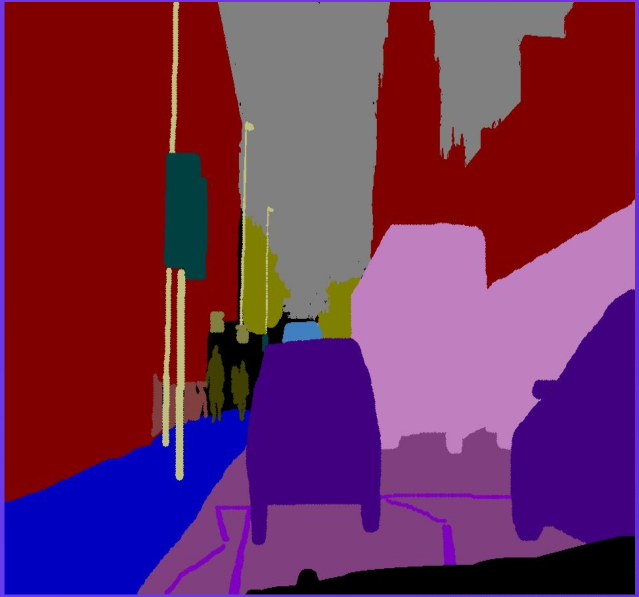
Semantic Mask


people, tennis racket, tree, sand, other
Input Label Set
Text Encoder
C
T_1 T_2 T_3 ... T_N
Input Image
Image Encoder
$\tilde{H}$
C
I_{11} I_{12} I_{13} ... $I_{1\tilde{W}}$
I_{21} I_{22} I_{23} ... $I_{2\tilde{W}}$
I_{31} I_{32} I_{33} ... $I_{3\tilde{W}}$
$I_{\tilde{H}1}$ $I_{\tilde{H}2}$ $I_{\tilde{H}3}$ ... $I_{\tilde{H}\tilde{W}}$
$\tilde{W}$
$\tilde{H}$
N
F_{11} F_{12} F_{13} ... $F_{1\tilde{W}}$
F_{21} F_{22} F_{23} ... $F_{2\tilde{W}}$
F_{31} F_{32} F_{33} ... $F_{3\tilde{W}}$
$F_{\tilde{H}1}$ $F_{\tilde{H}2}$ $F_{\tilde{H}3}$ ... $F_{\tilde{H}\tilde{W}}$
$\tilde{W}$
Output
people
tennis racket
tree
sand
other

Dataset: CamVid

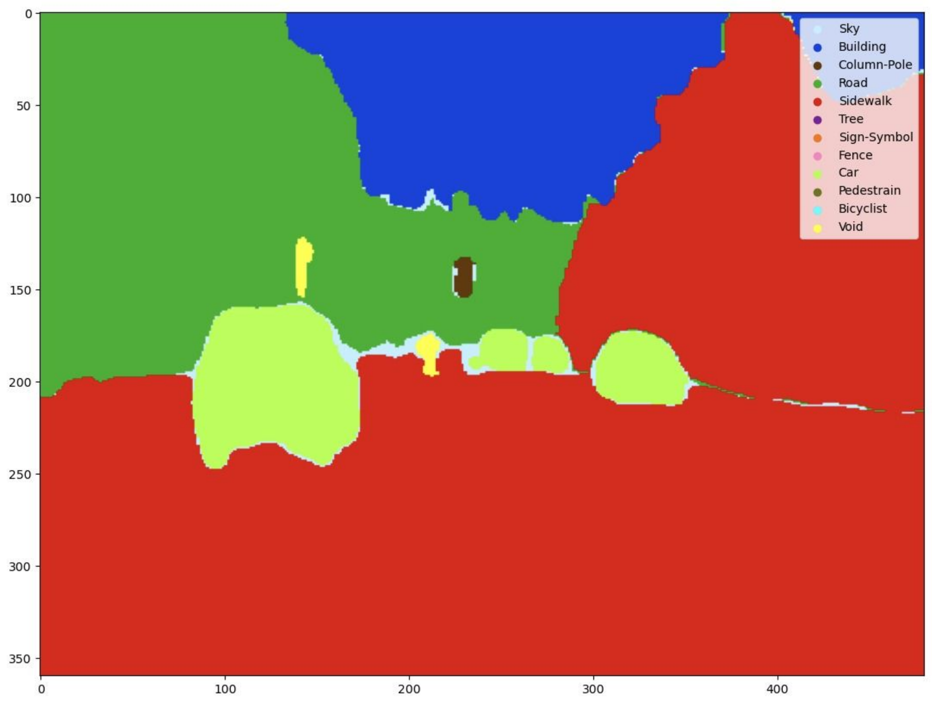
Dataset CamVid

367 images with annotated masks

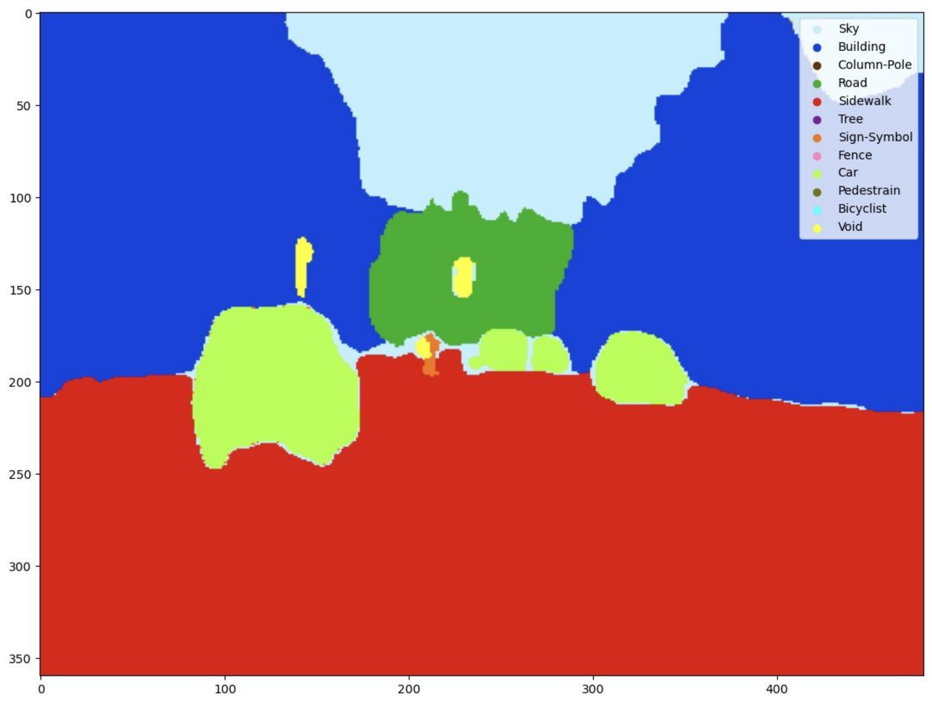


Results

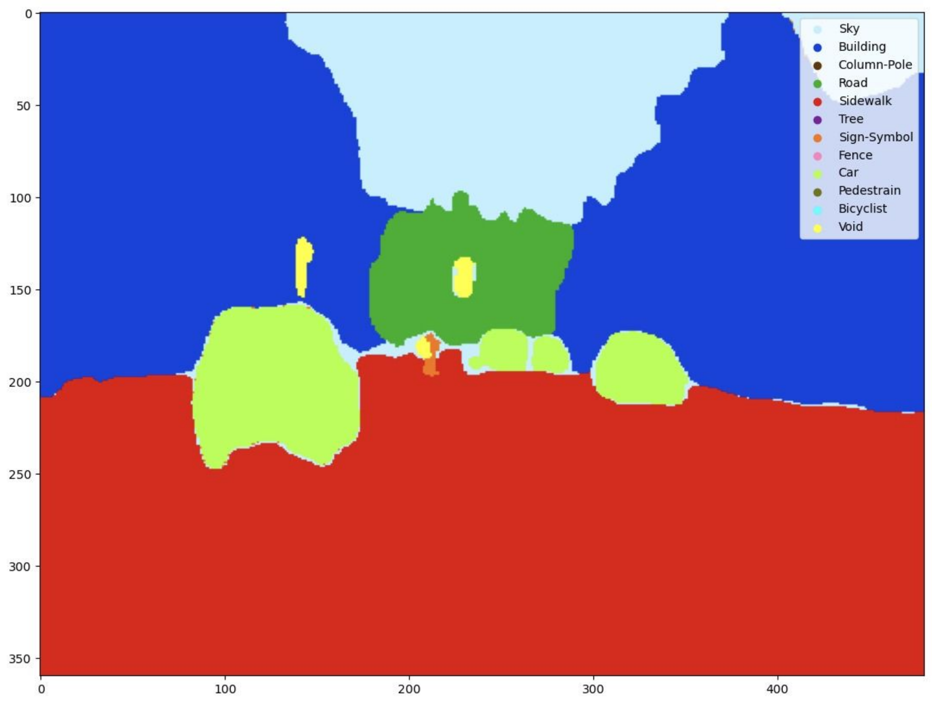
Qualitative Results: Unmasked + FF



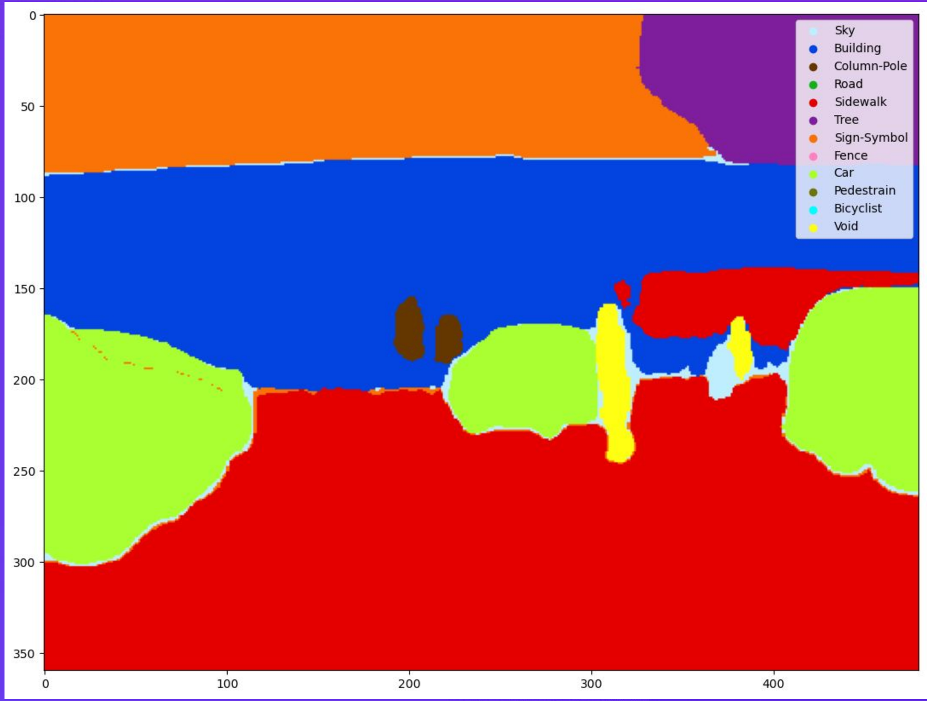
Qualitative Results: Masked + FF



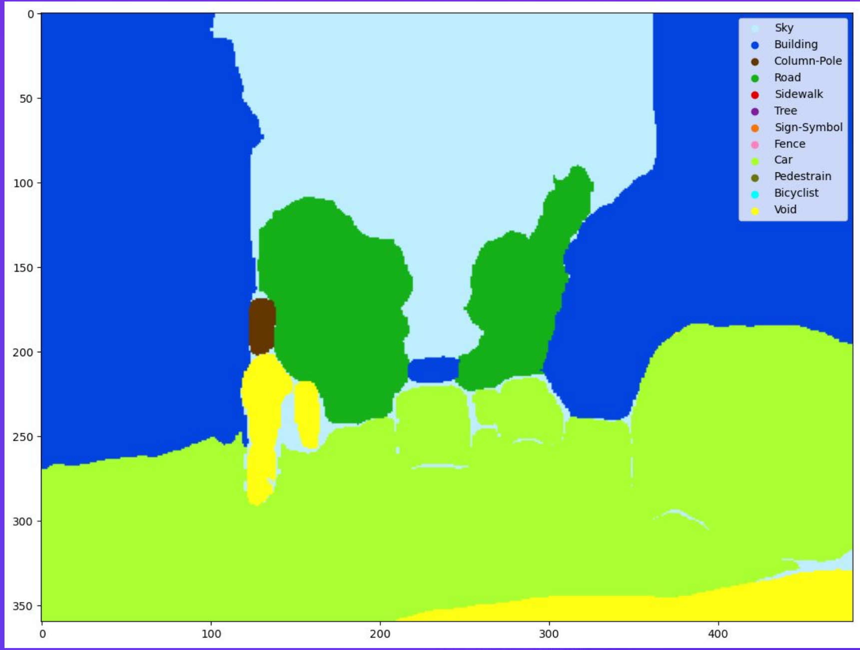
Qualitative Results: Masked - FF



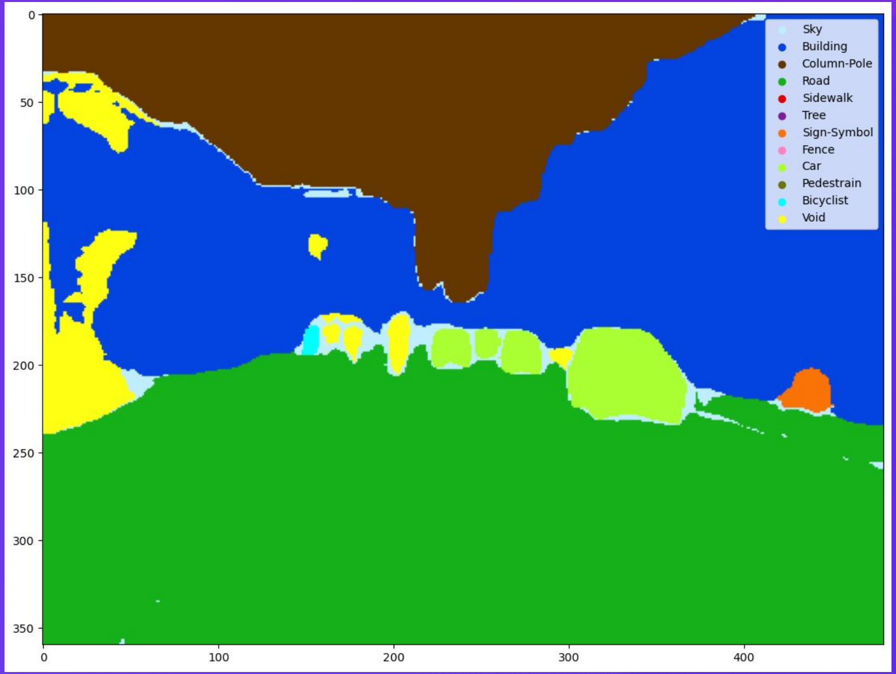
Qualitative Results: Masked + FF



Qualitative Results: Masked + FF



Qualitative Results: Masked + FF



Class	F1-Score	Recall	Precision	IOU (ref = 0.64)
Sky	0.19	0.20	0.21	0.18
Building	0.47	0.50	0.45	0.40
Column-Pole	0.02	0.13	0.10	0.01
Road	0.42	0.49	0.38	0.37
Sidewalk	0.01	0.15	0.00	0.00
Tree	0.42	0.47	0.44	0.32
Sign-Symbol	0.00	0.00	0.00	0.00
Fence	0.00	0.00	0.00	0.00
Car	0.51	0.80	0.43	0.40
Void	0.15	0.17	0.19	0.11

Conclusion

Conclusion

- Competitive results for the segmentation of sufficiently large objects is obtained
- Approach does not require fine-tuning, inference can be done on the CPU
- Algorithm does not work well with small objects, it may become the topic of the following works
- If a labeled dataset is available, it is possible to fine-tune the CLIP to get better results.

Possible ways of development

- Modify Feature Embeddings
- Fixing problems with small objects
- Feature Fusion Utility Testing