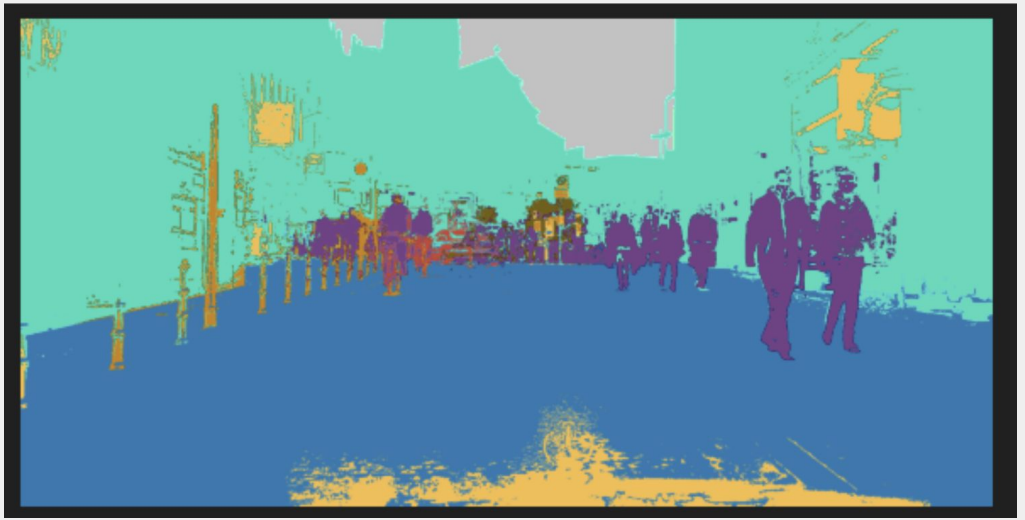
Semantic Segmentation

Pixelwise Labelling of Urban Environments



Project Presentation

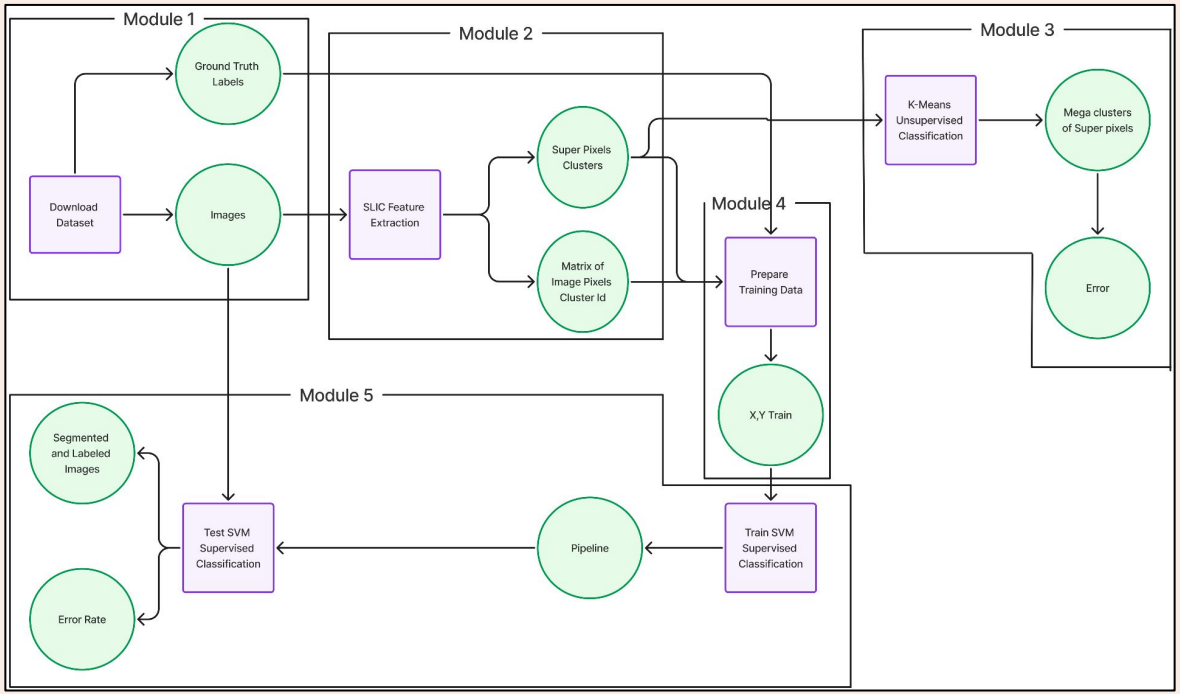
April 2025

Research Question

- The ability to detect different obstacles within an urban environment

- Label each obstacle to allow potential autonomous driving systems the ability to navigate urban environments

System Architecture



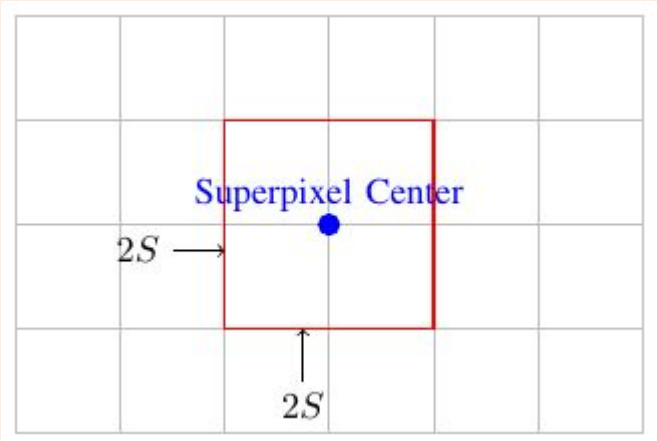
Algorithm

- Images are first segmented with SLIC to get super pixels
- Superpixels are sent through a SVM classifier to classify them to into classes

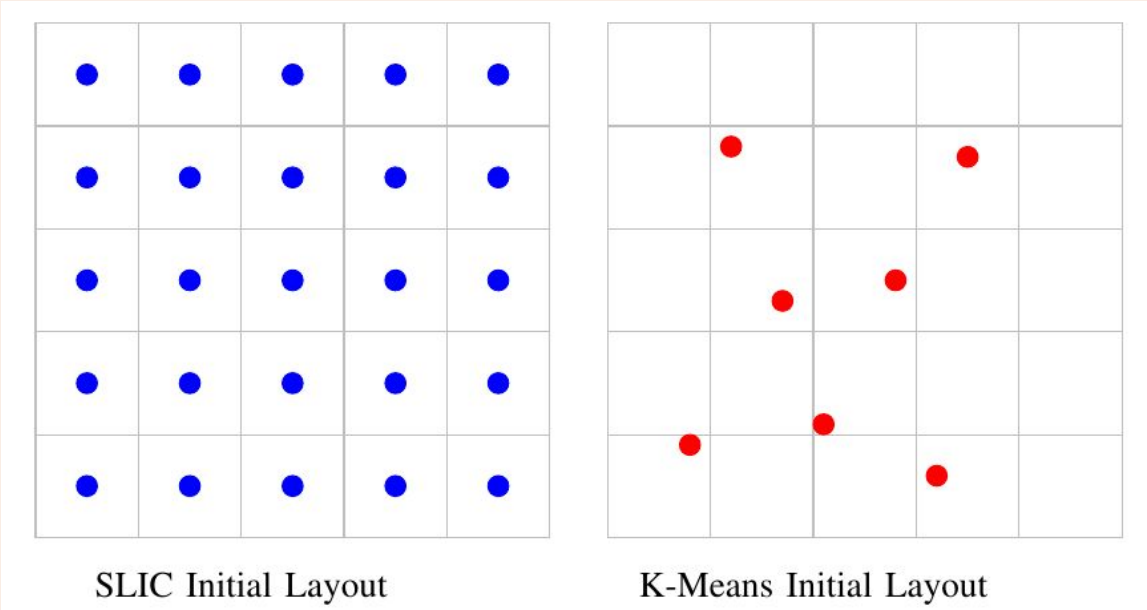
SLIC

- Modified version of K-Means
- Modifications:
  - Cluster initialization
  - Optimized Distance Calculation
  - Color data used to measure distance
  - GPU Acceleration

Distance Calculation



Cluster Initialization



SLIC Initial Layout

K-Means Initial Layout

Color Data

- CIELAB Color Space
  - Approximates Human Vision better than RGB
  - Better differentiation between similar colors
- “M” spatial compactness value

GPU Acceleration

Full SLIC implementation in C++/CUDA

200x performance compared to CPU

Support Vector Machine (SVM)

- SVMs are a class of machine learning models that classify data by mapping hyperplanes throughout the feature dimensions.
- Modifications:
  - Kernel Type
  - Regularization Parameters
  - Gamma Value

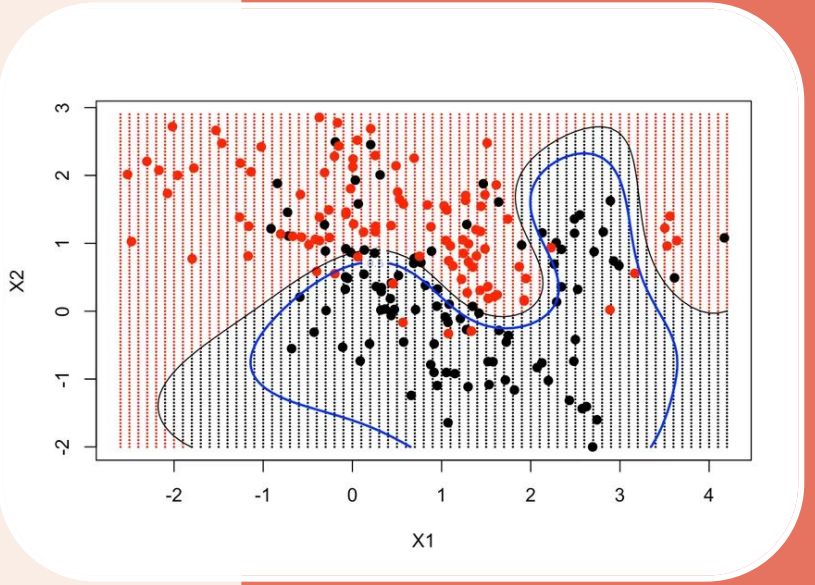


*Image sourced from "Support Vector Machines in R" from https://www.datacamp.com/tutorial/support-vector-machines-r

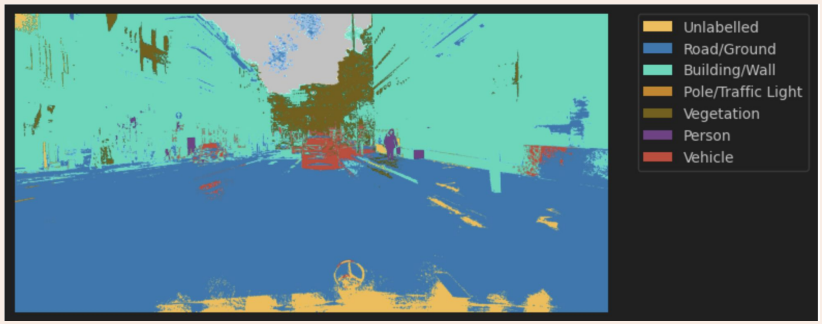
Parameter Tuning and Feature Selection

- SVM parameters were tuned using `sklearn.model_selection`, which optimized for various regularization strengths and kernel types.
- Features were selected through trial and error until the blend that offered the best efficiency and performance was found.

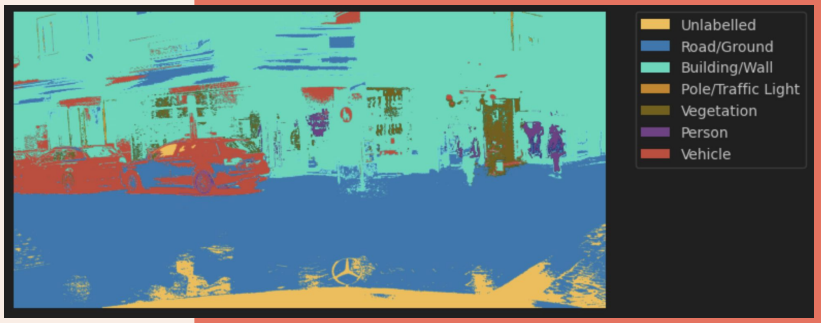
Sample SLIC Segmentation

SVM Sample Classifications

Average Outcome



Poor Outcome



K-Mean Sample Classification

6 Mega Clusters

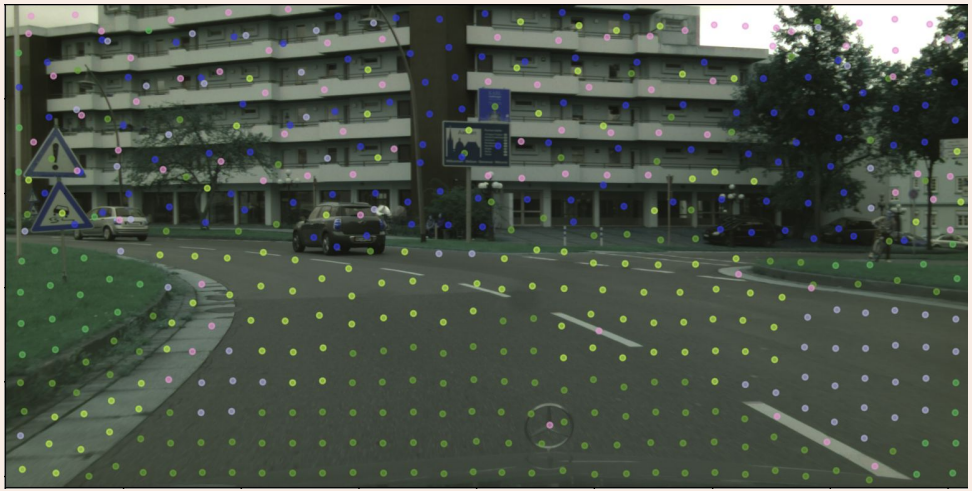


TABLE I: K-Means Accuracy for Varying Mega Clusters

Number of Mega Clusters	Image 1	Image 2	Image 3
2	23.18%	25.75%	29.48%
4	31.28%	29.18%	34.90%
8	34.01%	26.47%	33.32%
16	40.27%	30.47%	38.48%
32	41.96%	32.68%	37.97%
64	40.86%	31.09%	38.86%
128	33.59%	31.06%	30.88%

Results

With the architecture previously described we were able to achieve an accuracy of 81% over 100 images. The SVM was trained on only 5 images that had representation of all classes within them.

Live Demo

Q&A

Any questions?

https://github.com/SilverSix6/SemanticSegmentation