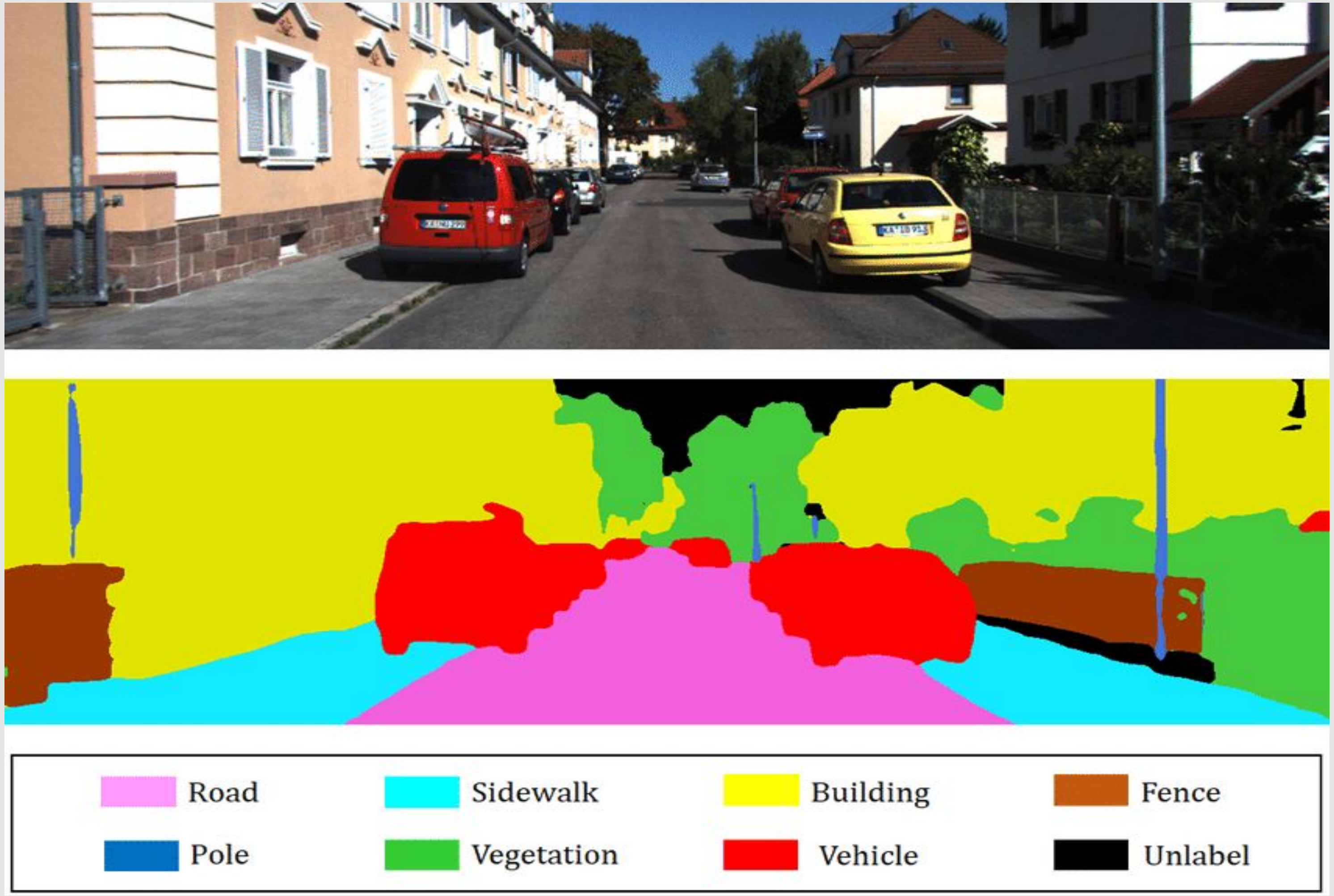
Coarse Semantic Segmentation with Multiscale Dimensionality Reduced Subblocks

By: Ben Kizaric, Jeff Hank, and Kevin Kim

Semantic Segmentation: Motivation

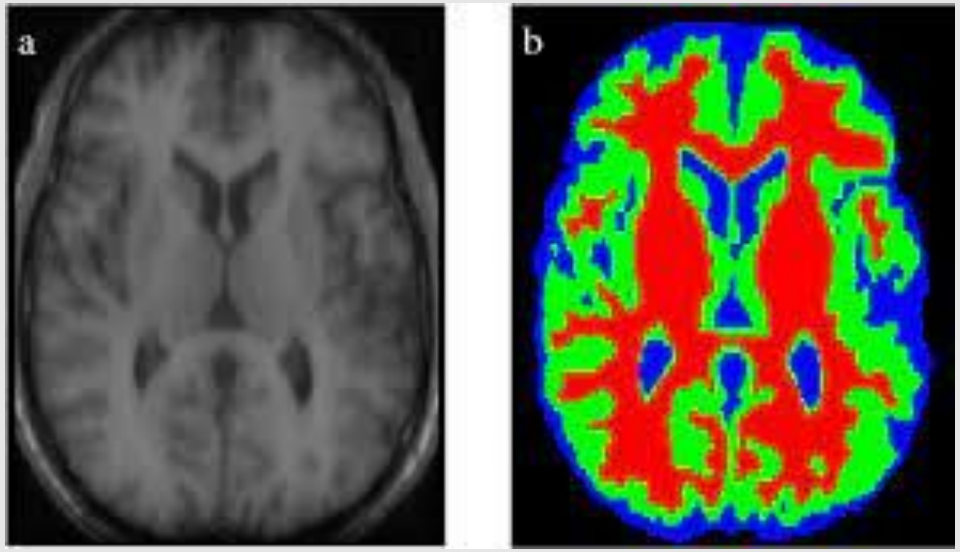
Task: Assign a semantic class to each pixel in an image



Semantic Segmentation: Motivation

Applications

- autonomous vehicles
- medical analysis
- specific classification tasks

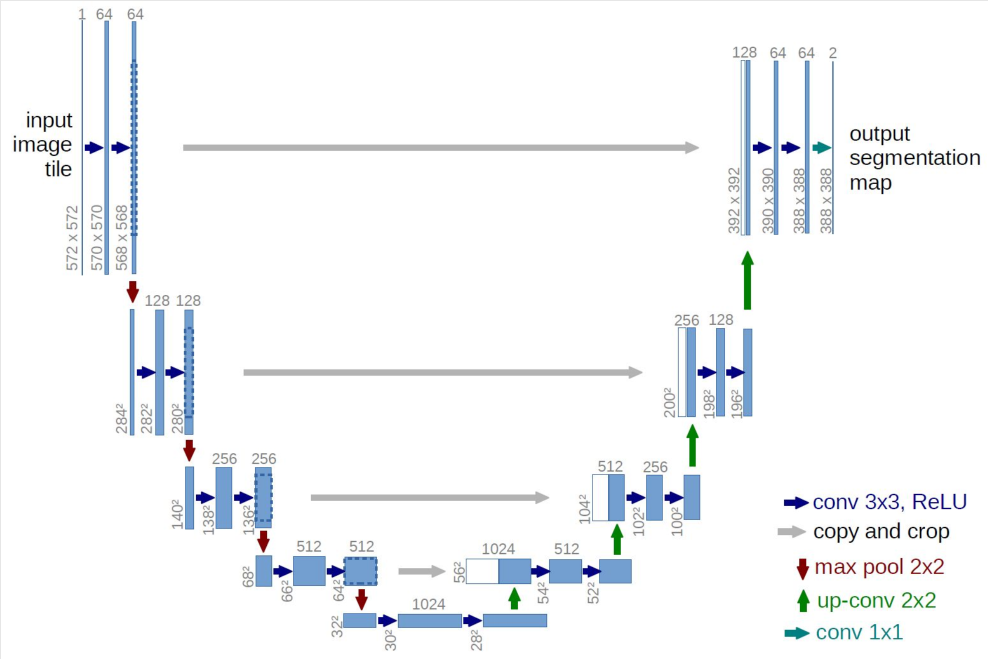


Current State of the Art: UNET

State of the art approach:

1) Encoder
   a) Classification network (convolution)
   b) Downsampling

2) Decoder
   a) Upsampling
   b) Concatenation



Our Approach: Motivation

Big Neural Networks & CNNs are Great!

BUT

They are huge, often in the Millions of parameters. Image data is very high-dimensional. Because of this:

- They have very long training time.
- They need lots of training data.
- Inference isn't super fast.

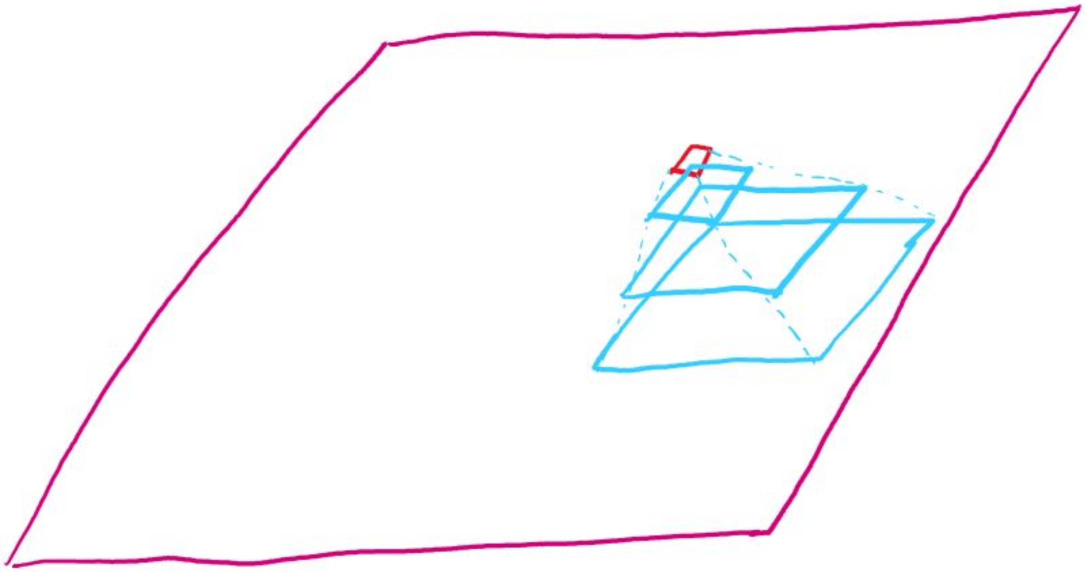
Our Approach: Goal

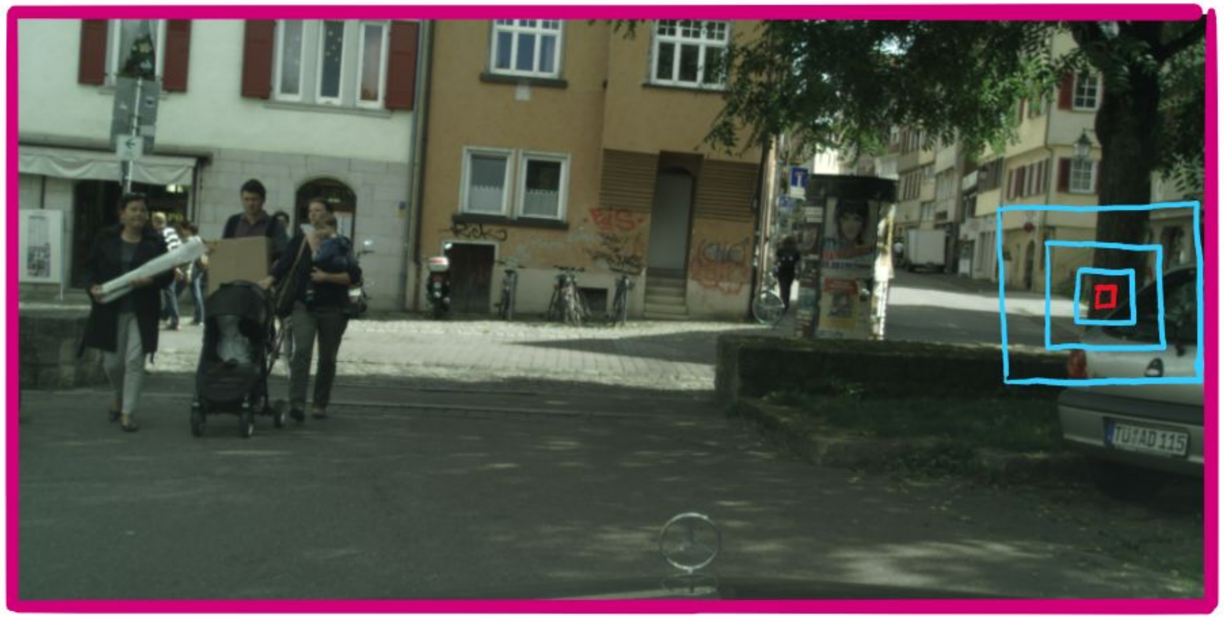
Create a smaller, simpler semantic segmentation algorithm that trains faster.

BY

Simplifying the semantic segmentation task.
Applying MASSIVE dimensionality reduction.

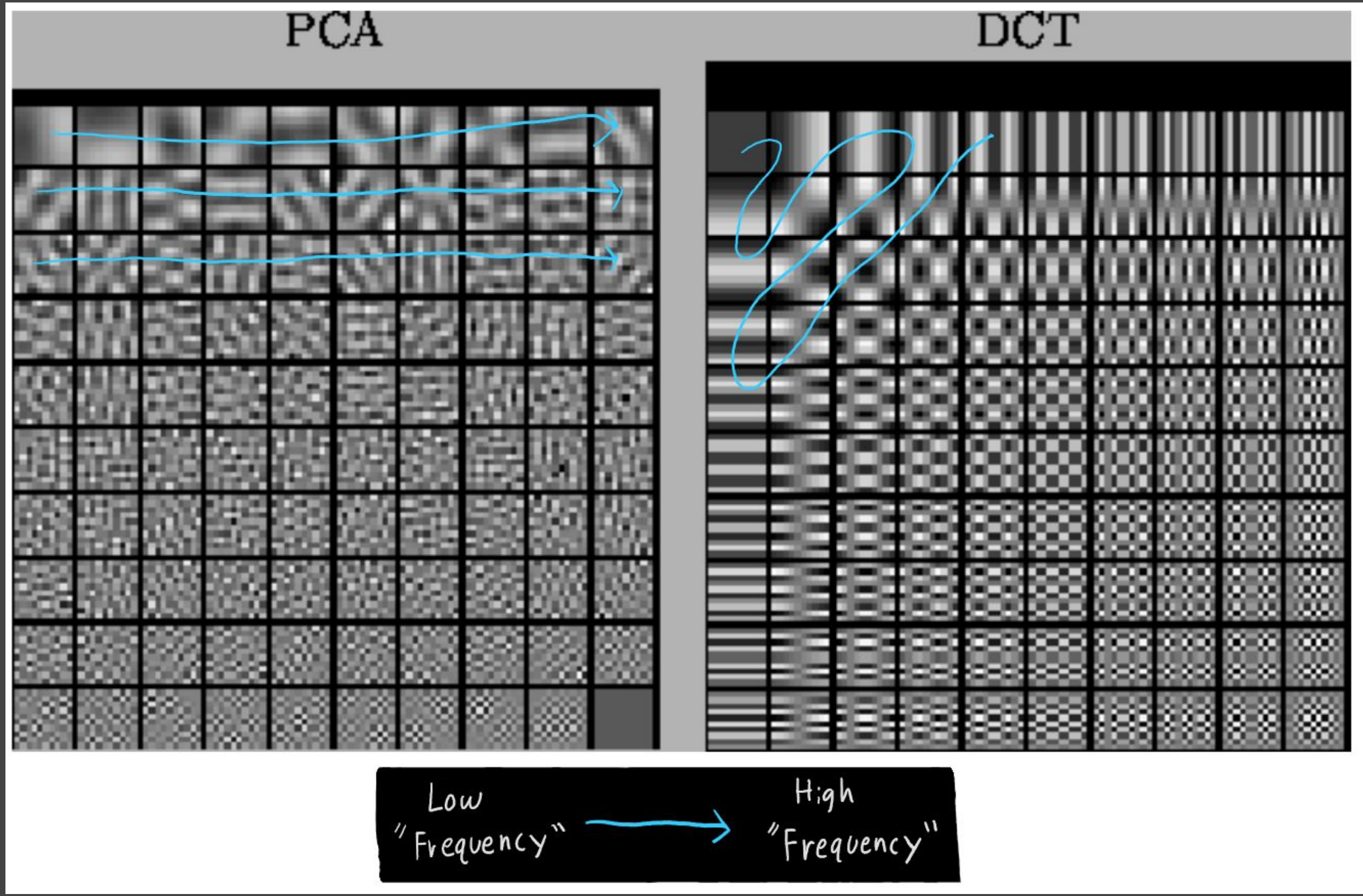
Our Approach: Multiscale Sliding Window





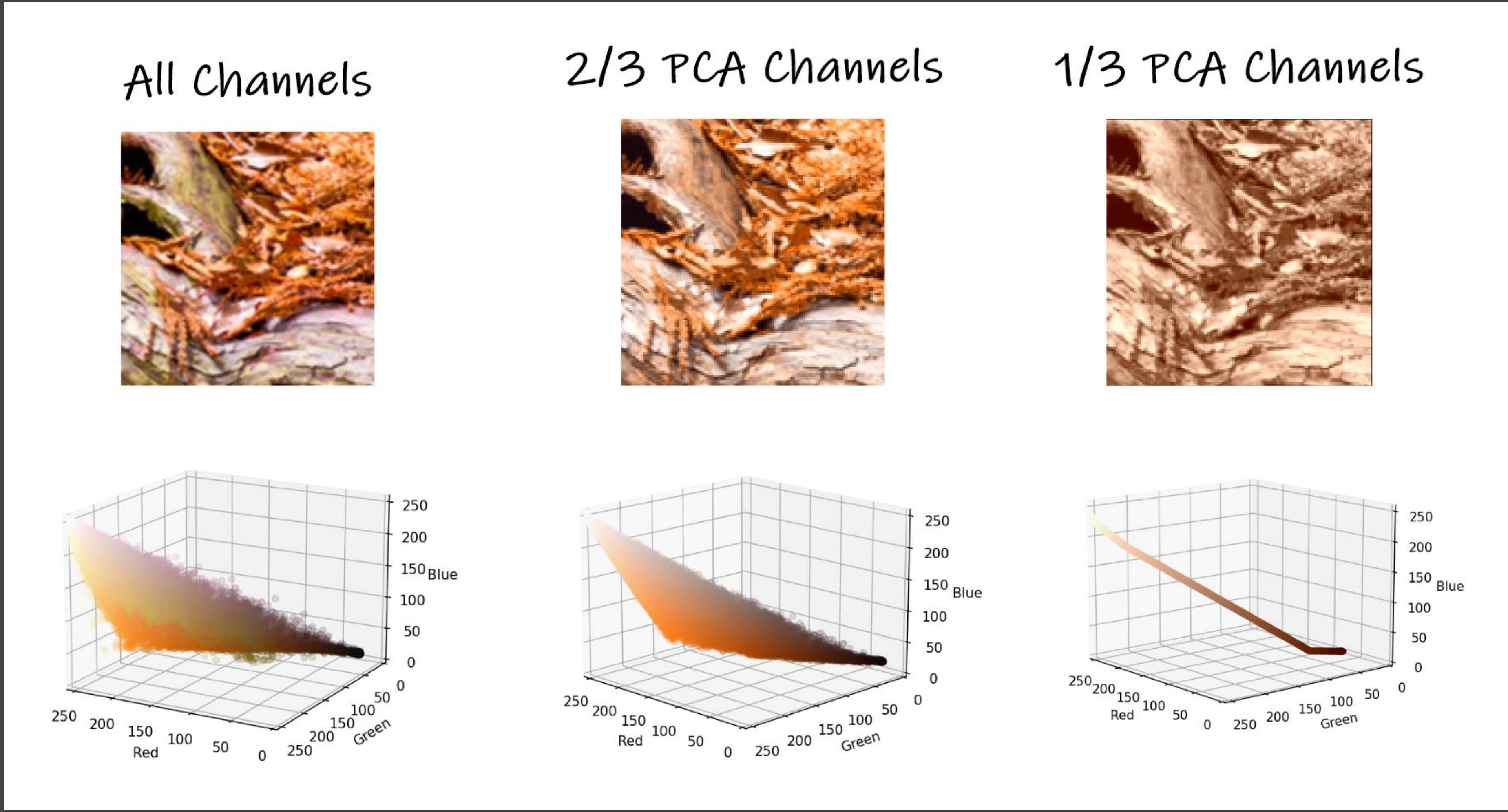
Our Approach: Multiscale Sliding Window

Our Approach: Texture Dimensionality Reduction using PCA



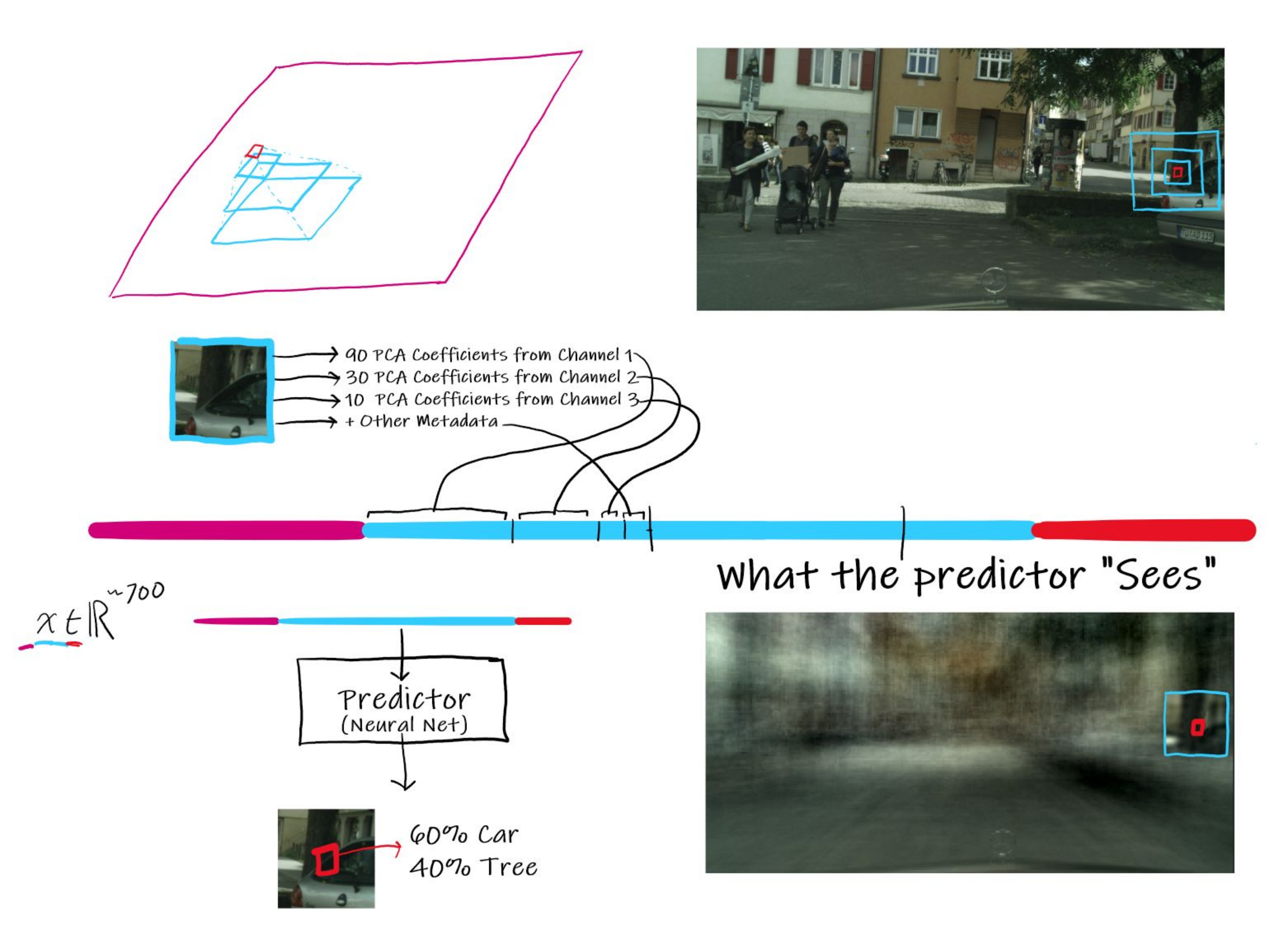
Use PCA Coefficients as input to a predictive model.

Our Approach: Color Dimensionality Reduction using PCA



Our Approach: Color Dimensionality Reduction using PCA

Our Approach: Encoding



Our Approach: Encoding

Our Approach: Prediction

NN Architecture

- 3 Dense layers
- Mean Squared Logarithmic Error loss function
- Adam optimizer with a learning rate of 0.0001
- Judged in terms of Accuracy metrics.
- Trained for 20 minutes.

Our Approach: Limitations & Constraints

- We use the Full-Resolution 2048 x 1024 images
- Coarse Segmentation
- No Corners

Task: Cityscapes

Goal: classify objects within the city

Inspiration: Self-driving cars

Results: Model Size

Our Model Size vs UNET

- Our model contains a total of 47,998 parameters.
- UNET’s model contains a total of 2,060,424 parameters.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 64)                44928
_________________________________________________________________
dense_1 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_2 (Dense)              (None, 30)                990
=================================================================
Total params: 47,998
Trainable params: 47,998
Non-trainable params: 0
_________________________________________________________________
```

```
----------------------------------------------------------------------------------------------------
------
add_6 (Add)                     (None, 512, 512, 32) 0           up_sampling2d_6[0][0]
                                                                 conv2d_7[0][0]
----------------------------------------------------------------------------------------------------
------
conv2d_8 (Conv2D)               (None, 512, 512, 8)  2312        add_6[0][0]
====================================================================================================
======
Total params: 2,060,424
Trainable params: 2,056,648
Non-trainable params: 3,776
----------------------------------------------------------------------------------------------------
```

Results: Time to Train

Our Model Training Time vs UNET

- Our model took 20 minutes to train vs UNET’s 400 minutes to train.
- Our model trained utilizing 30% of a laptop processor, specifically the i7 8750H.
- UNET’s model trained utilizing an unspecified gpu.
- Although we do not know UNET’s gpu, it is likely a lot more powerful than a cpu in terms of training neural networks.

Results: Accuracy

Our Model Accuracy vs Ground Truth

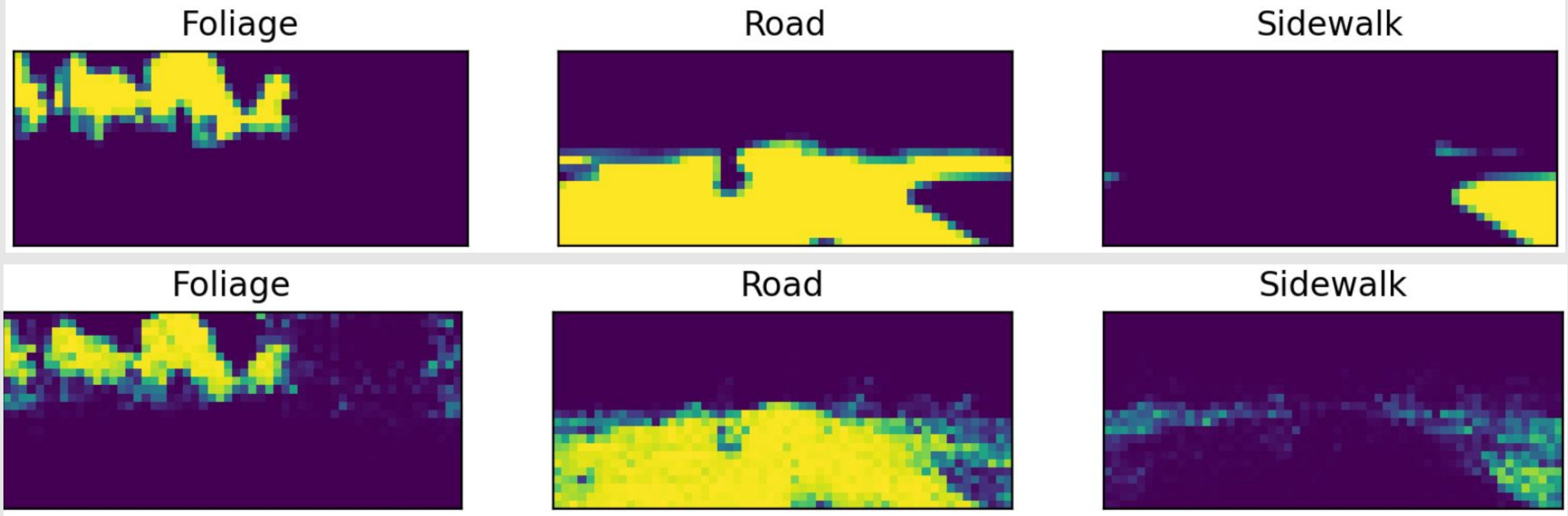
Caveat: Coarse

Our Model's accuracy after 20 minutes of training:

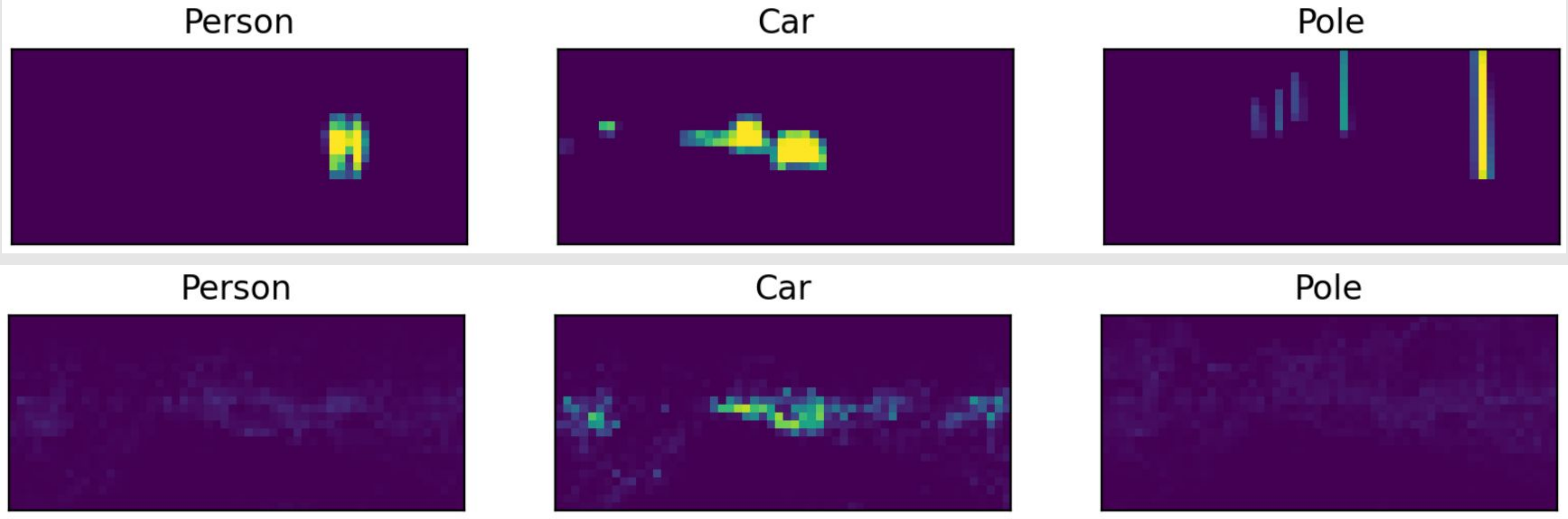
- 47,998 Parameters: 82.79%
- 203,630 Parameters: 90.08%
- 845,982 Parameters: 93.04%

UNET's pixel-wise segmentation model's accuracy: 91.69%

Results: Successes



Results: Areas of Improvement



Future Work

- Corners
- Pixel-Wise segmentation

Thank You!

References

- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., ... & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3213-3223).
- De Lathauwer, L., & Vandewalle, J. (2004). Dimensionality reduction in higher-order signal processing and rank-(R1, R2,…, RN) reduction in multilinear algebra. *Linear Algebra and its Applications*, *391*, 31-55.
- LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision* (pp. 319-345). Springer, Berlin, Heidelberg.
- Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.
- Vigueras-Guillén, J. P., Sari, B., Goes, S. F., Lemij, H. G., van Rooij, J., Vermeer, K. A., & van Vliet, L. J. (2019). Fully convolutional architecture vs sliding-window CNN for corneal endothelium cell segmentation. *BMC Biomedical Engineering*, *1*(1), 1-16.