**Semantic segmentation**

**Architecture:**

A pre-trained VGG-16 network pretrained on the ImageNet as the encoder and was converted to a fully convolutional network by converting the final fully connected layer to a 1x1 convolution and setting the depth equal of two. Performance is improved through the use of skip connections, performing 1x1 convolutions on previous VGG layers (in this case, layers 3 and 4) and adding them element-wise to upsampled lower-level layers. Each convolution and transpose convolution layer includes a kernel initializer and regularizer

### Setup:

Hyperparameters were chosen by the try-and-error process. Adam optimizer was used as a well-established optimizer. Weights were initialized by a random normal initializer.

The loss per batch tends to be below 0.100 and after 50 epochs, the average loss is around 0.044

**Rubric:**

**Build the Neural Network**

| Criteria | Meets Specifications |
| --- | --- |

|  |  |
| --- | --- |
| Does the project load the pretrained vgg model? | The function load\_vgg is implemented correctly. |
| Does the project learn the correct features from the images? | The function layers is implemented correctly. |
| Does the project optimize the neural network? | The function optimize is implemented correctly. |
| Does the project train the neural network? | The function train\_nn is implemented correctly. The loss of the network should be printed while the network is training. |

Neural Network Training

| Criteria | Meets Specifications |
| --- | --- |
| Does the project train the model correctly? | On average, the model decreases loss over time. Starts off at 1.5 and decreases to around 0.044 |
| Does the project use reasonable hyperparameters? | -Keep probability: 0.5  -Learning rate: 0.0009  -Number of epochs: 50 |
| Does the project correctly label the road? | few sample images from the output of the fully convolutional network, with the segmentation class overlaid upon the original image in green are shown after the table. |









