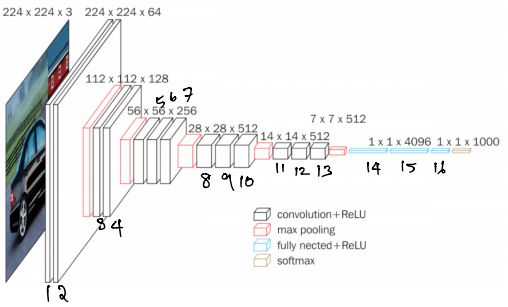
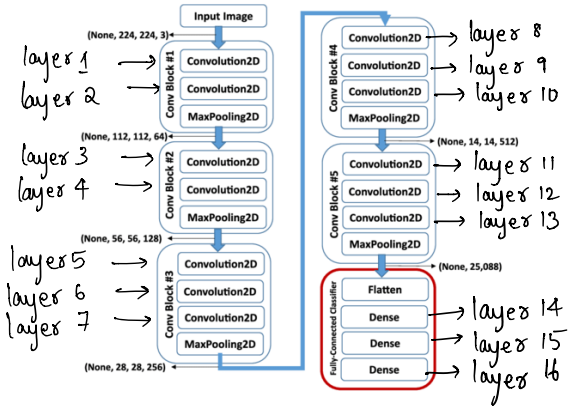
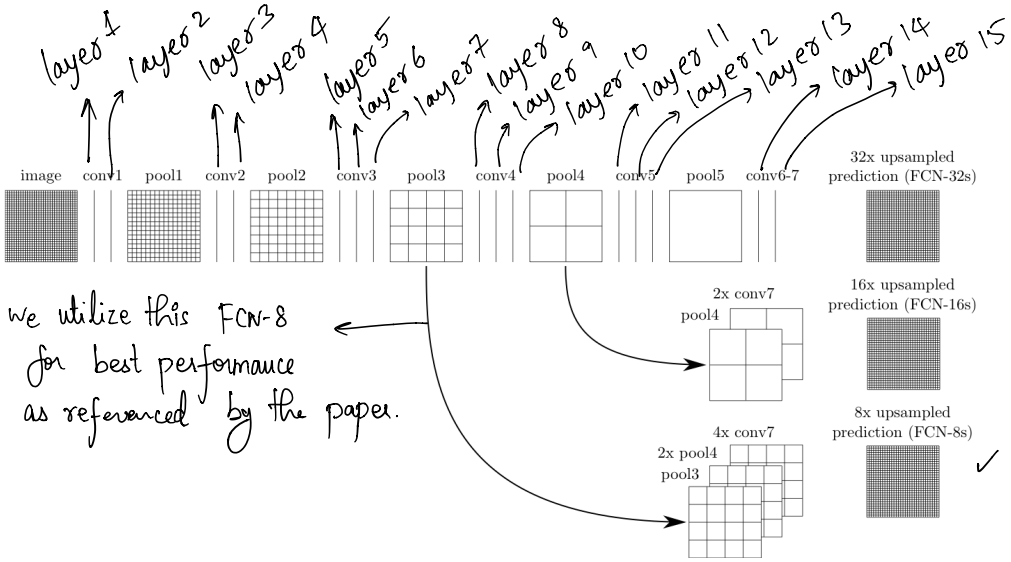
VGG-16



Vanilla VGG-16



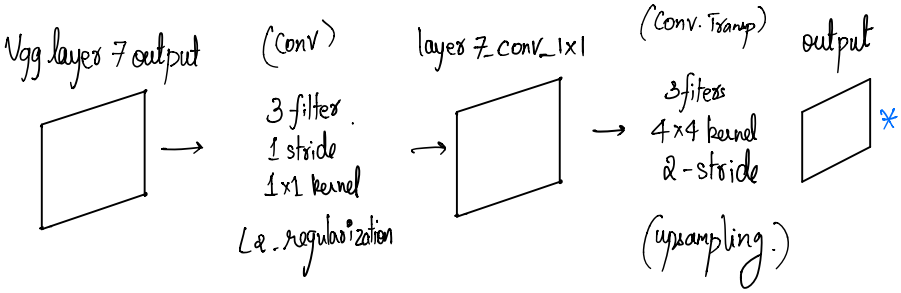


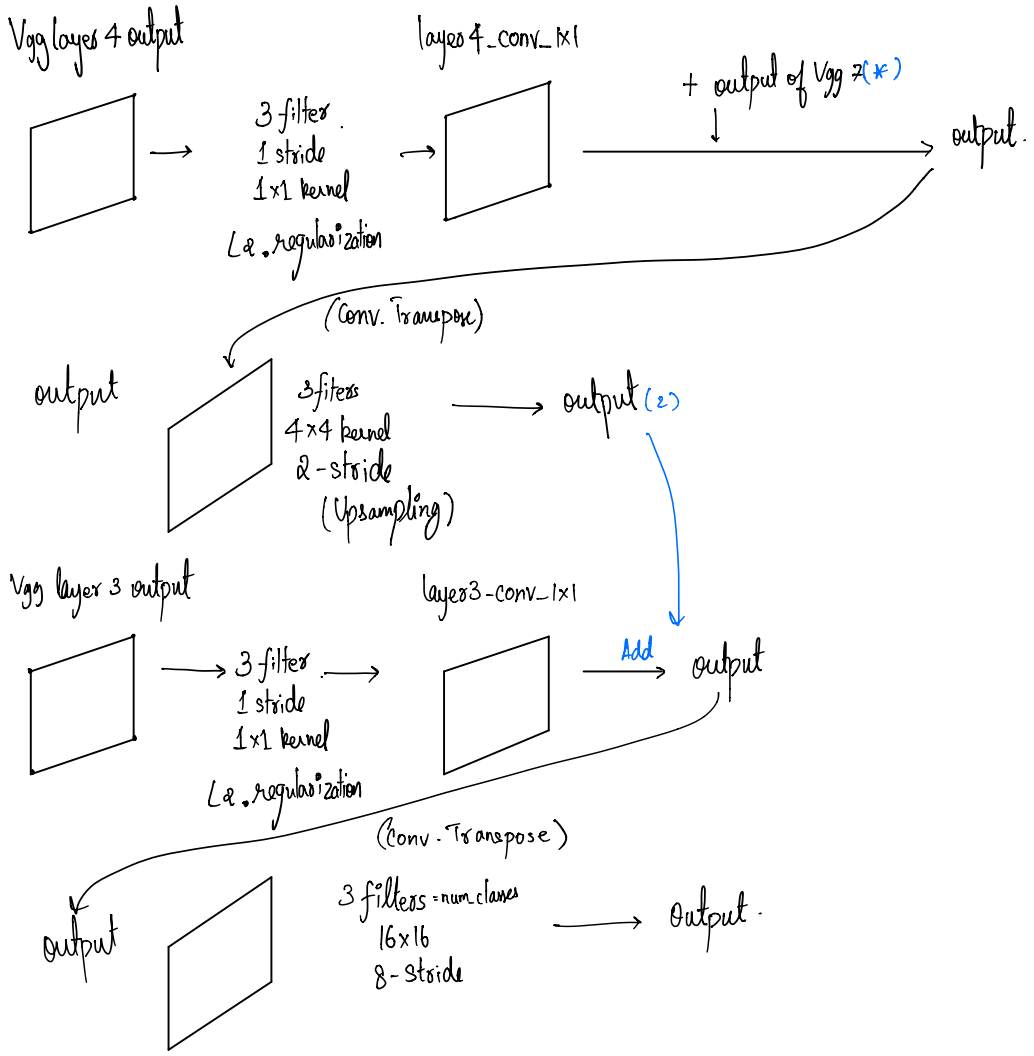
we utilize this FCN-8 for best performance as referenced by the paper.

Creating layers for a fully connected network

in this case just 3 → road, vehicles, nothing.

def layers(vgg_layer3_out, vgg_layer4_out, vgg_layer7_out, num_classes)





The vgg model utilized in the helper.py is already fully convolutionized. (it is not vanilla vgg-16)

Layer (type)	Output Shape	Param #	Connected to	
input_1 (InputLayer)	(None, 224, 224, 3)	0		
block1_conv1 (Convolution2D)	(None, 224, 224, 64)	1792	input_1[0][0]	1
block1_conv2 (Convolution2D)	(None, 224, 224, 64)	36928	block1_conv1[0][0]	2
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0	block1_conv2[0][0]	
block2_conv1 (Convolution2D)	(None, 112, 112, 128)	73856	block1_pool[0][0]	3
block2_conv2 (Convolution2D)	(None, 112, 112, 128)	147584	block2_conv1[0][0]	4
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0	block2_conv2[0][0]	
block3_conv1 (Convolution2D)	(None, 56, 56, 256)	295168	block2_pool[0][0]	5
block3_conv2 (Convolution2D)	(None, 56, 56, 256)	590080	block3_conv1[0][0]	6
block3_conv3 (Convolution2D)	(None, 56, 56, 256)	590080	block3_conv2[0][0]	7
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0	block3_conv3[0][0]	
block4_conv1 (Convolution2D)	(None, 28, 28, 512)	1180160	block3_pool[0][0]	8
block4_conv2 (Convolution2D)	(None, 28, 28, 512)	2359808	block4_conv1[0][0]	9
block4_conv3 (Convolution2D)	(None, 28, 28, 512)	2359808	block4_conv2[0][0]	10
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0	block4_conv3[0][0]	
block5_conv1 (Convolution2D)	(None, 14, 14, 512)	2359808	block4_pool[0][0]	11
block5_conv2 (Convolution2D)	(None, 14, 14, 512)	2359808	block5_conv1[0][0]	12
block5_conv3 (Convolution2D)	(None, 14, 14, 512)	2359808	block5_conv2[0][0]	13
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0	block5_conv3[0][0]	
flatten (Flatten)	(None, 25088)	0	block5_pool[0][0]	
fc1 (Dense)	(None, 4096)	102764544	flatten[0][0]	
fc2 (Dense)	(None, 4096)	16781312	fc1[0][0]	
predictions (Dense)	(None, 1000)	4097000	fc2[0][0]	

Total params: 138357544

weights = [0.3, 0.6, 0.3] → more weightage to the cars class. since classes are unbalanced.

optimization:

nn_last_layer, correct_label, learning_rate, num_classes.
(correct_label: Placeholder; learning_rate: Placeholder.; num_classes: 3)

Tensor from last layer.

Logits = Tensor from nn_last layer

Labels = Correct_labels

softmax = softmax(logits).