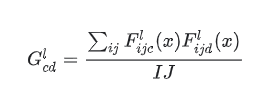
Customization



def gram\_matrix(input\_tensor):

result = tf.linalg.einsum('bijc,bijd->bcd', input\_tensor, input\_tensor)

gram\_matrix = tf.expand\_dims(result, axis=0)

input\_shape = tf.shape(input\_tensor)

i\_j = tf.cast(input\_shape[1]\*input\_shape[2], tf.float32)

return gram\_matrix/i\_j

If we use a style layer of “block3\_pool” and the output of (28, 28, 256), 256 means there are 256 number of filters in a layer (depth of an output). 28 shows the dimension of each filter. What gram matrix does it trying to measure the co-relation between two filters. In the equation ‘i’ indicates the pixels (28 pixels from both directions as taken from the above example) and ‘j’ indicates filters (which in this case 256 filters). ‘I’ and ‘J’ indicates the height and the width of each filter of the entire block layer.

def load\_vgg():

vgg = tf.keras.applications.VGG19(include\_top=True, weights=None)

vgg.load\_weights('/content/drive/MyDrive/modelWeights/vgg19\_weights\_tf\_dim\_ordering\_tf\_kernels.h5')

vgg.trainable = False

content\_layers = ['block4\_conv2']

style\_layers = ['block1\_conv1', 'block2\_conv1', 'block3\_conv1', 'block4\_conv1', 'block5\_conv1']

content\_output = vgg.get\_layer(content\_layers[0]).output

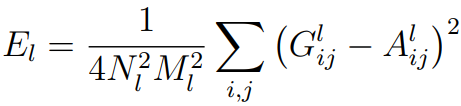
style\_output = [vgg.get\_layer(style\_layer).output for style\_layer in style\_layers]

gram\_style\_output = [gram\_matrix(output\_) for output\_ in style\_output]

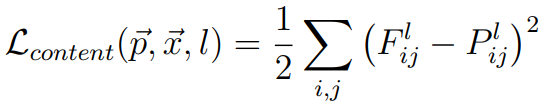
model = Model([vgg.input], [content\_output, gram\_style\_output])

return model

Content layers can be defined as 'block4\_conv2'; The style layers are defined as 'block1\_conv1', 'block2\_conv1', 'block3\_conv1', 'block4\_conv1', 'block5\_conv1'.



Style Loss [1]



Content loss [1]

To generate the images that mix the content of a photograph with the style of a painting we jointly minimise the distance of a white noise image from the content representation of the photograph in one layer of the network and the style representation of the painting in a number of layers of the CNN. So let ~p be the photograph and ~a be the artwork. The loss function we minimise is where α and β are the weighting factors for content and style reconstruction respectively [2].

References

[1]. Gatys LA, Ecker AS, Bethge M. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576. 2015 Aug 26.

<https://arxiv.org/pdf/1508.06576.pdf>

[2]. Khaligh-Razavi, S.-M. & Kriegeskorte, N. Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation. PLoS Comput Biol 10, e1003915 (2014).

<https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003915>