Creative Digital Media Design
ML Project: Image Styling & Recognition

Image Recognition

Code:

```
from tensorflow import keras
model = keras.Sequential([
   keras.Input((28, 28, 1)),
   keras.layers.Conv2D(512, 3, activation='relu'),
   keras.layers.MaxPool2D(pool_size=(3,3)),
   keras.layers.Dropout(0.2),
   keras.layers.Flatten(),
   keras.layers.Dense(64, activation='relu'),
   keras.layers.Dense(10, activation='sigmoid')
])

model.compile(optimizer='adam',
              loss=keras.losses.SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

model.summary()
```

Snippet:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 26, 26, 512)       5120
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 8, 8, 512)         0
_________________________________________________________________
dropout (Dropout)            (None, 8, 8, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 32768)             0
_________________________________________________________________
dense (Dense)                (None, 64)                2097216
_________________________________________________________________
dense_1 (Dense)              (None, 10)                650
=================================================================
Total params: 2,102,986
Trainable params: 2,102,986
Non-trainable params: 0
_________________________________________________________________
```

The result of WarsawByTytusBrzozowski.jpg is:

```
[('n03877845', 'palace', 0.20883702),
 ('n03447447', 'gondola', 0.20368342),
```

```
('n03874293', 'paddlewheel', 0.07497398),
('n03461385', 'grocery_store', 0.07299211),
('n03216828', 'dock', 0.043976508)]]
```

Vgg16 provides the following prediction about the WarsawByTutusBrzozowski.jpg.

The probability of this image being a palace = 0.20883702;
the probability of this image being a gondola = 0.20368342;
the probability of this image being a paddlewheel = 0.07497398;
the probability of this image being a grocery_store = 0.07299211;
the probability of this image being a dock = 0.043976508.

As it can be seen, it gives the probability for it being a palace, gondola, paddlewheel, grocery store and dock. Nonetheless, none of those things can actually be interpreted from the image itself. So, the prediction is not very accurate or effective.

The result of SanFrancisco.jpg is:

```
[('n03220513', 'dome', 0.23266125),
 ('n09428293', 'seashore', 0.21666795),
 ('n02894605', 'breakwater', 0.1352094),
 ('n04486054', 'triumphal_arch', 0.042043276),
 ('n03837869', 'obelisk', 0.03288306)]
```

The probability of this image being a dome = 0.23266125;
the probability of this image being a seashore = 0.21666795;
the probability of this image being a breakwater = 0.1352094;
the probability of this image being a triumphal_arch = 0.042043276;
the probability of this image being a obelisk = 0.03288306.

For SanFrancisco.jpg, it seems to predict that the image contains dome, seashore, breakwater, triumphal arch, and obelisk. In this image, the prediction seems to be more accurate as the seashore or some water body can be seen along with a breakwater structure as well as some buildings with an obelisk like structure.

After performing this analysis, we can observe that the vgg16 has a better accuracy at predicting real images than predicting artwork. The statement is justifiable for the fact that the model is trained using actual images, not art and/or animations. Therefore, it is hard to predict a reasonable (overall) estimation about the model's accuracy.

Image Styling

Trying different weightings of the model, we can make a few observations.

Firstly, the style weight needs to be extremely low if the white background should not be changed into a blue background and just the colour of the font should be changed.

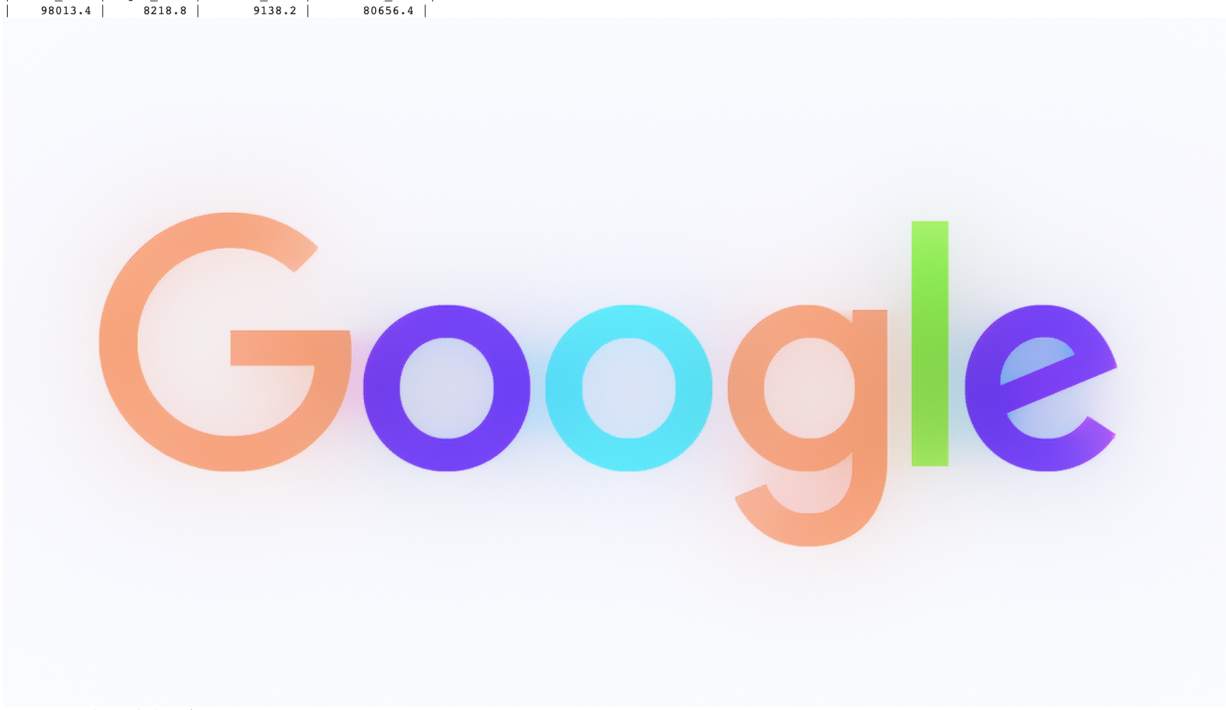
Next, in order to obtain a perfect shape and outline of the Google logo, the content weight must be significant enough to be able to achieve that.

For ensuring that the light is soft, the variation weight must be higher than 0.0002 and lower than 0.02. This ensures that the resulting/desired image is smooth and consistent with the spreading effect.

So, to achieve an image that is very similar to the output image, we use:

```
# Hyperparams
style_weight = 0.000002
content_weight = 1.0
variation_weight = 0.002
learning_rate = 5.0
ITERATIONS = 110
```

```
Iteration 110:
| total_loss | style_loss | content_loss | variation_loss |
|    98013.4 |     8218.8 |       9138.2 |       80656.4 |
```




```
save to google_styled110.jpg
```