

Deepdream Implementation with Tensorflow

Reference:

TensorFlow 2.0 Computer Vision Cookbook

<https://www.packtpub.com/product/tensorflow-2-0-computer-vision-cookbook/9781838829131>

<https://www.tensorflow.org/tutorials/generative/deepdream>

Deepdream can visualize what the model learns and help us understand the how the model differentiates two different classes of objects. Additionally, it helps us to understand how close the two classes of objects are since it might not be similar to human instincts.

The base model we use is InceptionV3

Forward pass an image through the base model and calculates the gradient of the image for each layer. By modifying the image by gradient ascent to create the visual of whatever model learned.

Deeper layers respond to higher-level features (such as eyes and faces),

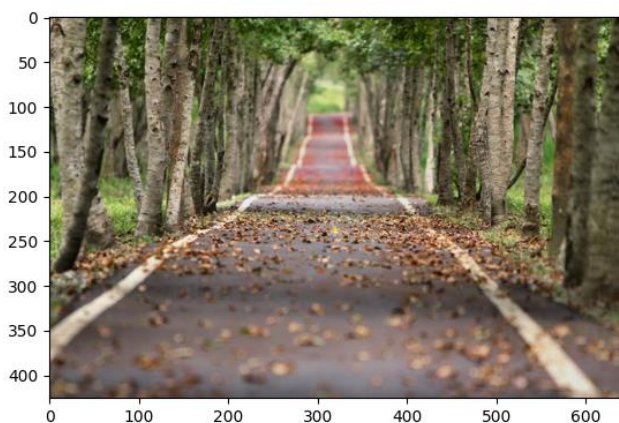
while earlier layers respond to simpler features (such as edges, shapes, and textures)

The loss of each layer will need to be normalized.

```
for activation in activations:
    loss = tf.math.reduce_mean(activation) # Normalize
    losses.append(loss)
```

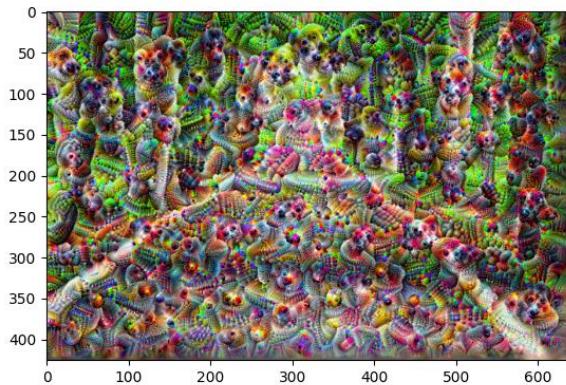
We need to apply gradient ascent at different scales otherwise the only the similar pattern will be used on the image. By increase the size of the image by an octave, a various of patterns can be created.

```
original_image = load_image('road.jpg')
show_image(original_image / 255.0)
```



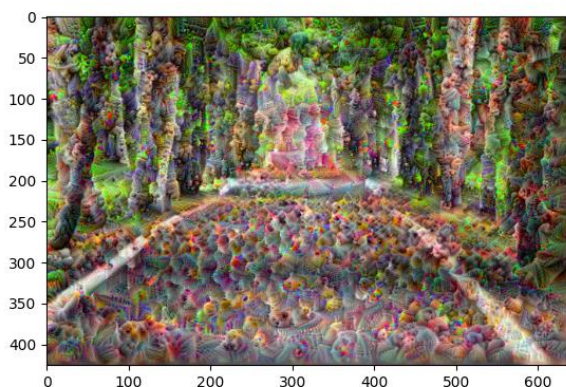
Default

```
dreamy_image = DeepDreamer().dream(original_image)
show_image(dreamy_image)
```



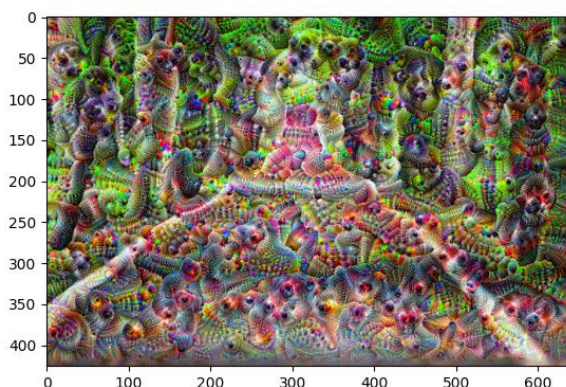
Add mixed7 layer

```
dreamy_image = (DeepDreamer(layers=['mixed2', 'mixed5', 'mixed7'])).dream(original_image)
show_image(dreamy_image)
```



Change the octave factor to -3, -1, 0, 3 (default -2 ~ 3)

```
dreamy_image = (DeepDreamer(octave_power_factors=[-3, -1, 0, 3])).dream(original_image)
show_image(dreamy_image)
```



Apply it on image contend a human.

Original Image



```
original_image = load_image('KP.jpg')
dreamy_image = (DeepDreamer(layers=['mixed2', 'mixed3'], octave_power_factors=[-5, -3,
0])).dream(original_image, steps = 150))
show_image(dreamy_image)
```

