MARIA PIOTROWSKA 38671 & KAMIL PODWORSKI 38672

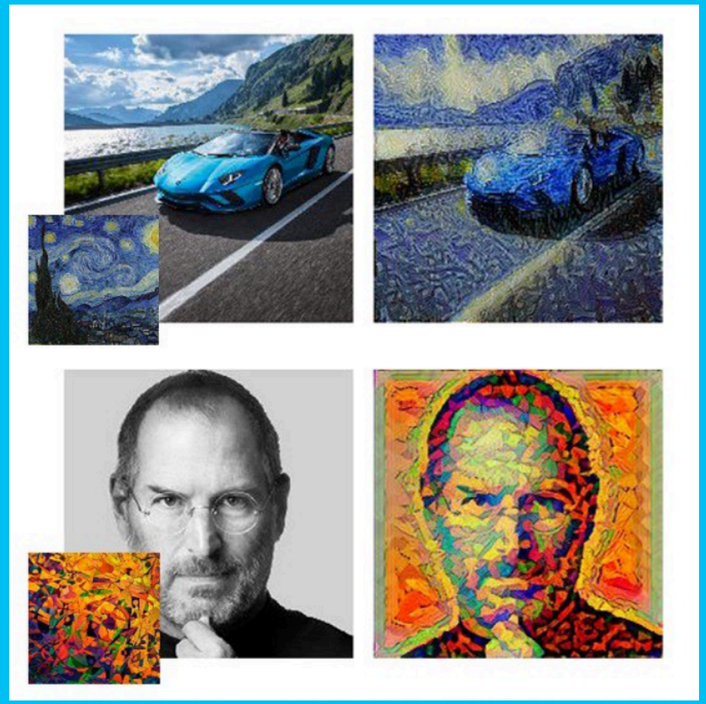
ADVANCED PHYTON PROJECT

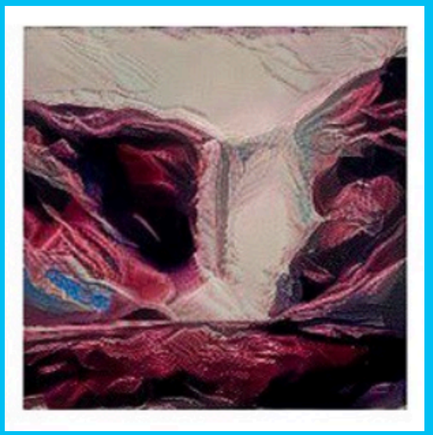
“NEURAL STYLE TRANSFER”

AT THE BEGINNING

We searched through various websites to look for:

- inspiration
- knowledge
- tips





Link:

https://medium.com/tensorflow/neural-style-transfer-creating-art-with-deep-learning-using-tf-keras-and-eager-execution-7d541ac31398

FIRST ATTEMPT

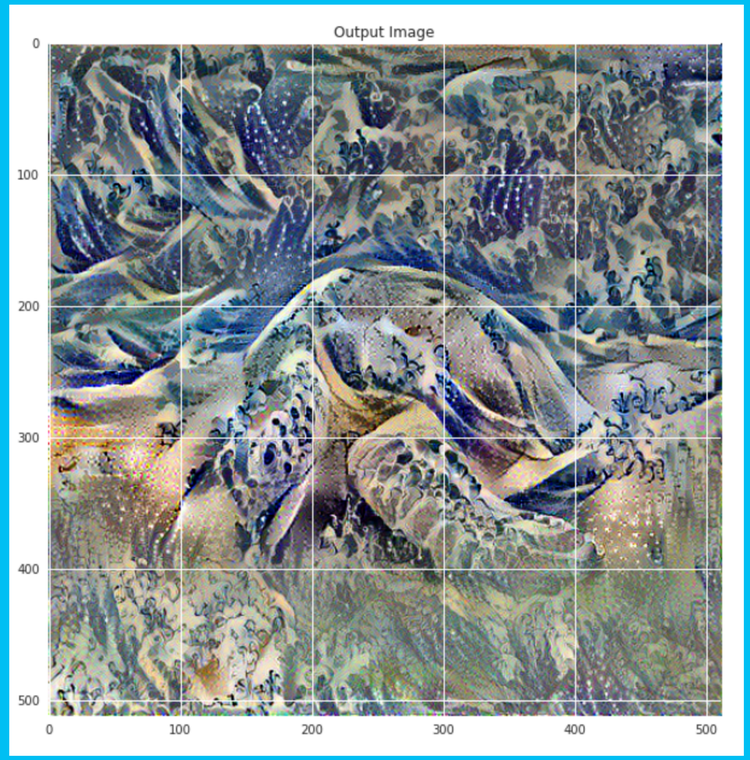
Was very difficult, cause we were ambitious!

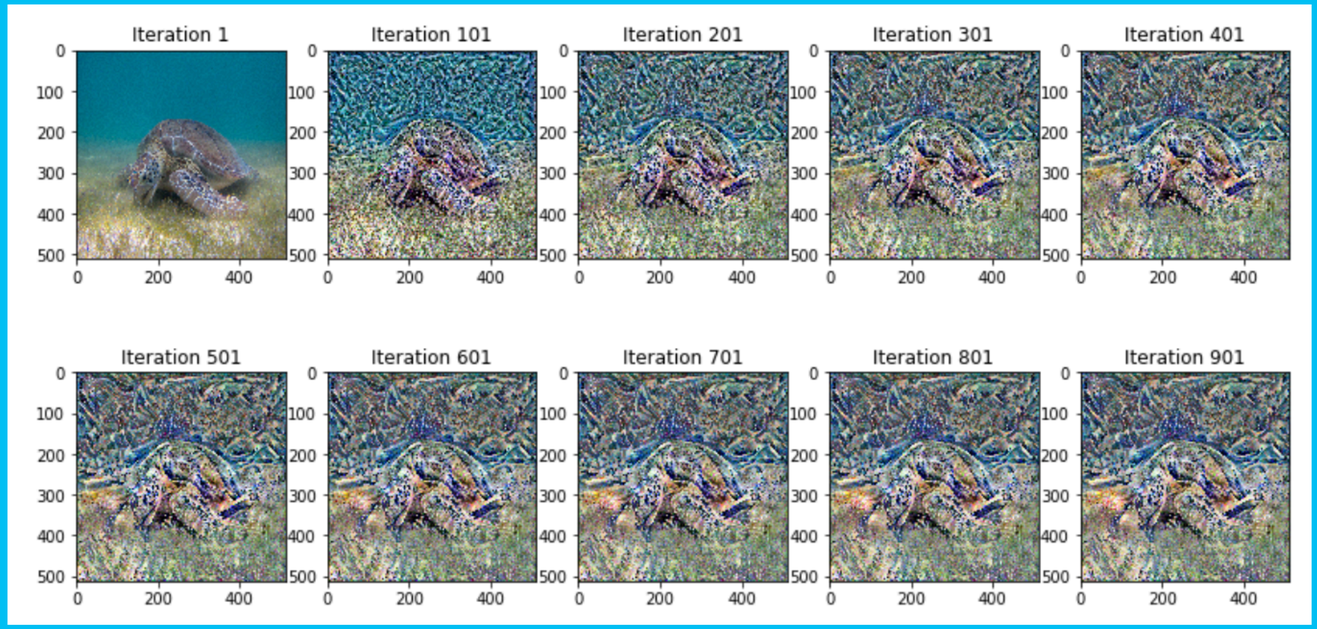
But we had a keras problem on both computers.

We decided to look for something else.

Link:

https://medium.com/tensorflow/neural-style-transfer-creating-art-with-deep-learning-using-tf-keras-and-eager-execution-7d541ac31398





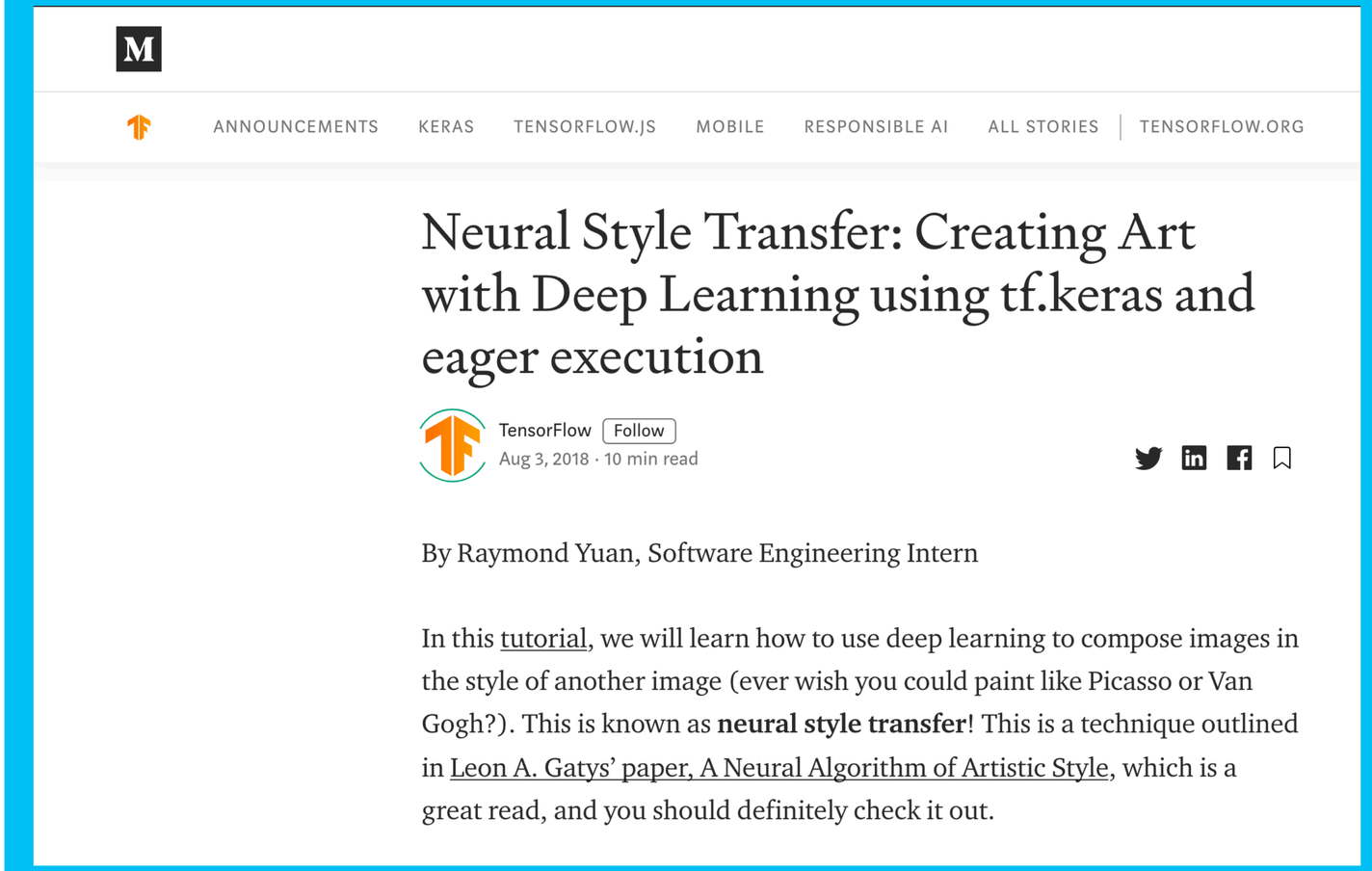
SECOND ATTEMPT

This instruction helped us solve a few problems.

We understood more, but the code was too complicated to tamper with it.

Link:

https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-neural-style-transfer-ef88e46697ee



A FEW ATTEMPTS LATER

ANOTHER ATTEMPT

From tensor flow.org we took our primary code.

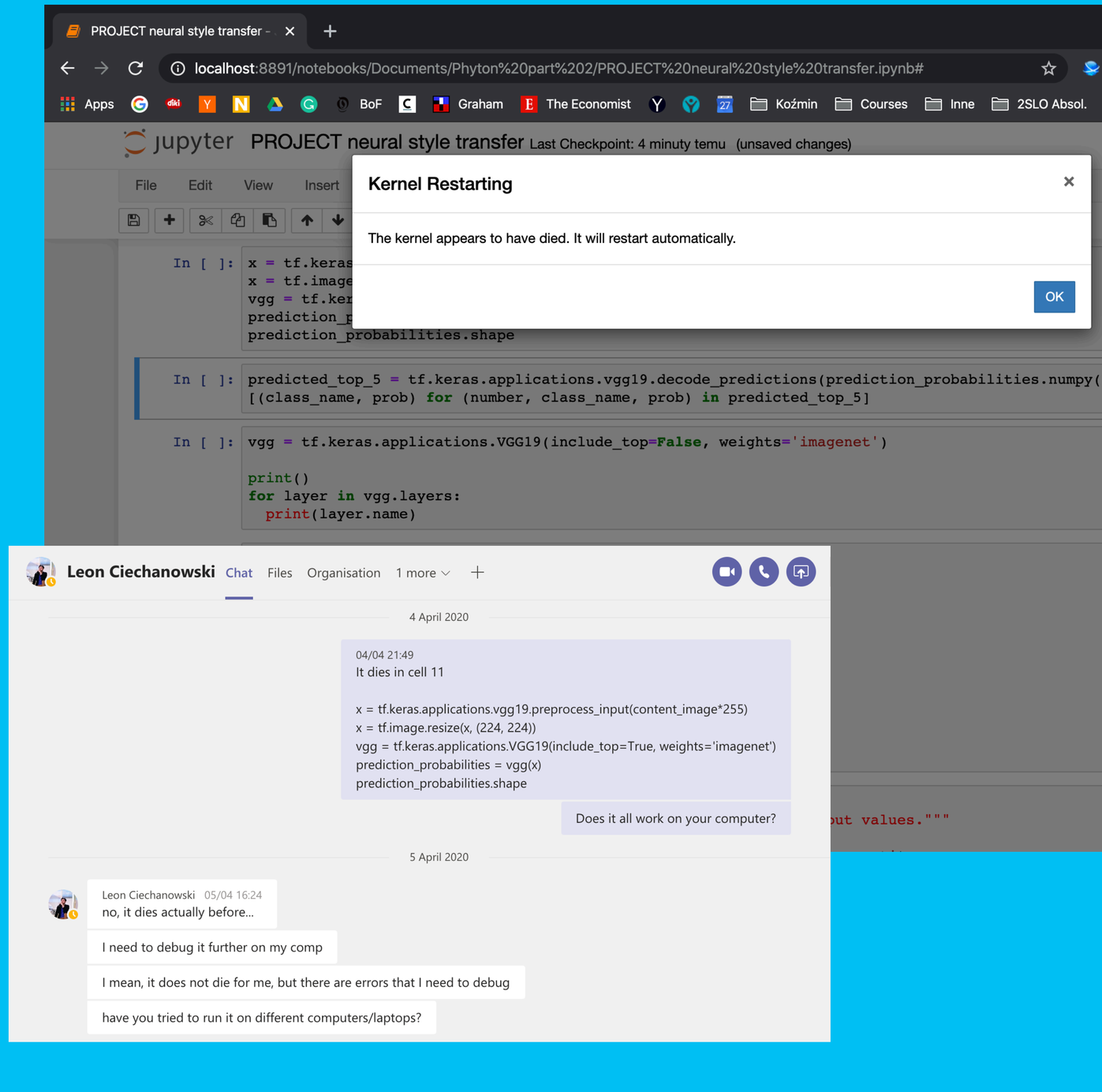
We changed it.

But we had a problem.

Even switching from new tenserflow to tensorflow 1.15.0 did not help.

Link:

https://www.tensorflow.org/tutorials/generative/style_transfer



BUT FINALLY WE MANAGED TO...

DO THIS

FROM THIS

BACKGROUND

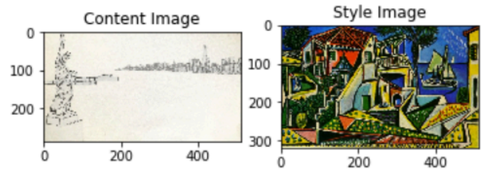
Because of a previously mentioned problem with tenserflow, we run just a part of the code.

But luckily we could see the results of neural style transfer.

```
content_image = load_img(content_path)
style_image = load_img(style_path)

plt.subplot(1, 2, 1)
imshow(content_image, 'Content Image')

plt.subplot(1, 2, 2)
imshow(style_image, 'Style Image')
```




```
import tensorflow_hub as hub
hub_module = hub.load('https://tfhub.dev/google/magenta/arbitrary-image-stylization-v1-256/1')
stylized_image = hub_module(tf.constant(content_image), tf.constant(style_image))[0]
tensor_to_image(stylized_image)
```

```
x = tf.keras.applications.vgg19.preprocess_input(content_image*255)
x = tf.image.resize(x, (224, 224))
vgg = tf.keras.applications.VGG19(include_top=True, weights='imagenet')
prediction_probabilities = vgg(x)
prediction_probabilities.shape
```

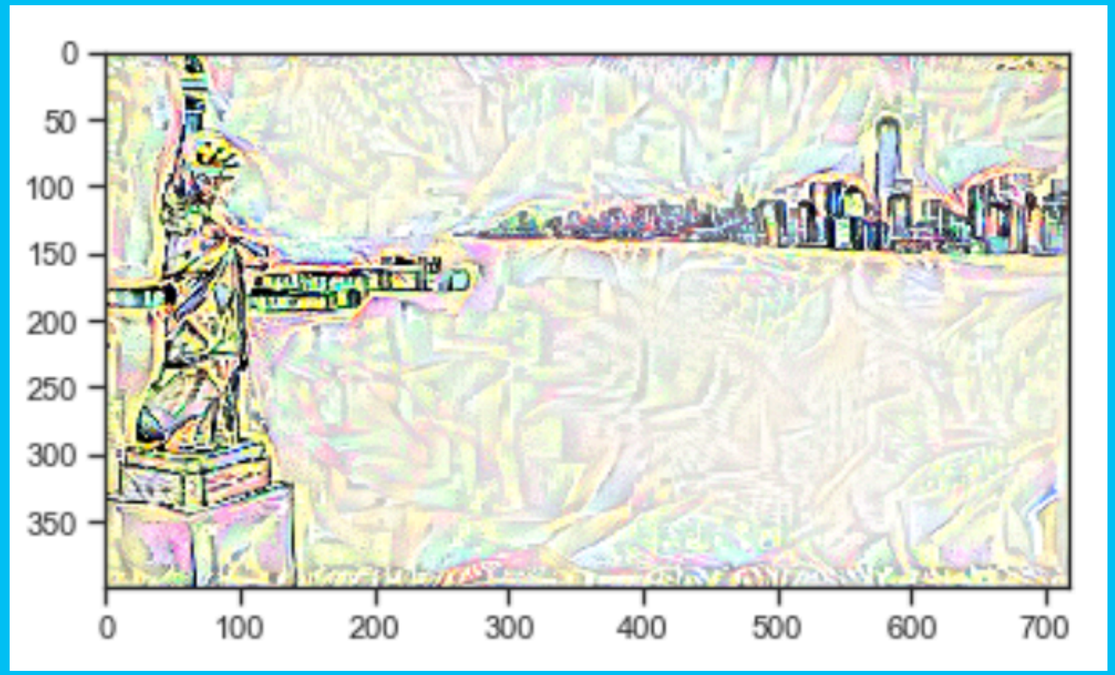
```
predicted_top_5 = tf.keras.applications.vgg19.decode_predictions(prediction_probabilities.numpy())[0]
[(class_name, prob) for (number, class_name, prob) in predicted_top_5]
```

ALSO...

AFTER NEURAL STYLE TRANSFER CLASSES:

We adjusted the code from our classes to our photos.

However the result was not as good as with the previous code.



THANK YOU FOR READING!

MARIA PIOTROWSKA 38671 & KAMIL PODWORSKI 38672