Neural Style Transfer

Aneesh Joshi

About me

- 4th year CSE undergrad
- Ex Game Developer
- ML Master race
- Information Extraction using DL

Goals of this talk

- Introduce Style Transfer (duh)
- Bring people into the DL master race
- Introduce tensorflow as python tools
- Provide something not found online

Style Transfer

Style Transfer

Ugh!

Style Transfer

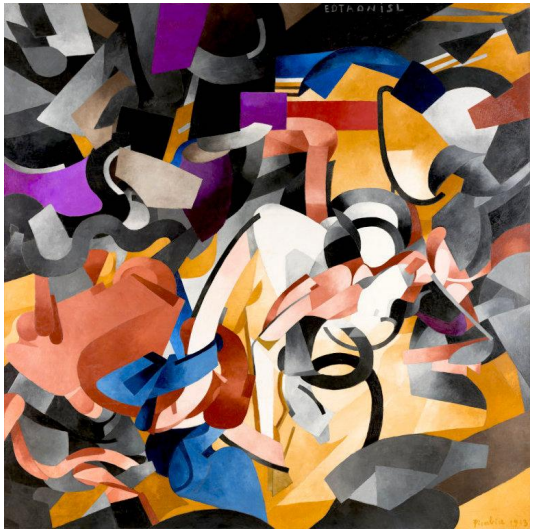

Pablo Picasso



Whoa!

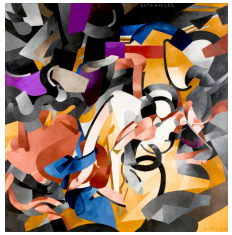
How does one go about this?

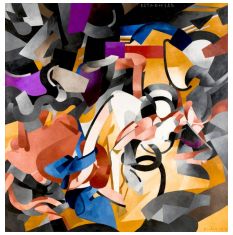
Content Image

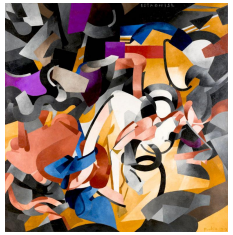


Style Image

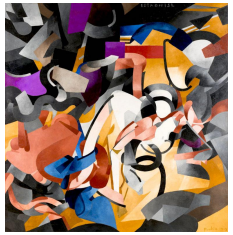
Combined Image







$$\mathcal{L}_{\text{content}}\left(\text{  }, \text{  }\right) \approx 0$$



$$\mathcal{L}_{\text{style}}\left(\text{  }, \text{  } \right) \approx 0 \qquad \mathcal{L}_{\text{content}}\left(\text{  }, \text{  } \right) \approx 0$$

Optimisation

Finding an alternative with the most cost effective or highest achievable performance under the given constraints, by maximizing desired factors and minimizing undesired ones.

$$\text{Maximize} : A = 2hr + \tfrac{1}{2}\pi r^2$$

$$\text{Constraint} : 12 = 2h + 2r + \pi r$$

Optimisation

Finding an alternative with the most cost effective or highest achievable performance under the given constraints, by maximizing desired factors and minimizing undesired ones.

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \left(\alpha \mathcal{L}_{\text{content}}(\mathbf{c}, \mathbf{x}) + \beta \mathcal{L}_{\text{style}}(\mathbf{s}, \mathbf{x}) \right)$$

Take the average?

No

Will produce bad non-intelligent results

content and style losses are based not on per-pixel differences between images, but instead in terms of higher level, more perceptual differences between them.

We need a system which:

Understands abstract concepts like lines and shapes

Teaching such concepts is difficult mathematically/programmatically

Enter : Deep Learning

Deep Learning would require you to understand:

- Neural Networks
- Convolutions
- Optimisation
- And on and on

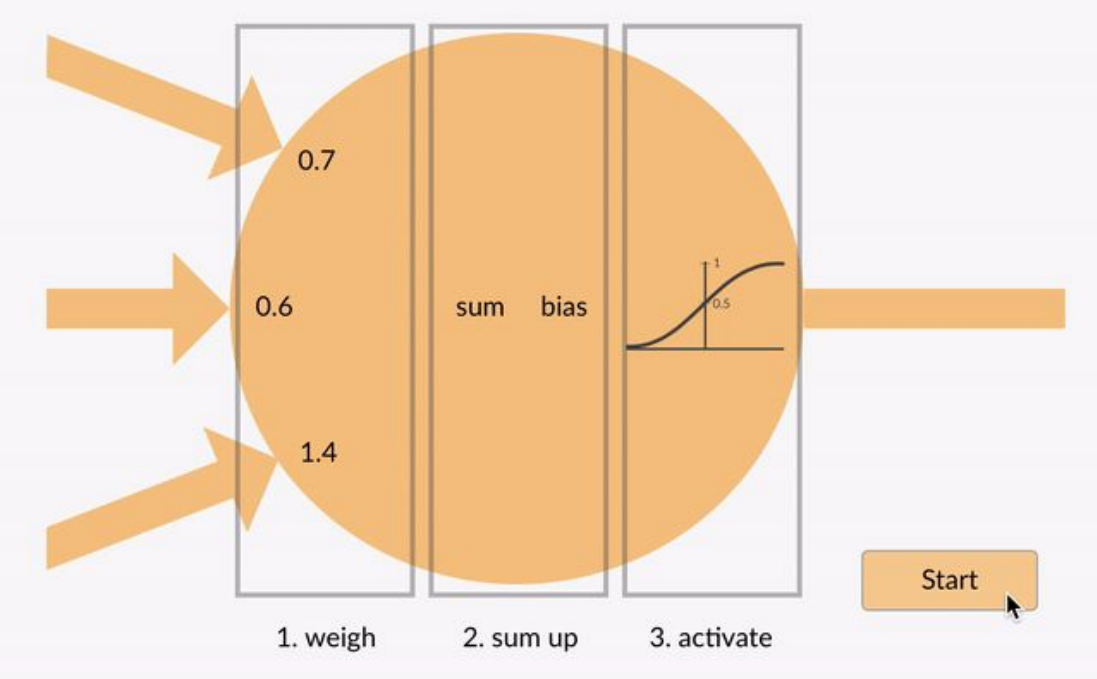
We don’t have time to go through this

Solution:

Let’s keep it simple.

Look at everything as an optimisation problem

But first, what is a neuron?



But what are they really?

A parameter wiggler.

It wiggles the parameters

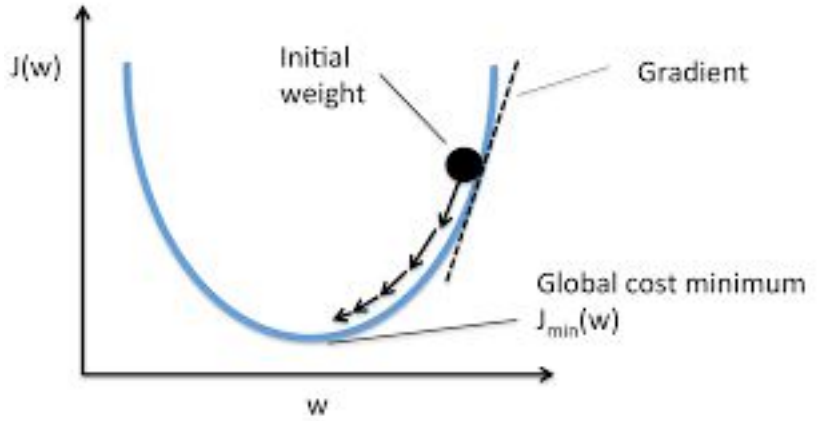
till it predicts correctly

I.e., till the error is minimised.

Most importantly:

The neuron decides “by itself” how to wiggle the parameters.

So, we just give it a task and it “figures out” what the correct set of parameters is.



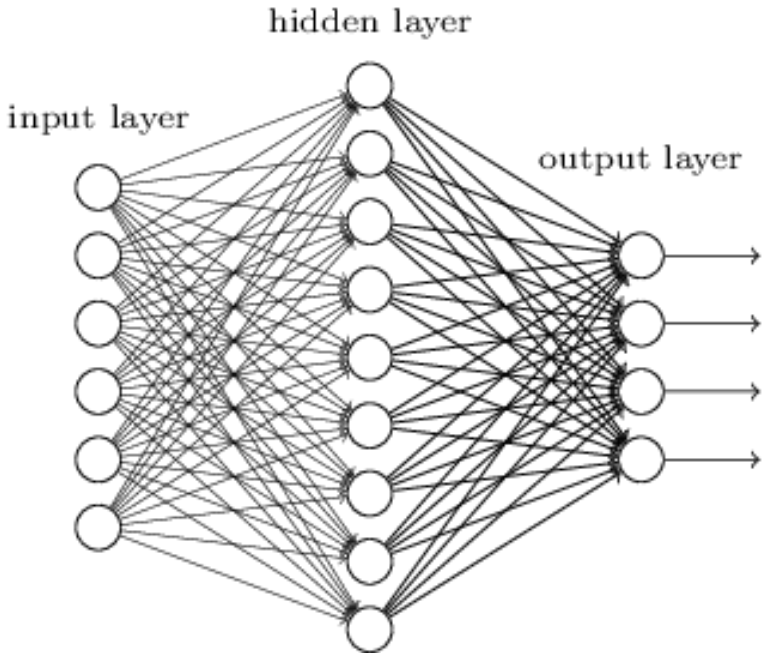
What can a simple parameter wiggler do?

Alone: not much

Combined: A lot!

This combined form is

a neural network

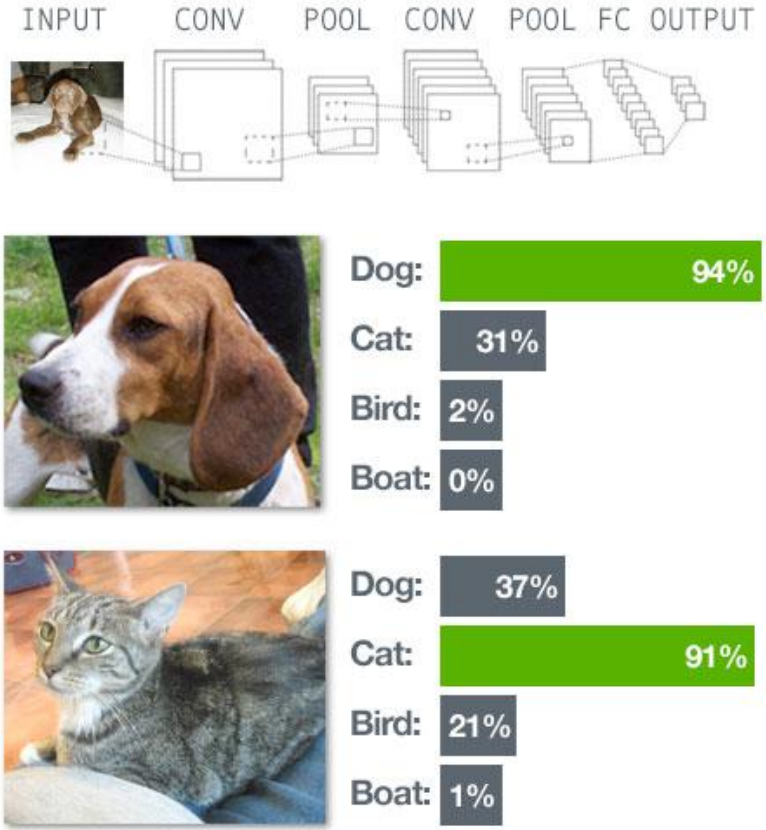


How it works

We give it an input

It predicts an output

We know the correct output



If the output is wrong we get a high error.

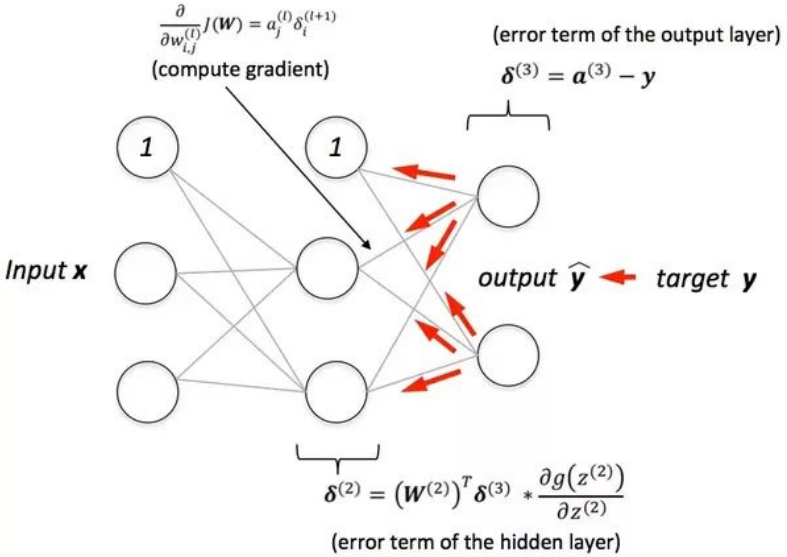
We let the network know:

The network feels bad.

It wiggles it parameters to

reduce the error.

How does it learn the direction to wiggle in?



Framed as an optimisation problem

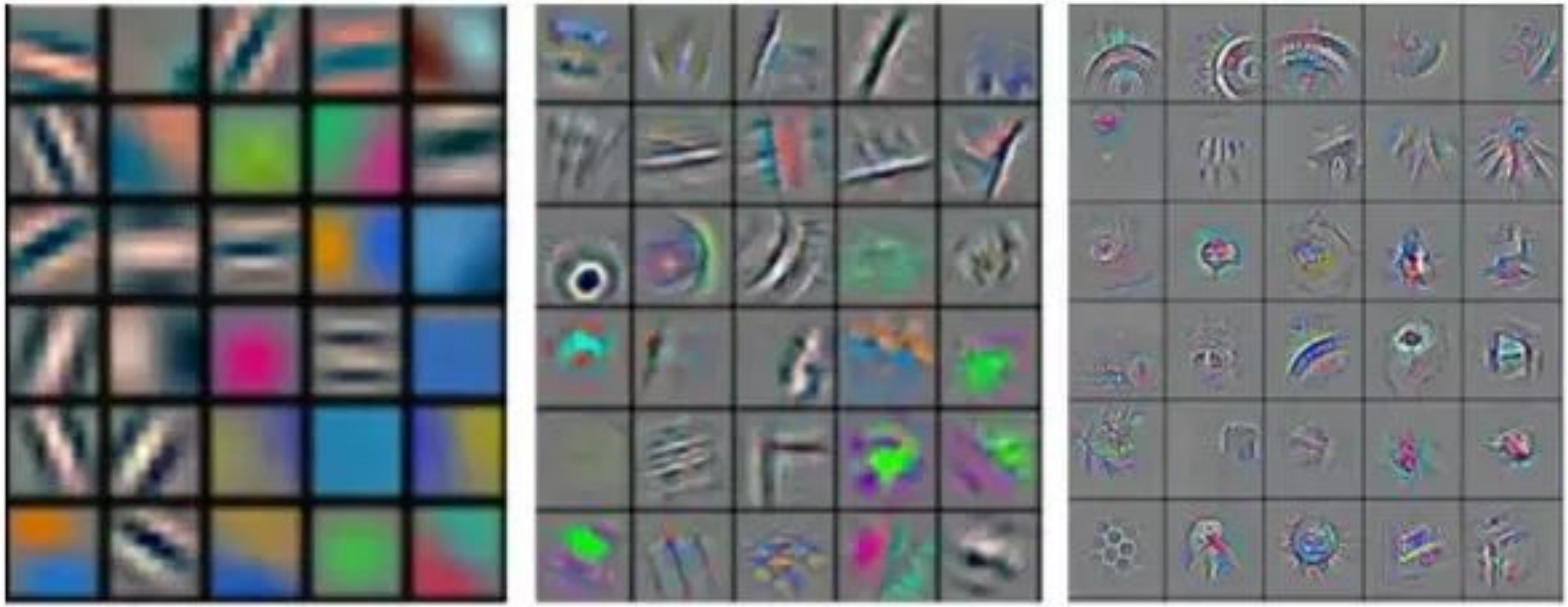
We want to minimise the error

That’s it.

Let the network decide the rest.

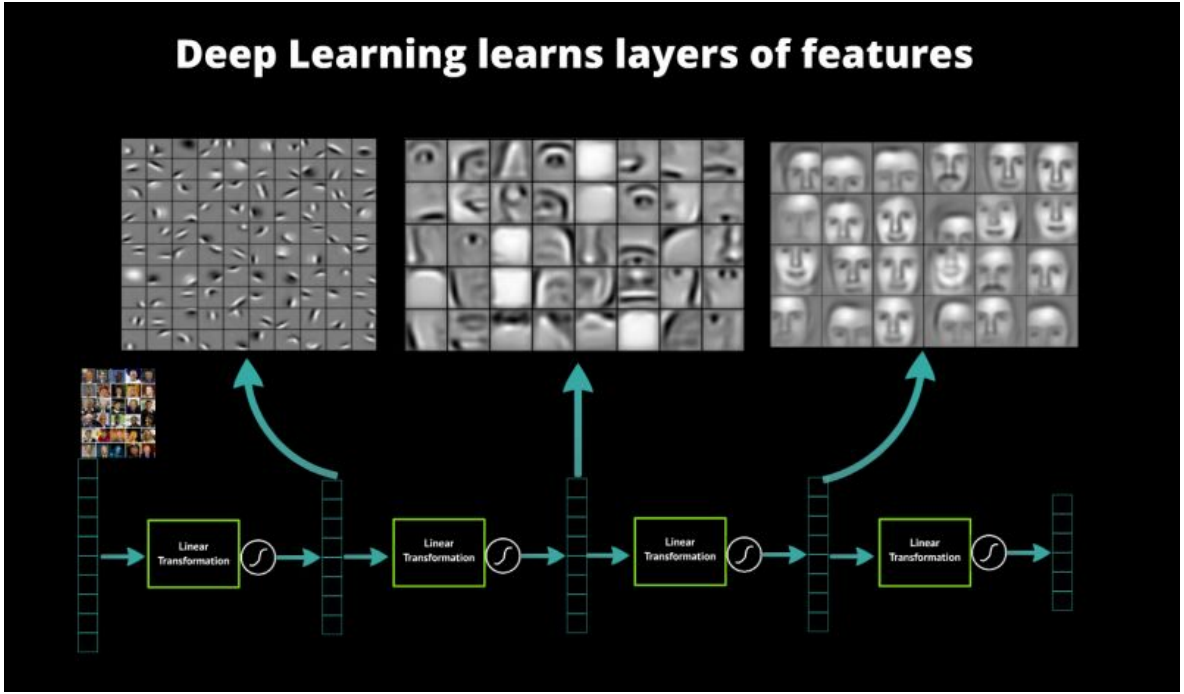
Error: distance between prediction and expected output

In the process of reducing errors, it learns interesting representations of the input data.



Learned features and their aggregation using ImageNet data

These representations are spread throughout the different layers



It extracts features because they allow it to reach its ultimate goal: to minimise error

Back to Style Transfer!

We don’t care about the predictions for now

We care about the extracted features

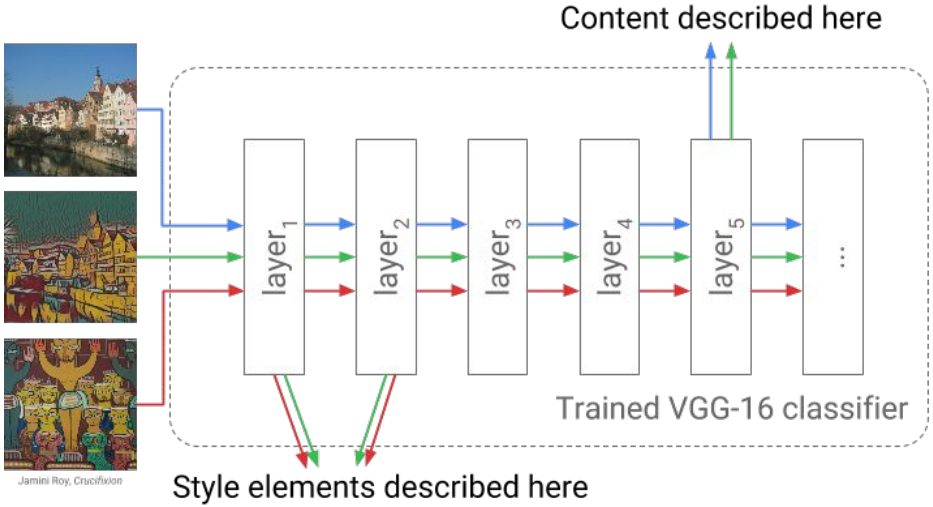
They capture abstract ideas like edges and shapes.

Perfect!

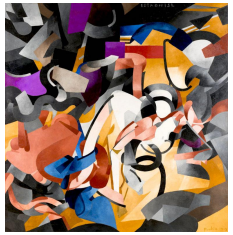
How do features help?

The features give us a way of measuring distance between images

based on features rather than pixels



We now have a way of calculating these losses in a meaningful way



$$\mathcal{L}_{\text{style}}\left(\ ,\ \right) \approx 0 \qquad \mathcal{L}_{\text{content}}\left(\ ,\ \right) \approx 0$$

Transfer Learning

For the network to learn features, it needs to be trained on a task.

We could train our own network.

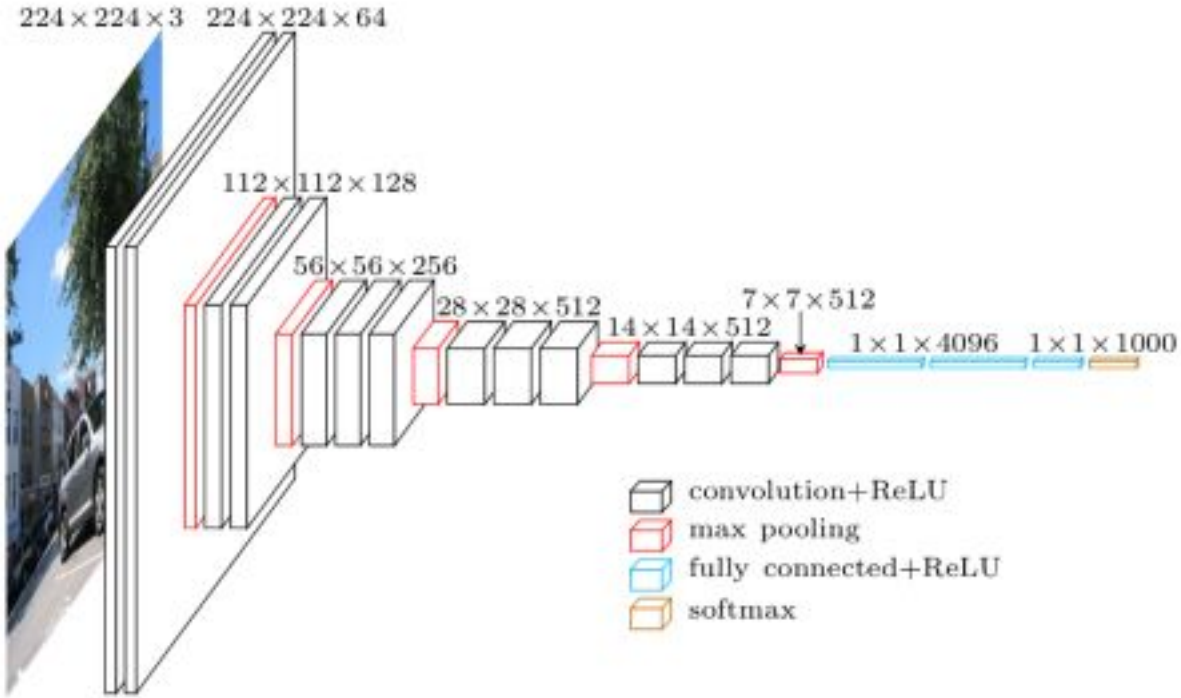
Takes time, data, machines

Transfer Learning

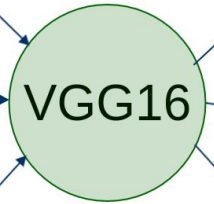
- We will use the pre trained VGG16 neural network for our task.
- ImageNet Competition
- Use 16 layers involved with feature extraction

VGG16

VGG16 Architecture



Architecture



Layer i

Content image features

Minimise Distance

Combined image features

Minimise Distance

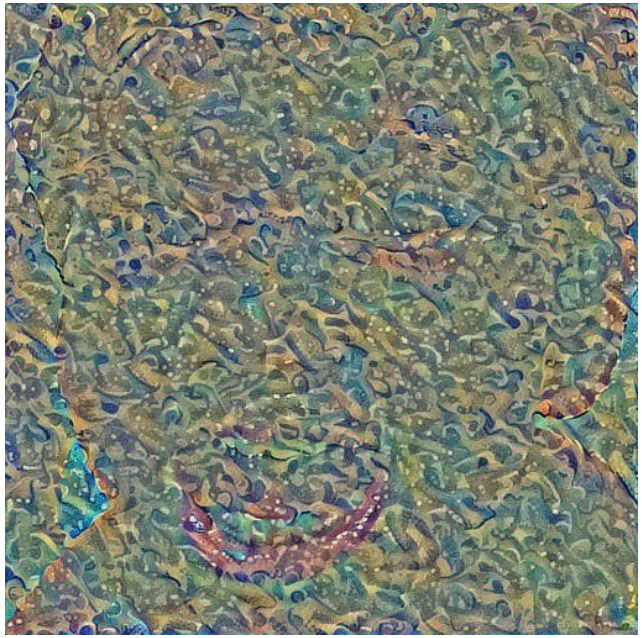
Style image features

Content Loss

Style Loss

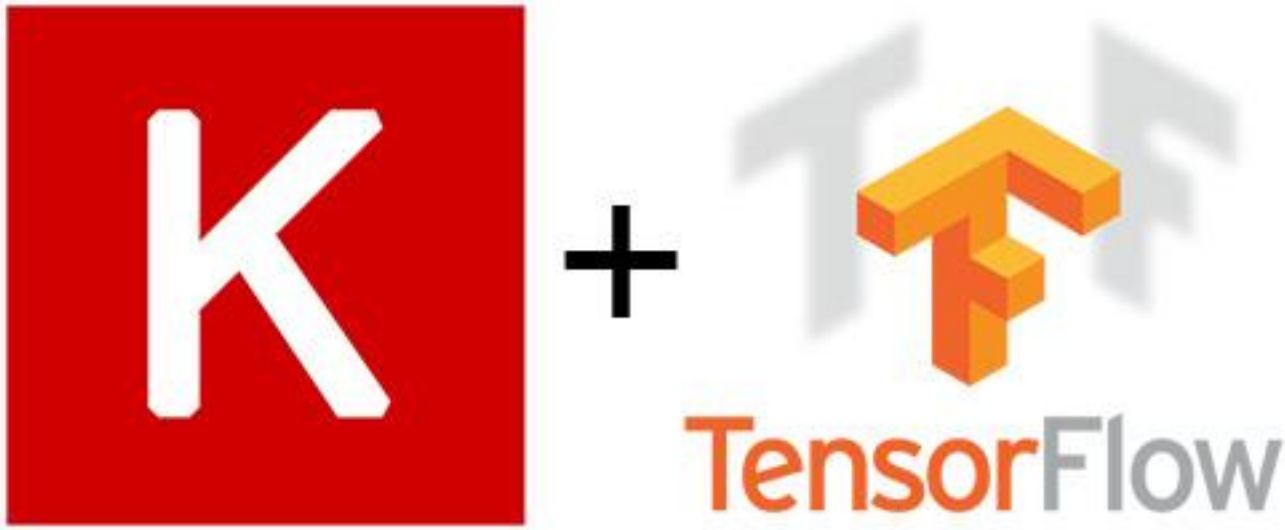
Total Loss

Over several iterations:



Tying it up with Python

Tying it up with Python



Thank you!