Artistic Style Transfer

Course: Computer Vision

Edgar Roman-Rangel.
francisco.roman@itam.mx

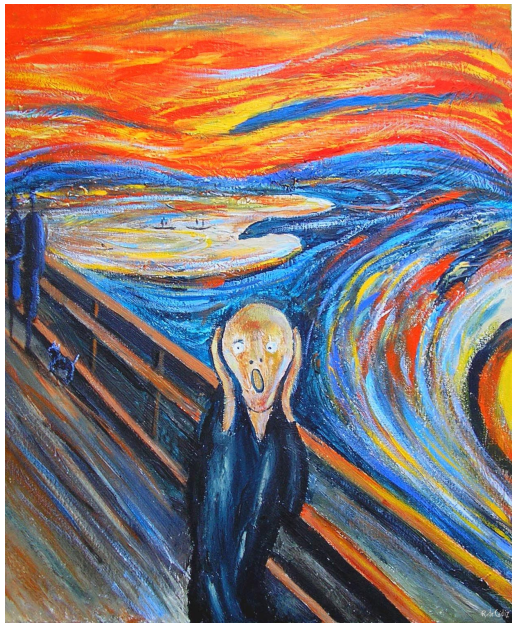
Instituto Tecnológico Autónomo de México, ITAM.

November 18th, 2020.

Outline

Goal

- Merge the content of one image with the style of another one.
- Apply the style of an image onto the content of another one.



Some available apps

Prisma

https://prisma-ai.com/

Lucid

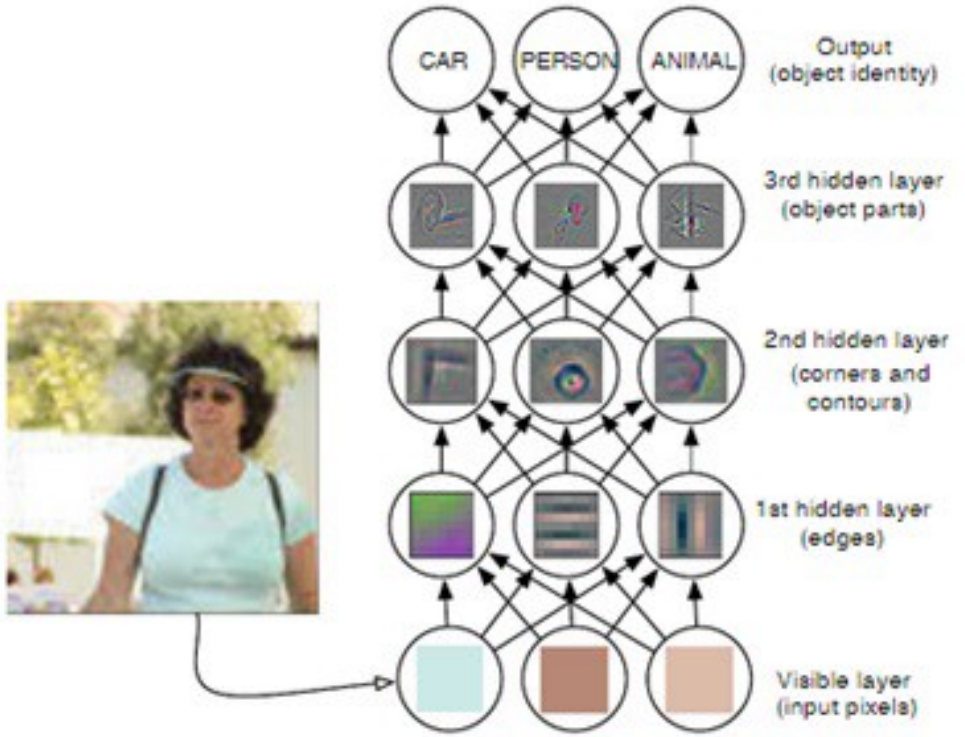
https://play.google.com/store/apps/details?id=com.doodle.doodle

Style

Q: How do we extract/isolate style from an image?

Style

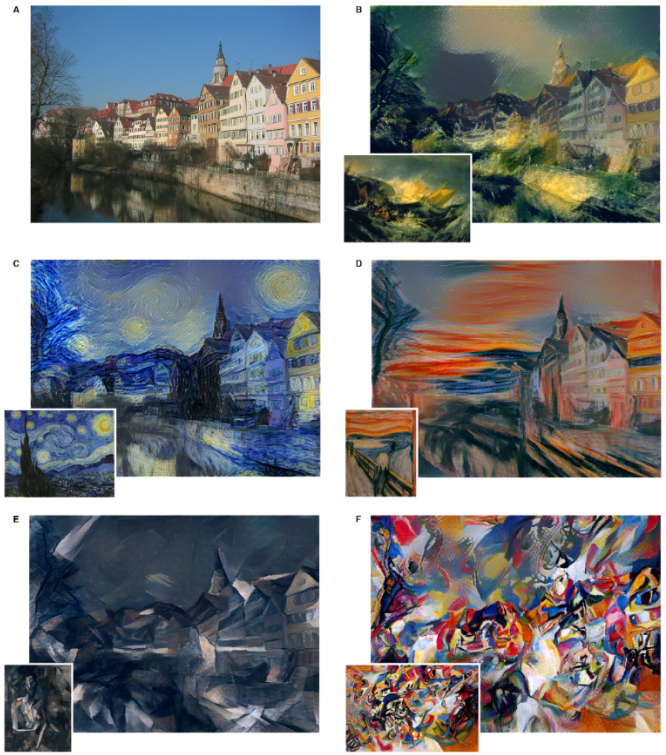
Q: How do we extract/isolate style from an image?
A: Remember the hierarchy of filters learned by CNNs. (Zeiler and Fergus, 2014).



Using filters for style transfer

Gatys et al., exploited this knowledge of filters.

- Gatys et al. 2014. A Neural Algorithm of Artistic Style. ArXiv.
- Gatys et al. 2016. Image Style Transfer Using CNNs. CVPR.



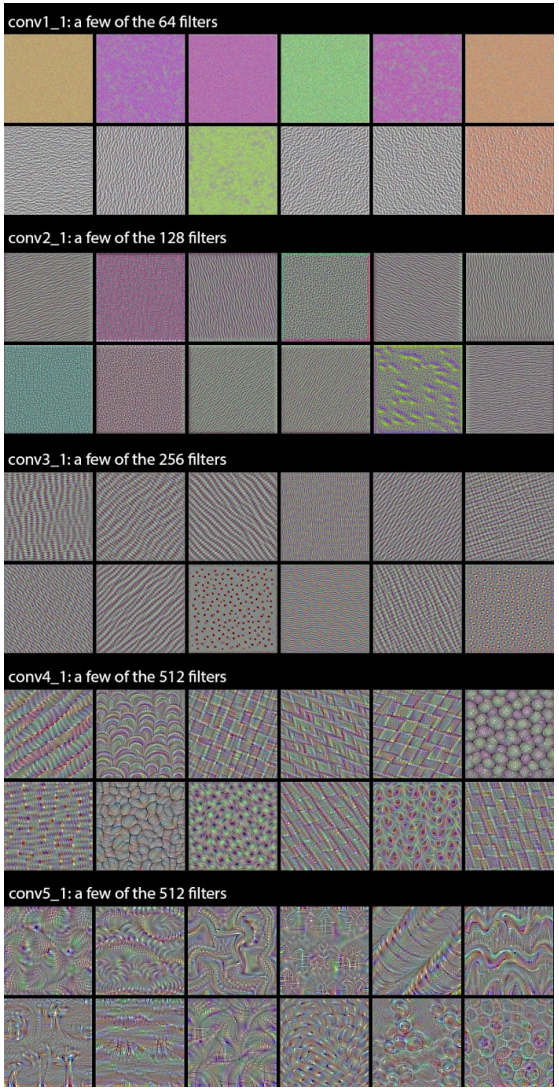
CNN layers

Content

Late hidden layers.

Style

Mid hidden layers.



Content loss

Mean squared error:

$$\mathcal{L}_c = \sum_l \sum_{i,j} \left(F^l_{i,j} - P^l_{i,j} \right)^2,$$

where, l is a chosen layer; F is the feature map for our content image; and P is the feature map of our generated image.

Style loss

Similarly, but measuring correlation between feature maps.

e.g., *if a blue patter often co-occurs with a square pattern, then highly probable there is a blue square pattern in the image*.

$$\mathcal{L}_s = \sum_l \sum_{i,j} \left(G^{s,l}_{i,j} - G^{p,l}_{i,j} \right)^2,$$

where, s and p indicate *style* and *produced* images, respectively; and G is a Gram matrix defined as,

$$G_{i,j} = \sum_k F_{i,k} F_{j,k},$$

where, k iterates over the different filters of a selected layer.

Total loss

Combine content and style losses:

$$\mathcal{L} = \alpha \mathcal{L}_c + \beta \mathcal{L}_s.$$

Pipeline

How do we do all this?

1. Load a pre-trained model, e.g., VGG Net.
2. Select some hidden layers for content and style outputs.
3. Pass content image and obtain content reference (output).
4. Pass style image and obtain style reference (output).
5. Pass a random image and get its outputs.
6. Compute loss against references.
7. Back-propagate error all the way to the input random image.
8. Update random image.
9. Repeat from step 5 until convergence.

Q&A

Thank you!

francisco.roman@itam.mx