JULY 11, 2021 - 11H

JUNE 15, 2021 - 15H

// APRIL 2021

FINAL PROJECT
STYLE TRANSFER

CAS CS 585

BY
Isabella Rocha, Skyler Gauuan,
Benjamin Reichelt

MARCH 22, 2021 - 15H

TOPIC

Our topic is style transfer. An optimization technique taking two images:

1. Content image
2. Style reference image

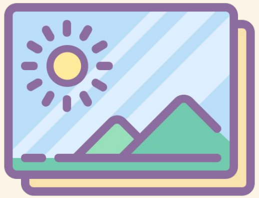
And blending them together so the output image looks like the content image, but portrayed in the style of the style reference image.

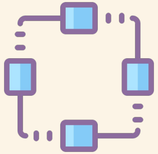
MOTIVATION

Style transfer allows you to manipulate images according to any specific art style you desire. Giving you an idea of how that artist would have painted your image.

BACKGROUND

There is an original style-transfer algorithm called *Neural Style Transfer* which optimizes the image content to a particular style. It does so by training from scratch on each specific style. We used a modern approach referenced as *Fast style Transfer for Arbitrary Styles*. This approach uses a neural network that can test on arbitrary styles without retraining. This avenue is much faster.





GOAL

INTERESTS

It was interesting to see what was interpreted as a ‘style’ of a given piece of art, and how the model attempted to recreate it with the content image. How different people’s faces changed depending on how a style was.

Our goal was to analyze how well the Fast Style Transfer algorithm worked on varying content images (faces) and varying style images (art pieces).

DIFFICULTIES

Aggregating our quantitative data with the results in a concise way. Compiling our findings and making generalised statements on how well the algorithm performs on different types of images. There were also a lot of data we had to measure.

METHODS

Uses deep learning to compose one image in the style of another image. This is also known as neural style transfer.

ORIGINAL-STYLE

NEURAL STYLE TRANSFER

$$G^l_{cd} = \frac{\sum_{ij} F^l_{ijc}(x) F^l_{ijd}(x)}{IJ}$$

- Takes the raw image as input pixels and builds an internal representation that converts these pixels into an understanding of the features present within the image. The CNN is able to capture the invariances and defining features within classes that are agnostic to background noise and other nuisances. The model serves as a complex feature extractor.
- The style of an image can be described by the correlation across the feature maps. Calculating a Gram matrix that takes the outer product of the feature vector with itself at each location and averaging that outer product over all locations.
- Style transfer algorithm: calculating the mean square error for your image’s output relative to each tager, then take the weighted sum of these losses.

METHODS

Uses deep learning to compose one image in the style of another image. This is also known as neural style transfer.

ORIGINAL-STYLE

NEURAL STYLE TRANSFER

Layer activations represent low-level features (edges + textures)

INPUT LAYER

Pretrained image classification network

INTERMEDIATE

Gets the content and style representations of the image.

FINAL LAYERS

Higher-level features (wheels, eyes, ...). Using VGG19 network architecture

METHODS

Uses deep learning to compose one image in the style of another image. This is also known as neural style transfer.

MODERN

FAST STYLE TRANSFER FOR ARBITRARY STYLES

This modern approach extends the model to train on more than 32 styles and perform stylizations for unseen painting styles never previously observed. The degree to which the network generalizes to unseen painting styles measures the degree to which the network (and embedding space) represents the true breadth and diversity of all painting styles. In this work, the proposed extension of the style prediction network, takes as input an arbitrary style image and predicts the embedding vector of normalization constants. The crucial advantage of this approach is that the model can generalize unseen style image by predicting its proper style embedding at test time

EXPERIMENTS

DATA

30 CONTENT IMAGES

We chose content images of human faces with various skin tones and emotions.

03 STYLE REFERENCE IMAGES

We chose the following artists for our style reference: Steven Harrington, Georgia O’Keeffe, and Van Gogh

FEATURE COMPARISONS

HIGH-LEVEL

Objects, shapes, wheels, eyes …

LOW-LEVEL

Pixel intensities, colors, textures, lines, edges, …

- (MSE) Mean Squared Error
- (SSIM) Structural Similarity Index

Estimating perceived errors

→ Launch Jupyter Notebook on Google Colab

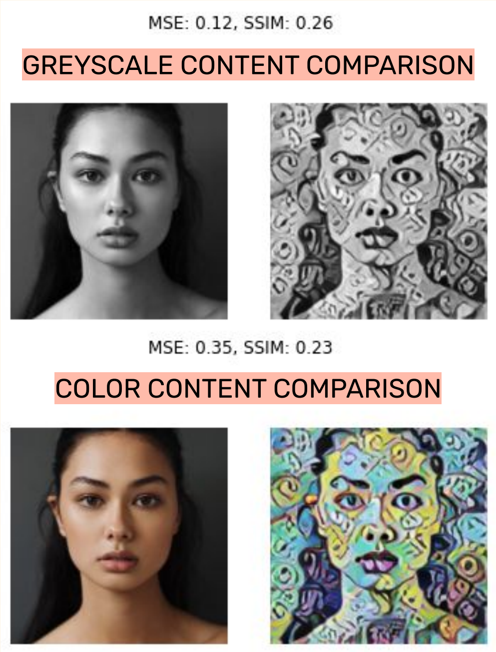
Python Compare Two Images

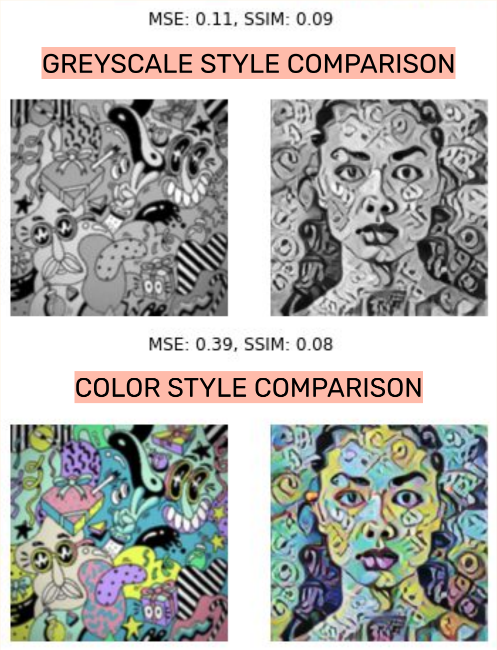
```python
def mse(imageA, imageB):
    # the 'Mean Squared Error' between the two images is the
    # sum of the squared difference between the two images;
    # NOTE: the two images must have the same dimension
    err = np.sum((imageA.astype("float") - imageB.astype("float")) ** 2)
    err /= float(imageA.shape[0] * imageA.shape[1])

    # return the MSE, the lower the error, the more "similar"
    # the two images are
    return err
```

We use SSIM to remedy some of the issues associated with MSE

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

FEATURE COMPARISONS





RESULTS: STEVEN HARRINGTON

		SKIN TONE 1 (DARKEST)		SKIN TONE 2 (MEDIUM)		SKIN TONE 3 (LIGHTEST)	
	AVERAGE	♂ MALE	♀ FEMALE	♂ MALE	♀ FEMALE	♂ MALE	♀ FEMALE
NEUTRAL	MSE	0.242	0.221	0.197	0.243	0.195	0.289
	SSIM	0.193	0.185	0.199	0.168	0.209	0.179
SMILING	MSE	0.246	0.3	0.281	0.198	0.274	0.254
	SSIM	0.191	0.16	0.2	0.208	0.199	0.2
ANGRY	MSE	0.28	0.261	0.237	0.192	0.233	0.178
	SSIM	0.166	0.2	0.217	0.188	0.199	0.233
SAD	MSE	0.206	0.215	0.193	0.211	0.277	0.262
	SSIM	0.219	0.213	0.212	0.201	0.192	0.204
PROFILE	MSE	0.207	0.238	0.179	0.199	0.246	0.297
	SSIM	0.188	0.188	0.186	0.186	0.19	0.181

MOST SIMILAR

LEAST SIMILAR

Images will have higher similarities when they have a low MSE and a high SSIM.

RESULTS: STEVEN HARRINGTON

LEAST

SIMILARITIES BETWEEN TESTED CONTENT IMAGES AND REFERENCE STYLE IMAGES

MOST

RESULTS: GEORGIA O'KEEFFE

		SKIN TONE 1 (DARKEST)		SKIN TONE 2 (MEDIUM)		SKIN TONE 3 (LIGHTEST)	
	AVERAGE	♂ MALE	♀ FEMALE	♂ MALE	♀ FEMALE	♂ MALE	♀ FEMALE
NEUTRAL	MSE	0.211	0.198	0.175	0.217	0.148	0.247
	SSIM	0.433	0.385	0.438	0.399	0.451	0.433
SMILING	MSE	0.238	0.245	0.255	0.185	0.238	0.243
	SSIM	0.405	0.377	0.425	0.403	0.431	0.415
ANGRY	MSE	0.242	0.21	0.223	0.159	0.203	0.146
	SSIM	0.34	0.412	0.411	0.441	0.408	0.451
SAD	MSE	0.161	0.183	0.168	0.179	0.257	0.244
	SSIM	0.434	0.432	0.452	0.436	0.43	0.407
PROFILE	MSE	0.159	0.21	0.168	0.171	0.24	0.281
	SSIM	0.451	0.374	0.423	0.427	0.411	0.415

MOST SIMILAR

LEAST SIMILAR

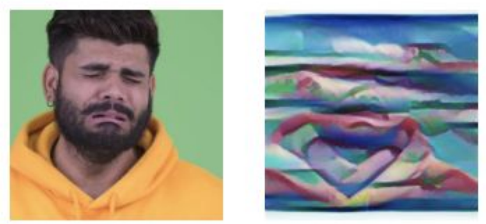
Images will have higher similarities when they have a low MSE and a high SSIM.

RESULTS: GEORGIA O'KEEFFE

LEAST

SIMILARITIES BETWEEN TESTED CONTENT IMAGES AND REFERENCE STYLE IMAGES

MOST



RESULTS: VAN GOGH

		SKIN TONE 1 (DARKEST)		SKIN TONE 2 (MEDIUM)		SKIN TONE 3 (LIGHTEST)	
	AVERAGE	♂ MALE	♀ FEMALE	♂ MALE	♀ FEMALE	♂ MALE	♀ FEMALE
NEUTRAL	MSE	0.195	0.174	0.157	0.191	0.14	0.295
	SSIM	0.322	0.308	0.317	0.28	0.349	0.284
SMILING	MSE	0.291	0.275	0.286	0.18	0.267	0.278
	SSIM	0.28	0.252	0.309	0.312	0.316	0.295
ANGRY	MSE	0.205	0.256	0.227	0.192	0.184	0.162
	SSIM	0.259	0.302	0.341	0.295	0.314	0.349
SAD	MSE	0.145	0.177	0.19	0.168	0.287	0.265
	SSIM	0.368	0.336	0.32	0.318	0.298	0.314
PROFILE	MSE	0.175	0.203	0.173	0.181	0.252	0.316
	SSIM	0.301	0.283	0.307	0.309	0.302	0.289

MOST SIMILAR

LEAST SIMILAR

Images will have higher similarities when they have a low MSE and a high SSIM.

RESULTS: VAN GOGH

LEAST

MOST

SIMILARITIES BETWEEN TESTED CONTENT IMAGES AND REFERENCE STYLE IMAGES

DISCUSSION

We believe our method was successful due to the following reasons:

- COLOR Our methods of measuring similarity averages the values from the colorful style images (mid range rgb values), which results in color values close to grey and green, and least similar to white, which is the end of the range of color values.

- TEXTURE : For the Van Gogh style, which had less variation in color, but a more consistent pattern and texture, the best matches were also images that had green and grey backgrounds, though ones with texture, rather than a smooth gradient like the previous ones. This makes sense, considering the style image was very textured. This shows the algorithm, as well as our method for measuring similarity, works better (in terms of transferring style while also maintaining the content) on similar textured backgrounds to the style of the image.

CONCLUSION

In conclusion, the *Fast Style Transfer* algorithm work relatively well on images with grey and green backgrounds. As well, styles tend to transfer better on content images with similar textures and contours to itself. This dominated the likelihood of similarity, as we did not notice any large disparities in across emotions and skin color.

JUNE 15, 2021 - 15H

FINAL PROJECT: STYLE TRANSFER

THANKS!

DO YOU HAVE ANY QUESTIONS?

Isabella Rocha
Skyler Gauuan
Benjamin Reichelt

- 18H

JUNE 15, 2021 - 15H

AUGUST 8, 2021 - 16H