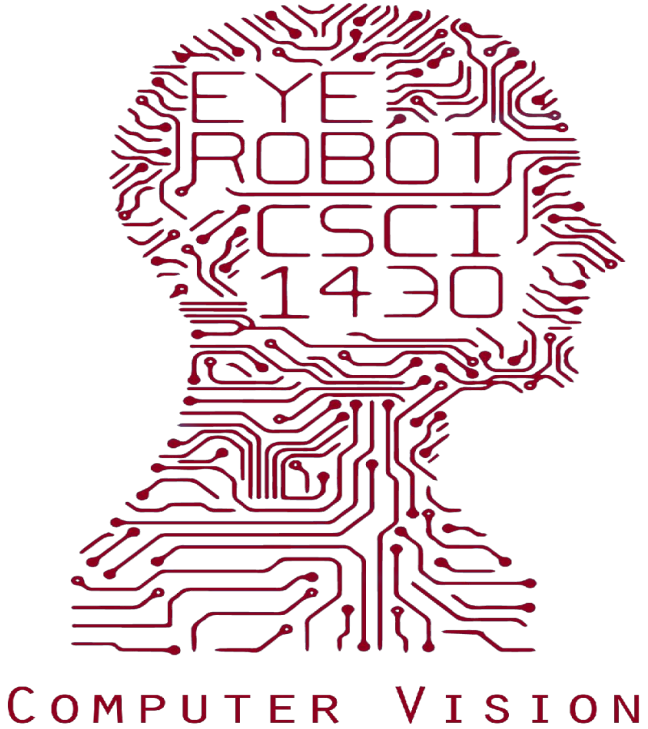
Neural Canvas

Aarav, Angel, Brian, Sophia



Neural Style Transfer

Image source:
https://research.adobe.com/news/image-stylization-history-and-future-part-3/

Optimizes image to have the content of a content image while also looking like the style image using the representations from a pre trained CNN image encoder. VGG 19 model was chosen.

Modified Neural Style Transfer

We wanted to be able to also combine the styles of multiple images.

Traditional neural style transfer: 1 style image + 1 content image

$$\mathbf{Loss_{Total} = Loss_{Style} + Loss_{Content}}$$

Modified neural style transfer: N style images + 1 content image

$$\mathbf{Loss_{Total} = Loss_{Style\ 1} + Loss_{Style\ 2} + \ldots + Loss_{Style\ N} + Loss_{Content}}$$

Result is image that is blend of style images.

Dataset

What images should we use as content images?

Met Collection API

- Gives access to the Met museum artwork without any restrictions
- Downloaded ~400 “highlighted” paintings

CLIP Image Retrieval

OpenAI CLIP model consists of an image encoder and text encoder and was trained such that embeddings of similar images/text are also similar to each other

~400 images

…

CLIP Image Encoder

Image embedding 1

Image embedding 2

…

Input queries = “Van Gogh Style”, “Impressionist”

CLIP Text Encoder

Van Gogh Embedding

Impressionist Embedding

Retrieve image embeddings most similar to the text embeddings and pass corresponding images to neural style transfer model

Making it faster

VGG model takes ~48 seconds

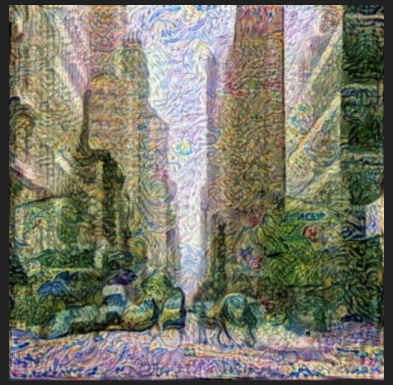
Improvements:

1. Quantization: Convert model weights/activations from 32 bit to 8 bit → ~47 seconds
2. Pruning: Removed filters based on l1 norm → ~44 seconds
3. Quantization + Pruning → ~43 seconds

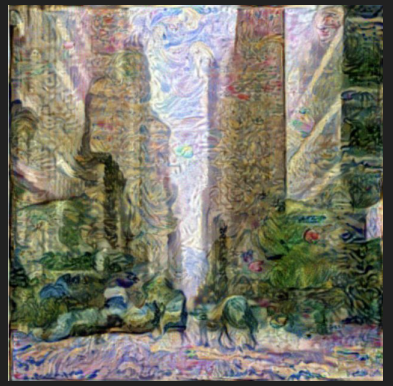
Changed image encoder

4. Efficient Net → ~24 seconds
5. MobileNet → ~11 seconds

Queries = “Van Gogh”, “Rainbow”



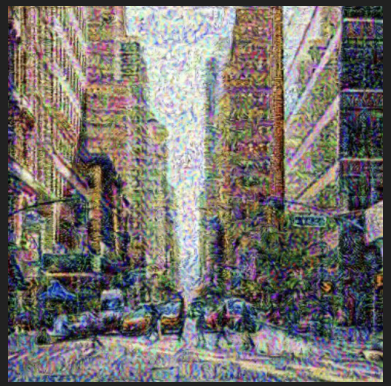
Original VGG



Quantization

Pruning

Pruning + Quantize

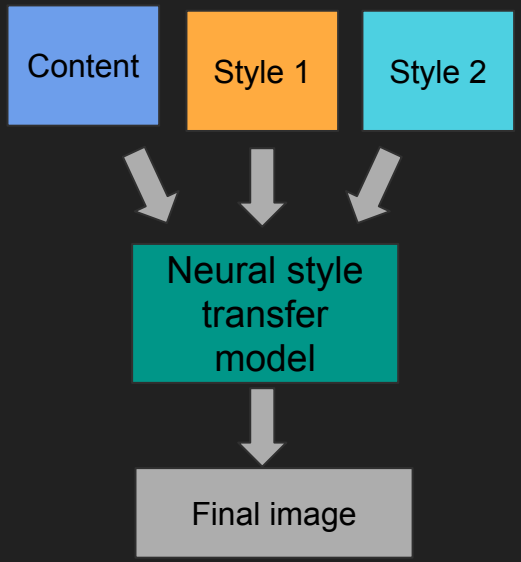


Mobile Net

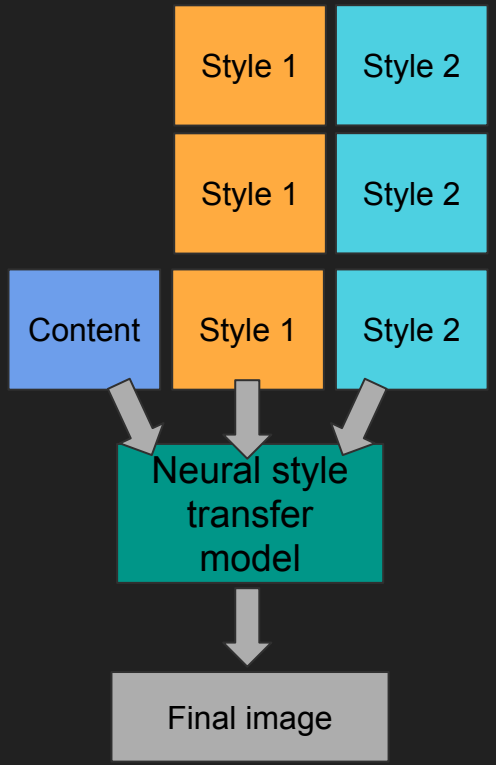
Efficient Net

Quality improvements

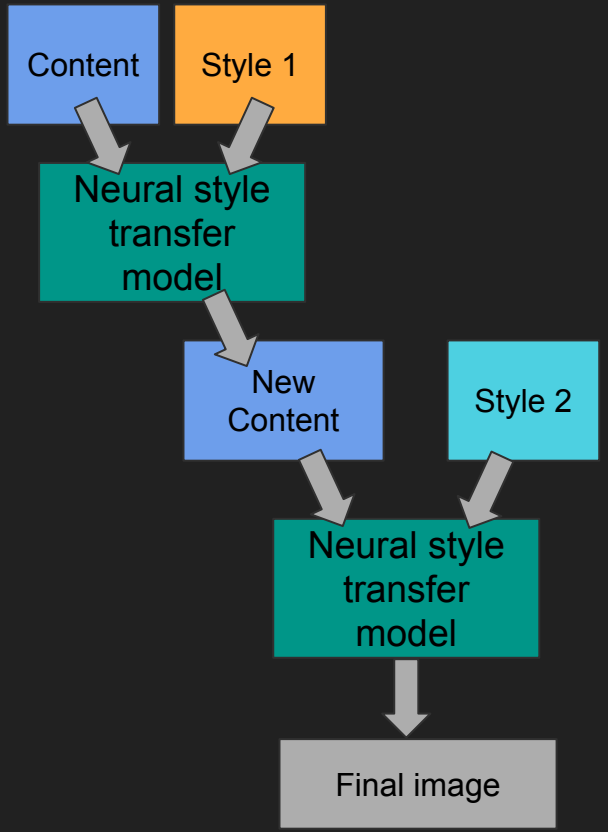
Individual
One image per style



Weighted
Multiple images per style



Chaining outputs



Results for different implementations

Text input: “Van gogh swirls”, and “Impressionistic Colors”

Content Image

Credit: Beth Garrabrant

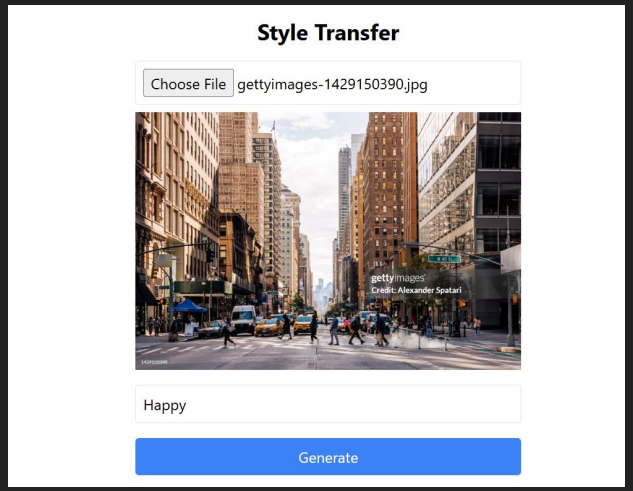
Individual

Weighted

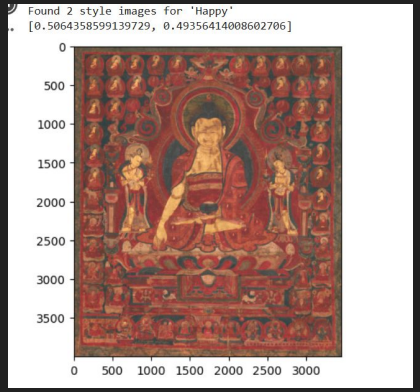
Chained

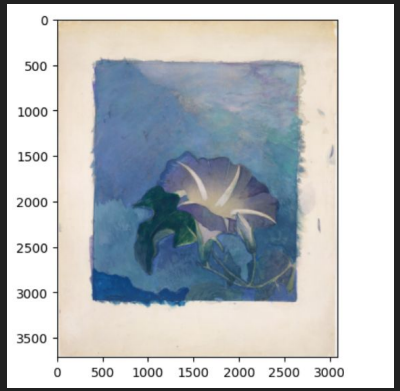
Web App

Content Image + Style Prompt



Styles





Result