

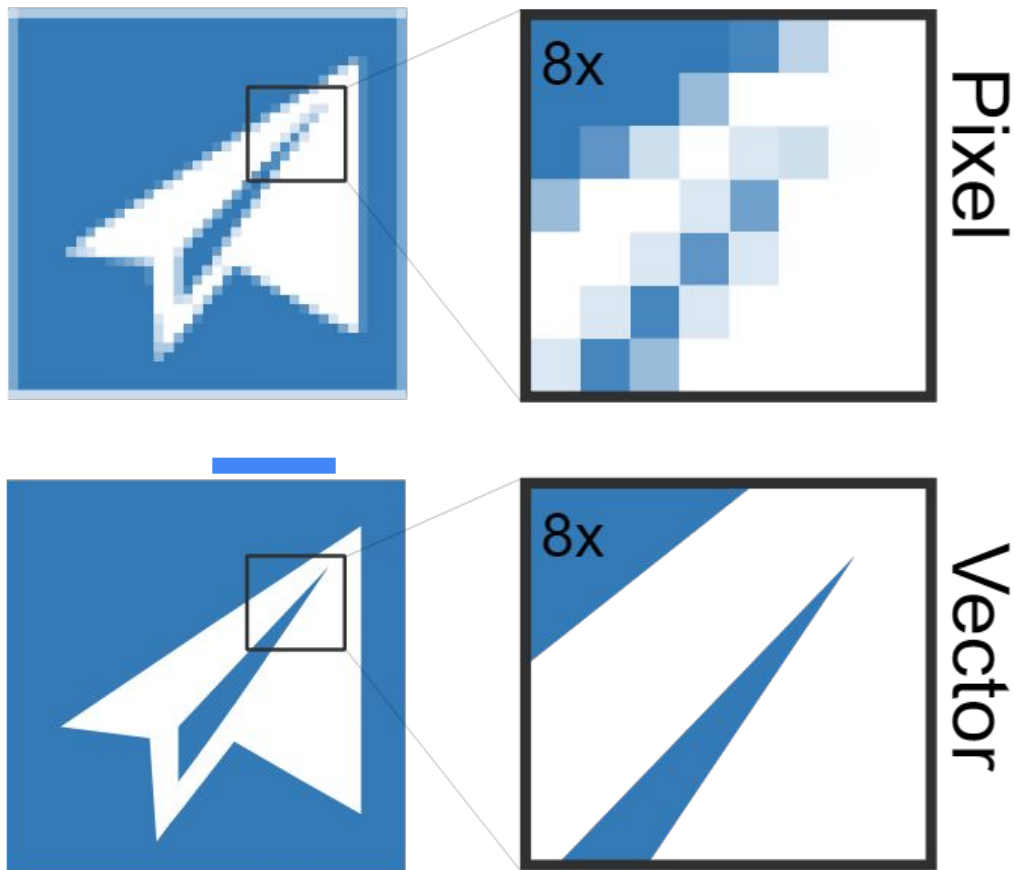
Image Vectorization

for Architecture

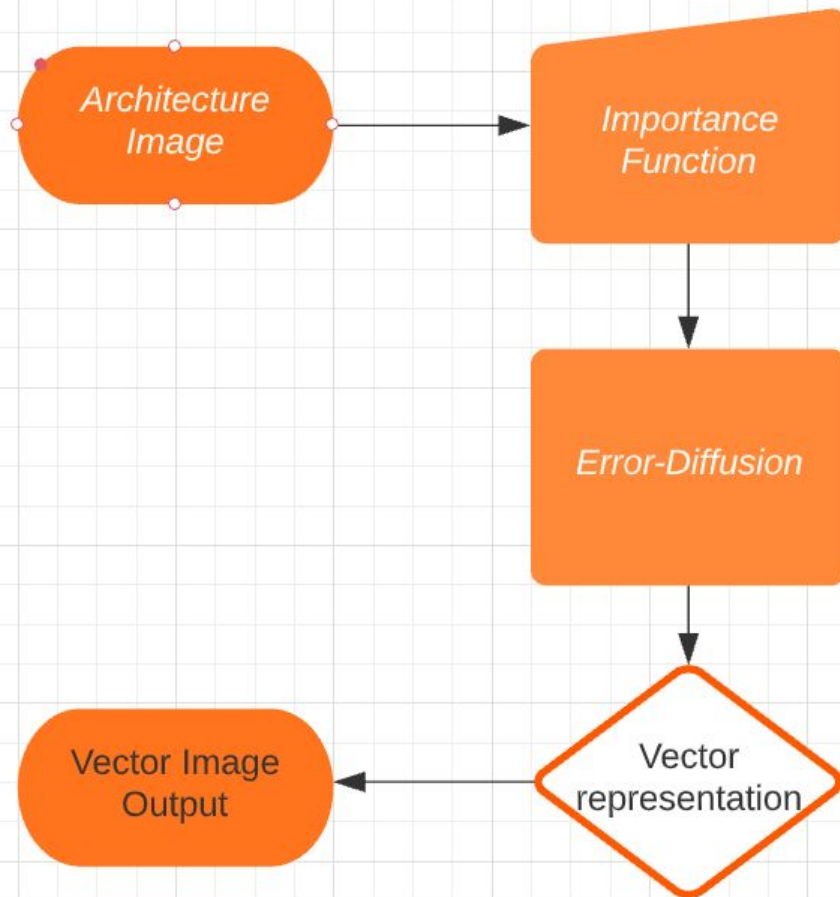


Derek Lee, Jon Lam, Victor Zhang

Y Vector



Structure Overview



Importance Matrix

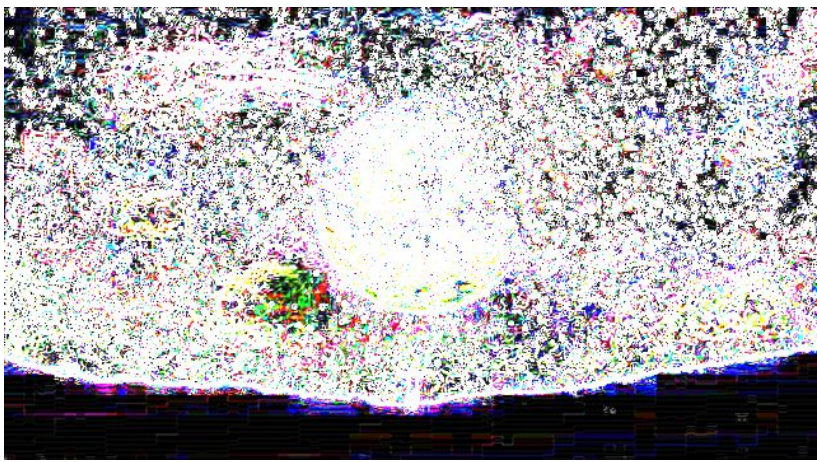
-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

2	1	0
1	0	-1
0	-1	-2

0	1	2
-1	0	1
-2	-1	0

$$F(x) = \left(\frac{x}{\max} \right)^{\frac{1}{\gamma}} * 255, \quad \gamma > 0, x \in [0, \max] \text{ and } x \in \mathbb{Z}^+,$$



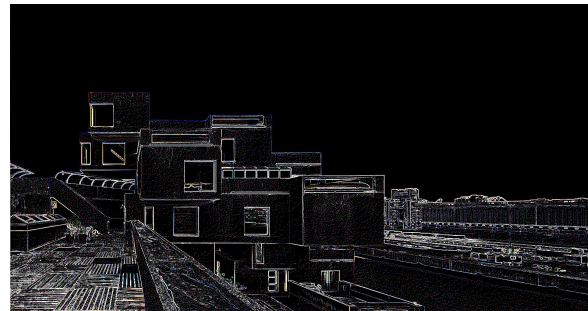
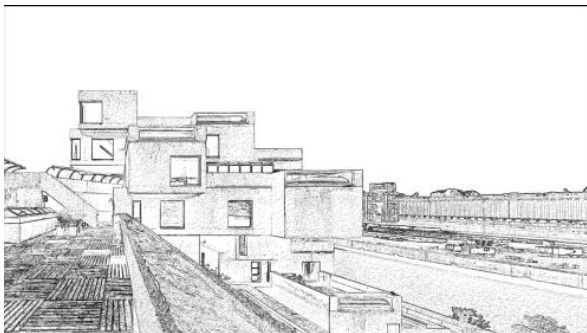
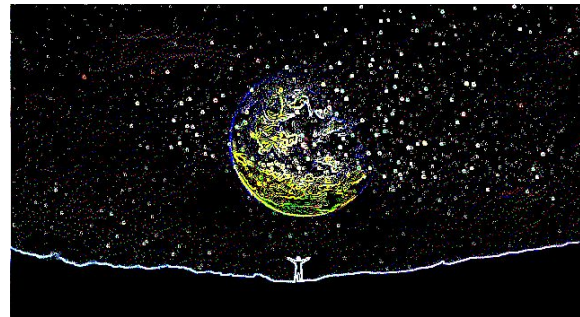
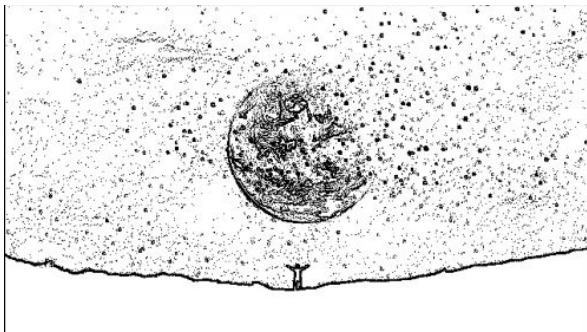
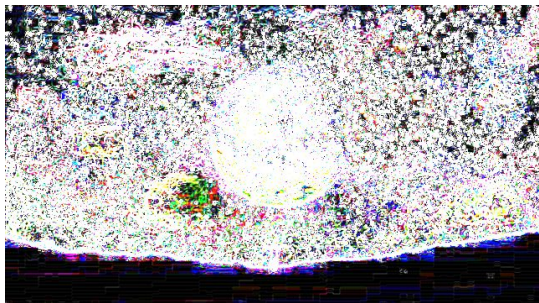
Error-Diffusion

Floyd–Steinberg dithering

$$\begin{bmatrix} & & * & \frac{7}{16} & \cdots \\ \cdots & \frac{3}{16} & \frac{5}{16} & \frac{1}{16} & \cdots \end{bmatrix}$$

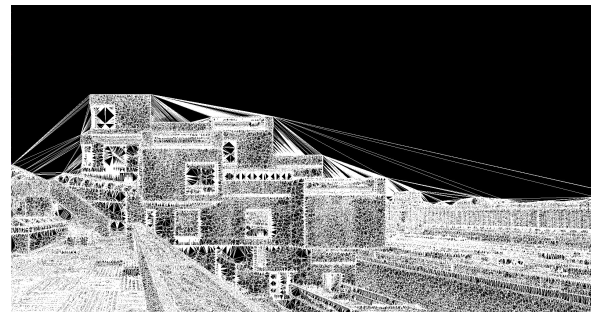
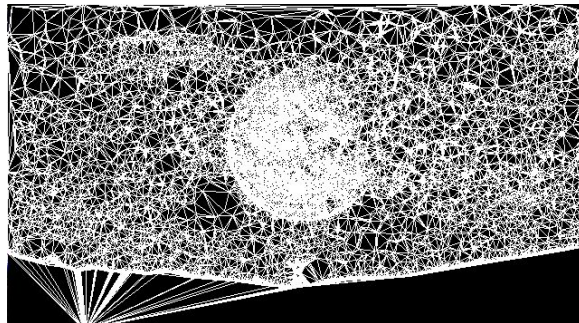
```
for each y from top to bottom do
  for each x from left to right do
    oldpixel := pixels[x][y]
    newpixel := find_closest_palette_color(oldpixel)
    pixels[x][y] := newpixel
    quant_error := oldpixel - newpixel
    pixels[x + 1][y] := pixels[x + 1][y] + quant_error × 7 / 16
    pixels[x - 1][y + 1] := pixels[x - 1][y + 1] + quant_error × 3 / 16
    pixels[x][y + 1] := pixels[x][y + 1] + quant_error × 5 / 16
    pixels[x + 1][y + 1] := pixels[x + 1][y + 1] + quant_error × 1 / 16
```


Importance -> Point Cloud (Threshold) -> Sampled Points



Vectorization

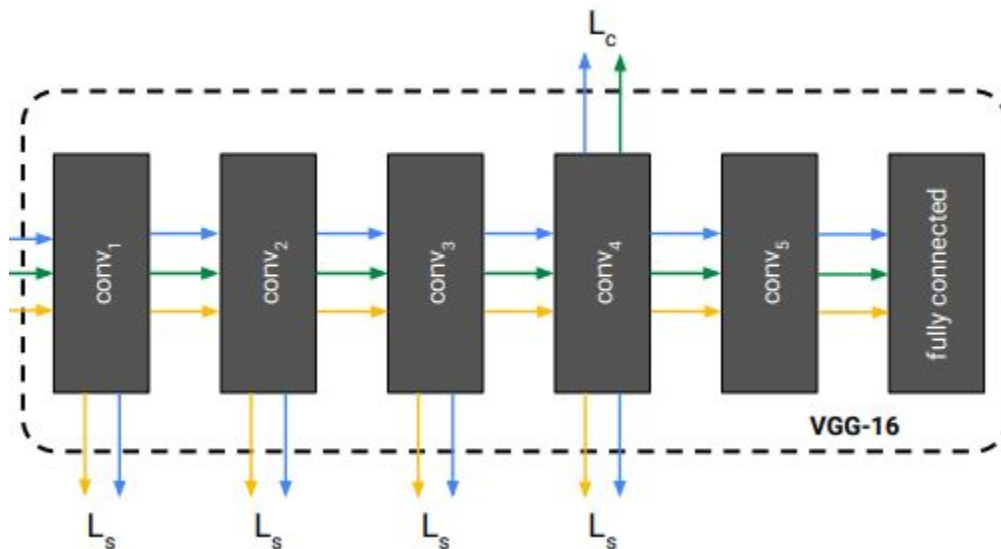
Current Implementation: Triangulation of Point cloud



Loss Evaluation

Currently using content loss with a VGG-19

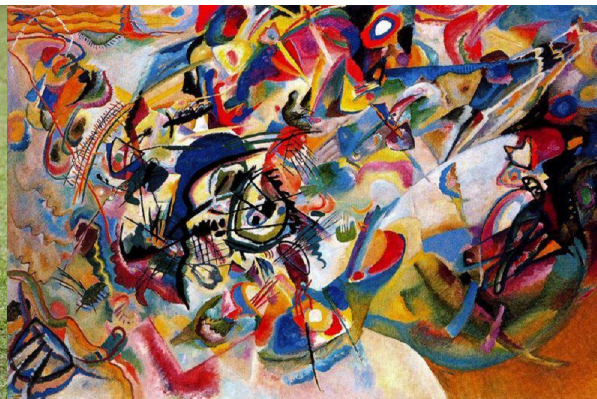
Euclidean distance between the intermediate representations of a trained classifier



Loss Evaluation



Original (Content)



Style



Combined

Image 1 & 2	Loss
Original + Combined	132352.05
Original + Style	190958.28
Style + Combined	190266.86

Future Plan

1. Implement Full SVG support
2. Mesh Simplification
3. Explore potential of NN to isolate Architecture

