COSC 499 Milestone 4

Team 9 order of aesthetics - section 003

Algorithm used to compare images

Recall: How to build a neural network

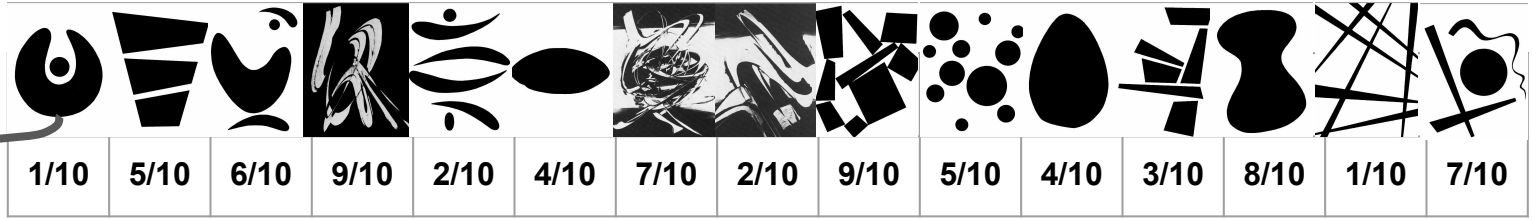
Start with unlabeled data

Recall: How to build a neural network

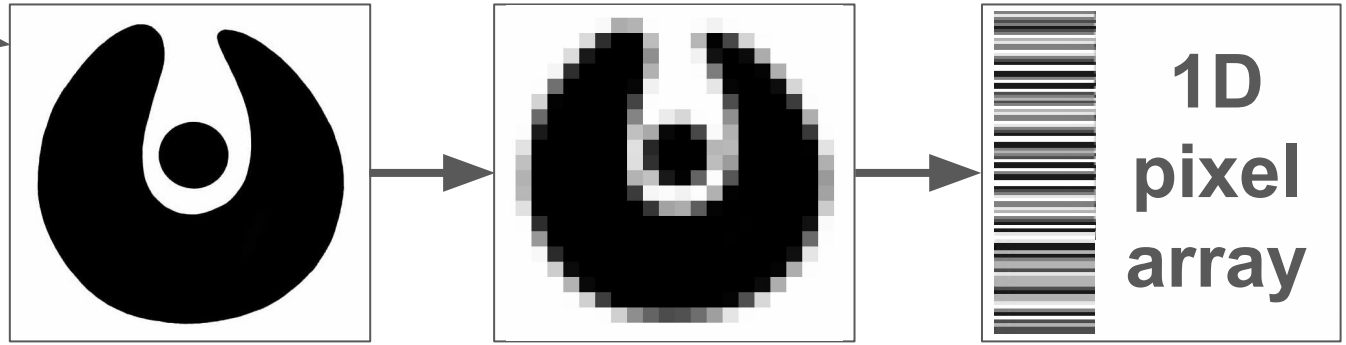
Label it using the survey in our apps

1/10	5/10	6/10	9/10	2/10	4/10	7/10	2/10	9/10	5/10	4/10	3/10	8/10	1/10	7/10

Recall: How to build a neural network

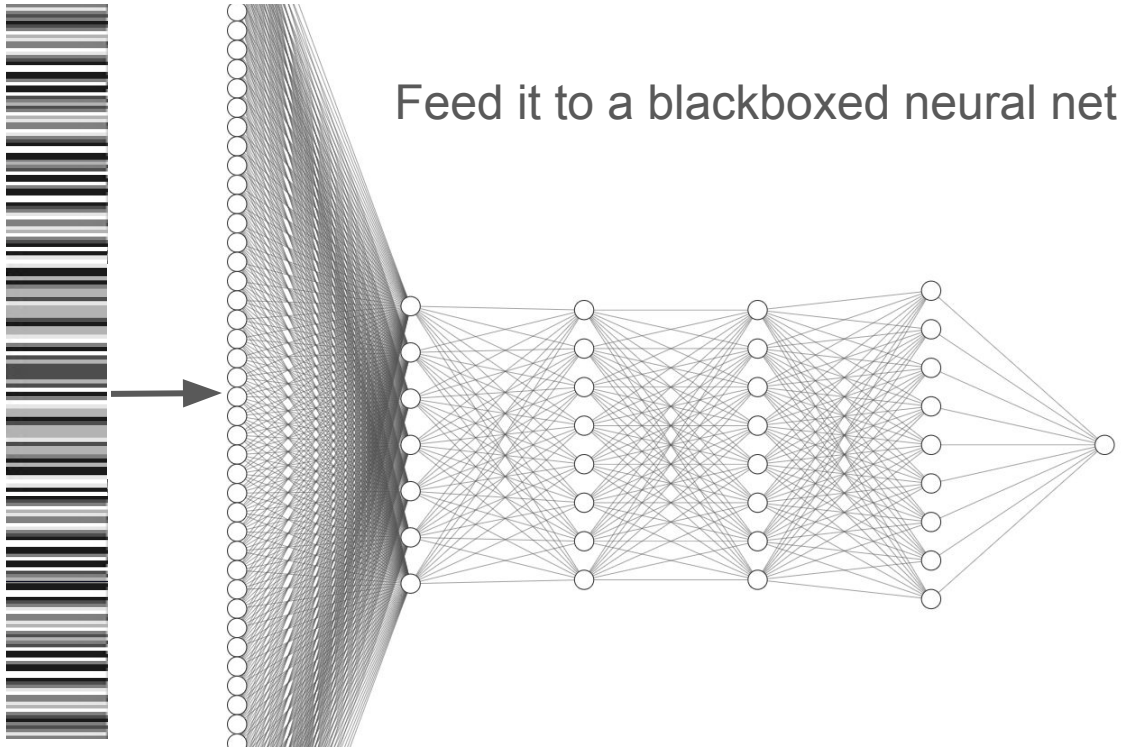


Convert each image to a set of pixels (pretraining)



Recall: How to build a neural network

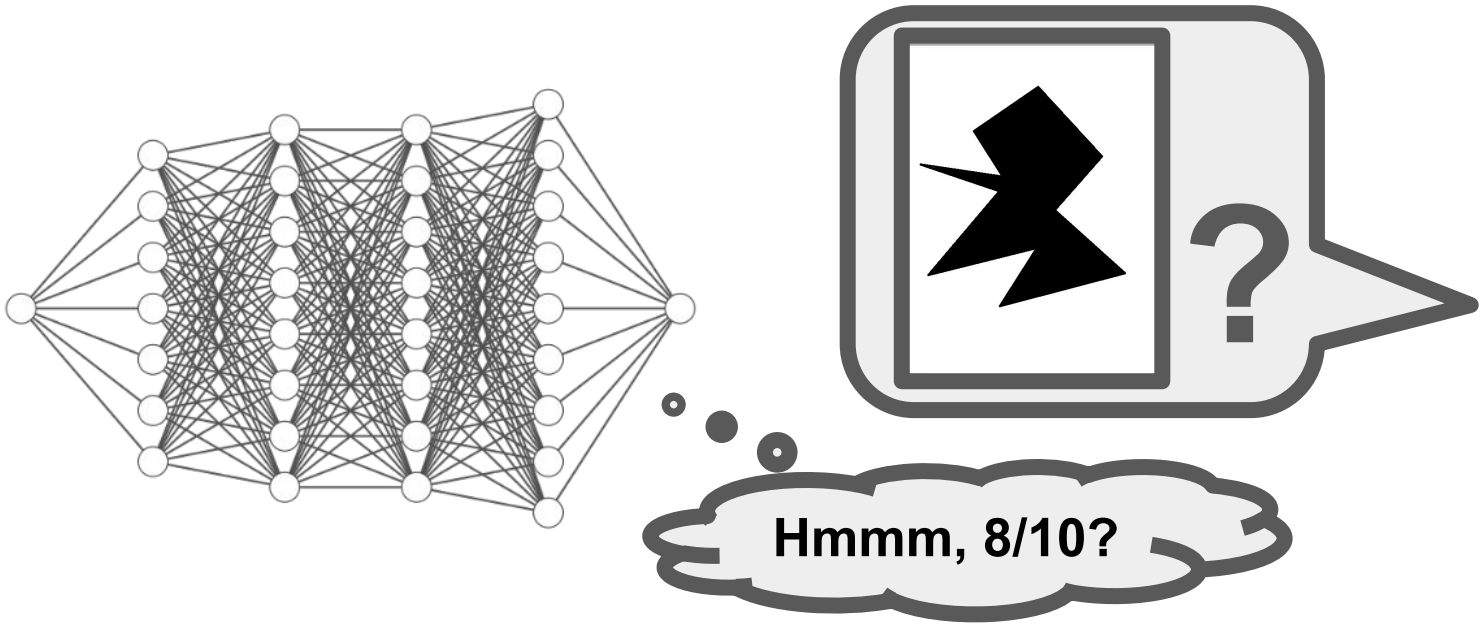
1D pixel array



Feed it to a blackboxed neural net

Recall: How a neural network works

And just like that we have a neural network that can make an educated guess as to how aesthetically pleasing an image is



Performance of image comparison

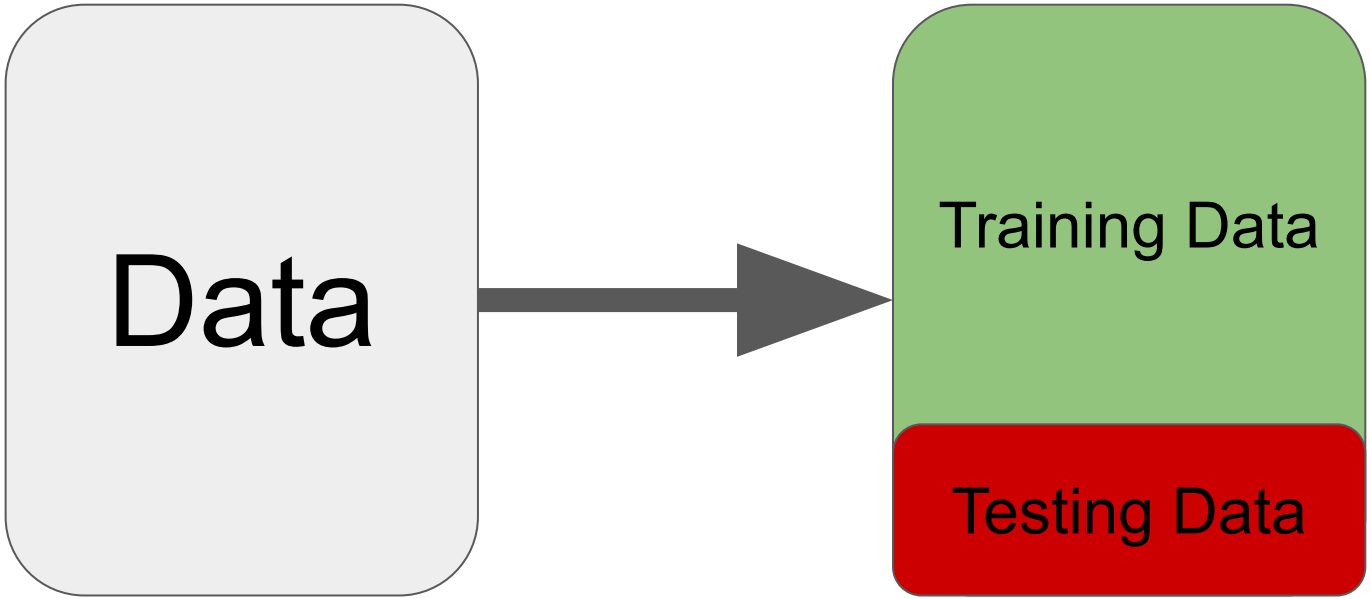
How do we measure performance?

How do we measure performance?

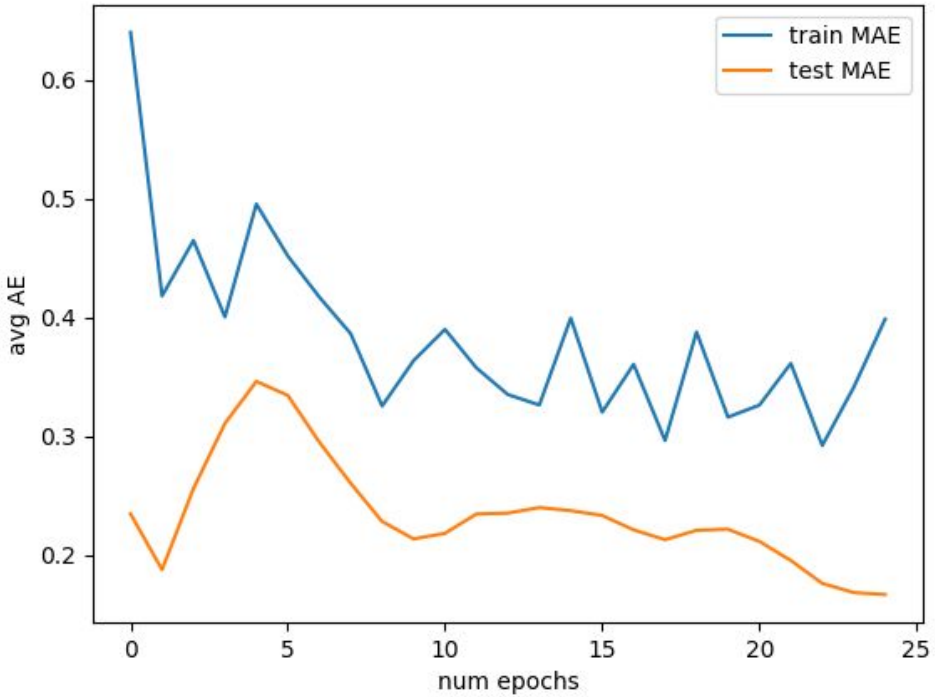
We implement a Train/Test split in our data

Train test split:

Train test split:



Train test split:



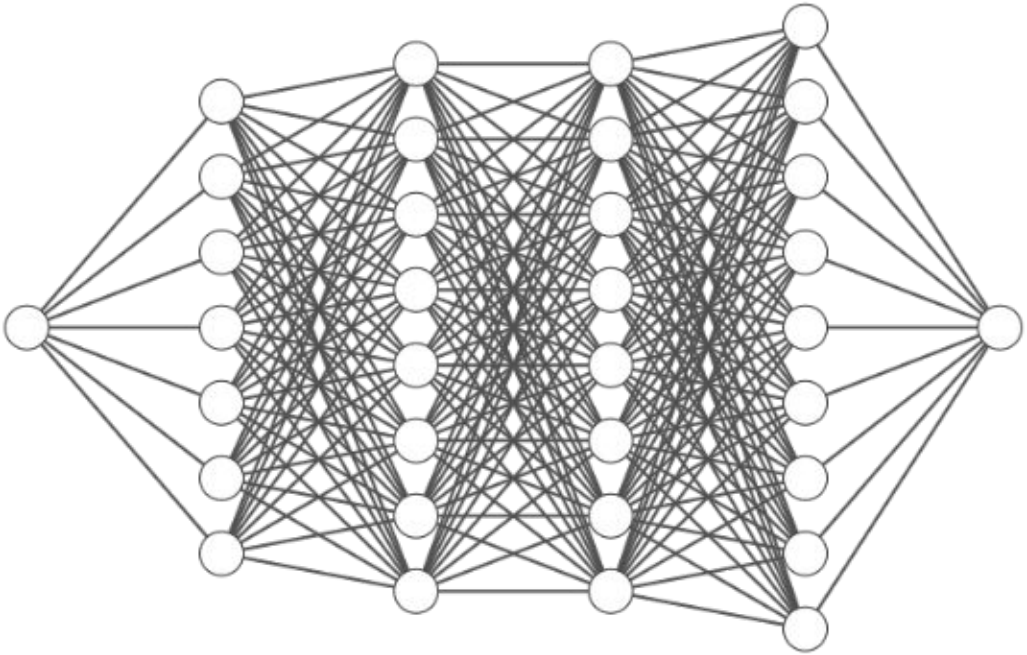
So Ultimately:

Our algorithm had an average error of 20% while trying to guess which image out of a pair was preferred by the majority of users (while guessing would be 50% +/-)

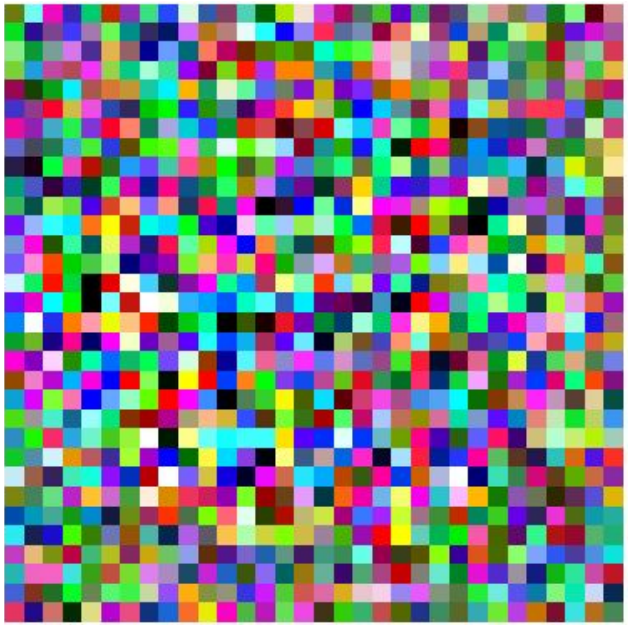
Meaning: for every 10 images, the Algorithm would align with the majority opinion for 8 of them (while guessing would have gotten 5 of them)

Algorithm used to generate images

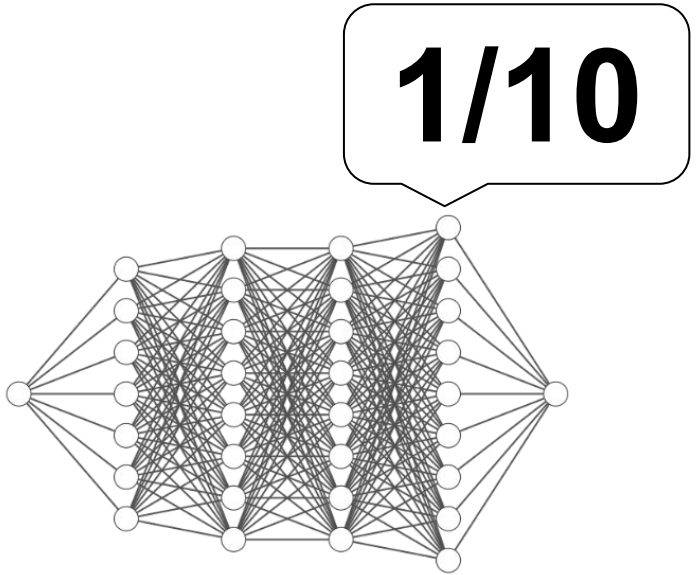
Starting out with our algorithm from earlier

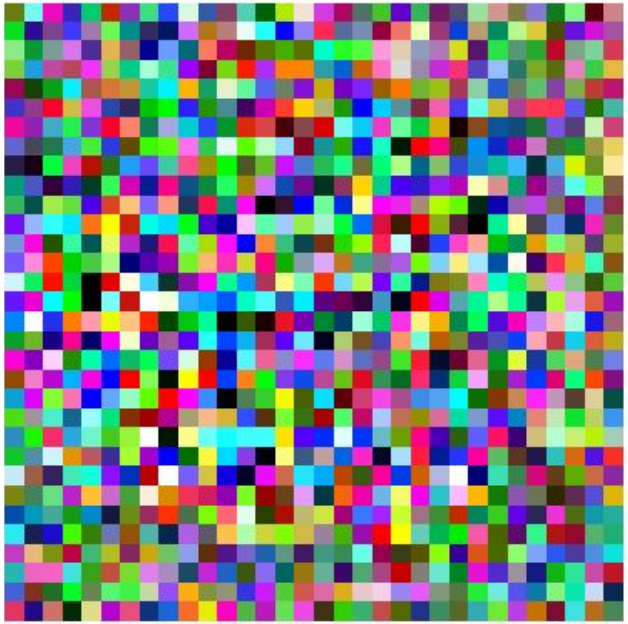


We generate an image from statistical noise

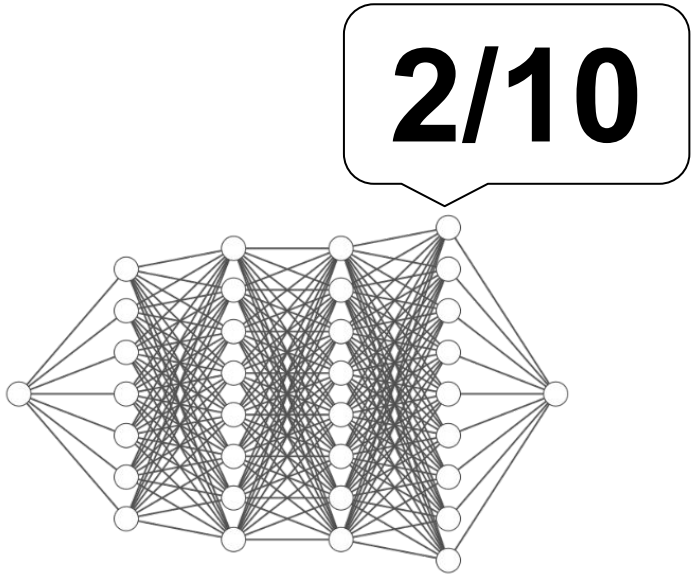


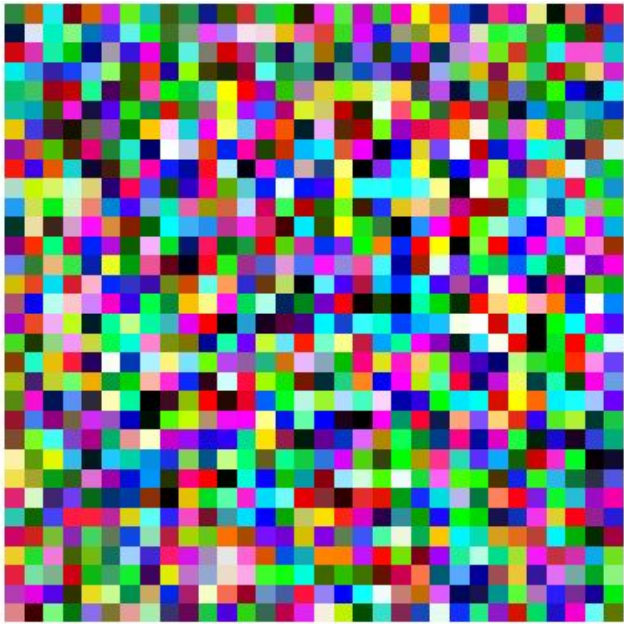
& ask our model what it thinks



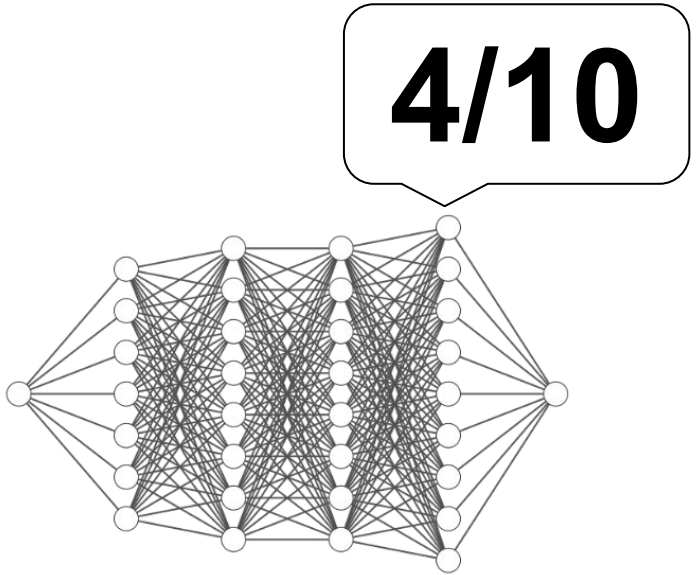


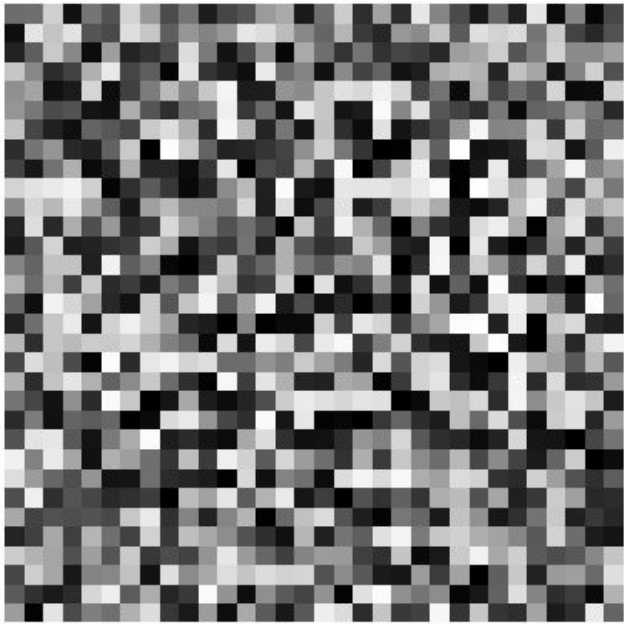
Using that we can generate a better image

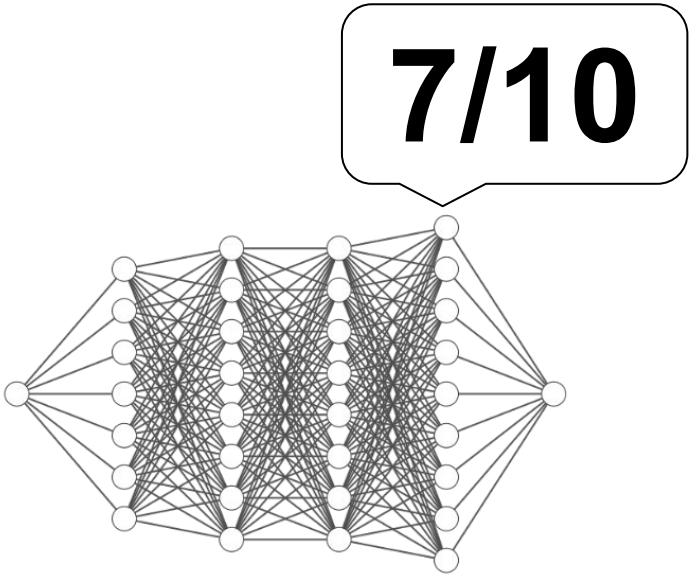


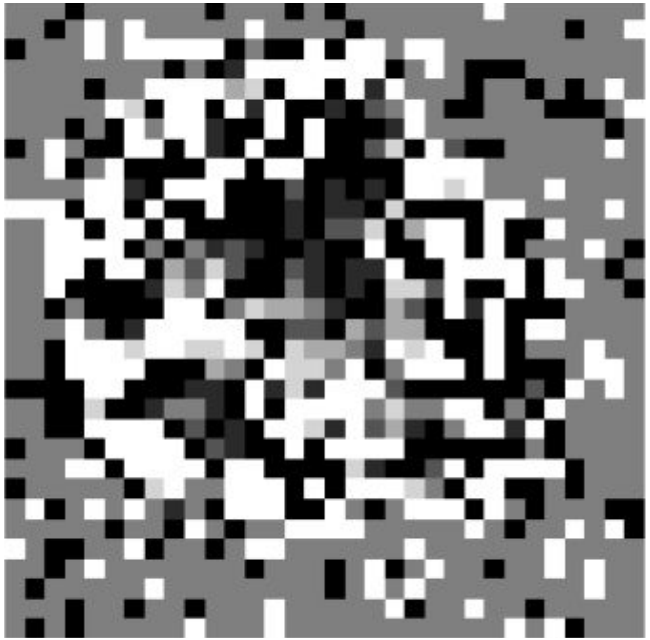


& then an even better image

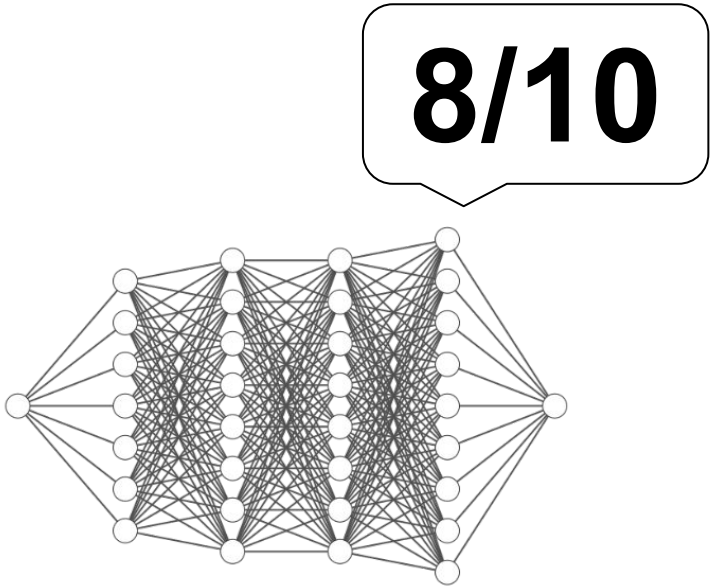


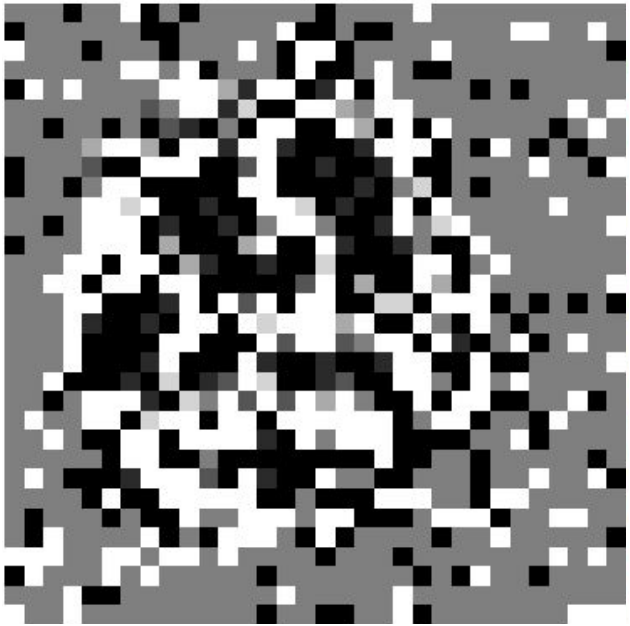



7/10



& then an even better image



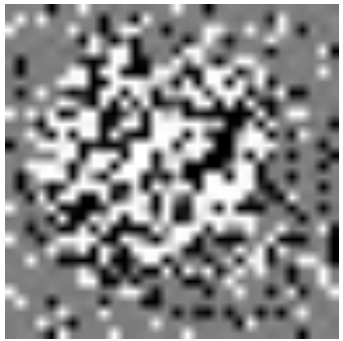


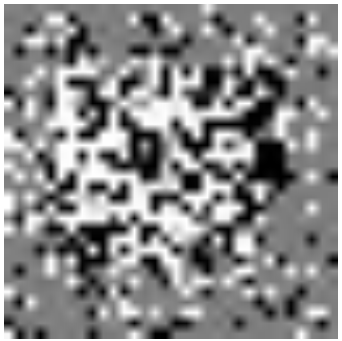
Then finally we can smooth to create a final product

Smoothing aims to take a set of raw math output,
& interprets it into an image:

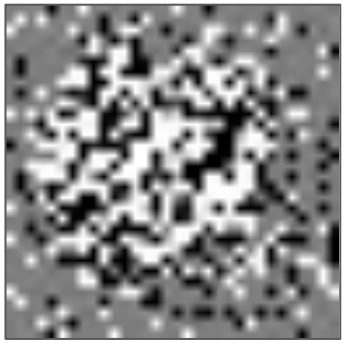
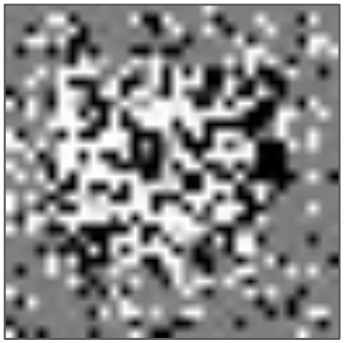
Smoothing aims to take a set of raw math output, & interprets it into an image:

Our generated images ended up looking like this raw:

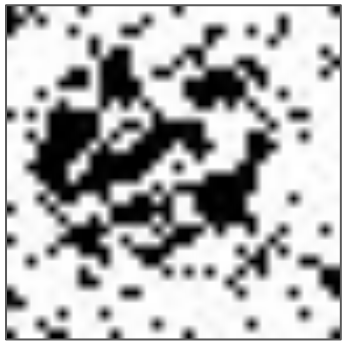
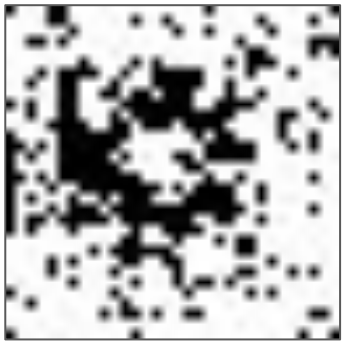
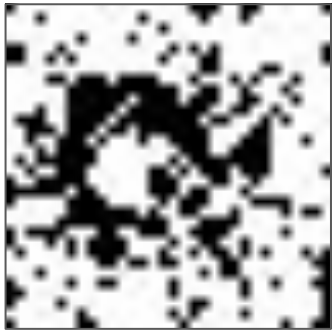




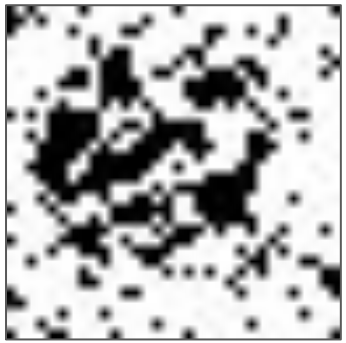
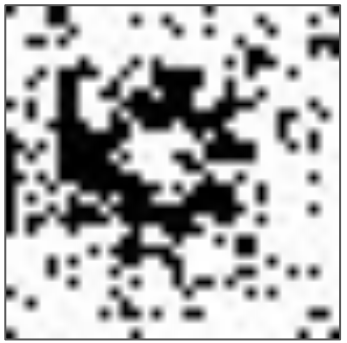
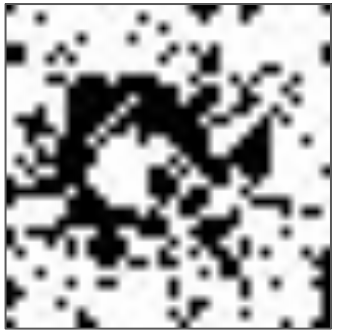
Our generated images ended up looking like this raw:


And like this after applying 4 Nearest Neighbor smoothing once:

And like this after applying 4 Nearest Neighbor smoothing once:

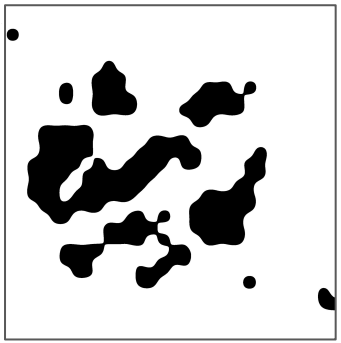
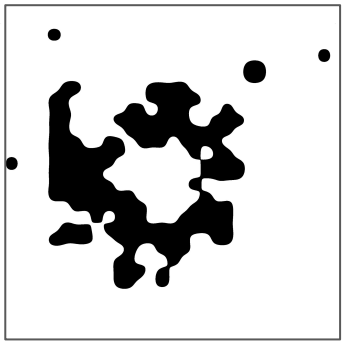
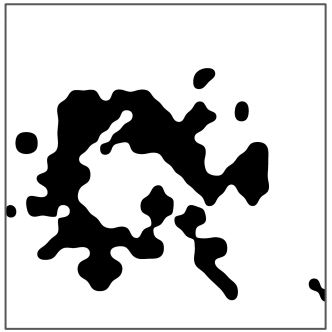
Then like this after applying Smoothing again:



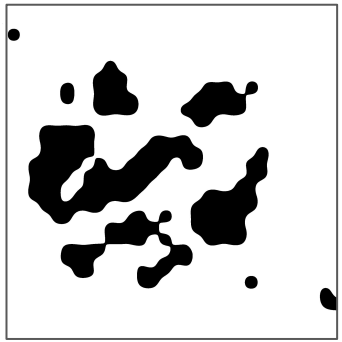
Then like this after applying Smoothing again:



Then like this after using interpolation to make HD versions:

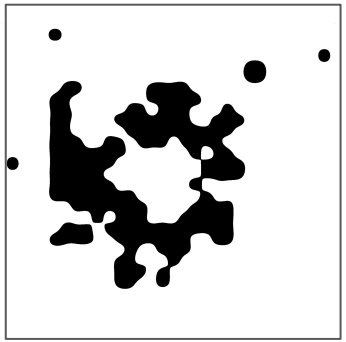




And rated by 5 people on a scale from 1-10:



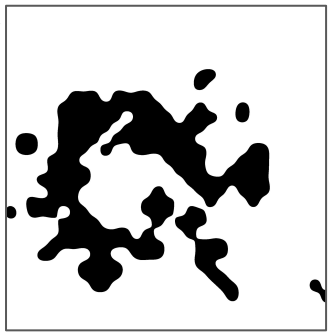
(9+8+10+7+9)/5

= 8.6/10



(7+8+9+8+7)/5

= 7.8/10



(6+8+9+5+8)/5

= 7.2/10

Therefore: the algorithms in place can make adequate but non excellent images - this serves as a proof of concept that with a sufficient pool of data a successful algorithm can generate images en mass

A note on small sample size

Thanks for listening

A note on dovetailing:

Some elements in this slideshow are copied from team member Samira Almuallim’s work in the client information session 3 & the M3 slides