

Image Classification With a Convolutional Neural Network

ANI KARENNOVNA K

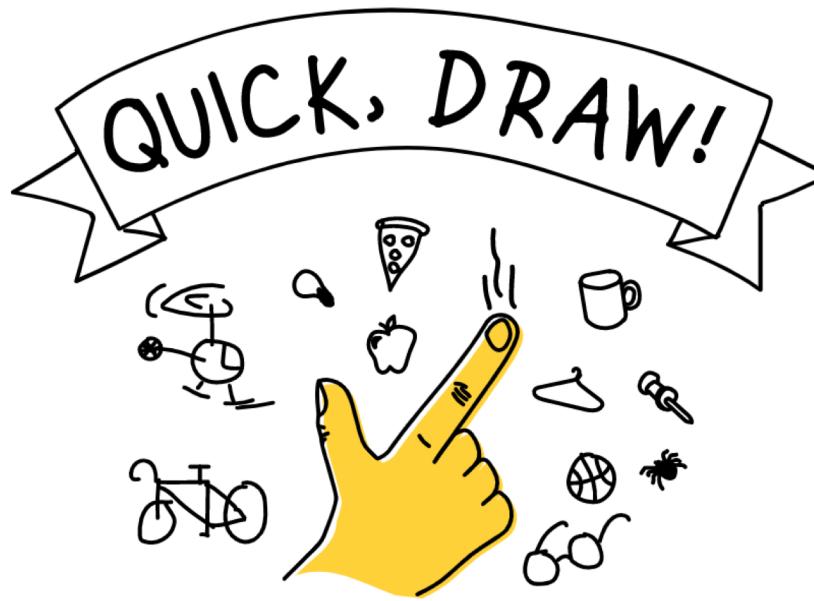
Motivation

- COMPUTER VISION
 - Facial recognition, self-driving-cars, facebook's friend suggestion algorithm
- SIMPLE TASK - TRANSFERRABLE INSIGHTS
 - Voice Recognition
- A DOOR INTO THE BLACK BOX OF DEEP LEARNING
 - More of a research twist
- AI PRODUCT DEVELOPMENT
 - Apple's Siri
 - Amazon's Alexa
 - iPhone facial recognition

The Data



The Data

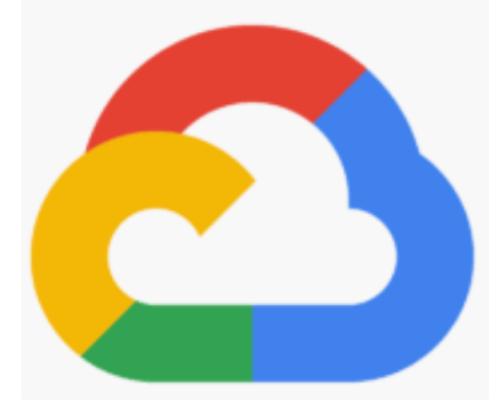


Can a neural network learn to recognize doodling?

[Click Here to Start Drawing](#)

The Data

- Google Cloud
- .ndjson files
- Each with ~12000 images with metadata
- Each image 288 X 432
- Over 300 classes



The Classes

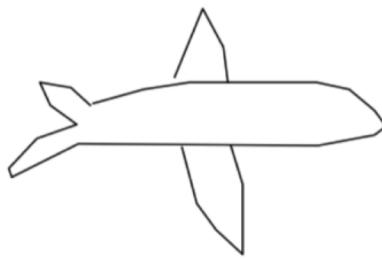
0



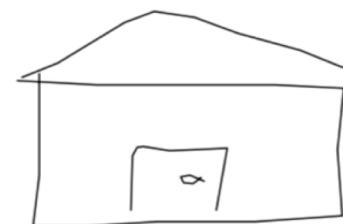
1



2



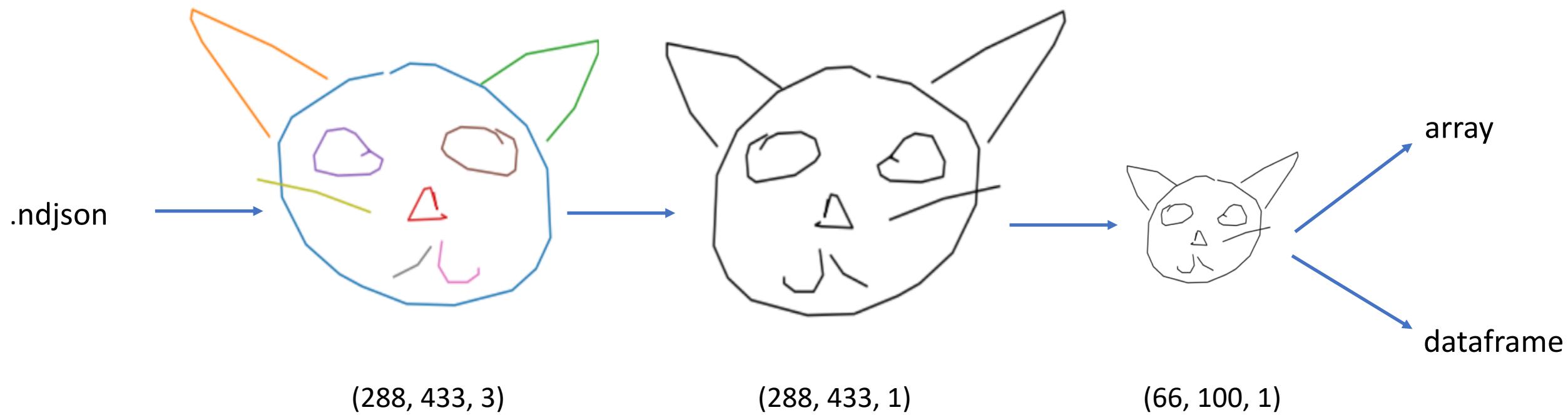
3



4



Data Preprocessing



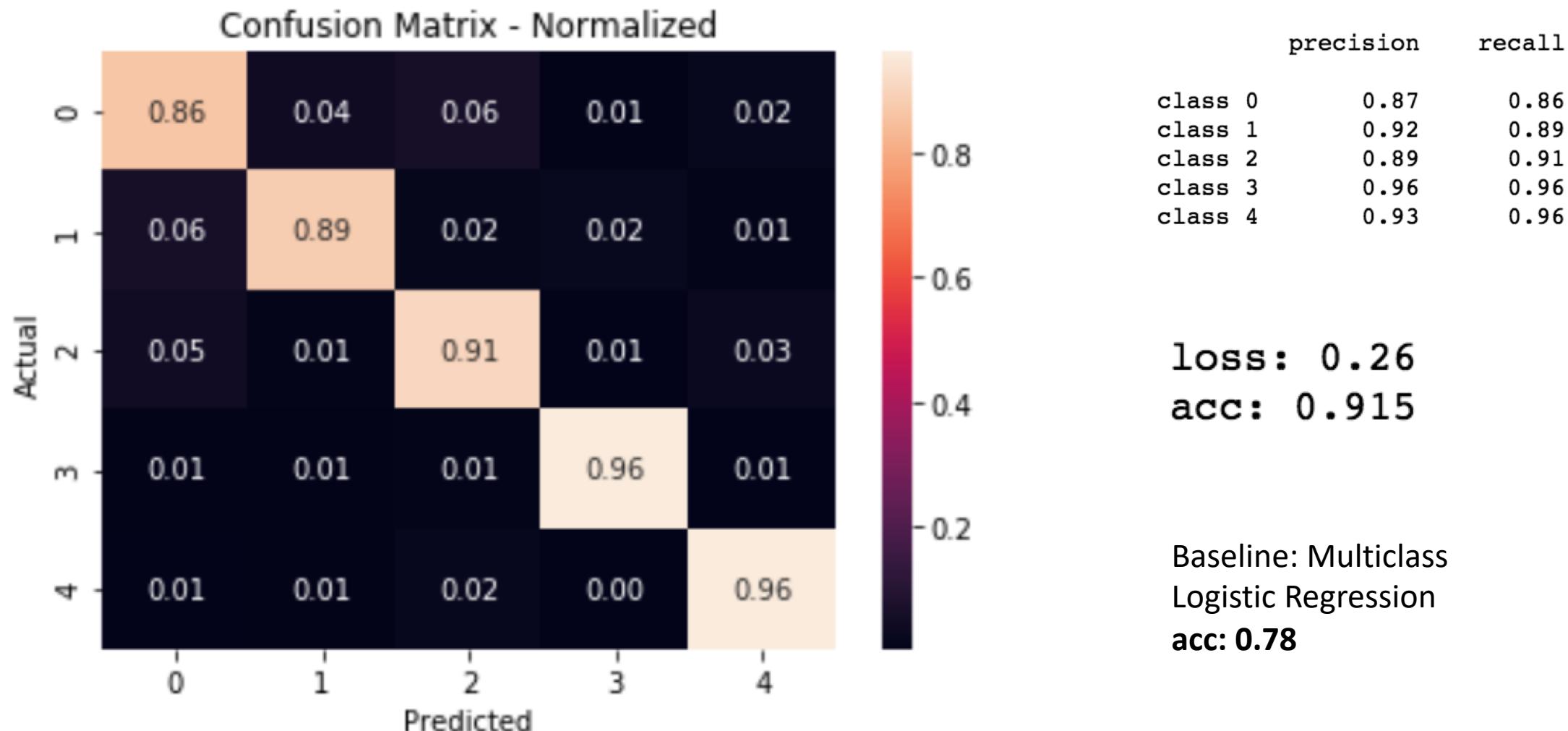
Variation in Images Within a Class



Modeling: Neural Network Architecture

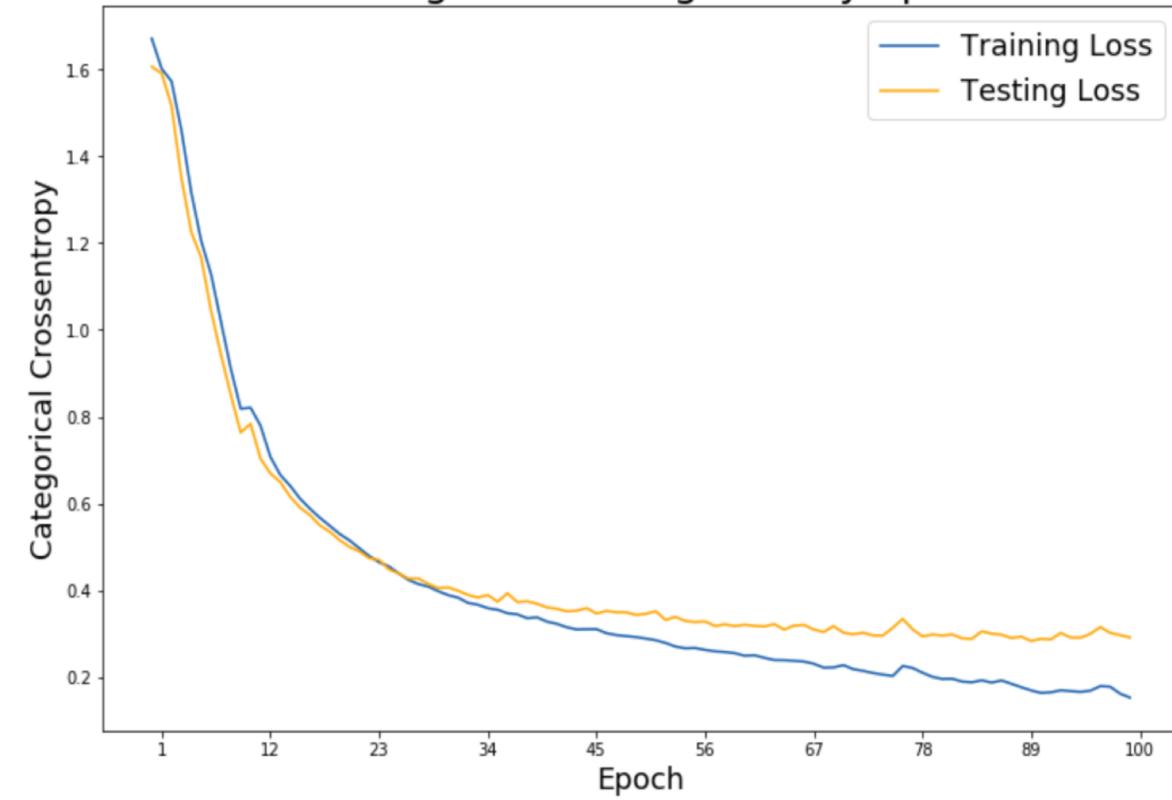
Layer (type)	Output Shape	Param #
=====		
conv2d_3 (Conv2D)	(None, 61, 95, 6)	222
max_pooling2d_3 (MaxPooling2D)	(None, 15, 23, 6)	0
conv2d_4 (Conv2D)	(None, 10, 18, 16)	3472
max_pooling2d_4 (MaxPooling2D)	(None, 2, 4, 16)	0
flatten_2 (Flatten)	(None, 128)	0
dense_4 (Dense)	(None, 3000)	387000
dense_5 (Dense)	(None, 3000)	9003000
dense_6 (Dense)	(None, 5)	15005
=====		
Total params: 9,408,699		
Trainable params: 9,408,699		
Non-trainable params: 0		

Model Evaluation

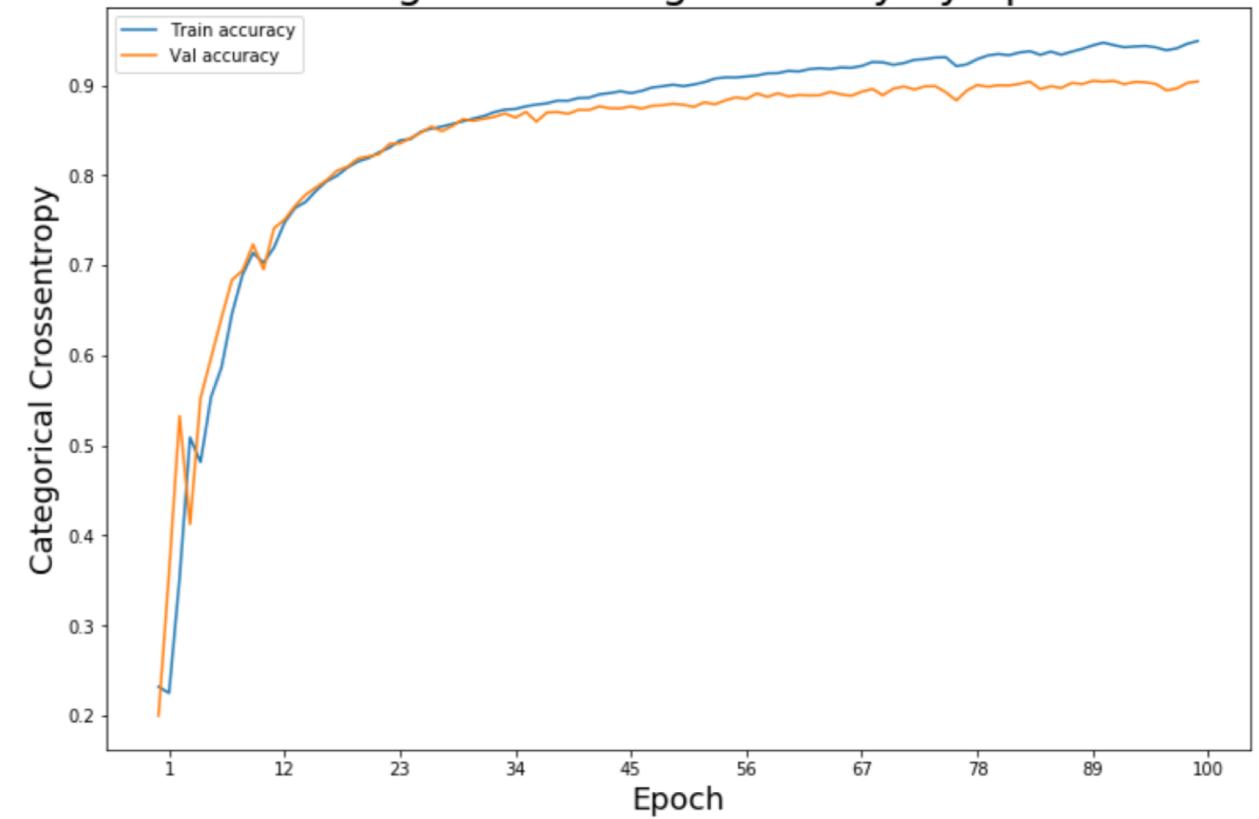


Model Evaluation

Training and Testing Loss by Epoch



Training and Testing Accuracy by Epoch



loss: 0.26
acc: 0.915

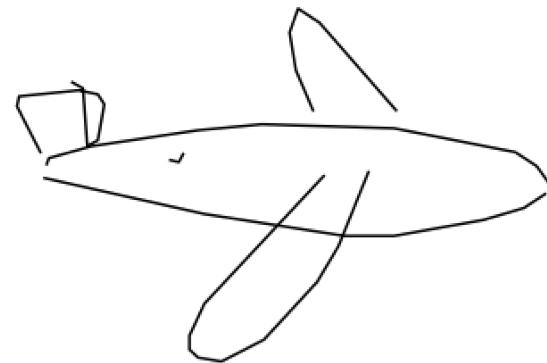
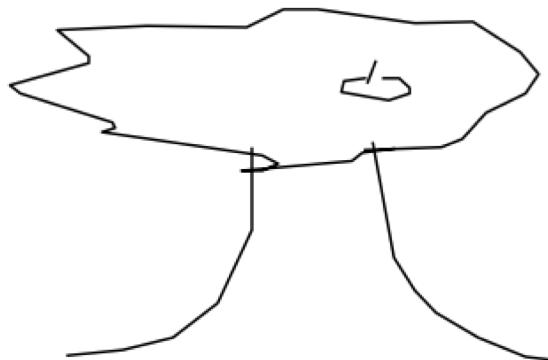
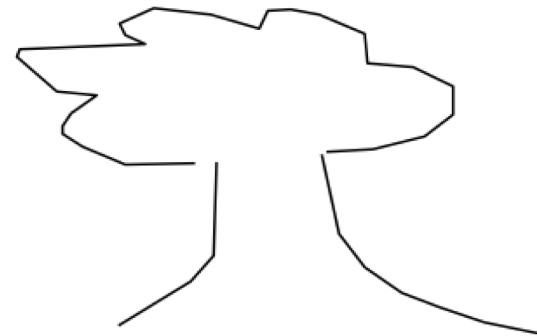
Model Understanding: Clustering Misclassifications

- Are there patterns in images that were misclassified that can be grouped together into clusters?
- Do these clusters make sense?
- If yes, understand when the model is expected to fail at correctly predicting
- Is this failure within a reasonable acceptance criteria?
- If no – what can be done?
- Is it worth it to try and improve?

Model Understanding: Clustering Misclassifications

- Unsupervised Learning
- 6250 images in test set - 525 misclassifications
- Misclassifications data frame: 525 rows by 6600 columns (unrowed data)
- Drop columns with all 0's
- PCA
- DBSCAN
- K-Means

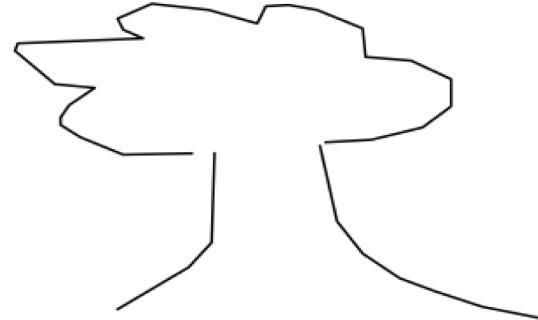
Model Understanding: K-Means Clustering



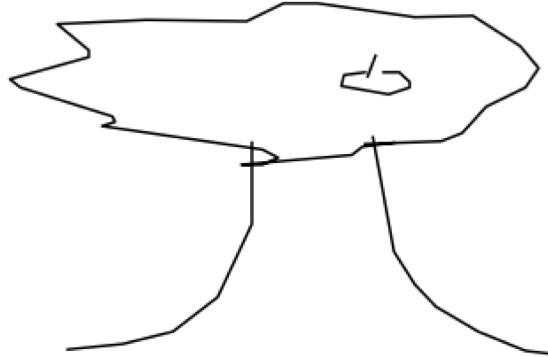
Model Understanding: K-Means Clustering



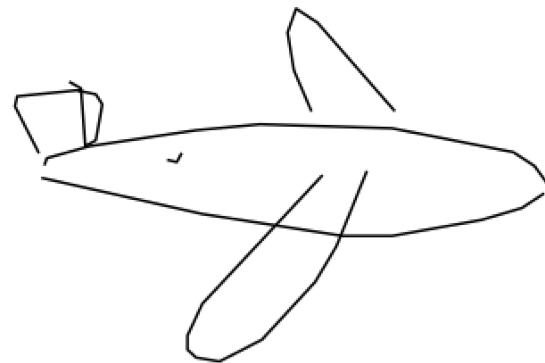
Misclassified as airplane



Misclassified as airplane

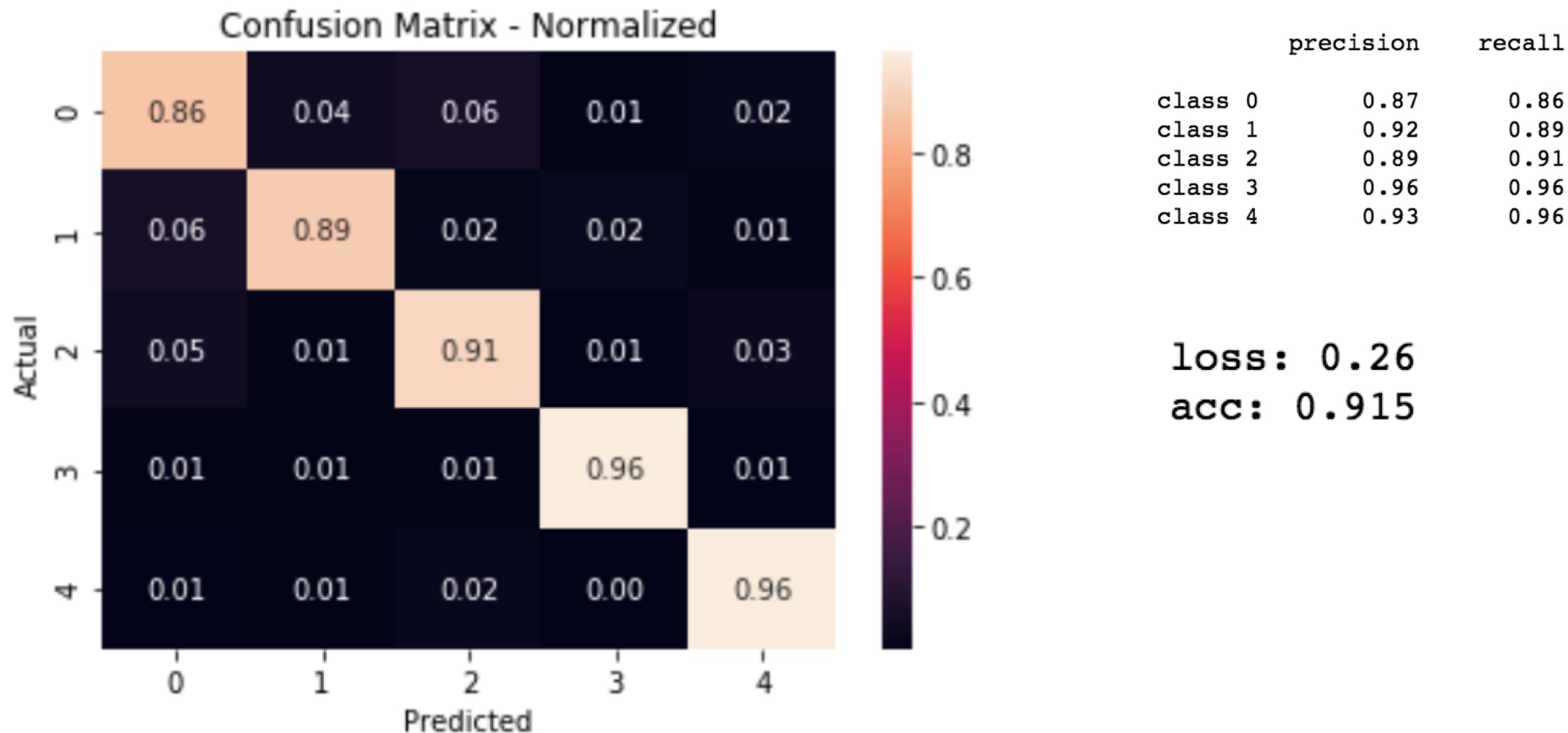


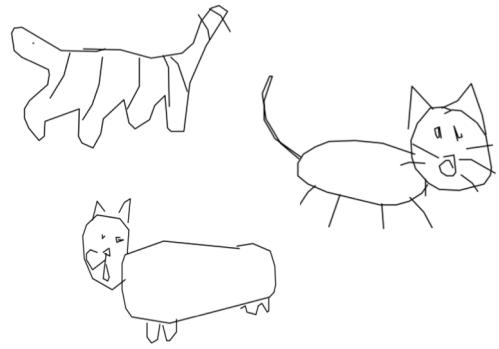
Misclassified as airplane



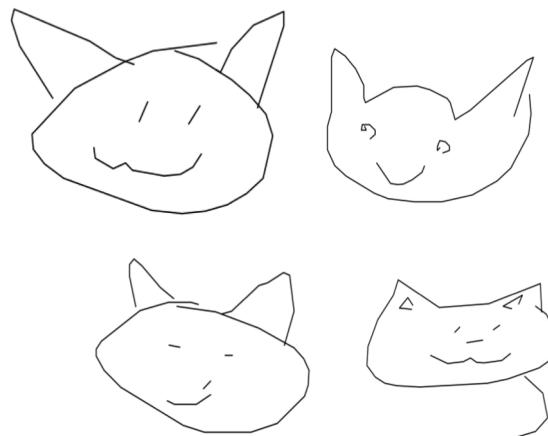
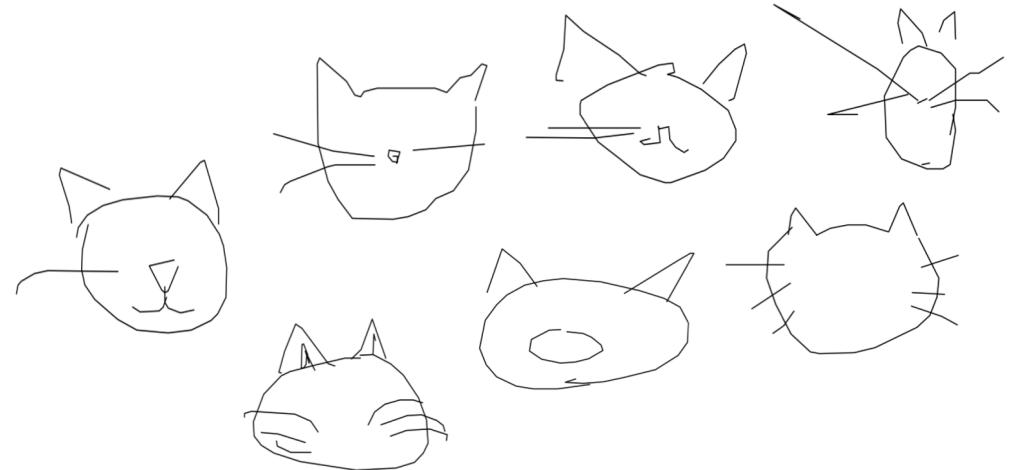
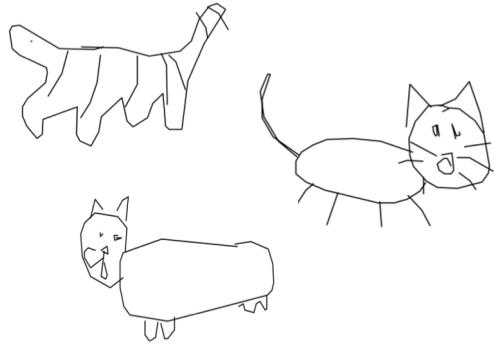
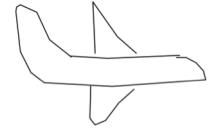
Misclassified as cat

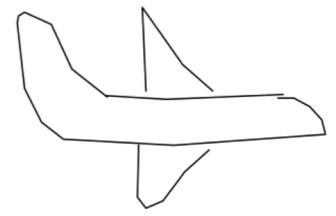
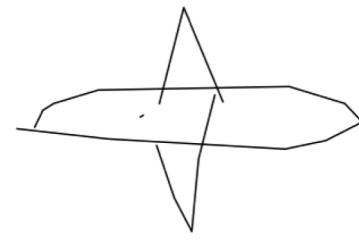
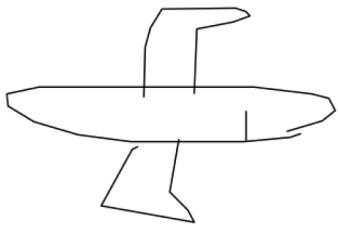
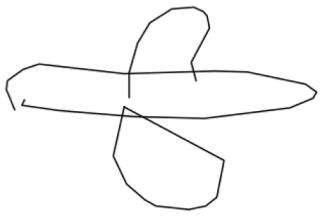
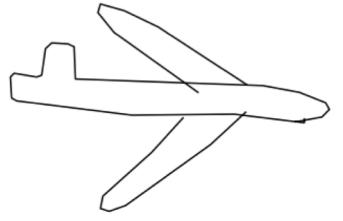
Model Evaluation



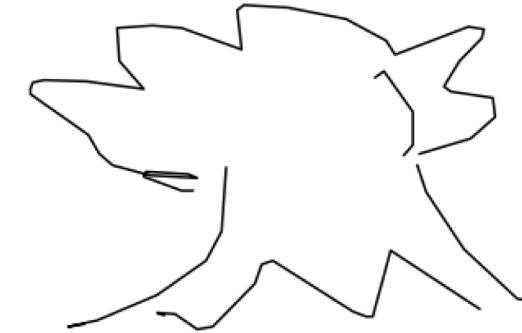
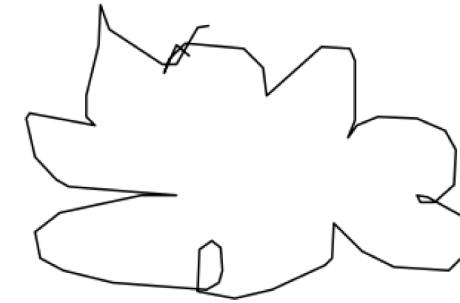
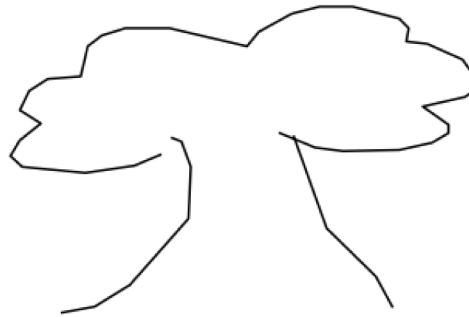
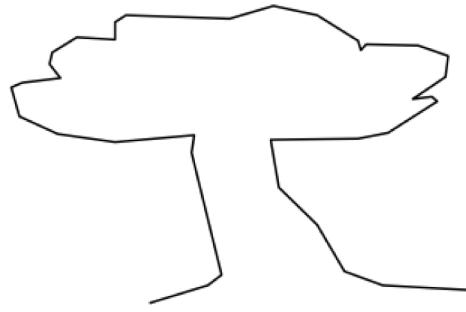
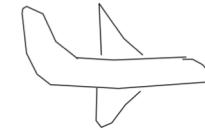


Misclassifications: Cats As Airplanes

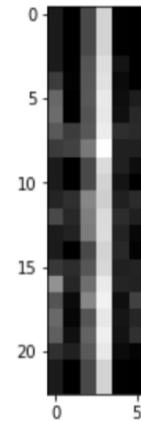
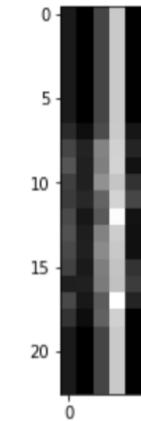
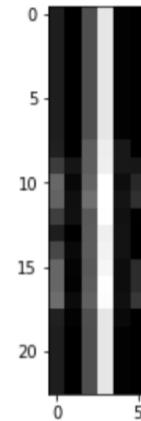
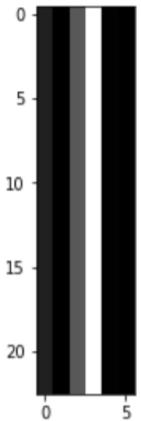
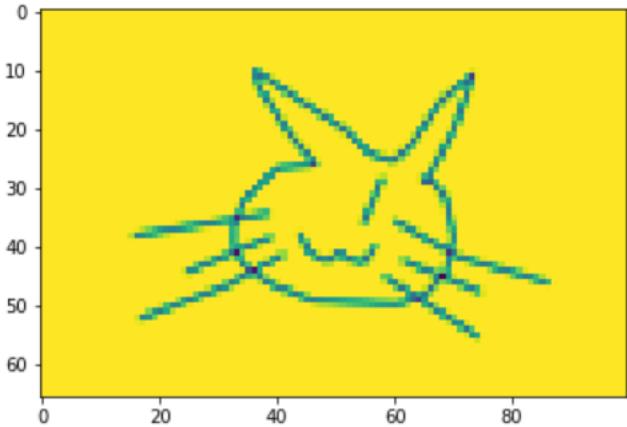




Misclassifications: Trees As Airplanes

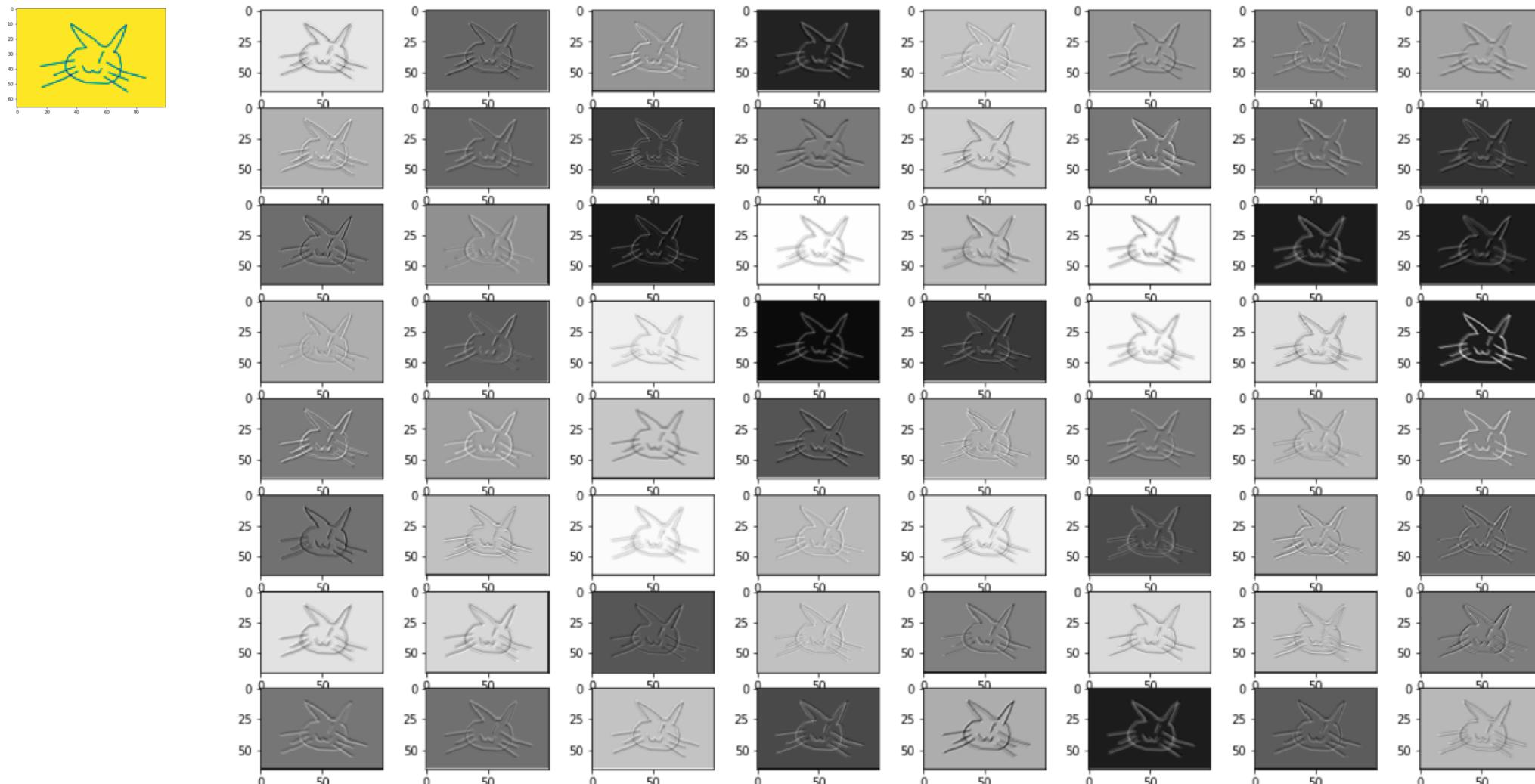


Model Understanding: Visualizing Hidden Layer Output



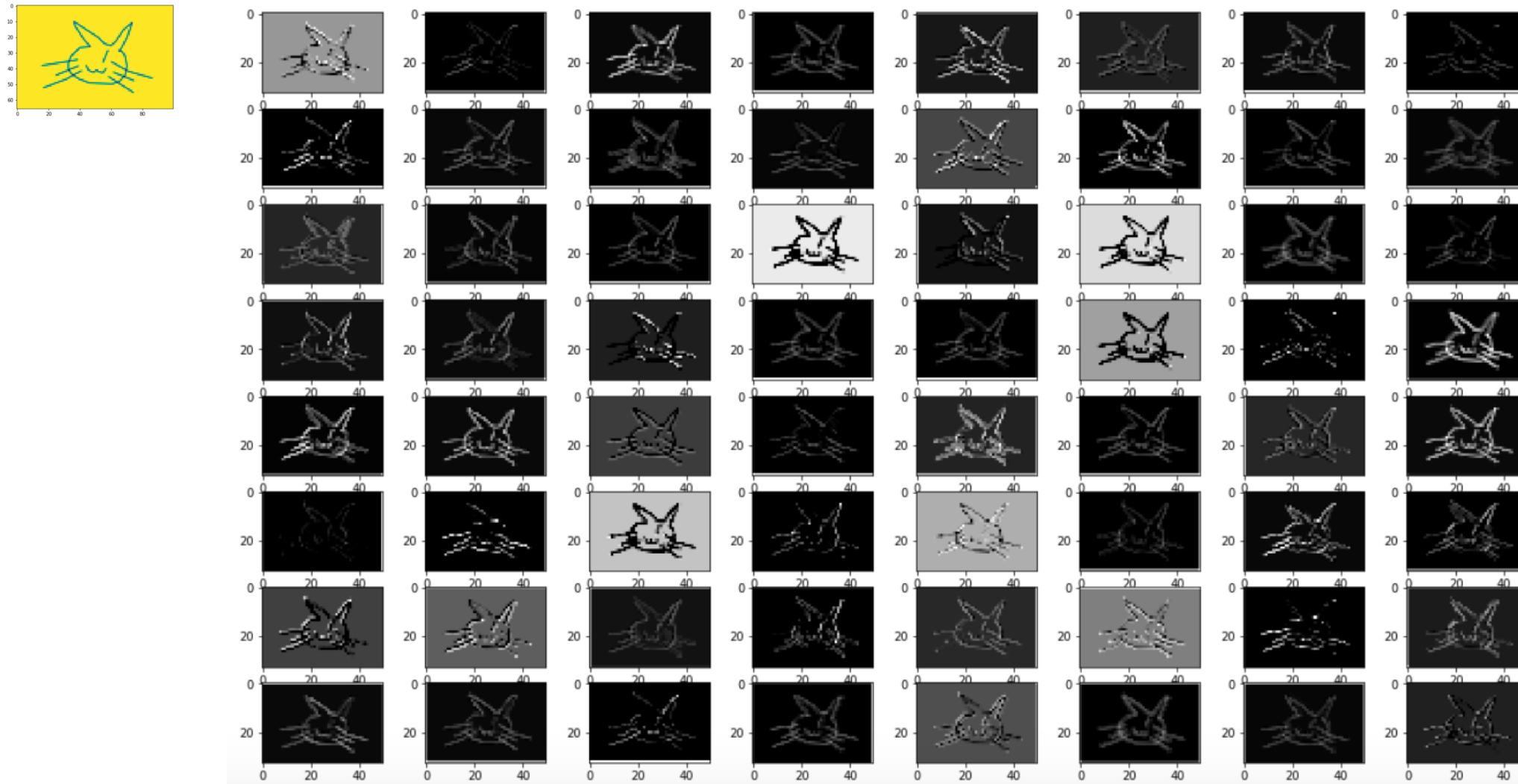
Model Understanding: Visualizing Hidden Layer Output

Activations After Layer 2



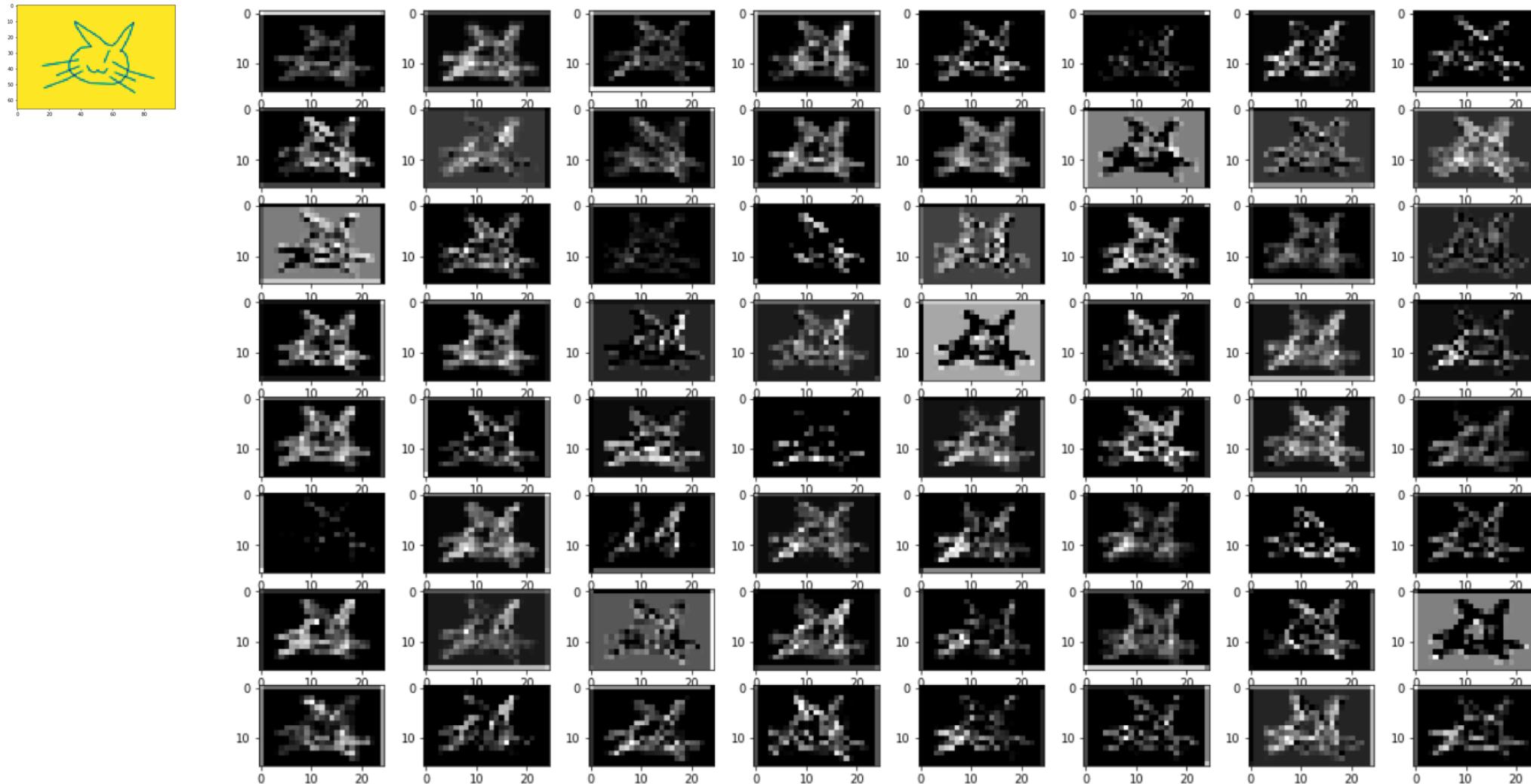
Model Understanding: Visualizing Hidden Layer Output

Activations After Layer 4



Model Understanding: Visualizing Hidden Layer Output

Activations After Layer 8



Next Steps and Conclusions

- Understand what my model hidden layers are outputting
- Manually label clusters of misclassification to better understand the model
- Investigate individual nodes contributions to the output
- Determine if the data is good
- Scale it up: expand the dataset and use cloud computing Amazon Web Services
- Retrain the model – tackle overfitting
- Understanding the inner workings of the algorithm to build better performing product – happy users

Thank You!

Layer (type)	Output Shape	Param #
<hr/>		
input_1 (InputLayer)	(None, 66, 100, 1)	0
layer_conv1 (Conv2D)	(None, 66, 100, 64)	640
batch_normalization_1 (Batch Normalization)	(None, 66, 100, 64)	256
activation_1 (Activation)	(None, 66, 100, 64)	0
maxPool1 (MaxPooling2D)	(None, 33, 50, 64)	0
layer_conv2 (Conv2D)	(None, 33, 50, 64)	36928
batch_normalization_2 (Batch Normalization)	(None, 33, 50, 64)	256
activation_2 (Activation)	(None, 33, 50, 64)	0
maxPool2 (MaxPooling2D)	(None, 16, 25, 64)	0
conv3 (Conv2D)	(None, 16, 25, 32)	18464
batch_normalization_3 (Batch Normalization)	(None, 16, 25, 32)	128
activation_3 (Activation)	(None, 16, 25, 32)	0
maxPool3 (MaxPooling2D)	(None, 8, 12, 32)	0
flatten_1 (Flatten)	(None, 3072)	0
fc0 (Dense)	(None, 64)	196672
dropout_1 (Dropout)	(None, 64)	0
fc1 (Dense)	(None, 32)	2080
dropout_2 (Dropout)	(None, 32)	0
fc2 (Dense)	(None, 5)	165
<hr/>		
Total params: 255,589		
Trainable params: 255,269		
Non-trainable params: 320		