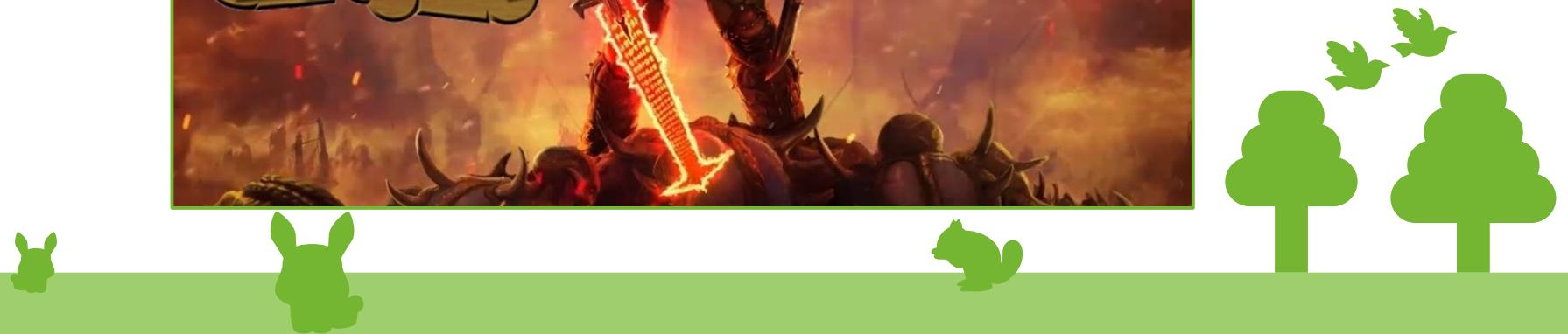


DoomCrossing

Using machine learning to identify the two side of an unlikely friendship



Background

- In March of 2020 two very different video games were released on the same day



Background

- In March of 2020 two very different video games were released on the same day
- This event lead to unlikely comradery between these disparate communities



Background

- In March of 2020 two very different video games were released on the same day
- This event led to unlikely comradery between these disparate communities
- ...and to a wealth of fan art and dank memes



The Data

- Collection of images scraped from the Doom and Animal Crossing subreddits
- Total of 1597 images; 757 from animal crossing and 840 from Doom
- Split into a training/test set
 - 10% into test set
- Images are highly varied
 - Some show the games
 - Others are text or memes



Data Examples: Animal Crossing



Me: *turns on Animal Crossing*

Isabelle:



Data Examples: Animal Crossing



Vesta

I'm playing this one game where you run your own town but also have to keep all these needy animals happy.

ACWII animalcrossingwii

that game sounds stupid

This block contains a screenshot from the video game Animal Crossing: Wii. A character with a purple head and a green body, identified as Vesta, is speaking. The text bubble says: "I'm playing this one game where you run your own town but also have to keep all these needy animals happy." Below the screenshot is a tweet from the user "animalcrossingwii" with the ACWII logo, stating: "that game sounds stupid".



Data Examples: Animal Crossing



kayla dimarco
@kaylamdimarco

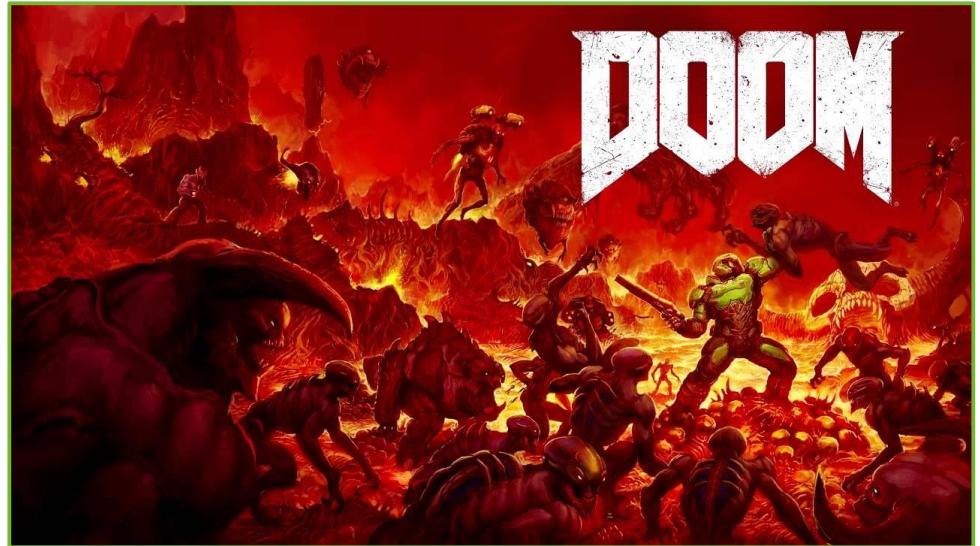
my mom didnt let me play animal crossing for a while because when i got a note saying that my neighbor Hazel the squirrel moved away i cried so hard i threw up on the carpet

11/19/18, 10:28 AM

funny.co



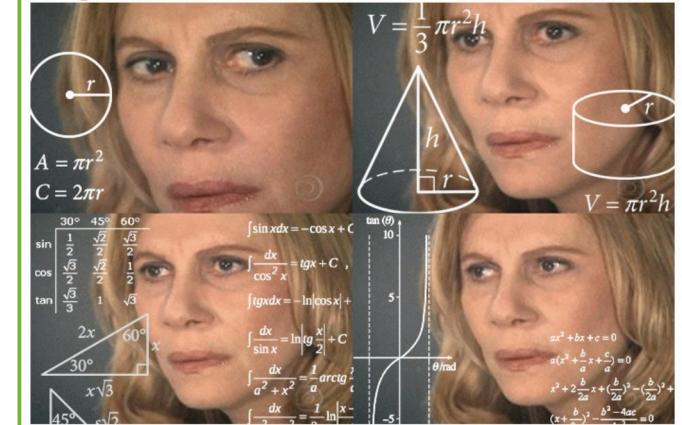
Data Examples: Doom



Data Examples: Doom



ARC soldier wondering why The Slayer took his plasma rifle even though he has had one for half the game.



Data Examples: Shared

Doom



Using sentinel batteries to unlock new abilities for the Slayer



Using sentinel batteries to unlock the original Doom Marine skin

Animal Crossing



WHEN APRIL 23 UPDATE ARRIVES

imgflip.com



Data Examples: Shared

Doom

Me realizing Id delaying Doom Eternal's release to March 2020 might upset a lot of the fanbase and newcomers



Me seeing this sub with genuine, patient support of the devs who are working hard to make it a perfect game



Animal Crossing

FAVORITE
VILLAGER HAS A
THOUGHT
BUBBLE



WANTS TO KNOW IF
THEY CAN CALL ME
BIG CHUNGUS TOO



Data Examples: Complicated

Doom



Doom



Animal Crossing



Data Examples: Complicated

Doom

**Look at them, they are scrolling through
memes and not playing our games**



Models

- Tree Based Models
- Simple Neural Net
- PCA Optimized Neural Net
- Custom Convolutional Neural Nets
- Edge Detecting Neural Nets
- Inception Model
- ResNet18 Model



Preprocessing

- Resize and crop
 - 200x200 for clustering/tree based model
 - 256x256 for neural nets
 - Random horizontal flip on training set
 - Normalization



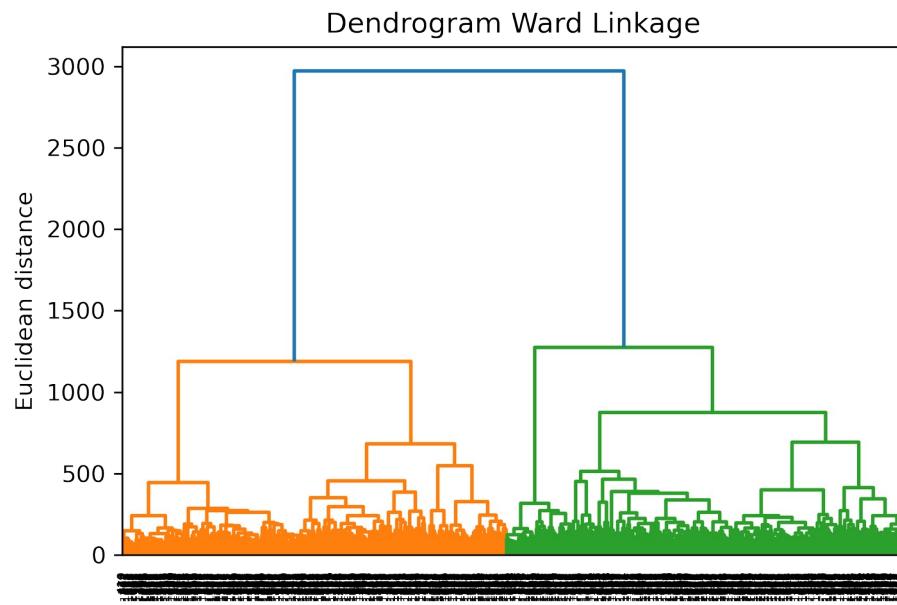
Tree Based Models

- Trees built using four different linkage methods
 - Ward Method, Complete, Average, and Single
 - Ward method seeks to minimize inter-cluster variance
- Although the classification based on clustering does not necessarily align with the true labels, the confusion matrix can give some sense of how close the cluster proportions match the true proportions.
- Mostly just an excuse to make dendograms



Tree Based Models

- Ward Linkage



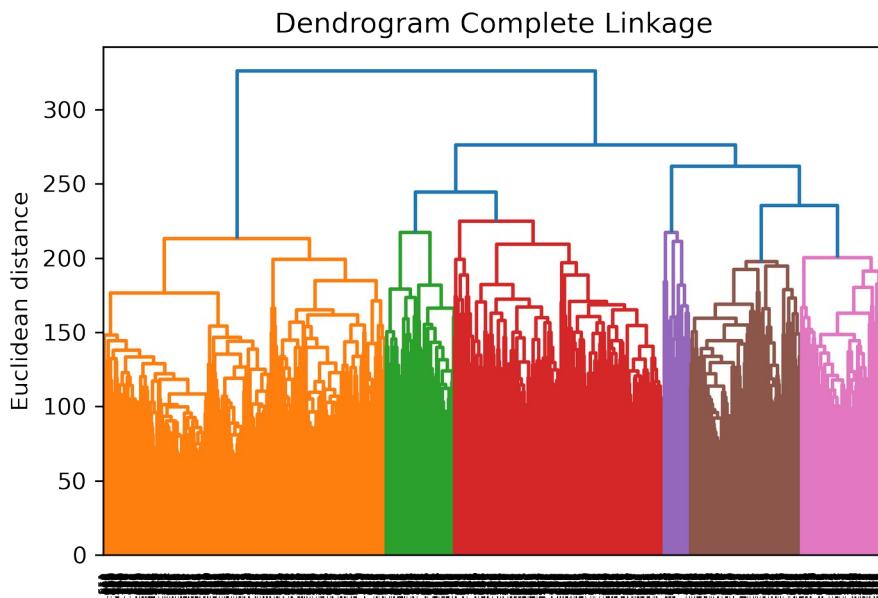
	precision	recall	f1-score	support
0.0	0.58	0.63	0.60	676
1.0	0.64	0.60	0.62	760
accuracy			0.61	1436
macro avg	0.61	0.61	0.61	1436
weighted avg	0.61	0.61	0.61	1436

Confusion Matrix
[[423 253]
 [307 453]]



Tree Based Models

- Complete Linkage



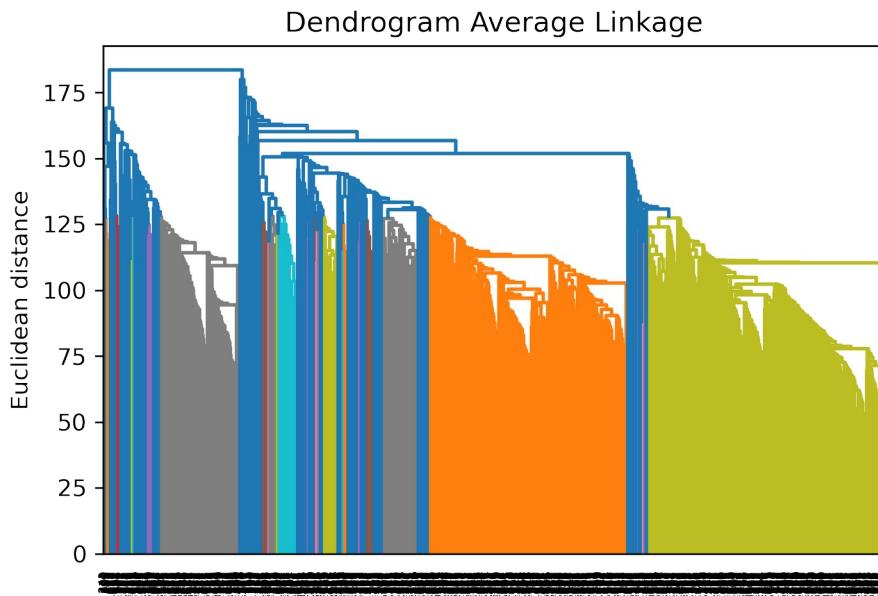
	precision	recall	f1-score	support
0.0	0.56	0.76	0.64	676
1.0	0.68	0.47	0.55	760
accuracy				0.60
macro avg	0.62	0.61	0.60	1436
weighted avg	0.62	0.60	0.59	1436

Confusion Matrix
[[511 165]
[406 354]]



Tree Based Models

- Average Linkage



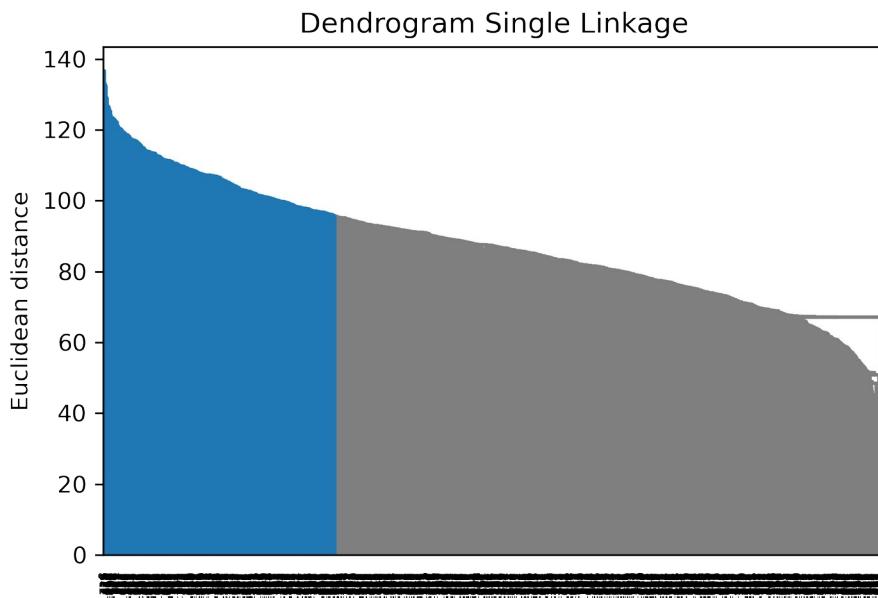
	precision	recall	f1-score	support
0.0	0.47	0.82	0.60	676
1.0	0.53	0.17	0.26	760
accuracy				0.48
macro avg	0.50	0.50	0.43	1436
weighted avg	0.50	0.48	0.42	1436

Confusion Matrix
[[557 119]
[628 132]]



Tree Based Models

- Single Linkage

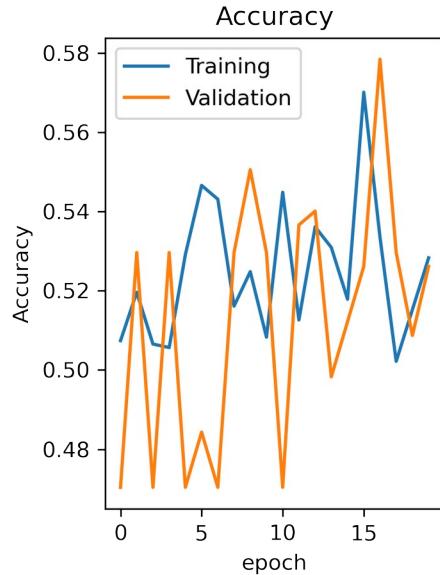
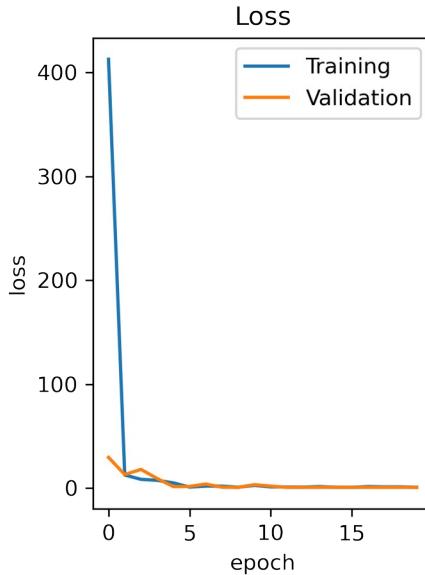


	precision	recall	f1-score	support
0.0	0.47	1.00	0.64	676
1.0	1.00	0.00	0.00	760
accuracy				0.47
macro avg	0.74	0.50	0.32	1436
weighted avg	0.75	0.47	0.30	1436

Confusion Matrix
[[676 0]
[759 1]]



Simple Neural Net



Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 1024)	302080
dense_1 (Dense)	(None, 1024)	1049600
dense_2 (Dense)	(None, 512)	524800
dense_3 (Dense)	(None, 256)	131328
dense_4 (Dense)	(None, 2)	514

Total params: 2,008,322

Trainable params: 2,008,322

Non-trainable params: 0

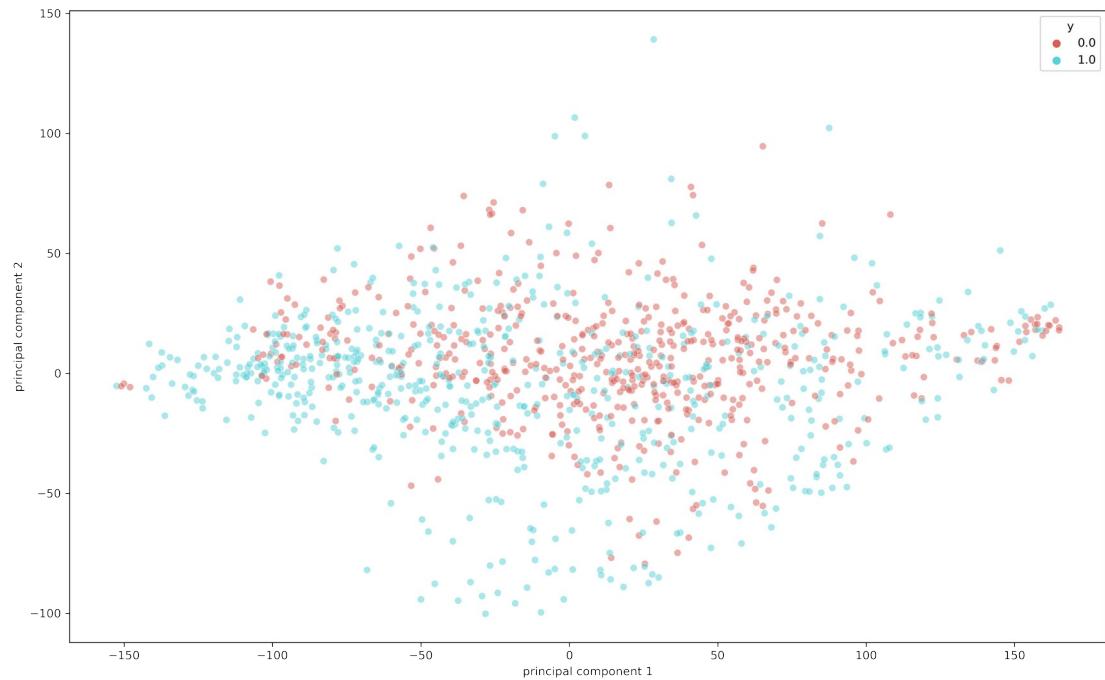
Maximum validation accuracy: 57.8 %

Test set accuracy: 53.3 %

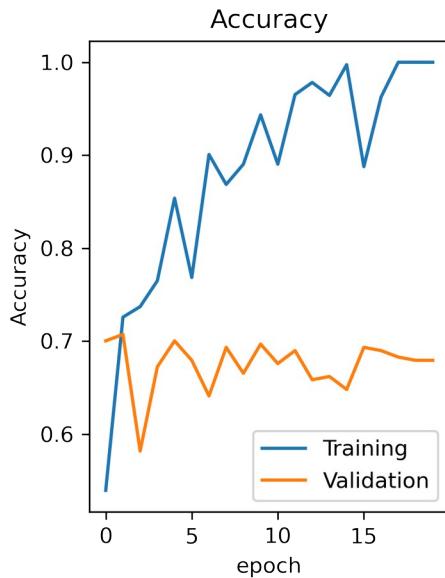
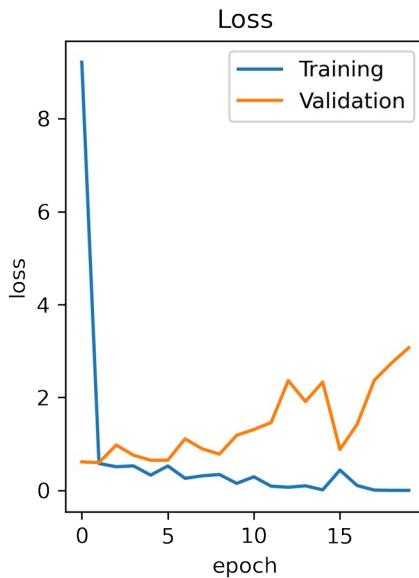


PCA Optimized Neural Net

- PCA is ran on the flattened images
- The model is ran on the principal components that capture 90% of the variance in the data
- Training runs extremely quickly
- Variance captured in the first two PCs
 - [0.379 0.070]
 - Many PCs required but much less than actual pixels



PCA Optimized Neural Net



Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 1024)	302080
dense_1 (Dense)	(None, 1024)	1049600
dense_2 (Dense)	(None, 512)	524800
dense_3 (Dense)	(None, 256)	131328
dense_4 (Dense)	(None, 2)	514

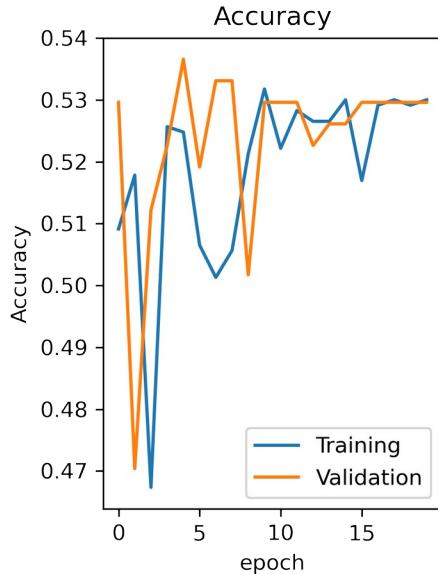
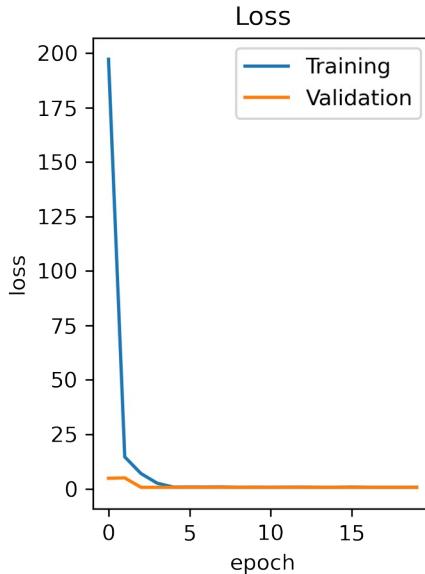
Total params: 2,008,322
Trainable params: 2,008,322
Non-trainable params: 0

Maximum validation accuracy: 70.7 %

Test set accuracy: 70.7 %



Simple Neural Net



Model: "sequential_3"

Layer (type)	Output Shape	Param #
dense_15 (Dense)	(None, 1024)	122881024
dropout_4 (Dropout)	(None, 1024)	0
dense_16 (Dense)	(None, 1024)	1049600
dropout_5 (Dropout)	(None, 1024)	0
dense_17 (Dense)	(None, 512)	524800
dropout_6 (Dropout)	(None, 512)	0
dense_18 (Dense)	(None, 256)	131328
dropout_7 (Dropout)	(None, 256)	0
dense_19 (Dense)	(None, 2)	514

Total params: 124,587,266

Trainable params: 124,587,266

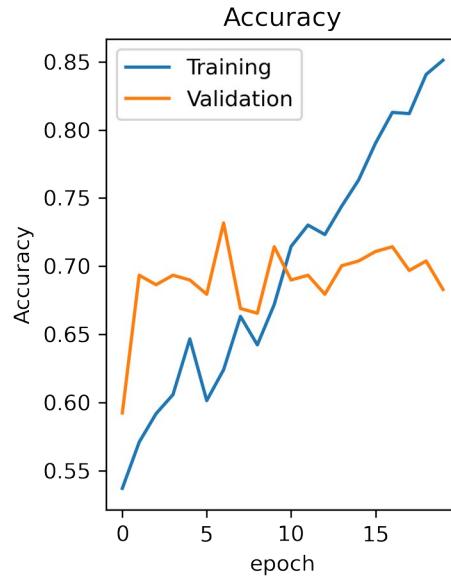
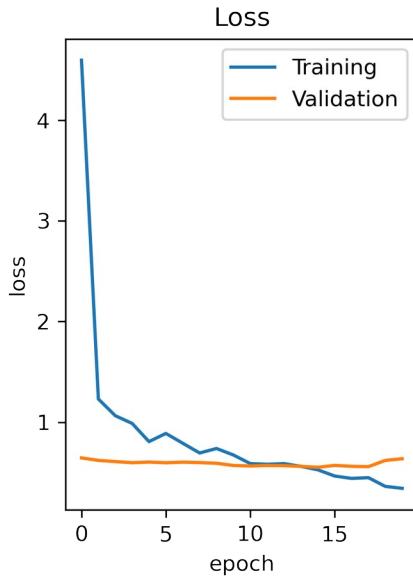
Non-trainable params: 0

Maximum validation accuracy: 53.6 %

Test set accuracy: 53.0 %



PCA Optimized Neural Net



Model: "sequential_2"

Layer (type)	Output Shape	Param #
dense_10 (Dense)	(None, 1024)	302080
dropout (Dropout)	(None, 1024)	0
dense_11 (Dense)	(None, 1024)	1049600
dropout_1 (Dropout)	(None, 1024)	0
dense_12 (Dense)	(None, 512)	524800
dropout_2 (Dropout)	(None, 512)	0
dense_13 (Dense)	(None, 256)	131328
dropout_3 (Dropout)	(None, 256)	0
dense_14 (Dense)	(None, 2)	514

Total params: 2,008,322

Trainable params: 2,008,322

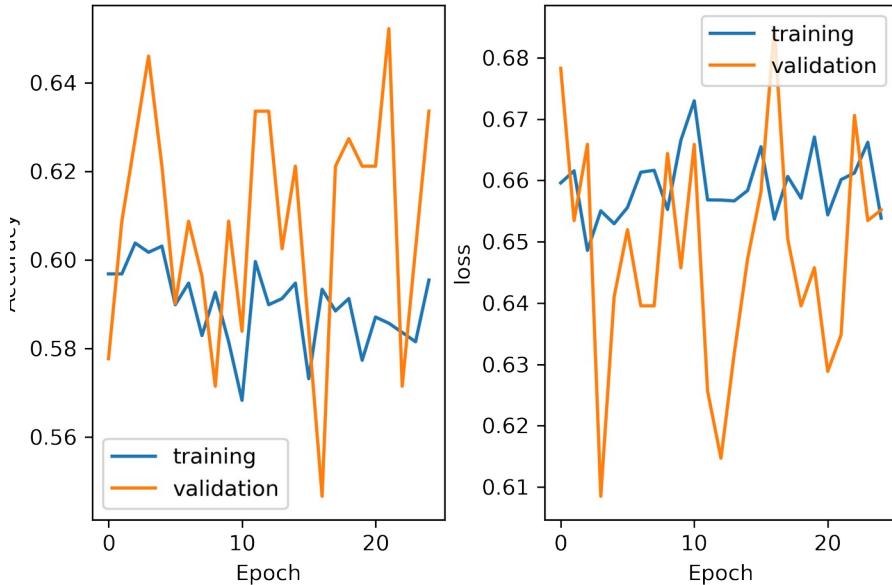
Non-trainable params: 0

Maximum validation accuracy: 73.1 %

Test set accuracy: 71.4 %



Custom Convolutional Neural Nets

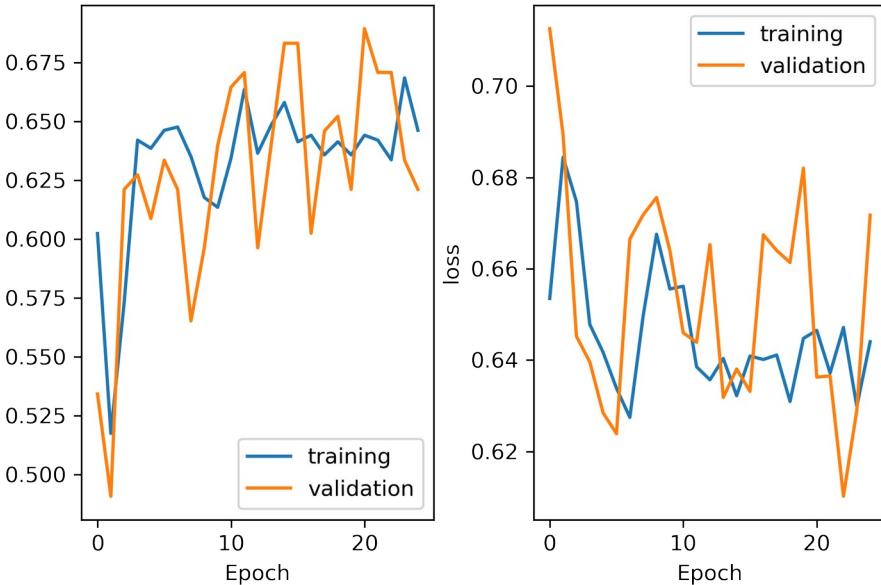


- Built with PyTorch
- One convolutional layer
 - out_channels: 128
 - Relu activation
 - 3x3 kernels
 - MaxPool (2x2 kernel)
- One fully connected layer
 - Softmax activation
- Adam Optimizer
 - With learning rate scheduler

Best Validation Accuracy: 65.2%



Custom Convolutional Neural Nets

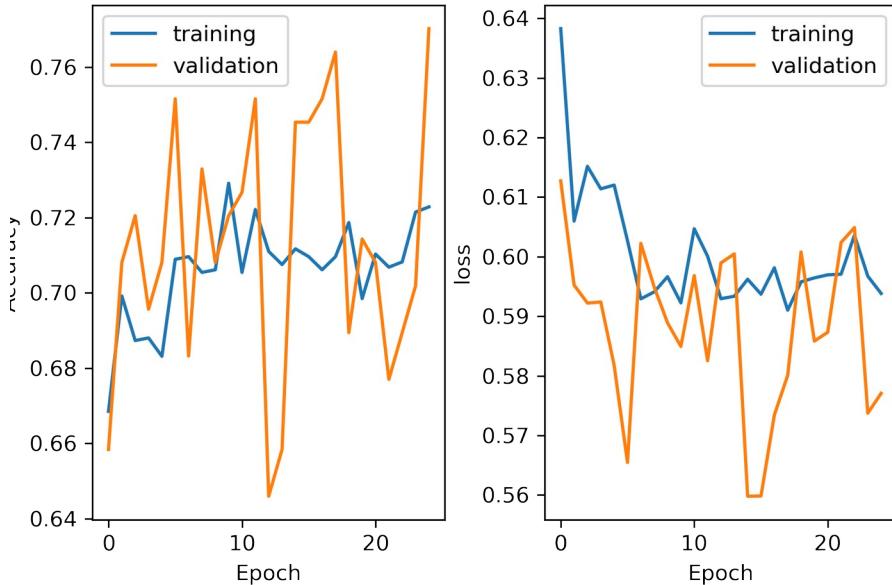


- Built with PyTorch
- Two convolutional layers
 - out_channels: 16/32
 - Relu activation
 - 3x3 kernels
 - MaxPool (2x2 kernel)
- One fully connected layer
 - Softmax activation
- Adam Optimizer
 - With learning rate scheduler

Best Validation Accuracy: 68.9%



Custom Convolutional Neural Nets

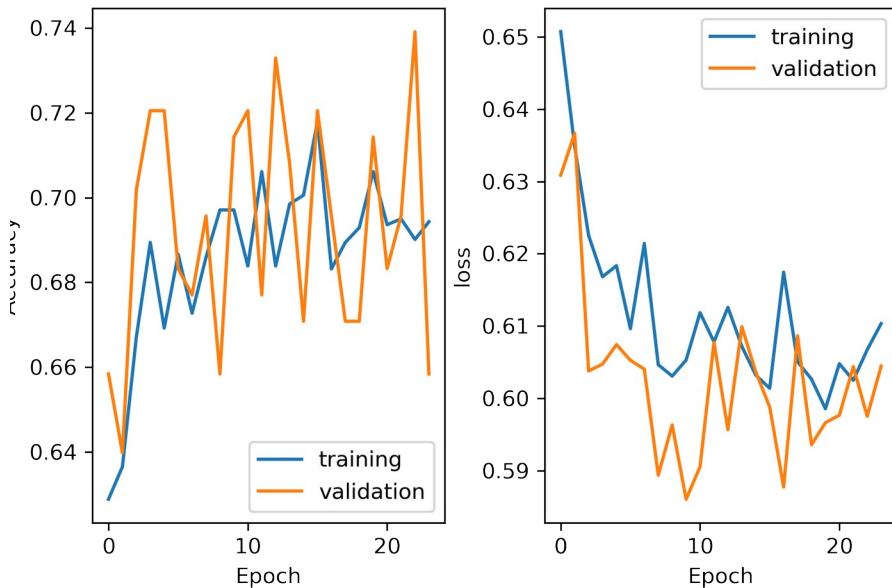


Best Validation Accuracy: 77.0%

- Built with PyTorch
- Two convolutional layers
 - out_channels: 16/32
 - Relu activation
 - 3x3 kernels
 - MaxPool (2x2 kernel)
- Two fully connected layer
 - Softmax activation
- Adam Optimizer
 - With learning rate scheduler



Custom Convolutional Neural Nets

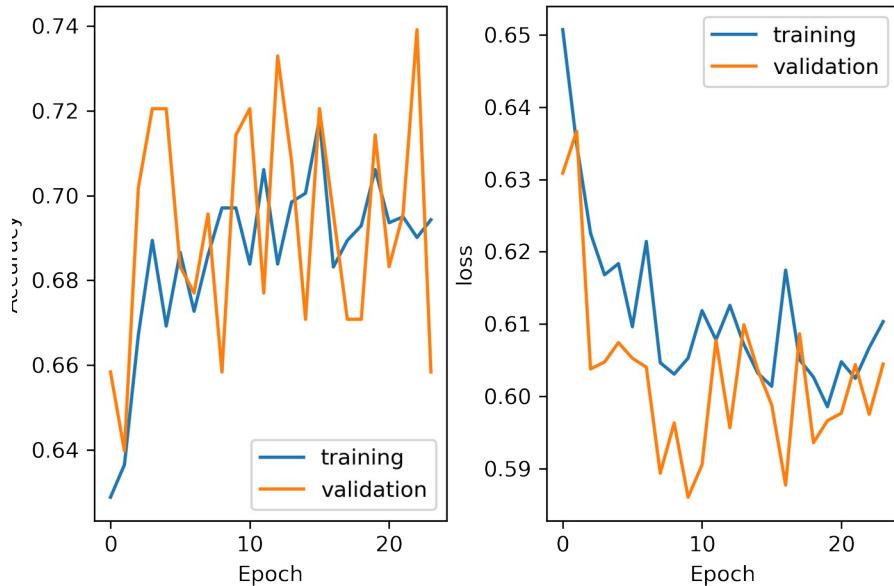


Best Validation Accuracy: 73.9%

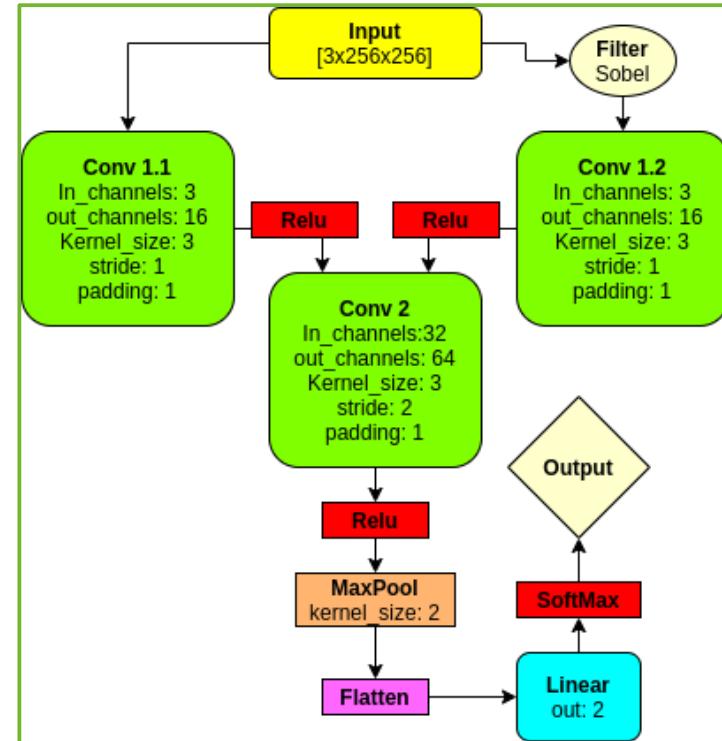
- Parallel convolutions
 - One on raw input and another with filtered input (sobel)
- Three convolutional layers
 - out_channels: 16/16/64
 - Relu activation
 - 3x3 kernels
 - MaxPool (2x2 kernel)
- Two fully connected layer
 - Softmax activation
- Adam Optimizer
 - With learning rate scheduler



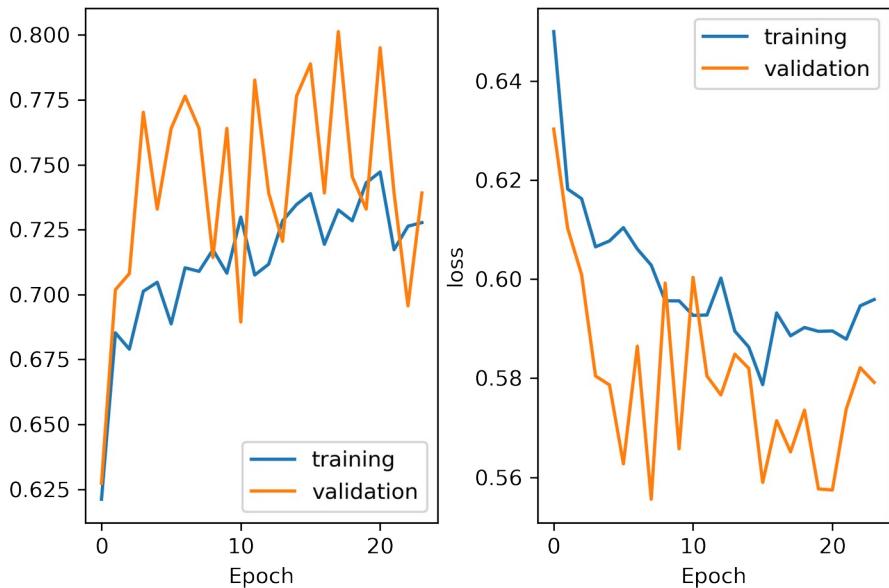
Custom Convolutional Neural Nets



Best Validation Accuracy: 73.9%



Custom Convolutional Neural Nets

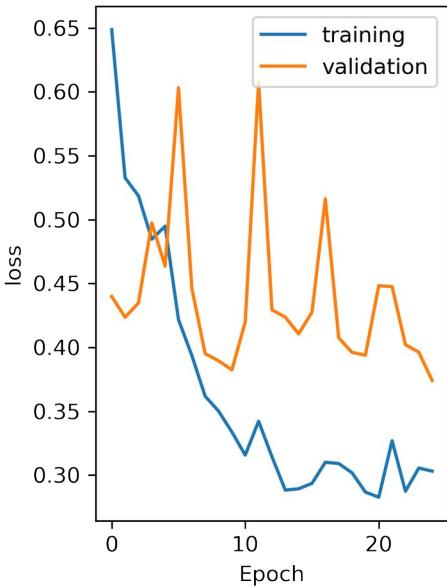
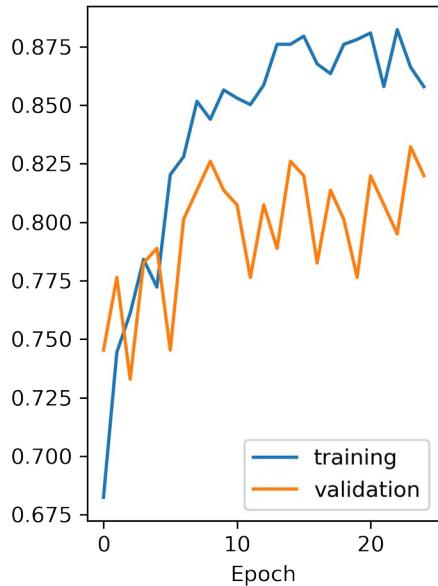


Best Validation Accuracy: 80.1%

- Parallel convolutions
 - One on raw input and another with filtered input (sobel)
- Five convolutional layer
 - out_channels: 8/8/16/16/64
 - Relu activation
 - 3x3 kernels
 - MaxPool (2x2 kernel)
- Two fully connected layer
 - Softmax activation
- Batch normalization of final convolution



Inception v3

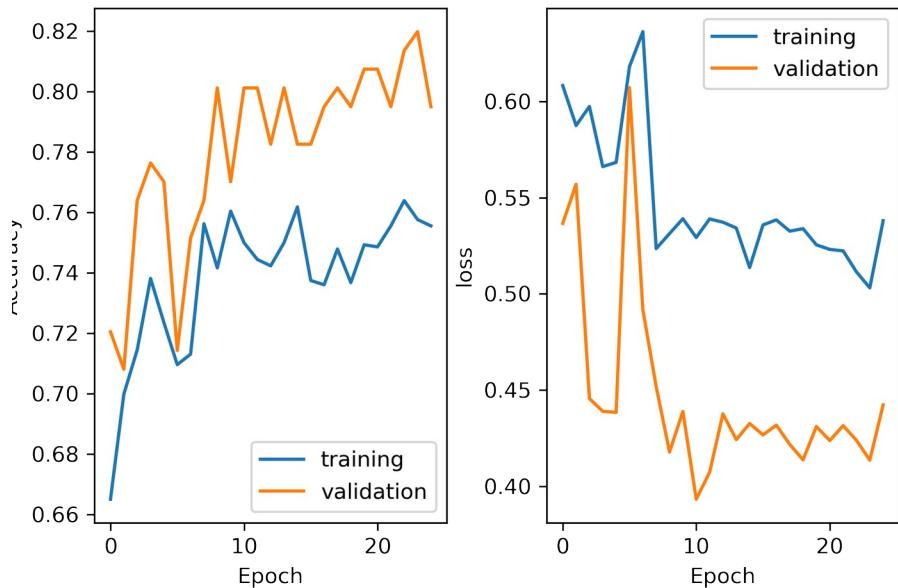


- All weights trained
- Cross entropy loss
- Stochastic gradient descent

Best Validation Accuracy: 83.2%



Inception v3

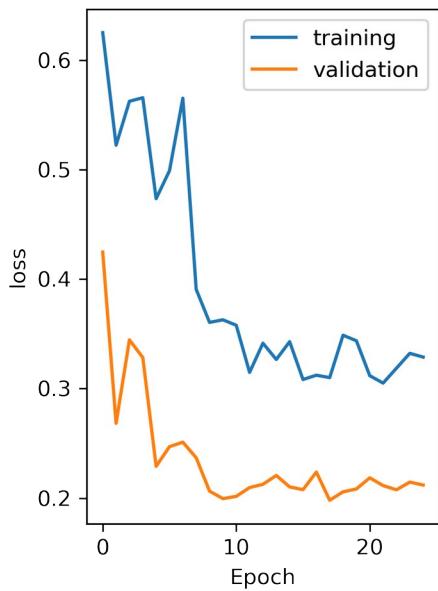
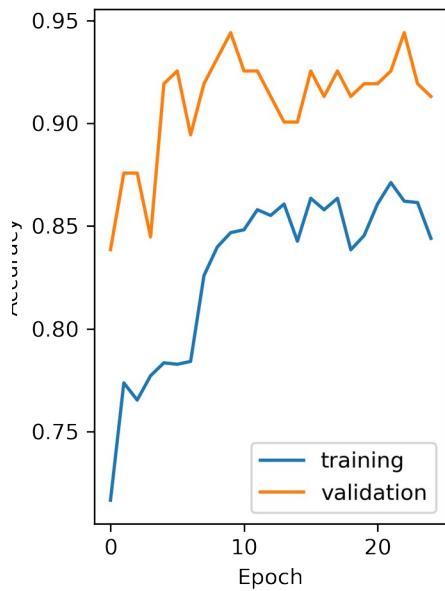


Best Validation Accuracy: 82.0%

- As a feature extractor
 - Only last layer trained
 - Cross entropy loss
 - Stochastic gradient descent



ResNet 18

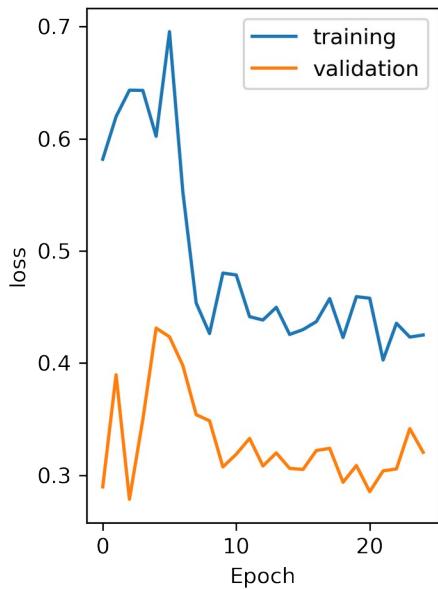


- All weights trained
- Cross entropy loss
- Stochastic gradient descent

Best Validation Accuracy: 94.4%



ResNet 18



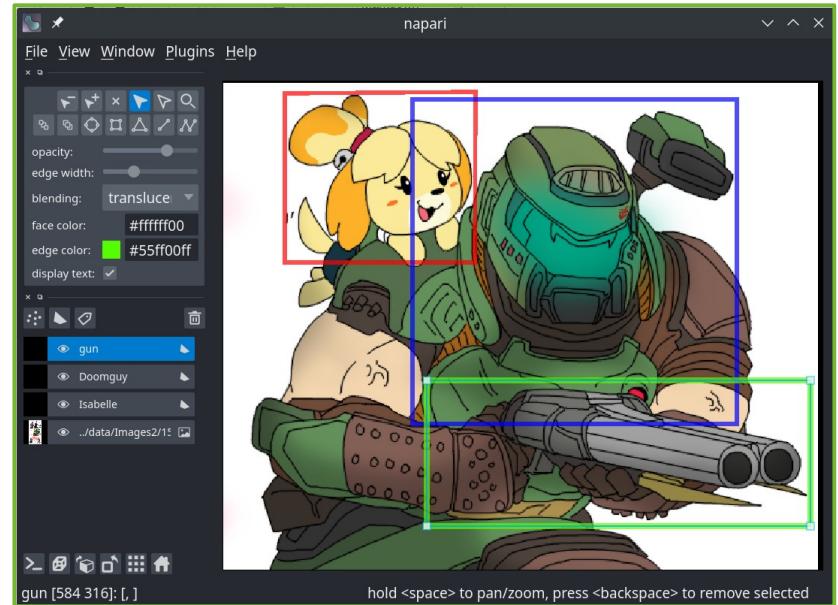
- As a feature extractor
 - Only last layer trained
 - Cross entropy loss
 - Stochastic gradient descent

Best Validation Accuracy: 87.6%



Next Steps

- Creating a model that generates a caption for an image
 - The data contains all of the reddit post information including the title which could be used to train a generative model
 - A region based CNN would be used to identify features in the image and a recurrent neural network would generate text based on this output using the true captions as examples
 - Requires annotation of classes in training images (in progress)
 - Success would be generation of a legible caption that had some level of relation to the image



Thank You

Questions?

