CAPSTONE: AGE ESTIMATION

Roy Kim

BACKGROUND: FACE DETECTION

- Heavy implications in government, business, advertisement
- Privacy concerns
- Social apps (Snapchat, Instagram)

DATA SCIENCE PROCESS

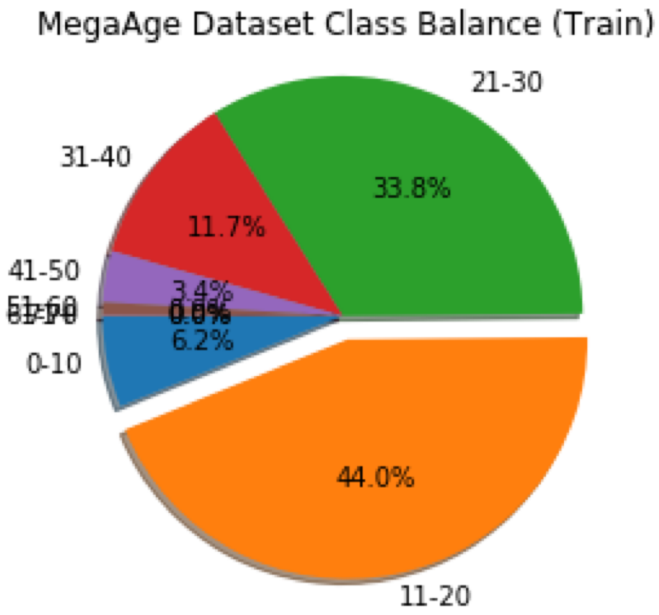
1. Define problem
2. Gather data
3. Explore data
4. Model with data
5. Evaluate data
6. Answer problem

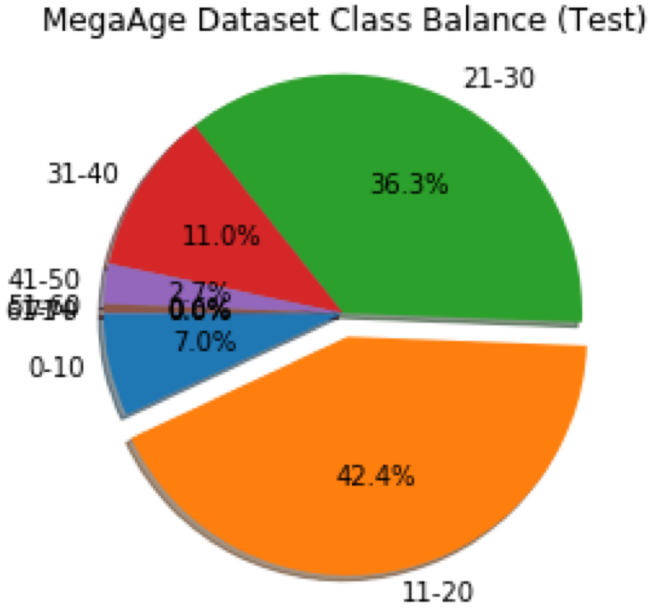
PROBLEM STATEMENT

Using convolutional neural networks, I will train a classification model to predict someone's age group based upon his/her picture. Model performance will be guided by cross entropy and accuracy (with consideration to the confusion matrix and other metrics), and the model will aim to achieve a score greater than baseline accuracy (determined by the number of classes).

DATA RETRIEVAL

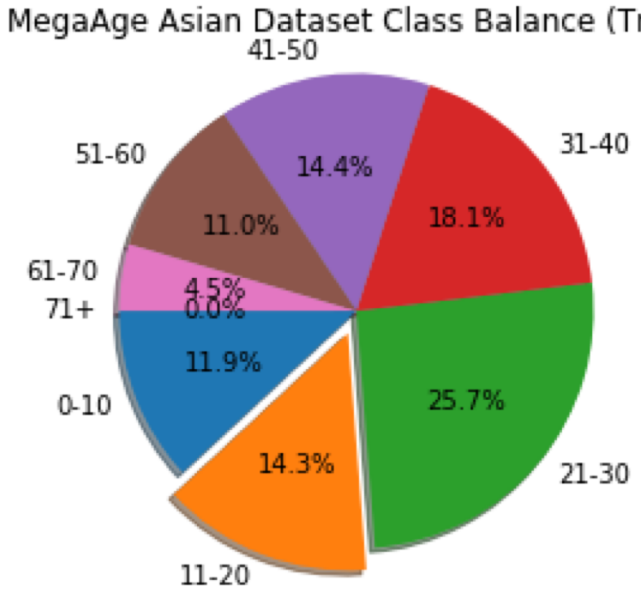
- MegaAge and MegaAge Asian dataset
- 41,941 annotated faces in MegaAge
- 40,000 annotated faces in MegaAge Asian

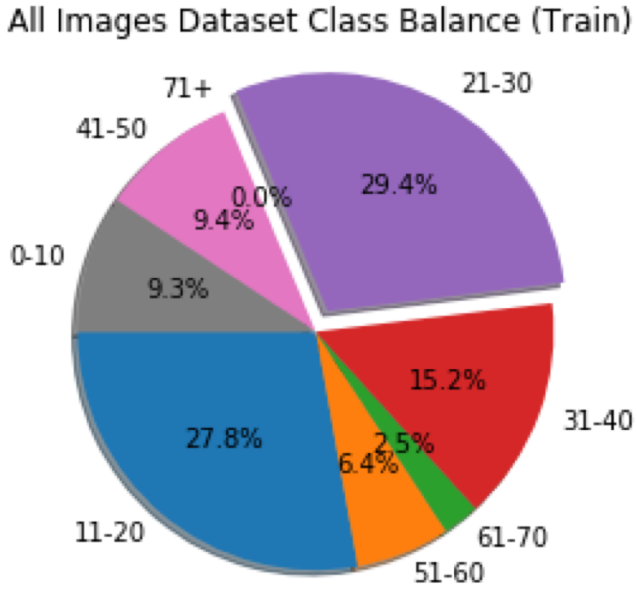


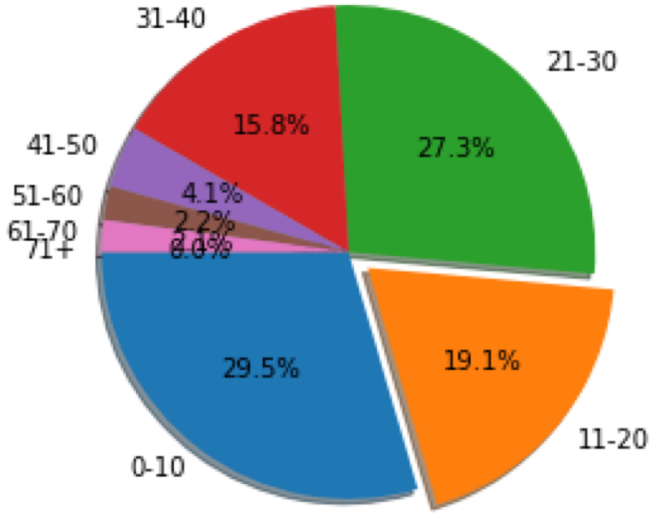


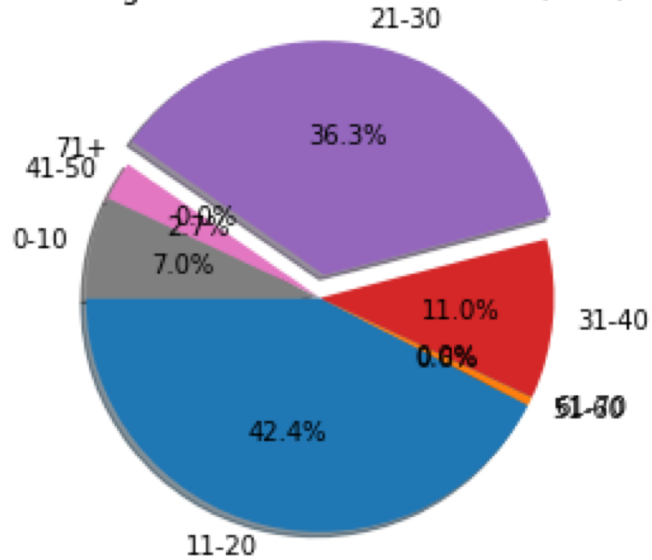
EXPLORING DATA: CLASS IMBALANCES

- First task was to classify each image into an age group programmatically
- Heavy class imbalance in the MegaAge dataset, causing the combined dataset to be imbalanced

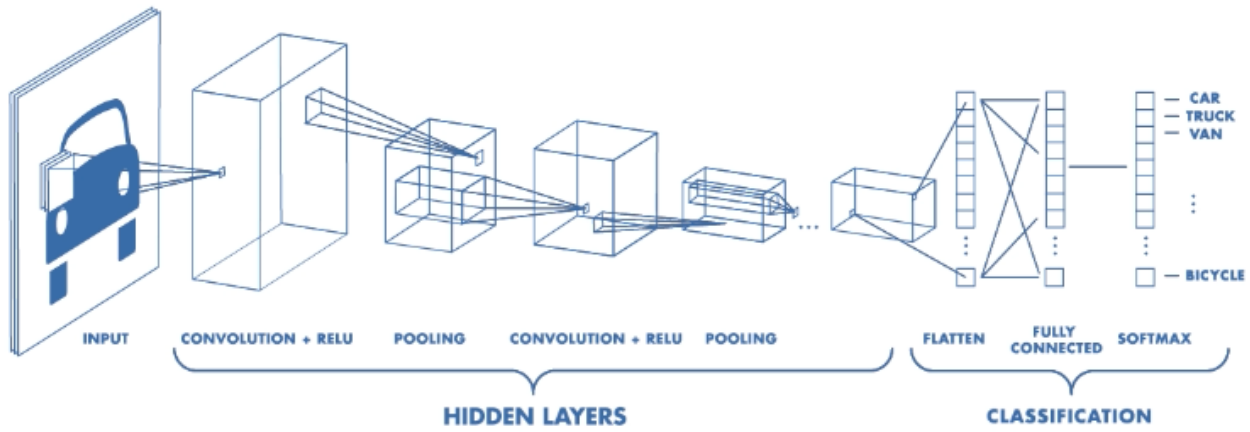

MegaAge Asian Dataset Class Balance (Train)
41-50
14.4%
31-40
18.1%
51-60
11.0%
61-70
4.5%
71+
0.0%
11.9%
0-10
25.7%
14.3%
21-30
11-20


All Images Dataset Class Balance (Train)
71+
21-30
41-50
0.0%
29.4%
9.4%
0-10
9.3%
15.2%
31-40
27.8%
2.5%
6.4%
61-70
11-20
51-60


MegaAge Asian Dataset Class Balance (Test)
31-40
21-30
15.8%
27.3%
41-50
51-60
4.1%
2.2%
61-70
71+
0.0%
29.5%
19.1%
11-20
0-10


All Images Dataset Class Balance (Test)
21-30
36.3%
71+
41-50
0.0%
2.7%
0-10
7.0%
11.0%
31-40
0.0%
42.4%
11-20

MODELING: CONVOLUTIONAL NEURAL NETWORKS (CNN)



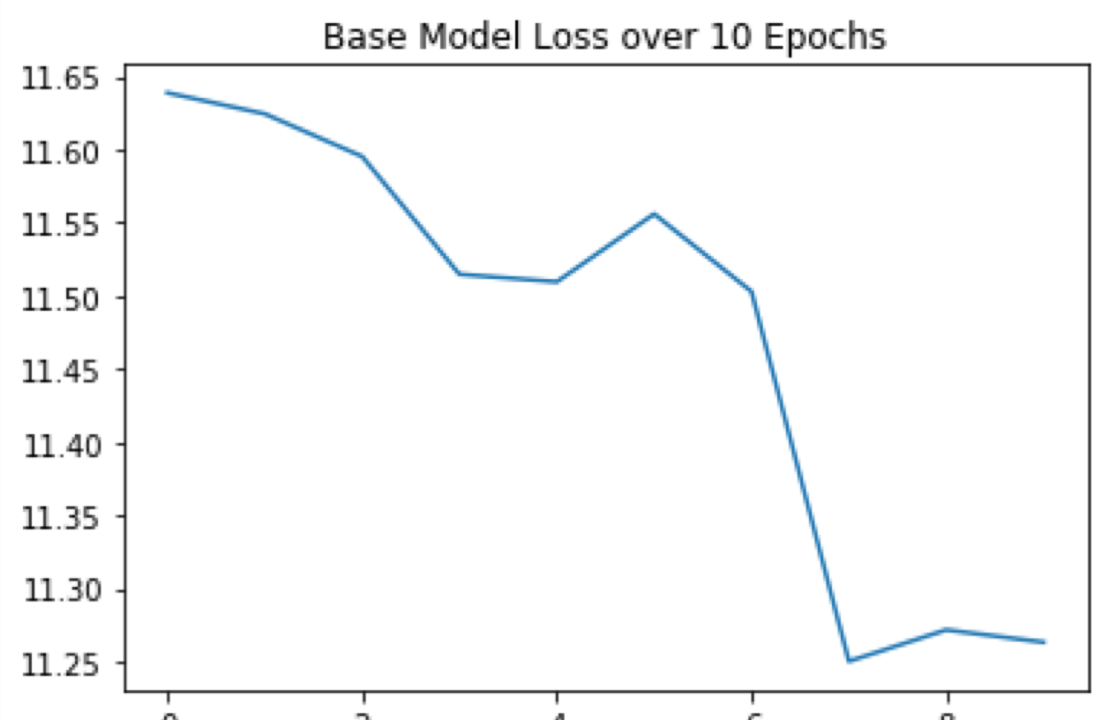
CNNs have convolutional layers that alter the dimensions of input observations, being more computational efficient than regular feed-forward neural networks for data with high dimensionality

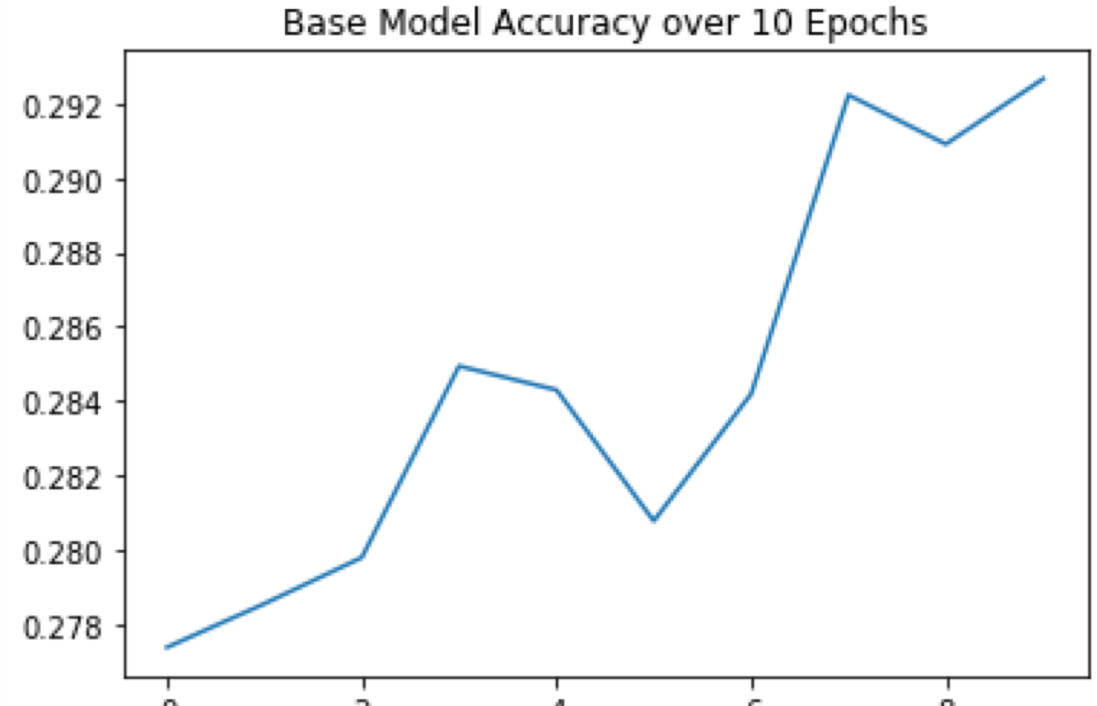
BASE MODEL: LOCAL AND AWS

- First CNN model was created from scratch
- 3 sets of convolutional and pooling layers, then 2 sets of dense layers to predict output

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 176, 216, 32)	320
activation_1 (Activation)	(None, 176, 216, 32)	0
max_pooling2d_1 (MaxPooling2	(None, 88, 108, 32)	0
conv2d_2 (Conv2D)	(None, 86, 106, 32)	9248
activation_2 (Activation)	(None, 86, 106, 32)	0
max_pooling2d_2 (MaxPooling2	(None, 43, 53, 32)	0
conv2d_3 (Conv2D)	(None, 41, 51, 64)	18496
activation_3 (Activation)	(None, 41, 51, 64)	0
max_pooling2d_3 (MaxPooling2	(None, 20, 25, 64)	0
flatten_1 (Flatten)	(None, 32000)	0
dense_1 (Dense)	(None, 64)	2048064
activation_4 (Activation)	(None, 64)	0
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 7)	455
activation_5 (Activation)	(None, 7)	0

Total params: 2,076,583
Trainable params: 2,076,583
Non-trainable params: 0



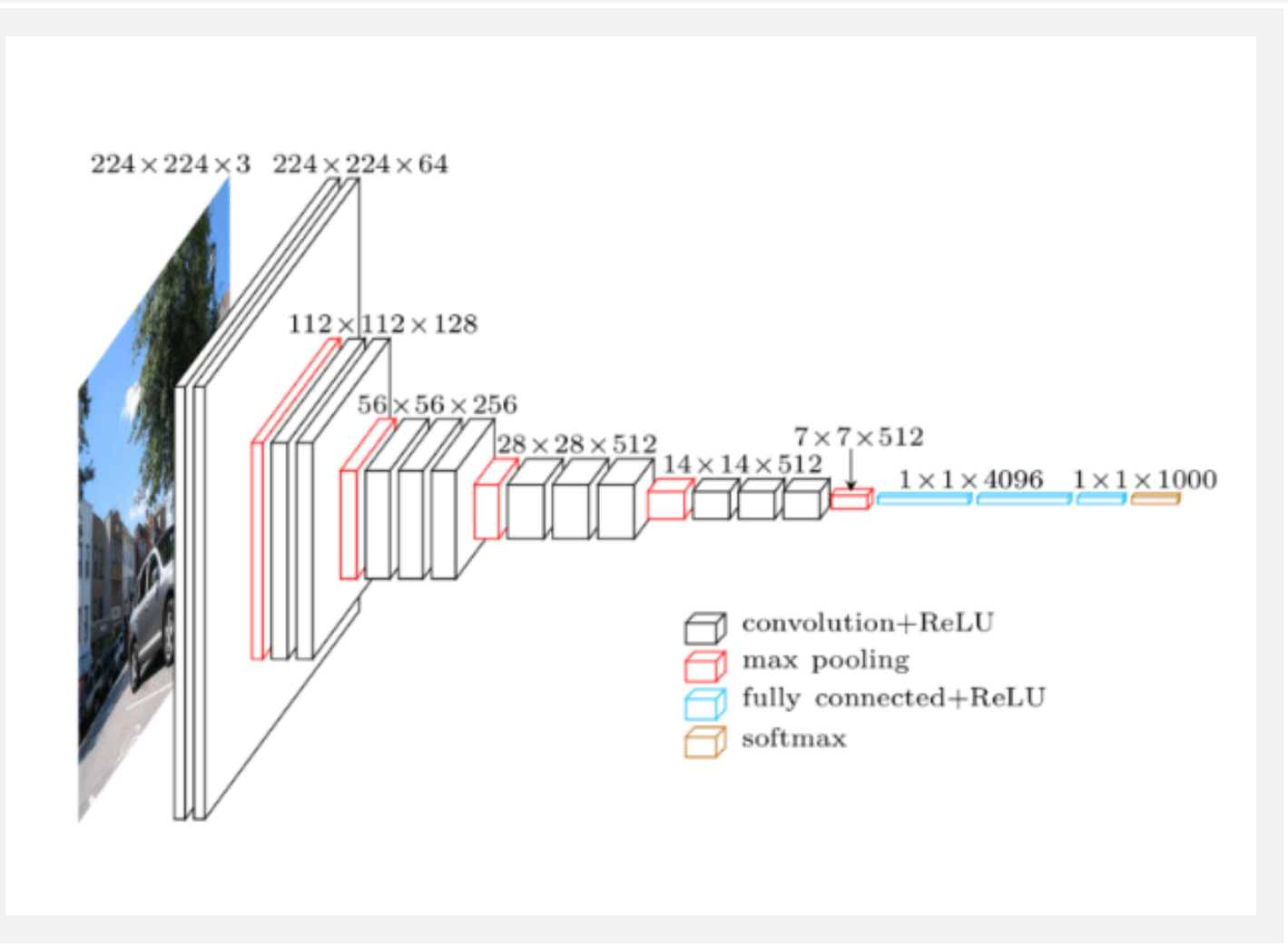


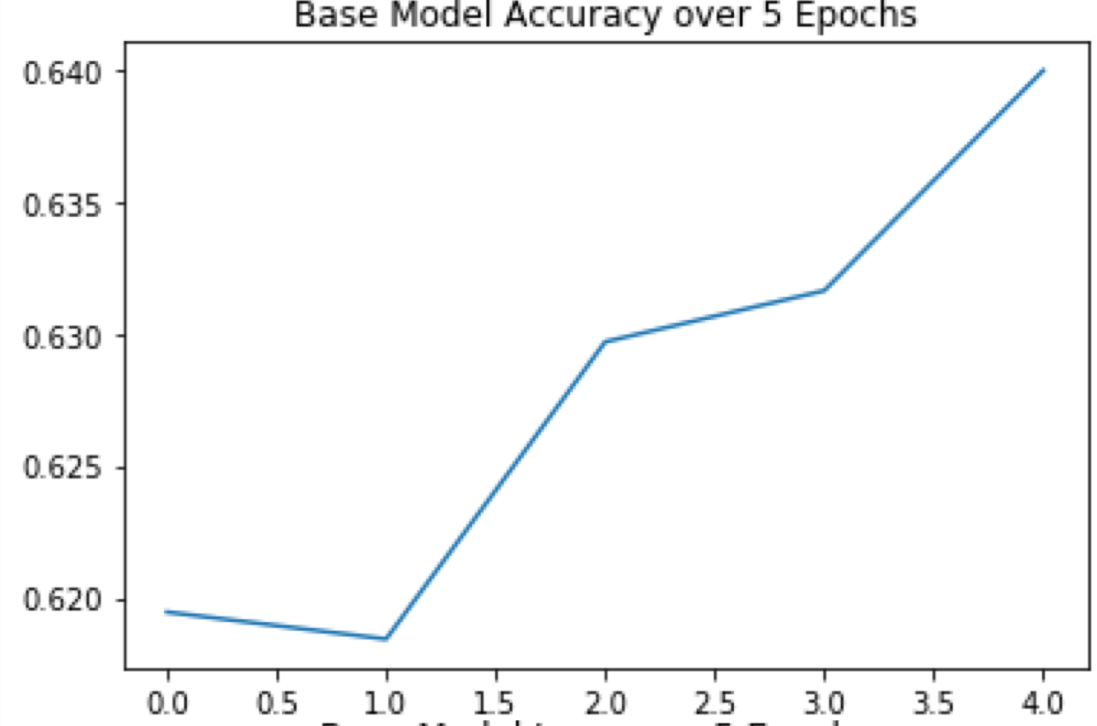
BASE MODEL PERFORMANCE

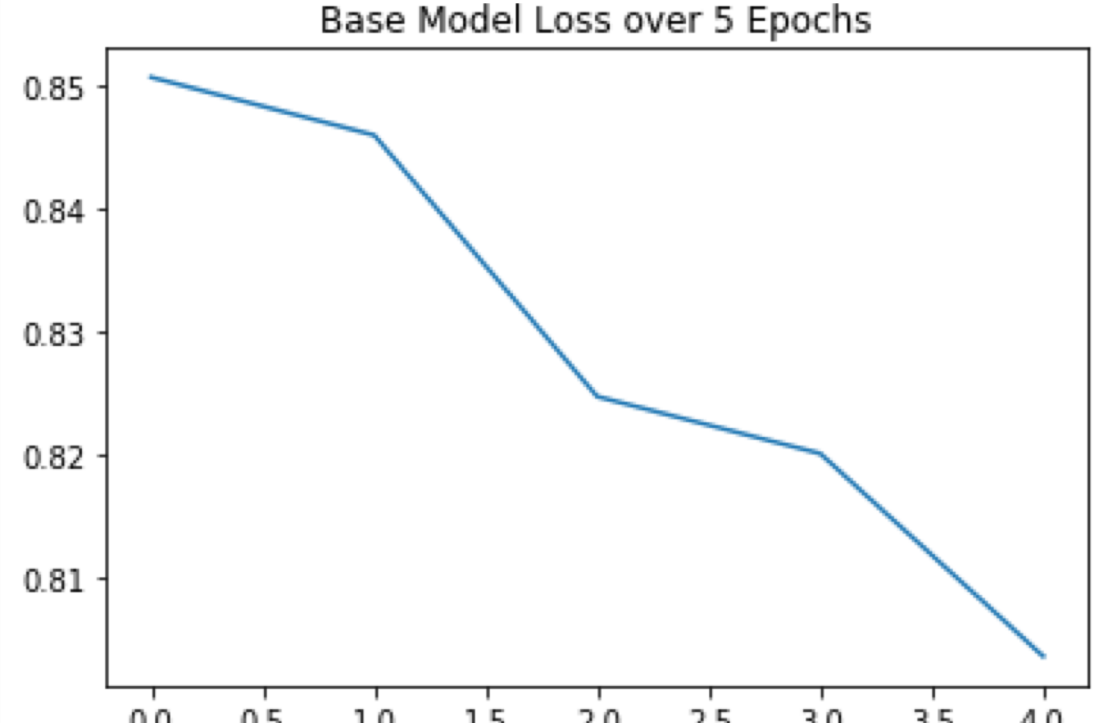
- Leveling off of accuracy at about 29~30%
- This was the baseline, meaning that the base model was not learning anything!

VGG-16 FACE MODEL

- Used a pre-trained model (VGG-16)
- Developed by Oxford University's Visual Geometry Group
- 16 layers deep
- 138 million parameters





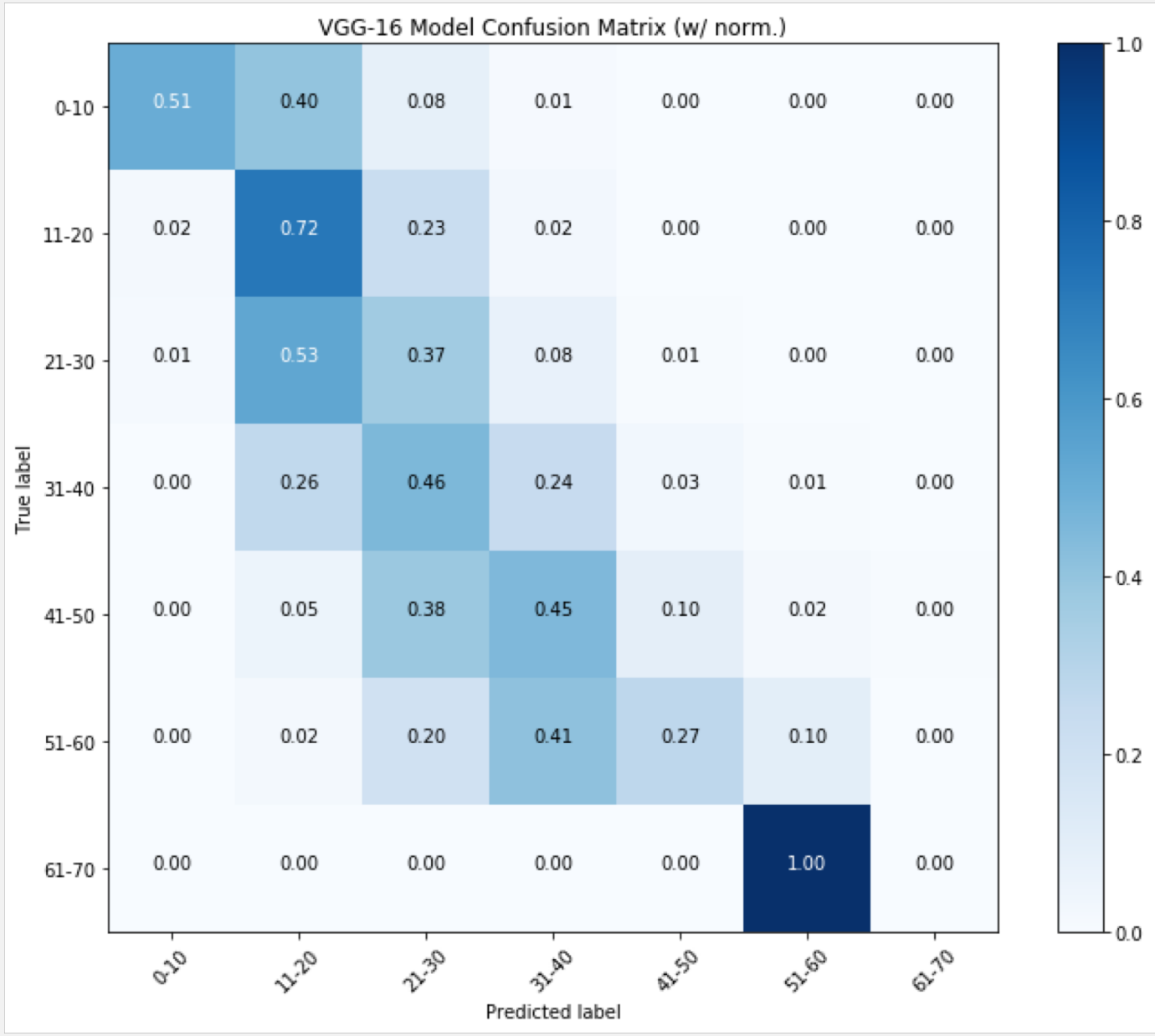


VGG-16 MODEL PERFORMANCE

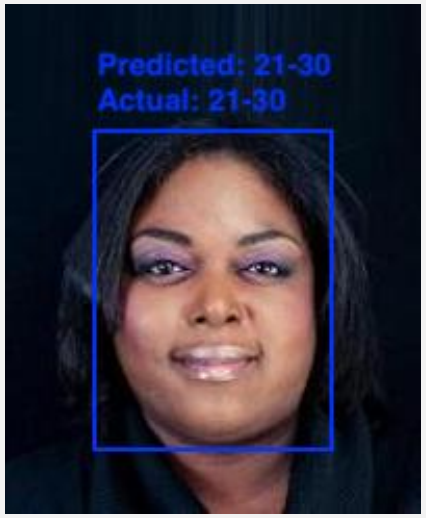
- Even on AWS, each epoch took 30 minutes (each run of the model was 5 or 10 epochs)
- On first run, the model was still learning nothing!
- After research, I changed batch size from 16 to 32 (which controls learning rate) and model started to learn
- After 20 epochs, train accuracy of 64% and cross-entropy of 0.8036

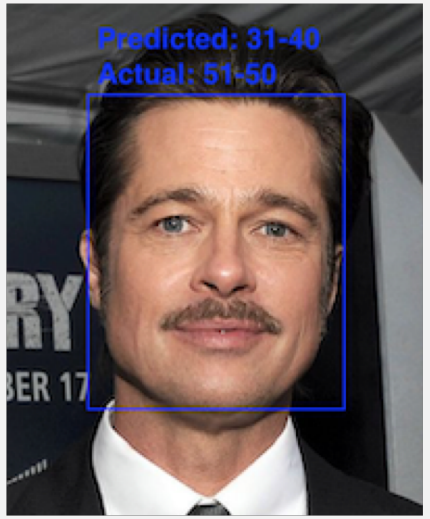
EVALUATING THE MODEL

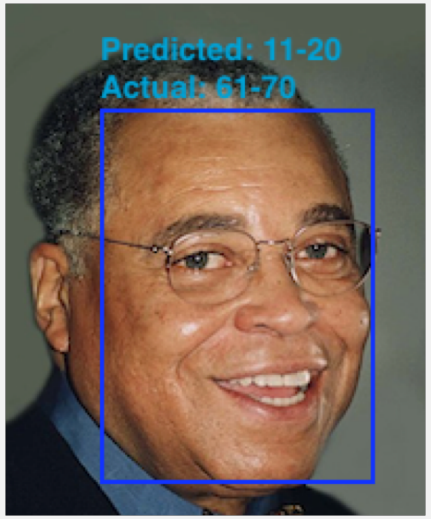
- To see if the model was learning, I created a confusion matrix
- One-Off accuracy metric: 70%
- Learning was stunted due to high class imbalance



FINAL PREDICTIONS







NEXT STEPS

- Use a live video feed via OpenCV to classify someone's age group based on their face (perhaps using a computer's webcam)
- Create a regression problem where the model predicts age, as opposed to age group
- implementing a validation training set (instead of only having a train and test split)

THANK YOU!