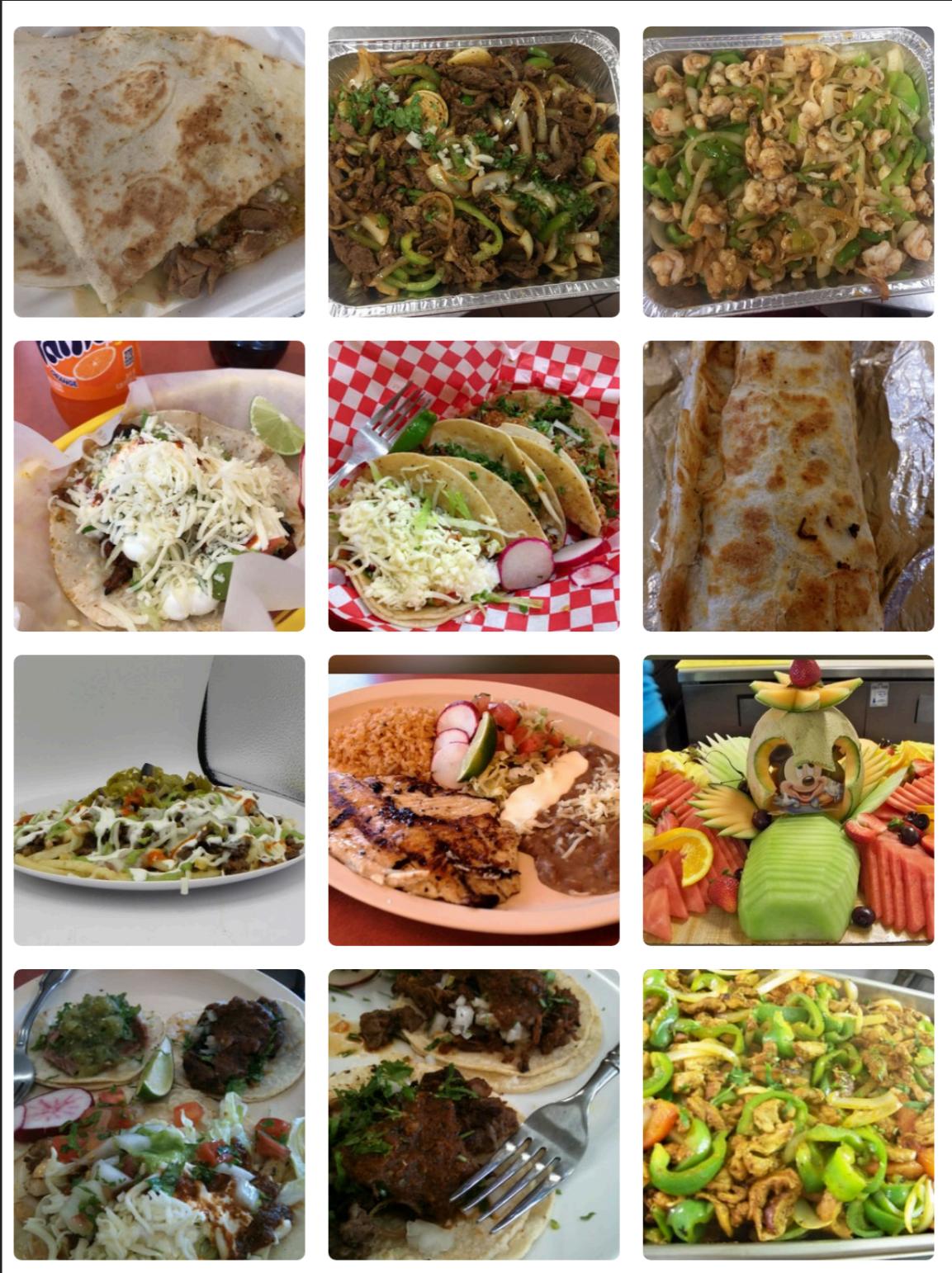


A BETTER SORT OF  
YELP PHOTOS

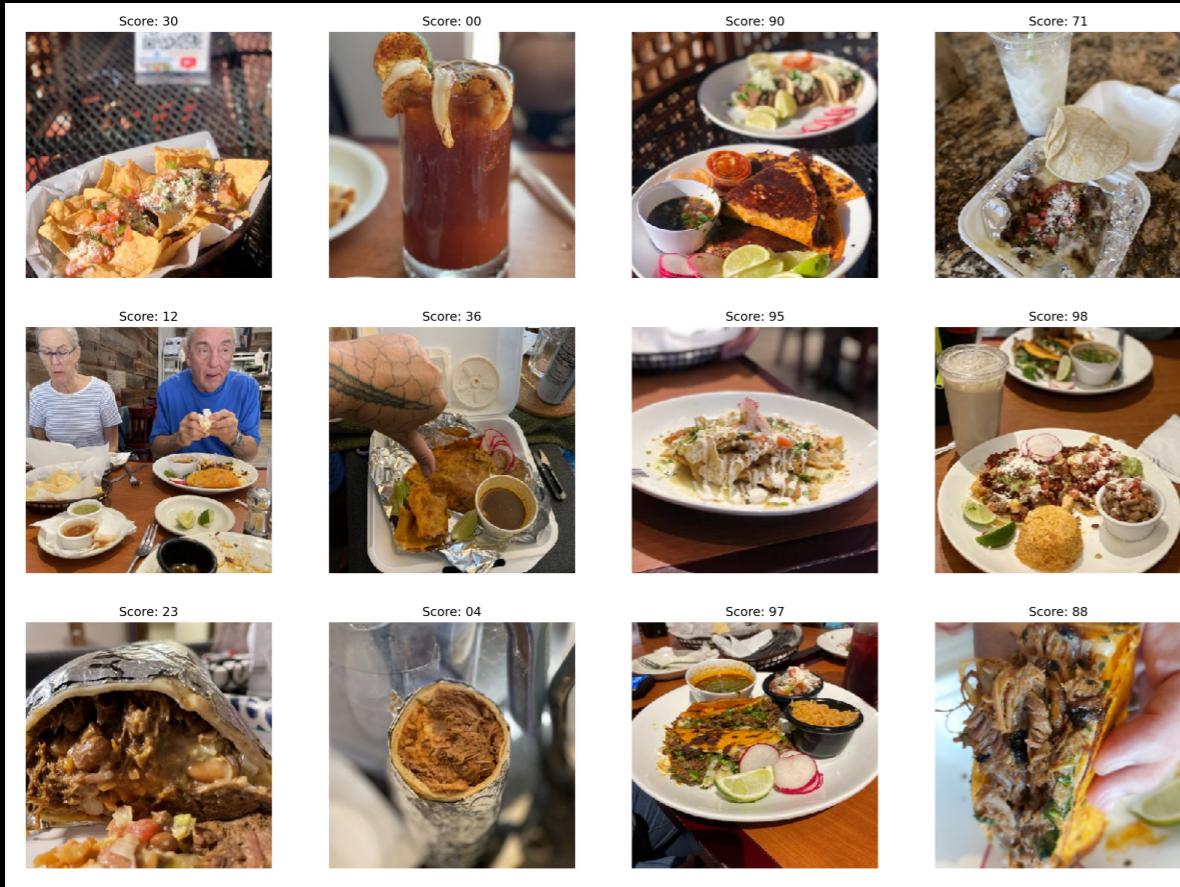
# PROBLEM STATEMENT

Why are the photos in many Yelp restaurant galleries still poor quality? App users and owners deserve a better sort option.

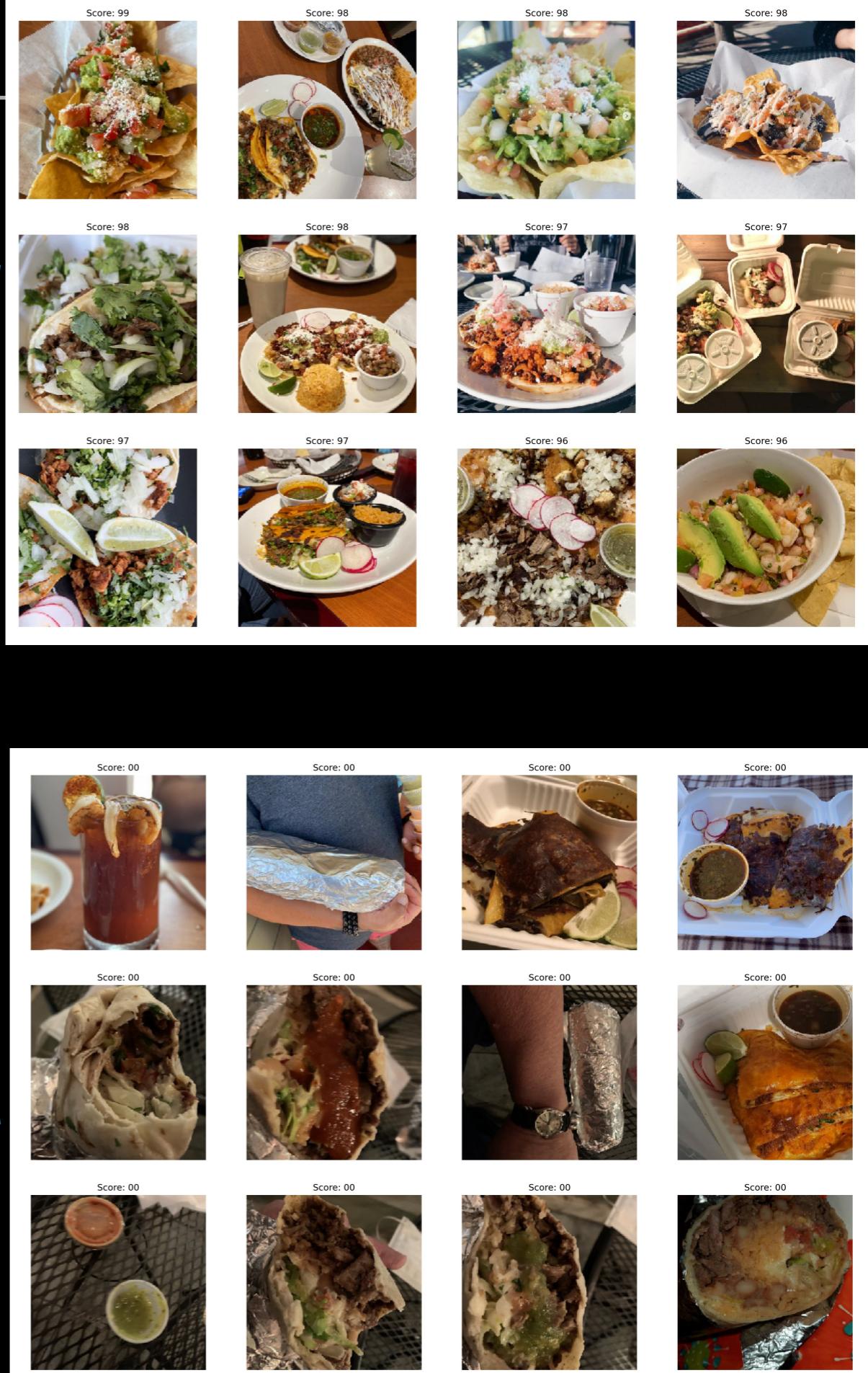


# USING MACHINE LEARNING TO SORT

SORT HIGH QUALITY PHOTOS TO THE TOP.



POOR QUALITY TO THE BOTTOM



# BACKGROUND



[Sources](#)

# METHODS

# PHOTO CRITERIA

- ▶ Focus: Subject in focus.

Label: [good]



Label: [bad]



# PHOTO CRITERIA

- ▶ Focus: Subject in focus.
- ▶ Exposure: Even

Label: [good]



Label: [bad]



# PHOTO CRITERIA

- ▶ Focus
- ▶ Exposure
- ▶ Subject
- ▶ Label: [good]
- ▶ Label: [bad]



# PHOTO CRITERIA

- ▶ Focus
- ▶ Exposure
- ▶ Subject
- ▶ Color

Label: [good]



Label: [bad]



# PHOTO CRITERIA

- ▶ Focus
- ▶ Exposure
- ▶ Subject
- ▶ Color
- ▶ Pattern / Composition

Label: [good]



Label: [bad]



# Photos for



1211 reviews

[+ Add photos](#)

All (633)

**All Food (454)**

Tacos (190)

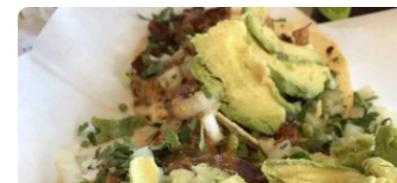
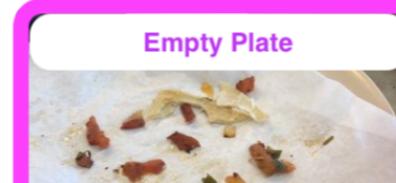
Burritos &amp; Wraps (111)

Inside (89)

Menu (26)

Or &gt;

Search photos...



# DATA VISUALIZATION

Good



Good



Good



Bad



Bad



Bad



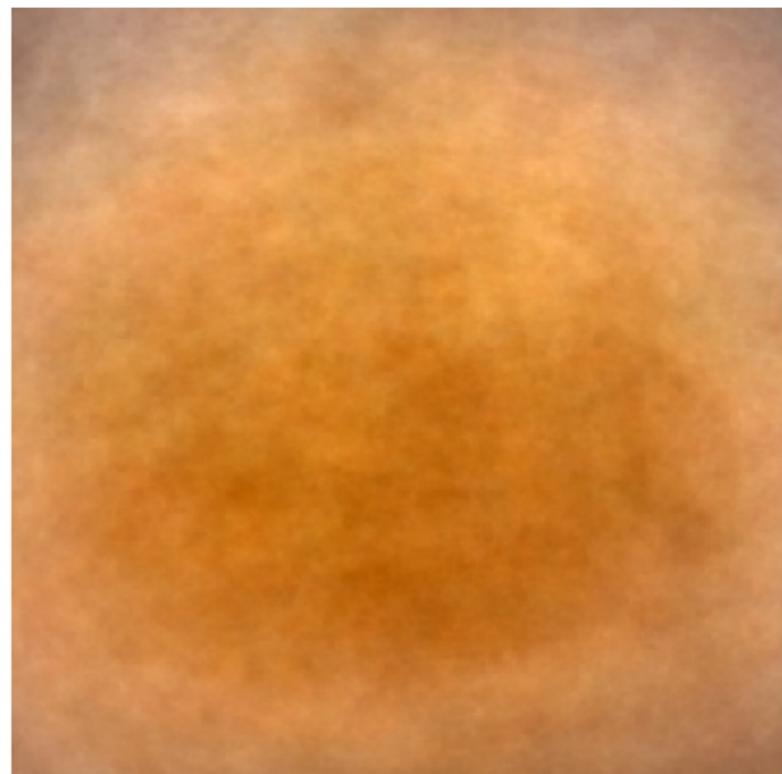
# DATA VISUALIZATION

- ▶ Composite image for each class (mean values)

Composite "Good" Image  
(Mean Values)



Composite "Bad" Image  
(Mean Values)



# DATA VISUALIZATION

- ▶ Average Grayscale Image Per Class

Average Good Image



Average Bad Image



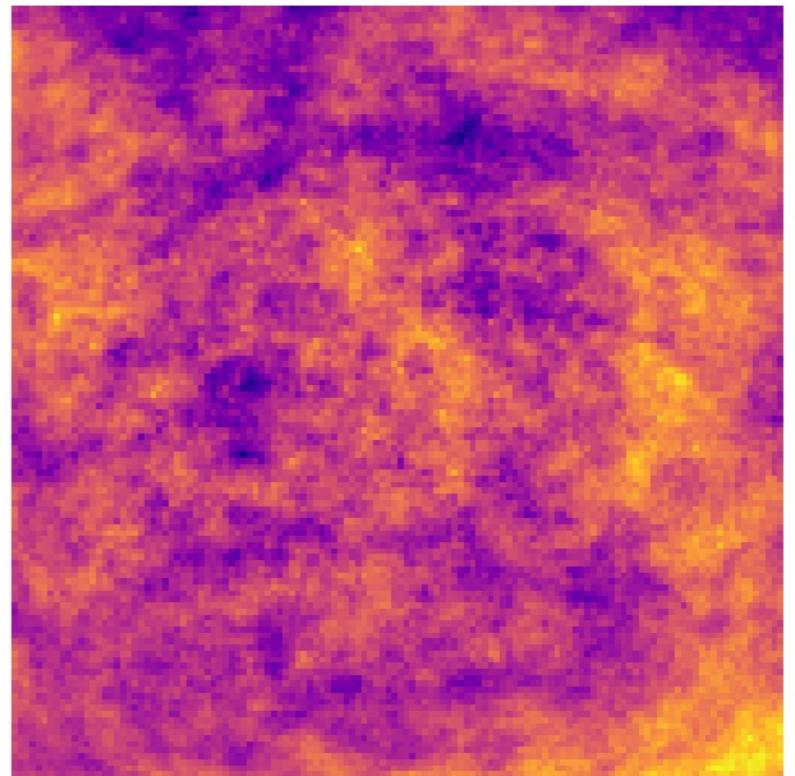
Perceptually Uniform Sequential colormaps



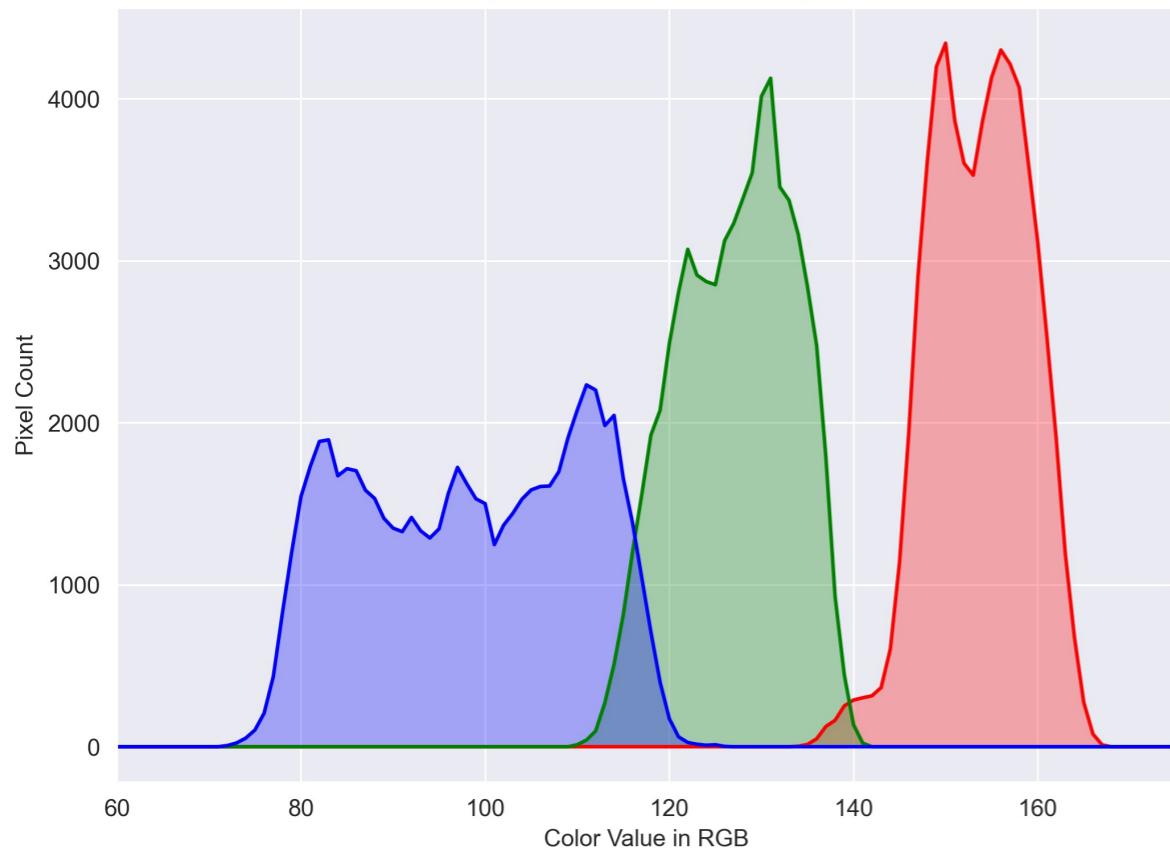
# DATA VISUALIZATION

- ▶ Contrast Image (Difference Between Average “Good” Image And Average “Bad” Image)

Difference Between Good and Bad Photo



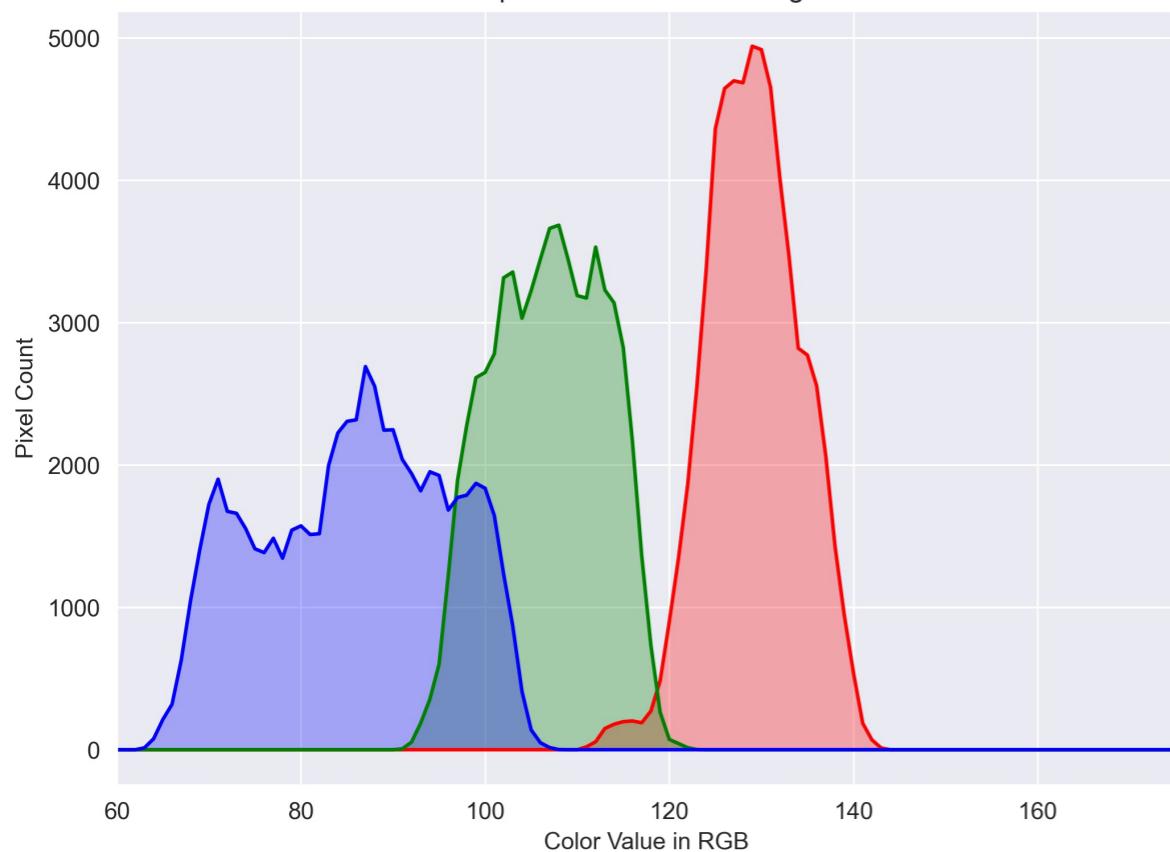
Composite Good Photo Histogram



# DATA VISUALIZATION

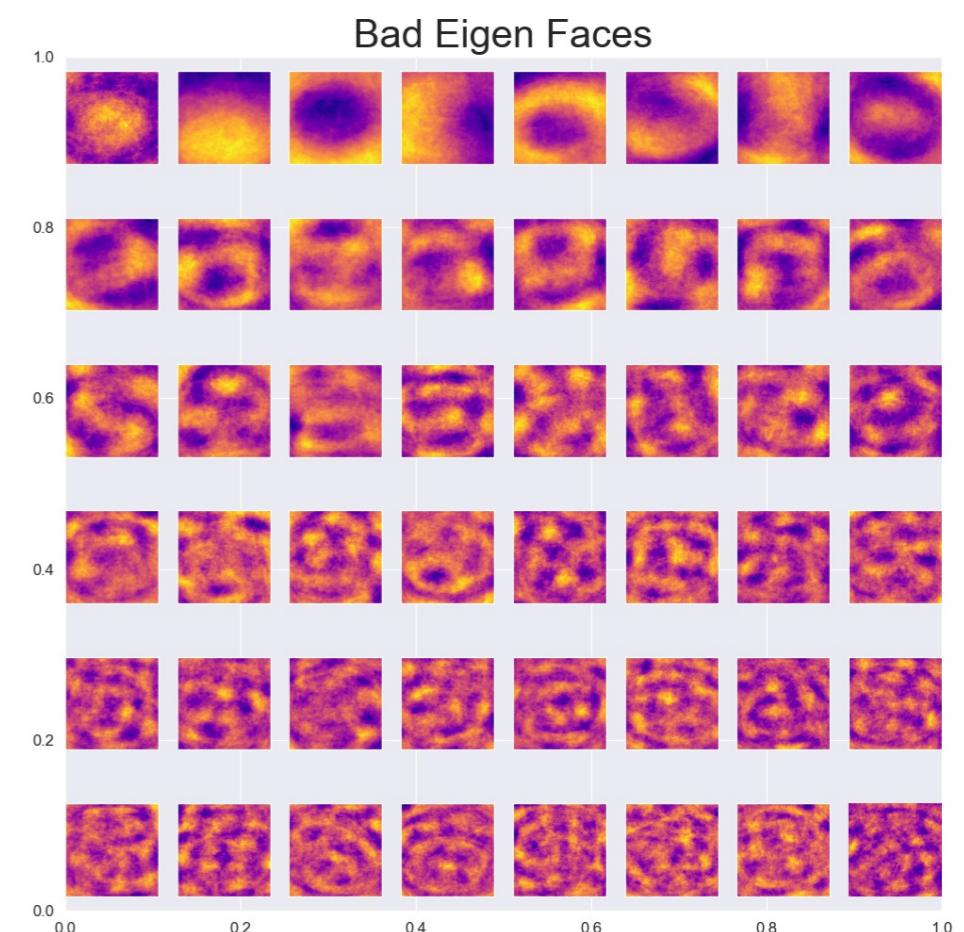
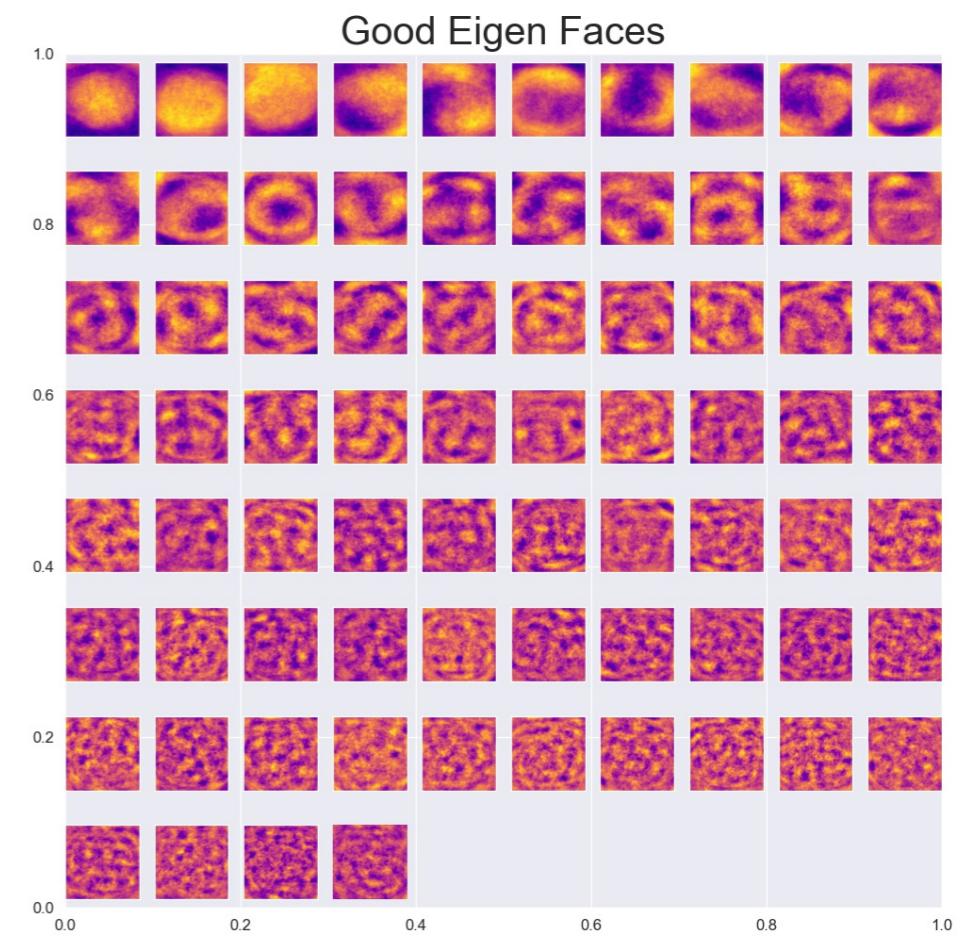
- ▶ Composite Histograms for Each Class

Composite Bad Photo Histogram



# DATA VISUALIZATION

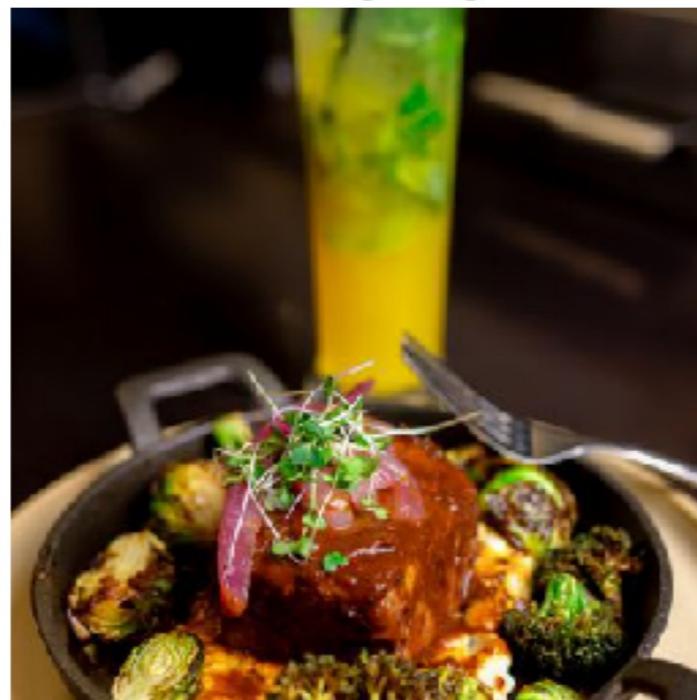
- ▶ Principle Components Analysis
  - ▶ Makes Composites of Each Class That Explain 70% of the Variance
  - ▶ Sometimes Called "Eigen Faces"; used in facial recognition research.



Good Photo High Eigen Value



Good Photo High Eigen Value



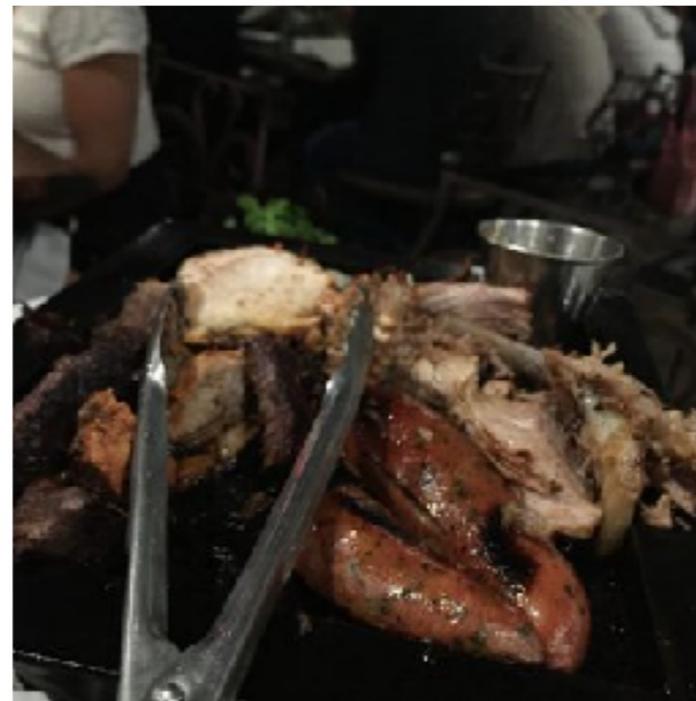
Good Photo High Eigen Value



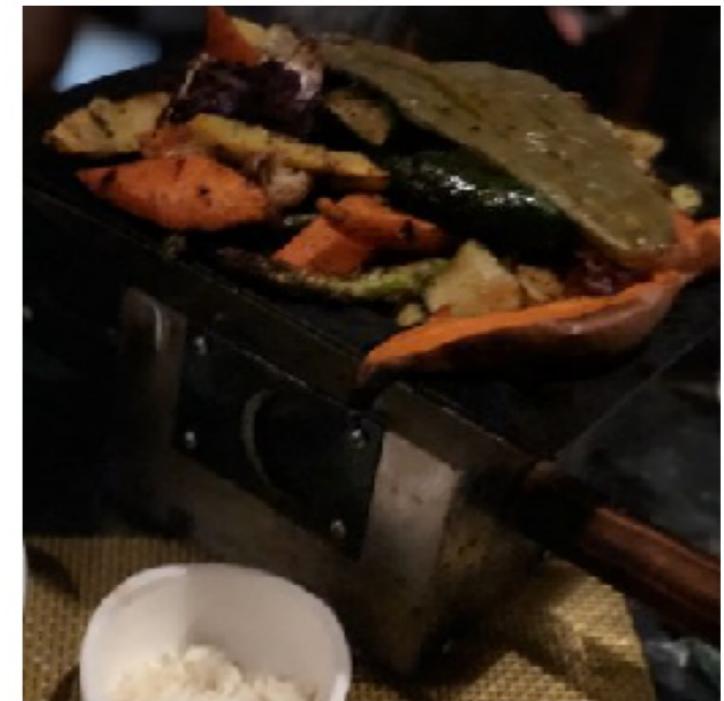
Bad Photo High Eigen Value



Bad Photo High Eigen Value



Bad Photo High Eigen Value



---

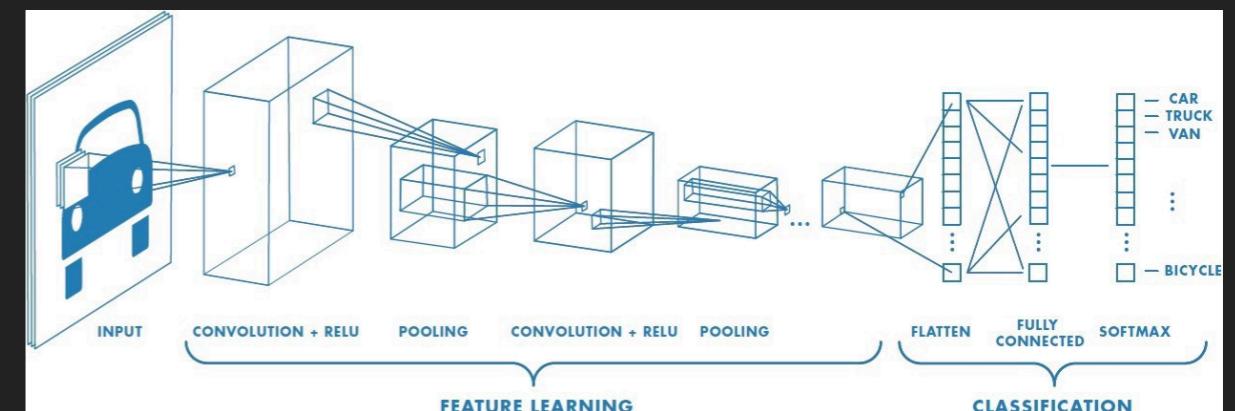
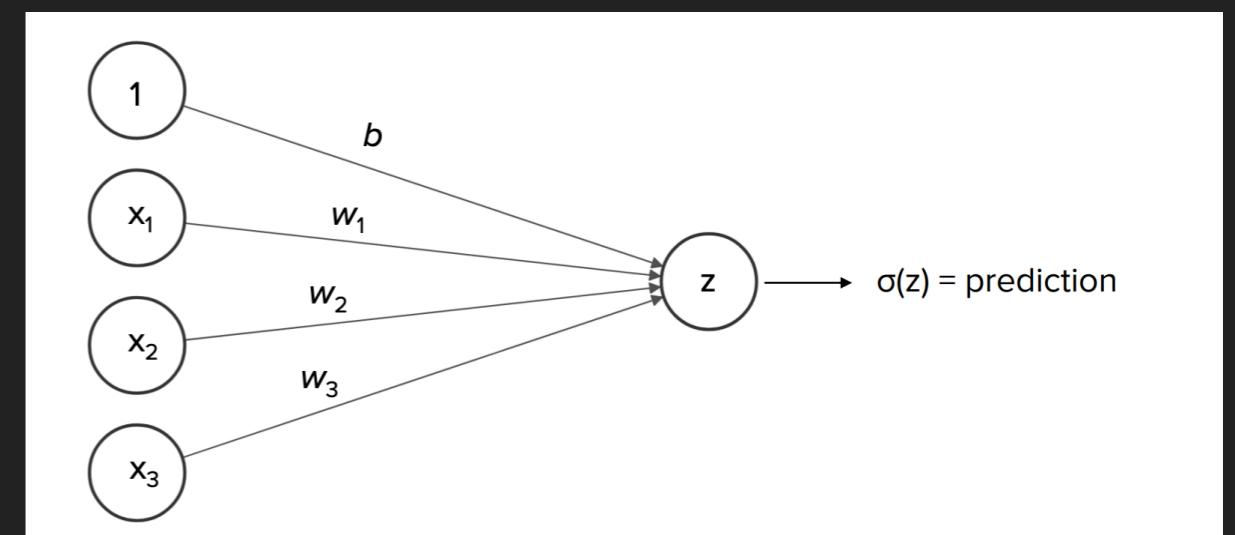
# MODELING

# CONVOLUTIONAL NEURAL NETS (CNN)

- ▶ What are they?
- ▶ CNNs are some of the best classifiers for images
- ▶ Two Methods Used. Two Models for EACH:
  - CNN from Scratch (Model 1, Model 2)
  - CNN with Transfer Learning (Model 3, Model 4)

$$p = \sigma(b + w_1x_1 + w_2x_2 + w_3x_3)$$

Curvy part      Linear part



# CNN IMAGE CLASSIFIER FROM SCRATCH

## MODEL 1: 76% ACC

Batch size: 16, Image size: 256x256

Model: "sequential\_5"

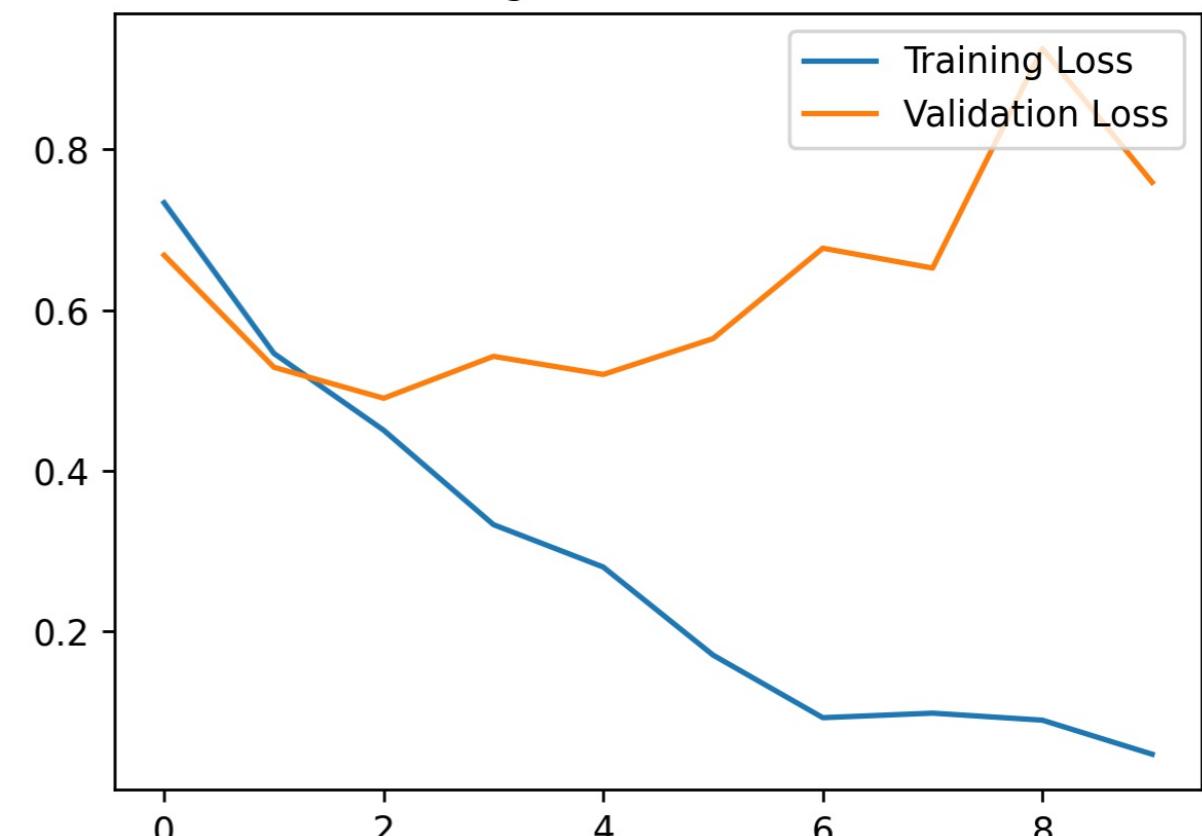
Layer (type)	Output Shape	Param #
<hr/>		
rescaling_8 (Rescaling)	(None, 256, 256, 3)	0
conv2d_15 (Conv2D)	(None, 256, 256, 16)	448
max_pooling2d_15 (MaxPooling2D)	(None, 128, 128, 16)	0
conv2d_16 (Conv2D)	(None, 128, 128, 32)	4640
max_pooling2d_16 (MaxPooling2D)	(None, 64, 64, 32)	0
conv2d_17 (Conv2D)	(None, 64, 64, 64)	18496
max_pooling2d_17 (MaxPooling2D)	(None, 32, 32, 64)	0
flatten_5 (Flatten)	(None, 65536)	0
dense_10 (Dense)	(None, 128)	8388736
dense_11 (Dense)	(None, 1)	129
<hr/>		

Total params: 8,412,449

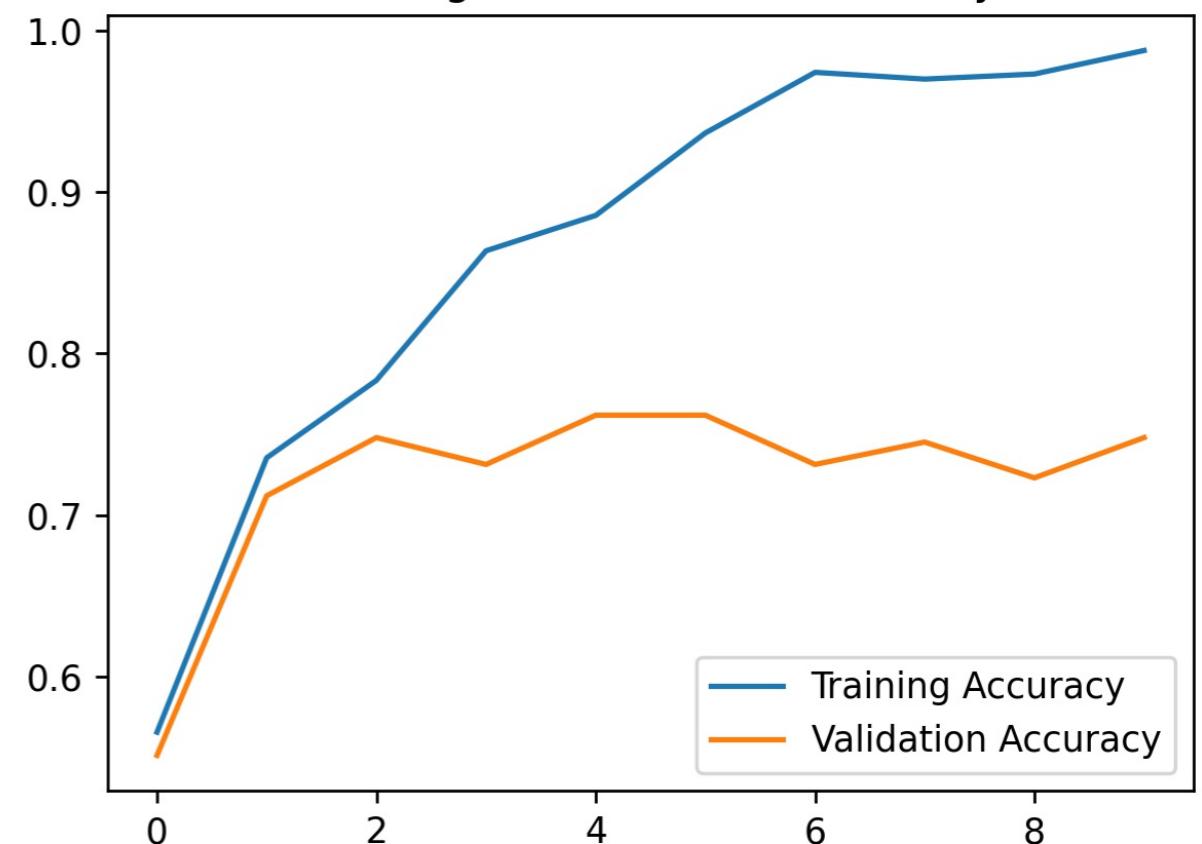
Trainable params: 8,412,449

Non-trainable params: 0

Training and Validation Loss



Training and Validation Accuracy



# CNN IMAGE CLASSIFIER FROM SCRATCH

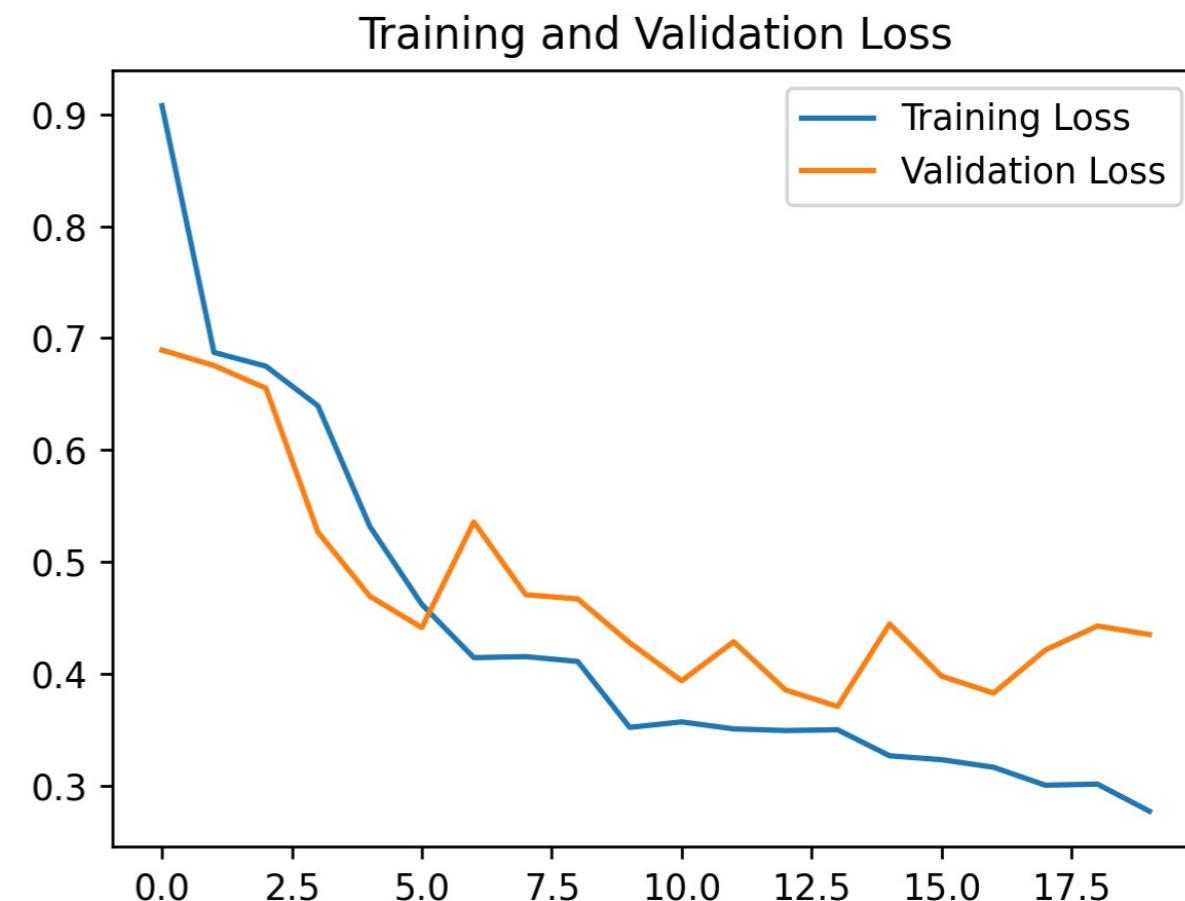
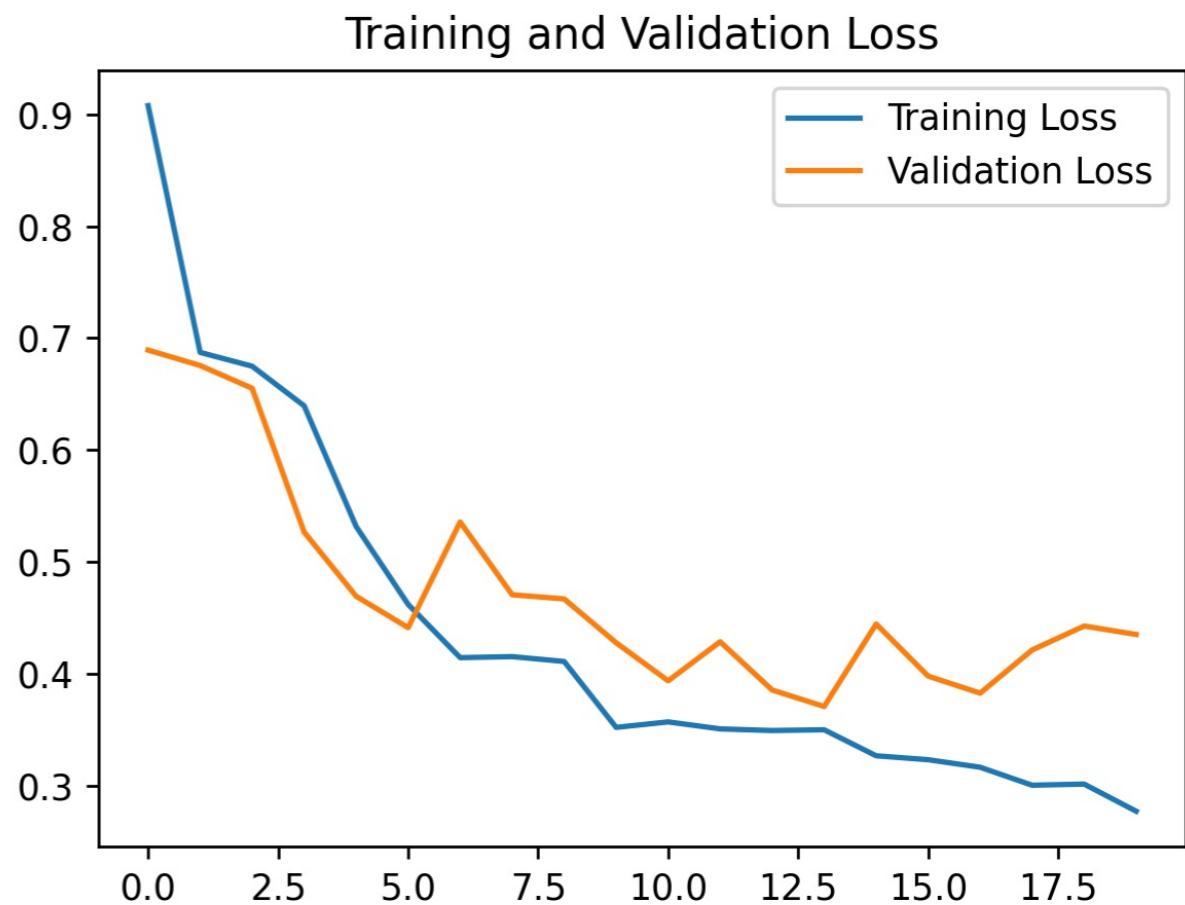
## MODEL 2: 82% ACC

Batch size: 16, Image size: 256x256

Adds Image Augmentation & Dropout

Model: "sequential\_8"

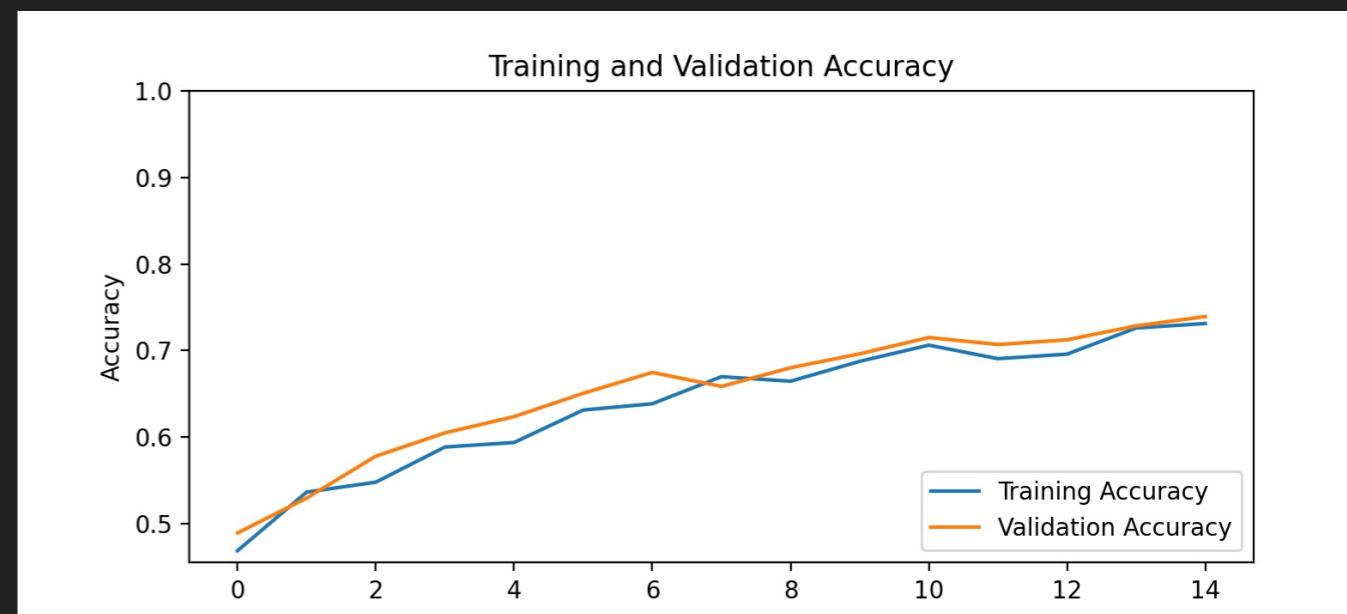
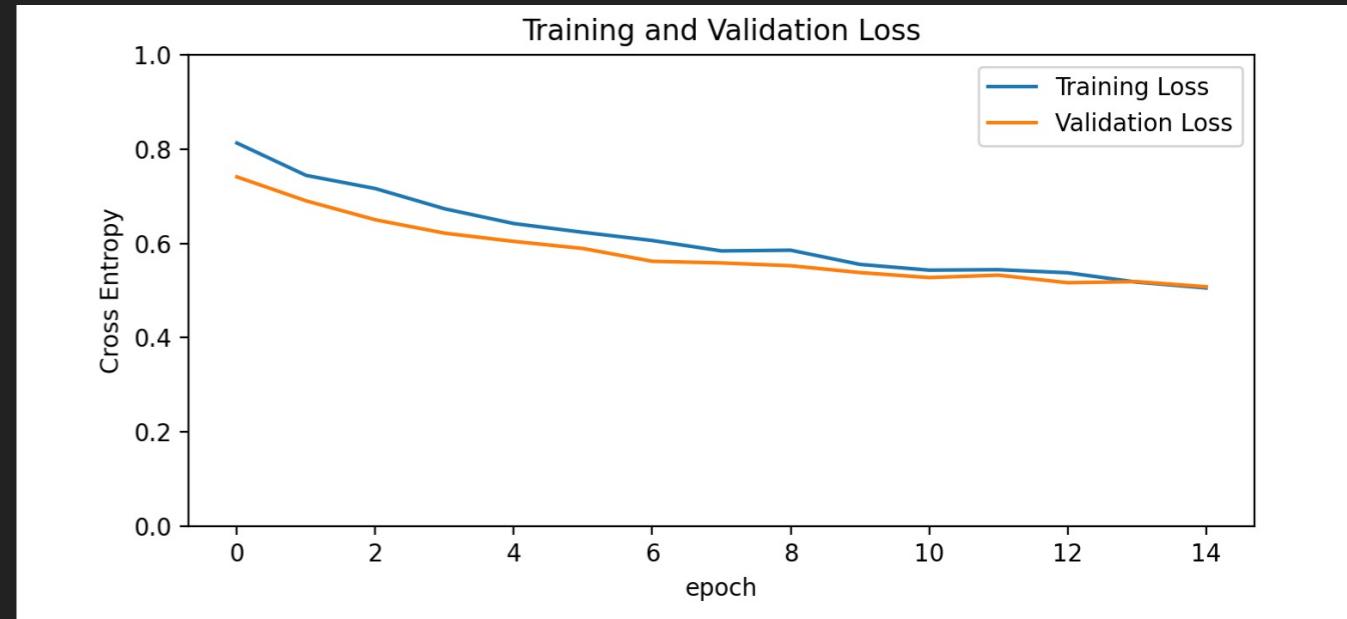
Layer (type)	Output Shape	Param #
sequential_7 (Sequential)	(None, 256, 256, 3)	0
rescaling_9 (Rescaling)	(None, 256, 256, 3)	0
conv2d_18 (Conv2D)	(None, 256, 256, 16)	448
max_pooling2d_18 (MaxPooling2D)	(None, 128, 128, 16)	0
conv2d_19 (Conv2D)	(None, 128, 128, 32)	4640
max_pooling2d_19 (MaxPooling2D)	(None, 64, 64, 32)	0
conv2d_20 (Conv2D)	(None, 64, 64, 64)	18496
max_pooling2d_20 (MaxPooling2D)	(None, 32, 32, 64)	0
dropout (Dropout)	(None, 32, 32, 64)	0
flatten_6 (Flatten)	(None, 65536)	0
dense_12 (Dense)	(None, 128)	8388736
dense_13 (Dense)	(None, 1)	129
<hr/>		
Total params:	8,412,449	
Trainable params:	8,412,449	
Non-trainable params:	0	



## MODEL 3: ~73% ACC

~73% on Validation Data after 15 epochs

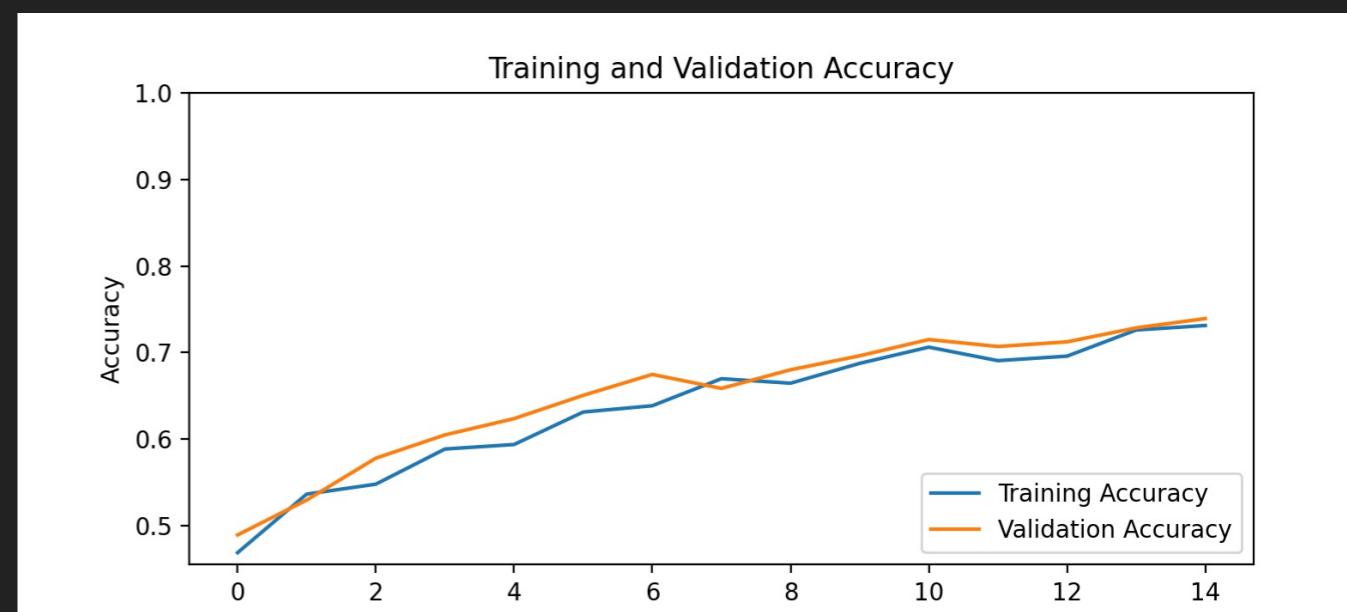
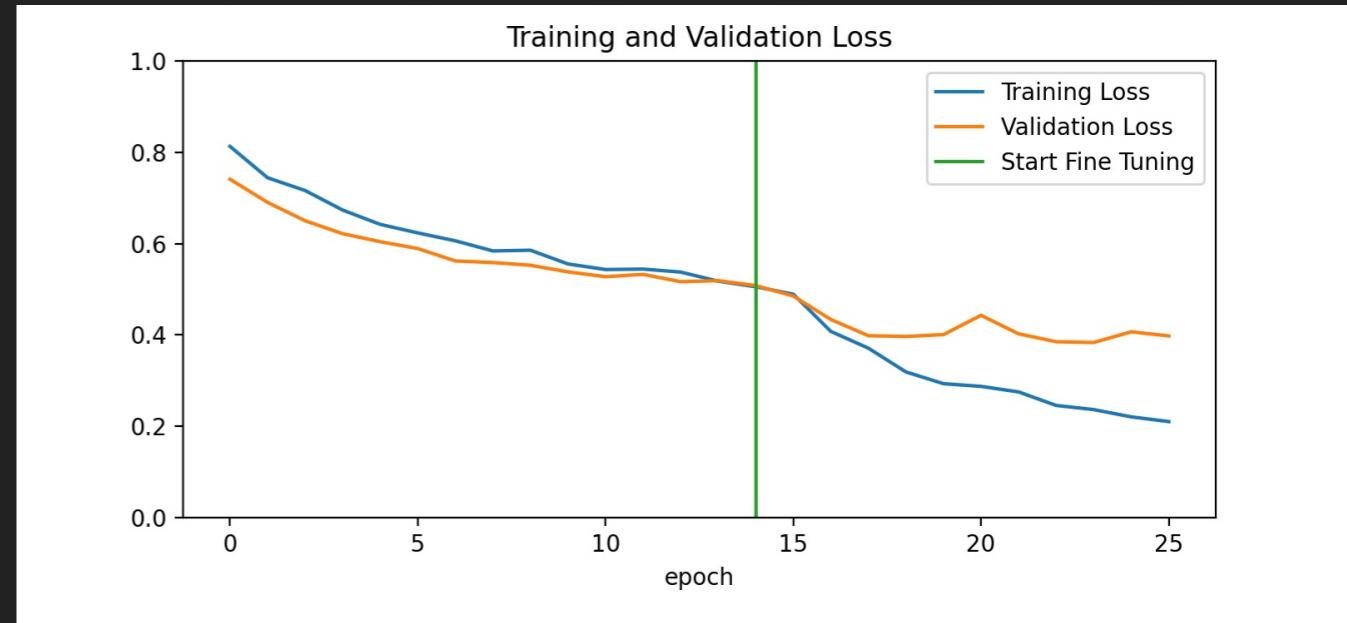
- ▶ A feature extraction transfer learning model
- ▶ Uses MobileNet V2 from Google (1.4 M photos, 1000 classes)
- ▶ Takes the top off and applies to this data.



## MODEL 4: ~83% ACC

~83% on Validation Data after 10 epochs

- ▶ A fine-tuning transfer learning model
- ▶ Unfreezes a few of the top layers from the MobileNet V2
- ▶ Lets you take advantage of basic features in lower part of pre-train, then get more specific for your data.



Label: good Predicted: good



Label: good Predicted: good



Label: good Predicted: good



Label: bad Predicted: bad



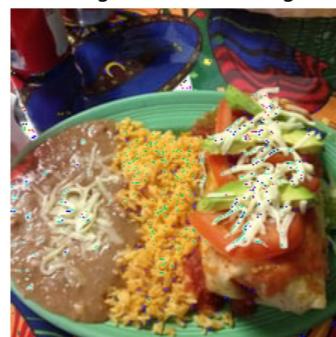
Label: good Predicted: good



Label: bad Predicted: bad



Label: good Predicted: good



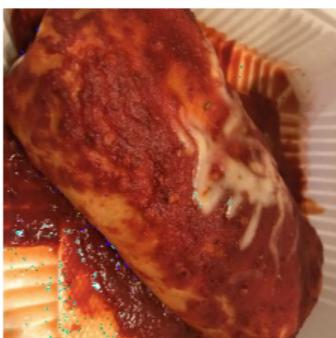
Label: bad Predicted: bad



Label: bad Predicted: bad



Label: bad Predicted: bad



Label: bad Predicted: bad



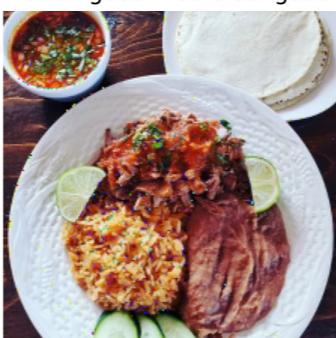
Label: bad Predicted: bad



Label: good Predicted: good



Label: good Predicted: good



Label: good Predicted: bad



Label: good Predicted: good



---

# CONCLUSIONS

## CONCLUSIONS

- ▶ More difficult to evaluate subjective labels.
- ▶ Sometimes CNNs with transfer learning don't perform better than those without.
- ▶ CNNs: are like poker.
- ▶ Try limiting down the dataset to more specific shapes(e.g. sushi).
- ▶ Try a CNN w/ specific architecture for color patterns.



---

# RECOMMENDATIONS



## RECOMMENDATIONS

- ▶ Review platforms with user generated photo galleries should consider implementing AI-based photo-quality sorting.
- ▶ This idea has other applications beyond food; probably many like this photo website are already in production: [www.pickpik.com](http://www.pickpik.com)