From Jello Molds to Surf & Turf: Culinary Deep Learning

By Sam Beardsley

PROJECT OVERVIEW

- Food is not just for nutrition, but it is also culturally important
- The hashtags #food, #foodporn, #instafood, #yummy and #foodie have been used over 1 billion times[1] and the volume of food blogs and cookbooks continue to grow despite expectations[2]
- Image datasets of food exist but they don’t capture the quality of the food image
- Having a tool that can automate the evaluation of food photography quality is beneficial for professional photographers and content creators such as bloggers

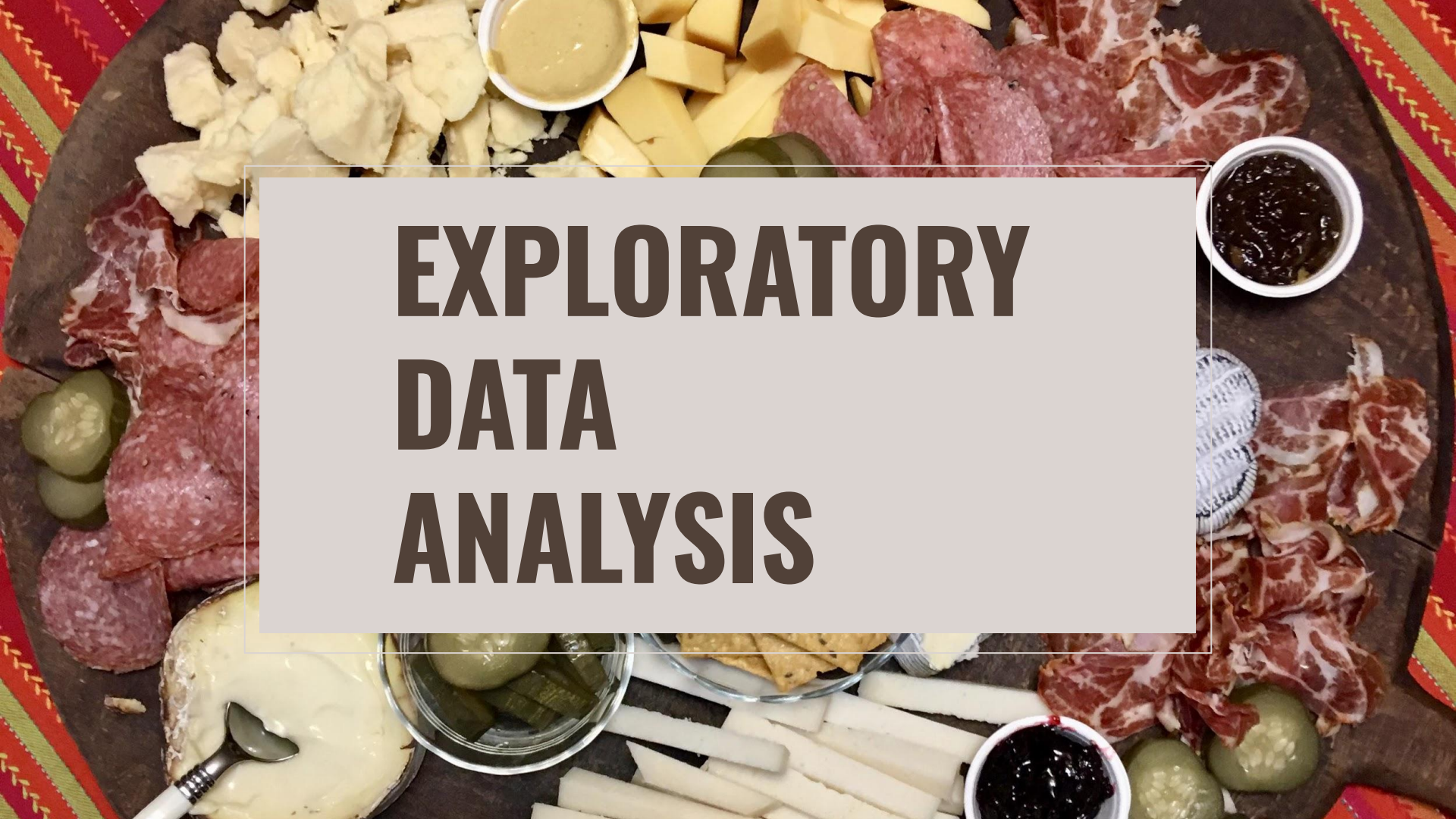
THE DATA

DATA WRANGLING

- Images sourced from the the /r/foodporn (FP) and /r/shittyfoodporn (SFP) Reddit communities
- Used the PushShift API to collect data on 333,866 FP and 317,486 SFP image posts between 1/1/15 and 9/30/20
- This was then narrowed down to the 20th percentile based on number of votes and those images were scraped from Reddit
- The final data set had 22,899 FP and 18,682 SFP images
- All images were resized to 224x224

IMAGE SELECTION

- Exploratory Data Analysis
  - Top 500 images from each subreddit by votes
- Model Selection
  - 4,500 training images, 1,000 validation images
  - Batch Size = 32
- Training and Fine-Tuning
  - 13,000 training images (including the previous 4,500)
  - The same 1,000 validation images
  - 1,000 test images
  - Batch Size = 32



EXPLORATORY DATA ANALYSIS

FP - SAMPLE IMAGES



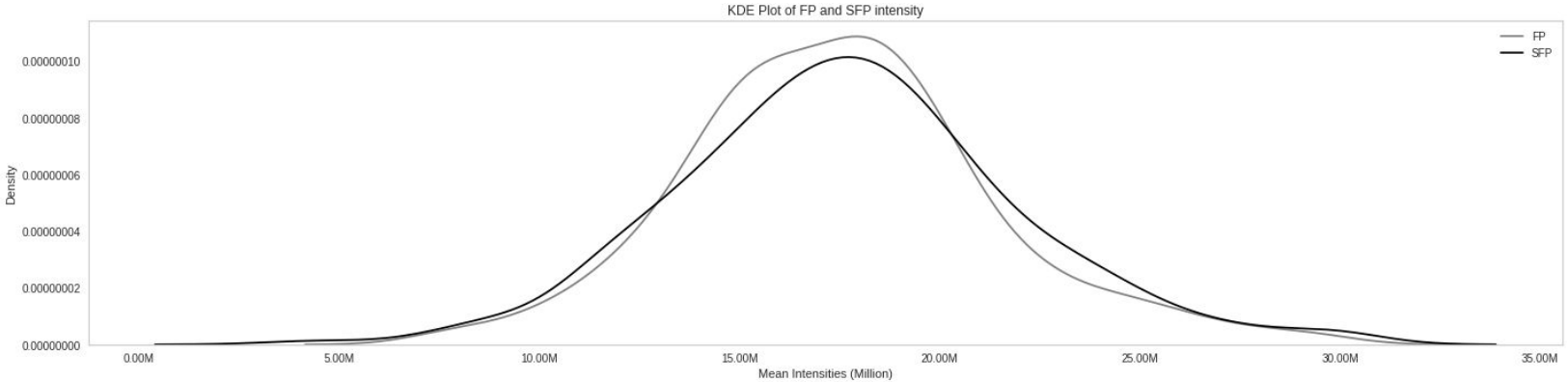


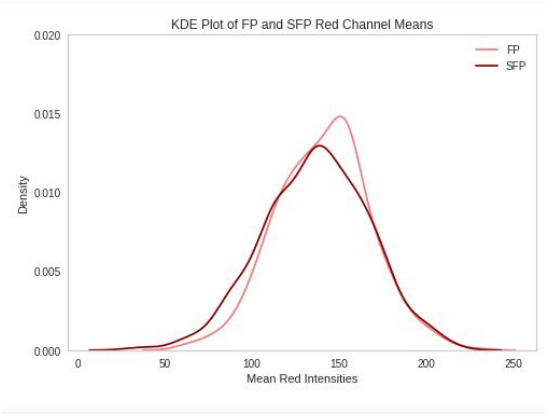


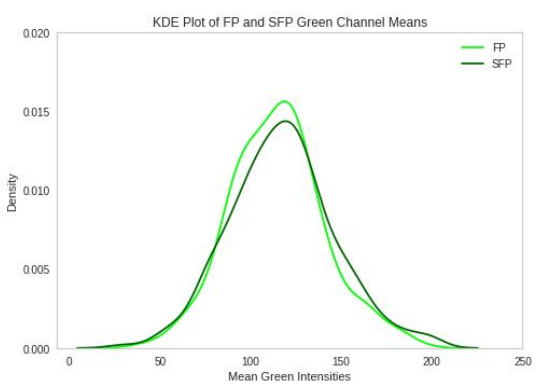


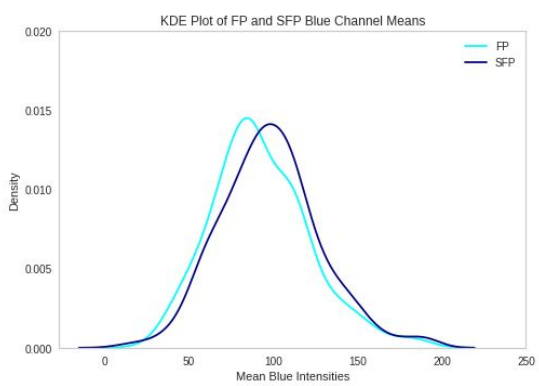
SFP - SAMPLE IMAGES

INTENSITIES





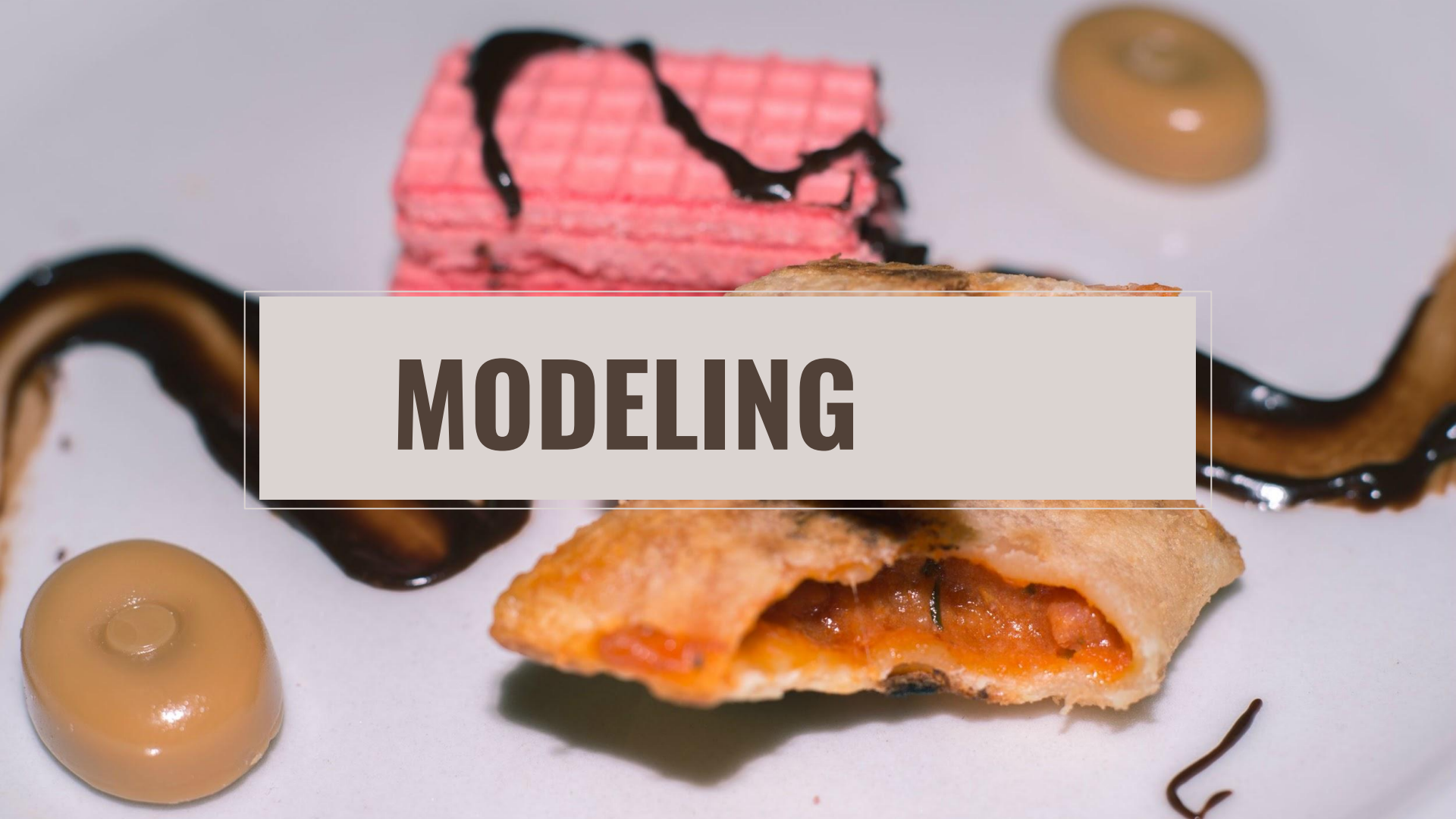




AVERAGE IMAGES

FP

SFP


MODELING

BASELINE MODEL

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
rescaling (Rescaling)	(None, 224, 224, 3)	0
conv2d (Conv2D)	(None, 222, 222, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 111, 111, 64)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	36928
max_pooling2d_1 (MaxPooling2	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 32)	18464
global_average_pooling2d (Gl	(None, 32)	0
flatten (Flatten)	(None, 32)	0
dense (Dense)	(None, 10)	330
dropout (Dropout)	(None, 10)	0
dense_1 (Dense)	(None, 1)	11

Total params: 57,525
Trainable params: 57,525
Non-trainable params: 0

- Comparison point to use with state of the art models
- Optimizer = Adam
- Loss Function = Binary Cross Entropy

TRANSFER LEARNING

- Developing and training neural networks from scratch requires significant trial and error and is computationally expensive
- Transfer learning bypasses these costs by using the weights from state of the art models pre-trained on large and diverse data sets as a starting point
- Models used: VGG16, ResNet50, InceptionV3
    - Pretrained on ImageNet
    - Optimizer = Adam
    - Loss Function = Binary Cross Entropy

HYPERBAND TRIALS

- Hyperband is an hyperparameter selection algorithm that adaptively allocates resources to focus on promising results while quickly eliminating unpromising ones.
- Each model conducted 30 trials with 4,500 training images and 1,000 validation images.

PARAMETER SPACE

Parameter	Values
Dropout Rate	0.0, 0.1, 0.2, 0.3, 0.4, 0.5
Learning Rate	0.1, 0.01, 0.001, 0.0001

RESULTS

Model	Best Parameters
Baseline	Dropout = 0.1 Learning Rate = 0.01
VGG16	Dropout = 0.3 Learning Rate = 0.01
ResNet50	Dropout = 0.4 Learning Rate = 0.001
InceptionV3	Dropout = 0.4 Learning Rate = 0.001

MODEL SELECTION

- Trained each model for 100 epochs with the best parameters
    - Only train the last layer of the transfer models
- Used checkpoints and early stopping with a patience of 10 monitoring validation loss
- 4,500 training images and 1,000 validation images

Model	Validation Loss	Validation Accuracy
Baseline	0.5184	74.8%
VGG16	0.4877	76.9%
ResNet50	0.4060	81.6%
InceptionV3	0.7270	66.7%

RESNET50 TRAINING

- 13,000 training images, 1,000 validation images, and 1,000 test images
- Trained (last layer only) for 1,000 epochs with the best parameters
- Used checkpoints and early stopping with a patience of 20 monitoring validation loss
- Reached local minimum loss after 3 epochs
    - validation loss = 0.3812
    - validation accuracy = 83.80%
    - Test loss = 0.3746
    - Test accuracy = 83.10%

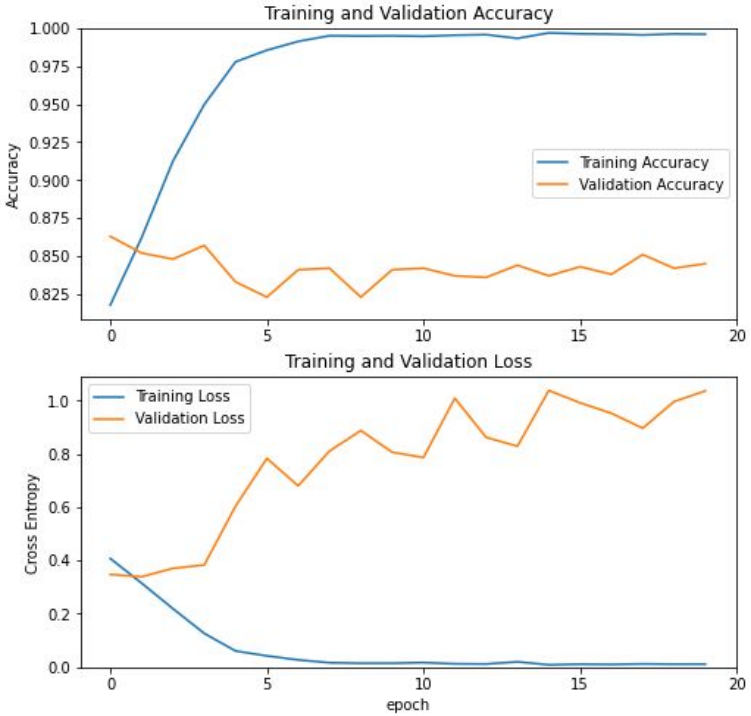
FINE-TUNING

- Unfroze lower layers of ResNet allowing for 20, 86, 150 and 214 trainable layers
- Trained each for 20 epochs with checkpoints monitoring validation loss, but no early stopping
- Same train, validation, and test images
- Learning Rate = (Best Learning Rate/10) = 0.0001

Trainable Layers	Best Val Loss	Val Accuracy	Test Loss	Test Accuracy
214	0.3484	84.4%	0.2102	91.4%
150	0.339	85.2%	0.1814	93.7%
86	0.3478	84.9%	0.2118	91.7%
20	0.3687	83.4%	0.2368	90.4%

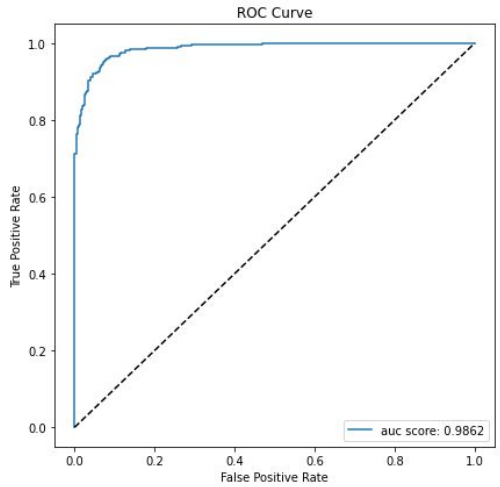
RESNET50 FINE-TUNING

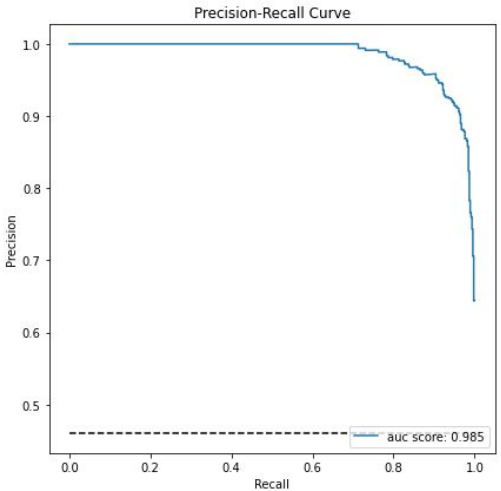
- Trained top 150 layers (out of 214)
- Reached local minimum loss after 2 epochs
  - Validation loss = 0.3390
  - Validation accuracy = 85.2%
  - Test loss = 0.1814
  - Test accuracy = 93.7%



MODEL VALIDATION

Training Summary	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
Initial Training	0.3812	83.8%	0.3746	83.1%
Fine-Tuning (150 layers)	0.3390	85.2%	0.1814	93.7%
Percent Change	-11.07%	1.67%	-51.58%	12.76%



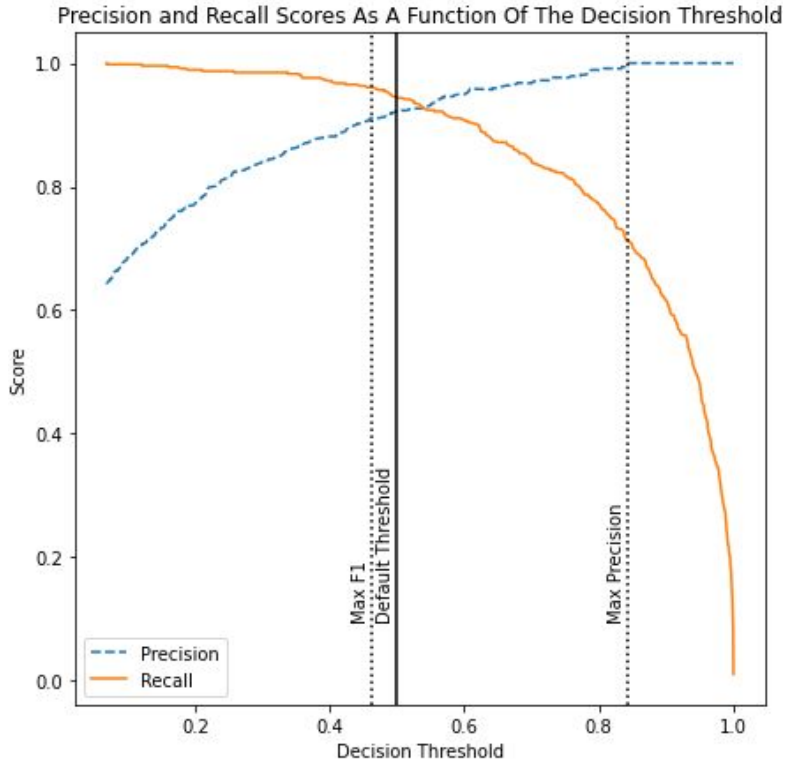


MODEL EXPLORATION

DECISION THRESHOLDS

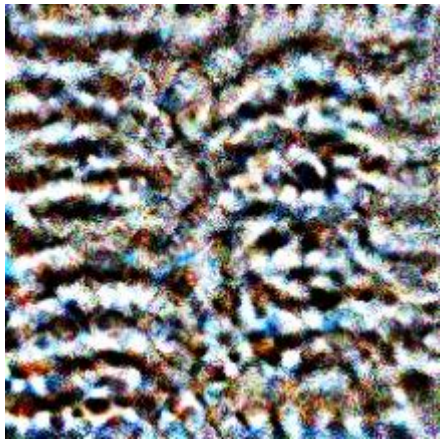
- Version A: Max F1
  - Useful for content creators
  - Precision: 0.9113
  - Recall: 0.9609

- Version B: Max Precision
  - Useful for professional photographers
  - Precision: 1.0000
  - Recall: 0.7130



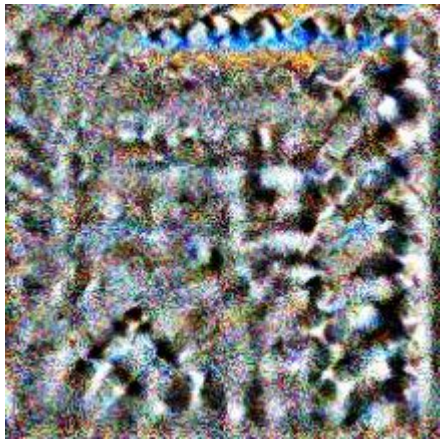
CLASS MAXIMIZATION

FP



Predicted Probability: 0.00025487

SFP



Predicted Probability: 0.9999112

CLASS MAXIMIZATION

FP

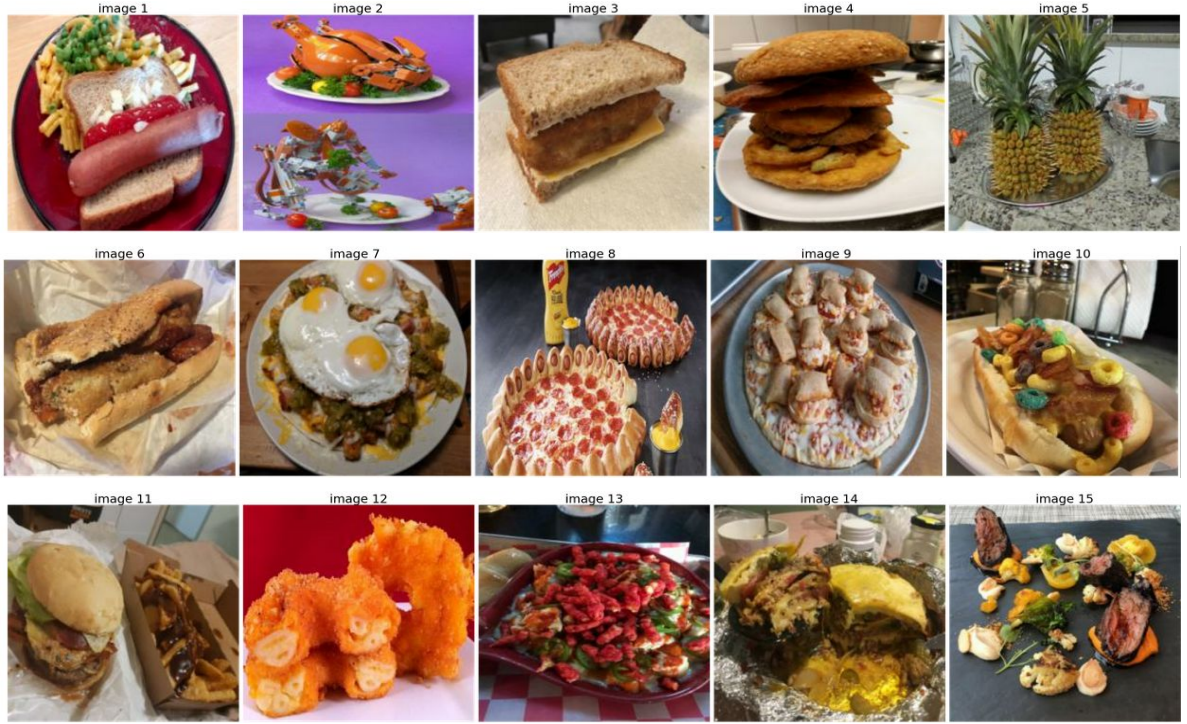
	Red	Green	Blue
Total	6,074,429	6,028,425	6,040,340
Mean	129.00	128.02	128.28
Standard Deviation	77.78	83.01	86.19

SFP

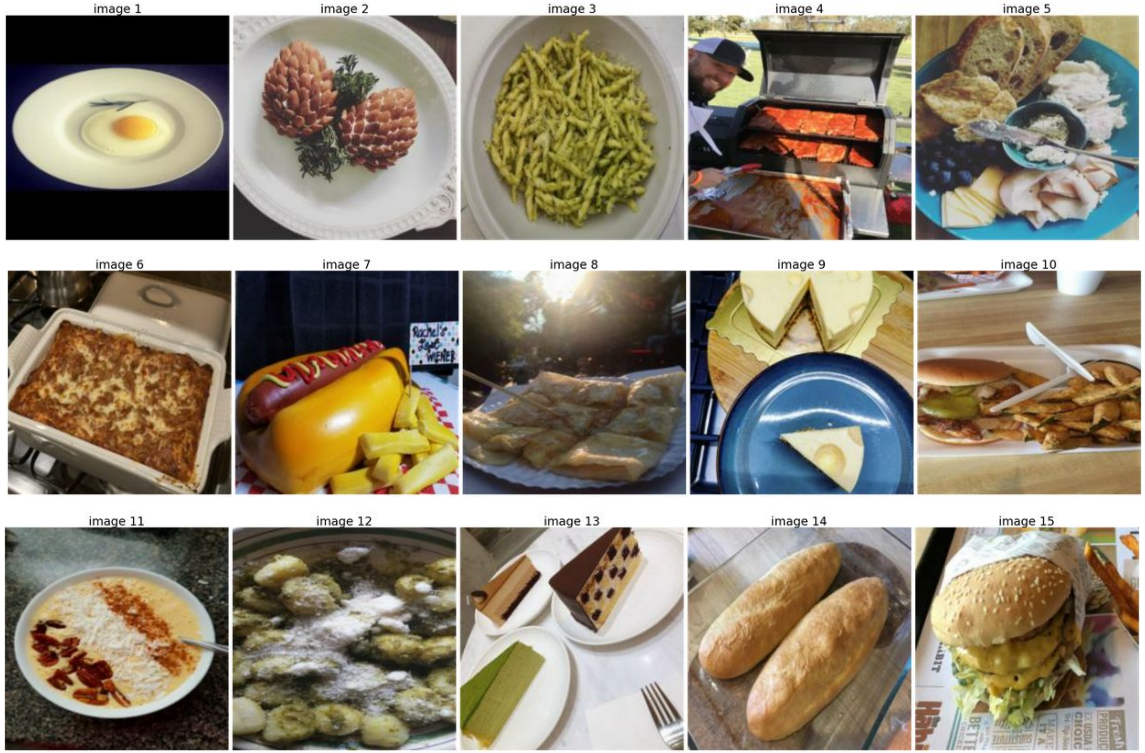
	Red	Green	Blue
Total	5,966,976	5,992,508	5,952,625
Mean	126.72	127.26	126.41
Standard Deviation	64.03	65.25	68.64

- T-tests show the red and blue mean values are statistically different with very small p-values.
- The green means p-value is 0.1170 indicating that we should not reject that the green means are the same.
- Bartlett’s tests on each channel’s standard deviation resulted in very small p-values showing that they are statistically different.

MISCLASSIFICATIONS (INCORRECTLY PREDICTED AS FP)



MISCLASSIFICATIONS (INCORRECTLY PREDICTED AS SFP)



CONCLUSION

- The final model used the ResNet50 architecture and had a 93.70% test accuracy
- Misclassifications suggest that the model has learned image composition
- Can be used by content creators and professional photographers to evaluate their food images
- Considerations for further improvement:
    - Use larger training set
    - Use higher resolution images
    - Try additional architectures

THANKS!

samtbeardsley@gmail.com