Dog vs Cat Image Classification

MSCA 31009 Machine Learning Final Project

Group 5: Aaron Gu, Lingyi Zhao, Miko Liu, Yuanyang Chang

TEAM

Aaron Gu

Lingyi Zhao

Miko Liu

Jerry Chang

THE UNIVERSITY OF CHICAGO

Agenda

- Problem Statement
- Assumptions & Hypothesis
- Exploratory Data Analysis
- Feature Engineering & Transformations
- Image Classification Models
- Conclusion & Future Work

Problem Statement

- Pets nowadays are inseparable parts of our daily life. Consequently, the pet supplies market is extremely hot. Merchants selling automated machines like litter robots or auto food feeders all take off in the market. However, most products can't distinguish the "user" and creating some chaos.
- We want to use image classification techniques to help products identify whether the "user" is a cat or a dog, so we can tailor better services to it.

Assumptions & Hypothesis

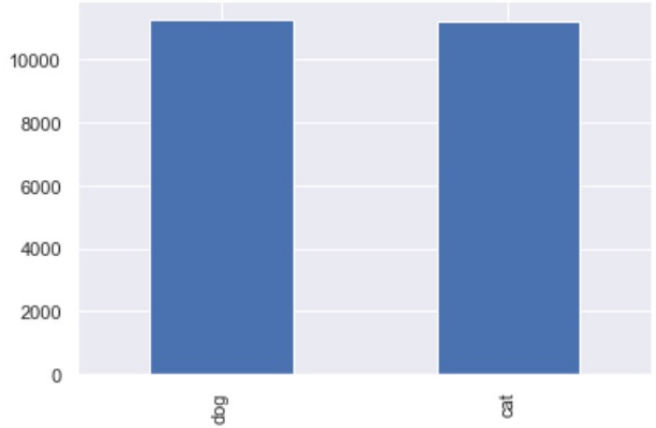
- We will use Dogs vs. Cats dataset from kaggle as our data source. It contains balanced quantity of images for each category. https://www.kaggle.com/c/dogs-vs-cats/overview
- Evaluation metrics:

Accuracy Score

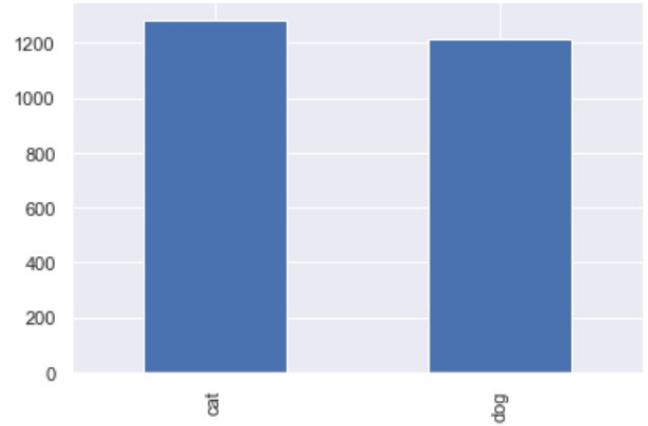
Confusion Matrix

Exploratory Data Analysis

- Train (Total 22500):



- Validation (Total 2500):




cats


cats


dogs


dogs

- Test (Total 12500): without any labels
- Images that are hard to classify from test data:

Feature Engineering & Transformations

- rescale: resize the image to uniform the dimension of each image & reduce data size to increase the running speed
- zoom_range: randomly zooming the images that allows more details of images through increasing the quantity of pixels
- width_shift_range & height_shift_range: translate the pictures vertically or horizontally
- rotation_range: randomly rotate the picture at a certain range
- brightness_range: set a brightness range will make sure images have similar brightness level
- horizontal_flip: randomly flipping half of the images horizontally
- fill_mode: the strategy used for filling in newly created pixels, which can appear after a rotation or a width/height shift.

CNN Model 1

- By Miko Liu

Basic simple layer CNN

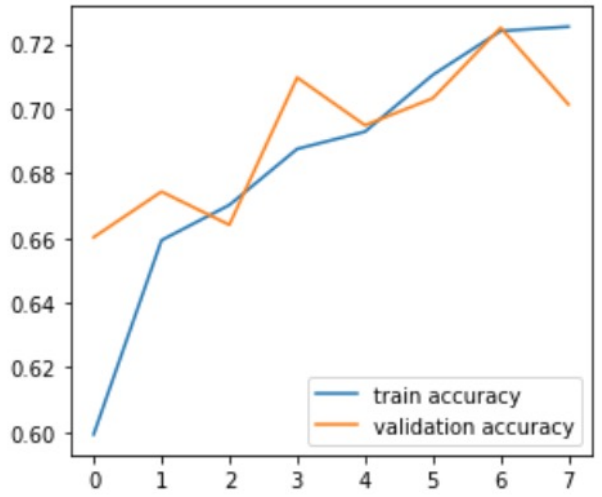
Architecture

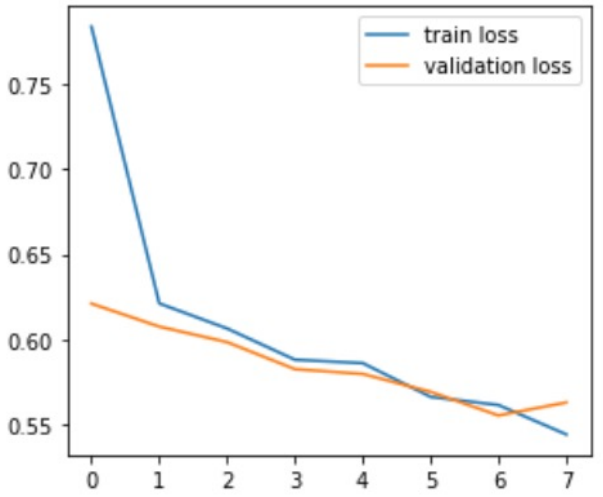
- Conv2D
- MaxPooling
- Flatten
- Relu Dense
- Sigmoid Dense

Fit

- Adam Optimizer
- Binary Crossentropy Loss
- Batch Size = 225
- Epochs = 8

	precision	recall	f1-score	support
0.0	0.74	0.59	0.66	1249
1.0	0.66	0.79	0.72	1251
accuracy			0.69	2500
macro avg	0.70	0.69	0.69	2500
weighted avg	0.70	0.69	0.69	2500

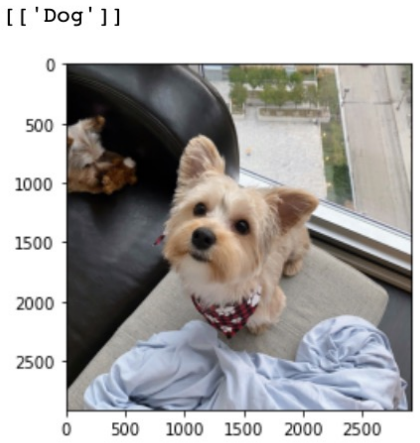


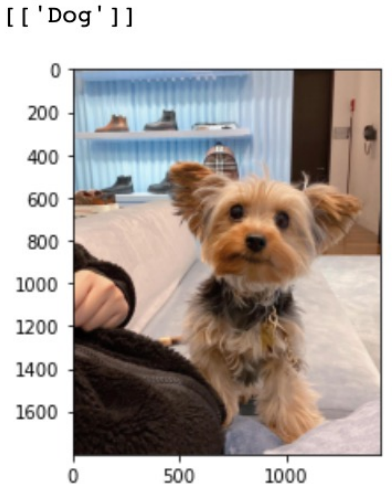


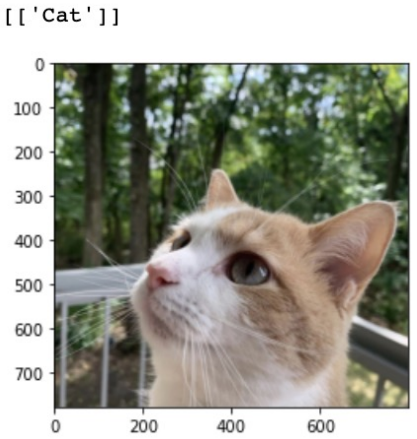
Model Performance

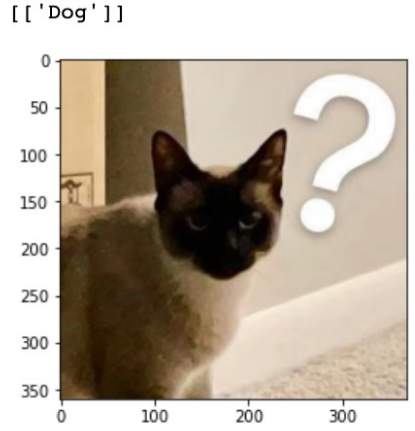
Randomly Selected Test Data

Our Pet









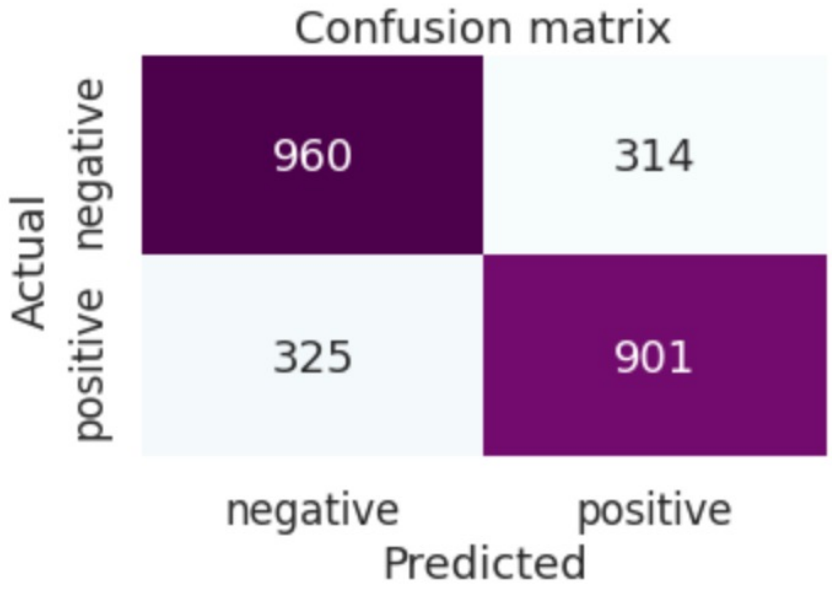
CNN Model 2

- By Jerry Chang

Two Conv2D CNN

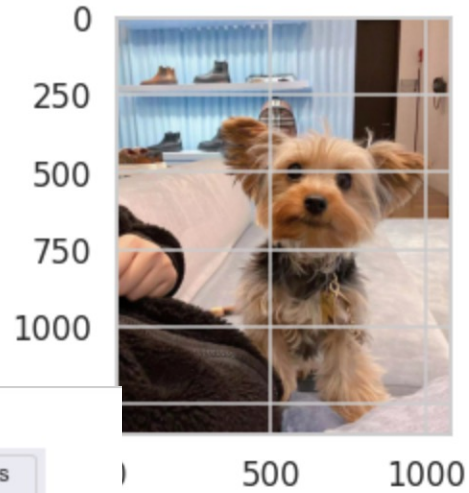
Architecture

- Conv2D_1
  - Batch Normalization
  - MaxPooling
  - Dropout
- Conv2D_2
  - Batch Normalization
  - MaxPooling
  - Dropout
- Flatten
- Relu Dense
  - Batch Normalization
  - Dropout
- Relu Dense
  - Batch Normalization
  - Dropout
- Sigmoid Dense

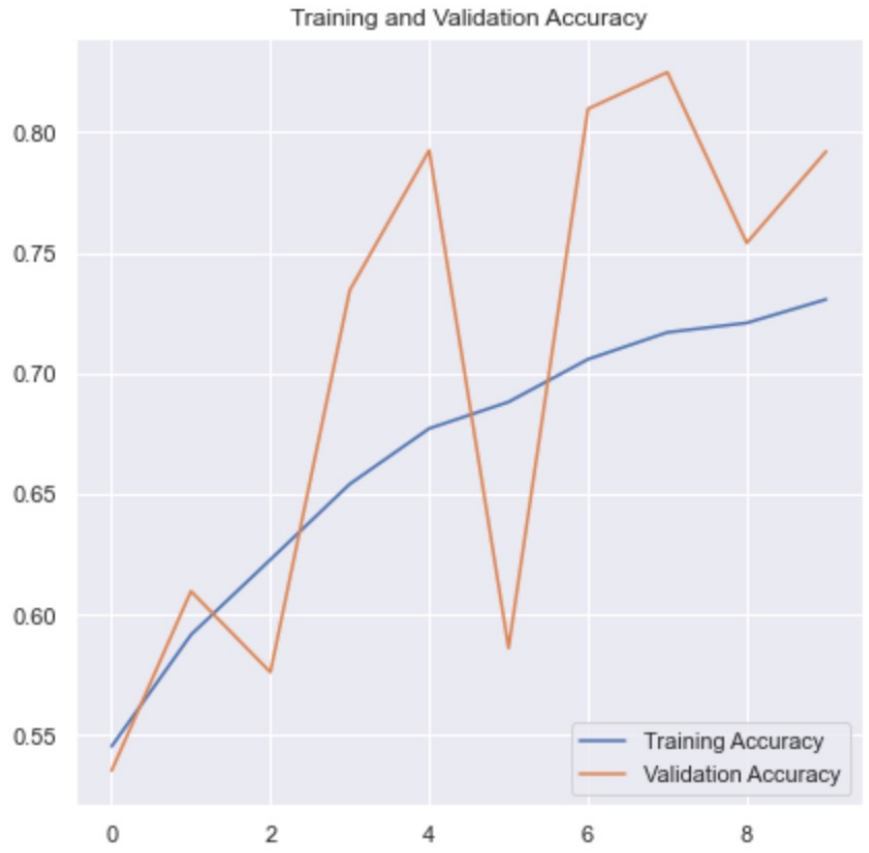


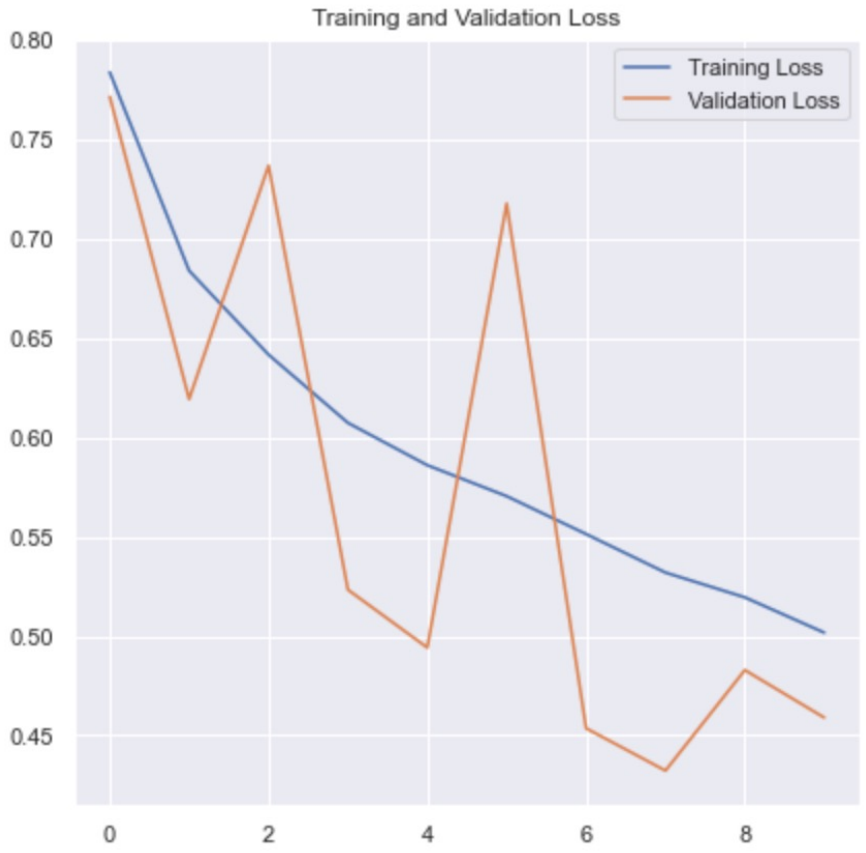
	precision	recall	f1-score	support
0.0	0.75	0.75	0.75	1274
1.0	0.74	0.73	0.74	1226
accuracy			0.74	2500
macro avg	0.74	0.74	0.74	2500
weighted avg	0.74	0.74	0.74	2500

[[0.552469]]



Accuracy: 74.44%

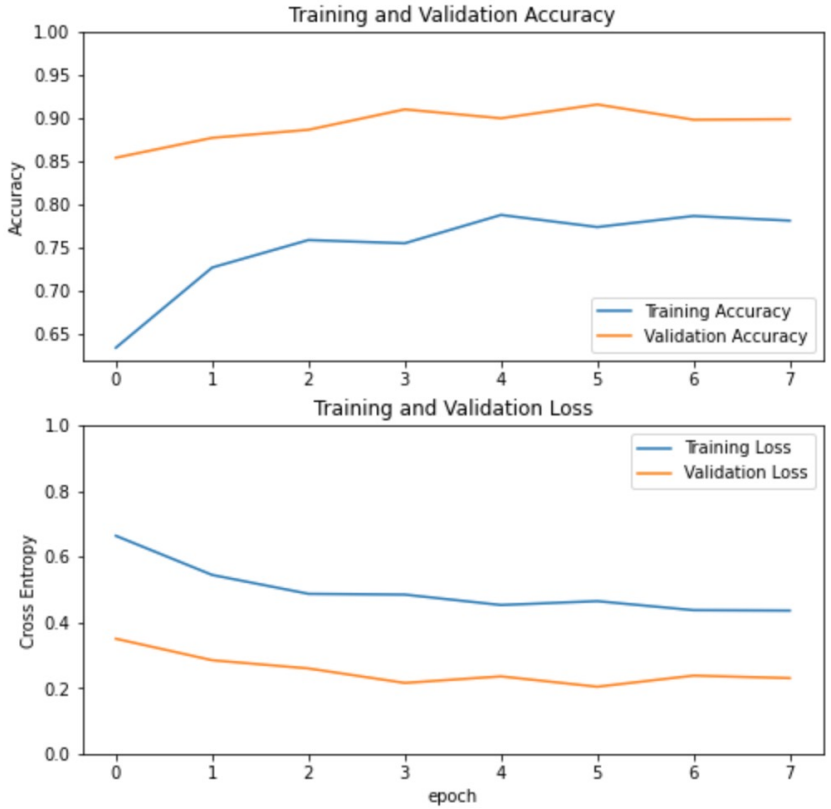


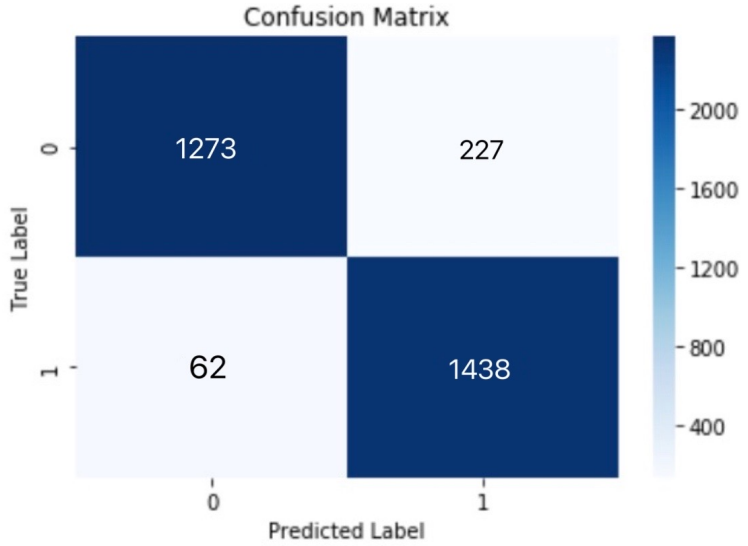


VGG16 Model

- By Lingyi Zhao

Base Model



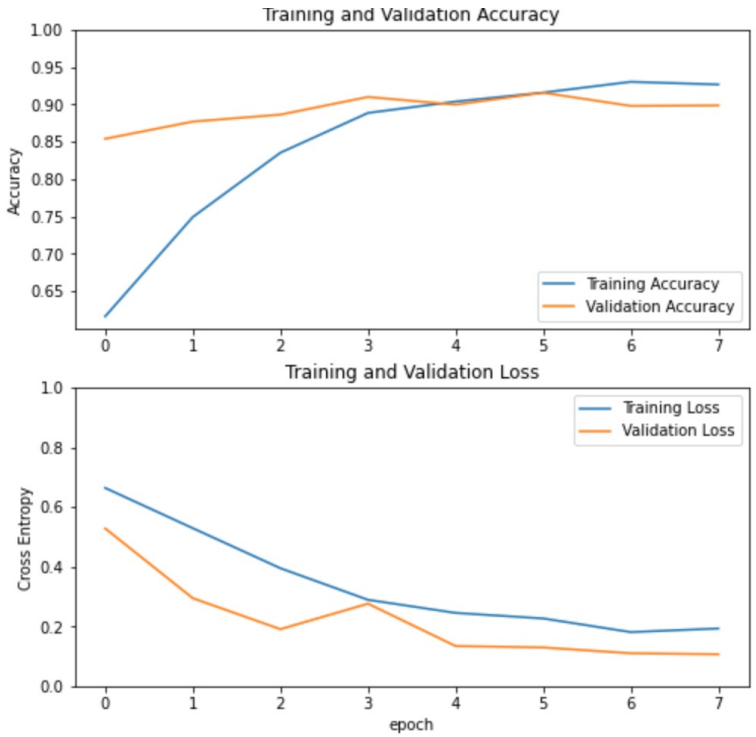


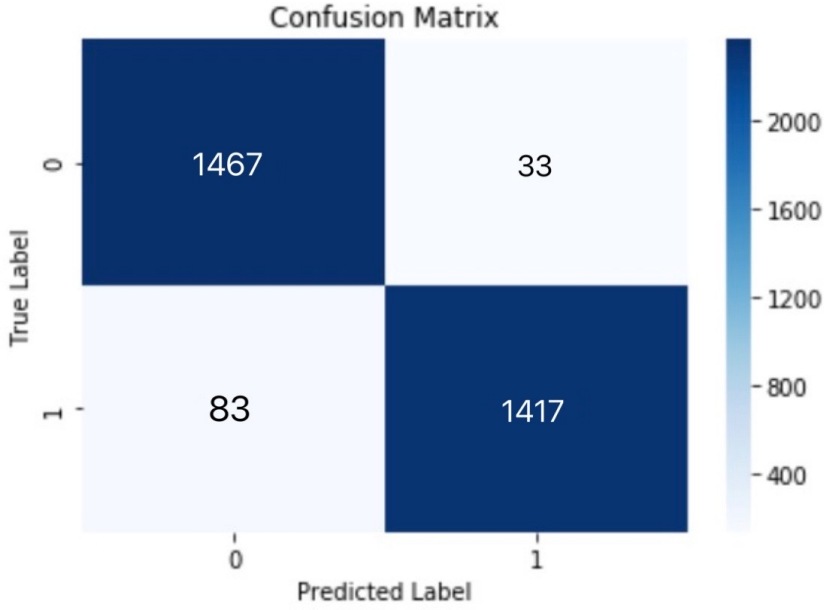

```
base_model.trainable = False
model = keras.Sequential([
base_model,
layers.Flatten(),
layers.Dense(2,activation='sigmoid') ])
```

	precision	recall	f1-score	support
0	0.96	0.83	0.89	1500
1	0.86	0.96	0.91	1500
micro avg	0.90	0.90	0.90	3000
macro avg	0.91	0.90	0.90	3000
weighted avg	0.91	0.90	0.90	3000
samples avg	0.89	0.90	0.89	3000

Class 0: Cat Class 1: Dog

Fine Tuned Model



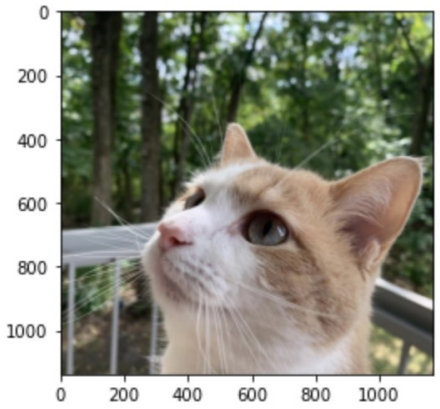


	precision	recall	f1-score	support
0	0.96	0.97	0.96	1500
1	0.97	0.95	0.96	1500
micro avg	0.96	0.96	0.96	3000
macro avg	0.96	0.96	0.96	3000
weighted avg	0.96	0.96	0.96	3000
samples avg	0.96	0.96	0.96	3000

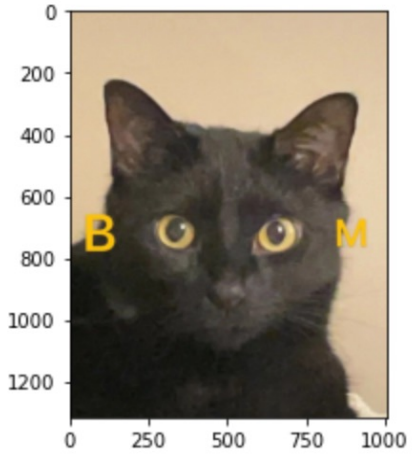
```
base_model.trainable = True

model2 = keras.Sequential([
    base_model,
    layers.Flatten(),
    layers.Dense(2,activation='sigmoid')])
```

Class 0: Cat Class 1: Dog

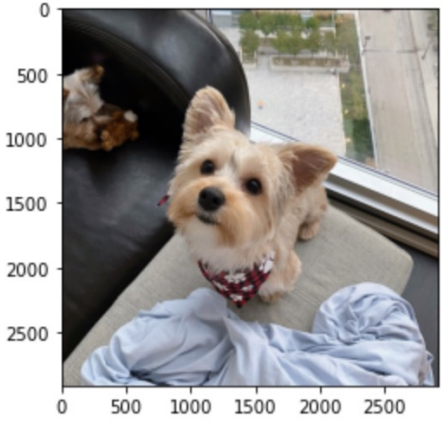
Model Results & Prediction



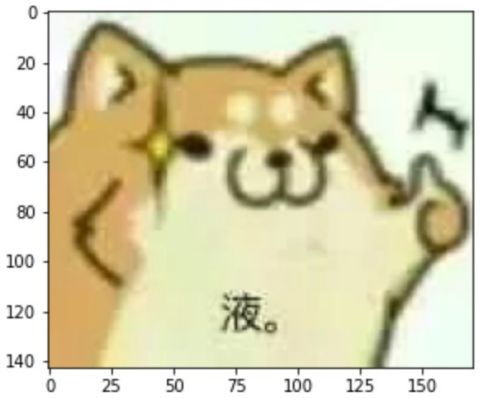
[[0.9094006 0.3354423]]



[[0.99874336 0.00228077]]



[[2.7747192e-06 9.9999213e-01]]



[[0.58293366 0.14366853]]

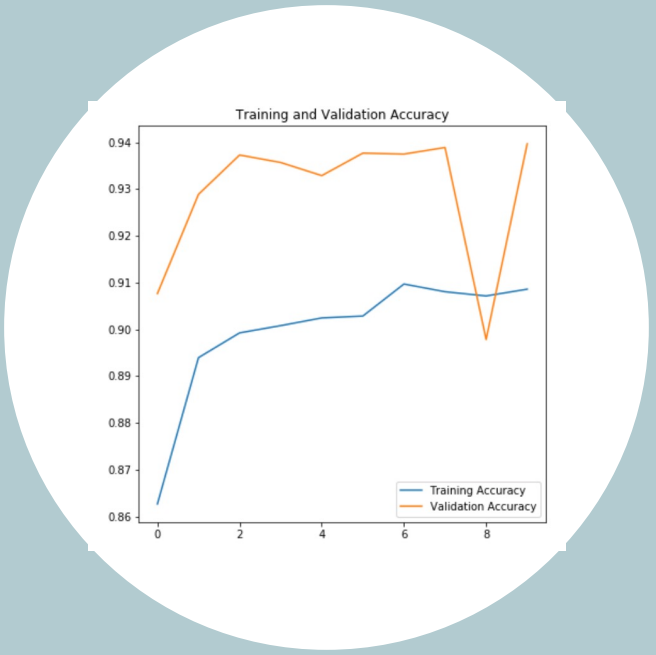
- VGG 16 model has better performance in identifying cats.
- Background may affect the prediction results.

Inception V3 Model

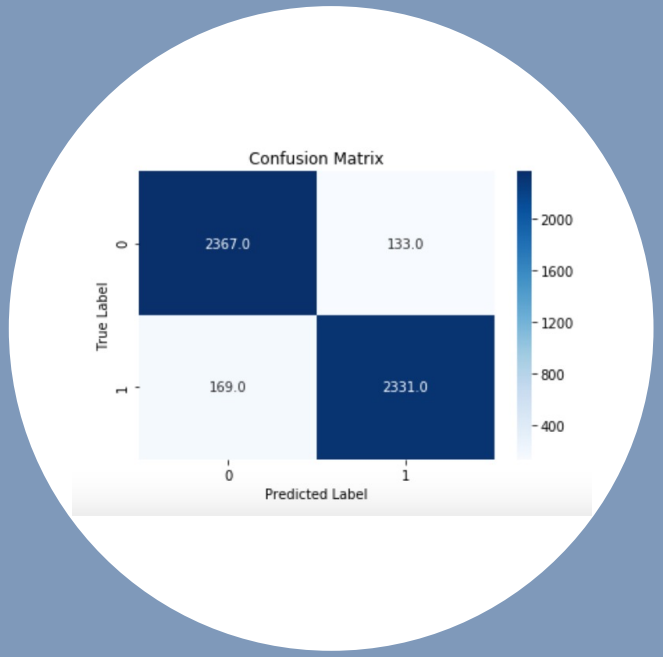
- By Aaron Gu

Inception V3 Fit

	precision	recall	f1-score	support
0.0	0.93	0.95	0.94	2500
1.0	0.95	0.93	0.94	2500
accuracy			0.94	5000
macro avg	0.94	0.94	0.94	5000
weighted avg	0.94	0.94	0.94	5000

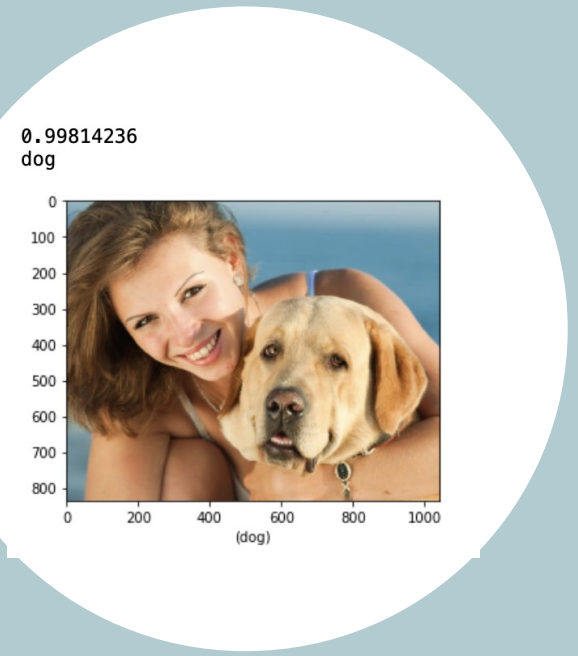


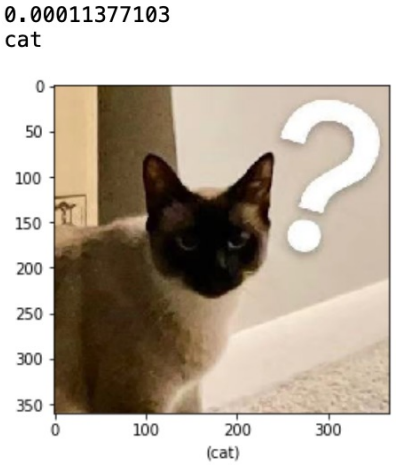
Validation Accuracy: 0.9399
Validation Loss: 0.1481
(Class 0: Cat ; Class 1: Dog)

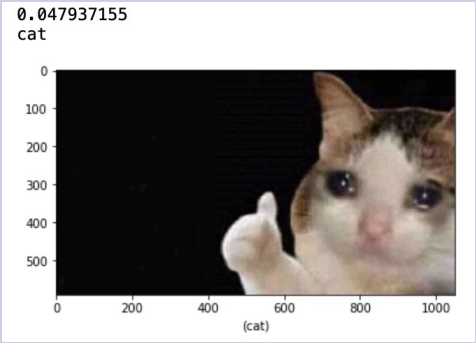


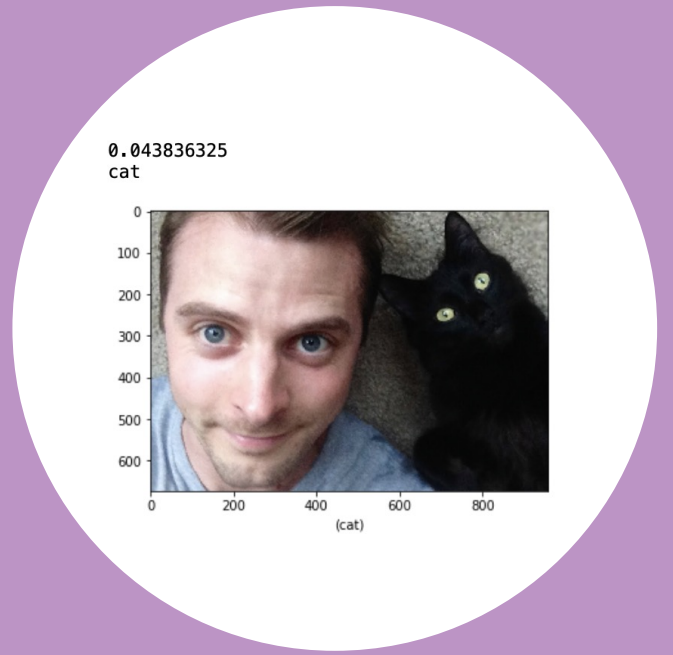
Inception V3 Sample Prediction

- Relatively high accurate predictions
- Works well on Cat/Dog with other objects
- Works well on all color breeds









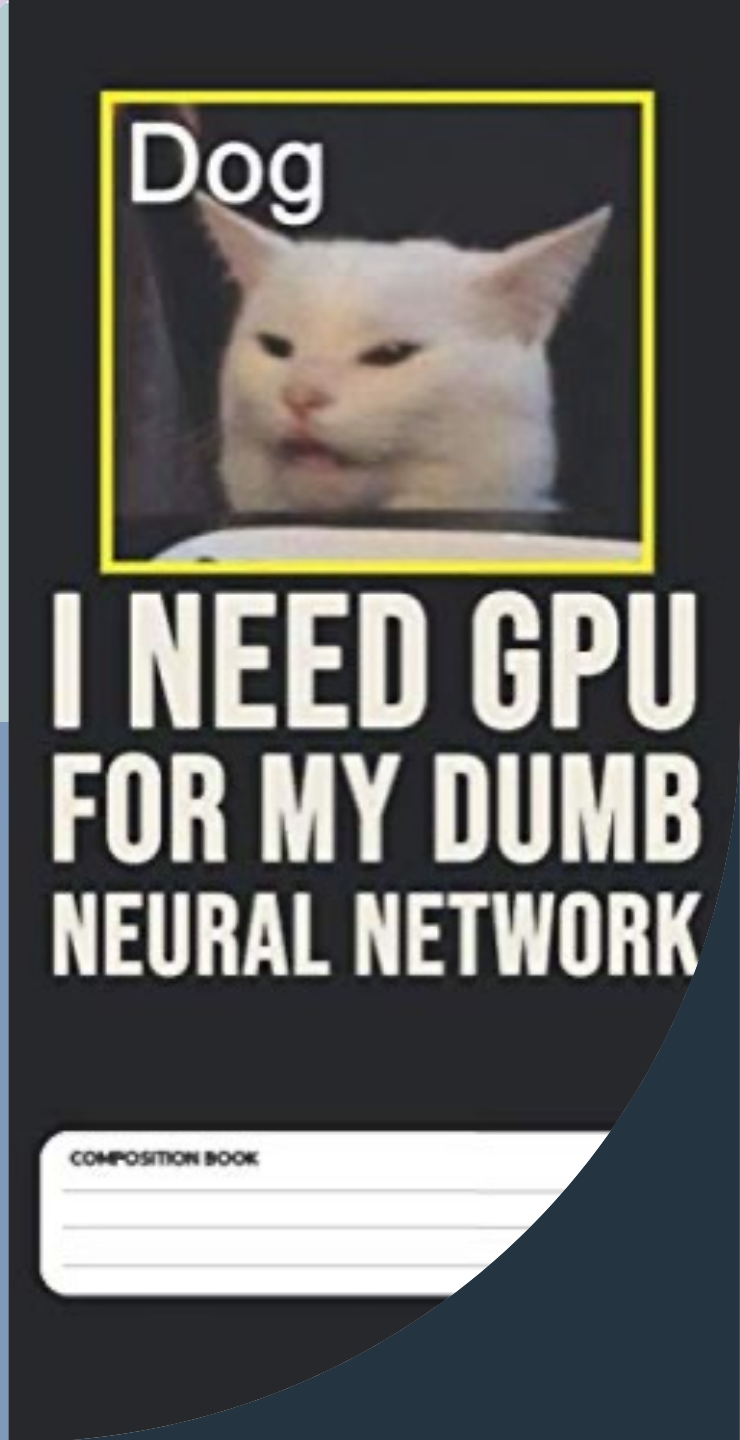
Model Comparison

	CNN_base	CNN	VGG16	VGG16_fine tuned	Inception V3
Accuracy	0.692	0.744	0.8987	0.9637	0.9399
Overfitting	No	No	Underfitting	No	Underfitting

Challenges & Future Work

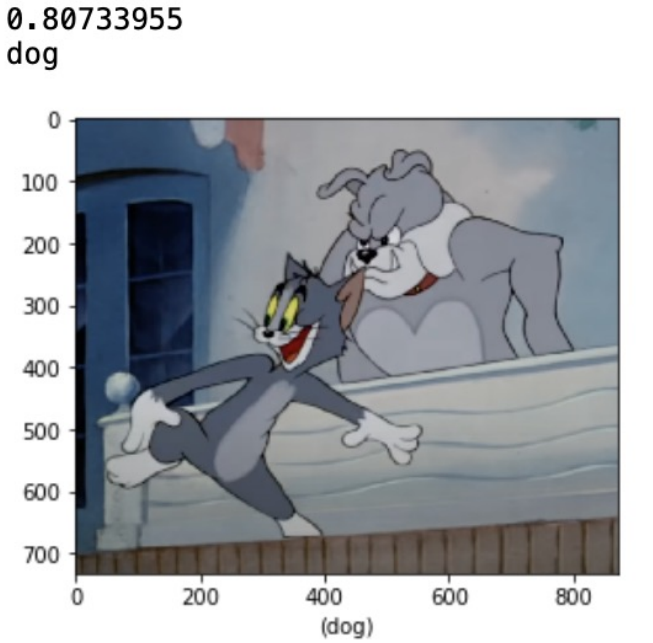
Efficiency & Accuracy Tradeoff

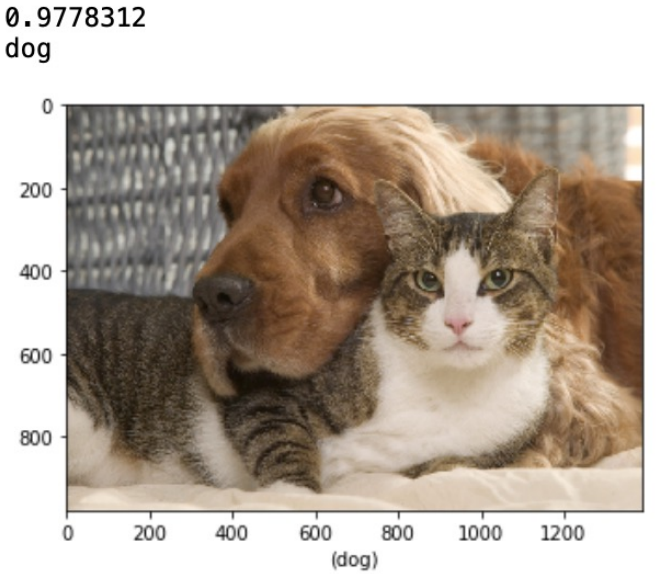
- Due to the hardware capacity, the size of the data, the number of layers, batch size, and epochs are often limited in order to have relatively acceptable training time.
- With better hardware or cloud services, we could have a larger training sets with more layers and more complex parameter settings to seek better model performances



Multi-Object Obstacles

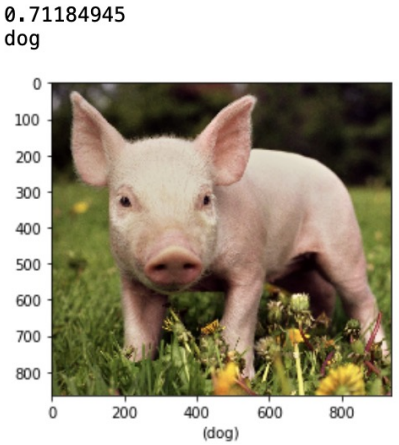
- Model has the limitation when multiple goal objects are in the same pictures. Specifically, most of the time model will predict "dog" when cats and dogs are in the same picture.



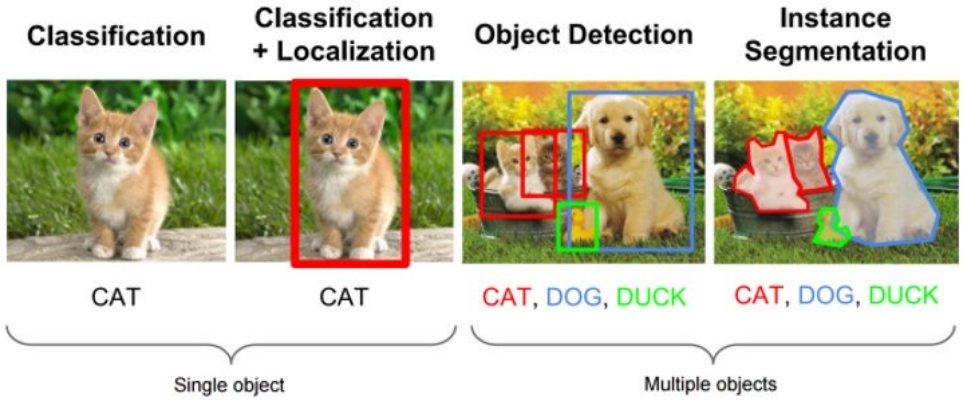


Binary Classification Limitation

- Binary classification model also has the limitation in detecting other non-relevant objects.



- In the future, we should continusouly imrpoving our model from binary classification to multi-categorical classification, with object detection and instance segmentation.



Thank You!