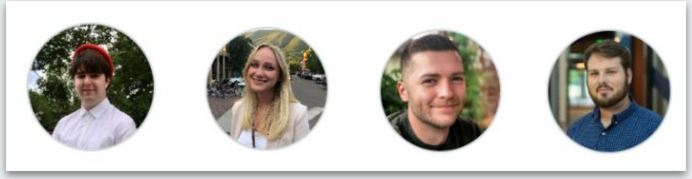
# Cats and Dogs: Final Phase

Group 2:
Owen Collier, Julia Donato, Jacob Scott, Gregory Shoda



# Outline

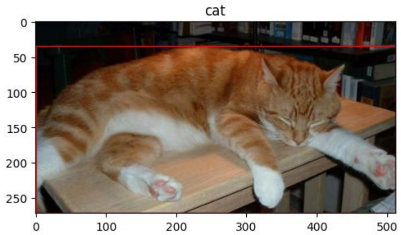
- Project Description
- Methods
  - Multi-headed Detector Using OOP API
  - EfficientDet D0-D7
  - YoloV8
  - FCN
- Conclusion

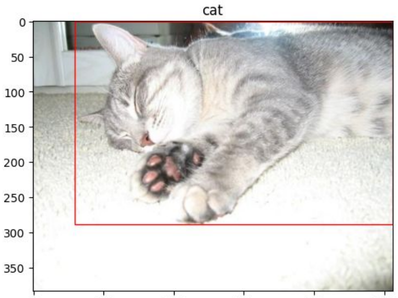
# Project Description

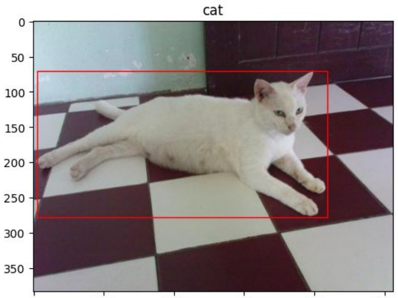
This project aims to address the complex challenge of object detection in computer vision by building pipelines for identifying the main object in an image, specifically cats or dogs.

Goals - Build a model that can detect and distinguish between cats and dogs in images.

Givens - Open Images V6 (subset)







# Progress so far:

Phase 2:

- Classification: Logistic Regression
- Bounding Box: Multiple Linear Regression

Phase 3:

- Pytorch Classifier
- Pytorch Regressor

Scores:

- Test Accuracy 0.54
- Test MSE of 0.028

- Test Accuracy 0.615
- Test MSE of 0.009

# Multi-headed Detector using OOP API

**Data Scaling and Conversion**

- Standardized using StandardScaler.
- Numpy arrays converted to PyTorch tensors for training, validation, and testing.

**Dataset and DataLoader Configuration**

- PyTorch TensorDataset and DataLoader Setup for Data Processing
- Tested across batch sizes

**Multi-Headed NN:**

- Shared layers for initial processing.
- Separate classification and regression layers.
- Configurable hidden layers and output units.
- Tested across activation functions.

**Loss Function and Optimization**

- Combined Loss: α * Binary Cross-Entropy Loss (Classification) + (1-α) * (MSE + L2 Regularization) (Regression)
- SGD optimizer

**Training and Evaluation Process**

- Evaluation across classification accuracy, regression loss, and intersection over union (IoU) for bbox accuracy.

**Best* Model**

Batch size of 32 with LeakyReLU for 100 epochs

* still poor performance, earlier data issue?

| Test Accuracy | Test Mean MSE | Test Mean IoU |
|---|---|---|
| 59.1 | 0.0365 | 0.440 |

# EfficientDet D0-D7

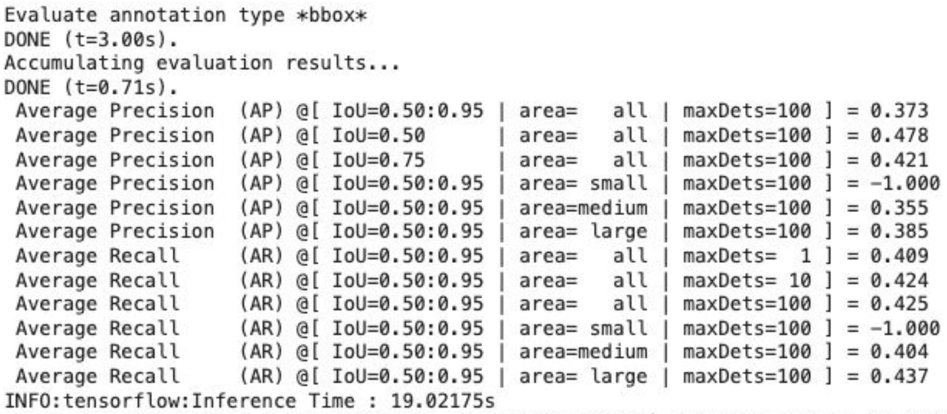
- EfficientDet-D0: Average Precision (AP) of 0.373 and Average Recall (AR) of 0.425, excelling in dog detection (AP 0.746) but failing in cat detection (AP 0.0).
- EfficientDet-D7: Improved performance with an AP of 0.394 and AR of 0.443, indicating slightly better overall detection capability.
- Limitation: Both models demonstrated an inability to detect small objects, with AP and AR for small areas at -1.000.

Loss Functions:
Combination of Focal Loss and Smooth L1

$$-\alpha(1-p)^{\gamma}\log(p)$$

$$L_{\text{Huber}}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases}$$

D0 Results- 10 epochs



```
Evaluate annotation type *bbox*
DONE (t=3.00s).
Accumulating evaluation results...
DONE (t=0.71s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.373
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.478
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.421
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.355
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.385
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.409
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.424
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.425
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.404
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.437
INFO:tensorflow:Inference Time : 19.02175s
```

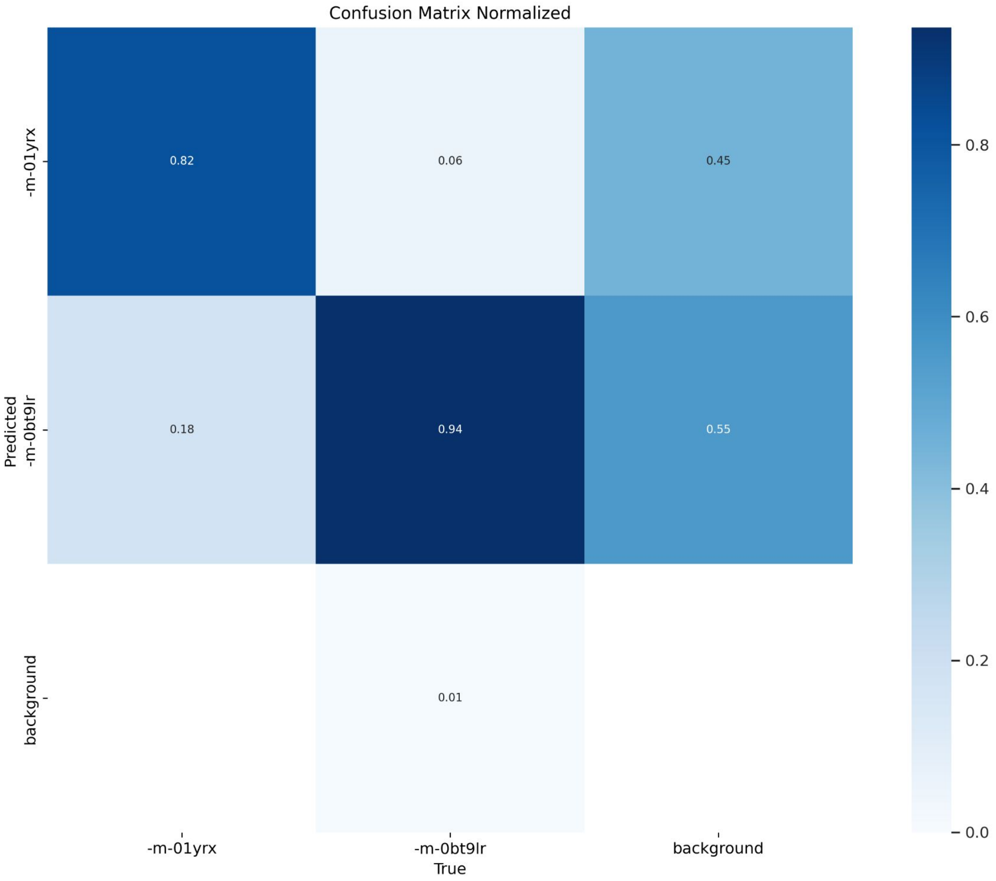
# YoloV8

-Created Custom Dataset Using RoboFlow to Import to YoloV8

-4850 Image Sample: 3413 Train Images, 966 Validate Image, 471 Test Images

-Used YOLOv8m Model for Training

- Showing 0.94 Accuracy for Dogs and 0.82 Accuracy for Cats

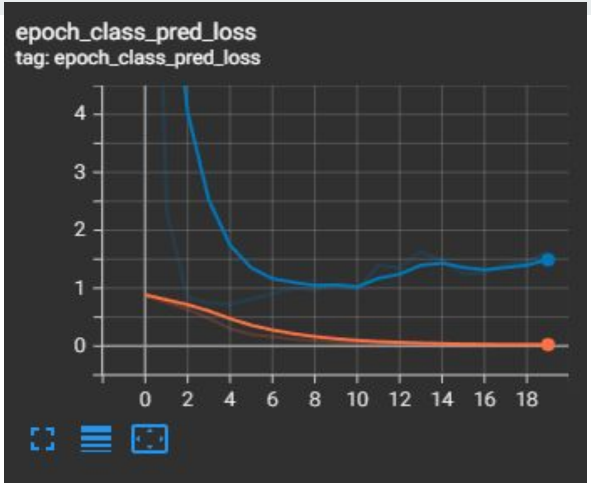


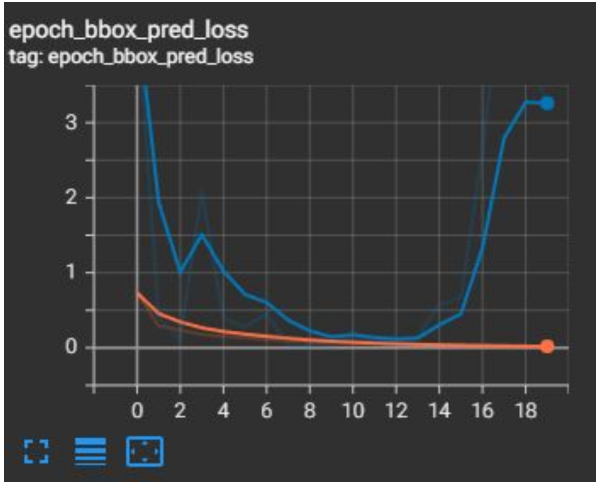
Blue - Validation
Orange - Train

# FCN Modelling

- Looking at loss functions alone, performs very well. However…
  - It only predicts dogs.
- Bounding box prediction is good.
  - training for ~11 epochs is best.
- Solutions?
  - Cutoff for logit - closer to class imbalance?
  - Maybe a discrepancy between loss function and predictions?

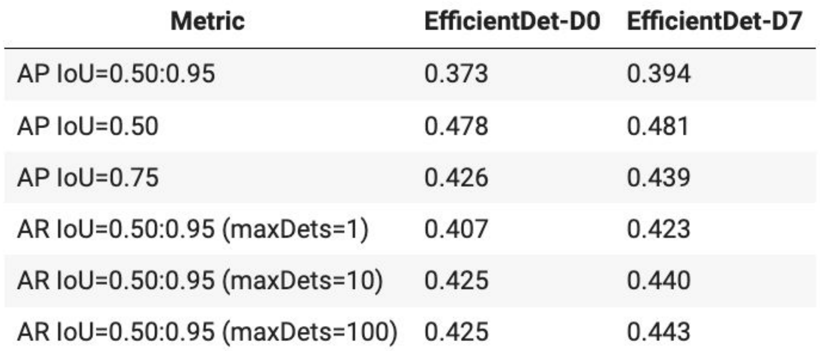
```
class pred accuracy: 0.5312 -
bbox_pred_mean_squared_error: 0.0208
```





# Conclusion

YoloV8 was our best model for classification, and our PyTorch regressor from phase 3 was our best bounding box predictor.



| Metric | EfficientDet-D0 | EfficientDet-D7 |
|---|---|---|
| AP IoU=0.50:0.95 | 0.373 | 0.394 |
| AP IoU=0.50 | 0.478 | 0.481 |
| AP IoU=0.75 | 0.426 | 0.439 |
| AR IoU=0.50:0.95 (maxDets=1) | 0.407 | 0.423 |
| AR IoU=0.50:0.95 (maxDets=10) | 0.425 | 0.440 |
| AR IoU=0.50:0.95 (maxDets=100) | 0.425 | 0.443 |

| Model | Classification Accuracy (%) | Test MSE / IoU |
|---|---|---|
| Multi-Headed | 59.1 | 0.0365 / 0.440 |
| YoloV8 | 88.1 | N/A |
| FCN | 53.12 | 0.0208 (MSE) |
| PyTorch (Phase 3) | 61.5 | 0.009 (MSE) |