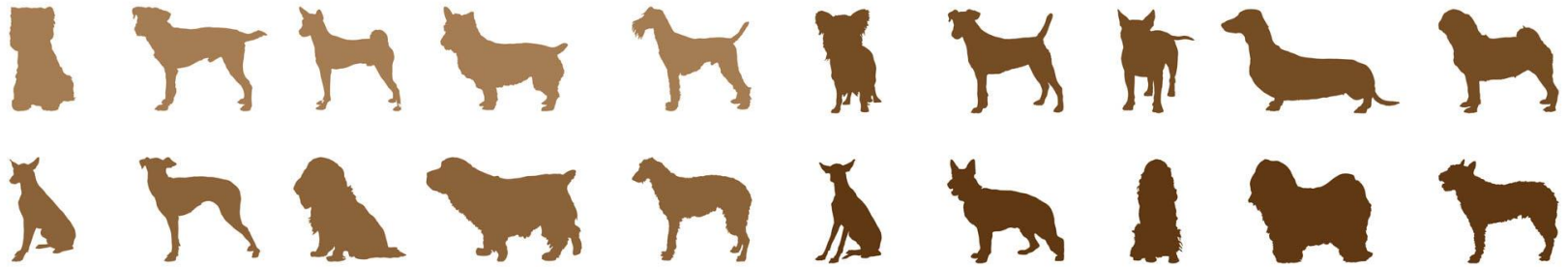
Image Classification and Object Detection:

Dog Breed Detector
&
Norwich Terrier Detector

Dog Breed Classification & Object Detection



Objectives

This capstone project encompasses two distinct yet interconnected tasks within the realm of computer vision:

Task 1 - Image Classification: develop a multi-class image classification application capable of accurately predicting various dog breeds.

- Provide breed predictions for both static images and real-time webcam captures.
- Success criteria: target score of 85% or higher

Task 2 - Object Detection: build an object detection model to identify a specific dog breed, Norwich Terriers, within images and videos.

- Success criteria mean Average Precision at IoU threshold 0.50 (mAP50) score of 0.80 or higher.

Task 1 - Image Classification

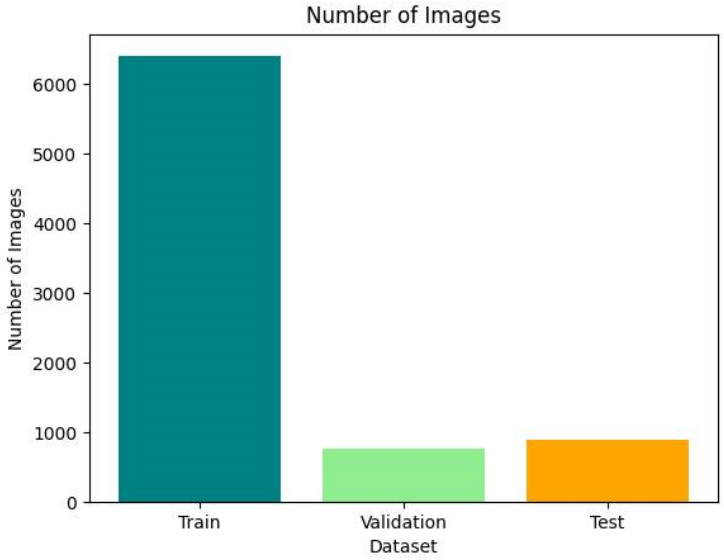
Image Data & Preprocessing

Images

Source: Kaggle

Data:

- Training images: 6,405
- Validation images: 766
- Test images: 891

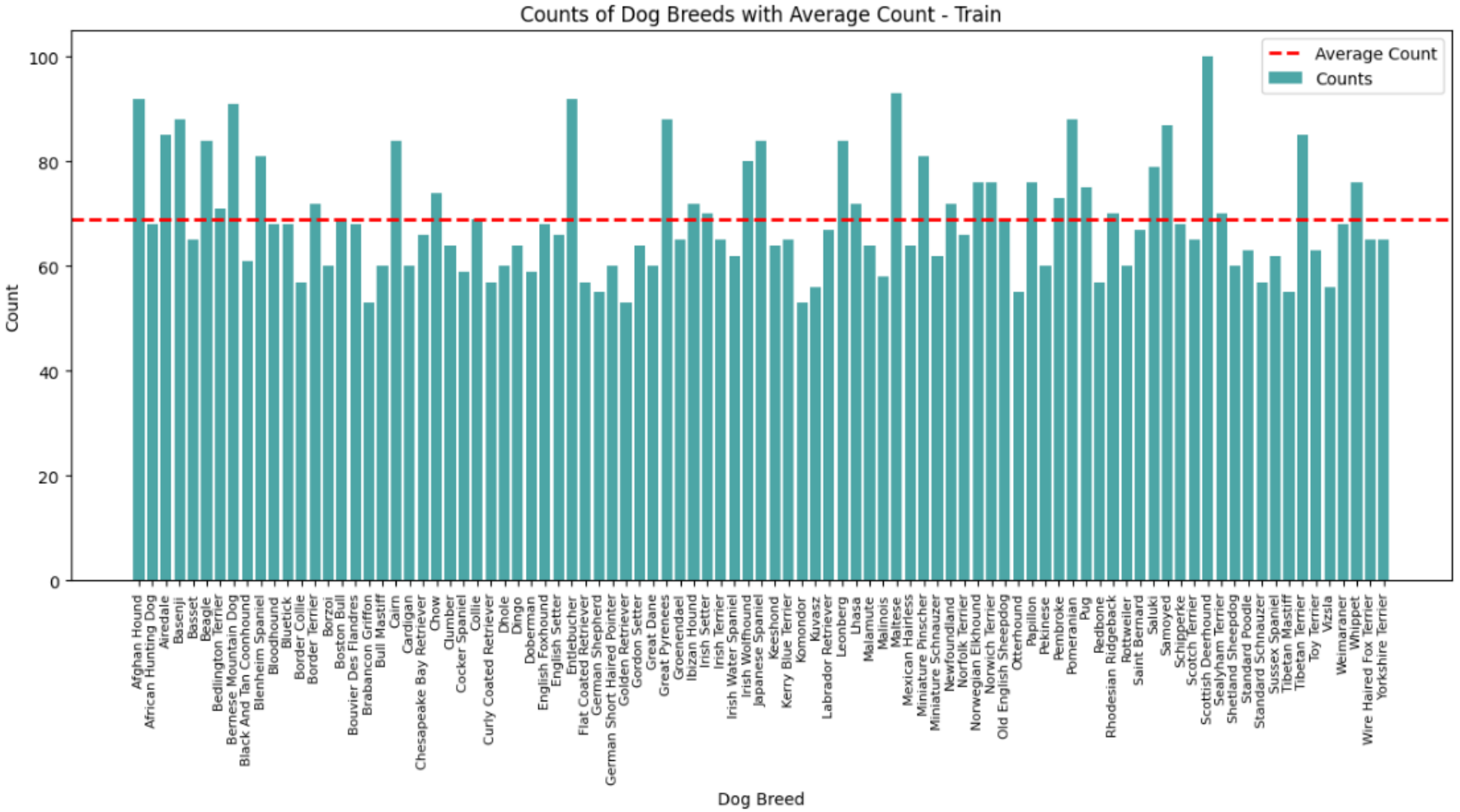


Sample Images

Preprocessing

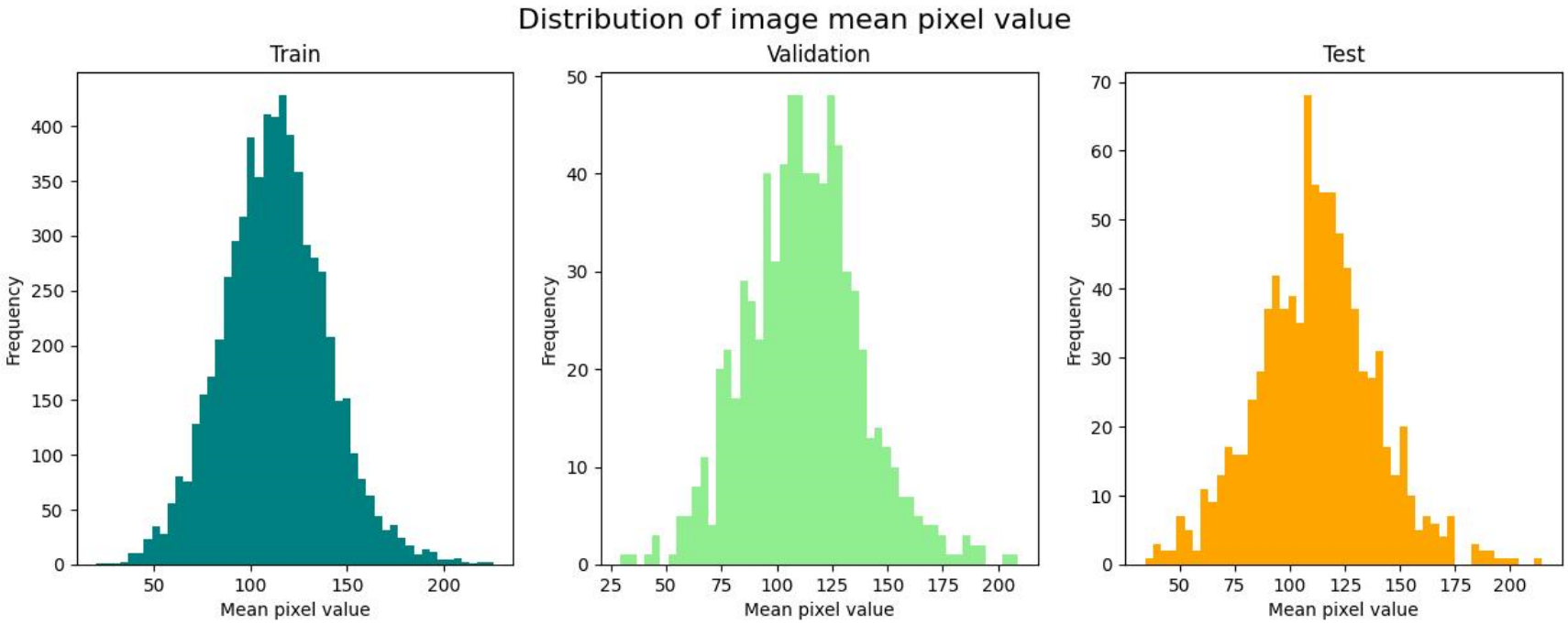
1) Convert to RGB
2) Image size 299 x 299
3) Normalized data

EDA: Number of images per breed (train)



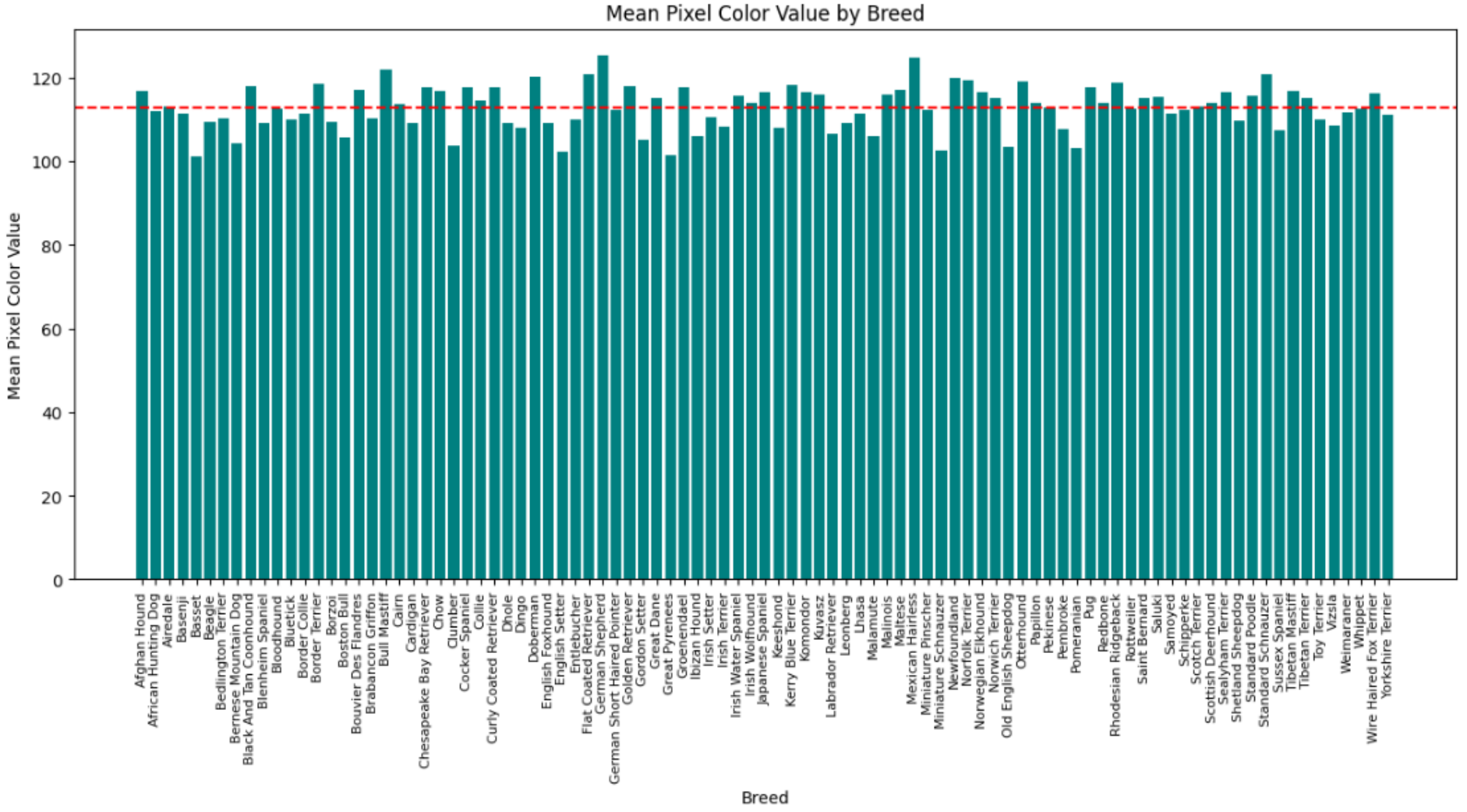
- 93 total number of dog breeds (classes)
- Avg. number of images per breed: 68.87
- Min: 53
- Max: 100

EDA: Distribution and average pixel intensity



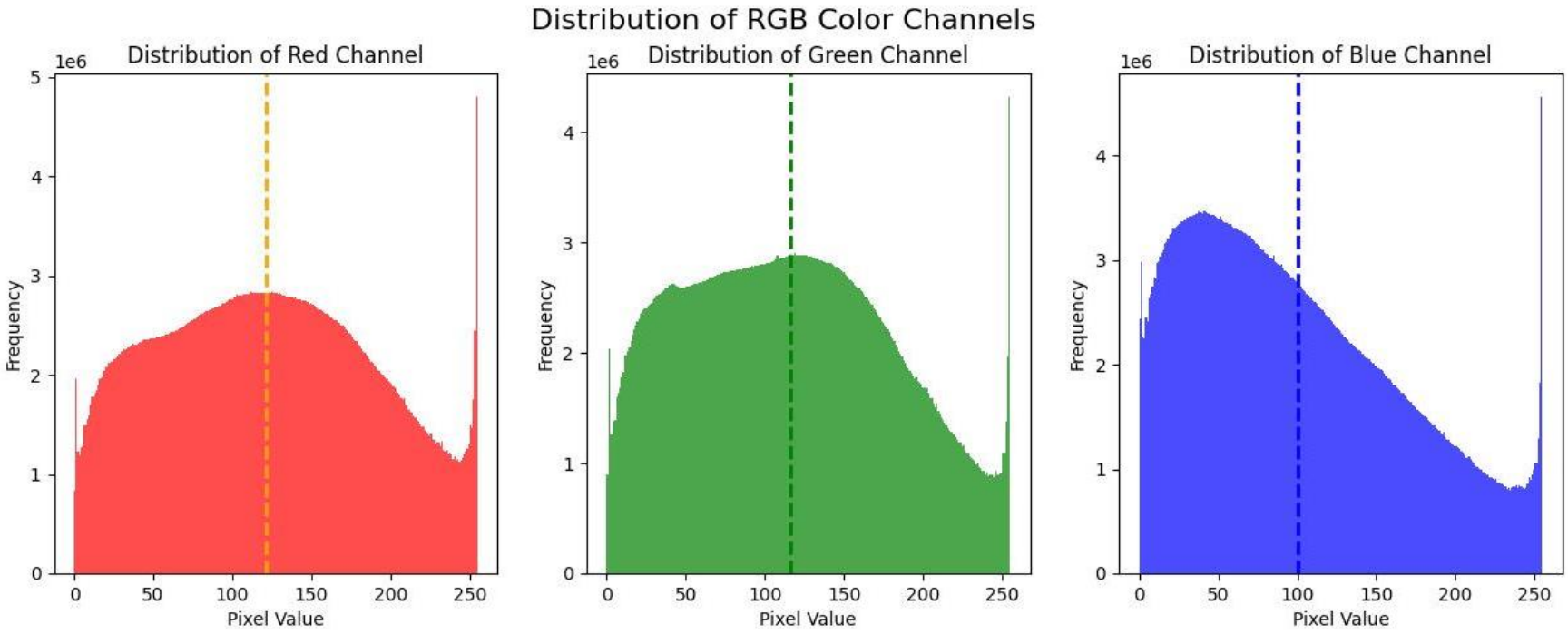
- The distribution and average pixel intensity of train, validation and test images are pretty similar:
  - Train: 112.65
  - Validation: 112.24
  - Test: 111.93

EDA: Pixel values by breed (train)



- Mean color value for all breeds: **112.80**

EDA: Distribution of RGB Color Channels

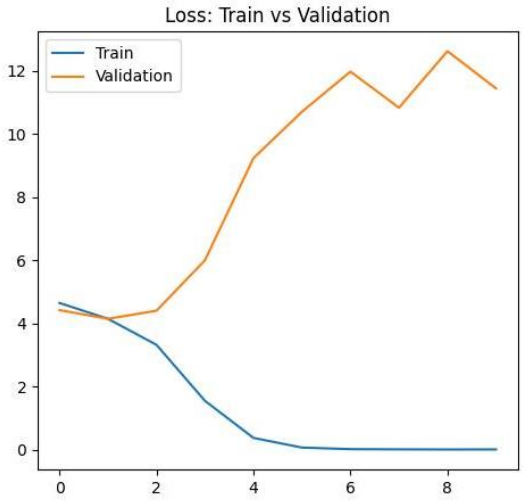


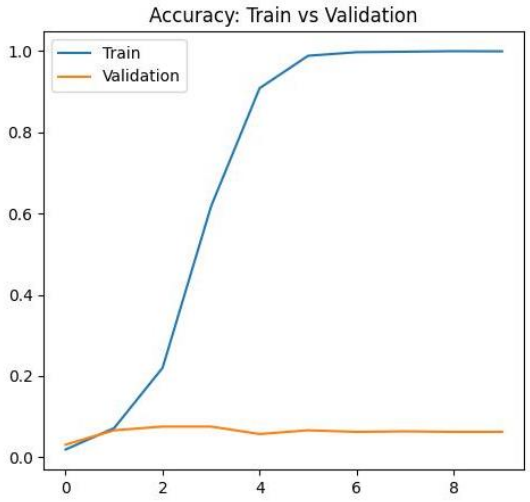
- Red is the most predominant color channel, followed closely by Green and Blue. However, in general, all three colors are used quite evenly in images.
  - Red: 121.79
  - Green: 115.83
  - Blue: 100.34

Modeling – Custom CNN Model Development

Model 1

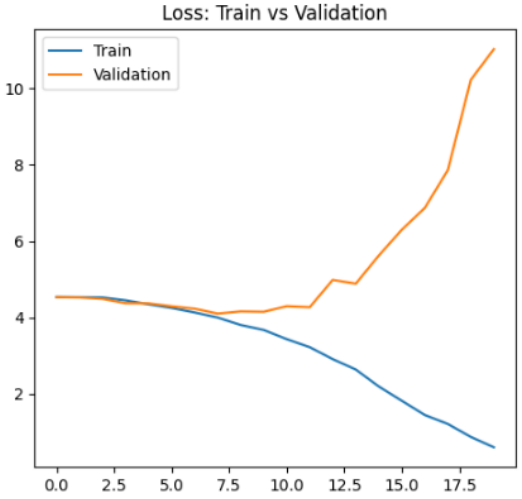
- 3 Convolutional layers
- 1 Dense layer
- Accuracy:
  - **Train: 0.999**
  - **Validation: 0.062**

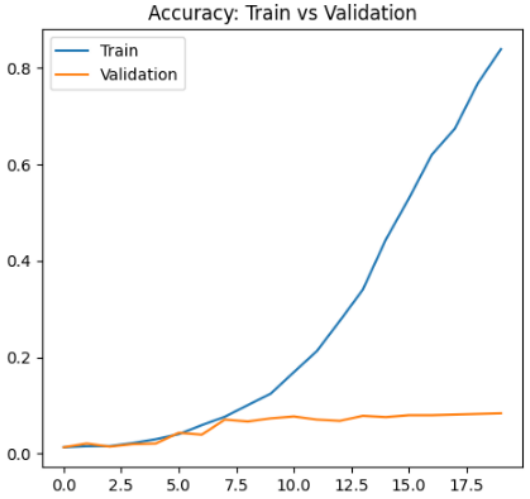




Model 2

- 4 Convolutional layers
- 2 Dense layer
- Accuracy:
  - **Train: 0.839**
  - **Validation: 0.084**

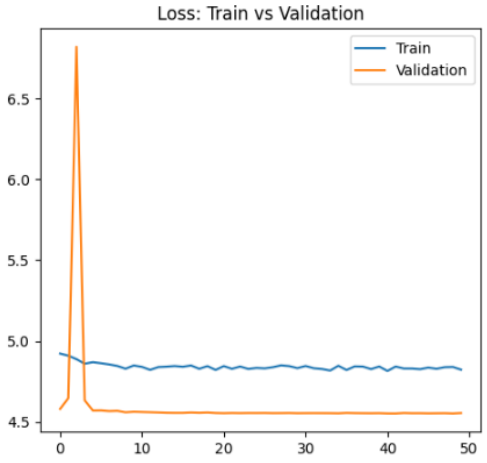


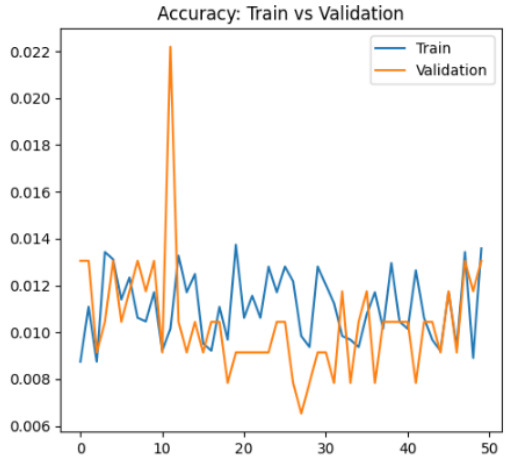


Modeling – Pre-trained Models

VGG16 + Custom 'top' model

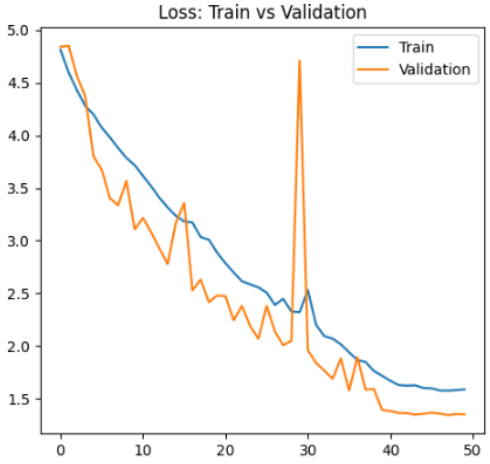
- Base model: VGG16
- Data Augmentation
- GlobalAveragePooling2D
- 3 Dense layers
- BatchNormalization
- Dropout
- Early stopping
- Accuracy:
  - **Train: 0.014**
  - **Validation: 0.013**

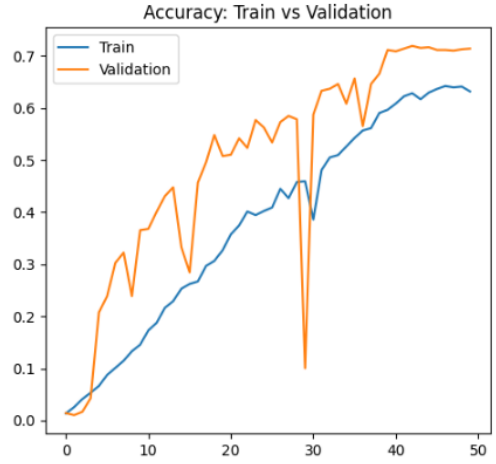




ResNet50 + Custom 'top' model

- Base model: ResNet50
- Data Augmentation
- GlobalAveragePooling2D
- 3 Dense layers
- BatchNormalization
- Dropout
- Early stopping
- Accuracy:
  - **Train: 0.632**
  - **Validation: 0.714**

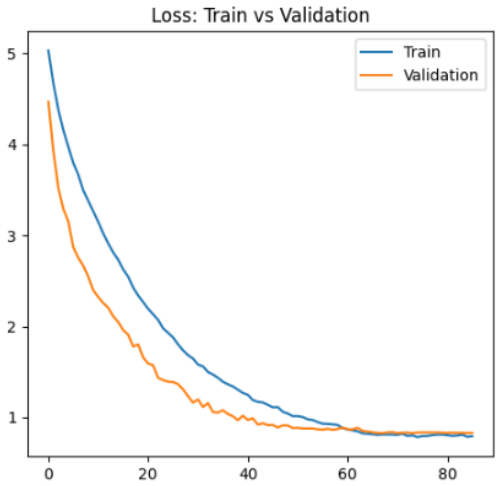


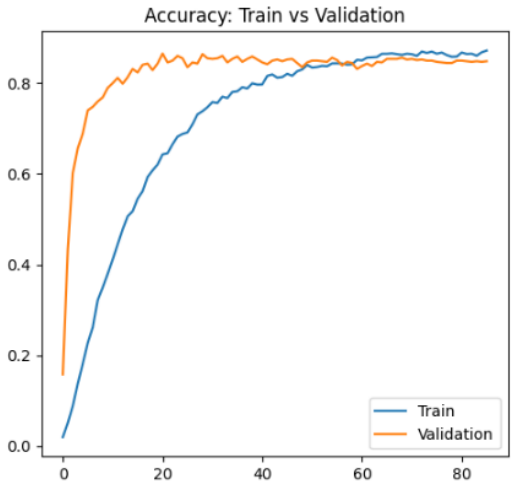


Modeling – Pre-trained Models (Best model)

Xception + Custom 'top' model

- Base model: VGG16
- Data Augmentation
- GlobalAveragePooling2D
- 3 Dense layers
- BatchNormalization
- Dropout
- Early stopping
- Accuracy:
  - **Train: 0.872**
  - **Validation: 0.849**
  - **21.1 mil parameters**



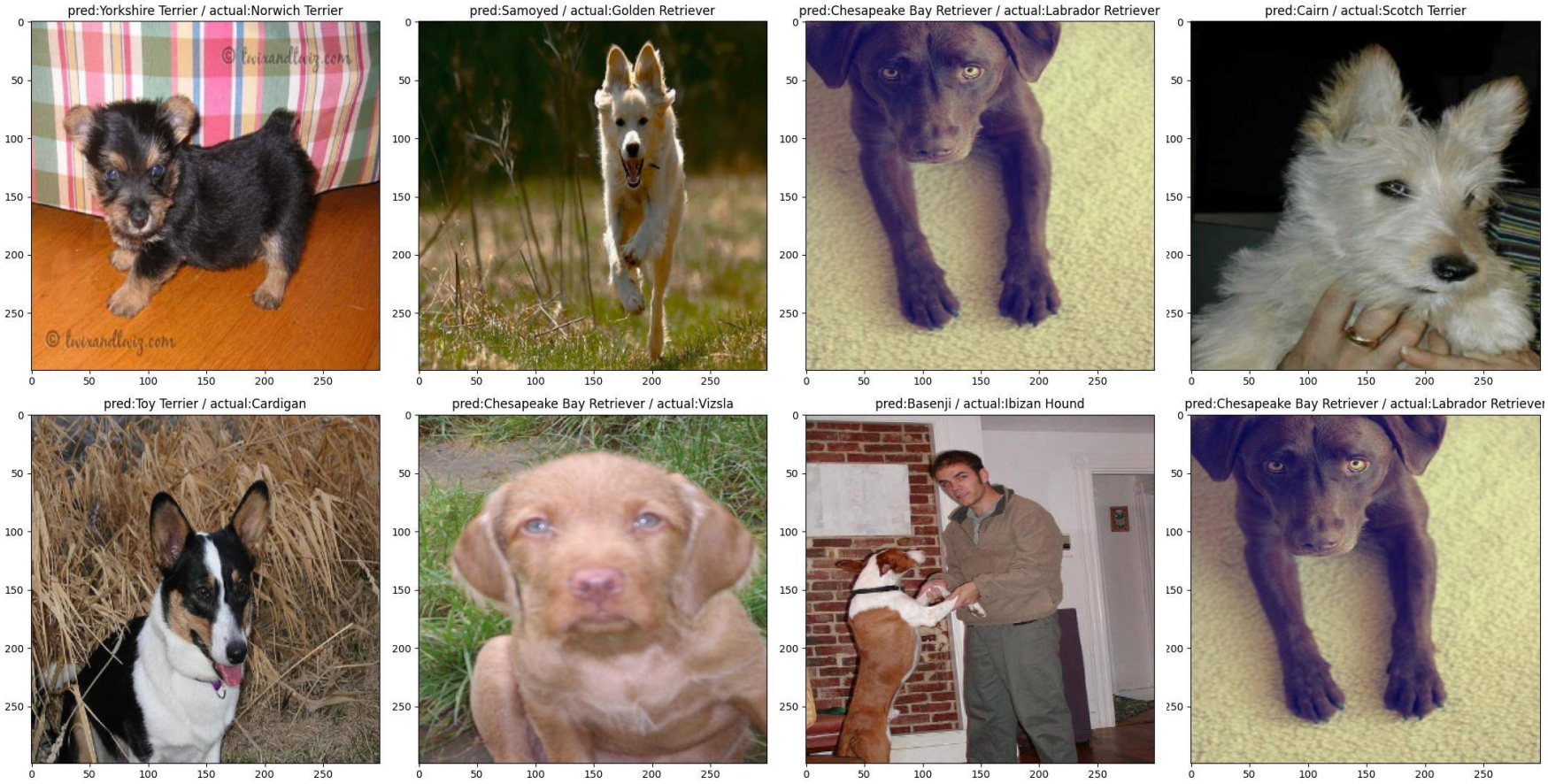


Evaluation – test images

Accuracy: 0.897

Accuracy	Precision	Recall	F1-Score
0.897	0.906	0.896	0.896

Evaluation - Misclassified images



Predictions

```
1/1 [==============================] - 0s 21ms/step
Your dog is a Doberman
Doberman: 1.0%
Black And Tan Coonhound: 0.0%
Malamute: 0.0%
```

```
1/1 [==============================] - 0s 21ms/step
Your dog is a Rottweiler
Rottweiler: 0.98%
Labrador Retriever: 0.0%
Doberman: 0.0%
```

```
1/1 [==============================] - 0s 21ms/step
Your dog is a Pug
Pug: 0.98%
Brabancon Griffon: 0.01%
Bull Mastiff: 0.0%
```

```
1/1 [==============================] - 0s 24ms/step
Your dog is a Papillon
Papillon: 0.62%
Pomeranian: 0.09%
Japanese Spaniel: 0.02%
```

Streamlit application

Dog Breed Detector

Choose an option:

- (•) Option 1: Upload a picture of your dog
- ( ) Option 2: Take a picture of your dog using webcam

Upload a picture of your dog:

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

Task 2 – Object Detection

Object Detection – YOLOv8

Task 2 - Object Detection: build an object detection model to identify a specific dog breed, Norwich Terriers, within images and videos.

- Success criteria mean Average Precision at IoU threshold 0.50 (mAP50) score of 0.80 or higher.

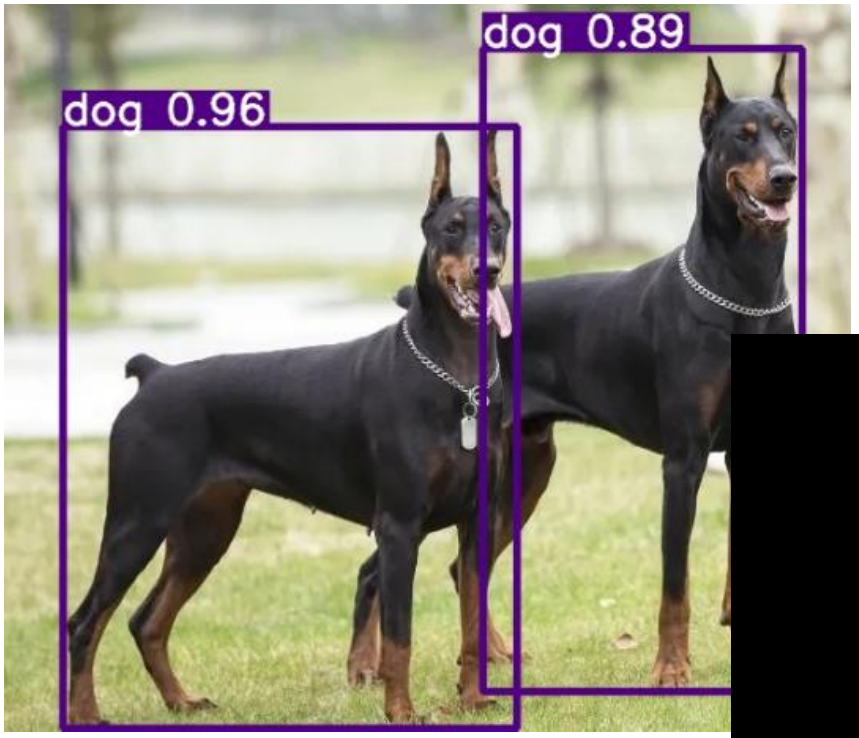
YOLO (You Only Look Once)

- YOLOv8 is the newest state-of-the-art YOLO model that can be used for object detection, image classification, and instance segmentation tasks
- Developed by Ultralytics
- 5 pre-trained models available (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x) with varying speed and performance

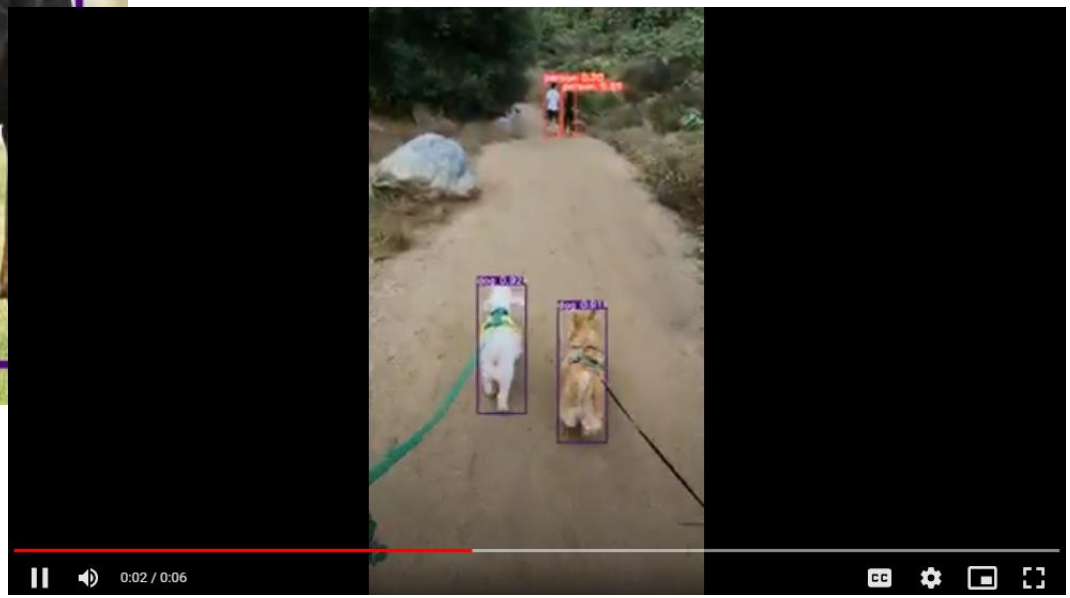
Pre-trained YOLOv8s model

Applied pre-trained YOLOv8s model to assess its ability to accurately detect objects such as dogs (and people) in both images and videos

Image



Video



YOLOv8l model on custom data

Process:

Annotate images
(drawing bounding boxes on the target object)

RoboFlow

→ **Train YOLOv8 model on annotated images** →

Make prediction

Iteration 1:

Source: Kaggle (same source for image classification task)
Data:

- 152 images of Norwich Terriers and Beagles
- 364 total number of images after data augmentation

Model: YOLOv8m

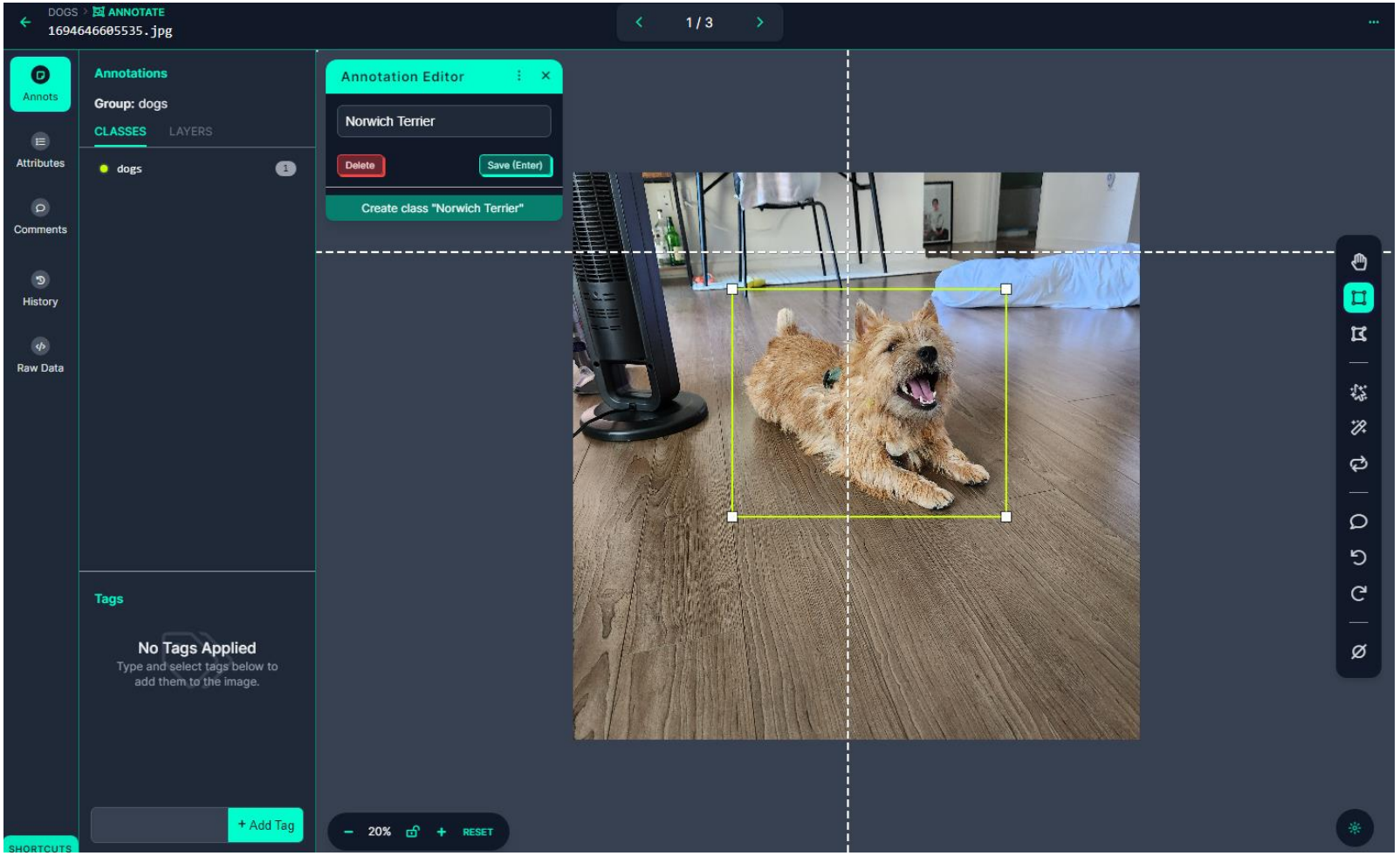
Iteration 2:

Source: Captured numerous images of a Norwich Terrier
Data:

- 211 images of a Norwich Terrier
- 507 total number of images after data augmentation

Model: YOLOv8l

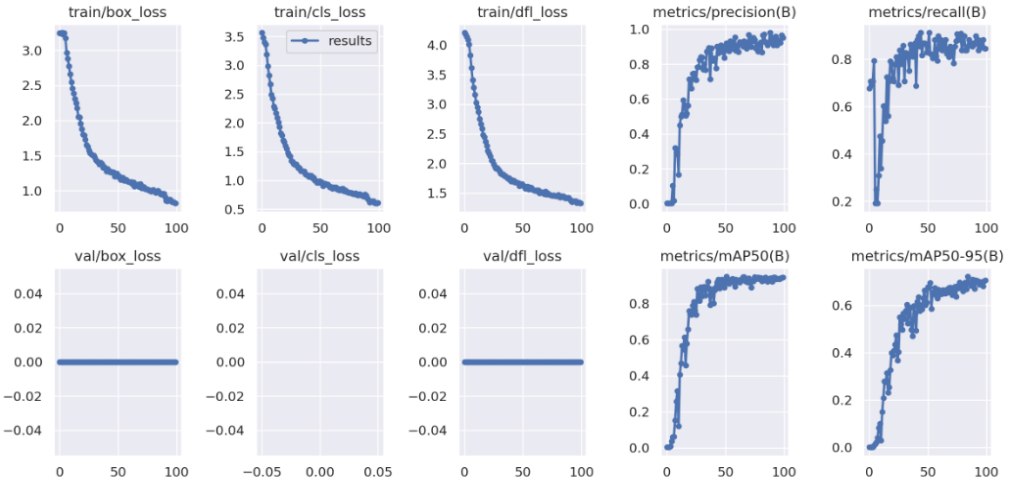
Annotating images using RoboFlow



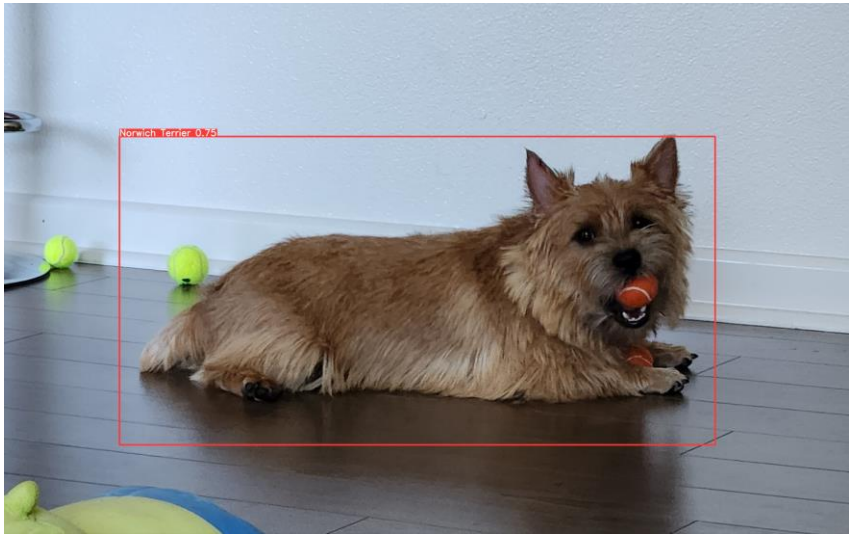
Modeling & Results

Training

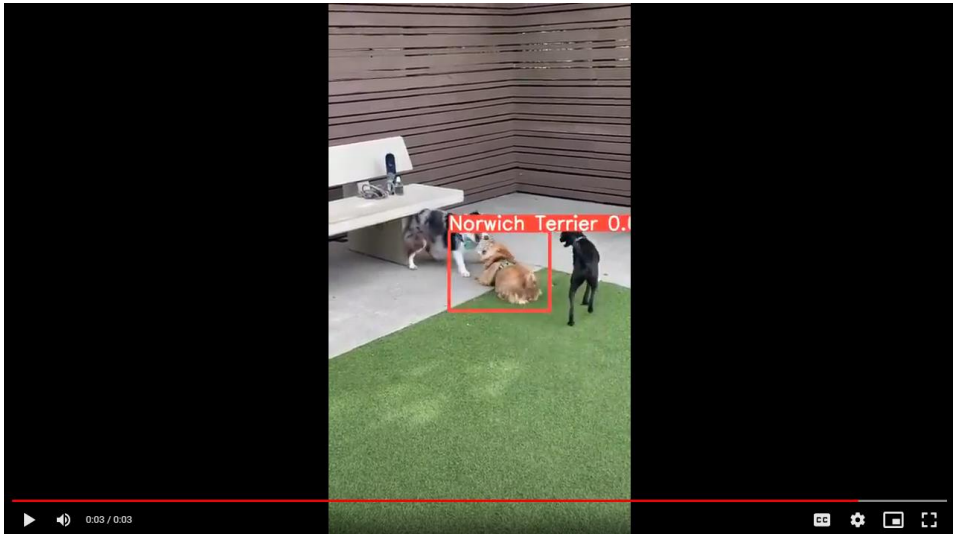
- Model: yolov8l
- Image size: 640 x 640
- Epoch: 100
- train: 444 images
- validation: 42 images
- test: 21 images
- **mAP50: 0.947**



Image



Video



Conclusion

- **Task 1 – Image Classification**
  - Utilizing pre-trained models to build a custom image classifier resulted in significantly superior performance compared to constructing a model from scratch.
  - Among the three well-known pre-trained models used for this task, Xception outperformed the others. VGG16 failed to converge, and while ResNet50 achieved substantial improvements over VGG16, it still showed signs of underfitting.
  - The superior performance of Xception can likely be attributed to its unique architecture (i.e., depthwise separable convolutional layer) → reduces the number of parameters, making it more memory and computationally efficient, helping to prevent overfitting and accelerate training
  - On the other hand, VGG16 is known for its deep architecture with numerous layers, which can hinder convergence due to vanishing gradients.
  - As for ResNet50, its residual connections, while effective in mitigating vanishing gradients, may lead to underfitting when the dataset is not extensive enough, as was the case for this task.

- **Task 2 – Object Detection**
  - The second YOLOv8 model, trained exclusively on Norwich Terrier images, not only effectively detected Norwich Terriers in both images and videos but also distinguished them from other dog breeds, achieving an impressive mAP50 score of 0.947. This enhanced model performance can be attributed to various factors:
    - Utilizing a more advanced and comprehensive model (YOLOv8l) compared to the initial iterations (YOLOv8m)
    - Increasing the number of training epochs from 70 to 100.
    - Expanding the dataset with a larger number of images.
    - Applying more sophisticated data augmentation techniques.
    - Focusing the training exclusively on Norwich Terriers without introducing other breeds, which allowed the model to learn more effectively.