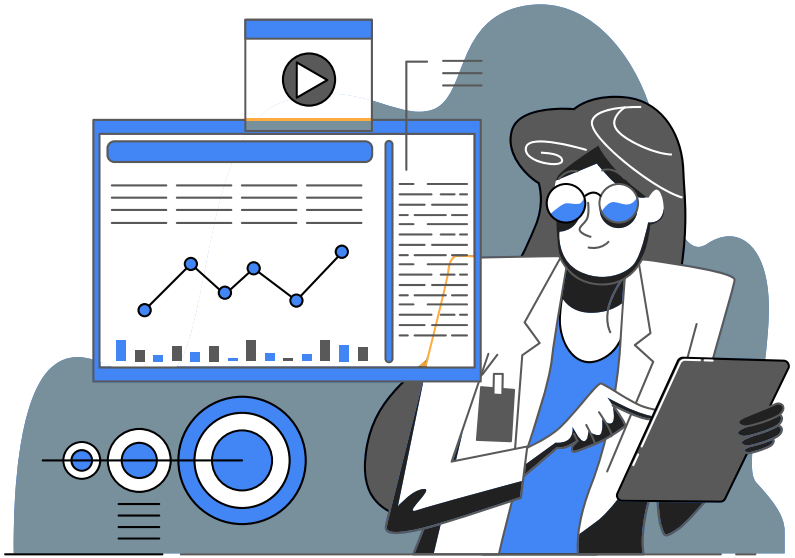

TikTok

Dog Breed Detector

By: Anirudh

CS 181 - Cs for inquiry

Table of Contents



Project Vision

Some Considerations

Is there a dog in this picture?...

What is this?

Its something I haven't done

1. Predict Dog Breeds from a given input Picture.
2. How is this done?
   a. Transfer Learning using existing Object Detector Models
   b. Deep Learning
   c. Trained on a lot of cute dog pictures!!

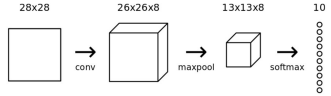
Starting Points

Libraries

TensorFlow

NumPy

Keras



jupyter

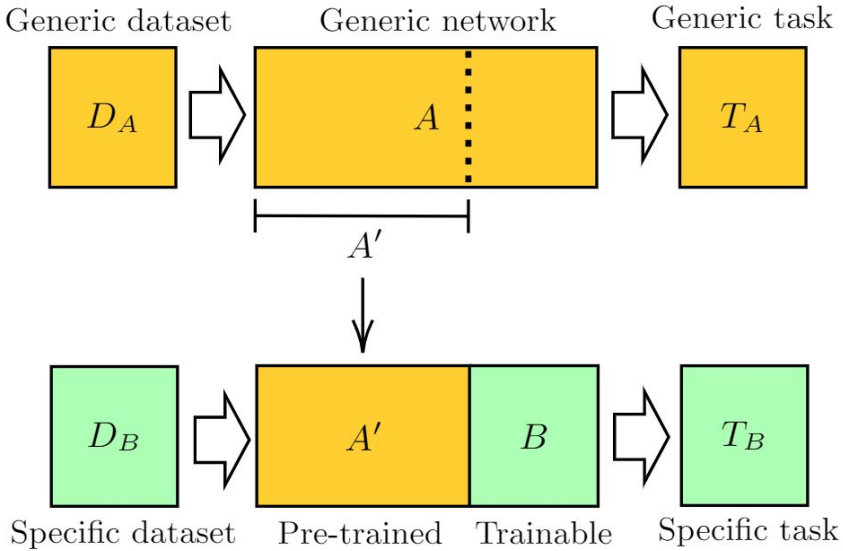
kaggle

matplotlib


Progress...

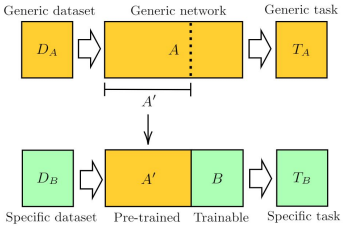
Methods

- **Transfer Learning**
  - The use of a previously learned model on a new problem



Methods

- **Transfer Learning:**
- Xception Model, trained on ImageNet from Keras
    - Removed the final convolutional fully connected layers
    - Used for Feature Extraction
    - This allowed me to define my input size
- Used Global Pooling layer to remove dimensions from output of Xception model; allows for variable input size.
- Other layers of Xception set to untrainable
- Added fully connected layer with as many outputs as breeds we want to classify



Training Procedure

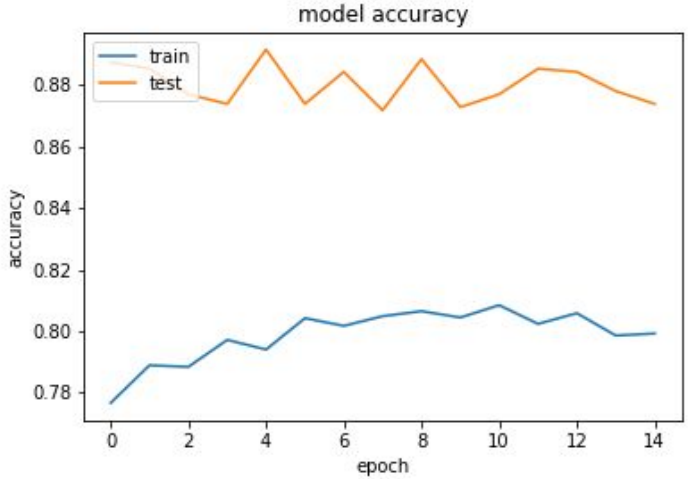
- Kaggle Data set of dogs.
- Data augmentation:
  - Orientation, scaling, color, brightness and more
- Training parameters:
  - Learning Rate: 0.001
  - Epochs: 50
  - Batch size: full set
  - **Took about 1.5 hours**

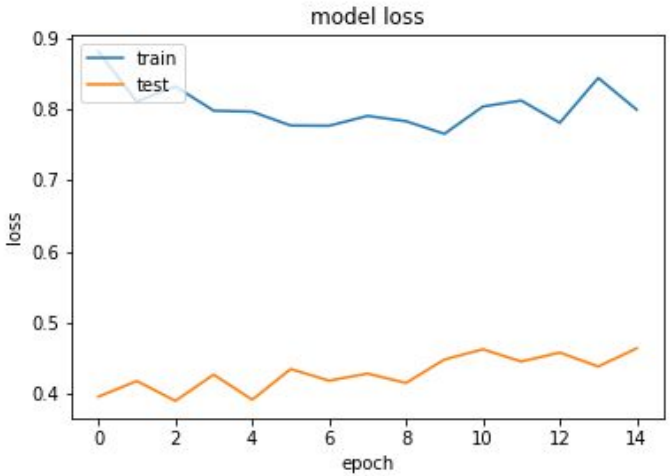
```
total number of breeds to classify 120
```

	id	breed	filename
0	000bec180eb18c7604dcecc8fe0dba07	boston_bull	train/000bec180eb18c7604dcecc8fe0dba07.jpg
1	001513dfcb2ffafc82cccf4d8bbaba97	dingo	train/001513dfcb2ffafc82cccf4d8bbaba97.jpg
2	001cdf01b096e06d78e9e5112d419397	pekinese	train/001cdf01b096e06d78e9e5112d419397.jpg
3	00214f311d5d2247d5dfe4fe24b2303d	bluetick	train/00214f311d5d2247d5dfe4fe24b2303d.jpg
4	0021f9ceb3235effd7fcde7f7538ed62	golden_retriever	train/0021f9ceb3235effd7fcde7f7538ed62.jpg

Results

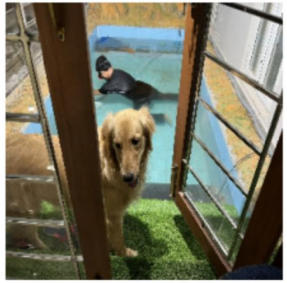
Model Training





Results

```
# image path
img_path = 'kyra.JPG'    # dog
predict_from_image(img_path)
```



'golden_retriever'

```
# image path
img_path = 'percy.JPG'    # dog
predict_from_image(img_path)
```

'toy_terrier'

```
# image path
img_path = 'twodogs.JPG'    # dog
predict_from_image(img_path)
```

'labrador_retriever'

```
# image path
img_path = 'poodle.jpeg'    # dog
predict_from_image(img_path)
```

'miniature_poodle'

```
# image path
img_path = 'cat.JPG'    # dog
predict_from_image(img_path)
```

'ibizan_hound'

```
# image path
img_path = 'goldenRetriever.jpeg'    # dog
predict_from_image(img_path)
```

'golden_retriever'

Results

```
# image path
img_path = 'bulldog.jpeg'    # dog
predict_from_image(img_path)
```

```
'bull_mastiff'
```

```
# image path
img_path = 'tester4.jpeg'    # dog
predict_from_image(img_path)
```

```
'great_dane'
```

```
# image path
img_path = 'doge.jpeg'    # dog
predict_from_image(img_path)
```

```
'dingo'
```

What's Next?...

- Fail Gracefully (only output when there is a dog)
- Make it a useable interface, instead of just a CLI
- Try different parameters during training for a better performing model
- Exploring with different baseline models

Thank you!