ARTIFICIAL INTELLIGENCE

REPORT

Prajwal B - 01FB15ECS209
Rahul Naren Pujari - 01FB15ECS223
Eshwa Gadkar - 01FB16ECS707

AIM:

(a) Use the PASCAL VOC2010 dataset, build a classifier. We will build the classifier to classify 20 across classes using Keras, CNN based models.
(b) Perform detecting object localization : predict bounding box for each image.
You will receive a separate question paper or lab instructions as per the progress.
(c) Evaluate the performance and experiment with transfer learning.

Requirements :

Python 2.7, TensorFlow, Keras, RAM >=8GB etc.

PART A: IMAGE CLASSIFIER

We've built a classifier using a CNN with Keras-

First, we used the annotation files present in the dataset to obtain the images and their corresponding labels.
Then we preprocessed the images to make them smaller and uniform, and then converted them into numpy arrays. The labels were converted into one-hot vectors. Thus, our training/testing data consists of the pixel arrays along with the one-hot labels.

- Optimizer used - Adam
- Loss function - Binary Cross Entropy
- Batch Size - 64
- Epochs - 20
- Accuracy - 19%

Due to inefficient way of coding we saw the accuracy to be poor. But we're planning to optimize the code in future.

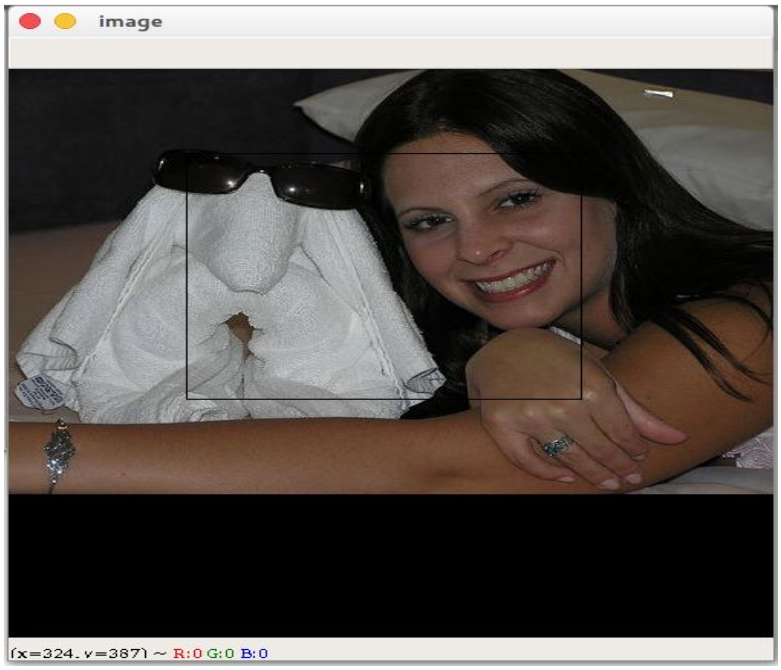
PART B: IMAGE OBJECT DETECTION

We've built an object detector which uses the concept of localization to put up a bounding box on the object to be detected on the image.
The coordinates of the bounding box are retrieved from the xml file(Annotations).

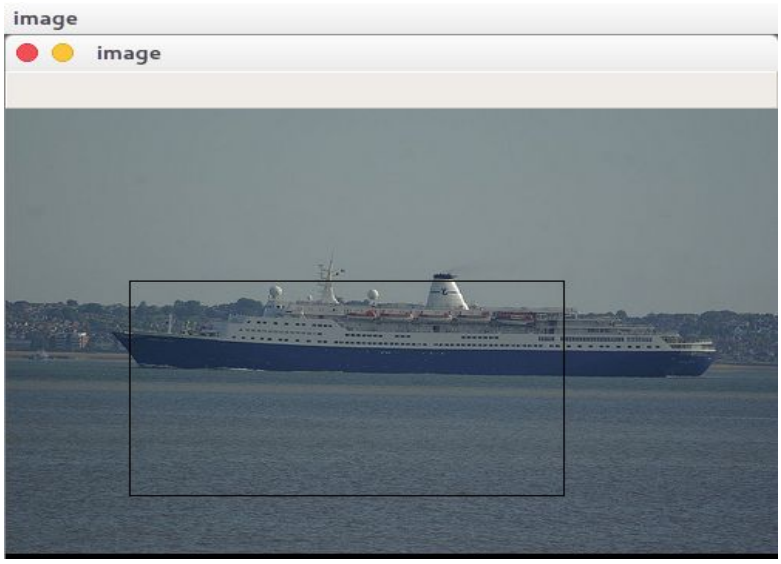
- Optimizer used - Adam
- Loss function - Mean Squared Error
- Batch Size - 1
- Epochs - 10
- Accuracy - 79%

Output Screenshots:

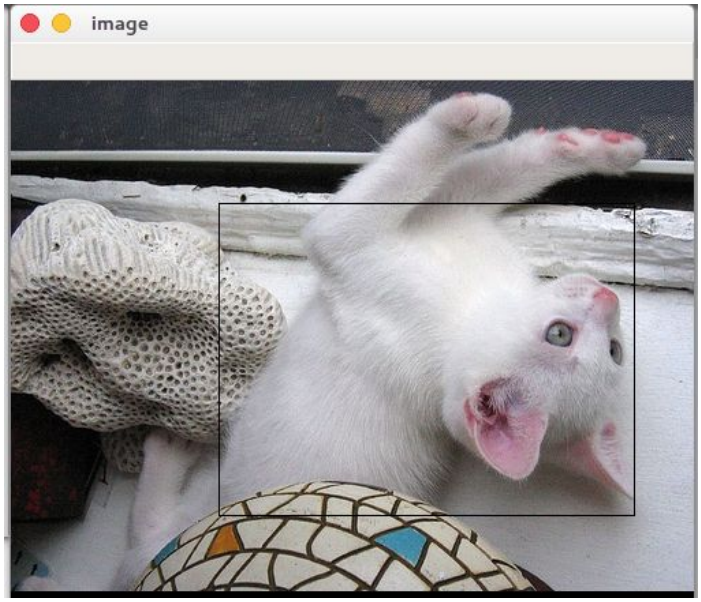
Person Class:



Boat Class:



Cat Class:



PART C: TRANSFER LEARNING

We have implemented the Classification of images using the pre-trained model **VGG16.**
To the existing VGG model, we added a dense layer with 1024 units with ReLU activation, and finally another dense layer with 15 units and a softmax activation.
We then froze the VGG layers, and only trained the new ones using images from the VOC dataset. After that, we froze the bottom 16 layers to train the only the top layers of the existing VGG network and trained it with more images.
We only trained it with around 500 images and 2 epochs due to resource constraints. Overall, we were able to achieve around 65% accuracy.

Output Screenshot: