Logo Detection using RCNNs

By: Kabir Ahuja and Satyam Singh

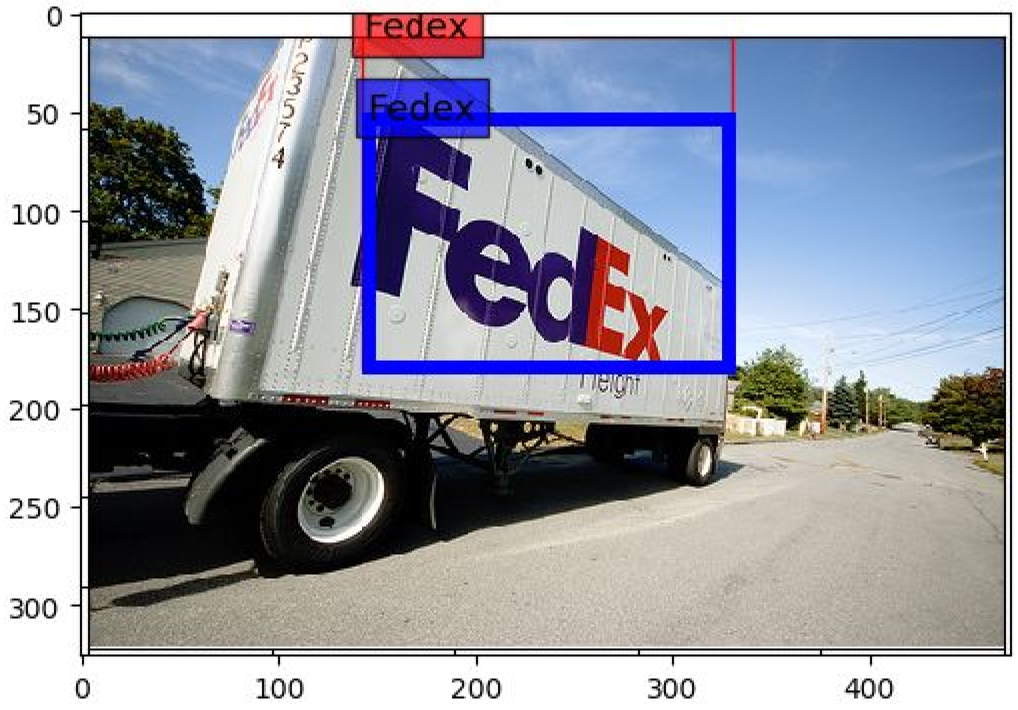


Fig1. Red bounding box represents the ground truth and blue bounding box represents the prediction of our model.

Approach:

Flickr's 27 logo dataset (link) for building our logo recognition model. It has logos of 27 categories like Apple, Coca Cola, McDonalds, FedEx etc. There are 4k images of logos in the dataset with the annotations which gives the coordinates of the logo in the image.

We used RCNN method which is an object detection method for this problem. In RCNN first we train a convolutional neural network on the patches of objects and backgrounds. Our CNN contains 3 convolutional and 3 pooling layers which are followed by 2 fully connected layer, the last layer giving the probability of the image patch belonging to a particular category. Relu activation function was used for all layers except for last layer where we used a softmax activation function to convert the logits into probabilities. Cross Entropy Loss function was used and Adam Optimizer with a learning rate of 0.0001 was used to optimize the loss. The code was implemented in python using TensorFlow 1.4.1

After training the model, to use this model for object detection we use Selective Search Algorithm to get the region proposals and then we pass these region proposals to our trained model and the model predicts to which category a region proposal works. Since our model is trained on the images of plain background too it is able to distinguish whether a proposal is a logo of a particular category or is just a plain background.

However due to the presence of many region proposals during training we get many detections of the same object. To overcome this method we use a technique called Non Max Suppression (NMS). In NMS what we take the results of the predictions on all the region proposals, first we only select those patches which were predicted with a probability greater than a certain threshold. After that we choose the proposal with the maximum probability and calculate its Intersection over Union (IOU) metric with the other proposals. IOU is defined as the ratio of the intersection of 2 patches and union of those 2 patches. We define a maximum IOU value and only choose those proposals whose iou with the the region of max probability is less than the maximum iou value. This allows us to reject the multiple detections of the same object in the image. After that we chose the next detection with highest probability and repeat the above steps.

In summary following are the sequence of steps performed to detect logos in an image:

1. Use Selective Search to get Region Proposals
2. Pass all these region proposals to the pretrained CNN model, and for each patch detect to which category of logo it belongs and also the probability with which it belongs to that class.
3. After that apply NMS to get the refined results.

To extend our model for detecting Exxon Mobil logos, we can take a few labelled images with Exxon Mobil logos and then fine tune our pre trained model by just training the last layer of our model while keeping the parameters of all the previous layers fixed.

Running the code:

Installing dependencies:

We use the following python libraries in our implementation:

a. Tensorflow 1.4.1 (pip install tensorflow)
b. numpy (pip install numpy)
c. Matplotlib (pip install matplotlib)
d. Selective Search (pip install selectivesearch)

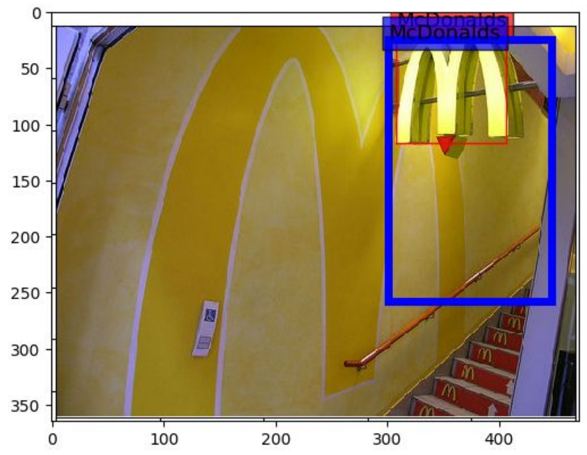
e. Skimage (pip install skimage)

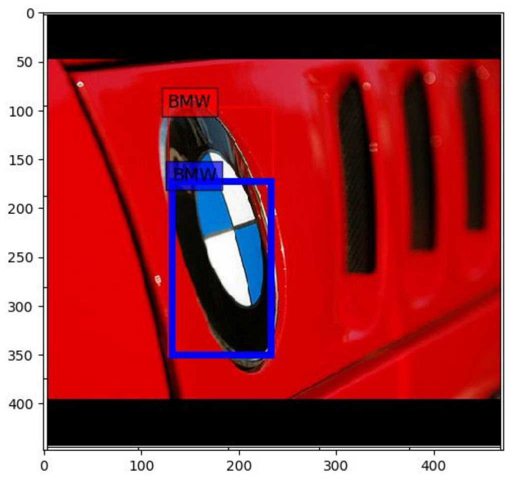
Following steps are to be followed for running the code:

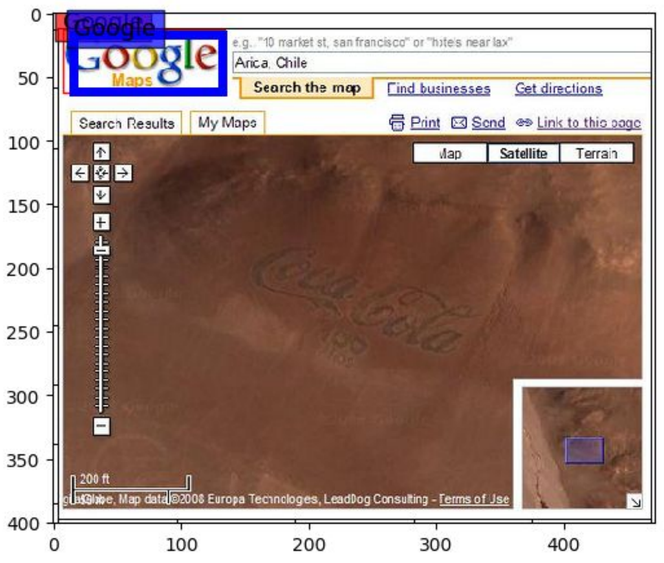
1. Run read_data.py which will essentially first generate the patches of background patches from the existing images and the creates a pickle file containing the image of different types of logos with their class labels. Note: Generating Background patches takes a lot time, hence we have provided the text file which contains the location and coordinates of the background images plus the logo images (A similar file for logos is provided in the dataset, we extend this file with the images of background) and have commented to the code which does that. Feel free to uncomment that if you want to run that too.
2. Run train.py which trains the CNN model on the patches of different logos. Since training might take a lot of time, one can skip this step and download our pretrained network from (link) and extract the contents to a folder checkpoints in the working directory.
3. Run detect_logos.py which tests the model on a set of images which are present in the folder test_images. The results are stored in the results folder. One can add the images of their choice in the test_images and run detect_logos.py to test the model on the custom images

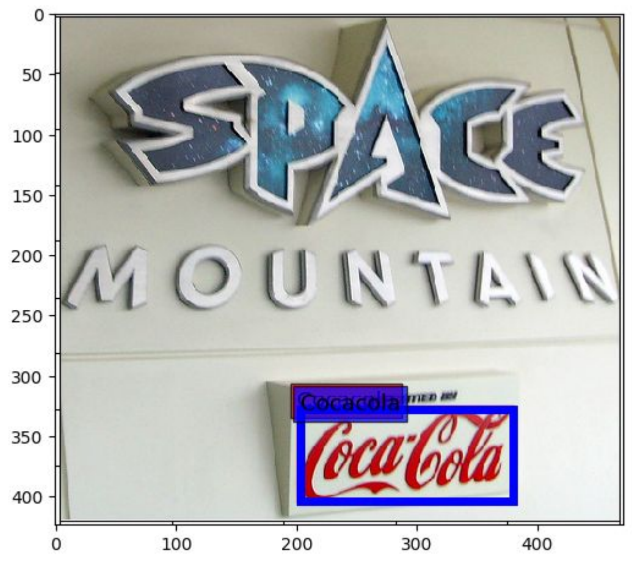
Results:

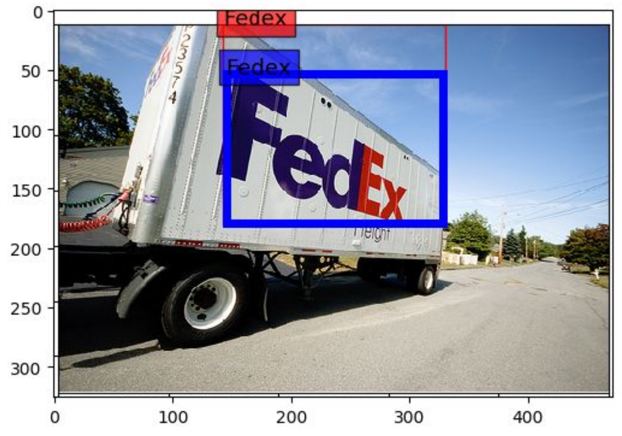
Following are the results of our model, red bounding box is the ground label and blue bounding box represents our model's predictions.

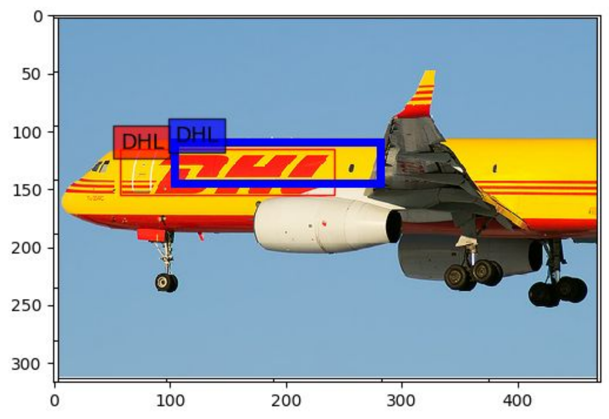


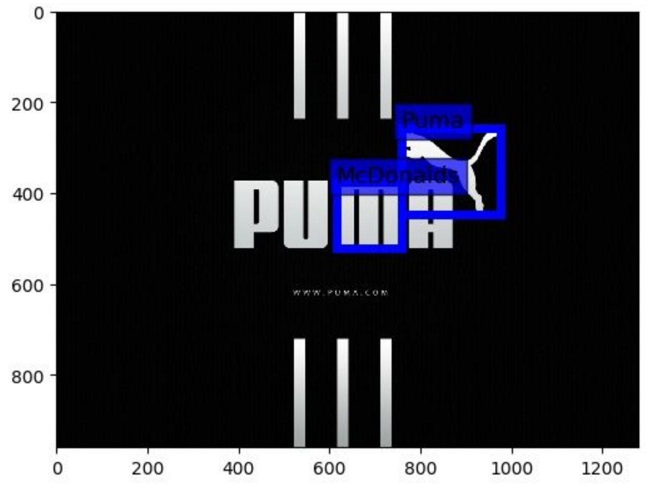



Google
e.g. "10 market st, san francisco" or "hotels near lax"
Arica, Chile
Search the map
Find businesses
Get directions
Search Results
My Maps
Print
Send
Link to this page
Map
Satellite
Terrain
200 ft
©2008 Europa Technologies, LeadDog Consulting - Terms of Use


SPACE
MOUNTAIN
Cocacola
Coca-Cola


Fedex
Fedex
FedEx
Freight


DHL
DHL
0
50
100
150
200
250
300
0
100
200
300
400


Puma
McDonalds
PUMA
WWW.PUMA.COM
0
200
400
600
800
0
200
400
600
800
1000
1200