Brand Exposure

Business Objective

Branding is one of the most important marketing strategies which helps a brand to grow. There are different ways these marketing strategies are carried out. For example, holdings, television ad breaks, during a cricket match, etc. Branding helps the brands stand out in a crowded market and hence they intend to spend resources on brand awareness in order to reach the targeted customers.

Return on investment (ROI) is a performance measure used to evaluate the efficiency or profitability of an investment made. For example, let’s say Pepsi company launched a new flavour of the drink and started advertising it via commercial breaks on television. The company invests some sort of amount in this branding of their drink and the return on investment will be nothing but the number of sales of the drink.

In our project, we are checking our brand exposure by calculating some KPI metrics on an IPL match video. Our main aim is to find the number of times a brand logo is shown during the match (total appearances), the largest and smallest area percentage, and the total area of the logo per count for every logo during the match.

Data Description

For our case study, we will be using a video clip of an IPL match played between CSK and RCB teams. This dataset is downloaded from YouTube and is 2 minutes 35 seconds long.
Further processing on this data will convert the video clip into frames. Using annotators, we will convert the data into xml files which will further be converted into CSV files on basis of our key performance metrics.

Aim

To find the KPI metrics for each brand logo, such as the number of appearances of the logo, the area, frames, shortest and longest area percentage, for the given input video clip.

Tech stack

- Language - Python
- Libraries – TensorFlow, pillow, opencv, matplotlib, NumPy, uuid

Approach

1. Download the input video
    - From YouTube
    - Using python (Youtube_downloader.py)
2. Use the annotation tool (LabelImg) to convert the images into XML file
3. Set up tensorflow for object detection i.e git clone
4. Convert the XML files to CSV files (the CSV files contain the width, height, depth, xmin, ymin, xmax, ymax per image)
5. Convert the csv file into tfrecords file (for train and test)
6. Download the base model from tensorflow model zoo1 (eg: ssd resnet 50fpn coco) and unzip the file.
7. From the folder (ssd resnet 50fpn coco) open the pipeline.config file and make the changes as shown in the video
8. Now open the model's folder, open the research folder, followed by object detection folder, and open the legacy folder. In the legacy, folder opens the train.py file and start the training.
9. Once the training is completed, the results are saved in ckpt files.
10. The results of the training can be monitored on the tensor board by using the events file generated along with the ckpt files.
11. Freeze the model - Cktp have 3 files, data, index, and meta, and these three files have to be combined into one file and this is called freezing of the model.
12. From the models/research/object_detection folder opens the export_inference_graph.py. Perform the command as shown in the video. This generates the frozen_inference_graph.pb which will be used for further inferences.
13. The frozen_inference_graph.pb, the labels.txt, and the test_video.mp4 are the final inputs on basis of which we will measure the KPI metrics.
14. Make predictions on the test images using the predict.py file
15. Tweak visualization utils to return box classes scores
16. Process the video into frames
17. Finally, run the video_processing.ipynb generate the output i.e find the KPI metrics for each brand logo.

Modular code overview

```
input
  |_test_video.mp4
  |_labels.txt
  |_frozen_inference_graph.pb


src
  |_Engine.py
  |_ML_Pipeline
            |_admin.py
            |_predict.py
            |_utility.py



lib
  |_youtube_downloader.py
  |_xml_to_csv.py
  |_generate_tfrecord.py
  |_Detection_scripts.ipynb
  |_video_processing.ipynb


output
  |_dictonary
```

Once you unzip the modular_code.zip file you can find the following folders within it.

1. input

2. src

3. output

4. lib

1. Input folder – The input folder consists of
    - test_video.mp4
    - labels.txt
    - frozen_inference_graph.pb

2. src folder - This is the most important folder of the project. This folder contains all the modularized code for all the above steps in a modularized manner. This folder consists of:
    - Engine.py

- ML_Pipeline

The ML_pipeline is a folder that contains all the functions put into different python files which are appropriately named. These python functions are then called inside the engine.py file.

3. Output folder **–**

The output will be a dictionary which will consists of key performance metrics per logo. They are, count i.e., the total number of appearances per logo in the video, the total area, frames, shortest and longest area percentage per logo.

4. lib folder **-** This is a reference folder. It contains the original ipython notebook that we saw in the videos. Also, the reference ppt is provided.

Project Takeaways

1. Understanding the business problem.
2. Learning how the download the video data via python.
3. Using annotation tool (LabelImg) for generating the XML files.
4. Understanding what and how to clone model.
5. Learning how to convert the XML files to CSV files.
6. Learning how to convert the CSV files to tfrecords files.
7. Understanding the base model concept (such as ssd resnet 50)
8. Understanding how to train the model from cloned models
9. Visualization via the tensor boards.
10. Importing the dataset and required libraries.
11. Understanding the concept of CKPT file in tensorflow
12. Generating frozen model from ckpts.
13. Making predictions using the trained model.
14. Perform tweaking to get bounding boxes in order to get KPI metrics
15. Understanding how to process the videos and break them down into frames.
16. Understanding how to process the frames to gain predictions.
17. Calculating the various KPI metrics such as the number of appearances of logo, the area, frame, shortest and largest area percentage.