Brand Visibility Calculation in Sport Videos

Lakshmi Chandana Yarramreddy (50418855),

Kshitij Subhash Chhajed (50432890)

{lyarramr, kchhajed} @buffalo.edu

***Abstract*—**In this paper we tried to derive best machine learning technique to detect the brand logos from sport videos like cricket match footage. We constructed two different machine learning models using YOLOv3 algorithm, Detectron2 algorithm to identify shortlisted brand logos from given video input. We trained the models with training data extracted manually from IPL 2022 cricket videos. As a result of our project, we analyzed the performance of both models and concluded that Detectron2 is good choice for the targeted task as its object detection capacity is good with very less training epochs.

Keywords—Brand visibility, YOLO algorithm, Detectron algorithm, IPL (Indian Premier League), Object detection algorithms

I. INTRODUCTION

Advertising is the main source of income for many sports leagues now a days. Considering this fact many advertising agencies are quoting huge amounts for the brand to advertise their products in these games. Companies on the other hand wants to advertise their products in the best front for reasonable amount of money. This forms the basis for our project. Machine learning tools can be employed to calculate the cost of advertisement based on the brand visibility. These tools can help advertising agencies to showcase the brand visibility related to advertisement at the event locations and can also help brands to understand the effectiveness of their investment in advertisements. It's very difficult to identify the brand logos from a streaming video and then to identify the brand value. Hence, we decided to create different models to detect the logos accurately and then to calculate the brand value based on the results from these models. Identifying the correct model for logo detection in this scenario is cruel for overall model performance.

II. DATA PREPARATION

One of the major problem we faced in this project was that there was no existing dataset for the problem statement we have chosen. Most of the datasets available for object detection are for solid objects like cars, banners etc. But our problem is completely related to moving frames containing different angles of the players and stadium. We needed more relevant dataset with which the models can be trained accurately.

We opted for using Indian Premier League 2022, Cricket match recordings for our model's training and testing purposes. Extracted key frames from each video and annotated them as per model requirements. We have used LabelImg [12] and LabelMe [8] tools for annotating the frames with targeted logos. YOLOv3[16] model accepts training images with rectangular annotations with txt files, unlike this algorithm, Detectron2[3] accepts training frames with polygonal annotations in json format. Hence, we opted for two different tools to annotate the training frames.

III. DATA SOURCE

Data used in this project are manually extracted from highlights of famous cricket matches in the IPL 2022 season, which are collected from the IPL official website [9] (https://www.iplt20.com). For this project we chose three brands namely Jio, Tata and Paytm. We extracted all the frames from the video with a span of 1 sec using python script. Then manually shortlisted the frames with above mentioned brand logos. Total training data is of around 500 frames for each model, including all the brands. We annotated all the test data with both the tools LabelImg and LabelMe, so that this data can be used to train both the models.

IV. Related Work

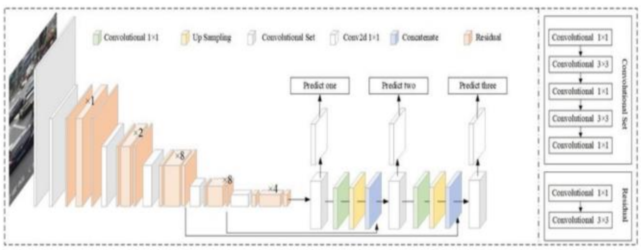
Currently, the impact of advertisement placement is calculated using a variety of statistical and analytical approaches, and the display time is estimated manually for most of the cases. However, [1] has attempted a serious usage of object detection algorithms for the analysis of brand visibility. They have made use of Yolov2 model and calculated the logo exposure using product of video duration with the ratio of number of frames a logo was detected and the total number of frames. [14] and [15] have made similar attempts using Yolo algorithm, which are specifically targeted for using object detection for identifying and classification of brand logos.

V. Proposed approach

A. YOLOv3 Model

To improve upon the existing works, we tried to use Yolov3 Model which is improved version from Yolov2 Model. It is based on Darknet-53 architecture which is significant improvement over Darknet-19 architecture for Yolov2.

Darknet is a C-based implementation of neural network (especially CNN) which works using CUDA. It is an open-source framework which forms basis for different version of Yolo algorithm.



Structure detail of YOLOv3.It uses Darknet-53 as the backbone network and uses three scale predictions.

B. Training of model:

We used pre trained darknet53.conv.74 weights as base weights and trained YOLO model using our custom dataset to generate new weights for every 1000 epochs. These final weights obtained after 6000+ epochs were used for testing the model in the next steps.

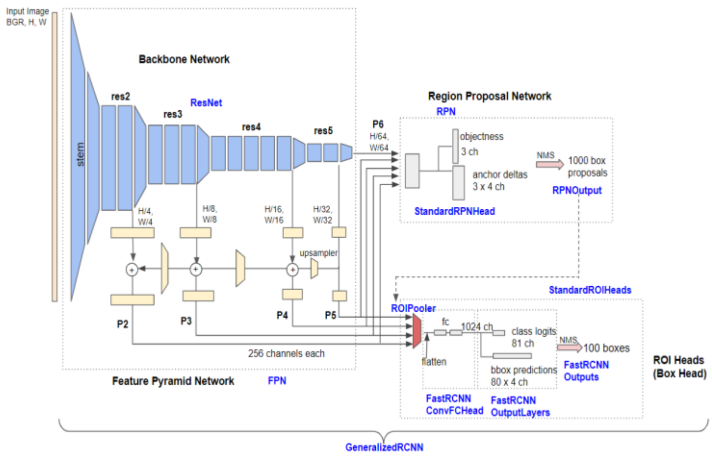
C. Testing of the model:

We gave the custom labels in the order jio, paytm, tata. Hence the label values for these brands would be 0,1,2 respectively. We used the final weights generated form the training model to calculate the brand visibility in each test video input.

D. Detectron2 Model

Detectron2 is developed by Facebook AI Research (FAIR) team to achieve state-of-the-art results in object detection and segmentation. It is preceded by Detectron[4] model and Mask-RCNN benchmark.

It has various models like Mask-RCNN, RetinaNet, RPN, TensorMask, etc. We have used a version of Faster R-CNN for current project name as ‘faster_rcnn_R_101_FPN_3x’. It is based on the COCO dataset and its configurations.



VI. Results

A. YOLOv3:

Yolov3 model we implemented is successfully detecting the logos and surpassing the unwanted or logo less frames. Model extracted the frames and highlighted the brand logos with corresponding similarity scores as shown below. We have chosen a threshold of 0.5 similarity score to count the identified box as a logo.

Fig1: Jio logo is detected with 0.89 similarity

Fig2: Paytm logo is detected with 0.9 similarity

Fig3: tata logos are detected with 0.97 and 0.93 similarity

Fig4: Multiple logo (tata, paytm) detection in the same frame

B. Brand visibility calculation from YOLO results:

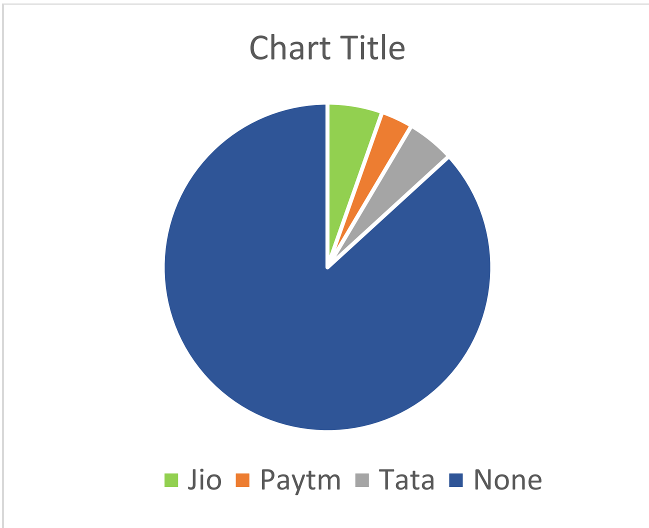
For testing, we have chosen a video of length of 5 min 40 sec (340 seconds) from IPL 2022 matches.

After testing the results are fallows:
Total number of frames processed: 704
Total number of frames with Jio logo: 38
Total number of frames with Paytm logo: 22
Total number of frames with Tata logo: 33
Total duration of video = 340 sec

Brand visibility = (total number of frames one brand is detected * Total duration of match)/No of frames a logo is detected.

Brand visibility for Jio = (38*340)/93 = 140
Brand visibility for Paytm = (22*340)/93 = 80
Brand visibility for Tata = (33*340)/93 = 120

Below is the pie chart to represent the brand visibility for each brand in total input video.



C. Detectron2 results:

Detectron2 was able to detect the logos correctly. However, we were unable to calculate the visibility of different brands with this model. The model has inbuilt provision to provide evaluation results for all classes however, there is no method for class-wise accuracy calculations.

```
[08/20 14:04:37 d2.engine.train_loop]: Starting training from iteration 0
[08/20 14:04:52 d2.utils.events]:  eta: 0:03:32  iter: 19  total_loss: 1.671  loss_cls: 1.487  loss_box_reg: 0.0004611
[08/20 14:05:07 d2.utils.events]:  eta: 0:03:17  iter: 39  total_loss: 1.339  loss_cls: 1.119  loss_box_reg: 0.002989
[08/20 14:05:22 d2.utils.events]:  eta: 0:03:02  iter: 59  total_loss: 0.7342  loss_cls: 0.5483  loss_box_reg: 0.008233
[08/20 14:05:38 d2.utils.events]:  eta: 0:02:47  iter: 79  total_loss: 0.3912  loss_cls: 0.2522  loss_box_reg: 0.01425
[08/20 14:05:53 d2.utils.events]:  eta: 0:02:33  iter: 99  total_loss: 0.2299  loss_cls: 0.1512  loss_box_reg: 0.04567
[08/20 14:06:08 d2.utils.events]:  eta: 0:02:18  iter: 119  total_loss: 0.2094  loss_cls: 0.1124  loss_box_reg: 0.05379
[08/20 14:06:24 d2.utils.events]:  eta: 0:02:03  iter: 139  total_loss: 0.2275  loss_cls: 0.1127  loss_box_reg: 0.06104
[08/20 14:06:39 d2.utils.events]:  eta: 0:01:47  iter: 159  total_loss: 0.2253  loss_cls: 0.116  loss_box_reg: 0.06965
[08/20 14:06:54 d2.utils.events]:  eta: 0:01:32  iter: 179  total_loss: 0.2484  loss_cls: 0.1309  loss_box_reg: 0.1006
[08/20 14:07:09 d2.utils.events]:  eta: 0:01:17  iter: 199  total_loss: 0.3319  loss_cls: 0.1566  loss_box_reg: 0.1379
[08/20 14:07:25 d2.utils.events]:  eta: 0:01:02  iter: 219  total_loss: 0.2166  loss_cls: 0.1021  loss_box_reg: 0.07605
[08/20 14:07:40 d2.utils.events]:  eta: 0:00:47  iter: 239  total_loss: 0.2982  loss_cls: 0.1394  loss_box_reg: 0.1247
[08/20 14:07:55 d2.utils.events]:  eta: 0:00:31  iter: 259  total_loss: 0.2405  loss_cls: 0.124  loss_box_reg: 0.1044
[08/20 14:08:10 d2.utils.events]:  eta: 0:00:15  iter: 279  total_loss: 0.321  loss_cls: 0.1445  loss_box_reg: 0.129
[08/20 14:08:26 d2.utils.events]:  eta: 0:00:00  iter: 299  total_loss: 0.275  loss_cls: 0.1215  loss_box_reg: 0.1461
[08/20 14:08:26 d2.engine.hooks]: Overall training speed: 298 iterations in 0:03:46 (0.7594 s / it)
[08/20 14:08:26 d2.engine.hooks]: Total training time: 0:03:47 (0:00:01 on hooks)
```

In comparison with Yolov3, Detectron2 took very less time for training and reached minimum loss function within 150-160 iterations. However, the detection was not at par with Yolov3. Major reason for this could be the availability of enough training data.

VII. CONCLUSION AND FUTURE WORK

The Detectron2 model can be fine-tuned to provide accuracy for a given class. This model can be developed into a stand-alone app to calculate the brand visibility. Both the advertising agencies and brands can use this model for brand recognition and advertising cost estimations.

VIII. BIBLIOGRAPHY

1. Identification of Brand Logos from Video Feed; Pranshu Garg; Anmol Tripathi; Ashish Kumar Sahani: https://ieeexplore.ieee.org/document/9342654
2. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proc. IEEE Computer Society Conference Computer Vision Pattern Recognition., vol- 2016 December
3. https://github.com/facebookresearch/detectron2
4. https://github.com/facebookresearch/Detectron
5. https://towardsdatascience.com/object-detection-in-6-steps-using-detectron2-705b92575578
6. https://github.com/pjreddie/darknet
7. https://www.analyticsvidhya.com/blog/2021/08/your-guide-to-object-detection-with-detectron2-in-pytorch/
8. https://github.com/wkentaro/labelme
9. https://www.iplt20.com/video/46226/q2-rr-vs-rcb--match-highlights?tagNames=2022
10. https://github.com/jakkcoder/Training_Yolo_Custom_Object_Detection_files
11. https://pysource.com/2020/04/02/train-yolo-to-detect-a-custom-object-online-with-free-gpu/
12. https://github.com/heartexlabs/labelImg
13. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks https://arxiv.org/pdf/1506.01497v3.pdf
14. https://github.com/dqhuy140598/LogoDetectionV2#brand-and-logo-detection-in-image-using-yolov3
15. https://github.com/akarshzingade/Logo-Detection-YOLOv2
16. YOLOv3: An Incremental Improvement - https://arxiv.org/pdf/1804.02767.pdf