OBJECTIFIER

Muhammad Abdullah Usmani | Usama Saif

National University Of Computer And Emerging Sciences, Karachi

19K-1377
19K-0248
Section: F

ABSTRACT

Object detection is one of the exemplary issues in computer vision. It is very significant for computers/machines to perceive the common objects in life like the human mind and the human eye can. This article makes sense of the working of an object identification framework that uses both pretrained and custom trained models that have a place within the group of most advanced Convolutional Neural Networks [3] there are to recognize and detect objects. Some of the ever so growing applications of object detection problem are also discussed in this article.

OBJECTIVES

The development of this system occurred considering the accompanying targets:

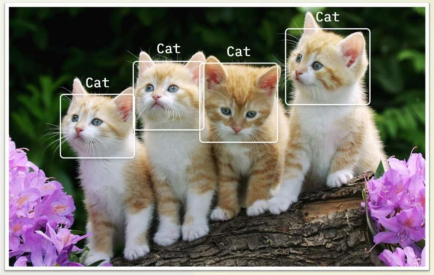
1. To comprehend the use of most recent pretrained Convolutional Neural Network architectural models.
2. To perform object detection using pretrained EfficientDet-D0 512x512 and mobilenet_v2_fpnlite_320x320 architectural models.
3. To perform object detection using uniquely prepared EfficientDet-D0 model and mobilenet_v2_fpnlite_320x320 architectural models.

INTRODUCTION

Object recognition and detection is a critical computer vision task used to recognize instances of visual articles of explicit classes (for example, individuals, animals, vehicles, or designs) in advanced pictures, for instance, photos or video outlines. The target of object detection is to encourage computational models that give the most focal information expected by computer vision applications.

Object discovery is inseparably connected to other comparative computer vision strategies and problems like image recognition and image segmentation, in that it helps us comprehend and investigate scenes/frames in pictures or videos. In any case, there are significant contrasts between image recognition, image detection and image segmentation. image recognition just results a class mark for a distinguished item/object, and image segmentation makes a pixel-level comprehension of a picture's components. What isolates object detection from these different errands is its extraordinary capacity to find objects inside a picture or video. This then permits us to count and afterward track those objects.

METHODOLOGY



- **Pre-trained models setup Applications:**

To comprehend the use of pretrained object detection models, we, utilizing python language and its ever so developing arrangement of functionalities, figured out how to concoct a python script, upon providing the link of a particular CNN model from TensorFlow model zoo[], downloads, loads and sets up that model for sure fire use, all in no time.

- **Displaying annotations:**

Utilizing cv2, a well-known python computer vision package, we figured out how to draw annotation boxes over objects detected by the models.

- **Dataset:**

We utilized the notorious PASCAL VOC2011 [1] dataset to prepare our Efficient Net and MobileNet models. This dataset contains 14,000+ pictures with 20 different article classes.

- **Data cleaning:**

We fostered a python content to clean the PASCAL VOC2011 dataset

- **Training models:**

We utilized EfficientNet-D0 512x512 and mobilenet_v2_fpnlite_320x320 architectures given in TensorFlow zoo [2] as a base for making our own special modified Efficient Net and MobileNet models.

We arranged the previously mentioned architectures in a manner so that they becomes viable with our chosen dataset and environment.

RESULTS

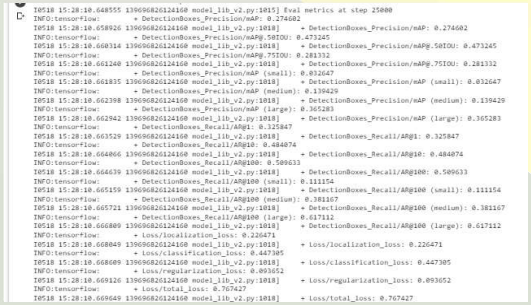
Four different types of Convolutional Neural Networks were used in this system with each of them having their own precision, accuracy and loss.

- When it comes to pre-trained Efficient Net D0 512x512, its mean average Precision is 33.6%.
- When it comes to pre-trained SSD MobileNet V2 FPNLite 420x320, its mean average precision is 22.2

The above stated facts can be verified from Tensor Flow zoo [2]

- When it comes to pre-trained Efficient Net D0 512x512 and SSD MobileNet V2 FPNLite 420x320 models, their precision and accuracy is not the best due to the fact that they were trained for only 10,000 steps in average due to the unavailability of GPUs. Still, these models showed somewhat promising results and gave us hope that with GPUs, they can perform much better. The Efficient Net model was evaluated on 3665 images and the results are presented below:

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.275
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.473
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.281
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.033
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.139
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.365
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.326
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.484
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.510
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.111
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.381
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.617
```



APPLICATIONS

Object detection is one of the basic issues of computer vision. It shapes the premise of numerous other downstream computer vision assignments, for example, instance segmentation, image captioning, object tracking, pedestrian detection, People counting, face detection, text detection , pose detection , number-plate recognition , Crowd counting, Self-driving cars, Video surveillance ,Face detection , Anomaly detection etc.

CONCLUSION

In summary, we developed a system known as Objecifier allows its users to perform object detection utilizing 2 of the latest Tensor Flow pretrained models. This system also allows users to detect objects using custom trained Efficient Det and MobileNet models.

We achieved these features by performing the following steps:

1. Cleaning the ever so popular PASCAL VOC2011 dataset [1] so that it can be used for the purpose of training our models.
2. Downloading and configuring MobileNet and Efficient Net architectures from Tensor Flow zoo [2] to make them compatible with our environment
3. Training our configured models.
4. Testing our models and analyzing their result.

REFERENCES

The following sites/ articles were referenced in this article:

- **[1] http://host.robots.ox.ac.uk/pascal/VOC/voc2011/index.html**
- **[2] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md**
- **[3] arXiv:1511.08458v2**