FORMAT-2

Web Development for Object Identification

BY

Jallepalli Harsha Vardhan(19030141CSE047)

Shaik Sameer Roshan(19030141CSE013)

Bhimavarapu Dhanush Krishna Sai Reddy(19030141CSE086)

Podalakunta Kottigi Suchith(19030141CSE083)

Semester: V

Department of Computer Science & Engineering

Alliance College of Engineering and Design

ALLIANCE UNIVERSITY

Chikkahagade cross, Chandapura- Anekal Main Road Bangalore-562106

Batch: 2019-23

INDEX

ABSTRACT

Real time object detection is a vast, vibrant and complex area of computer vision. If there is a single object to be detected in an image, it is known as Image Localization and if there are multiple objects in an image, then it is Object Detection. This detects the semantic objects of a class in digital images and videos. The applications of real time object detection include tracking objects, video surveillance, pedestrian detection, people counting, self-driving cars, face detection, ball tracking in sports and many more. Convolution Neural Networks is a representative tool of Deep learning to detect objects using OpenCV (Opensource Computer Vision), which is a library of programming functions mainly aimed at real-time computer vision.

Introduction

Object Detection is the process of finding and recognizing real-world object instances such as car, bike, TV, flowers, and humans out of an images or videos. An object detection technique lets you understand the details of an image or a video as it allows for the recognition, localization, and detection of multiple objects within an image.

It is usually utilized in applications like image retrieval, security, surveillance, and advanced driver assistance systems (ADAS). Object Detection is done through many ways:

- Feature Based Object Detection
- Viola Jones Object Detection
- SVM Classifications with HOG Features
- Deep Learning Object Detection

Algorithm/Design flow

Step-1: We first take a pretrained keras framework
Step-2: From keras framework we import keras applications
Step-3: We create a Django project to upload an image
Step-4: We install TensorFlow packages for backend of project
Step-5: The keras model is retrained based on number of classes that needed to be detected
Step-6: Finally, whenever the image is upload the trained keras model identifies the image with accuracy

Circuit diagram / Model information

Keras

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

Tensorflow

Tensor flow is an open source software library for high performance numerical computation. It allows simple deployment of computation across a range of platforms (CPUs, GPUs, TPUs) due to its versatile design also from desktops to clusters of servers to mobile and edge devices. Tensor flow was designed and developed by researchers and engineers from the Google Brain team at intervals Google's AI organization, it comes with robust support for machine learning and deep learning and theversatile numerical computation core is used across several alternative scientific domains.

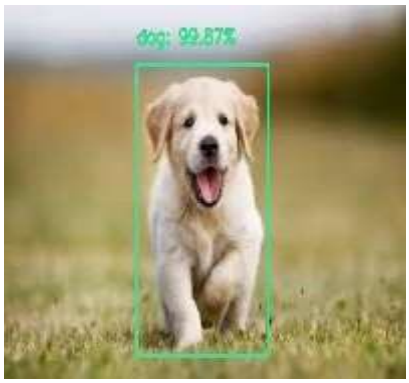
ResNet

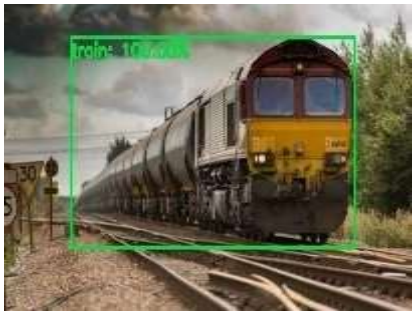
To train the network model in a more effective manner, we herein adopt the same strategy as that used for DSSD (the performance of the residual network is better than that of the VGG network). The goal is to improve accuracy. However, the first implemented for the modification was the replacement of the VGG network which is used in the original SSD with ResNet. We will also add a series of convolution feature layers at the end of the underlying network. These feature layers will gradually be reduced in size that allowed prediction of the detection results on multiple scales. When the input size is given as 300 and 320, although the ResNet–101 layer is deeper than the VGG–16 layer, it is experimentally known that it replaces the SSD's underlying convolution network with a residual network, and it does not improve its accuracy but rather decreases it

Results (Comparative analysis)

Input

Output





Conclusion

Deep learning-based object detection has been a research hotspot in recent years. This project starts on generic object detection pipelines which provide base architectures for other related tasks. With the help of this the three other common tasks, namely object detection, face detection and pedestrian detection, can be accomplished. Authors accomplished this by combing two things: Object detection with deep learning and OpenCV and Efficient, threaded video streams with OpenCV. The camera sensor noise and lightening condition can change the result as it can create problem in recognizing the object. The result is a deep learning- based object detector that can process around 6-8 FPS.

Future scope Reference

Here are a some of the future implementation of object detection.

1. Face detections and recognition:

Face detection perhaps be a separate class of object detection. We wonder how some applications like Facebook, Face app, etc., detect and recognize our faces. this is often a sample example of object detection in our day-to-day life. Face detection is already in use in our lifestyle to unlock our mobile phones and for other security systems to scale back rate.

2. Object tracking:

Object detection is additionally utilized in tracking objects like tracking an individual and his actions, continuously monitoring a ball within the game of Football or Cricket. As there's an enormous interest for people in these games, these tracking techniques enables them to know it during a better way and obtain some additional information. Tracking of the ball is of maximal importance in any ball-based games to automatically record the movement of the ball and adjust the video frame accordingly.

3. Self-driving cars:

this is often one among the main evolutions of the planet and is that the best example why we'd like object detection. so as for a car to travel to the specified destination automatically with nonhuman interference or to form decisions whether to accelerate or to use brakes and to spot the objects around it. this needs object detection.

4. Emotion’s detection:

this permits the system to spot the type of emotion the person puts on his face. the corporate Apple has already tried to use this by detecting the emotion of the user and converting it into a respective emoji within the smart phone.

5. Biometric identification through retina scan:

Retina scan through iris code is one among the techniques utilized in high security systems because it is one among the foremost accurate and unique biometric.

6. Smart text search and text selection (Google lens)

In recent times, we've encountered an application in smart phones called google lens. this will recognize the text and images and search the relevant information within the browser without much effort.

Bibliography

References

Krizhevsky, I. Sutskever and G. E. Hinton, ""Imagenet classification with deep convolutional neural networks"" in Advances in Neural Information Processing Systems 25, Curran Associates, Inc., pp. 1097-1105, 2012. Google scholar

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going deeper with convolutions", *The IEEE Conference on ComputerVision and Pattern Recognition (CVPR)*, June 2015. Google scholar

K. Chatfield, K. Simonyan, A. Vedaldi and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets", *British Machine Vision Conference*, 2012. Google scholar