IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

REAL TIME OBJECT DETECTION MODULES Core description about the modules used for real time object detection

[1]Rinkesh U Patel,[2]Meet S Patel,[3]Dev A Thakkar, [4]Bhumika Bhatt

[1]B.E. I.T,[2]B.E. I.T,[3]B.E. I.T, [4]Assistant Professor
[1]Information Technology
[1]Sardar Patel College of Engineering, Bakrol, Anand,India

Abstract:The advancement of technology leads to increasing accuracy of computer vision such as machine learning in which a machine learns itself, without explicit programming. The field deep learning which is a subset of machine learning in Artificial intelligence uses patterns used in decision making has made task of image processing much easier. This paper will focus on TensorFlow by google. The trained cocomodels are taken with input which in turn produces output boxes where objects are identified within frames with percentage match and labels. This survey paper will elaborate the actual working of TensorFlow. The working of COCOAPI models includes capturing, storing and comparing thousands of images with real time objects and percentage matches are described with its process in ensuring literature surveys. Apart from this, paper also includes some details about yolo for training object detection models.

Keywords: TensorFlow, cocoapi, weights, models, OpenCV, image processing, deep learning, artificial intelligence, machine learning.

I. INTRODUCTION

Many prominent and intellectual citizens from different walks of life uses machine learning knowingly or unknowingly. As per current scenario the near future seems to follow tradition based on image processing. Nowadays face detection, eye scannerare used to unlock devices, recommendation systems are also widely used and all of them are based on image processing. The most popular machine learning library in the world is Google's Tensorflow developed by their team named as google brain. Every single product of google use machine learning in some or the other way whether its image searching, image capturing, caption translation and so on. Google needs machine learning to take advantage of such bulky datasets. Tensorflow library was built to scale it and made to run on multiple CPUs and GPUs. There are wrapper classes available in several languages with python being the most used language where python package manager is used to install TensorFlow commonly known as pip. pip install TensorFlow isthe command used to install TensorFlowon different machines based on their CPU and GPU. Initially, libraries are loaded and models are downloaded as per the requirement. There are various Coco dataset models available based on cnn, rcnn and faster_rcnn but among them faster_rcnn is found most accurate. Widely used model is ssd_mobilenet_v2_coco model which follows the architecture of SSD (Screen Shot Detection). In addition, protobuf-compiler is also used which convert proto files into c++, java and python source code that are cloned inside TensorFlowmodel.

II. LITERATURE SURVEY

2.1Detection of Real Time Objects Using TensorFlow and OpenCV

Object recognition and detection can be done in structured as well as in the unstructured environment. It is the most interesting and challenging job in computer science. It will be helping hand for people who had lost their eyes. Object detection technique has wide methods to implement. One of those is, by using sensors. It includes much complex hardware such as ultrasonic sensor, radar, stereo vision optical flow. Those made the product more expensive. A prototype model can be developed which is cost-effective.

There are two techniques for detecting living or non-living things, named as TensorFlow and PyTorch. TensorFlow is one of the source software libraries of python which is used for machine learning applications along with math as a core library. Whereas, PyTorch was developed by the AI research team of Facebook and Uber. It is an open-source library.

2.1.1 Appearance-based object detection

This method has shown very well result on detecting range of 3-D objects from smaller ones to larger. For instance, it could recognize cars, buses, trucks, airplanes, too. Additionally, it is programmed to find out which is the animal against the web camera. The project has been done on a large-scale that it has verified around more than 2000 separate images, thus, ultimately, it increases the performance of recognition and always keep updating rigorous dataset/database. The major problem happens with this is that the whole object frame must not separate as bottom-up. In the current system, it is done automatically for 2-D views of 3-D objects

2.1.2 Implementation

YOLO is an object detector which is used in the object recognition system and real-time object detection. YOLO is an abbreviation of “You only look once". It can capture 45 frames per second. Another thing to install is the webcam that is operated by programming in python. Specifically, it is run when we call the function of OpenCV. Functions for example, cv2.VideoCapture() is for reading incoming frames and method like read() is reading the images' pixels and return it to script. According to the dataset, it will match the pixels with objects resided in the dataset and will fetch the name of it.

2.2 Tensorflow: A system for large scale machine learning

Google brain team is developed specifically for making machines intelligent. One of their projects was development of TensorFlow system, model making and training them on larger datasets. The functional computation a traditional method previously used for graphical representation on immutable data is now replaced by tensor (multidimensional array) on mutable data. However, TensorFlow allows vertices to represent computations that update the state or own them. Previously used system was DistBelief developed for training neural networks on distributed system that google had used since 2011. DistBelief follows parameter server architecture and its limitations were tackled by development of TensorFlow.TensorFlow has benefits such as along with python it also provides API's in C++ which makes it very easier and flexible to implement, it also has a relatively faster compilation time than other deep learning libraries such as Keras and Torch. Unlike others libraries TensorFlow supports both CPU and GPU computing for any device, which is very beneficial because nowadays the volume of datasets used for deep learning is huge and CPU's computing power is not capable of handling such complex operations, so using CPU along with GPU is really the key to process hybrid deep learning applications especially games having a high graphics requirement. However, Tensorflow was not always considered as a tool for deep learning, originally it was developed to run large numerical computations.

Tensorflow uses Tensors which are multidimensional arrays to accept data. Input in neural networks are taken in the form of arrays having different dimensions and ranks. On the basis of rank a tensor dimension can be classified, for example, s=[200] is a tensor of rank 0 which contains only one element also known as a scalar, a tensor of rank 1 can be described as v=[10,11,12] a single row matrix and higher rank tensors are also possible due to multidimensionality. Tensorflow programs are executed on Data Flow Graphs which consists of a number of nodes and edges. Unlike typical programming, in TensorFlow we prepare graphs containing various nodes and which run in the form of a session using data from a tensor. TensorFlow operations can be arranged into a generally called a computational graph which is the graph of main logic of a program which TensorFlow builds in the memory. Very large-scale neural networks as computing can be distributed across several CPUs or GPUs in a same manner can be allowed by using it. blocks can be created from computational graphs which can run across distinct GPU and CPU simultaneously. This process is called parallel computation.

2.3 Microsoft COCO: Common Objects in Context

The dataset developed by Microsoft commonly known as COCO: Common Objects in COntext that means it contains objects that are easily recognized by a 4-year-old boy. This dataset contains 330K images with 1.5M object instances for 80 object categories including 5 captions per image. This is stored in annotation file stored with extension. json. The coco images are trained and validated and tested for all 80 categories. All object instances are annotated with detailed segmentation mask. This json file contains five different sections info, licenses, images, annotations, and categories. The info and licenses section only contains information related to dataset like date, contributor, year, URL and licenses. The images section has its own unique ID which is used to reference in annotations. Apart from that, cocourl and flickerurl specifies the path to images hosted. Widths and heights are stored in pixels. For each image they have distinct annotations that have different ID's, category_id that specifies the category to which the object belongs. Categories have their name, id and supercategory. To illustrate, if the name is bicycle then the supercategory is vehicle and that is having id as 2. iscrowd is a field of annotation that is specified by 0 and 1. If iscrowed is 0 that means image contains only one lion and is iscrowed is 1 than there is more than one lion in the image. Whole image is described in segmentation and values are stored in form of array as shown [[x,y,x,y][x,y,x,y]]. So, segmentation is list of points that defined image where actual object is present in the image. Counts are also defined in annotations which identifies the unshaded part or the part where object is not present. In other words, let us consider the array [147,3,1, 3, 89, ...], 147 represents that all these pixels are not the part of object followed digit 3 indicates these 3 pixels represents image. Next digit 1 shows that pixel is not a part of image and so on continues for rest of array.

By this way one can also create their own dataset for detecting their own specified images. Now cocoapi had grouped images into three different types called iconic-scene images, iconic-object images and non-iconic images. If we consider iconic-scene images they are aimed to focus on sharing two views in one common image whereas iconic-object images have only single large object and last non-iconic images lack important information regarding the contextual viewpoints. The goal of cocoapi was to create dataset that contains majority of non-iconic images that can improve object detection in general scenario. There are many popular datasets like ImageNet, PASCAL VOC 2012 and SUN. All these datasets differ in various ways like labeling, categories and types of images. But the fact that makes coco dataset emphasize is the use of non-iconic images which can easily recognize the objects during real time. Moreover, the comparison of average performance of DPMv5 suggests that performance of MS COCO was dropped by nearly 2 factors because of the non-iconic images selected to create dataset.

III. Conclusion

Agglomerating all the points elaborated aboveit can be concluded that, all modules explained above plays a vital role in image processing. The Tensorflow module act as a pathway for detecting objects. On top of it, a large set of images collected together in dataset by coco unable us for identifying objects of real world. Thus, this all modules helps me to create a project based on real time object detection. In contrast, there are certain aspects that need more research and have clear statistical information and improving the accuracy of object detection in real time scenario.

REFERENCES

[1] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[2] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... &Kudlur, M. (2016). Tensorflow: A system for large-scale machine learning. In 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16) (pp. 265-283).

[3] Zainab, Ameema. "Realtime Object Detection on Android using Tensorflow." Qatar Foundation Annual Research Conference Proceedings Volume 2018 Issue 3. Vol. 2018. No. 3. Hamad bin Khalifa University Press (HBKU Press), 2018.

[4] Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., ... &Zitnick, C. L. Microsoft COCO: Common objects in context. arXiv 2014. arXiv preprint arXiv:1405.0312.

[5] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[6] Shafiee, M. J., Chywl, B., Li, F., & Wong, A. (2017). Fast YOLO: A fast you only look once system for real-time embedded object detection in video. arXiv preprint arXiv:1709.05943.

[7] Francois, R., Eddelbuettel, D., Stokely, M., Ooms, J., &Eddelbuettel, M. D. (2019). Package 'RProtoBuf'.

[8] Pietraszek, T., & Tanner, A. (2005). Data mining and machine learning—towards reducing false positives in intrusion detection. Information security technical report, 10(3), 169-183.