International Journal Of Advance Research, Ideas And Innovations In Technology

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 4 - V7I4-1511)
Available online at: https://www.ijariit.com

Object segregation using R-CNN

Nihal Kumar Singh
nihal.kumar.singh@gmail.com
RajaRajeswari College of Engineering, Bengaluru, Karnataka

Aakash Singh
aakashsingh1999@gmail.com
RajaRajeswari College of Engineering, Bengaluru, Karnataka

Ashish Prasad
ashishamredra@gmail.com
RajaRajeswari College of Engineering, Bengaluru, Karnataka

S. Usha
sakthivelusha@gmail.com
RajaRajeswari College of Engineering, Bengaluru, Karnataka

ABSTRACT

Computer vision is the science of computers and package systems that can also recognize how images and scenes are perceived. PC Vision consists of a variety of solutions that include image recognition, object recognition, image generation, super resolution of images and much more. widely used in facial recognition, vehicle recognition, pedestrian counting, network mapping, security systems, and autonomous cars, but here we have a tendency to specialize in completely different sensible objects, those with different kinds of fruits, buttons, coins, etc. In this project we use extremely correct object recognition algorithms and methods like RCNN, FastRCNN, FasterRCNN, Mobilnet and fast but extremely correct methods like SSD. If we understand frameworks by using dependencies like TensorFlow, OpenCV etc., we can recognize every single object in the image through the realm object in a highlighted area and determine every single object and assign its label to the object. This also includes the precision of every technique used to distinguish between objects.

Keywords: *Object Detection, Tensorflow, Opencv, Python, Tensorboard, R-CNN*

1. INTRODUCTION

Object detection has already been the sole analysis for direction and thus the focus among the computer vision, which can be applied within the driverless car, robotics, police investigation, and pedestrian detection. Segmenting the image for object detection is the most important part of computer vision where the model needs to determine objects either from the background or any of the body parts or just static objects. Object detection could be a method wherein the objects are detected with the assistance of an algorithm detecting live or through an image or video feed. Before the emergence of deep learning technology, the ways of object detection are primarily accomplished by establishing the mathematical models supported by some previous knowledge. This further involve task that distinguish objects in digital photographs or through a live feed via a camera. Classification of image suggests that predicting the category of an object in an image. Object detection states that an image is known to examine 1 or more objects in an image and thereby encloses a bounding box around the boundary of the object detected. Object detection could also be used for detecting the hair follicles and could localize the follicles individually. First, we tend to construct different categories for training, then after training it would be able to detect an object we trained it for, and once after partitioning the image and generating records it would be able to load the model much more efficiently. The subsequent steps are mentioned below. during this project, we are going to be essentially focusing on the various reasonably objects that we see in our day to day life like different varieties of fruits, buttons, coins etcetera. The ways we tend to implement our project as:

1 Image DataSet Preparation
2 Generating TensorFlow Records
3 Implementation of R-CNN
4 coaching And Testing

Object detection will be done using R-CNN, the primary stage of R-CNN is that the generation of region proposals. It uses selective search algorithmic rules. The algorithm works by generating a sub-segmentation of the image that belongs to at least one object, either supported colour, size.

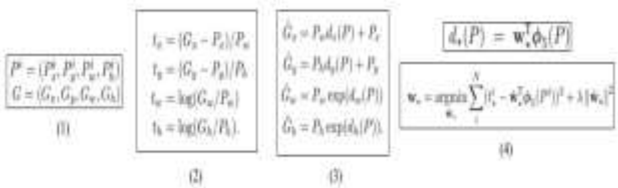

Fig 1: Bounding Box Equations used in R-CNN

The subsequent stage is property SVM categorizes and learns for every class that detects the presence or an absence of object belonging to specific category or class. The output of each stage could be a set of positive object proposals for every class. The subsequent stage is that the bounding box regression, this improves localization performance by employing a target transformation between our foreseen proposal and target proposal. The output of this stage is the positive region proposals of each class prognosticative from the SVM and thus, we tend to get the bounding box around the object.

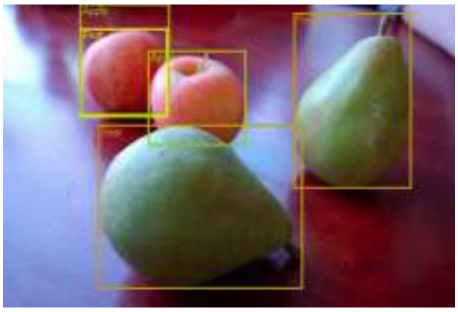

Fig 2: Bounding Box Example

2. EXISTING SYSTEM

In the existing paper, they have discussed about a graph based Image Segmentation and object identification method, especially where the background is not chaotic and do not contain several same type of object. Their method successfully identified the search object from the background. However the method lacks its identification where the object seems multipart such as same shape and colour objects. The existing system was not that accurate and was time consuming process for the object identification. The existing system was implemented using graph based image segmentation which made the model slow and less accurate as compared using Neural Networks.

3. PROPOSED SYSTEM

In the proposed system, we will be basically focussing on the different kind of object that we see in our day to day life like different kinds of fruits, buttons, coins etc. In this we will be focusing on images having various objects in an image it can be more than one object also. First we create a image dataset of the classification of different objects and then from the test image we search for the object after partitioning the image using a TensorFlow record generator. The following steps are discussed below.

1. Image Dataset Preparation
2. Creating TensorFlow record
3. Implementation of R-CNN
4. Training
5. Object Identification(Testing)

A. Objective of the work

The main objective of the work is to Identify different objects in an image with more than one type of object in the image.

B. *Methodology*

In this project, images are collected to form a data set with different types of objects present. After the data set is collected we train our machine in such a way that it identifies different objects in an image using neural networks. For testing we will take a image and let it pass through the algorithm and will check for the output. This process is done in few simple steps, which are:

1. Sample an open source dataset of images for detecting different objects in the image.
2. Train R-CNN to automatically detect objects in images via the dataset we created.
3. Evaluate the results from an educational perspective.

4. SYSTEM REQUIREMENTS

A. System analysis

Study of the feasibility: In terms of information, yield, projects and techniques, the assessment of this attainability depends on the design plan of the system prerequisite. Having defined a model system, the analysis will continue to recommend the kind of gear needed to build the framework, the technique required to run the framework once it has been designed. The undertaking is expanded to such a degree that, within the imperatives, the basic capacities and execution are fulfilled. Within the latest innovation, the task is created. Despite the fact that the technology may be obsolete after a certain time frame due to the way it never forms more developed variants in the same programming, the system is still being used. The system has been developed using convolution neural networks algorithm.

B. Functional requirement

Useful requirements describe the product's internal activities: that is, the technical subtleties, monitoring and handling of data and other specific functionality demonstrating how to satisfy the use cases. They are upheld by non-utilitarian prerequisites that force the plan or execution of imperatives.

- System should Process the dataset.
- System should detect the Objects.
- System should predict Objects using Training algorithm.

C. *Non-functional requirement*

Unnecessary prerequisites are requirements that suggest parameters that can be used to assess a framework's operation rather than specific activities. This should be distinguished from useful necessities indicating explicit behaviour or capabilities. Reliability, flexibility are common non-practical necessities. Non-practical preconditions are often referred to as system utilizes. Different terms for non-practical necessities are "limitations", "quality characteristics" and "prerequisites for administration". On the off chance that any special cases occur during the product execution, it should be obtained and keep the framework from slamming along these lines. The architecture should be created in order to incorporate new modules and functionalities, thereby promoting application development. The cost should be small as a result of programming packages being freely accessible.

- Usability System Should be user Friendly
- Reliability System should be Reliable
- Performance System Should not take excess time in detecting the object
- Support-ability System should be easily updatable for future enhancement

D. *TOOLS AND TECHNOLOGY USED*

1) HARDWARE:

- System: Intel i3 2.4 GHz.
- Hard Disk: 40 GB.
- Ram: 512 Mb Minimum

2) SOFTWARE:

- Operating system: Windows XP/Windows 7 or more
- Software Tool: Tensorflow 2.5, Open CV
- Coding Language: Python.

5. SYSTEM DESIGN

The system is design in such a way that any image can be classified and will help in identifying the object for future reference. The steps or modules involved in the design are:

1. Image DataSet Preparation:

For each object class we prepared a set of characteristics from sample images where the objects are labelled manually. Each entry is defined by the different values sets in the dataset. All the possible images or a particular image will be classified in one entry so that if the image is taken from some other angle or some other view point even then image can be identified.

2. Creating TensorFlow Records:

It is a simple file format for storing a sequence of binary records. This is done with the help of protocol buffers which is a cross language library for efficient serialisation of structured data. The tf.train.Example represents mapping and is done using TFX and thus records are created with .tfrecord extension.

3. Implementing R-CNN:

The region proposals produces a square which is feed into a CNN that produces 4096 dimentational feature vector as output and then CNN acts a feature execrator and the output layer consist of the features extracted from the image and is feed into SVM. Thus, we will use R-CNN for training our dataset.

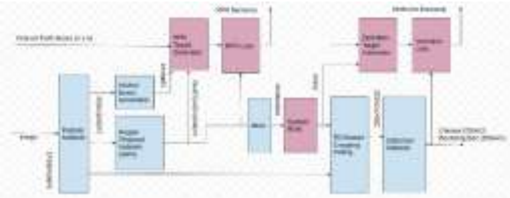

Fig 3: R-CNN Pipeline

4. Testing:

In this phase we will take different types of random images of objects. Images can be combined with different objects or of a single object. We will be taking the images and then tally with the dataset given and will identify what is the object along with it our system will also identify objects using the live feed irrespective of number of objects appearing in from of the camera.

5. Object Identification:

After training and testing the system for normal and edge cases our system would be able to identify objects using the image or live feed.

6. SYSTEM IMPLEMENTATION

A. *Module Description*:

1) *Image Collection*:

Input to the proposed system is Classification of Scan images of different objects are taken.

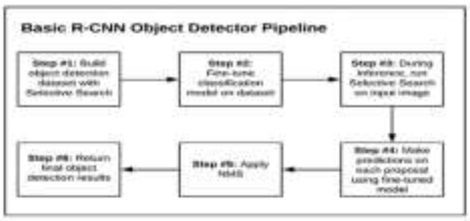

Fig 4: Basic Pipeline

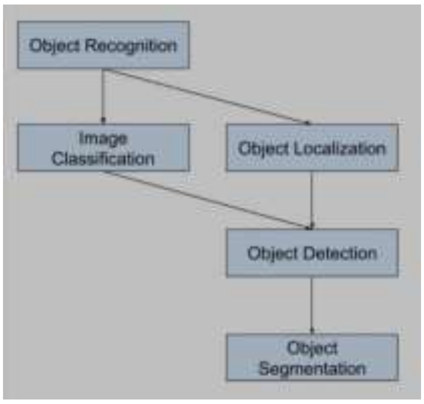

Fig 5: segmentation of object

2)*TernsorFlow Records:*

While working with the large dataset which we would be using in our system TFrecords compresses the dataset into a binary file which can be used during the training of our model. Binary takes-up less space on our disk thus, it makes our training much more efficient. It also stores the data in a time series or word encodings this is done with the help of tensorflow comprising of two components those being:

a)tf.train.Example

b)tf.train.SequenceExample

These are further used to serialize using TFRecordWriter

To save it in disk.

tf.train.Feature wraps a list of data of a specify type so that TensorFlow can understand it. It comprises of stored list which is stored in tf.train.BytesList.

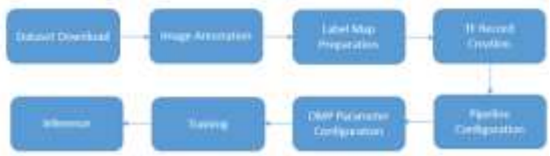

Fig 6: TensorFlow Record

3)*Testing:*

In this we use the R-CNN to differentiate the objects in an image. R-CNN is the convolution neural network based on the region proposal brought up in 2014 by Girshick who came up with the concept of region proposal for the first time. The principle of R-CNN is that it utilizes the region segmentation method of selective search to extract the region proposals in the image, which include the possible object candidates, and loads them into convolution neural network to extract the feature vectors. After merging by non-maximal suppression (NMS), the model outputs the precise object classifications and object bounding boxes to achieve object detection. The method of mapping region proposal of input image to the feature layer in Fast R-CNN shares the convolution computation, which substantially reduces the calculation. In addition, in order to decrease the parameters of full connection, Fast R-CNN adopts truncated SVD to enable that the single fully connected layer corresponding to weight matrix is replaced by two small fully connected layers, which further lessens the network calculation.

The proposed R-CNN model is comprised of three modules; they are:

- Module 1: Region Proposal. Generate and extract category independent region proposals, example candidate bounding boxes.
- Module 2: Feature Extractor. Extract feature from each candidate region, example using a deep convolution neural network.
- Module 3: Classifier. Classify features as one of the known class, example linear SVM classifier model.

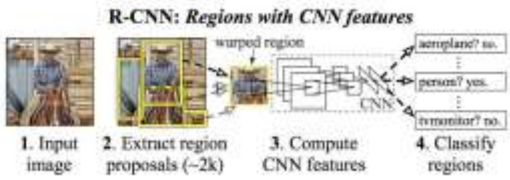

Fig 7: R-CNN Regions with CNN features

4) *Object Identification:*

After all the above steps the objects present in the image will be segmented or will be in a box stating the class of object if different than other. All the objects will be easily identified and will reduce the time for the recognition of the object in the image.

7. CHALLENGES AND SOLUTIONS

With the advancement of deep learning technology, the object recognition model based on deep learning has been continuously improved. Currently, object recognition is a better and more active perspective than object recognition in deep learning. Learning is heavily dependent on object recognition, so improving object recognition ability will promote improved object recognition ability. Now, various kinds of visual recognition and recognition competitions are being held to advance the further development of deep learning in object recognition and object recognition. Meanwhile, the ever-changing hardware upgrade is also driving deep learning technology applications forward. Based on the general analysis of current deep learning-based object recognition methods, it can be concluded that there are two main challenges, namely real-time and robustness of the computation. In the future, object recognition will undoubtedly develop in better real-time and robustness.

8. CONCLUSION

In previous paper, we discussed about a graph based Image Segmentation and object identification method, especially where the background is not chaotic and contains several same type of object. Our method successfully identified the search object from the chaotic background. However the method lacks its identification where the object seems multipart such as human body. It can identify the types of objects in a particular image or the input given in it. It can be useful when we have many kinds of object in a single picture. By differentiating objects with our own eyes or physically makes things difficult and time consuming. So we came up with this idea of making an object segregation project which will help and reduce the time and effort of a human and will provide with utmost accuracy. The object detection capability of R-CNN was demonstrated successfully. Our system shows that R-CNN has much more accuracy than the other architecture available but then we have a trade off in between speed and accuracy. Here in R-CNN we get higher accuracy with higher time detection. The bounding box regression incorporated within the R-CNN helps us to classify images with much more accuracy.

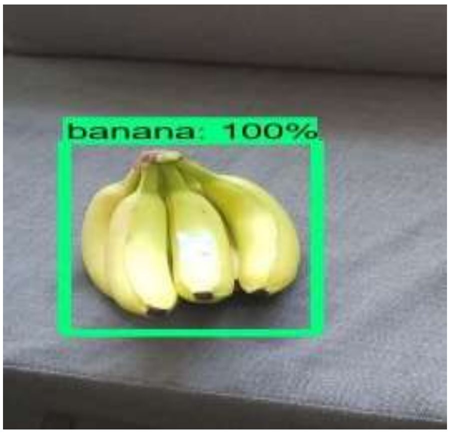

Fig 8: Objects identified in an image

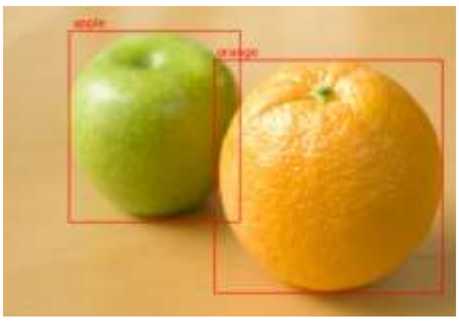

Fig 9: Multiple object detection in an image

9. RESULT

Our model was trained on NVIDIA GEFORCE GTX 1650 which took around 11 hours for training on 62789 steps. The below metrix was monitored using TersorBoard which is a visualisation tool for TensorFlow. The label for fig 10, were Total loss versus Time.

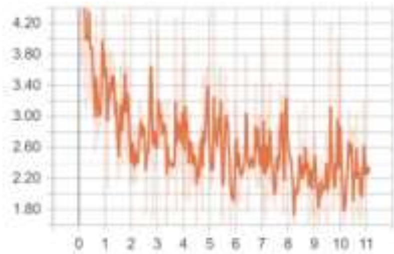

Fig 10: Total versus time (hours) using SSD with R-CNN

For fig 11, the labels for the graph is total number of steps used under training versus the loss.

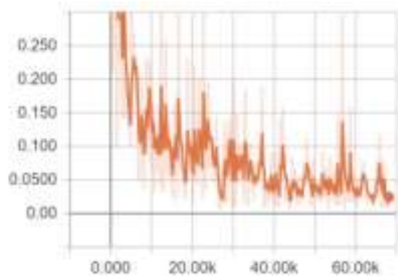

Fig 11: Total loss versus number of steps used in R-CNN

After training our model will be able to classify objects which was used during the training process. Thus, it would be able to detect objects via a static image or live using a camera. Our model's mean Average Precision (mAP) is 0.817 @IOU = 0.50:0.95.

10. REFRENCES

[1] Debalina Barik and Manik Mondal, Object Identification For Computer Vision using Image Segmentation in:201O 2nd International Conforence on Education Technology and Computer (ICETC).

[2] Cong Tang, Yunsong Feng, Xing Yang, Chao Zheng and Yuanpu Zhou, The Object Detection Based on Deep Learning in:2017 4th International Conference on Information Science and Control Engineering.

[3] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu and Xindong Wu, Object Detection with Deep Learning in: IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.

[4] Alexe, B., Deselaers, T., and Ferrari, "What is an object?," in ComputerVision and Pattern Recognition (CVPR) in:2010 IEEE Conference on (San Francisco,CA: IEEE), 73–80.

[5] Borenstein, E., Ullman, S.: Class-specific, top-down segmentation. In: ECCV (2). (2002).

[6] Levin, A.,Weiss, Y.: Learning to combine bottom-up and top-down segmentation. In: ECCV(2006).

[7] Srinivasan, P., Shi, J.: Bottom-up recognition and parsing of the human body. In: CVPR. AC(2007).

[8] FJ. Estrada, A.D. Jepson, and C. Chennubhotla. Spectral embedding and min-cut for image segmentation. In British Machine Vision Conference, 2004 ..

[9] An efficient algorithm for Co-segmentation Dorit S. Hochbaum Vikas Singh.

