Object Detection Applying Deep Neural Networks

ASMIT MISHRA (17BCE0886)
GUIDED BY: SURESH KUMAR N

Abstract

In today's world, with the evolution of the deep neural network the methodology adopted for implementing Object Detection mechanism has been taken to the next level. In our proposed mechanism we will be implementing different model versions of R-CNN data set model to train our network. This can be implemented by following the steps as follows: (a) We will collect the training data set which will consist of the daily recognizable object images. (b) Using the framework of Tenser flow we can train our model for object detection. (c) Applying different model we will compare the time complexity for the object detection. In our project we will implement the technique of object classification as well as object labeling at the same time in the same framework. This project concludes with the implementation of one of the enhanced technique of Object Detection methods compared to the one used in the past 3-4 years that means by implementing the concept of neural network.

Key Words: Neural Network, Object Detection, Data Model, Framework, Tenser Flow, Object Classification, R-CNN model (Region- Convolution Neural Network).

1. Introduction

Object classification is an extremely well known research bearing in the vision field. Propelled in 70's, object classification and detection is started to be on track until 90's when PCs turned out to be amazing and application ample. It is simple for us as human to classify different objects in pictures, in any case, things wind up troublesome for PCs. Including the distinctive stance of articles and the intricate condition around, object discovery is of greater uncertainty.
As we are well known, the advancements in algorithms is separated into two phases. Stage one depends on the customary highlights of the arrangement, and the second stage is the application of neural networks. Prior to 2013, a large portion of researchers were in try to optimize the traditional way of object detection. From that time onwards, both scholarly world and industry swung to the neural networks implementation.
With the expanding measure of location information, the conventional object classification strategy execution will end up immersed. The discovery execution will step by step enhance, yet the enhancement diminishes after a specific measure of information. Be that as it may, the technique of neural networks and deep learning is extraordinary promoted. While the information of the scene circulation get collected, the accuracy of the detection algorithm will go on increasing.
In this project, a lot of information of every day objects is gathered, and afterward unique neural network models are being made to train the detection algorithm. By comparing the result of the tradition method and the deep neural network, we can ensure the speed and accuracy of deep neural networks is more enhanced as compared to traditional algorithms.

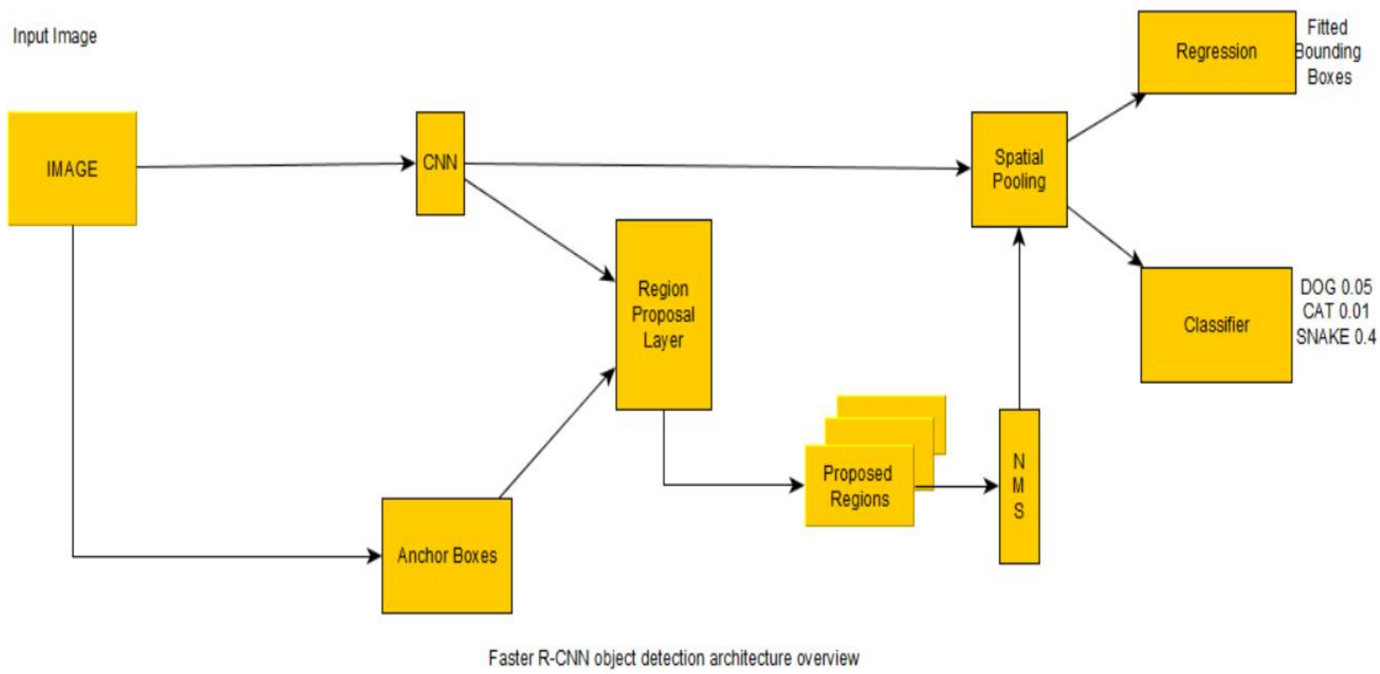
2. Background

Using ILSVRC, deep learning has increased the average object detection rate upto 44%. R-CNN which was proposed by Ross Girshick introduced CNN(Convolution Neural Network) method into object detection field for the first time. Previously, Selective Search window extraction algorithm was meant to replace the traditional sliding window extraction method. Now Object detection is also implemented in the field of automatic driving cars [1] using convolutional neural network named as SqueezeDet, which is implemented using small data-set so as to face the real world scenario without any delay. On of the Salient Object detection method [2] uses the power to compare similar looking objects with reference to their colors and size and shape. This method uses the power of 7 data-sets to implement the model and apply the object detection technique. [3] You Only Look Once is the another kind of algorithm developed in the hope to classify and the label the object in the same frame work. In this technique a bounding boxes are made so as to classify different objects present in the image. The processing speed of this algorithm is more than 40 frames per second.
Now a days there is a huge development in the technique of R-CNN model [4] . In this model various kind of datasets have been introduced which perform computation at various speeds. The faster R-CNN is the fastest to be known so far from the results. The convolution network is being trained before applying to the real world scenario. The another method which came into role play in the field of object detection is Hybrid object detection technique [5]. This method uses the method of recursive calling technique which forms the frame to classify the object in the continuous image . This method was fast but for large object detection there was a minor chance of encountering stack overflow during run time of the algorithm.But still at present world this technique of classifying objects outperforms many previously known techniques.

3. Problem Statement

Till now the techniques which we are using for the object detection is quite slow. The algorithm takes huge amount of time to train the network using the data set provided, because the data set usually contains thousands and sometimes millions of sample images. So in order to reduce the training time of the algorithm, in this project we will train the neural network using various kinds of data set models and compare the performance of the neural network approach and the previously followed approaches in terms of time taken by algorithms to get trained to satisfy the required correctness and generate the output more accurately.[1]
In the proposed work which has been implemented comprised of the small size model. The exact size or the number of objects used in the model is not fixed.With the help of SqueezeDet algorithm opnly small size models can be trained . So the time for the computation is variable. [2] The proposed model in this project deals with the object detection of objects with similar appearance and colors.So, If we put objects like remote and cellphone in one image then it cannot classify or differentiate between then. [3]
In YOLO detection method, accuracy is compromised if the background is complex. In case of blurred images, YOLO cannot perform image classification. [4] Dynamic Neural Network algorithms can be applied in future works to non-linear networked network. Understanding of complicated dynamics will help us in improving the filtering capacity effectively and efficiently.
[5] This model implements the technique of Recursive approach. Son in case of bigger objects more number of recursive computations are required to classify the object. In this method, due to more number of computations there will be a chance of Stack Overflow which imposes an impractical limit on size of the image. [6] Detection of an object includes two major parameters that means Color and Shape of the Object. In open-cv method when there is light illumination or reflection on the object surface, algorithm compromises with the accuracy and fails to give the result correct object type. Based on color approaches and shape based approach this method is weak. This method sometimes fail to detect the boundaries of various connected objects in an image.

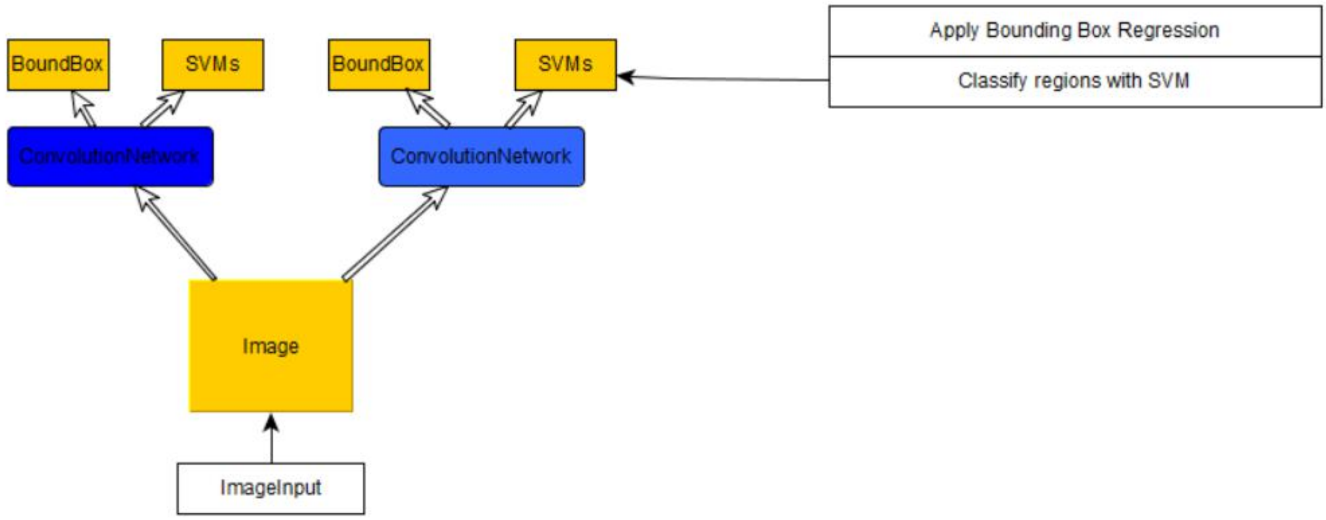
4. Proposed Architecture



Fig(1): Overview of the Image detection and Labeling using Fast R-CNN algorithm

Fig(1) shows the overview of Fast R-CNN architecture. An instance of the webcam is given as input along with the object proposal. Firstly, several convolution network processes the whole image and max pooling algorithm is applied to produce fully convolution feature map. After this, a feature vector is extracted for each of the proposed object. Above this layer, the R-CNN layer is applied which helps in image classification using its convolution network and label them with high accuracy.

4.1 R-CNN Algorithm:



Fig(2) : Overview of the Faster- RCNN working mechanism

The steps followed in the RCNN algorithm to classify the objects includes: Since we have to deal with the real time scenario, the input to our algorithm will be a snapshot of our live scene which will be taken from webcam. This input is feed in to the pre-trained region convolution neural network for retraining purpose. In the next step, the last layer of the network is trained depending on the number of objects to be detected. Algorithm then extracts the region of interest inn the image, then combine all the matching regions by reshaping them. Support vector machines are then trained to classify the objects. At the last step, the training of linear regression model is done to generate more accurate bounding boxes for all the objects that have been identified. Through the above process, many convolution inner layers can be implemented to improve the accuracy of the algorithm, but this may cost the computational power of the system used in the implementation.

4.2 Spatial Pooling Algorithm:

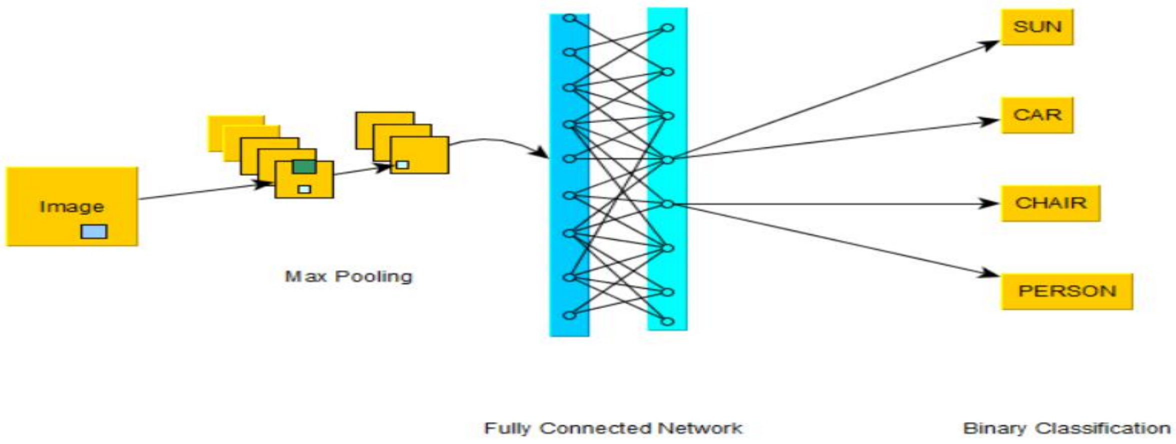
While dealing with real time scenario, there may be the chance of encountering same object with different height to width ratio. So it becomes quite challenging to classify the object. To ignore this problem, we implement spooling algorithm (max spooling). This algorithm ensure the r-cnn model that the object detection will work with no issues even if the object is deformed or the size of the object is changed. From the perspective of our algorithm spatial pooling divides the image into distinct sub regions and compute feature vector for them. Final image produced is by concatenation of all the sub regions.

4.3 Anchor boxes:

To classify different boxes, anchor boxes are generated around each of the distinct objects present in the same image. Rcnn algorithm firstly generates anonymously large number of boxes of various sizes. Then it calculate the probability of recognizing the object within a particular box. By comparing the neighbouring pixel values, the objects are differentiated. If full object falls within a particular box, tight anchor box are generated with the corresponding labels. So this algorithm is implemented as a model within the R-CNN algorithm for classifying and object labeling.

4.4 Classifier:

There are various available image classification algorithm present. In our implementation we have used supervised learning algorithm under which linear regression and support vector machine algorithm have been used. In simple words, this algorithm is based on the concept of Python Dictionaries. Various class of objects are included as the key along with their values. So after the anchor box is formed around the image, the object is compare with the components present in our training data set model. If matching occurs, then the corresponding value from the dictionary is displayed as the label in our output snapshot of the webcam. In this way , after the last layer of our convolution network, object labeling is performed.



Fig(3): Image Classification technique using neural networks

4.5 Region Proposal Layer:

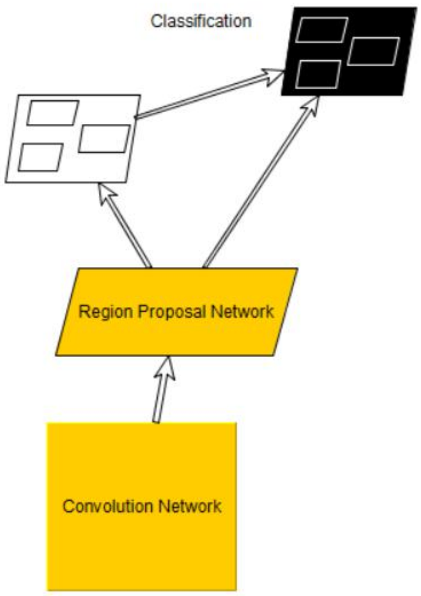
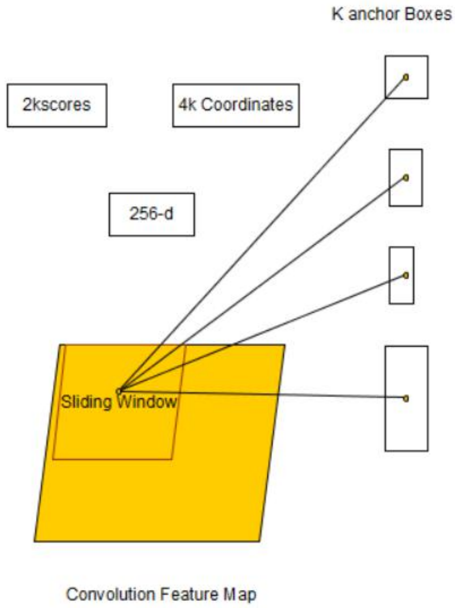


Fig (4) : Mechanism of RPN technique



fig(5) : Determining the number of anchor boxes.

RPN is used to generate the ‘Proposal’ for a particular regions when the object is known to exist. For this purpose a small network is implemented over the feature map (which is the output of the last layer of convolution network). Above this small layer the R-CNN model is implemented for further classification and labeling purpose. Both the classifier and the regressor is contained in the RPN technique. Classifier estimates the probability of a proposal(bounding box) bounding any object and the Regressor returns the coordinates of the region where the object has been detected.
For instance, let us consider the aspect ratio to be = width of image/ hight of image, and the developer selects the scale value to be 3 and 3 aspect ratio. So the total number of possible proposal will become 9 (k = 9). Hence , the number of anchor boxes generated for a particular object in the image would be 9. In this way, a large number of anchor boxes are generated around the proposals, and estimates the probability of detecting the object.

5. Pseudo-Code/Algorithm

1.BEGIN
Import necessary modules
Model_name = ‘’
Model_file = modelname + ‘ .tar.gz’

2.#Set name of training model
.set_model_path =   MODEL_NAME + '/frozen_inference_graph.pb'

3.# List of the strings that is used to add correct label for each box.
Path_labels = os.path.join('data', 'mscoco_label_map.pbtxt')

4.#downloading model
```
if not path.exists(MODEL_NAME + '/frozen_inference_graph.pb'):
        print ('Downloading the model')
        opener = urllib.request.URLopener()
        opener.retrieve(DOWNLOAD_BASE + MODEL_FILE, MODEL_FILE)
        tar_file = tarfile.open(MODEL_FILE)
```

5.#‘Download complete’
6.#extracting the downloaded model
```
  if 'frozen_inference_graph.pb' in file_name:
                tar_file.extract(file, os.getcwd())
```

7.#extraction complete
8.#load tensorflow model and labels into the memory
```
detection_graph = tf.Graph()
with detection_graph.as_default():
    od_graph_def = tf.GraphDef()
    with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
        serialized_graph = fid.read()
        od_graph_def.ParseFromString(serialized_graph)
        tf.import_graph_def(od_graph_def, name='')
Label_map = load_labelMap()
```

9.# import open-cv packages(cv2) for accessing the systems webcam service
```
cap = cv2.VideoCapture(0)
  with tf.Session(graph=detection_graph) as sess:
      ret = True
```

10.# accessing the scene through the webcam
11.#setting the dimension of the image axis
```
  image_np_expanded = np.expand_dims(image_np, axis=0)
```

12.# image accessing
```
  image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
```

13.#generating boxes
```
  boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
```

14.#calculate the probability of correctness of object detection and classify them
```
  scores = detection_graph.get_tensor_by_name('detection_scores:0')
  classes = detection_graph.get_tensor_by_name('detection_classes:0')
```

15.#displaying the output of the above steps in the system display monitor
Image = Visualization_array(image,boxes,scores,classes)
cv2.imshow(‘image’)
16.END

6. Results and Discussions

6.1 Data set

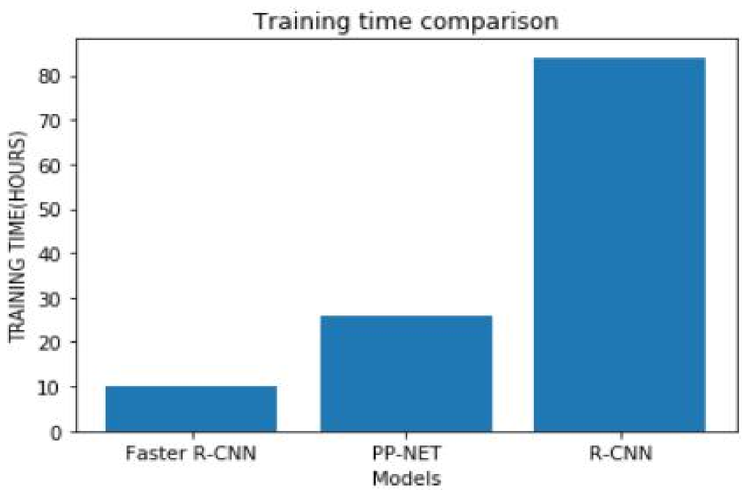
With the increase in the implementation of the deep neural networks in the field of object detection, there are huge number of good datasets available. Some of them are COCO data-set, Oxford-IIIT pets data set, PASCAL VOC data set, etc. In this project we implemented the COCO data-set. This data-set contains images of objects of 91 categories including chair, person, animal, daily basis usage things etc.

6.2 Experimental Environment

The implementation of this project is done on the Tensor flow frame work. Tensor flow is the open source deep learning library supported and developed by Google. The frame work of the tensor flow is quite easy and fluent. The tensor flow API can be used on various other platforms which makes it highly compatible for the machine learning model, this feature of tensor flow made implementation of this project successful.

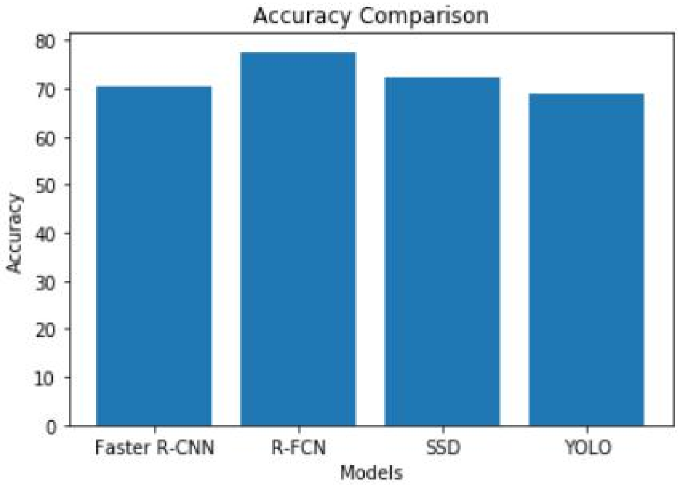
6.3 Experiment

By implementing various training models to our data-set we came with the following statistical result.



Fig(6): Training time comparison of various models.

By looking in graph in figure(6), it can be seen clearly that the training time of the Faster R-CNN model is very less as compared to other models. Training time of our model can make a huge difference in computational performance of our algorithm when it has to be implemented in the real time scenario, where new objects will keep on adding to our data-sets. So we have to train our model more than once to maintain the accuracy and performance of the model.



Fig(7) : Comparing accuracy of different models.

From the above analysis, it can clearly be seen while we gain the best training time output for the Faster R-CNN model in the cost of accuracy. But there is not much difference between the accuracy between diffenrent models. The performance of a particular model is not depend only on the one factor, there are numerous factors such as training time of the kodel, testing time of the model, the accuracy level, frames per second on which iur model can capture the image and process on it etx. So, it can bve said that on the basis of the accuracy level, R-CNN model is good, because it has the advantage over other models in terms of the training time Fig(6) and testing time.

Conclusion

This project mainly established to implement an efficient object detection system on the COCO data set. The data set was trained on different model and compared the performance and finally we came to a result getting a good daily object detection system.There well trained model can be deployed in our daily gadgets such as smart phones or computers for the accurate object detection. In the future, we can improve the performance of our model by adding more images to our data seta and tuning the parameters (iff any) to make our model give more accurate results.

Future works

This project works well for the real time based object detection scenario very well. In some cases it still lags in the accuracy as compared to the other models. So to improve on its scalability and accuracy, more number ob classified objects can be added to the data set. Upon increasing the number of layers in the neural network, we can achieve higher level of accuracy.

References

[1] Bichen Wu1 , Forrest Iandola1 , Peter H. Jin1 , Kurt Keutzer et al. SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving, IEEE conference of pattern recognition.

[2] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, Jia Li et al. Salient Object Detetection : A BenchMark . IEEE conference of pattern recognition, and computer vision.

[3] Joseph Redmon, Santosh kumar Diwala, Ross B. Girshik, Ali Farhadi.,”You only look once: Unified, Reatl-Time Object Detection”, IEEE conference Computer Vision and Pattern.

[4] Alex Dominguez-Sanchez, Miguel Cazorla , Sergio Orts-Escolano. A New Dataset and Performance Evaluation of a Region-Based CNN for Urban Object Detection. IJCNN-2018: 1-8

[5] J. Martin Herrero et al. Hybrid Object Labelling in Digital Images. Springer: Machine version and Application.

[6] Juan Wu1, Bo Peng , Zhenxiang Huang , Jietao Xie. Et al. Computer Vision Based object Detection and Classification . AICT-392, pp.183-188, 2013

[6] Ross Girshick. FAST R-CNN. Computer Vision and pattern recognition

[7] Suresh Kumar.N, Shantharajah.S.P., Senthil Murugan.B., X.Z.Gao, "Satellite Image Processing for Biomass Estimation", Journal of Global Pharma Technology(JGPT).

[8] Ross B. GirshickJeff DonahueTrevor DarrellJitendra Malik, "Rich Feature hierarchy for Accurate Object Detection using Convolution Networks", IEEE Conference On computer Vision and pattern recognition 2014.

[9] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. "ImageNet Classification with Deep Convolution Neural Networks". Semantic Scholar 2017.

[10] Xiaotong ZhaoWei LiYifan ZhangThomas Aaron GulliverShuo ChangZhiyong Feng,"Faster R-CNN for Pedestrian Detection System", 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)2016.

[11] Xiangxin ZhuCarl VondrickDeva RamananCharless C. Fowlkes. "Do we need More Training Data or Better Models for Object Detection", BMVC 2012.