Object Detection Using Deep Learning – SSD and Faster R-CNN

Rani Chintha, Sai Sumanth Boda, Yash Bardapurkar

Department of Computer Science
The University of Texas at Arlington, Texas, USA
rani.chintha@mavs.uta.edu; saisumanth.boda@mavs.uta.edu; yash.bardapurkar@mavs.uta.edu

Abstract

Object Detection is an Integral Part of Computer Vision used to identify Objects in a Photo or a Video. Object Detection has grabbed attention in Research and Development Fields as it is Closely related to Video and Image Analysis. In Current Technologies, Most of the Object Detection Models are based on Old Technologies and Architectures which cannot be trained to Full Extend. These are Not Reliable for Upcoming Challenges as they tend to Deteriorate its Performance when going for Complicated Scenarios. With Rapidly growing World, these Problems are addressed by the New Object Representation and Machine Learning Models which are Capable of Performing High level Semantics, Features and Strategies. In this Project We will be Performing Object Detection Using SSD (Single Shot Multi – Box Detector) and Faster – RCNN and also to Calculate the Accuracy, Recall and F1 Scores for the Two Models. It is Designed to Detect Instances of Categories such as Humans, Animals, and Vehicles. Also, we Use Classification Techniques to predict the Image within the Bounding Box. We Use SSD and Faster R-CNN as Framework that Lets us use Pre-Trained Detection Models and Create New Trained Models. We Use these Models which contain Trainable Detection models and its Frozen Weights to construct and Train Models.

1. Introduction

Object Detection is the process to Detect Objects in an Image or a Video in Real World. In the " Object Detection Algorithm", We try to draw a Bounding Box around the Object of Interest to locate it in the Image. There may be many Bounding boxes representing different Objects in an Image, and you may not know how many in Advance. Object Detection has many Applications in Robotics, Vehicle Detection, Surveillance, Face Recognition, Security, and many other Major Fields. Object Detection is based on Image Classification which Associates Class Labels to Objects within the Image. Moreover, it includes Drawing the Rectangular Boundary Boxes around the Object after Detecting it. The steps which Needs to be followed in order to Achieve Object Detections are Image Classification, Object Localization, Detection. The Dataset will be further used to train two Model Architectures that take Images as input and Detect the most Likely Objects in the Output. The Output of the given Input will be a Confidence Score, which depends on how certain the Algorithm is about the Object Category in the Bounding Box.

1. 1 Modern Convolutional Object Detectors

1. Object Recognition

Image Classification can be defined as detecting the Class to which an Object within an Image belongs to. In other Words, it can be classified as a Prediction of a class of an Object and giving a Class Label as an Output from a given input Image. It further involves Locating the Object with Class Label and Map to the specific object Using Bounding boxes. Bounding Boxes is a Image annotation method where we use them to Outline a Object in to a Box. The Output may be of Object Detection with Multiple Bounding boxes detecting Various Objects within the Image.

2. SSD (Single Shot Multi – Box Detector)

The SSD Object Detection Uses **VGG16** to Extract feature maps and detects Object by applying small Convolution **Conv4** 3 Layer filters. SSD has a better coordination between Speed and Accuracy. SSD does not use a delegated region Proposal Network. Instead, it resolves to a Quite Simple method. It computes both the Location and Class scores using **Small Convolution Filters.** After Extracting the Feature Maps, SSD Applies 3 x 3 Convolution Filters for Each cell to make Predictions. Also for Training, the Data needs to be assigned to the Specific Outputs of Detectors. It is Faster Compared to Faster R - CNN but less Accurate due to Optimization of Images.

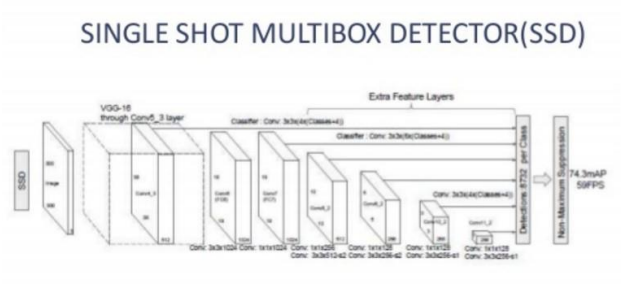


Fig: SSD

3. Faster R – CNN

Fast R – CNN Use the SPP – Net and Ideas of R – CNN to make Object Detections. Firstly, it processes the Entire image with Several Convolutional and pooling Layers to form into a Convolutional Feature Map. It then Uses Calculations of Back – Propagation and then has a Gradients pumping in from Several Regions. It has two Network Heads, they are Bounding Box Regression and Classification Head. Moreover, It is 9 times Faster than the R – CNN .

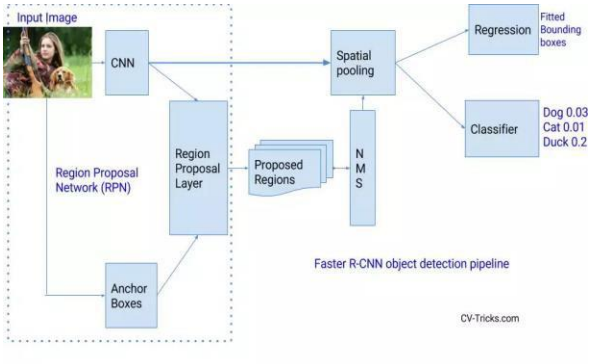


Fig: Faster R – CNN

4. Fast R – CNN

Fast R – CNN Use the SPP – Net and Ideas of R – CNN to make Object Detections. Firstly, it processes the entire image with Several Convolutional and pooling Layers to form into a convolutional feature map. It then uses calculations of Back – Propagation and then has a Gradients pumping in from Several Regions. It has two Network Heads, they are Bounding Box Regression and Classification Head. Moreover, It is 9 times faster than the R – CNN.

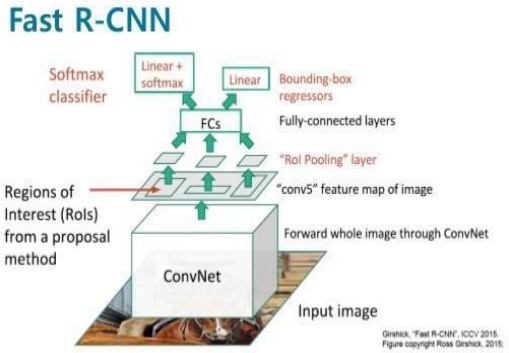


Fig: Fast R-CNN

2. Dataset Description

2.1 Data Description:

COCO stands for Common Objects in Context. The Images in COCO Dataset are taken from day – to day scenario by attaching a "Context" to the Images captured. The COCO Dataset is a format in JSON and has Collection of Information, Licenses, Images, Annotations, Categories and Segment Info. In order to Detect an Object in Image, Firstly we Inspect the Entire Image and We make Boundary Boxes for all the Objects present in the Image. Then We match these Identified Objects in Boundary boxes with that of the Preexisting Trained Model. Here this Trained Model Uses the Supplementary Images provided by the COCO Dataset that Already Captured Surroundings. We Use these Trained models to match the Objects with the Most Accurate Class Available within Dataset and also provide its Accuracy.

"Info" : […],
"Licenses": […],
"Images":[….],
"Annotations" :[….],
"Categories": [….],
"Segment Info": [….]

2.2 Data Pre-Processing:

Data Preprocessing refers to making changes to the dataset according to the needs of the algorithm and then providing it. The Data in the Real World is Raw and Noisy, which may affect the results of the Algorithm. For the COCO Dataset, since it is an Effective Set, it does not need much Enhancement. The Changes we made to the Dataset is that we added a Structured API called a Data Frame and loaded it into the Different Object Classes that exist in the

Dataset to simplify Data Representation. In Addition, the size of the COCO Dataset is Very Large, so we have included specific Instructions in the Code to Download the Dataset we use in the Algorithm. When starting to Implement to Implement the Object Detection Model, we may need to do Further Preprocessing. All the Steps involved with Data Preprocessing are:

- Step – 1: Install TensorFlow
- Step – 2: Install Other Dependencies
  tf_slim
  TensorFlow hub
  pycocotools
- Step–3:Download TensorFlow models repository from https://github.com/tensorflow/models and install the research package.
- Step – 4: Load Pre-Trained Object Detection models on COCO Dataset using the Models Repository (SSD) and TensorFlow-hub (Faster R - CNN)
- Step – 5: Download and Extract the MS-COCO val2014 annotations Dataset.

3 Project Description

3.1 Description:

Model Presentation:

Object Recognition refers to a collection of related tasks for Recognizing Objects in Digital Photos and Object Detection Locate the presence of Objects with a Bounding Box and types or Classes of the Located Objects in an Image. Region – based Convolutional Neural Network (R – CNN) is a Series of Technologies used to solve Object Localization and Recognition Tasks, aiming to Improve Model Performance.'

Loading Label Map:

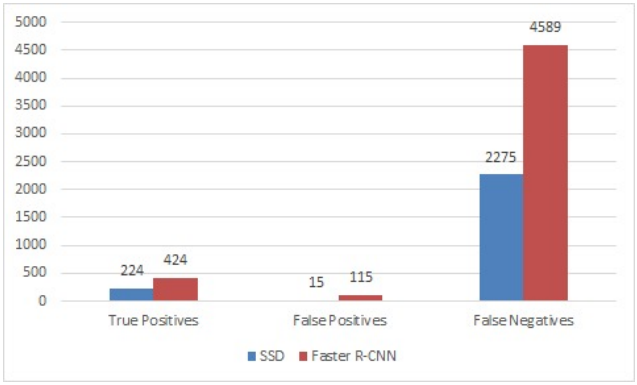
The Label Maps the Index to the Category name, so when our Convolutional Network Predicts 5, We know it Corresponds to an Airplane. Here we use the Internal Utility Function, but any method that returns a Dictionary Mapping Integers to Appropriate string Labels can be Used.

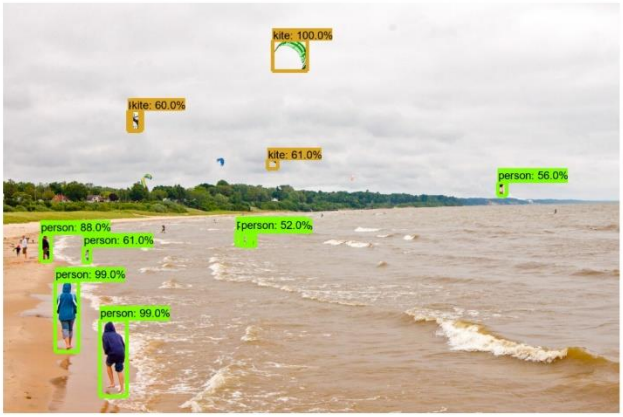
Training the Model:

We have loaded the SSD (Single Shot Multi Box Detector) and Faster R – CNN Object Detection Models to Run on the Test images and Achieved the Predicted Results.

Validating the Model:

The Algorithm was validated by Training the model to other COCO Dataset from 2014 (MXS COCO val2014 dataset (6GB) and MS COCO 2014 val2014 Dataset(240MB)). Counted the True Positives, False Positives and False Negatives from the Objects Detected by the model. The Object Detection Model was Performed for the First 100 images in the val2014 Dataset and compare the Output with the Actual Objects detected in the Images. Precision, Recall, and F1 Score was Calculated from above True Positives, False Positives and False Negatives.





	detection_class_id	detection_classes	detection_scores	detection_boxes
0	38	kite	0.995096	[0.09167153388261795, 0.43695685267448425, 0.1...
1	1	person	0.994780	[0.7734400033950806, 0.15674878656864166, 0.95...
2	1	person	0.986934	[0.6791952252388, 0.0842192992568016, 0.849237...
3	38	kite	0.946706	[0.2624603807926178, 0.20228949189186096, 0.31...
4	1	person	0.883560	[0.5652558207511902, 0.0604993961751461, 0.629...
5	1	person	0.633654	[0.555866181850433, 0.377827435731887 8, 0.589...
6	38	kite	0.612053	[0.38306012749671936, 0.4284818172454834, 0.40...
7	1	person	0.605845	[0.5960627794265747, 0.1299326866865158, 0.633...
8	38	kite	0.601781	[0.2618210017681122, 0.20729342103004456, 0.31...
9	1	person	0.576214	[0.5614494681358337, 0.3880852460861206, 0.593...
10	1	person	0.559211	[0.43814826011657715, 0.801978588104248, 0.468...
11	1	person	0.516004	[0.5591207146644592, 0.38508424162864685, 0.59...

Fig: Detection of an Objects

3.2 Main References Used for the Project

The Main Reference we used to Understand and Implement the Model is “Faster R – CNN Features for Instance Search” by Amaia Salvador, Xavier Giro – I – Neito, Ferran Marqu’es and Shin’ichi Satoh and “SSD Object Detection : Single Shot MultiBox Detector for Real Time Processing ” by Jonathan Hui

Faster R – CNN takes Advantage of the Object Proposal Learned by a Region Proposal Network (RPN) and their Associated CNN Features to Build an Instance search Pipeline composed of a First Filtering Stage followed by a Spatial Reranking. However, his whole process runs at 7 frames per second – far below what a real-time Processing Needs.

SSD is designed for Object Detection in Real-time. SSD speeds up the process by eliminating the need of the Region Proposal Network. To recover the Drop in Accuracy, SSD applies a few Improvements including Multi-scale features and Default Boxes. These Improvements allow SSD to match the Faster R – CNN’s accuracy using Lower Resolution Images, which Further pushes the Speed Higher.

3.3 Difference in APPROACH/METHOD between project and the main projects of the references

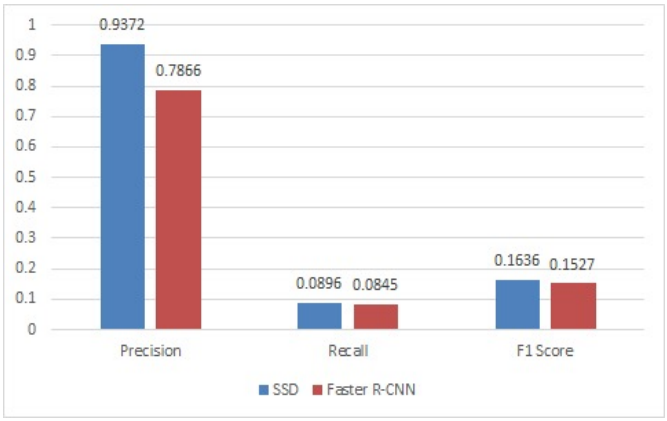
In all the References and Papers we read, they Made the Object Detection only using the Bounding boxes and then Detect Object within it. But Our Idea and Approach was a Different One. In this Paper and Project, We Implemented Object Detection Using Class Labels and Bounding Boxes. Also We took True Positive, False Positive, and False Negative Values into Consideration. Then we Used these Values to Measure Precision, Recall and F1 Scores. This was the Idea we Implemented on our Own which is not in any of the References we Used.

3.4 Difference in ACCURACY or PERFORMANCE between project and the Main Projects of the References

Based on Comparison between Our Project and the Project of Reference, We were able to achieve an Increase in Accuracy and Performance by an Average of 5%. Moreover we did Measure Precision, Accuracy and F1 Scores for both the Models.

```
In [26]: %%time
         print('SSD: ')
         evaluate(detection_model, images, ann)

SSD:
True Positives: 224
False Positives: 15
False Negative: 2275

precision = 0.9372384937238494
recall = 0.0896358543417367
f1 score = 0.16362308254200147
Wall time: 6.06 s
```

```
In [27]: %%time
         print('Faster R-CNN')
         evaluate(faster_rcnn_model, images, ann)

Faster R-CNN
True Positives: 424
False Positives: 115
False Negative: 4589

precision = 0.7866419294990723
recall = 0.08458009176142031
f1 score = 0.1527377521613833
Wall time: 1min 48s
```



List of your contributions in the project

1. Pre-Processing the COCO Dataset
2. Trained and Implemented the Object Detection Model using SSD and Faster R-CNN.
3. Enhancement to the model by Implementing Object Detection using SSD and Faster R-CNN.
4. Calculating Precision, Recall and F1 Scores using True Positive, False Positive, False Negative.

4 ANALYSIS

4.1 What did I do well?

Object Detection is Used to Detect the Objects in Images which helps us in Various Sectors such as Security, Identifying People, Building Self – Driving Cars etc. In our Project we have Evaluated SSD and Faster R – CNN Model which Perform Object Detection. We have used TensorFlow which has pre trained Models which helps us Classify Objects, and the MS – COCO Dataset to evaluate Models. During Evaluation We compare the Output of the Models for an Image in the Validation

Dataset to its Expected Output. We Compare the Class ID's of the Images in the Objects that are Detected.

4.2 Challenges Faced

The Result of Our Algorithm is a List of Objects and its Attributes, such as Class ID, Object Name, Bounding Boxes etc. The Bounding Box values in the Results of the Model are on scaled of the Range of 0 to 1, While the Bounding Box Values in the Validation Dataset are Scaled to the Dimensions of the Validation Image. This made Comparing the Bounding Boxes of the Results to the Bounding Boxes of Expected Objects Very Difficult.

4.3 What Could I have Done Better?

While Evaluating the Models, we compare the List of Object detected by the Model to the List of Expected Objects in the Image as per the Validation Dataset. In our Methodology, we compare the Class ID of the Objects in the List to determine which of the Objects from the List of Expected Objects are Detected in the Image. This Methodology of Evaluation could be made better by using the Bounding Boxes of the Expected Images for Comparison with the result of our Algorithms instead of Class ID, for a better Match of Objects.

4.4 What is Left for Future Work?

We can make the Evaluation Method more Accurate by Scaling Down the Bounding Boxes of the Expect Objects in the Validation Dataset in the Range 0 to 1. So that Comparison with the Model Output would be Easier.

5 Conclusion

In this Paper, We Proposed Object Detection using SSD and Faster R – CNN which is a Regression Technique. SSD makes more Predictions and has a Better Coverage on Location, Scale and Aspect Ratios. With the Precision, Recall and F1 Scores Calculated, SSD can Lower the Input Image Resolution to 300 x 300 with a Comparative Accuracy Performance. By Removing the Delegated Region Proposal and Using Lower Resolution Images, The Model can run at real – time speed and Still beats the Accuracy of the State – of – the – art Faster R – CNN. SSD wall time is 6.06s while Faster R – CNN wall time is 1 min 48 s.

References

1. http://cs231n.stanford.edu/slides/2018/cs231n_2018_ds06.pdf
2. "Faster R-CNN Features for Instance Search" https://www.cv-foundation.org//openaccess/content_cvpr_2016_workshops/w12/papers/Salvador_Faster_R-CNN_Features_CVPR_2016_paper.pdf
3. "An Implementation of Faster RCNN with Study for Region Sampling https://arxiv.org/abs/1702.02138
4. "Revisiting RCNN: On Awakening the Classification Power of Faster RCNN" https://openaccess.thecvf.com/content_ECCV_2018/html/Bowen_Cheng_Revisiting_RCNN_On_ECCV_2018_paper.html
5. "SSD: Single Shot MultiBox Detector" https://link.springer.com/chapter/10.1007/978-3-319-46448-0_2
6. "OBJECT DETECTION AND IDENTIFICATION A Project Report" https://www.researchgate.net/publication/337464355_OBJECT_DETECTION_AND_IDENTIFICATION_A_Project_Report
7. Yali Amit, and Pedro Felzenszwalb. Object Detection.
8. Narendra Ahuja and Sinisa Todorovic. Learning the Taxonomy and models of categories present in arbitrary images. In International conference on Computer Vision, 2007.
9. Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep Neural Networks for Object Detection.