PROJECT REPORT



OBJECT DETECTION WITH SINGLE SHOT MULTI BOX DETECTOR

Arbuda Sivani Majeti

811181382

[amajeti@kent.edu](mailto:amajeti@kent.edu)

**I-Summary:**

Object detection is a computer vision technique for locating and identifying objects in an image or video. For example, if an image contains a dog, a cat and a car, object detection helps us to classify the types of things in the image and to locate them within the image. Object detection can be done with several Neural Network techniques like R-CNN, Fast R-CNN, Faster R-CNN. YOLO and SSD were introduced to address the shortcomings of R-CNN and its successors. Hence, in my project I have made a comparison on why Object detection using SSD is better than YOLO. How and why to use Caffe model with pre-trained input classes. All the justifications were explained very well in detail in my project. I have not used any technical implementation of YOLO but implemented SSD and described why it is preferrable.

**II-Problem Statement:**

**Caffe:**

Caffe is a deep learning framework developedas a faster and far more efficient alternative to other frameworks to perform object detection. Caffe can process ***60 million images per day*** with a single NVIDIA K-40 GPU. That is 1 ms/image for inference and 4 ms/image for learning. [Caffe](http://caffe.berkeleyvision.org/) is an open-source deep learning framework allows you to leverage your GPU for training neural networks.

**Data Collection:**

I have not taken any dataset but used the images from the internet with the pre-determined classes. Caffe has helped me to get the pre-trained model with the classes that I would be needing for building my model. I have used the following classes, and this has been derived from the pre-trained caffe model.

Text, letter

Description automatically generated

My model will be able to recognize all the objects mentioned above and if at all I would like to have new objects, I should train the caffe model that already exists. After the training, the model will be able to identify the new objects as well. It will not be able to detect the image if anything that is beyond the trained classes.

**III. Techniques:**

The main objectives of my project are:

* To perform Object Detection using Single Shot mutlibox Detector
* Architectural differences between YOLO and SSD
* To understand the advantages of using a pre-trained model like Caffe for object detection.
* How is SSD Different

*A. Object detection using SSD:*

* Object detection is detecting, predicting, and localizing an area of an image.
* SSD Object Detection extracts feature maps using a ***base deep learning network,***which are CNN based classifiers, and applies ***convolution filters*** to finally detect objects.
* SSD starts with a VGG-16 architecture but with the fully connected networks removed. Some extra convolutional layers are attached for handling bigger objects.
* The SSD model is based on feed-forward convolutional network. It produces fixed-sized bounding boxes and scores if an object class instance is present in those boxes. During final predictions, a non-maxima suppression algorithm is used.
* SSD consists of two parts - a **backbone** model and **SSD head**. Backbone is a standard pre-trained image classification architecture which acts as a feature extractor. SSD head is an additional layer on top of the backbone model.
* Object detection in SSD takes place with the following features:
  + Multi-scale feature maps - Convolutional layers are added to the end of the base network. SSD uses lower resolution layers for detecting larger objects.
  + Convolutional predictors - After going through certain convolutions a feature size of m x n is obtained with p channels. A 3 x 3 convolution is applied on this m x n x p layer.
  + Default boxes and aspect ratios - There are k bounding boxes of different size and aspect ratios for each location. For each box c class scores and 4 offsets relative to the original default bounding box shape are computed. Thus we get (c+4)kmn outputs for a feature map of m x n

Python implementation for building the model:

* Libraries: Following libraries have been used to for the model:
  + Numpy
  + Argparse
  + Open CV2 - Open CV is used for image processing and computer vision tasks
* Caffe: In my model, a pretrained caffe model with input classes is used for detecting the images. Initialized labels for the SSD to generate bounding box for the trained images and colors for each class

Text

Description automatically generated

* Constructed an input blob of the image and normalized the image.

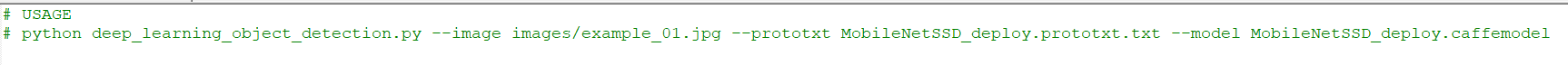
A picture containing application

Description automatically generated

* Filtered out the weak detections by ensuring that the confidence is greater than a minimum confidence and only images with strong index ID were stored. Extracted the index of the class label from the `detections’ and then computed the (x, y)-coordinates of the bounding box for the object.

Text

Description automatically generated

* In the python interface, a comment has been added to detect the input image and to check if the model is able to take the input classes given and detect the image. 
  + images/example\_01.jpg \ - this is the route path for the input image give. In my model, input images are from example\_01 to example 07. Here, the example\_01 can be changed to any images from 01 to 07.
  + prototxt MobileNetSSD\_deploy.prototxt.txt – prototxt files are **model definition files that are required when you train a CNN model**. The snapshot. caffemodel and mean. binaryproto files are output files that are created after the model training is complete.
  + MobileNetSSD\_deploy.caffemodel – path to the caffe model.
* Output:

A couple of cars on a road

Description automatically generated with low confidence

I have taken example\_01 which is the image of the cars, and the model has accurately detected the cars. I tried several other objects that are mentioned in the above classes and the model was able to detect them accurately.

*B. Architectural differences between YOLO and SSD*

**YOLO Vs SSD:**

* YOLO (You Only Look Once) is an object detection algorithm where the image is looked at only once to sight multiple objects. In YOLO every image is divided into a grid of size s x s and for every grid N bounding boxes and confidence are predicted. Confidence gives accuracy of the bounding box and tells whether any object is present in the bounding box. So, a total of s x s x N boxes are predicted.
* SSD looks at the image only once and develops a multiscale feature map which helps in detection of objects at different scales. SSD adds custom convolution layers at the end and convolution filters to make predictions. SSD also uses anchor boxes which are a collection of boxes at different spatial locations and aspect ratios. During training appropriate anchor boxes are matched with bounding boxes of each ground truth. Any box which has a Jaccard Index greater than 0.5 is considered as a match.

Diagram

Description automatically generated

* In SSD 8732 boxes are developed in comparison to 98 boxes in YOLO. As a result, SSD outperforms YOLO in terms of accuracy but there is an impact on speed.
* YOLO **struggles to detect and segregate small objects in images that appear in groups**, as each grid is constrained to detect only a single object. Small objects that naturally come in groups, such as a line of ants, are therefore hard for YOLO to detect and localize.
* SSD makes more predictions and has better coverage on location, scale, and aspect ratios. With the improvements above, SSD can lower the input image resolution to 300 × 300 with a comparative accuracy performance.
* SSD a good balance between speed and accuracy. SSD runs a convolutional network on input image only once and calculates a feature map.

*C. To understand the advantages of using a pre-trained model like Caffe for object detection.*

Caffe Model:

* Caffe is a deep learning framework developedas a faster and far more efficient alternative to other frameworks to perform object detection. Caffe can process ***60 million images per day*** with a single NVIDIA K-40 GPU. That is 1 ms/image for inference and 4 ms/image for learning.
* [Caffe](http://caffe.berkeleyvision.org/) is an open-source deep learning framework allows you to leverage your GPU for training neural networks.
* Caffe internally works with a data structure called [blobs](http://caffe.berkeleyvision.org/tutorial/net_layer_blob.html) which is used to pass data forward and gradients backward. It’s a four-dimensional array whose four dimensions are referred to as:
  + N or batch\_size
  + channels
  + height
  + width
* For my project, I have used a pre trained caffe model which has pre-defined classes that are given to the model like “aero plane”, “sky” etc. In my model, I should give the input of images with such classes and then the model will be able to detect. If we give any other image which does not have any of these classes, then the model will not be able to detect the object.

*D. How is SSD Different:*

* Other algorithms like R-CNN and Fast(er) R-CNN use object proposal methodology where the image is first broken down into different parts and suggests where the object could be potentially located. This leads to longer time for training and compromises with accuracy. YOLO and SSD were introduced to address the shortcomings of R-CNN and its successors. They speed up the process by removing the concept of object proposal method.
* SSD predicts category scores and box offsets for fixed default bounding boxes using filters that are applied to feature maps. To achieve high accuracy, different scale predictions are produced from the feature maps which are then separated by aspect ratio. As a result, even on images with low resolution, high accuracy is achieved.

Diagram

Description automatically generated

Table

Description automatically generated

**IV. Conclusions:**

* After working on constructing a model using Object Detection with SSD, I can conclude that Single Shot multibox detector is better when we had to use millions of objects and it is great when we have small objects in the images. The SSD and Caffe for object detection, my model was able to accurately predict an image up to 99%.
* While comparing the architectures of YOLO and SSD, understood that SSD is more efficient than YOLO.
* Caffe model’s expressive architecture, extensible code, speed has made it more preferrable for training the images.
* Finally, I can conclude that choosing the proper method is important and largely depends on the problem and the outcome we are looking for.

**V. Contributions:**

* In the class, we saw object detection with the basic Neural Network and in my project, I have used Single Shot Multibox Detector (Advanced and much Faster Neural Network) for object detection.
* In my learning on the topic, I have learnt about Caffe which is a pre trained model with input classes. This is very different and effective and can take a large amount of data for training. Though my project didn’t require usage of Caffe, I tried to implement this in my model to train my model with pre-defined classes.
* While working on my project, I was able to learn about much faster Object Detection algorithms followed by an advanced training model for millions of images.

**VI. References:**

* <https://medium.com/acm-juit/ssd-object-detection-in-real-time-deep-learning-and-caffe-f41e40eea968>
* [Object Detection by Image Processing using Deep Learning and Caffe with Python and OpenCV](https://www.youtube.com/watch?v=Il_S2cf7x_0)
* <https://www.mathworks.com/discovery/object-detection.html>
* <https://towardsdatascience.com/ssd-single-shot-detector-for-object-detection-using-multibox-1818603644ca>
* <https://www.joyofdata.de/blog/neural-networks-with-caffe-on-the-gpu/>
* <https://cv-tricks.com/object-detection/faster-r-cnn-yolo-ssd/>
* <https://iq.opengenus.org/ssd-model-architecture/>