**CSE 6363-003 Machine Learning**

**Term Project final report**

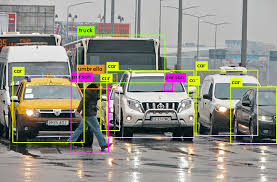
Topic: Object Detection using YOLOv3 algorithm

Group members:

1. Meetkumar Patel- 1001750000
2. Yash Dani –

**Abstract:**

During this high technological era, under artificial intelligence and machine learning magnificent researches are conducted. Robotics is one of the significant technical development nowadays. Object detection deals with identifying semantic objects and certain classes from image. Object detection has a vast amount of applications in not just only robotics, but in day to day life also. Some major applications in the areas of object detection are security surveillance, self-driving automated vehicles, tracking objects, activity recognition, pedestrian detection etc. Detecting more than one objects from given image is one of the most difficult tasks in unsupervised learning. Object detection mainly concerns with detecting multiple objects/ things from single image or video. Whereas, Image classification algorithms identifies, main or primary (single) object from the image only. Multiple times classification algorithm is applied to the image to dig out all the objects from image. In this paper we would like to propose an object detection model that is capable of identifying objects from the given image.



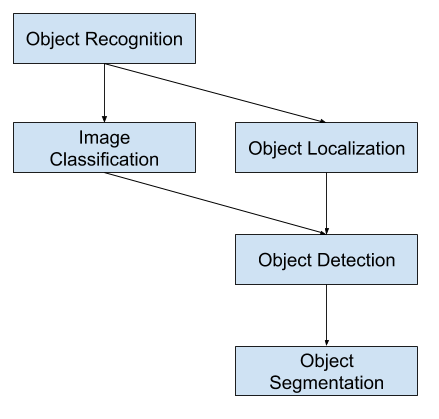
**Introduction:**

In computer vision, there are three types of tasks:

1. Object localization is primary concept behind all image detection algorithms. Object localization refers to detect objects and present them in different ways.

Instance segmentation is one of those ways in localization. In instance segmentation, a bounding box is created around the object and a label to which that object belongs is created at the top of the object. whereas, in image segmentation, only boundary is created around the object and its also pixel wise. Instance segmentation created square or rectangle boundaries.

1. Image classification: it concerns with type or class of the object in the given image. Output for this algorithm is a class label for input photograph.
2. Object detection: Object detection is a computer vision technique to identify various objects from an image or from a video. Object detection outputs class label with proper boundary boxes for objects in the image.



Various algorithms can be used for object detection using deep learning. Such as:

1. Yolo

YOLO uses convolutional neural networks for prediction of class labels and object’s location also. YOLOv3 is more accurate and faster than previous versions and SSD (Single Shot MultiBox Detector).

Let me introduce to previous versions of yolo in brief:

Yolo v1:

Yolo v 1uses darknet framework and works exactly same as yolov3 works. But main problem with yolov1 was, cluster classification. It cannot recognize small objects which are closer to each other. Algorithm reads them as a cluster and identifies only some of them.

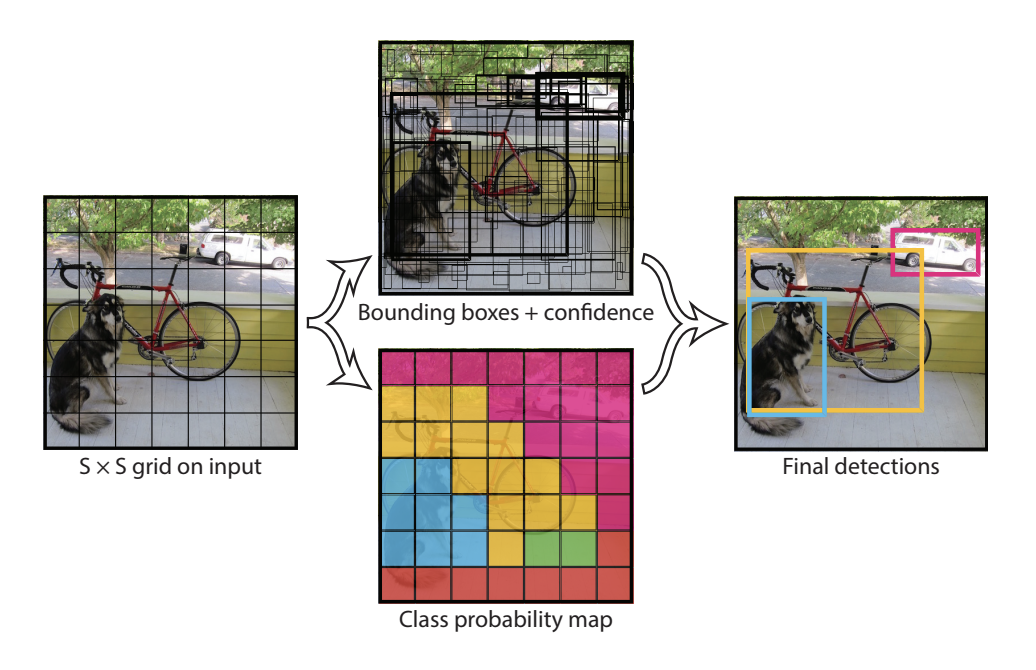
Yolov2:

Yolov2 introduced many new features over version 1. Such as, batch normalization, anchor boxes, multiscale training and also new darknet architecture. It works better than yolo v1 with a good accuracy, also detect small objects from image.

How does YOLO work?

YOLO looks at image only once. And then apply only one neural network on the image. The image is divided into **SxS** grid. Each cell can predict **N** bounding boxes. A rectangle is formed from these bounding boxes to enclose an object within it. So total, **SxSxN** boxes can be predicted from the given image. Each bounding boxes have its own predicted probabilities.

YOLO outputs a confidence score that tells us how certain it is that the predicted bounding box encloses the object.



1. SSD

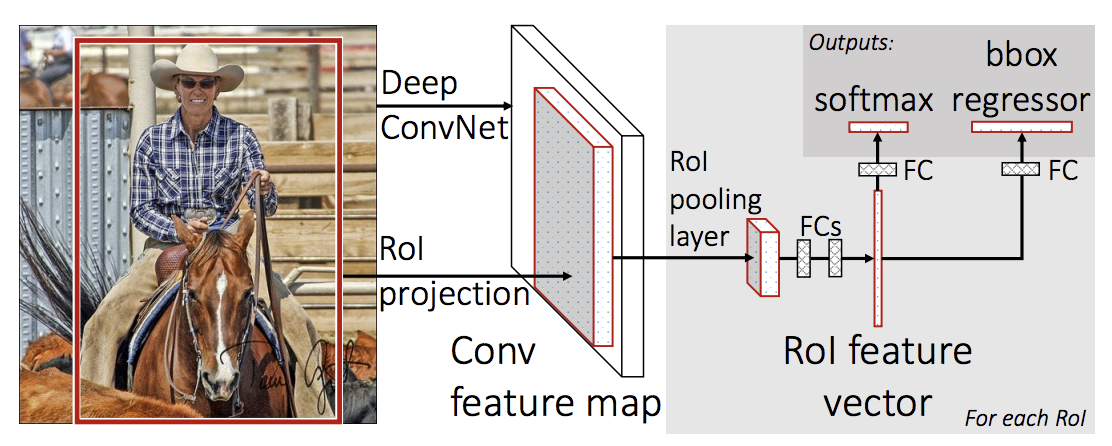
SSD (Single Shot MultiBox Detector) algorithms have a good balance between its accuracy and speed. SSD runs a small two-dimensional convolutional kernel on the feature map generated by CNN and image. By doing so, it automatically predicts bounding boxes and classification probability.

SSD works better on large objects than smaller one. We can say that its performance is as good as Faster -RCNN for large objects.

1. Faster R-CNN

Faster R-CNN consists of Region Proposal Network (RPN) and Fast-RCNN. Anchor boxes are introduced in faster-RCNN. There are three types of anchor boxes, 128x128, 256x256, 512x512. Additionally, it has three aspect ratios. (1:1, 2:1, 1:2). This gives a total of 9 boxes to predict the probability. This is output of RPN, which is given to Fast-RCNN. Here, spatial pooling technique is applied along with regression.

Faster R-CNN is 10 times faster than Fast R-CNN for the same accuracy output.



**Dataset description:**

For this project we used the dataset provided by our professor, MS COCO (microsoft common objects in context) dataset. The data is accomplished by gathering pictures of complex ordinary scenes containing basic items in their common setting. It is large-scale segmentation, captioning and object detection dataset. The task is to categorize each object based on the class shown in the image. The image feature consists of object segmentation. Also, there are more than 330 thousand images out of which more than 200 thousand images are labeled. There are 91 stuff, 80 object categories along with 1 and a half million object instances. Each image consists of at least 5 captions. The training set consists of 118 thousand images along with annotations - 2017 Train Images (18 GB), and 2017 Train/Val annotations (241 MB)

You can download dataset from here: <http://cocodataset.org/#download>

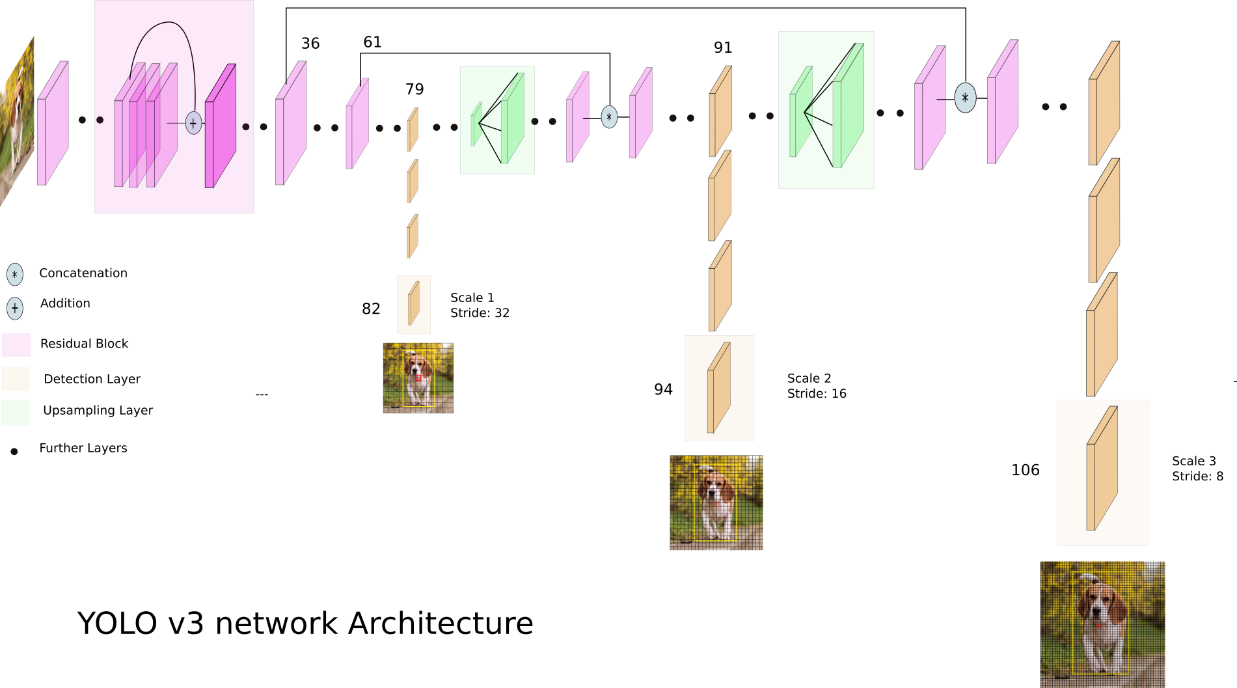
**Project description:**

1. Description:

We used MSCOCO dataset as mentioned above for model training. For yolo model implementation, we used keras library using tensorflow as backend. We used keras because it is easy to implement and works better with computer vision models and it is capable of running on the top of tensorflow.

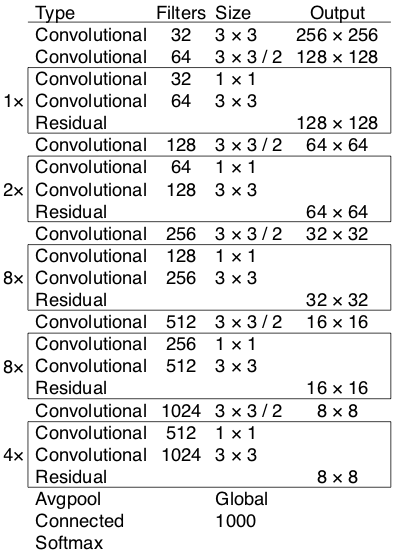
YOLO v3:

Following are primary concepts of yolov3 algorithm.



Darknet architecture:

Yolo v3 uses darknet – 53 architecture for feature extraction from image. As name suggest it has 53 convolutional layers. We know that the number of layers suggests how deep the architecture is. Yolov3 is a way deeper architecture than yolov2, as it goes only till 19 layers. There are 3x3 and 1x1 filters in yolov3 architecture which helps to extract features from the image, in every different layer.

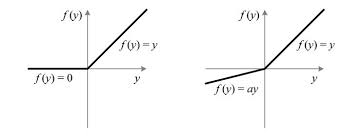


Leaky ReLU:

We all know that ReLU(Rectified Linear Unit) is a good activation function for neural networks. But onne step ahead, leaky relu is way better than ReLU itself. One of the most benefits given by leaky relu is, it almost removes “dying relu” problem. Dying relu problem refers to tha output given by relu function for negative inputs (It always 0). In leaky relu, there is a small negative slope for negative input values. Such that, output for that input values will be tends to zero but not actually zero, which helps in getting better neurons at each layes of convolutional network and so in output results.

Here is basic code to implement leaky relu in program and a chart to represent it’s functionality. On the left hand side is, ReLU function and other chart postulates leaky relu. The function, f(y) = ay is the slope line for negative values, and in practical, value of a is nearly around 0.01 – 0.001.

|  |  |
| --- | --- |
|  | def leaky\_relu(alpha, x):   if x&amp;amp;amp;amp;amp;amp;amp;lt;=0:    return x   else:    return alpha \* x |



Bounding boxes :

Bounding boxes are used to differentiate identified objects from one another. Logistic regression is used to predict confidence score of box.

Anchor boxes:

In single network layer, anchor boxes are responsible for classification and prediction. It uses k means clustering method for prediction.

Batch normalization:

Generally, we normalize our input data with activation functions or some other techniques to improve performance of our model, then can we normalize our hidden layers to improve model efficiency? Batch normalizarion is used for normalize input and hidden layers of the architecture for improvement. Conventionally, it improves accuracy by 2% and also overfitting can be reduced to negligible.

Accuracy threshold:

In our model, we used accuracy threshold value. Mostly what model does, is that it generates bounding boxes for the objects which are not clearly visible (which are not required to be identified) with lesser accuracy than other objects. After giving threshold accuracy value to model, it will not consider objects which gives less accuracy than threshold, which outputs in better detection.

Configuration parameter:

Additionally, model can predict more than one bounding boxes for one object which sometimes overlap on each other. Configursation parameter is set as overlap amount for boxes. We used different values for the parameter to check how those boxes are made for different values. If we consider its value as 1.0, then it will show all the bounding boxes possible around an object. And for 0.1, it will remove some required boxes from account. So generally, 0.5 or 0.6 are used to get better and systematic output.

1. Main references used for your project:
2. From a blog by “Ayoosh Kathooria” on medium.com, we get to know about overall architecture and other features of YOLOv3. Like, bounding boxes, improvements in yolov3 over yolov2, anchor boxes, loss function. Reference no. [6]
3. From this blog we get to know about how yolov3 works. Reference no. [[7]](https://pjreddie.com/darknet/yolo/)
4. Working of keras in object detection. [8]

Update the refereneces used numbers as per our references list below:

1. Difference in APPROACH/METHOD between your project and the main projects of your references

The references that we used are using mostly pre trained models and weights of dataset. Also, they were some complex models with lesser accuracy than what we got in out model. Some models were working with a lot of data, which were out of our resources. They have used some limited dataset as well as some of them have used a lot data. But in our project, ms coco dataset is used for getting accuracy. We implemented configuration parameter and threshold accuracy with a good combination to achieve best accuracy.

For object detection, generally, leaky relu is used as activation function and we are using that one only in the model. But for testing purposes we once tried simple relu function but did not get desired results. Other than that, all the parameters like, anchor boxes, bounding boxes, cnn layers are retained in our model.

1. Difference in ACCURACY/PERFORMANCE between your project and the main projects of your references

saw a lot of implementation codes and snippets we got some idea about implementation.

1. Configuration parameter

In our project there is a function which sets configuration parameter. What this parameter does is, it generalizes the bounding boxes which generate around one object only. Foe a single object we get multiple boxes. This parameter chooses a best box from them which can get a label accordingly. In one of those references, one person did not set the value for the configuration parameter and gets the accuracy around 70%. But after implementing this parameter our accuracy has risen by 8 to 10%.

1. Accuracy threshold value

While using different threshold values for accuracy, we get different number of objects in each output. This is because, evry time we change threshold value, objects with accuracy below threshold value will be dropped from final accuracy measurement and output. And model will not get same number of objects each time.

These are some results we get for same image with different threshold parameters, but with same configuration parameter value (0.5).

|  |  |  |
| --- | --- | --- |
| Accuracy Threshold value | Accuracy | Number of objects: |
| 60% (0.6) | 88.165% | 7 |
| 40% (0.4) | 83.94% | 8 |
| 80% (0.8) | 98.54% | 4 |

1. List of your contributions in the project

Data augmentation. (Augmented annotation files with dataset). With the threshold and configuration parameters ideal combination we took out the best accuracy. Trained the Keras model using Yolov3 to generate new weights to detect the objects in the image. Tried different activation functions to see the variation in the accuracy of the image detection.

**Analysis:**

1. What did you do well?

At the starting phase of the project we did not have any idea about keras, tensorflow, neural networks, activation functions, yolo models. But now we have a good idea about these terms and we are able to create a basic model to classify some objects from image.

The references we have used are calculating accuracy for each individual object, separately. We combine all accuracies and achieve a more general accuracy for a single image. Which helps to get more general idea about overall performance of algorithm.

1. What could I have done better?

At the starting phase of the project we were thinking about implementing the project not only for image or single photo, but also for video. Such as, detecting cars and people in given video of a city traffic. But because of less time, more and more difficulty to implement project, limited amount of resources we were not able to implement this idea.

If we were able to implement this thing than it could be helpful in big areas of computer vision like surveillance system, unmanned vehicle systems, robotics, etc.

1. What Is left for future work?

For future work, we can work on getting better and better accuracy for just a single image. After that, we can think of our basic and main idea to implement it for real time video, for tracking application. This application requires a lot of high computation computers to generate an effective algorithm and proper environment to deploy that system.

One more idea is that, making a shopping application which detect the object and gives best price, link to buy it from and reviews of that item.

Conclusion:

Object detection is key concept in robotics and computer vision areas’ researches. It needs a high amount of attention to give superior results in better advancements of humankind. although, there have been so many researches done and are going in computer vision, but they are not enough as compare to technological advancements of 21st century. Object detection can be highly used in real time applications like, tracking applications, surveillance systems, pedestrian detection, unmanned vehicle systems.

But what we have learn so far in this project, for object identification, Yolov3 is a well-defined approach. It gives good accuracy even for small datasets, because of having deep architecture and complex model network.

References:

1. MS COCO Dataset: <https://arxiv.org/abs/1405.0312>
2. Understanding of project - <https://towardsdatascience.com/object-detection-using-yolov3-and-opencv-19ee0792a420>
3. Working of model - <https://pjreddie.com/darknet/yolo/>
4. Implementing object detection for video- <https://towardsdatascience.com/real-time-mobile-video-object-detection-using-tensorflow-a75fa0c5859d>
5. Other papers:

<https://medium.com/@venkatakrishna.jonnalagadda/object-detection-yolo-v1-v2-v3-c3d5eca2312a>

<https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b>

<https://towardsdatascience.com/object-detection-with-neural-networks-a4e2c46b4491>