iJRASET

International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: XI Month of publication: November 2021

DOI: https://doi.org/10.22214/ijraset.2021.39044

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

Real Time Object Detection Using Yolo

Ankith I[1], Akshaya H P[2]
[1]UG Scholar, Dept. of CSE, BNM Institute of Technology, Bengaluru, Karnataka
[2]UG Scholar Dept. of CSE, BNM Institute of Technology, Bengaluru, Karnataka

Abstract: Object detection is related to computer vision and involves identifying the kinds of objects that have been detected. It is challenging to detect and classify objects. Recent advances in deep learning have allowed it to detect objects more accurately. In the past, there were several methods or tools used: R-CNN, Fast-RCNN, Faster-RCNN, YOLO, SSD, etc. This research focuses on "You Only Look Once" (YOLO) as a type of Convolutional Neural Network. Results will be accurate and timely when tested. So, we analysed YOLOv3's work by using Yolo3-tiny to detect both image and video objects.

Keywords: YOLO, Intersection over Union (IOU), Anchor box, Non-Max Suppression, YOLO application, limitation.

I. INTRODUCTION

Object detection is a computer technology which is related to computer vision and image processing, which deals with the detection of semantic objects that belong to a particular class (such as humans, buildings, or cars) appearing in digital images and videos. Among the well-researched areas of object detection, there is the detection of faces and pedestrians. There are many applications of object detection in computer vision, including image retrieval and video surveillance, including finding objects in images. The term “analysis of the scene” is usually used as a synonym for detecting every object in the scene, classifying each of their labels, and determining their bounding box (or polygons) in a more convenient way. The process of detecting objects includes many approaches including fast R-CNNs, Retina-Nets, and Single-Shot MultiBox Detectors (SSDs). It is under these approaches that data limitation and modelling challenges have been solved in the area of object detection. However, still they are not capable of detecting all objects in a given algorithm run. In recent years, the Yolo algorithm has gained a great deal of attention due to its superior performance when compared to other methods of object detection.

A. What is YOLO?

The term YOLO is an abbreviation for the expression ‘You Only Look Once’. In real-time, this algorithm is used to detect and recognize objects in a picture that are hidden within it. With YOLO, object detection is accomplished in real time by implementing the convolutional neural networks (CNNs) that are employed to identify objects. YOLO is implemented as a regression problem and our algorithm provides the values of cluster probabilities for the detected images. The algorithm uses a neural network to sense objects and essentially only requires a single forward propagation through the network to do so. Accordingly, the entire image is predicted for a single algorithm run, i.e., the prediction is run on the whole image. With the CNN, multiple class probability predictions are made, and bounding boxes are projected at the same time simultaneously. Different variations of the YOLO algorithm are available. These include the tiny Yolo and the Yolov3 which are some of the most common ones.

B. Why YOLO Algorithm is Important?

Because of the following reasons, the YOLO algorithm is important:

1) *Speed:* Because this algorithm can predict objects in real time, it improves the speed of the detection process.
2) *High Accuracy:* YOLO is a method of predicting that produces a high level of accuracy when compared to a comparison technique.
3) *Learning Capabilities:* This algorithm exhibits excellent learning capabilities, which allow it to acquire considerable knowledge of how objects are represented, and then apply that knowledge to object detection.

C. Different Techniques used in YOLO?

Three techniques are used in the YOLO algorithm to achieve its goal:

1) Residual blocks
2) Bounding box regression
3) Intersection Over Union (IOU)

a) *Residual Blocks:* To begin with, the image is divided into different grids. It is important to note that each grid has a dimension of S x S. Here is an illustration of how a picture is divided into grids based on its input.

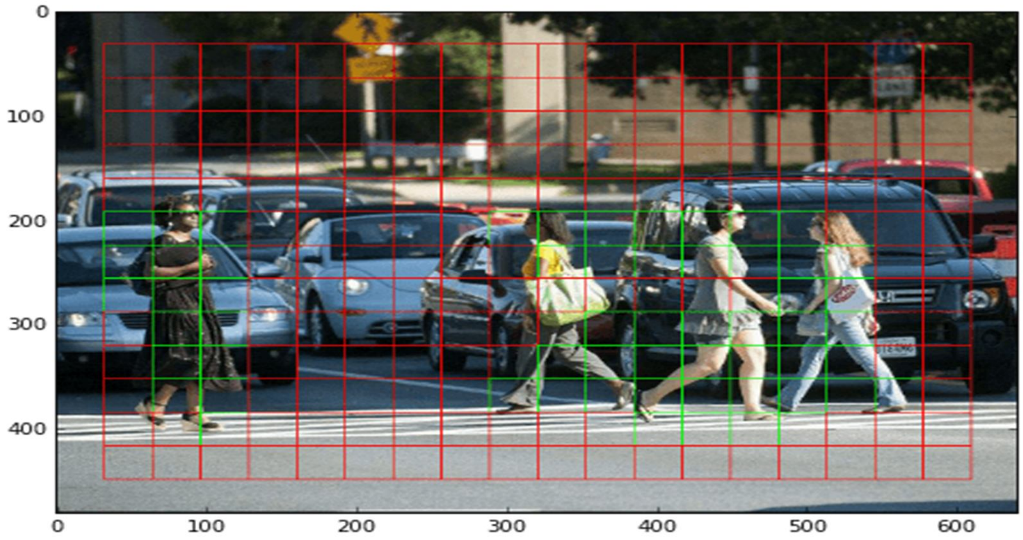


Fig 1. Residual blocks

It is apparent from the image above that there are many grid cells of equal dimensions. Therefore, any object that appears within a grid cell will be detected. It can be understood from the above example that if an object centre appears within a specific grid cell, then it is this cell which will be responsible for detecting it.

b) *Bounding Box Regression:* In an image, a bounding box is an outline that highlights an object in the picture.
The following attributes are associated with every bounding box in the image:

- Width (in bw).
- Height (bh).
- Class (for example, person, car, traffic light, etc.)- This is represented by the letter c.
- Bounding box center (bx,by).

In the image below, you can see an example of a bounding box. In the image below, you can see that the bounding box has been represented by a yellow outline.

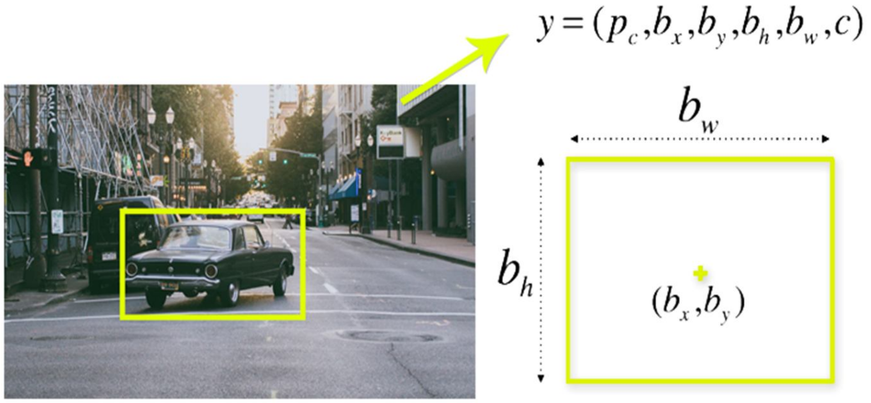


Fig 2. Bounding box regression

In general, YOLO is used in order to predict height, width, centre, and class of objects using a single bounding box regression. According to the image above, represents the probability that an object will appear within the bounding box.

c) *Intersection Over Union (IOU):* In object detection, intersection over union (IOU) describes the way in which boxes overlap each other. YOLO provides an output box that encircles each object perfectly with the use of IOU. The bounding boxes must be predicted by each grid cell as well as the confidence scores associated with these boxes. A predicted bounding box is defined as being equal to a real box if the IOU is equal to 1. By this mechanism, bounding boxes that don't align to the real box will be eliminated.

This image illustrates how a simple IOU works by providing a simple example.

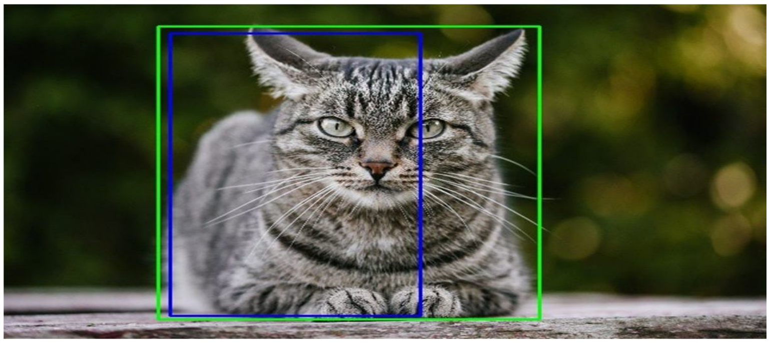


Fig 3. Intersection over union (IOU)

It is evident from the image above that there are two bounding boxes, one that is green and one that is blue. It would appear that the blue box is a predicted box, while the green box is the actual box. YOLO ensures that the bounding boxes of the two boxes are equal, as expected.

D. Combination of the Three Techniques

Using the following image, we can see how the three techniques are applied throughout the detection process to produce the final detection results.

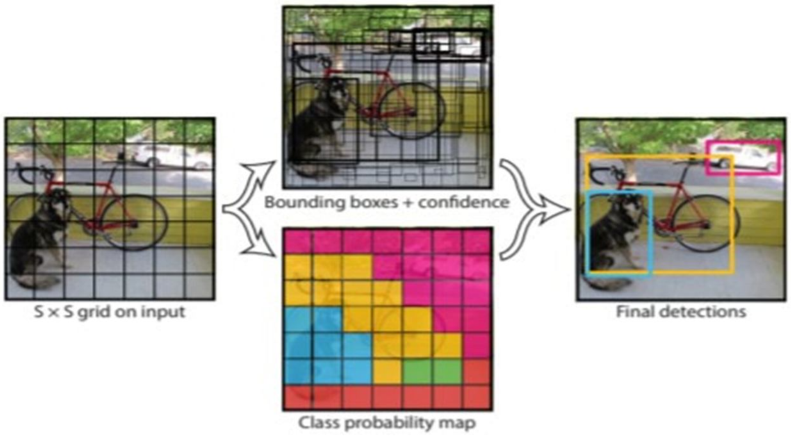


Fig 4. Combination of three techniques

In the first step, the image is divided into grid cells. Based on the bounding boxes generated by each grid cell, confidence scores are calculated. The cells predict the probabilities of classifying each object based on those bounding boxes.

As an example, we can notice that at least three classes of objects are visible at any given time: a car, a dog, and a bicycle. In this method, a single convolutional neural network is used to make all the predictions simultaneously. It is important to note that intersection over union is simply the method of ensuring that the predicted boxes of bounding are equal to the real boxes of objects. By using this phenomenon, it is possible to remove unnecessary bounding boxes that do not correspond to the characteristics of the objects (such as height and width). As the result of the final detection, we will be provided with bounding boxes that are unique and fit the objects perfectly. There are two different types of bounding boxes. For example, the pink bounding box surrounds the car while the yellow bounding box surrounds the bicycle. Blue bounding boxes have been used to highlight the dog in this image.

II. MOTIVATION

Although a convolutional neural net (CNN) is used under the hood of YOLO, it’s still able to detect objects with real-time performance. It’s possible thanks to YOLO’s ability to do the predictions simultaneously in a single-stage approach.

Other, slower algorithms for object detection (like Faster R-CNN) typically use a two-stage approach:

1) in the first stage, interesting image regions are selected. These are the parts of an image that might contain any objects.
2) in the second stage, each of these regions is classified using a convolutional neural net.

Usually, there are many regions on an image with the objects. All these regions are sent to classification. Classification is a time-consuming operation, which is why the two-stage object detection approach performs slower compared to one-stage detection. YOLO doesn’t select the interesting parts of an image, there’s no need for that. Instead, it predicts bounding boxes and classes for the whole image in a single forward net pass.

III. WORKING OF YOLO

As part of YOLO, the idea of dividing images into grid cells is unique, with the grids being defined as S x S. The centre of an object will be determined by the grid of the cells in which that object is located.

A. Each of the cell Grid will Have

$$y = \begin{bmatrix} pc \\ bx \\ by \\ bw \\ bh \\ c \end{bmatrix}$$

Fig 5. Outputs each grid cell

- ***pc*** represents the probability that the object will be present in the grid cell. If there is no object in the grid cell, then ignore it.
- ***bx, by*** points to the center of the object found in the grid cell. In the event that the center of the object does not fall into that grid cell, then it does not need to be calculated.
- ***bw, bh*** are measurements of the box's width and height respectively.
- ***c*** is the number of categories that will be calculated according to the specified category.

B. As an Example

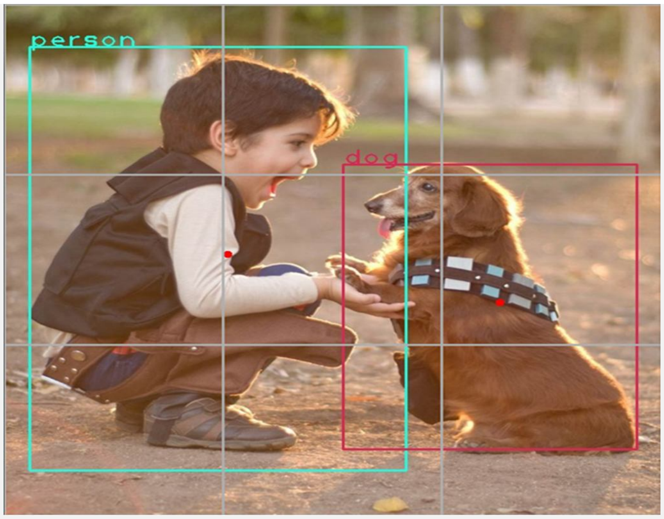
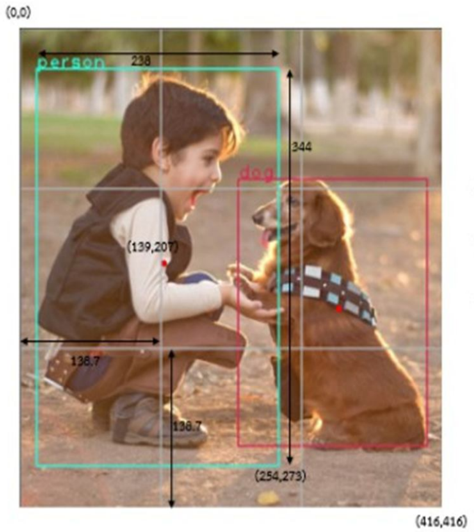


Fig 6. Sample picture to explain how YOLO works

From this example, we are going to divide the grid cell into 3x3 cells. We will define 3 different classes, that are, you guessed it, people, dogs, and cats. As a result, in this image, Y will be as follows:

$$y = \begin{bmatrix} pc \\ bx \\ by \\ bh \\ bw \\ c1 \\ c2 \\ c3 \end{bmatrix}$$

From the picture, you can see that the center of the image is in a different grid cell, so you can find y from:



x = (139-138.7)/138.7 = 0.002

y = (207-138.7)/138.7 = 0.49

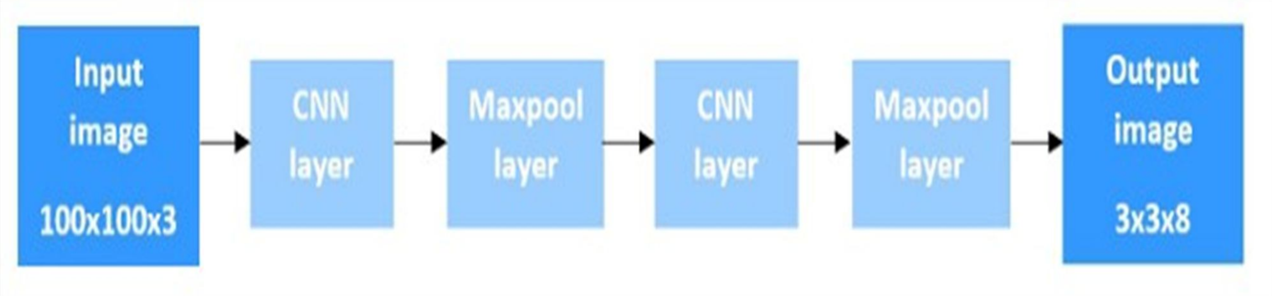
w = 238/417 = 0.57

h = 344/417 = 0.82

$$y_{person} = \begin{bmatrix} pc \\ bx \\ by \\ bh \\ bw \\ c1 \\ c2 \\ c3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.002 \\ 0.49 \\ 0.82 \\ 0.57 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad y_{dog} = \begin{bmatrix} pc \\ bx \\ by \\ bh \\ bw \\ c1 \\ c2 \\ c3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.22 \\ 0.78 \\ 0.55 \\ 0.44 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Fig 7. Calculating the y value of each object

For all grids, we have a vector output. This result will be in **3 X 3 X 8.**



C. Anchor Box

Anchor boxes are YOLO's algorithm that separates objects if multiple image centers are in the same grid cell.

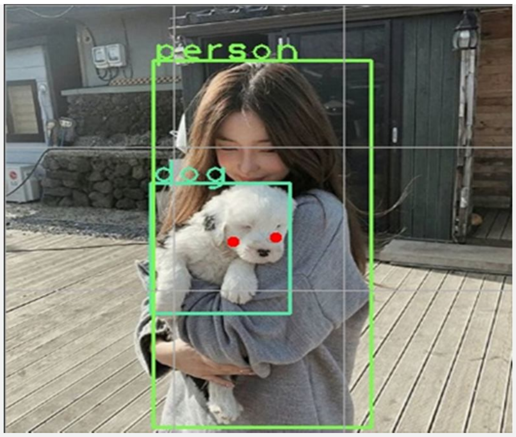


Fig 8. Anchor box example

Seeing the picture, you are able to see that the centre of both objects is inside the same grid cell. We can solve this issue by adding a "dimension" to the output. For the purpose of this discussion, we are going to look at these two objects separately.

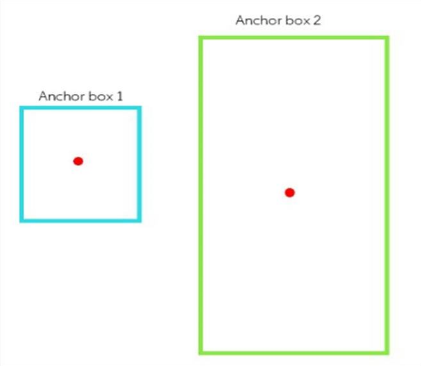


Fig 9. Anchor box

Find the y of each anchor box and then add them together.

$$y = \begin{bmatrix} pc \\ bx \\ by \\ bw \\ bh \\ c1 \\ c2 \\ c3 \\ pc \\ bx \\ by \\ bw \\ bh \\ c1 \\ c2 \\ c3 \end{bmatrix}$$

Therefore, the prediction of all vector class outputs will be **3 X 3 X (2 * 8)** will be **3 X 3 X 16**, which can be obtained from **S X S X (B * 5 + C)**.

- S x S is the grid
- B is anchor box
- C is class

D. Intersection over Union (IoU)

The accuracy of the detector can be assessed using this indicator because it is an evaluation indicator. As we are looking at the predicted box against the detectable box, how can we determine whether or not this is a good prediction? The area can be calculated by using the formula below:

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union}$$

If the IoU value is close to 1, then our predictions are highly accurate.

E. Non-Max Suppression

Basically, it is a general algorithm used to predict multiple boxes for the same object. Probability of an object and IOU value are considered in tandem, which means that when an object is detected along with the probability of multiple objects, we will consider the object with the highest probability first. After you have found the value for each of the other boxes, you then need to find the box with the higher value for the IoU.

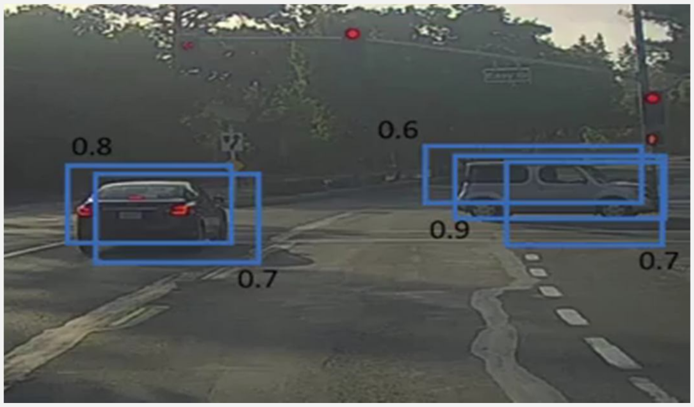


Fig 10. Non-Max Suppression image

IV. APPLICATION

YOLO algorithm is applicable in all the following fields:

1) *Autonomous Driving:* As part of autonomous vehicles, Yolo can detect objects around the car including vehicles, people, and even parking signals in order to avoid collisions. It is the goal of autonomous cars to avoid colliding with objects in the path, given the fact that no human is driving them.
2) *Wildlife:* There are a number of types of animals that can be spotted through this algorithm in forests. The detection of animals using this method is used by wildlife rangers and journalists to identify animals in images (both recorded and real-time) and videos. There are several animals that can be detected in this area, including giraffes, elephants, and bears.
3) *Security:* Yolo can also be used in security systems as a method of enforcing security in an area. Let's imagine for a moment that certain people have been restricted from passing through a certain part of the city due to security concerns. As someone passes through the restricted area, the YOLO algorithm will identify him/her, requiring security personnel to take further action.

V. LIMITATION

As each grid cell can only predict two boxes and have one class, YOLO imposes strong spatial constraints on bounding box prediction. Because of this spatial constraint, our model can only predict a limited number of objects nearby. As a result, our model is unable to handle small objects that appear in groups, such as bird flocks. Due to the way our model learns to predict bounding boxes from data, it has difficulty generalizing to objects with new or unusual aspect ratios or configurations. Since our model uses multiple down sampling layers from the input image, it also uses relatively coarse features to predict bounding boxes. In conclusion, while we train on a loss function that approximates detection performance, our loss function treats errors the same way regardless of whether the bounding boxes are small or large. It is generally benign to make a small error in a large box, however, a small error in a small box will have an adverse effect on the IOU. The main cause of error is incorrect localization.

VI. CONCLUSION

Whenever we view images or videos, we can easily identify and locate the objects of our interest within moments of viewing. To pass on this intelligence to computers is nothing more than a process of object detection, location, and identification. During its development, Object Detection has found its applications in a wide variety of domains, such as video surveillance, image retrieval systems, autonomous driving vehicles, and many others. Various algorithms can be used for object detection, but we will As a real-time object detection algorithm, YOLO is one of the fastest (45 frames per second) as compared to the R-CNN family of algorithms (R-CNN, Fast R-CNN, Faster R-CNN, etc.). -CNN, Fast R-CNN, Faster R-CNN, etc.). Thus, in the following paper we will show how Yolov3 works & how it can be used for Object detection using YOLOv3-tiny as a testing tool, since this software has less memory and is the fastest of all Yolo. YOLOv3-tiny was shown in the experiment to be able to detect objects quickly and precisely enough.

REFERENCES

[1] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In Computer Vision– ECCV 2008, pages 2–15. Springer, 2008.

[2] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In International Conference on Computer Vision (ICCV), 2009.

[3] H. Cai, Q. Wu, T. Corradi, and P. Hall. The crossdepiction problem: Computer vision algorithms for recognising objects in artwork and in photographs. arXiv preprint arXiv:1505.00110, 2015.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886–893. IEEE, 2005.

[5] T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijaya narasimhan, J. Yagnik, et al. Fast, accurate detection of 100,000 object classes on a single machine. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 1814–1821. IEEE, 2013.

[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. arXiv preprint arXiv:1310.1531, 2013.

[7] J. Dong, Q. Chen, S. Yan, and A. Yuille. Towards unified object detection and semantic segmentation. In Computer Vision–ECCV 2014, pages 299–314. Springer, 2014.

[8] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2155–2162. IEEE, 2014.