International Journal of Research in Engineering, Science and Management
Volume-3, Issue-3, March-2020
www.ijresm.com | ISSN (Online): 2581-5792

Classification and Identification of Object in an Image

Akshay Manjunath Naik[1], S. S. Krushitha[2], C. V. Jeevan[3], N. Hariprasad[4], Padmanayana[5]
[1,2,3,4]*B.E. Student, Dept. of Computer Science & Engineering, Srinivas Institute of Technology, Mangaluru, India*
[5]*Professor, Dept. of Computer Science and Engineering, Srinivas Institute of Technology, Mangaluru, India*

***Abstract*: Computer Vision is the branch of the science of computers and software systems which can recognize as well as understand images and scenes. Computer Vision is consists of various aspects such as image recognition, object detection, Supply vector machine. Object detection is widely used for face detection, vehicle detection, web images. In this project, we are using highly accurate object detection-algorithms, SVM and Fast yet highly accurate ones like YOLO. Using these methods and algorithms, based on machine learning requires lots of mathematical and deep learning frameworks understanding by using dependencies such as OpenCV, imageai etc, we can detect each and every object in image by the area object in a highlighted rectangular boxes and identify each and every object and assign its tag to the object. This also includes the accuracy of each method for identifying objects. The Objective is to detect of objects using You Only Look Once (YOLO) approach. This method has several advantages as compared to other object detection algorithms.**

***Keywords*: Convolutional Neural Network, Fast-Convolutional Neural Network, Bounding Boxes, YOLO.**

1. Introduction

Object detection is a technology that detects the semantic objects of a class in digital images and videos. One of its real-time applications is self-driving cars. In this, our task is to detect multiple objects from an image. The most common object to detect in this application is the car, motorcycle, and pedestrian. For locating the objects in the image we use Object Localization and have to locate more than one object in real-time systems. There are various techniques for object detection, they can be split up into two categories, first is the algorithms based on Classifications. CNN and RNN come under this category. In this, we have to select the interested regions from the image and have to classify them using Convolutional Neural Network. This method is very slow because we have to run a prediction for every selected region. The second category is the algorithms based on Regressions. YOLO method comes under this category. In this, we won't select the interested regions from the image. Instead, we predict the classes and bounding boxes of the whole image at a single run of the algorithm and detect multiple objects using a single neural network. YOLO algorithm is fast as compared to other classification algorithms. In real time our algorithm process 45 frames per second. YOLO algorithm makes localization errors but predicts less false positives in the background title also can be copied and paste it, when you need new section and type the section heading as per your requirement.

2. Literature survey

You Only Look Once: Unified, Real-Time Object Detection, by Joseph Redmon. Their prior work is on detecting objects using a regression algorithm. To get high accuracy and good predictions they have proposed YOLO algorithm in this paper. Understanding of Object Detection Based on CNN Family and YOLO, by Juan Du. In this paper, they generally explained about the object detection families like CNN, R-CNN and compared their efficiency and introduced YOLO algorithm to increase the efficiency. Learning to Localize Objects with Structured Output Regression, by Matthew B. Blaschok. This paper is about Object Localization. In this, they used the Bounding box method for localization of the objects to overcome the drawbacks of the sliding window method.

3. Working of YOLO algorithm

First, an image is taken and YOLO algorithm is applied. In our example, the image is divided as grids of 3x3 matrixes. We can divide the image into any number grids, depending on the complexity of the image. Once the image is divided, each grid undergoes classification and localization of the object. The objectness or the confidence score of each grid is found. If there is no proper object found in the grid, then the objectness and bounding box value of the grid will be zero or if there found an object in the grid then the objectness will be 1 and the bounding box value will be its corresponding bounding values of the found object. The bounding box prediction is explained as follows. Also, Anchor boxes are used to increase the accuracy of object detection which also explained below in detail.

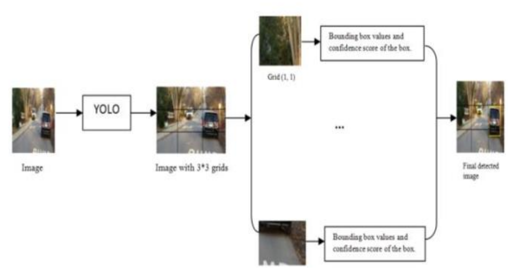

Fig. 1. Working of YOLO

International Journal of Research in Engineering, Science and Management
Volume-3, Issue-3, March-2020
www.ijresm.com | ISSN (Online): 2581-5792

A. Bounding box predictions

YOLO algorithm is used for predicting the accurate bounding boxes from the image. The image divides into S x S grids by predicting the bounding boxes for each grid and class probabilities. Both image classification and object localization techniques are applied for each grid of the image and each grid is assigned with a label. Then the algorithm checks each grid separately and marks the label which has an object in it and also marks its bounding boxes. The labels of the gird without object are marked as zero.

1) Real-time object detection with YOLO

Fig. 2. Example image with 3x3 grids

Consider the above example, an image is taken and it is divided in the form of 3 x 3 matrixes. Each grid is labelled and each grid undergoes both image classification and objects localization techniques. The label is considered as Y. Y consists of 8 values.

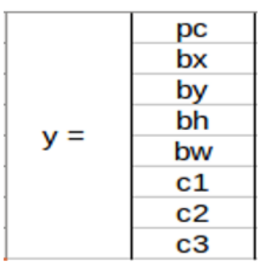

Fig. 3. Elements of label Y

Pc – Represents whether an object is present in the grid or not. If present pc=1 else 0. bx, by, bh, bw – are the bounding boxes of the objects (if present). c1, c2, c3 – are the classes. If the object is a car then c1 and c3 will be 0 and c2 will be 1. In our example image, the first grid contains no proper object. So it is represented as,

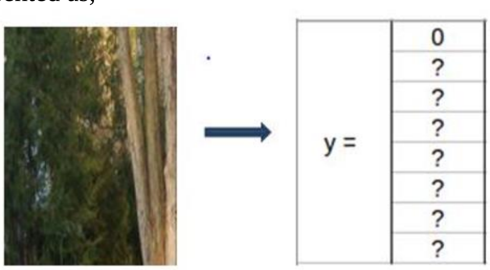

Fig. 4. Bounding box and Class values of grid 1

In this grid, there exists no proper object so the pc value is a "0".

And rest of the values are doesn't matter because there exists no object. Consider a grid with the presence of an object. Both 5th and 6th grid of the image contains an object. Let' consider the 6th grid, it is represented as,

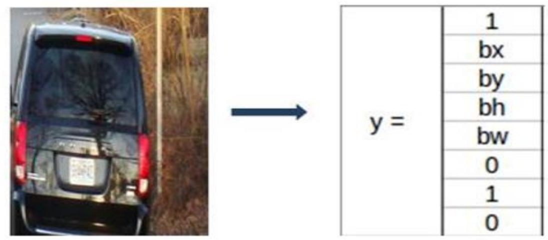

Fig. 5. Bounding box and Class values of grid 6

In this table, 1 represents the presence of an object. And bx, by, bh, bw are the bounding boxes of the object in the 6th grid. And the object in that grid is a car so the classes are (0,1,0). The matrix form of Y in this is Y=3x3x8.

For the 5th grid also the matrix will be little similar with different bounding boxes by depending on the objects position in the corresponding grid.

If two or more grids contain the same object then the center point of the object is found and the grid which has that point is taken. For this, to get the accurate detection of the object we can use to methods. They are Intersection over Union and Non-Max Suppression.

In IoU, it will takes the actual and predicted bounding box value and calculates the IoU of two boxes by using the formulae,

IoU = Area of Intersection / Area of Union.

If the value of IoU is more than or equal to our threshold value (0.5) then it's a good prediction. The threshold value is just an assuming value. We can also take greater threshold value to increase the accuracy or for better prediction of the object.

B. Accuracy improvement anchor box

By using Bounding boxes for object detection, only one object can be identified by a grid. So, for detecting more than one object we go for Anchor box.

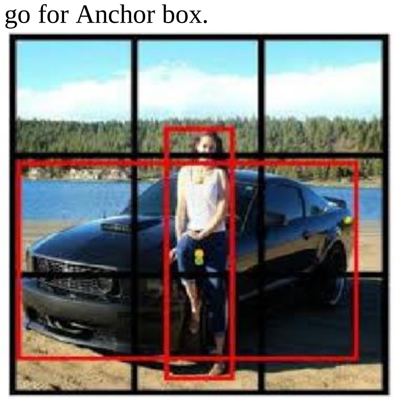

Fig. 6. An example image for anchor box

Consider the above picture, in that both the human and the car's midpoint come under the same grid cell. For this case, we use the anchor box method. The red color grid cells are the two anchor boxes for those objects. Any number of anchor boxes can be used for a single image to detect multiple objects. In our case, we have taken two anchor boxes.

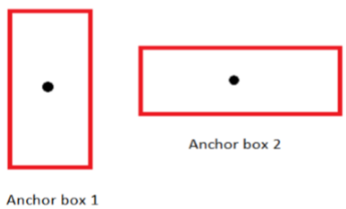

Fig. 7. Anchor boxes

The above figure represents the anchor box of the image we considered. The vertical anchor box is for the human and the horizontal one is the anchor box of the car. In this type of overlapping object detection, the label Y contains 16 values, the values of both anchor boxes.

4. System requirement

Install Python on your computer system

1. Install ImageAI and its dependencies like tensorflow, Numpy, OpenCV, etc.
2. Download the Object Detection model file (Retinanet)

Steps to be followed:

1. Download and install Python version3 from official Python Language website, *https://python.org*
2. Install the following dependencies via pip.

1) Tensorflow

Tensorflow is an open-source software library for dataflow and differentiable programming across a range of tasks. It is an symbolic math library, and is also used for machine learning application such as neural networks, etc. It is used for both research and production by Google.

Tensorflow is Google Brain's second-generation system.1st Version of tensorflow was released on February 11, 2017.While the reference implementation runs on single devices, Tensor flow can run on multiple CPU's and GPU (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). TensorFlow is available on various platforms such as64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

The architecture of tensorflow allows the easy deployment of computation across a variety of platforms (CPU's, GPU's, TPU's), and from desktops - clusters of servers to mobile and edge devices.

- pip install tensorflow-command

2) Numpy

NumPy is library of Python programming language, adding support for large, multi-dimensional array and matrice, along with large collection of high-level mathematical function to operate over these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several developers. In 2005 Travis Olphant created NumPy by incorporating features of computing Numarray into Numeric, with extension modifications. NumPy is open-source software and has many contributors.

- pip install numpy –command

3) SciPy:

SciPy contain modules for many optimizations, linear algebra, integration, interpolation, special function, FFT, signal and image processing, ODE solvers and other tasks common in engineering. SciPy abstracts majorly on NumPy array object, and is the part of the NumPy stack which include tools like Matplotlib, pandas and SymPy, etc., and an expanding set of scientific computing libraries. This NumPy stack has similar uses to other applications such as MATLAB, Octave, and Scilab. The NumPy stack is also sometimes referred as the SciPy stack.

- pip install scipy -command

4) ImageAI

ImageAI provides API to recognize 1000 different objects in a picture using pre-trained models that were trained on the ImageNet-1000 dataset. The model implementations provided are SqueezeNet, ResNet, InceptionV3andDenseNet.

- pip3 install imageai—upgrade

5. Methodology

A. SqueezeNet

SqueezeNet is name of a DNN for computer vision. SqueezNet is developed by researchers at DeepScale, University of California, Berkeley, and Stanford University together. In SqueezeNet design, the authors goal is to create a smaller neural network with few parameters that can more easily fit into memory of computer and can more easily be transmitted over a computer network.

Table 1

DNN Model	Application	Original Implementation	Other Implementations
SqueezeDet	Object Detection on Images	TensorFlow	Caffe, Keras
SqueezeSeg	Semantic Segmentation of LIDAR	TensorFlow	
SqueezeNext	Image Classification	Caffe	TensorFlow, Keras, PyTorch
SqueezeNAS	Neural Architecture Search for Semantic Segmentation	PyTorch	

SqueezeNet is originally released in 2016. This original version of SqueezeNet was implemented on top of the Caffe deep learning software framework. The open-source research community ported SqueezeNet to a number of other deep learning frameworks.

SqueezeNet ships as part of the source code of a number of deep learning frameworks such as PyTorch, Apache MXNet, and Apple CoreML. In addition, 3rd party developers have created implementation of SqueezeNet that are compatible with frameworks such as TensorFlow.

B. InceptionV3

Inception v3 is widely used as image recognition model that has showed to obtain accuracy of greater than 78.1% on the ImageNet dataset. The model is the culmination of many ideas developed by researchers over years. It is based on “Rethinking the Inception Architecture Computer Vision” by Szegedy.

The model is made of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batch norm is used more throughout the model and applied to activation inputs. Loss is computed via SoftMax.

C. DenseNet:

DenseNet stands for Densely Connected Convolutional Networks it is one of the latest neural networks for visual object recognition. It is similar to ResNet but has some fundamental differences.

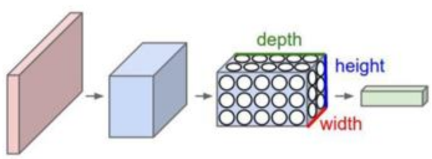

Fig. 8. DenseNet with other convolution networks

6. Results and Discussion

Fig. 9. Before detection

This is a sample image we feed to the algorithm and expect our algorithm to detect and identify objects in the image and label them according to the class assigned to it.

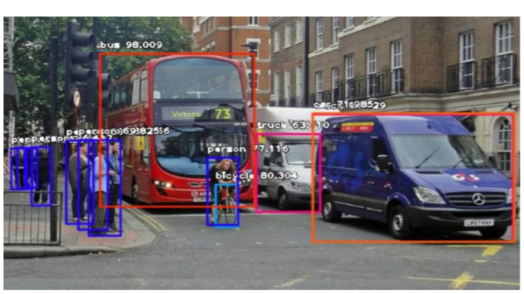

Fig. 10. After detection

As expected our algorithm identifies the objects by its classes an assigns each object by its tag and has dimensions on detected image.

ImageAI provides many more features useful for customization and production capable deployments for object detection tasks. Some of the features supported are:

Adjusting Minimum Probability: By default, objects detected with a probability percentage of less than 50 will not be shown or reported. You can increase this value for high certainty cases or reduce the value for cases where all possible objects are needed to be detected.

Custom Objects Detection: Using a provided Custom Object class, you can tell the detection class to report detection so none or a few number of unique objects.

Detection Speeds: You can reduce the time it takes to detect an image by setting the speed of detection speed to “fast”, “faster” and “fastest”.

Input Types: You can specify and parse in file path to an image, Numpy array or file stream of an image as the input image.

Output Types: You can specify that the detect Objects from Image function should return the image in the form of a file or Numpy array.

7. Conclusion

In this paper, we proposed about YOLO algorithm for the purpose of detecting objects using a single neural network. This algorithm is generalized, it outperforms different strategies once generalizing from natural pictures to different domains. The algorithm is simple to build and can be trained directly on a complete image. Region proposal strategies limit the classifier to a particular region. YOLO accesses to the entire image in predicting boundaries. And also it predicts fewer false positives in background areas. Comparing to other classifier algorithms this algorithm is much more efficient and fastest algorithm to use in real time.

8. Future enhancements

The object recognition system can be applied in the area of surveillance system, face recognition, fault detection, character recognition etc. The objective of this thesis is to develop an object recognition system to recognize the 2D and 3D objects in the image. The performance of the object recognition system depends on the features used and the classifier employed for recognition. This research work attempts to propose a novel feature extraction method for extracting global features and and obtaining local features from the region of interest. Also, the research work attempts to hybrid the traditional classifiers to recognize the object. The object recognition system developed in this research was tested with the benchmark datasets like COIL100, altech101, ETH80 and MNIST. The object recognition system is implemented in MATLAB7.5.

Specifically, the contributions towards this research work are as follows, An object recognition system is developed, that recognizes the two-dimensional and three-dimensional objects.

- The feature extracted is sufficient for recognizing the object and marking the location of the object. X The proposed classifier is able to recognize the object in less computational cost.
- The performance of the One-against-One classifier is efficient.
- Global feature extracted from the local parts of the image.
- Local feature PCA-SIFT is computed from the blobs detected by the Hessian-La place detector.
- Along with the local features, the width and height of the object computed through projection method is used.

As a scope for future enhancement,

- Features either the local or global used for recognition can be increased, to increase the efficiency of the object recognition system.
- Geometric properties of the image can be included in the feature vector for recognition.150
- Using unsupervised classifier instead of a supervised classifier for recognition of the object.
- Fully occluded object cannot be tracked and considered as a new object in the next frame.
- Foreground object extraction depends on the binary segmentation which is carried out by applying threshold techniques. So blob extraction and tracking depends on the threshold value.
- Splitting and merging cannot be handled very well in all conditions using the single camera due to the loss of information of a 3D object projection in 2D images.

References

[1] Agarwal S, Awan A., and Roth D. (2004), Learning to detect objects in images via a sparse, part-based representation. IEEE Trans. Pattern Anal. Mach. Intel. 26, 1475–1490.

[2] Aloimonos, J., Weiss, I., and Bandyopadhyay, A. (1988). Active vision. Int. J. Comput. Vis. 1, 333–356.

[3] Andreopoulos, A., and Tsotsos, J. K. (2013). 50 years of object recognition: directions forward. Comput. Vis.ImageUnderst.117, 827-891.

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, “You Only Look Once: Unified, Real-Time Object Detection”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.

[5] YOLO Juan Du,” Understanding of Object Detection Based on CNN Family”, New Research, and Development Center of Hisense, Qingdao 266071, China.

[6] Joseph Redmon, Ali Farhadi, “YOLO9000: Better, Faster, Stronger”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7263-7271.