International Journal of Research in Engineering, Science and Management
Volume-3, Issue-2, February-2020
www.ijresm.com | ISSN (Online): 2581-5792

Real Time Object Detection and Localization Using Deep Learning and OpenCV

T. Vineela[1], B. Naga Sowjanya[2], B. Raasi[3], Ch. Sree Ranjani[4], A. Durga Sri[5]
[1]*Assistant Professor, Department of Electronics and Communication Engineering, Vasireddy Venkatadri Institute of Technology, Guntur, India*
[2,3,4,5]*Student, Department of Electronics and Communication Engineering, Vasireddy Venkatadri Institute of Technology, Guntur, India*

***Abstract*: Deep Learning is a subset of Machine Learning. It teaches computer to learn how to predict and classify information, that may be images, text, audio etc. Deep Learning becomes more popular from the past few years in image classification tasks. Real time object detection and tracking is becoming simple by using deep learning. There are many deep learning algorithms for object detection there are CNN, R-CNN, fast R-CNN Faster R-CNN, SSD, YOLO. In our project we are using combination of SSD and MobileNet because it gives more performance, accuracy, speed compared to the other algorithms. The main objective of our project is to guide the visually impaired people by telling the what are the objects present and alerts the person when there are vehicles around them.**

***Keywords*: Object detection, CNN, SSD, MobileNet.**

1. Introduction

A few years ago, the creation of software and hardware image processing system was mainly limited to development of user interface. Image processing is used in various applications like face recognition, analyzing medical images, road signs counting people etc.

Object recognition is to describe a collection of computer vision tasks that involve activities like identifying objects in images. Image classification involves activities such as predicting the class of one object in image. Object localization is referring to identifying the location of one or more objects in an image and drawing bounding box around their extent. Object detection is estimating the class and location of objects in an image.

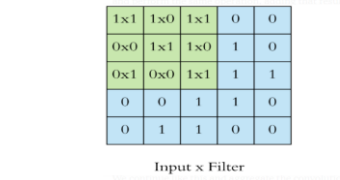
Human brain very powerful it recognizes multiple objects in an image within in second without any processing. But it is difficult to the machine to recognize the objects in an image without any training. The first step of image processing is, how to represent the image for understanding to the machine. For that we need to arrange a pixel of image in network that process is Convolutional Neural Network (CNN).

There are 3 basic components needed for convolutional neural network.

1. Convolutional Layer
2. Pooling Layer
3. Output Layer

A. Convolutional layer

In this layer we perform convolution to the input image and filter/kernel. Input size may be 5x5 or 6x6 and so on and kernel size may be 2x2 or 3x3 and so on. After convolution of the input image pixel values and coefficients of kernel gives a feature map.



B. Pooling Layer

In this layer the size of feature map is reduced by performing max pooling or average pooling and so on. Now we perform max pooling on feature map is shown in fig below.

429	505	686	856
261	792	412	640
633	653	851	751
608	913	713	657

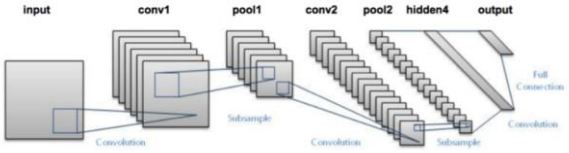
Feature map

792	856
913	851

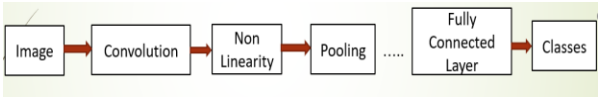
After Max pooling

C. Output Layer

After multiple layers of convolution and padding, we would need the output in the form of class. The convolution and pooling layers are able to extract features and reduce the number of parameters in original image. For getting the output we need to apply fully connected layers to generate output equal to the number of classes we need.



2. Block Diagram of CNN



Initially the input image is processed through the convolution block, which performs multiplication of input pixel values with the corresponding coefficients of the kernel and aggregate of that matrix gives corresponding output value in feature map.

The obtained feature map is given to the non-linearity circuit if there are any negative values in that map, they get replaced with zero by using ReLU function and obtained feature map becomes linearly separable.

At pooling layer, the size of the feature map gets reduced based on the stride value.

All the above four processes are repeated until the features of the image are fully extracted.

The objects in the image are classified in the fully connected layers.

There are many algorithms in deep learning object detection among those we are using Single Shot Multibox Detector, because it gives more performance, accuracy, it gives output within seconds in real time applications.

For running these algorithms, we are using tensor flow framework which is artificial intelligence library it uses model flow graphs to build models. It allows developers to create large scale neural network with many layers and OpenCV python library which is used to solve computer vision problems and it supports many programming languages.

A. Single Shot MultiBox Detector

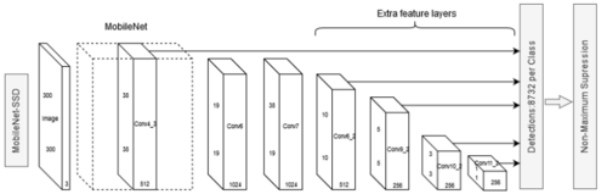
The performance of object detection in SSD scores over 74%mAP at 59 frames per second on standard dataset such as coco etc. Single Shot means object localization and classification are done in single forward pass of the network; the network simply “looks” once at the image. Detector is an object detector that classifies the detected objects. The SSD approach is based on a feed-forward convolution network that produces a fixed size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections.

SSD starts from core network, it is pre-trained on a large data set, such as ImageNet, which allows it to learn a rich set of different features. The core network is used to transfer training, spread the input image to a predetermined layer, obtain an object map, and then move this map forward to the object detection layers.

B. MobileNet

There are so many base layers like VGG or ResNet architecture, MobileNet architecture. but in those we are using MobileNet, because it is fastest and it gives more accuracy. MobileNetV1 is an architecture from google, which is known primarily for its smaller set of parameters and less network complexity, achieved by fewer addition and multiplication operations. MobileNet is a convolutional neural network architecture that applied on devices with limited computing power.

3. Architecture of MobileNet-SSD



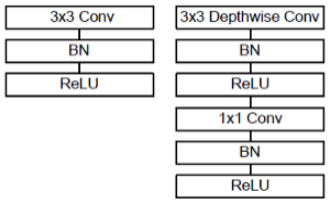
MobileNet-SSD architecture

SSD object detection composes of 2 parts:

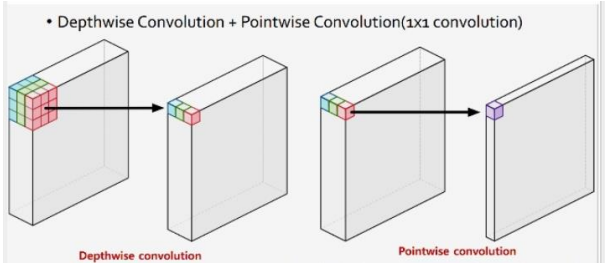
1. Extract feature maps
2. Apply Convolution filters to detect objects

In MobileNet-SSD architecture, we use MobileNet until conv_6 and then it consists additionally 6 auxiliary convolutional layers. In MobileNet it performs depth wise convolution followed by point wise convolution compared to standard convolution it multiplications are very less.

Depth wise convolution is the channel wise DK x DK spatial convolution. Point wise convolution actually is a 1 x 1 convolution to change the dimensions.



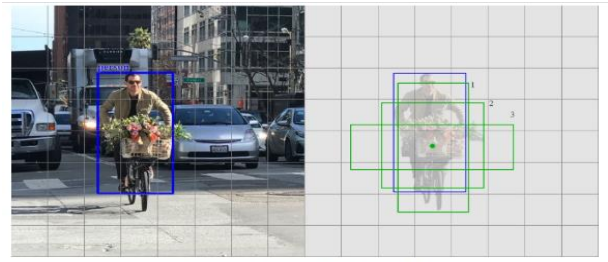
Standard Convolution (Left), Depthwise separable convolution (Right) With BN and ReLU



After performing depth separable convolution, we extracted feature maps. Those feature maps are given to the convolution filters to compute both location and class scores. For each cell it makes 4 object predictions. Each prediction composes of a boundary box and 21 scores for each class and we pick highest score as the class for bounded object. Conv4_3 makes a total a

total of 38 x38 x 4 predictions. Many predictions contain no object. SSD reserves a class "0" to indicate it has no objects.

SSD predictions are classified as positive matches or negative matches. SSD only uses positive matches in calculating the localization cost. If corresponding default boundary box has an IoU greater than 0.5 with ground truth, the match is positive. Otherwise, it is negative. IoU is intersection over the union it is ratio between the intersected area over the joined area for two regions.



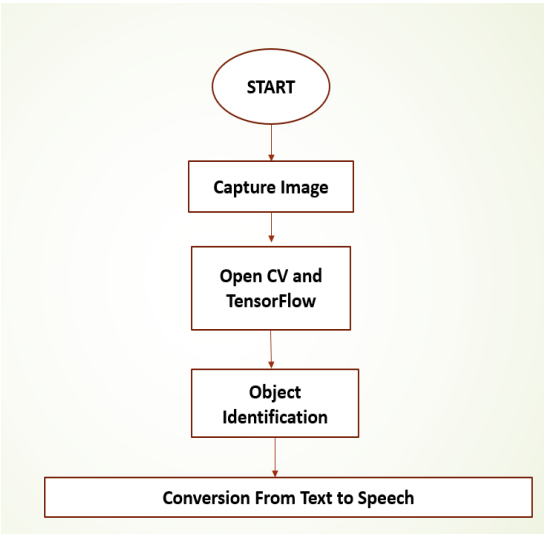
The ground truth object (blue) and 3 default boundary boxes (green).

In above fig there is only default box 1 and 2 have an Iou greater 0.5 with ground truth box above (blue box). So only box 1 and 2 are positive matches. Once we identify the positive matches, we use the corresponding predicted boundary boxes to calculate cost.

4. Non-Maximum Suppression (NMS)

Large number of boxes are generated during a forward pass of SSD at inference time, it is essential to prune most of bounding box by applying a technique known as non-maximum suppression: boxes with a confidence loss threshold less than ct (e.g.0.01) and IoU less than it (e.g. 0.30) are discarded and only top N predictions are kept. This ensures only most likely predictions are retained by the network, while the noisier ones are removed.

Flowchart:



Initially it captures a video by using webcam and by executing the python program which is trained by using combination of SSD and MobileNet on standard dataset coco under the environment of OpenCV and TensorFlow, we got objects present in video with labels and scores and bounding boxes surrounding the corresponding objects.

5. Results

The above figure shows real time object detection chairs, cell phone, bottle, laptop with confidence levels of 51%, 69%, 76%, 55%, 64% respectively. The model was trained to detect object class like car, truck, dog, cat, horse, book, bottle, laptop, bicycle, clock etc.

6. Conclusion

In this paper we detect objects and classify the objects by using combination of SSD and MobileNet accurately within seconds in real time object detection, but also it gives alerts message if there are any vehicles around them. Mainly it is very helpful for the visually impaired for detecting the objects around them and for their navigational purposes. It is also used in military applications, medical field, security surveillance, vehicle driver assistance etc.

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. Intell. 26,1475–1490.

[2] Alexe, B., Deselaers, T., and Ferrari, V. (2010). "What is an object?" in ComputerVision and Pattern Recognition (CVPR), 2010 IEEE Conference on (San Francisco, CA: IEEE), 73–80.

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

[4] Andrianopoulos, A., and Tsotsos, J. K. (2013). 50 years of object recognition: direc-tions forward. Comput. Vis. Image Underst. 117, 827–891.

[5] Andrianopoulos, A., and Tsotsos, J. K. (2013). 50 years of object recognition: directions forward. Comput. Vis. Image Underst. 117, 827–891.

[6] Azzopardi, G., and Petkov, N. (2013). Trainable cosfire filters for key point detection and pattern recognition. IEEE Trans. Pattern Anal. Mach. Intell. 35, 490–503.

[7] Azzopardi, G., and Petkov, N. (2014). Ventral-stream-like shape representation: from pixel intensity values to trainable object-selective cosfire models. Front. Comput. Neurosis. 8:80.

[8] Benbouzid, D., Busa-Fekete, R., and Kegl, B. (2012). "Fast classification using sparse decision dags," in Proceedings of the 29th International Conference on Machine Learning (ICML-12), ICML '12, eds J. Langford and J. Pineau (New York, NY: Omni press), 951–958.

[9] Bengio, Y. (2012). "Deep learning of representations for unsupervised and transfer learning," in ICML Unsupervised and Transfer Learning, Volume 27 of JMLR Proceedings, eds I. Guyon, G. Dror, V. Lemaire, G. W. Taylor, and D. L. Silver (Bellevue: JMLR.Org), 17–36.

[10] Bourdev, L. D., Maji, S., Brox, T., and Malik, J. (2010). "Detecting people using mutually consistent pose let activations," in Computer Vision – ECCV2010 – 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part VI, Volume 6316 of Lecture Notes in Computer Science, eds K. Daniilidis, P. Maragos, and N. Paragios (Heraklion: Springer), 168–181.