Real-Time Object Detection Using Pre-Trained Deep Learning Models MobileNet- SSD

Ayesha Younis
Department of Electronics Engineering,
Tianjin University of Technology and Education
300222, P.R China
+86-18526799742
ayesha@tute.edu.cn

Li Shixin
Department of Electronics Engineering,
Tianjin University of Technology and Education
300222, P.R China
+86-13920986298
li_shixin@sina.com

Shelembi JN
Department of Electronics Engineering,
Tianjin University of Technology and Education
300222, P.R China
+86-17622604244
shelembijn@gmail.com

Zhang Hai
Department of Electronics Engineering,
Tianjin University of Technology and Education
300222, P.R China
+86-15822775604
1459399409@qq.com

ABSTRACT

Mobile networks and binary neural networks are the most commonly used techniques for modern deep learning models to perform a variety of tasks on embedded systems. In this paper, we develop a technique to identify an object considering the deep learning pre-trained model MobileNet for Single Shot Multi-Box Detector (SSD). This algorithm is used for real-time detection, and for webcam feed to detect the purpose webcam which detects the object in a video stream. Therefore, we use an object detection module that can detect what is in the video stream. In order to implement the module, we combine the MobileNet and the SSD framework for a fast and efficient deep learning-based method of object detection. The main purpose of our research is to elaborate the accuracy of an object detection method SSD and the importance of pre-trained deep learning model MobileNet. The experimental results show that the Average Precision (AP) of the algorithm to detect different classes as car, person and chair is 99.76%, 97.76% and 71.07%, respectively. This improves the accuracy of behavior detection at a processing speed which is required for the real-time detection and the requirements of daily monitoring indoor and outdoor.

CCS Concepts

•Computing methodologies→Artificial intelligence→ Computer vision→Computer vision problems→Object detection

Keywords

Computer vision, Real-time Object Detection, Single Shot Detector, Deep Learning Neural Network

1. INTRODUCTION

Now a days**,** object detection is used globally in numerous fields such as, video surveillance, pedestrian displays, defamation detection, self-driving cars and appearance recognition. Within the Deep Learning field, the sub-discipline which is called Object Detection, includes an image such as a photo, video, or webcam feed. Since R-CNN predicted in 2014, built on DNN, object detection has grown tremendously. Finally, a set of upgraded approaches based on R-CNN as SPP-NET, Fast-RCN, Faster RCNN, and R-FCN appear in object detection methods (see Figure 1). Range of These methods attained high precision; however, a network structure is quite consisting of many elements in a complex relationship.

ICCDE '20, January 4–6, 2020, Sanya, China

ACM ISBN 978-1-4503-7673-0/20/01…$15.00
https://doi.org/10.1145/3379247.3379264

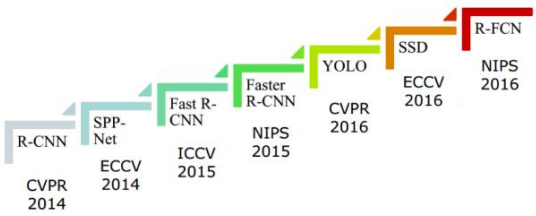

Figure 1. Object detector using deep learning

The complex DNN model for the detection of items can achieve high precision. But they required a large quota of calculation and configuration variables that is inside the model, which are not suitable for the embedded devices. According to this problem, the MobileNet was developed, which reduces the number of parameters and complexity of an algorithm to a number of resources demanded running by the embedded devices. The most crucial implication of the MobileNet is the usage of a separate deep configuration. There are two hyperparameters that are called width and resolution multiplier [1]. Shraddha and Supriya [2] emphasized the detection of a moving object by video order. Most of them followed a variety of techniques, and unlike the method, there are mixtures and contacts. Previously, there were some techniques involved in object detection, such as encode texture using the traditional approach of computer vision. [3, 4] shape [5, 6] and color [7].many Schemes were introduced to use motion and color information to detect with video orders, but these approaches cannot produce strong segmentation results [8]. Histogram of the Orient Gradients algorithm [9, 10] proved to be searching for objects in the image if the condition of the image has not been significantly altered. Han et al. [11] showed that the Support Vector Machine is highly compact and can be used to train highly precise objects such as the HOG algorithm.

To overcome these limitations, one of the simplest techniques in this approach is the SSD we are using with the DL (deep learning) Pre-Trained Model MobileNet through CNN development. In this way, the item is easily realized in complex video scenes and intricate background conditions.

First, recreate the background image with input frames. From this background image, we can imagine animation based on the motion information and the color indicator, which is based on the current color information in the frame. Then combine this motion and color indicator into the Markoff Random Field (MRF) framework to get an object that detects data from the background. The idea of background reduction is to cut the current image from a steel backdrop, which is assembled before being inserted into the item. After subtraction, only non-stationary or original items are left.

This technique is particularly suitable for video conferencing and surveillance application, where the background remains throughout the entire conference or monitoring period. Nevertheless, there is still a lot of madness as both the foreground and background colors look the same, changing the quality of the light, and the noise that makes us a simple way to isolate a video object. The use of change and limitation techniques is prohibited.

On the basis of survey, it was found that there has been a registered increase in the need to build compact and efficient neural networks [12, 13, 14, 15, 16].
Many diverse methods can usually be considered either compressing already trained networks or directly training minor networks. Two smooths global hyperparameters that are offered to trade properly between delay and accuracy which allow the model builder to choose the exact size of the MobileNet for their application [17]. However, the majority of papers on twisted networks only emphasize size but do not address speed [18].
In the detection of leaf diseases, a smart mobile application design which built on deep CNN to detect tomato leaf diseases. To develop the application, the model is based on the MobileNet CNN model, which is able to identify ten essential tomato leaf disease types. To accomplish the tomato leaves dataset for the development of the tomato disease diagnose application, the mobile application utilizes 7176 images of tomatoes. The MobileNet is primarily formed from the deep-removable ones introduced in [19] and then used in the incision model [20] to reduce calculation to the rare initial layers.

2. METHODOLOGY

SSD's have dual mechanisms: a spinal model and SSD head. The spinal model is generally a network of pre-trained image classifications as fact-makers. Here we usually use a network called MobileNet that trains over a million images that have been completely removed from the associated ranking layer. SSD Head This waist contains only one or more fixed layers, and the results are defined as classes of boxes and objects bound to the dimensional place of the closing layer activation. The first few layers (see Figure 2) are white cells. The spinal cord, the layers of blue cells signify the head of SSD.

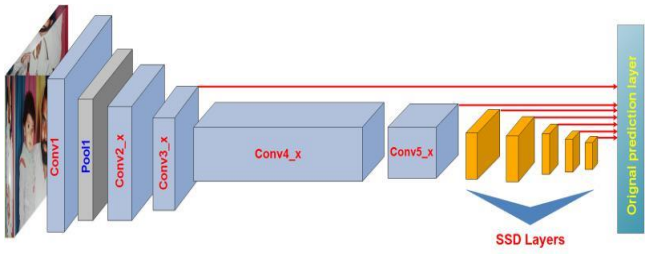

Figure 2. The architecture of CNN with an SSD detector [22]

2.1 Some Parameters in SSD

2.1.1 Grating/Grid Cell

Detecting objects means estimating the class and location of an element directly in this area. For example (see Figure 3) we use the 4x4 grid. Individually, the grating focuses on creating the space and the shape of the space that suits it. If the picture contains different elements in grating individually or wants to detect various things of unlike objects, the accessible field called Anchor box is included to complete this part.

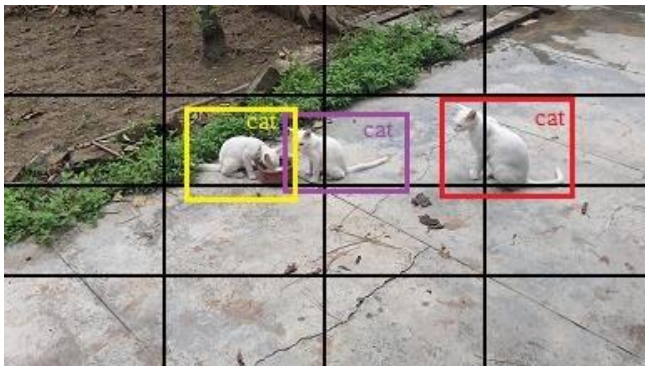

Figure 3. Example of a 4x4 grid and three anchor boxes

2.1.2 Anchor Box

The grid cells individually allocated to SSD with different anchors or prefixes. In each grid cell, these anchor boxes can control any shape and size. The cats (see Figure 3) corresponds to different anchor boxes, one high anchor box, while the other is wider, hence different sizes of the anchor boxes. These anchor boxes with an abundance of intersection through an object will finalize the class and its place of that object. This stuff is used for training the network and for predicting the detected object and its location after the network has been accomplished.

2.1.3 Zoom Level

Anchor boxes do not have to be the same size as the grid cells. It is used to identify with grid cells to what degree of the anchor box wants posterity individually upward or downward.

2.1.4 Aspect Ratio

Some objects in shape are wider whereas some are longer (see Figure 3) with different grades. The SSD framework permits the aspect ratio of anchor boxes towards it. The range of proportions is used to describe the variable aspect ratios of anchor box links.

2.1.5 Receptive Field

The acceptable field input area is separated as a region of viewed by a particular CNN. Zeiler and Rob [21] used the distinctive attribute and an actuate to present them as a back-line combination at the relative location. Due to the compromised operation, the properties of different layers indicate the various size of region in the image. Place (See Figure 4) the lowest layer (5x5) and then a console, which will result in a central layer (3x3) in which a single green pixel represents a 3x3 region of the input layer (bottom layer). The convolution is then applied to the middle layer (green), having the upper red layer (2x2) where the individual attribute equals the 7x7 area of the input image. This green and a red 2D array are referred to as the feature map, which in the form of an indicator window points to a set of features created using a similar feature extractor at different points of the input map. Similar map features have a similar field, which in return tries to identify similar patterns in different positions. Hence a local level of Convolutional Network is created.

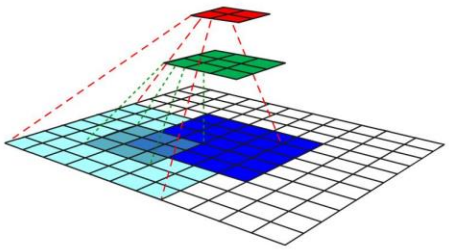

Figure 4. Visualizing CNN feature maps and the receptive field

2.2 Development of Algorithm

We try to do object detection using OpenCV library and deep learning pre-trained models it's almost similar to real-time face recognition. First we train the system using a familiar faces or reference faces in case if the face appears any of the image or the video feeding in the system, which will recognize that face. In this paper, which are dealing with object detection we cannot predict the number of objects or predict the objects such as a car, people and cat. If we have possible images of a car to train a system, then a system can predict these objects from the image or video. But it's not practically possible because there are plenty of objects around us. We relayed some pre-trained models. These pre-trained models have been trained by some third-party person. Most of the objects already pre-trained in these models. Finally, System is ready to detecting objects using pre-trained models with SSD method. We use pre-trained models MobileNets to implement with the SSD method in python code. This model can classify labels on the bases of training data and a set of bounding box colors for individual classes. Load the input video (frame by frame) and make it an input drop for a single frame by resizing each frame with a fixed size (300x300) pixels.

MobileNet method is utilized to expand the SSD algorithm and speed rating accuracy on a real-time basis. This approach requires taking a single shot to detect multiple objects. The SSD is a neural network architecture design for detection purposes. This means localization and classification occurring at the same time but other methods such as the R-CNN series require two shots, SSD technique detaches the output space of bounding into a set of default boxes over dissimilar fact ratios and scales. SSD reveals the banned output space in the default box set. The network rapidly scans the presence of individual object classes in a default box and unites the box to fit what is inside it. Also, this network fits many models with different sizes of natural adhesives with different resolutions. If no item is present, it is considered a background and the location is ignored.

2.3 Detection Contains Three Steps

1-Using OpenCV's deep neural network (DNN) module to load a pre-trained object detection network.

2-Set of input to the network and compute the forward pass for the input. Storing the result as Detections.

3-Then, loop through the Detection and determine what and where the objects are in the images.

Besides, the place where a particular object is in the frame (see Figure 6) a person and the place and then the accuracy of the detection. A Person image and then the label for that particular detection it's already there in the pre-trained model. A label called a person (see Figure 6) then the accuracy of the label. So these three steps retrained and used a loop through these three steps and drawing a bounding box around that particular object in the frame.

2.3.1 Model files

These are the files of our pre-trained models, one is configuration, and the second is the weights. So, the model is actually how neurons are arranged in a neural network.

1- Configuration
2- Weight

3. EXPERIMENTAL RESULTS

The algorithm of this object detection is up to 14 fps, so low-quality cameras of any fps can produce good results. In this case, we consider a 6 fps webcam. In our experiments, the SSD algorithm demonstration indoor and outdoor feed video frames via webcam but the position of the objects between two consecutive frames is different. The video captured by webcam and the algorithm convert the size of a single frame that is considered to be 300 × 300. SSD can detecting objects frame by frame with the accuracy of a class label with creating the bounding box around the detected object. The obtained results from this procedure on the homemade video's frame are shown in the Figures 5.

Input frame of a video sequence detection is a TV monitor (see Figure 6) with a confidence level of 76.46% and a person with the level of 97.86% (i.e., probability), although the full face of the person is not shown, CNN has a highly accurate detection algorithm for human characteristics.

The SSD can produce multiple bounding boxes for different classes with a different confidence level (see Figure 7) using a higher proportion of default boxes that can have a better effect, where different boxes are used for each location. This proposed method of single-shot multi-box detection is based on frame difference (see Figure 8). Frames analyzed the effectiveness of the proposed method. The detection results in foggy weather conditions (see Figure 8) verified the accuracy and sturdiness of the proposed method.

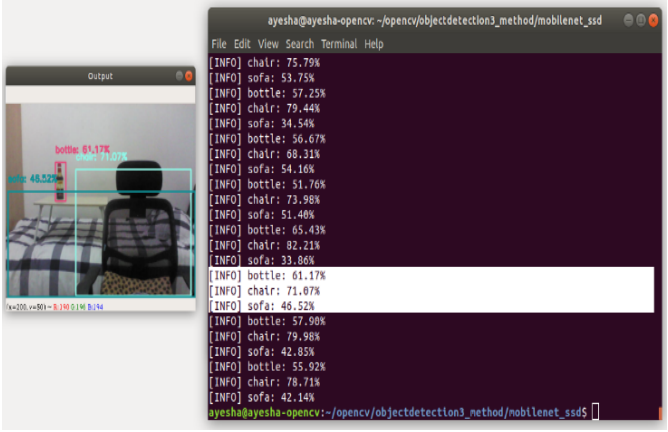

Figure 5. A window appears on the screen

After executing the program and inputting the video, the following window appears on the computer screen (see Figure 5).

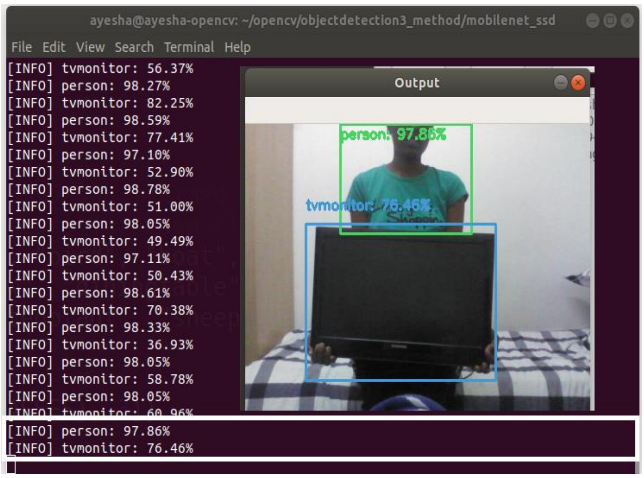

Figure 6. System detecting a person and TV monitor

Figure 7. SSD multiple bounding boxes for different classes with different confidence level

Figure 8. Car detection from video frame sequence

4. CONCLUSIONS

A high accuracy object detection procedure has been achieved by using the MobileNet and the SSD detector for an object detection, which can push a processing speed to 14 fps, and make it efficient to all camera that can often process at only 6 fps. This system can detect the items within its dataset, such as a car, bicycle, bottle, chair, etc. The dataset can be expanded by adding an unlimited number of items through using images, referred to as deep learning technology. We used Ubuntu 18.04.2, OpenCV 3.4.2 and Python programming language for the experiment of SSD algorithm. The goal of this research is to develop an autonomous system where the recognition of objects and scenes helps the community to make the system interactive and attractive. For future work, this work will be primarily deployed to identify the item with better features in the external environment.

5. ACKNOWLEDGMENTS

The authors are special thanks to Editor in chief and anonymous referees for the valuable comments and suggestions. This work was partially supported by the Tianjin science and technology (19JCTPJC54800) and Tianjin graduate research (2019YJSS194).

6. REFERENCES

[1] Hong, Y. C., Chung, Y. S, *An Enhanced Hybrid MobileNet 2018 9th International Conference on Awareness Science and Technology (iCAST)*

[2] Shraddha, M., Supriya, M. Moving object detection and tracking using convolutional neural networks *IEEE Xplore* ISBN:978-1-5386-2842-3

[3] Ojala. T., Matti. P., Topi, M. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis & Machine Intelligence, no. 7, pp. 971–987, 2002.

[4] Haralick R.M., Karthikeyan. S., Hak. D. Textural features for image classification, IEEE Transactions on systems, man, and cybernetics, no. 6, pp. 610–621, 1973.

[5] Hu. M.K., Visual pattern recognition by moment invariants, IRE transactions on information theory, vol. 8, no. 2, pp. 179–187, 1962.

[6] Khotanzad. A., Yaw. H. H. H. Invariant image recognition by Zernike moments. IEEE Transactions on pattern analysis and machine intelligence, vol. 12, no. 5, pp. 489–497, 1990.

[7] Huang. J, Kumar S. R., Mitra. M, Zhu. W. J., Zabih. R. Image indexing using color correlograms. in cvpr. IEEE, 1997, p. 762.

[8] Prajakta, A. P., Prachi, A. D. Moving Object Extraction Based on Background Reconstruction. International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 4, April 2015

[9] Dalal, N., Triggs, B. Histograms of oriented gradients for human detection. An International conference on computer vision & Pattern Recognition (CVPR'05), vol. 1. IEEE Computer Society, 2005, pp. 886–893.

[10] Rosebrock, A. Deep Learning for Computer Vision with Python: Starter Bundle. Pyimagesearch, 2017.

[11] Han. F., Shan, Y., Cekander R., Sawhney, H.S., Kumar R, A two-stage approach to people and vehicle detection with hog-based SVM in Performance Metrics for Intelligent Systems 2006 Workshop, 2006, pp. 133–140.

[12] Jin, J., Dundar, A., Culurciello, E. Flattened convolutional neural networks for feedforward acceleration. arXiv preprint arXiv:1412.5474, 2014

[13] Wang, M., Liu, B., Foroosh, H. Factorized convolutional neural networks. arXiv preprint arXiv:1608.04337, 2016.

[14] Iandola, F.N., Moskewicz, M.W., Ashraf, K., Han, S., Dally, W. J., Keutzer, K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 1mb model size. arXiv preprint arXiv:1602.07360, 2016.

[15] Wu, J., Leng, C., Wang, Y., Hu, Q., J. Cheng, J. Quantized convolutional neural networks for mobile devices. arXiv preprint arXiv:1512.06473, 2015.

[16] Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A. Xnornet: Imagenet classification using binary convolutional neural networks. arXiv preprint arXiv:1603.05279, 2016.

[17] Howar, A.G., Zhu,M., Che, b., Kalenichenko, D., Wang, W., Wey, T., Andreetto, M., Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications *arXiv*: 1704.048861

[18] Azeddine Elhassouny, *Florentin Smarandache* , " Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks" IEEE/ICCSRE2019, 22-24 July, 2019, Agadir, Morocco.

[19] Sifre, L., Rigid-motion scattering for image classification. PhD thesis, Ph. D. thesis, 2014.

[20] Ioffe, S., Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[21] Zeiler, M. D., Rob, F. "Visualizing and understanding convolutional networks." In European conference on computer vision, pp. 818-833. springer, Cham, 2014.

[22] Liu, W (October 2016). SSD: Single shot Multi-Box detector. European Conference on Computer Vision. Lecture Notes in Computer Science. 9905. *arXiv:1512.02325.*