International Journal of Academic Research and Development
ISSN: 2455-4197
Impact Factor: RJIF 5.22
www.academicsjournal.com
Volume 3; Issue 2; March 2018; Page No. 1311-1316

Deep learning based detection and recognition of objects using mobile nets and SSDs

T Naga Lakshmi [1], N Janaki [2]
[1] Assistant Professor, Department of C.S.E, AITS, Rajampet, Jntua, Andhra Pradesh India
[2] Assistant Professor, Department of C.S.E, SITAMS, Jntua, Andhra Pradesh, India

Abstract
Object detection and recognition is a sub filed of computer vision that is greatly based on machine learning. From the past few decades, it is observed that, the field of machine learning has been overwhelmed by deep neural networks called as CNN, i.e. Convolution Neural Networks. Computational power and the availability of the data are the two important characteristics to made CNN as powerful and such kind of neural network is well appropriate for image processing application for example object detection and recognition. The model is trained by identifying numerous features such as pixel positions, corners, hues, edges in the image and merge these features in to more complex shapes. To detect the objects, the model must compute the locations of the objects and classifies accordingly. There exists several methods to detect the objects using deep neural networks module in deep learning and works based on a pre-trained networks via Caffe, Tensor Flow and Torch/Py Torch for image classification. The work is comprised by combining Mobile Nets and Single Shot Detectors (SSDs) and more efficient and fast to detect the objects using deep learning. Our experimental results shows the better results compared with other methods for example Faster R-CNN, YOLO, and SSDs.

Keywords: object detection, CNN, F-CNN, Mobile Nets, SSDs

1. Introduction
Detecting and Tracking of objects in image plays a vital part in many computer vision applications for example in image and video classification. Detecting the object in the image denotes that identifying the correct patterns present in the image or a pattern in a specific frame in a particular video. A technique is always required to track the object in an image or video. There exists numerous techniques for this purpose, which use the temporal information and this information can be further analyzed in a sequence of frames to minimize the false detections to maximize the accuracy of detection and recognition [1]. There are several challenges exists in object recognition. Although there exists various algorithms and techniques to recognize the objects present in the images, the following conditions always poses several challenges to these algorithms. Lighting conditions may be different throughout the day along with weather conditions may also affect the lighting in an image. For example, image captured at In-door and out-door may have different lighting conditions. Shadows in the image also affects the quality and difficult to recognize the object. Position of the objects in an image also poses challenge to recognition and classification tasks. The image may be rotated in different angles in some cases and difficult for the object recognition algorithms and mirrored images also able to recognize. Several conditions are there for an object in the image to be occluded. Size of the object in an image also affects the result. The same objects may of small, medium, large but the algorithm has to detect the objects accurately. Efficient and robust algorithms are required for object detections with the above stated issues.

CNN has been generally utilized as a part of visual recognition since 2013 because of its high accurate detection and classifying images. It is similar to traditional neural networks, in which it contains one or two hidden layers, but in CNN it may have many number of hidden layers. The traditional neural networks is shown in Figure 1. The authors of [2] proposed improved methods to detect the accurate results using ILSVRC data set and CNN became a most important and critical challenge among all the other research domains. Kept apart, CNNs also have other applications such as localization [3], image/video segmentation [4], caption generations [5], object detection [6] etc. The computation of activation function in traditional neural network is shown in Figure 2. This task is almost similar in CNN except the number of hidden layer in between the input and output layers.

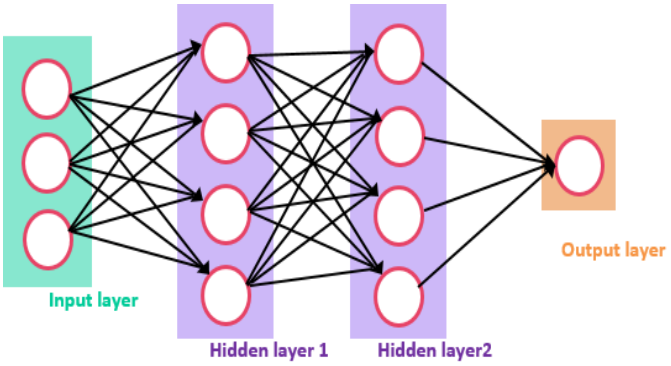


Fig 1: Structure of Neural Networks.

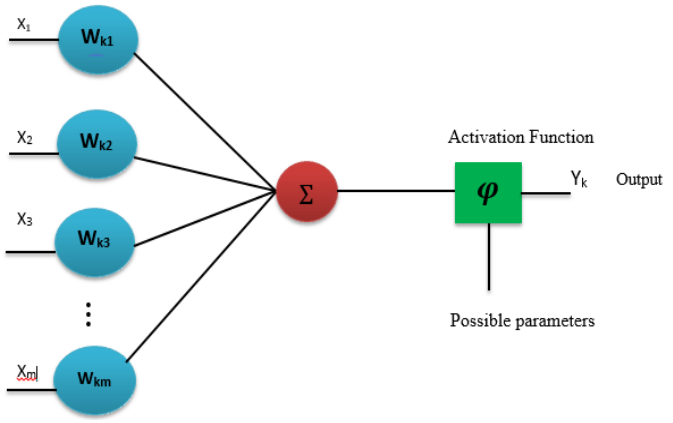


Fig 2: Computation of Activation Function

2. Background Work

Videos are usually made with the sequences of images, solely of which is known as a frame, and are displayed with speed enough frequency and hence human eyes can capture the content's flow. It is clear that every technique of image processing can be applied on frames individually. In addition to this, the two consecutive frame's has closely related content usually [7]. Detection of interested domains is typically said as initial step in most of the computer vision applications including event detection, video surveillance and robotics. It is necessary to have a general algorithm of object detection, but there is complexity in proper handling of remote objects or objects with required variations in color, shape and texture. Hence most of practical computer vision systems consider a static camera environment that makes the process for object detection much easier [8]. An image from a video sequence is partitioned in two complimentary sets of pixels. The corresponding pixels of foreground objects relate to first set whereas the background pixels relates to complimentary set. This outcome is usually represented as a binary image or a mask. It is not so easy to state an exact standard related with what is to be decided as foreground and what should be taken as background because this definition is partially specified in application. In general, like human, vehicles else foreground objects are movable and everything else is background [10]. In most of the times shadow is considered as a foreground object that yields improper output. In the literature, the following are the fundamental steps in tracking an object.

2.1 Object Detection

Object Detection is to find interest objects in a video sequence and to integrate these objects' pixels. This Object Detection can be carried out by different techniques like frame differencing, Optical flow and Background subtraction. The initial step in object tracking is to find interest objects in a video sequence and to integrate these objects' pixels. As movable objects are the key source of information, many methods concentrate on the identification of such objects.

2.2 Object Classification

Object can be categorized as vehicles, swaying tree, birds, floating clouds and any other movable objects. The approaches for objects classification are Shape-based classification, Motion-based classification, Color based classification and texture based classification.

2.3 Object Tracking

The problem in predicting the object path in an image plane as it moves around a scene is called as Tracking. For tracking the objects there exist the approaches like point tracking, kernel tracking and silhouette. While tracking an object, few of the challenges need to be taken into sight and those are mentioned in below similar to the description in [9]. 1. Loss of evidence caused by estimate of the 3D realm on a 2D image, 2. Noise in an image, 3. Difficult object motion, 4. Imperfect and entire object occlusions, 5. Complex objects structures. The significance of tracking an object is to provide the way by identifying the position of that object in every individual frame of the video [11]. The purpose of tracking an object is to extract an object, recognize and object and decisions regarding activities. According to paper [9], Object tracking can be classified as point tracking, kernel based tracking and silhouette based tracking. To illustrate this, the point trackers include detection in every frame; while geometric area or kernel based tracking or contours-based tracking needs detection at the time of displaying the object in the scene for the first time.

3. Related Work

The recent advancements and rise of autonomous vehicles, video surveillance, and face detection there is huge demand in developing fast and accurate object detection systems. The developed systems are not only recognizing and classifying the class of that objects, it localize the object by drawing a box around the object which is recognized. This makes question that object detection becomes a hard task than the conventional computer vision and its predecessors and image classification algorithms actually does [14]. Luckily, be that as it may, the best way to deal the detections is use the extensions of classification models. A couple of months prior, Google developed an API for object detection TensorFlow and it is one among the most popular models. The other models include Single Shot Multi box Detector SSD with MobileNets, SSD with V2, R- FCN, and Faster RCNN etc. We will briefly discuss few of these models in this section.

3.1 Convolution Neural Networks

The task of taking an input image and getting a related class of the object i.e. a cat, chair, person etc. called as Image classification. For a human beings recognition is quite easy since we learn from childhood and that comes naturally and effortless as adults. Without even thinking twice, we're able to quickly and seamlessly identify the environment we are in as well as the objects that surround us. When we see an image or just when we look at the world around us, most of the time we are able to immediately characterize the scene and give each object a label, all without even consciously noticing [13]. These skills of being able to quickly recognize patterns, generalize from prior knowledge, and adapt to different image environments are ones that we do not share with our fellow machines.

3.2 Inputs and Outputs

When a computer sees an image (takes an image as input), it

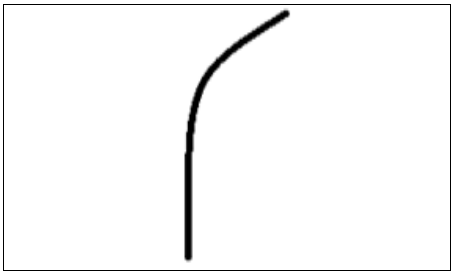
will see an array of pixel values. Depending on the resolution and size of the image, it will see a 32 x 32 x 3 array of numbers (The 3 refers to RGB values). Just to drive home the point, let's say we have a color image in JPG form and its size is 480 x 480. The representative array will be 480 x 480 x 3. Each of these numbers is given a value from 0 to 255 which describes the pixel intensity at that point. These numbers, while meaningless to us when we perform image classification, are the only inputs available to the computer. The idea is that you give the computer this array of numbers and it will output numbers that describe the probability of the image being a certain class (.80 for cat, 15 for dog, 05 for bird, etc.) [15].

3.3 Structure of CNN

Convolution Neural Networks takes an image as input, pass it through a series on convolutions, nonlinear, pooling and fully connected layers and gets corresponding output. In this process, the input is a single image and output may be a single class or probability percentage of the class that the object best describes about the image. The primary layer in CNN is actually called Convolutional Layer. The input for this convolutional layer is the image in the form of RxCxD. Here R, C, D represents the number of rows, columns, dimensions respectively. For example 256x256x3 denotes the size of the image is 256x256 and contains three colors channels, i.e. red, green, and blue. In other words, the input for the convolutional layer is array of pixels. The depth of filter or kernel is always same as the depth of the image. The process of convolution is now similar to the normal convolution operation. We have shown the First layer in high level perspective in Figure 3. Figure 3.a. shows the pixel representation of a kernel and Figure 3.b. is visualization of a curve detector filter. We have given an example of convolution process for a sample image and the resultant value of the convolution of these two images is 6600 (it includes the multiplications and summations). The result of convolution process of the image shown in Figure 3.c with the filter shown in Figure 3.a. is 6600. ((50*30) + (50*30) + (50*30) + (20*30) + (50*30)).

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

(a) filter



(b) Visualized curve filter

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

(c) Original image

Fig 3: Pixel representation of filter with original image

3.4 R- CNN

R-CNN abbreviated as Region based Convolutional Neural Network, which comprises three steps. Firstly, it scans the input image and find the possible objects using Selective Search algorithm (assume ~1000 regions). In other words, it finds the appropriate regions. Secondly, we will run a CNN model on the top of these regions, obtained from the previous step. Finally consider the output of each CNN and feed them in to either SVM or to Linear Regression method to tighten the bounding box of the object, if any such kind of object present in the image. These three steps are shown in Figure 4. In other words, we first collect regions, classify those regions based on the features and we will give it to a classification algorithm. R-CNN is very straight forward but it is too slow.

3.5 Fast R-CNN

The immediate descendant of R-CNN was Fast R-CNN. This is an improved method over R-CNN in detection speed. It performs the feature extraction on the image before getting the regions unlike in R-CNN and it replace SVM with a softmax layer. It extends the neural network predictions rather than creating a new model.

3.6 Faster R-CNN

Faster R-CNN replaces the Slower Selective Search algorithm with a fast Neural net. This method introduces the region proposal network (RPN). At each location of our feature map we will consider the k-different boxes centered on it. It may be a tall, side, large. We will loop over each box and check whether this box contains our expected object. This location is actually called as sliding window location. We have shown the comparison of these three models in Table 1.

Table 1: Comparison of Three Convolution Models

	R-CNN	Fast R-CNN	Faster R-CNN
Test time	50 s	2 s	0.2 s
Speed	1x	25x	250x
mAp	66.0%	66.9%	66.9%

4. Proposed Work

The main goal of our work is to detect and recognize the objects present in the image. We are using a pre-trained image model with the help of deep neural network module from Open CV version 3.3. This newly introduced module contains different algorithms and support numerous deep learning frameworks includes Caffe, PyTorch, TensorFlow etc. The work is summarized in three steps. First read the input image from the disk or from the network. Pre-process the image to

reduce the noise or for improving the quality of the input image. As a third step, pass this image through the network and get the classified image. The sample convolution process is shown in Figure 4. This model categorize the image but it could not locate the object position in the image. To detect the location of the image, we have applied the object detection algorithm in our work to detect where the object is actually resides in the image. We obtained the bounding box co-ordinates assume (x, y) for an object in image by combining Single Shot detectors (SSD) and MobileNets. All the images are passed through a network and we get bounding box co-ordinates of each object i.e. classified image as output. There exists various deep learning based object detection methods such as Faster R-CNN, YOLO and SSDs. The Faster R-CNN is most popular method for object detection and it is hard to understand for the beginners to implement. The task of training phase is also a challenging in Faster R-CNN. The faster implementation of the regions in fast R-CNN is a little slow on the order of 7 FPS. YOLO is another algorithm, it is much faster on the order of 40-90 FPS on GPU. Another advanced model in this called, faster YOLO is capable of 155 FPS also. The YOLO is less accurate. In our work we opted SSDs, originally developed by Google Corporation, in which it balances the speed and accuracy. SSD is faster than fast R-CNN and it is straight forward. This method is based on a feed-forward convolutional model and produces a fixed size 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 discretizes the output by the bounding boxes in to a group of default boxes over numerous aspect ratios and scales per feature map location. While prediction, for each object category a default score is generated and in further iterations few adjustments are made to get the better match based on the object shape. Multiple number of feature maps with numerous resolutions handles the objects with different size. These models are easy to train and we can directly incorporate these systems and which requires a component for detection.

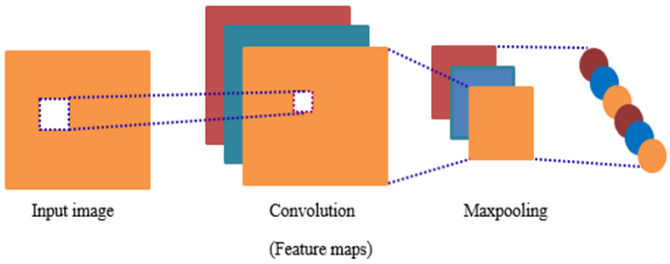


Fig 4: Convolution Process

When developing object detection networks, a network architecture is used. For example VGG network can be used inside the object detection pipeline. The size of these networks can be vary in between 150-500 MB and hence not suitable for the memory constrained applications. We used MobileNets by Google, because they are well suited for resource constrained devices for example smartphones. These are different compared to convolution networks since the convolution process is based on depth. We have shown the usage of MobileNet in the object detection process in Figure 5. The idea of depth wise convolution is nothing but, splitting the convolution process in to two stages. 3X3 Convolution followed by 1X1 point convolution.

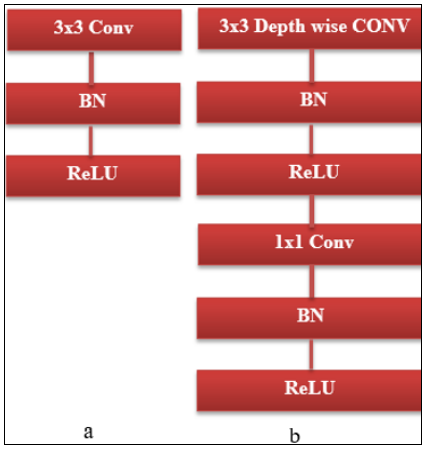


Fig 5: (a) Standard Convolution Layer. **(b)** Depth wise Convolution Layer.

5. Results and Discussion

Our idea is the combination of MobileNet and SSD framework gives a fast and better results in the view of deep learning based object detection. Our work is implemented by using MobileNet SSD along with deep neural network library of OpenCV module. The process begins with the input image. The image is pre-processed first and will be given to trained model. As a first step, we have to prepare a model from the set of images. The MobileNet was trained with COCO (Common Objects in the Context) Dataset. This trained model contains approximately 2000 number of classes of objects. This input will processed through our trained model and the objects present in the image are classified with their class name along with bounding boxes. We took different images as input and applied the SSD network model and observed results. We can detect around 15-20 objects in an image includes airplanes, birds, people, sofas, trains, cars, cats etc. We consider few input images and the results object detection are shown in Figure 6. This method require four parameters. One is Input image, Caffe model text file, the trained model and the confidence (20% is used by default). Along with our study, in some cases our work is giving the result less accurate of the objects with complete white background. The results are shown in Figure 7. Our method gives the results correctly in Figure 7.b for the input image, and in correctly classified result which is highlighted in Figure 7.c.

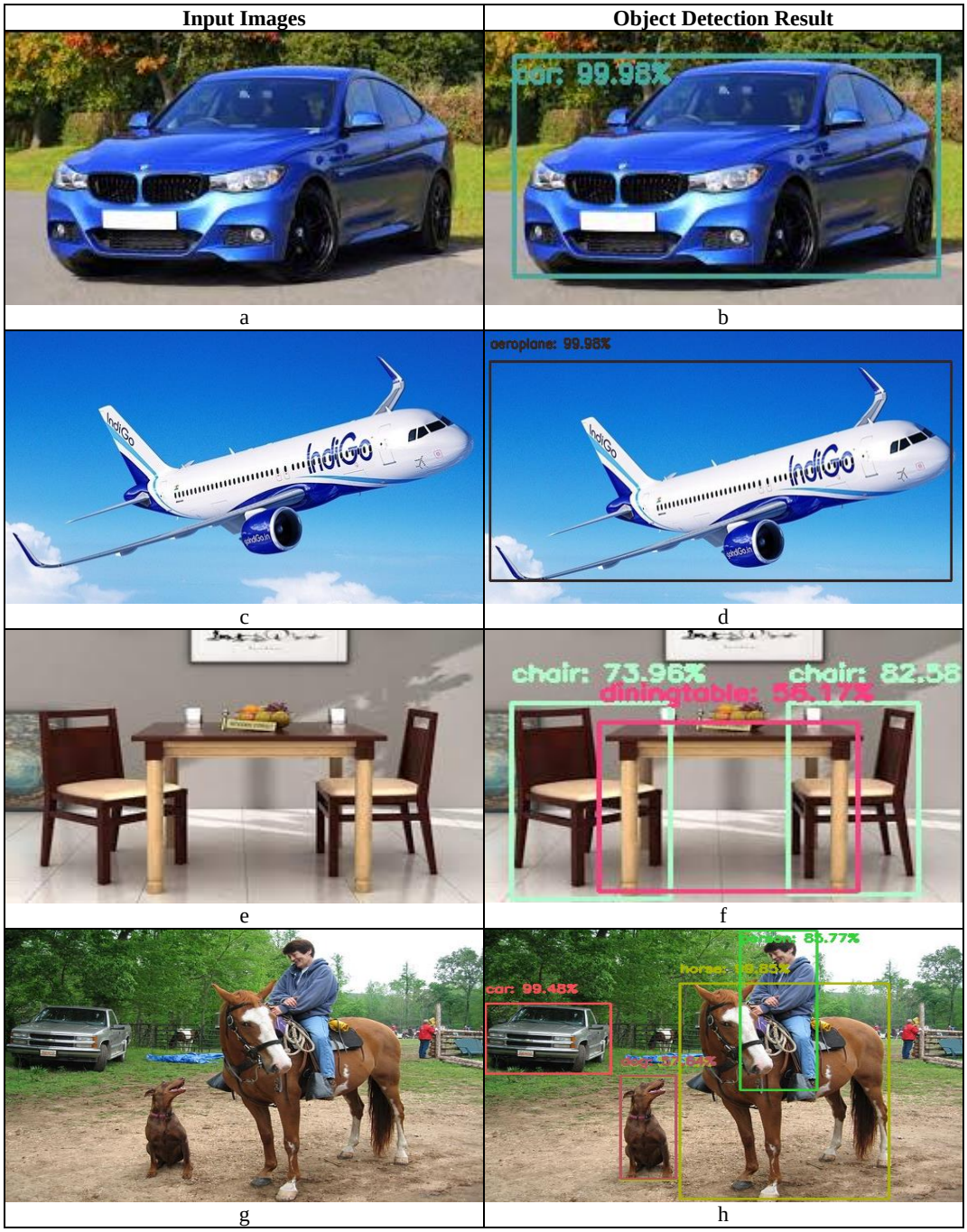


Fig 6: Object classification for different input images

a **b** **c**

Fig 7: (a) Input Image. **(b)** Correctlry classfied result. **(c)** Incorrectly classified result.

6. Conclusion

Object detection and recognition is a sub filed of computer vision that is greatly based on machine learning. It plays an important role in many computer vision and pattern recognition applications such as video classification, vehicle navigation, surveillance and autonomous robot routing. Convolutional neural networks (CNNs) has been widely used in visual recognition from 2012 due to its high capability in correctly classifying images. In this paper, we concentrated on detecting and classifying the objects present in the images using deep learning using computer vision model. We used MobileNets and SSDs along with deep neural network model to detect the objects in the image. We conducted experiments by considering different real time images captured in out-door environment and our model is giving good results less than the computational time taken by other state of the art algorithms. Although our work is giving the result faster than Fast R-CNN, the model is less accurate in some cases as described. Increasing the accuracy and applying the same model to the images captured in different environmental conditions and applying this model to the real-time video streams will be considered as our future work.

7. References

1. Himani S, Parekh1, Darshak G. Thakore Udesang K. A Survey on Object Detection and Tracking Methods, International Journal of Innovative Research in Computer and Communication Engineering. 2014; 2(2).
2. Krizhevsky A, Sutskever I, Hinton G. Image Net classification with deep convolutional neural networks. In NIPS. 2012; 1(3):4-7.
3. Szegedy C, Toshev A Erhan D. Deep neural networks for object detection. In NIPS, 2013, 2.
4. Jonathan L, Evan S, Trevor D. Fully convolutional networks for semantic segmentation, to appear in CVPR, 2015.
5. Andrej K. Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions.
6. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation in Proc. CVPR, 2014.
7. Abhishek Kumar Chauhan, Prashant Krishan. Moving Object Tracking Using Gaussian Mixture Model And Optical Flow, International Journal of Advanced Research in Computer Science and Software Engineering, 2013.
8. Cheng-Laing Lai, Kai-Wei Lin. Automatic path modeling by image processing techniques, Machine Learning and Cybernetics (ICMLC), 2010 International Conference on. 2010; 5:2589-2594.
9. Joshan Athanesious J, Suresh P. Systematic Survey on Object Tracking Methods in Video, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2012, 242-247.
10. Sen-Ching S. Cheung Chandrika Kamath, Robust techniques for background subtraction in urban traffic video.
11. Ruolin Zhang, Jian Ding. Object Tracking and Detecting Based on Adaptive Background Subtraction, International Workshop on Information and Electronics Engineering, 2012, 1351-1355.
12. Object detection with deep learning and OpenCV by Adrian Rose Brock on in Deep Learning, OpenCV 3, Tutorials, 2017.
13. Springenberg JT, Dosovitskiy A, Brox T, Riedmiller MA. Striving for simplicity: The all convolutional net. CoRR abs/1412.6806, 2014.
14. Uijlings JRR, van de Sande KEA, Gevers T, Smeulders AWM. Selective search for object recognition. International Journal of Computer Vision. 2013; 104(2): 154-171.
15. Yang B, Yan J, Lei Z, Li SZ. Craft objects from images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, 6043-6051.