ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2019.0301

DISCOVERY OF COMPOUND OBJECTS IN TRAFFIC SCENES IMAGES WITH A CNN CENTERED CONTEXT USING OPEN CV

G. Arun Sampaul Thomas and G. Manisha
Department of Computer Science and Engineering, J.B. Institute of Engineering and Technology, India

Abstract

Vision based traffic scene perception (TSP) is one of many fast-emerging areas in the intelligent transportation system. This field of research has been actively studied over the past decade. TSP involves three phases: detection, recognition and tracking of various objects of interest. Since recognition and tracking often rely on the results from detection, the ability to detect objects of interest effectively plays a crucial role in TSP. The aim of traffic sign detection is to alert the driver of the changed traffic conditions. The task is to accurately localize and recognize road signs in various traffic environments. Prior approaches use colorant shape information. However, these approaches are not adaptive under severe weather and lighting conditions. Additionally, appearance of traffic signs can physically change over time, due to the weather and damage caused by accidents. Instead of using color and shape features, most recent approaches employ texture or gradient features, such as local binary patterns and histogram of oriented gradients. These features are partially invariant to image distortion and illumination change, but they are still unable to handle severe deformations.

Keywords:

Object Detection, CNN, Traffic Scenes Images, Traffic Sign Detection, Image Identification

1. INTRODUCTION

1.1 GENERIC OBJECT DETECTION

Object detection is a challenging but important application in the computer vision community. It has achieved successful outcomes in many practical applications such as face detection and pedestrian detection complete survey of object detection can be found in. This section briefly reviews several generic object detection methods. One of the viable classical object detector is the detection framework of Viola and Jones, which uses a sliding-window search with a cascade classifier to achieve accurate location and efficient classification. The other commonly used framework is a linear Support Vector Machine (SVM) classifier with histogram of oriented gradients (HOG) features and that has been applied successfully in pedestrian detection. These frameworks achieve excellent detection results on rigid object classes. However, for object classes with a large intra-class variation, their detection performance falls down dramatically. In order to deal with appearance variations in object detection, a deformable parts model (DPM) based method has been proposed in. This method relies on a variant of HOG features and window template matching, but explicitly models deformations using a latent SVM classifier. It has been applied successfully in many object detection applications. In addition to the DPM, visual sub categorization is another common approach to improve the generalization performance of detection model. It divides the entire object class into multiple subclasses such that objects with similar visual appearance are grouped together. A sub-detector is trained for each subclass and detection results from all sub detectors are merged to generate the final results.

1.2 TRAFFIC SIGN DETECTION

Many traffic sign detectors have been proposed over the last decade with newly created challenging benchmarks. Interested reader should see which provide a detailed analysis on the recent progress in the field of traffic sign detection. Most existing traffic sign detectors are appearance-based detectors. These detectors generally fall into one of four categories, namely, color-based approaches, shape-based approaches, texture-based approaches, and hybrid approaches.

Color-based approaches usually employ a two-stage strategy. First, segmentation is done by a thresholding operation in one specific color space. Subsequently, the shape detection is implemented and is applied only to the segmented regions. Since RGB color species very sensitive to illumination change, some approaches convert the RGB space to the HSI space which is partially invariant to light change. Other approaches implement segmentation in the normalized RGB space which is shown to outperform the HIS space. Both the HIS and the normalized RGB space can alleviate the negative effect of illumination change, but still fail on some severe situations.

Shape-based approaches detect edges or corners from raw images using canny edge detector or its variants. Then, edges and corners will be connected to regular polygons or circles by using Hough-like voting scheme. These detectors are invariant to illumination change, but the memory and computational requirement is quite high for large images. A genetic algorithm is adopted to detect circles and is in variant to projective deformation, but the expensive computational requirement limits its application.

Texture-based approaches firstly extract hand-crafted features computed from texture of images, and then use these extracted features to train a classifier. Popular hand-crafted features include HOG, LBP, ACF, etc. Hybrid approaches are a combination of the fore mentioned approaches. Usually, the initial step is the segmentation to narrow the search space, which is same as the color-based approaches. Instead of only using edges features or texture-based features, these methods use them together to improve the detection performance. All traffic signs have been fully annotated with the rectangular regions of interest (ROIs). Researchers can conveniently compare their work based on this benchmark.

To detect all kinds of objects in an image, we can directly use object localization. The difference is that we want our algorithm to be able to classify and localize all the objects in an image, not just one. So, the idea is, just crop the image into multiple images and run CNN for all the cropped images to detect an object.

The proposed algorithm works is defined in the following steps:

1. Make a window of size much smaller than actual image size. Crop it and pass it to ConvNet (CNN) and have ConvNet make the predictions.
2. Keep on sliding the window and pass the cropped images into ConvNet.
3. After cropping all the portions of image with this window size, repeat all the steps again for a bit bigger window size. Again, pass cropped images into ConvNet and let it make predictions.
4. At the end, you will have a set of cropped regions which will have some object, together with class and bounding box of the object.

2. LITERATURE SURVEY

The aim of traffic sign detection is to alert the driver of the changed traffic conditions. The task is to accurately localize and recognize road signs in various traffic environments. Prior approaches [8] [9], use color and shape information. However, these approaches are not adaptive under severe weather and lighting conditions. Additionally, appearance of traffic signs can physically change over time, due to the weather and damage caused by accidents. Instead of using color and shape features, most recent approaches employ texture or gradient features, such as local binary patterns (LBP) [2] and histogram of oriented gradients (HOG) [7]. These features are partially invariant to image distortion and illumination change, but they are still unable to handle severe deformations. Car detection is a more challenging problem compared to traffic sign detection due to its large intra-class variation caused by different viewpoints and occlusion patterns. Although sliding window-based methods have shown promising results in face and human detection [7], they often fail to detect cars due to a large variation of viewpoints. Recently the deformable parts model (DPM) [1], which has gained a lot of attention in generic object detection, has been adapted successfully for car detection. In addition to the DPM, visual sub categorization based approaches [1] have been applied to improve the generalization performance of detection model.

Generic Object Detection: Object detection is a challenging but important application in the computer vision community. It has achieved successful outcomes in many practical applications such as face detection and pedestrian detection [2] [7]. Complete survey of object detection can be found in [7]. This section briefly reviews several generic object detection methods. One classical object detector is the detection framework of Viola and Jones which uses a sliding-window search with a cascade classifier to achieve accurate location and efficient classification [3]. The other commonly used framework is using a linear support vector machine (SVM) classifier with histogram of oriented gradients (HOG) features, which has been applied successfully in pedestrian detection [7]. These frameworks achieve excellent detection results on rigid object classes. However, for object classes with a large intra-class variation, their detection performance falls down dramatically [4]. In order to deal with appearance variations in object detection, a deformable parts model (DPM) based method has been proposed in [6]. This method relies on a variant of HOG features and window template matching, but explicitly models deformations using a latent SVM classifier. It has been applied successfully in many object detection applications [5], [7], [9]. In addition to the DPM, visual sub categorization [5] is another common approach to improve the generalization performance of detection model. It divides the entire object class into multiple subclasses such that objects with similar visual appearance are grouped together. A sub-detector is trained for each subclass and detection results from all sub detectors are merged to generate the final results. Recently, a new detection framework which uses aggregated channel features (ACF) and an AdaBoost classifier has been proposed in [1]. This framework uses exhaustive sliding-window search to detect objects at multi-scales. It has been adapted successfully for many practical applications.

Traffic Sign Detection: Many traffic sign detectors have been proposed over the last decade with newly created challenging benchmarks. Interested reader should see [43] which provides a detailed analysis on the recent progress in the field of traffic sign detection. Most existing traffic sign detectors are appearance-based detectors. These detectors generally fall into one of four categories, namely, color-based approaches, shape-based approaches, texture-based approaches, and hybrid approaches.

Color-based approaches [8], [9] usually employ a two-stage strategy. First, segmentation is done by a thresholding operation in one specific color space. Subsequently, shape detection is implemented and is applied only to the segmented regions. Both the HSI and the normalized RGB space can alleviate the negative effect of illumination change, but still fail on some severe situations.

Shape-based approaches [5], [7] detect edges or corners from raw images using canny edge detector or its variants. Then, edges and corners will be connected to regular polygons or circles by using Hough-like voting scheme. These detectors are invariant to illumination change, but the memory and computational requirement is quite high for large images. In [8], a genetic algorithm is adopted to detect circles and is in variant to projective deformation, but the expensive computational requirement limits its application.

Texture-based approaches firstly extract hand-crafted features computed from texture of images, and then use these extracted features to train a classifier. Hybrid approaches [8] are a combination of the aforementioned approaches. Usually, the initial step is the segmentation to narrow the search space, which is same as the color-based approaches. Instead of only using edges features or texture-based features, these methods use them together to improve the detection performance. One standard benchmark for traffic sign detection is the German traffic sign detection benchmark (GTSDB) [8] which collects three important categories of road signs (prohibitory, danger, and mandatory) from various traffic scenes. All traffic signs have been fully annotated with the rectangular regions of interest (ROIs). Researchers can conveniently compare their work based on this benchmark.

3. PROPOSED SYSTEM MODEL

YOLOV3 (You Only Look Once) which is much more accurate and faster than the sliding window algorithm. It is based on only a minor tweak on the top of algorithms that we already

know. The idea is to divide the image into multiple grids. Then we change the label of our data such that we implement both localization and classification algorithm for each grid cell. Let me explain this to you with one more info graphic.

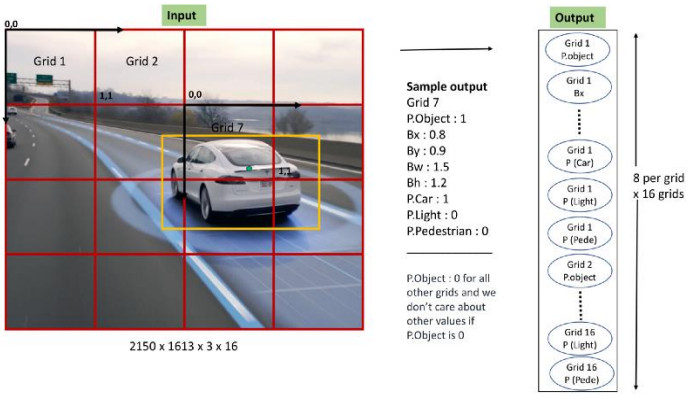

Fig.1. Bounding boxes, input and output for YOLO

For an input image of same size, YOLO v3 predicts more bounding boxes than YOLO v2. For instance, at its native resolution of 416×416, YOLO v2 predicted 13×13×5 = 845 boxes. At each grid cell, 5 boxes were detected using 5 anchors.

On the other hand YOLO v3 predicts boxes at 3 different scales. For the same image of 416×416, the number of predicted boxes are 10,647. This means that YOLO v3 predicts 10x the number of boxes predicted by YOLO v2. You could easily imagine why it's slower than YOLO v2. At each scale, every grid can predict 3 boxes using 3 anchors. Since there are three scales, the number of anchor boxes used in total are 9, 3 for each scale.

3.1 PROPOSED SYSTEM ARCHITECTURE

YOLOV3 with OpenCV: YOLOv3 is the third object detection algorithm in YOLO (You Only Look Once) family. It improved the accuracy with many tricks and is more capable of detecting small objects. Let's take a closer look at the improvements.

What is Mask YOLOV3?

- YOLOv3 is the latest variant of a popular object detection algorithm YOLO – You Only Look Once.
- The published model recognizes 80 different objects in images and videos, but most importantly it is super-fast and nearly as accurate as Single Shot MultiBox (SSD).
- Starting with OpenCV 3.4.2, you can easily use YOLOv3 models in your own OpenCV application.
- YOLO v3 now performs multilabel classification for objects detected in images.
- Earlier in YOLO, authors used to softmax the class scores and take the class with maximum score to be the class of the object contained in the bounding box. This has been modified in YOLO v3 as shown in the Fig.2.
- Softmaxing classes rests on the assumption that classes are mutually exclusive, or in simple words, if an object belongs to one class, then it cannot belong to the other. This works fine in COCO dataset.
- However, when we have classes like Person and Women in a dataset, then the above assumption fails. This is the reason why the authors of YOLO have refrained from softmaxing the classes. Instead, each class score is predicted using logistic regression and a threshold is used to predict multiple labels for an object. Classes with scores higher than this threshold are assigned to the box.

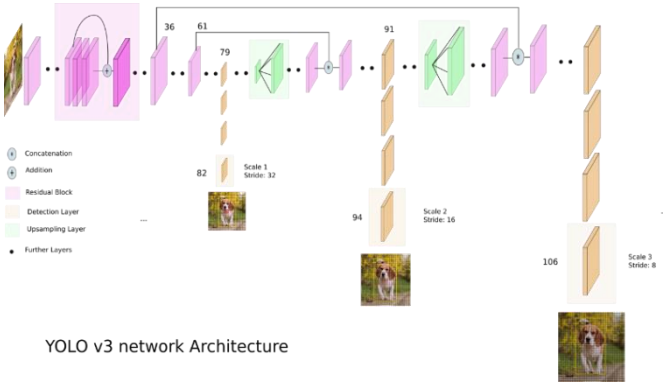

Fig.2. YOLOV3 Network Architecture

3.2 SYSTEM MODULES

The following modules are used in our project

- Data collection and pre-processing
- Training the model
- Testing the model
- Evaluating the experimental results

3.2.1 Data Collection and Preprocessing:

We used new dataset MSCOCO with the goal of advancing the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. Our dataset contains photos of 91 objects types that would be easily recognizable by a 4 years old. With a total of 2.5 million labeled instances in 328k images, the creation of our dataset drew upon extensive crowd worker involvement via novel user interfaces for category detection, instance spotting and instance segmentation.

This is achieved by gathering images of complex everyday scenes containing common objects in their natural context. Objects are labeled using per-instance segmentations to aid in precise object localization.

Data preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. In the pre-processing phase, the first step of the moving object detection process is capturing the image information using a video camera. In order to reduce the processing time, a grayscale image is used on entire process instead of the colour image. The grayscale image only has one color channel that consists of 8 bits while RGB image has three colour channels. Image smoothing is performed to reduce image noise from input image in order to achieve high accuracy for detecting the moving objects.

3.2.2 Training the Model:

Any machine learning training procedure involves first splitting the data randomly into two sets.

1. ***Training set***: This is the part of the data on which we train the model. Depending on the amount of data you have, you

can randomly select between 70% and 90% of the data for training.

2. ***Test set***: This is the part of the data on which we test our model. Typically, this is 10-30% of the data. No image should be part of the both the training and the test set.

We split the images inside the JPEG Images folder into the train and test sets. You can do it using the splitTrainAndTest.py scripts as follows, passing on the full path of the JPEG Images folder as an argument.

When you train your own object detector, it is a good idea to leverage existing models trained on very large datasets even though the large dataset may not contain the object you are trying to detect. This process is called transfer learning.

Instead of learning from scratch, we use a pre-trained model which contains convolutional weights trained on ImageNet. Using these weights as our starting weights, our network can learn faster. Let's download it now to our darknet folder.

3.2.3 Testing the Model:

20% of the data is used for testing. Test data is used only to assess performance of model. Training data's output is available to model whereas testing data is the unseen data for which predictions have to be made. The testing data is used to assess how well your algorithm was trained, and to estimate model properties.

Detailed description of the simulated experimental results with the YOLOV3 defined metrics are elucidated in the next section.

4. EXPERIMENTAL RESULTS

The results are obtained using Anaconda, pytrorch Prediction for the image by showing the Object was detected with the following three example output screens. The Proposed model accurately predicts the object classes in the image or video file and it also take the input from the webcam. Produce the output file which is having the class of the object and mask around each individual object in the input file and save the file as filename.py.

Prediction across Scales:

- 3 different scales are used.
- Features are extracted from these scales like Feature Pyramid Network (FPN).
- Several convolutional layers are added to the base feature extractor Darknet-53 (which is mentioned in the next section).
- The last of these layers predicts the bounding box, objectness and class predictions.
- On COCO dataset, 3 boxes at each scale. Therefore, the output tensor is $N \times N \times [3 \times (4+1+80)]$, i.e. 4 bounding box offsets, 1 objectness prediction, and 80 class predictions.
- Next, the feature map is taken from 2 layers previous and is up sampled by 2x. A feature map is also taken from earlier in the network and merge it with our up sampled features using concatenation. This is actually the typical encoder-decoder architecture, just like SSD is evolved to DSSD.
- This method allows us to get more meaningful semantic information from the up sampled features and finer-grained information from the earlier feature map.
- Then, a few more convolutional layers are added to process this combined feature map, and eventually predict a similar tensor, although now twice the size.
- K-Means clustering is used here as well to find better bounding box prior. Finally, on COCO dataset, (10×13), (16×30), (33×23), (30×61), (62×45), (59×119), (116×90), (156×198), and (373×326) are used.

YOLOv3 is pretty suitable! In terms of COCOs weird average mean AP (Average Precision) metric it is on par with the SSD variants but is 3x faster. It is still quite a bit behind other models like Retina Net in this metric though. However, when we look at the "old" detection metric of mAP at Intersection Over Union (IOU) = 0.5 in the simulated environment YOLOv3 is very strong. It is almost on par with Retina Net and far above the SSD variants. This indicates that YOLOv3 is a very strong detector that excels at producing decent boxes for objects. However, performance drops significantly as the IOU threshold increases indicating YOLOv3 struggles to get the boxes perfectly aligned with the object. In the past YOLO struggled with small objects. However, now we see a reversal in that trend. With the new multi-scale predictions, we see YOLOv3 has relatively high APS performance. However, it has comparatively worse performance on medium and larger size objects. More investigation is needed to get to the bottom of this. When we plot accuracy vs speed on the AP50 metric (see Fig.3-Fig.5) we see YOLOv3 has significant benefits over other detection systems. Explicitly, it's quicker and better.

Output Screens

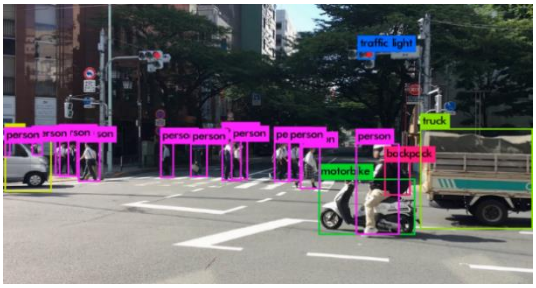

Fig.3. Output file 1 - Traffic signal object detection

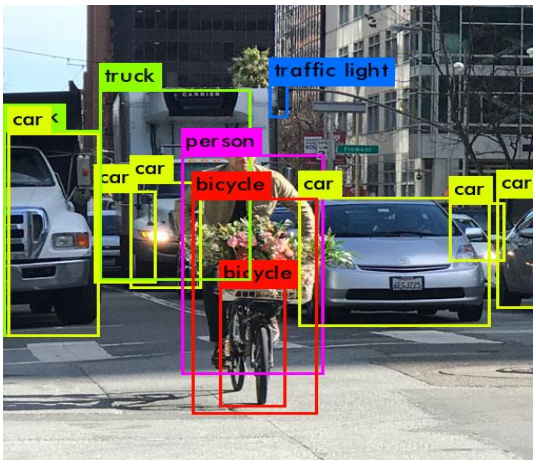

Fig.4. Output file 1 - Traffic signal object detection

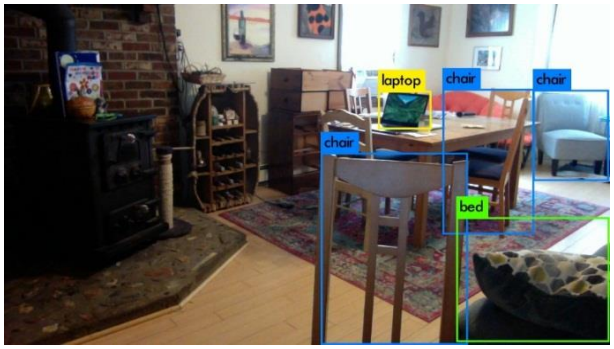

Fig.5. Webcam Output file 3 - Home environment object detection

5. CONCLUSION AND FUTURE WORK

This paper introduces an object detection and instance segmentation, a fast object detector for multiple categories. A key feature of our model is the use of multi-scale convolutional bounding box outputs attached to multiple feature maps at the top of the network and segmentation is done to each object. This representation allows us to efficiently model the space of possible box shapes. We experimentally validate that given appropriate training strategies, a larger number of carefully chosen default bounding boxes and instance segmentations results in improved performance. We built Yolov3 models with at least an order of magnitude more box predictions Instance segmentation sampling location than the existing methods. Accuracy and the speed on the AP metric in YOLOV3 has its significant benefits over other detection systems. Explicitly, which is a quicker and better.

The future enhancement of this project is we can use the small dataset with more accurate pixel segmentation to improve the accuracy and confidence score of object detection. Extending the object detection in images to videos (one can exploit temporal redundancy to come up better networks for video). For the extension of object detection, the action recognition will be implemented. We are focusing to extend our project scope to make it work for self-driving cars to security and tracking. Multi view tracking can be implemented using multiple cameras because of wide coverage range with different viewing angles for the objects to be tracked.

REFERENCES

[1] M. Andriluka, L. Pishchulin, P. Gehler and B. Schiele, "2D Human Pose Estimation: New Benchmark and State of the Art Analysis", *Proceedings of 27th IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 103-108, 2014.

[2] P. Arbelaez, J. Pont-Tuset, J.T. Barron, F. Marques and J. Malik, "Multi Scale Combinatorial Grouping for Image Segmentation and Object Proposal Generation", *Proceedings of 27th IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1-14, 2014.

[3] A. Arnab and P.H.S. Torr, "Pixel Wise Instance Segmentation with a Dynamically Instantiated Network", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1-21, 2017.

[4] M. Bai and R. Urtasun, "Deep Watershed Transforms for Instance Segmentation", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 341-349, 2017.

[5] S. Bell, C.L. Zitnick, K. Bala and R. Girshick, "Inside Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1-10, 2016.

[6] Z. Cao, T. Simon, S.E. Wei and Y. Sheikh, "Real Time Multi Person 2D Pose Estimation using Part Affinity Fields", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 231-239, 2017.

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding", *Proceedings of European Conference on Computer Vision*, pp. 43-52, 2016.

[8] J. Dai, K. He, Y. Li, S. Ren and J. Sun, "Instance-Sensitive Fully Convolutional Networks", *Proceedings of European Conference on Computer Vision*, pp. 551-559, 2016.

[9] J. Dai, K. He and J. Sun, "Convolutional Feature Masking for Joint Object and Stuff Segmentation", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 101-110, 2015.