Real-Time Object Detection with Audio Feedback

Vignesh R, Sandhya Rani, Manoj Yadav and Manjunath S.M
Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal, India

***Abstract*—We propose and innovative object detection method which enables the visually impaired to become aware of the objects in the frame of a camera. In addition to informing the person about the class of objects present in the image, we also include features that helps to identify the relative positions of the objects in the image. We achieve this objective using YOLO, a fast object identification and classification framework. We implement the modified framework in GPU and compare its results with that of the CPU. We develop an Android application to make this feature accessible and user friendly. Applications of our project include object location in an environment, road-crossing assistance for the blind, person identification from traffic webcams etc.**

Keywords*—*Convolutional Neural Networks(CNN), Deformable Parts Model (DPM), R-CNN (Recurrent Neural Network), Darknet, MS-COCO Dataset, Object Detection, Audio feedback, CUDA, Blind.

I. Introduction

The goal of object detection is to detect all instances of objects from a known class, such as people, cars or faces in an image or video. Real time object detection should be fast, accurate, and able to recognize a wide variety of objects. Since the introduction of neural networks, detection frameworks have become increasingly fast and accurate. Each detection is reported with some form of pose information. This could be as simple as the location of the object with respect to a fixed point. For example if an image showing a dog and a bicycle is given as input image, it may compute the locations of the dog as left/right to the bicycle, in addition to the bounding box of the dog and bicycle separately. The modern detection techniques use a classifier repeatedly. They take the object-classifier and use it at multiple locations within the image. Some systems use a more flexible approach such as a sliding window. In this mechanism, the image is divided equally and classifier is evaluated at each of these locations. Deformable Parts Model is one such system. On the other hand, region-based systems use a slightly different approach. They first generate a set of bounding boxes for each image and the classifier is evaluate at each of these enclosures. Following this, there is a refining and duplicate elimination phase and the bounding boxes are restored to the class of objects they represent. Convolutional Neural Networks are examples of such systems. Since each individual component must be trained separately, it leads to complex pipelines and difficult optimization mechanisms. Most methods used in practice have been designed to detect a single object class under a single view, thus these methods cannot handle multiple views, or large pose variations. Some works have tried to detect objects by learning subclasses or by considering views/poses as different classes; in both cases improving the efficiency and robustness. Also, multi-pose models and multi-resolution models have been developed. Efficiency is an issue to be taken into account in any object detection system. However, using specialized hardware like GPU some methods can run in real-time like deep learning. Integrating contextual information (e.g., about the type of scene, or the presence of other objects) can increase speed and robustness, Some proposed solutions include the use of (i) spatio-temporal context (ii) spatial structure among visual words and (iii) semantic information aiming to map semantically related features to visual words among many others. While most methods consider the detection of objects in a single frame, temporal features can be beneficial. The rest of the paper is organized as follows. In Section II we review related past work, and their drawbacks. In Section III we explain the YOLO and provide necessary background. Section IV elucidates our work. Section V explains briefly the limitations of the system. Finally, we conclude and describe our plan of future work in Section VI.

II. LITERATURE REVIEW

Research has been going on related to object detection system based on multiscale deformable part models [1]. This is able to represent highly variable object classes. However, their value had not been demonstrated on difficult benchmarks such as the PASCAL datasets. It totally depends on new methods for discriminative training with partially labeled data. It has a formalism called as latent SVM that is susceptible to local minima and thus sensitive to initialization. This is a common limitation of other methods that use latent information as well. [7] shows how multiscale and sliding window approach be efficiently implemented within a ConvNet using a feature extractor called OverFeat. The technique is analogous to DPM in that the localizer considers only the information available in the immediate neighbouring regions. There is a lack of global context in OverFeat. Hence, the initial results must undergo a high degree of processing to produce accurate results. With regard to prior detection systems, visual object detection framework that processes images extremely rapidly simultaneously achieving high detection rates was proposed [8]. This includes introduction of the Integral Image, a learning algorithm based on AdaBoost, to yield extremely efficient classifiers and a method for combining classifiers in a cascade. Though this system yields face detection performance comparable to the best previous systems, its face detection proceeds at 15 frames per second, which is not sufficient for todays world. Other approaches include application of convolutional neural networks (CNNs) to bottom-up region proposals in order to localize and segment objects followed by supervised pre-training and domain-specific fine-tuning, to yield a good performance boost [2]. This method makes training a multi-stage pipeline, expensive in space and time and object detection is slow taking more than 40 seconds per image at test time.

An extension to this was later proposed namely, Fast Region-based Convolutional Network method (Fast R-CNN) for object detection [3]. Fast R-CNN classifies object proposals using deep convolutional networks, uses several smart techniques to decrease training and testing time. DeepMultiBox is another proposed detector, which generates a few bounding boxes as object candidates [5]. Rather than using a specific search for an object (Selective Search), it uses a CNN. It can also perform detection of a particular object. However, detection of general objects is not very accurate in MultiBox and hence it is used as a component of a larger pipeline for detection.

III. BACKGROUND

The proposed work is intended to perform real-time object detection providing audio feedback for the detected objects and their relative positions.

The methodology re-frames object-detection as a single re-gression problem, straight from picture pixels to bounding box co-ordinates and class-probabilities. It is simple and extremely fast. It prepares on full pictures and speeds-up detection. This brought-together model has a few advantages over con-ventional strategies for object-detection. A single convolution neural-network all the while predicts various bounding boxes and class-probabilities for those containers. It also gives the audio feedback of the labels of the objects detected, along with the location semantics.

A. Single Neural Network for Detection

This model is not quite the same as others as the different parts of object-detection are brought together into a single convolution neural-network . This system utilizes features from the whole picture to anticipate each bounding box. It additionally predicts all bounding boxes over all classes for a picture all the while. This design empowers end-to-end training and real-time speed while keeping up high precision accuracy.

This framework divides the input picture into a S S matrix. In the event that the center point of an object falls into a matrix cell, that matrix cell is in charge of recognizing that object. Every matrix cell predicts B bounding-boxes, confidence-scores for those boxes and C class probabilities. The confidence-scores reflect how sure the model is that the box contains an object and furthermore how precise it supposes the box is that it predicts. The confidence-score is characterized as Pr(Object) IOUtruthpred. On the off chance that no object exists in that cell, the confidence-scores ought to be zero. This is shown in Fig 1.

Each bounding box comprises of 5 forecasts: x, y, w, h, and confidence-score. The (x, y) co-ordinates tell the center point of the box with respect to the limits of the matrix cell. The width and height are anticipated in respect to the entire picture. At long last the confidence expectation tells the IOU between the anticipated box and any ground truth box.

The C conditional class-probabilities, Pr(Classi—Object) are molded on the matrix cell containing an object. We just foresee one arrangement of class-probabilities per matrix cell, paying little heed to the quantity of boxes B. At test time we multiply the conditioned class- probabilities and the individual box confidence-forecasts,

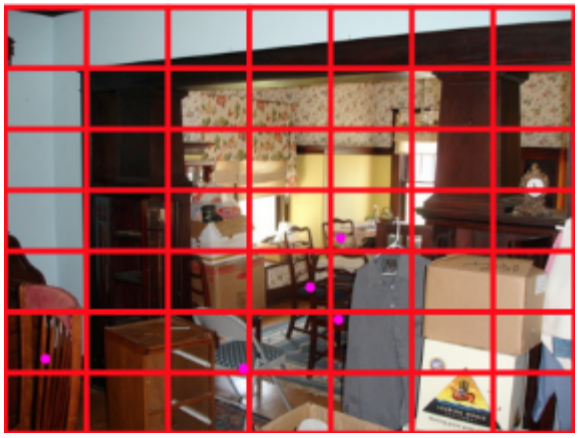

Fig. 1. Observe the object centers (see pink dots) within each grid cell. These respective grid cells will be responsible for these chair objects, Image taken from [10]

$$Pr(Class_i|Object)*Pr(Object)*IOU_{pred}^{truth} = Pr(Class_i)*IOU_{pred}^{truth}$$

which gives us class-particular confidence-scores for each box. These scores encode both the likelihood of that class showing up in the box and how well the anticipated box fits the object.

B. CNN Architecture

The proposed model re-frames object-detection as a single-regression problem, straight from picture pixels to bounding-box co-ordinates and class probabilities. It partitions the input picture into a S S matrix. In the event that the center point of an object falls into a matrix cell, that matrix cell is in charge of recognizing that object. A description of the CNN architecture can be seen in Figure 2.

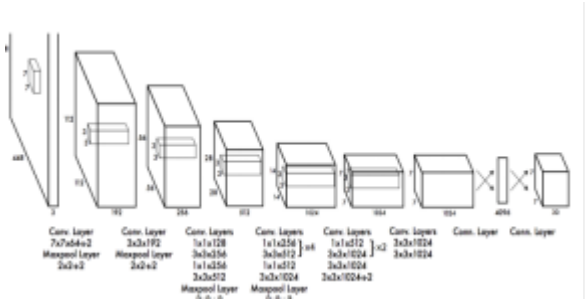

Fig. 2. The Convolutional Neural Network Architecture, Image taken from [10]

C. Dataset and Training

The MS-COCO dataset 2014 with annotations (key points) is used to train the given CNN model for performing detections. The dataset is created and maintained by Microsoft and is open-source. It contains a set of high-quality RGB images labeled with bounding box coordinates and 80 different class categories. The CNN learns high-quality, hierarchical features automatically, eliminating the need for hand-selected features. These convolutional layers are pre trained on the ImageNet 1000-class competition dataset. The Darknet framework [9] is used for all training and inference. This model is then changed over to perform detection. Including both convolution and associated layers to pre-trained networks can enhance execution as per past work. So four convolution layers and two fully connected layers are included with arbitrarily instated weights. The input-resolution of the system is expanded from 224 224 to 448 448 to give fine-grained visual data to the detection. The last layer predicts both class-probabilities and bounding-box coordinates. The bounding-box width and height are standardized by the picture width and height with

the goal that they fall in the vicinity of 0 and 1. The bounding-box x and y coordinates are likewise parameterized to be balances of a specific matrix cell area so they are additionally limited in the vicinity of 0 and 1. A linear-activation function is utilized for the last layer and every single other layer utilize the accompanying leaky rectified linear-activation:

$$\phi(x) = \left\{ \begin{array}{cc} x & \text{if x greater than 0} \\ 0.1x, & \text{otherwise} \end{array} \right\}$$

The aggregate-squared error is utilized in light of the fact that it is anything but difficult to enhance. The loss from bounding-box co-ordinate predictions is expanded and the loss from confidence forecasts for boxes that don't contain objects, is diminished.

This system predicts various bouncing-boxes per matrix cell. At training-time just a single predictor is "capable" for anticipating an object in light of which prediction has the most noteworthy current IOU with the ground truth. This prompts specialization between the bounding-box predictors. Every predictor shows signs of improvement at foreseeing certain sizes, perspective proportions.

The system is prepared for around 135 "epochs" on the preparation and approval datasets from "MS-COCO 2014". A batch-size of 64, a "momentum" of 0.9 and a "decay" of 0.0005 is utilized all through the preparation. To abstain from over-fitting we utilize dropout and broad data-augmentation.

IV. PROPOSED WORK

A. Audio Feedback

We take the classes predicted by the model and export it to a text file. This is achieved by adding a segment of code to the file image.c in the src folder of home directory. We then run a script that extracts the contents of the text file and converts it to audio. This can be performed by using Linux system utilities such as espeak. Thus, a visually impaired person can hear the objects in the camera frame.

B. Location Semantics

The system gives information about the positions of objects that are currently in the camera frame. So, the system can actually tell a visually impaired person the position of an object in terms of relative positions of other objects in the frame as to whether it is on the left or the right hand side. The system can also provide audio feedback about which objects are on the left hand side, right hand side and in the center of the frame. So, the system in this way, is beneficial for visually impaired as it can process in real time and give audio feedback about locations of objects detected.

C. Algorithm For Finding Respective Locations

We get the coordinates for object detections from the CNN. For finding the location of an object with respect to other objects, we use the following algorithm:-

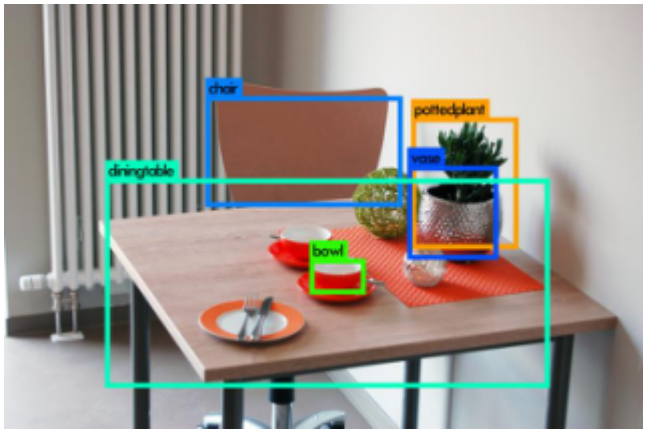

Fig. 3. Sample image

Algorithm 1 Algorithm for Finding Locations

Input: Set of coordinates (left, right, top, bottom) for detected objects

Output: Location of detected objects with respect to each other

Initialisation :

count = total_detected_objects

detections[0..count].coordinates = set of coordinates of detection

LOOP Process

for $i = 0$ to $count$ **do**
  for $j = i + 1$ to $count$ **do**
    if $detections[i].left > detections[j].right$ **then**
      print detections[i].class_name is right to detections[j].class_name
    else if $detections[i].right < detections[j].left$ **then**
      print detections[i].class_name is left to detections[j].class_name
    else if $detections[i].left < detections[j].left$ and $detections[i].right < detections[j].right$ **then**
      print detections[i].class_name is left to detections[j].class_name
    else if $detections[i].left < detections[j].left$ and $detections[i].right < detections[j].right$ **then**
      print detections[i].class_name is right to detections[j].class_name
    else
      print detections[i].class_name and detections[j].class_name are in front of you
    end if
  end for
end for

```
chair is left to pottedplant

chair is left to vase

chair and bowl are in front of you

chair and diningtable are in front of you

pottedplant is right to vase

pottedplant is right to bowl

pottedplant and diningtable are in front of you

vase is right to bowl

vase and diningtable are in front of you

bowl and diningtable are in front of you
```

Fig. 4. Output of Location Prediction

D. Android Application

In order to increase the utility of the application, an android application was made that takes an image as input and uploads it to an online web server. On the server, this object detection system is run. The output is the same image with the objects detected, having bounding boxes around them. The object classes that were trained on the system, will be detected based on their confidence scores. The output image can be downloaded using the app. The labels of the objects detected is converted into audio using Google APIs. This

application is mainly intended for visually impaired people so that an image of the surroundings can be taken and using our application, the objects around them can be known and the semantics feature gives an idea of objects location.

V. LIMITATIONS

This single CNN detection system still has an accuracy less than the best detection systems in practice today. The bounding box predictions are very restrictive in terms of spacial locality. Each cell has an upper limit of the number of boxes it can predict. There can be only one class associated with each cell. These restrictions result is moderate accuracy when many small objects are grouped together such as flock of birds. The audio output sentences do not convey a clear meaning for these closely-placed objects. This requires further fine-tuning of the location coordinates and the mapping from coordinate to locations.

VI. CONCLUSION AND FUTURE WORK

YOLO is a very efficient and fast framework and provides results with acceptable accuracy, which makes it suitable for deployment on mobile devices. Since all modern mobile operating systems provide a text-to-audio conversion engine, it is feasible to run the entire process viz. Image acquisition, Detection, Prediction and audio feedback in smartphone itself. This makes it very convenient for the end-user. Although our project is aimed at providing aid to the visually impaired, there are numerous applications of this approach and we look forward to exploring these avenues in our future work.

In further work, we intend to improve the accuracy of audio feedback service of the system. The current system can provide information about locations of objects properly in terms of left and right orientations. The system can be improved for better audio feedback for other orientations. In order to increase the utility of the system, we intend to add more features to our Android application such as ability to control the application with voice for visually impaired, provide real time object detection along with audio feedback. Another future work that we are considering is to add image question answering through NLP (Natural Language Processing) that can answer to questions like describing the current image scene, the shape and color of various objects detected etc.

ACKNOWLEDGMENT

The authors would like to thank Mr. Joseph Redmon, University of Washington, and his team for invention of the YOLO framework. This framework that has been used extensively in this project. We would also like to thank the open source community for their contribution towards making the framework fast and efficient.

REFERENCES

[1] P. F. Felzenszwalb, R. B. Girshick, D.McAllester, and D. Ramanan, *Object detection with discriminatively trained part based models,* 2010.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation.*,v5, IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[3] R. B. Girshick, *Fast R-CNN.* Proceedings of the International Conference on Computer Vision (ICCV). 2015.

[4] M. Everingham, S. M. A. Eslami, L. Van Gool, C.K. I. Williams, J. Winn, and A. Zisserman, *The Pascal Visual Object Classes Challenge: A Retrospective*, International Journal of Computer Vision, 2015.

[5] D. Erhan, C. Szegedy, A. Toshev and D.Anguelov, *Scalable object detection using deep neural networks*, Computing Research Repository, 2013.

[6] S. Ren, K. He, R. Girshick, and J. Sun. Faster, *R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, Computing Research Repository, 2015.

[7] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat, *Integrated Recognition, Localization and Detection using Convolutional Networks*, Computing Research Repository, 2013.

[8] P. Viola and M. J. Jones, *Robust Real-time Face Detection*, International Journal of Computer Vision, 2004.

[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L, *Microsoft COCO (Common Objects in Context*, Computing Research Repository, 2014.

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, *You Only Look Once:Unified, Real-Time Object Detection.*,v5, 1em plus 0.5em minus 0.4emThe IEEE Conference on Computer Vision and Pattern Recognition, 2016.