International Journal of Computer Trends and Technology
ISSN: 2231-2803/ https://doi.org/10.14445/22312803/IJCTT-V68I8P105

Volume 68 Issue 8, 33-38, August 2020
© 2020 Seventh Sense Research Group®

Original Article

Object Recognition for Visually Impaired People

Kavitha Srinivasan[1], Shanmuga Velayutham V[2], Vignesh G[3], Subash R[4]

[1,2,3,4]*Department of CSE, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603110, India*

Received Date: 28 June 2020
Revised Date: 06 August 2020
Accepted Date: 07 August 2020

Abstract - *Deep learning techniques are evolving rapidly in computer vision for many real-time applications, namely object detection, recognition, classification, segmentation, prediction and analysis. In this paper, an object recognition model for visually impaired people is proposed and validated using deep learning techniques for multiple datasets. The proposed model identifies multiple objects in a frame with its corresponding text, and the identified objects are converted into speech to guide the visually impaired people in real-time. The object identification process is carried out using a bounding box technique and a single convolutional neural network. The resulting bounding boxes with less probability than the threshold are eliminated, and the remaining objects are identified using a pre-trained Darkflow model. Then the identified objects are mapped to relevant text and converted to speech using Text-to-Speech (TTS) tool. The proposed model has been validated using four types of datasets, such as the Pascal VOC dataset, COCO dataset, BROID challenge dataset and Auto Rickshaw detection challenge dataset. The novelty of this work is: modified intersection over union algorithm for better recognition, chosen datasets have different sets of images, and the weight file is modified to recognize the objects of the challenge dataset. OpenCV and Compute Unified Device Architecture (CUDA) are used for image manipulation and graphics processing along with Tensorflow. The final output is obtained in audio format by applying TTS to the objects identified using Pyttsx, which is a python package that converts simple text to the speech signal.*

Keywords *- Object identification, Object recognition, YOLO, SSD, Intersection over Union, Darkflow, Text to speech.*

I. INTRODUCTION

Technological growth paves the way for huge data acquisition, processing and automation. Even though the information is abundant, object recognition, image auto–annotation and image search are still challenging tasks since it gives wrong results for many predictions in real-time. Usually, humans glance at an image and instantly know what objects are in the image, where they are, and how they interact through visual perception. However, for the visually impaired, fast, accurate algorithms are required for object detection and analysis. Some of the popular algorithms are related to neural networks and detection frameworks. Also, a colossal dataset is required for creating models using these algorithms.

One of the deep learning approaches, namely Region-based Convolutional Neural Networks (R-CNN), uses the region proposal method for generating potential bounding boxes in an image and then runs a classifier on these proposed boxes. After classification, post-processing is carried out for redefining the bounding boxes, eliminating duplicate detections, and the boxes are rescored based on other objects in the scene. But in this method, a single convolutional network simultaneously predicts multiple bounding boxes and their class probabilities, which consumes more time to detect and identify objects. Hence this method is not suitable for real-time object detection. A selection search is applied to compare the training set data with test data. Instead of applying the selection search technique to every sliding window, the whole frame is compared with the pre-trained model, thus making it much faster than its predecessors.

General-purpose object detection should be fast and must be able to recognize a wide variety of objects. The most common detection datasets contain thousands to hundreds of thousands of images with dozens to hundreds of tags. Labelling images for detection is far more expensive than labelling for classification or tagging.

The proposed model does preprocess, object recognition and object identification. Images are rescaled by running them through a tensor model, and object recognition is done using the Intersection over Union algorithm (IoU). The class label with maximum confidence value is returned as output. The image labels are given as audio output using the pyttsx engine to aid the visually impaired in their daily activities.

Pascal VOC is used as one of the datasets because of its wide variety of images. This system proposes a method to harness a large number of classes available in the Pascal VOC dataset and use it to expand the scope of current detection systems and to reduce the time complexity.

The remaining part of this paper spans the following subsections. In Section 2, the Literature survey related to deep learning techniques, object detection and datasets are explained. In Section 3, the design of the proposed object detection method, along with implementation details, are described. A summary of the results and discussions of four datasets are given in Section 4, and the conclusion is given at the end.

III. RELATED STUDIES

The research work on object identification and detection has been carried out using machine learning and deep learning techniques. Some of the significant disadvantages of machine learning over deep learning are (i). Acquisition of relevant feature sets is the major challenge. Also, the feature set needs to be modified as per the requirement of specific algorithms. This modification leads to a significant impact on results to be achieved or obtained. (ii). If the model is trained on a smaller dataset, the decisions might be wrong for the biased dataset. The main advantage of deep learning is that it reduces the need for feature engineering, which is one of the most time-consuming portions of machine learning practice. Some of the deep learning techniques are (i). Fast Region-based Convolutional Neural Network (Fast R-CNN) (ii). Faster R-CNN (iii). Single Shot Multi-Box Detector (SSD) (iv). You Only Look Once (YOLO). Fast R-CNN trains the network faster than R-CNN has been evaluated and justified using the Pascal VOC dataset. This method has several advantages, like the training process updating the weight of all network layers in a single stage supports sparse object representation to improve the object detection quality [1]. Faster R-CNN achieved better object detection performance by addressing the multi-scale Region Proposal Network (RPN) issue of COCO and Pascal VOC dataset [2] and outperforming the Fast R-CNN method. In the SSD method, the training process is easy and straightforward, and it performs better than Faster R-CNN. The dataset used in [3] is Pascal VOC, COCO, and ILSVRC datasets. YOLO accepts different size images as input and outperforms Fast R-CNN, Faster R-CNN and SSD using COCO detection and ImageNet classification dataset [4]. The advantages of using YOLO over other deep learning models are: learns a very general representation of object, fast and outperforms than the other detection models and less positive error rate [5].

A. Algorithm: YOLO

Input: A consecutive set of real-time images

Output: Returns the coordinates of the bounding box and the name of each object.

Step 1: Encode the bounding box using top-left (x_{min}, y_{min}) and bottom-right (x_{max}, y_{max}) corner coordinates.

$$\begin{bmatrix} x_{min} \\ y_{min} \\ x_{max} \\ y_{max} \end{bmatrix} \in \mathbb{R}^4 \Rightarrow b_{corner} = \begin{bmatrix} x_{min}/W \\ y_{min}/H \\ x_{max}/W \\ y_{max}/H \end{bmatrix} \in [0,1]^4$$

Where W and H represent image width and height, respectively.

$$b_{center} = \begin{bmatrix} (x_{min} + x_{\max}\ )/2W \\ (y_{min} + y_{max})/2H \\ (x_{max} - x_{min})/2W \\ (y_{max} - y_{min})/2H \end{bmatrix} \in [0,1]^4$$

$$\Rightarrow \quad b_{center} = \begin{bmatrix} x_c \\ y_c \\ x_c \\ y_c \end{bmatrix} \in [0,1]^4$$

$$b_{center} = \begin{bmatrix} (x_{min} + x_{\max}\ )/2W \\ (y_{min} + y_{max})/2H \\ (x_{max} - x_{min})/2W \\ (y_{max} - y_{min})/2H \end{bmatrix} \in [0,1]^4$$

$$\Rightarrow b_{center} = \begin{bmatrix} x_c \\ y_c \\ x_c \\ y_c \end{bmatrix} \in [0,1]^4$$

where $_{corner}$ represents corner-normalized bounding box, and b_{center} Represents centre-normalized bounding box respectively.

Step 2: Compute the YOLO bounding box from the centre-normalized bounding box. The width and height are predicted directly.

$$b_{yolo} = \begin{bmatrix} f(x_c, g_x) \\ f(y_c, g_y) \\ sqrt(w_c) \\ sqrt(h_c) \end{bmatrix} \in [0,1]^4$$

where $g_x = \lfloor 7.x_c \rfloor,\ g_y = \lfloor 7.y_c \rfloor\ and$

$$f(x_c, g_x) = 7.x_c - g_x$$

Step 3: Construct the truth vector y_{g_x, g_y} $from$ bounding box predictions of the grid cell (g_x, g_y) using IoU

$$b = \begin{bmatrix} x \\ y \\ w \\ h \end{bmatrix}, \widehat{b_1} = \begin{bmatrix} \widehat{x_1} \\ \widehat{y_1} \\ \widehat{w_1} \\ \widehat{h_1} \end{bmatrix}, \widehat{b_2} = \begin{bmatrix} \widehat{x_2} \\ \widehat{y_2} \\ \widehat{w_2} \\ \widehat{h_2} \end{bmatrix}$$

$$c = \max_{\hat{b} \in \{\widehat{b_1}, \widehat{b_2}\}} IoU\ (b, \hat{b}\ )$$

Step 4: YOLO loss for the grid cell with objects, no object assigned, and no purpose is given below for each bounding box.

$$Q_{obj} = \begin{pmatrix} 1 & \\ & \Lambda_{coord} \end{pmatrix} \in \mathbb{R}^{5x5}$$

$$z = \begin{bmatrix} \widehat{c_1} \\ \widehat{b_1} \end{bmatrix} - \begin{bmatrix} IoU\left(b, \widehat{b_1}\right) \\ b \end{bmatrix} \in \mathbb{R}^5$$

$$\mathcal{L}_{g_x, g_y} = z^T Qobj^z + \lambda_{noobj} \widehat{c_2}^2 + (p - \widehat{p})^T (p - \widehat{p})$$

$$\mathcal{L}_{g_x, g_y} = \lambda_{noobj}. (\widehat{c_1}^2 + \widehat{c_2}^2)$$

Step 5: The overall loss $\mathcal{L}$ for the whole image is computed from the sum of the loss of all grid cells.

$$\mathcal{L} = \sum_{g_x=0}^{6} \sum_{g_y=0}^{6} \mathcal{L}_{g_x, g_y}$$

In the Inception framework, object detection can be done by crossing a sliding window across the image [1]. At each step, the classifier predicts the label or the tag of the object inside the current window. A sliding window gives several hundred or thousand predictions of the current window image, even though the prediction with confidence score greater than the threshold value are taken up to the next step. This approach works better, but it's prolonged since the classifier runs many times. To overcome this drawback, a more efficient region proposal method has to be adopted in real-time [5]. In this approach [5], a single neural network is applied to the entire image only once, which results in a tensor batch size of 7x7x30. That is the image divided into a 7x7 grid with 30 class predictions.

Some of the datasets used for object detection includes Microsoft Common Objects in Context (COCO), SUN Database, ImageNet, PASCAL VOC, Broad Challenge and Auto-Rickshaw detection dataset. COCO is a large-scale dataset used in object detection, segmentation and captioning task. SUN database helps the researchers of many domains like computer vision, human perception, cognition and neuroscience, machine learning and data mining, computer graphics and robotics. It has a comprehensive collection of annotated images covering a large variety of environmental scenes, places and objects. ImageNet is an image dataset organized according to the WordNet hierarchy. Each meaningful concept in WordNet, possibly described by multiple words or word phrases, is called a "synonym set" or" synset". There are more than 100,000 synsets in WordNet, and the majority of them are nouns (80,000+). Pascal VOC dataset is suitable for object detection since the classes are labelled with a bounding box. Broad Challenge dataset consists of 329 images, and the task is to identify the bird and distinguish the bird from other objects using a bounded box. Auto-Rickshaw detection challenge dataset includes 698 images, and the task is to distinguish auto from other objects.

From the above study and analysis, the overall inference is that the YOLO is a suitable model for real-time object identification. The Pascal VOC dataset, COCO dataset, BROID challenge dataset can be used for testing purposes and performance analysis. Auto-Rickshaw detection challenge dataset can be used for training the model and validation.

III. SYSTEM DESIGN

The proposed system identifies multiple objects in a frame, converts each object into its corresponding text and is uttered as speech for the visually impaired, as given in Figure 1. The modules are dataset collection, preprocessing, object recognition, object detection and text-to-speech conversion. In dataset collection, four datasets, namely BROID dataset, COCO dataset, Pascal VOC and Auto Rickshaw detection challenge datasets, are used to analyze the efficiency of different models. In preprocessing, each image in the dataset is rescaled into a 7x7 grid and compared with pre-trained weights. Each preprocessed object in the image is recognized effectively by reducing the overlapping problem and identifying the object(s) by the box prediction method. Then, the recognized objects are detected by comparing with weights file, and the object label with the probability value greater than the threshold is returned along with the confidence score. Finally, the object labels returned are converted to speech using pyttsx, a python package that converts simple text to speech.

B. The workflow of the proposed system

Input: A consecutive set of real-time images

Output: An audio output describing the labels of the objects in the images.

Step 1: Dataset collection - Pascal VOC dataset, COCO dataset, BROID challenge dataset, Auto Rickshaw detection challenge dataset

Step 2: Pre-processing: For each image, rescale to 7x7 grids, compare with pre-trained weights and the results.

Step 3: Object Recognition: For each image, apply the Intersection over Union algorithm, which addresses the overlapping problem and recognizes the objects by the box prediction method.

Step 4: Object Detection: For each image, the recognized objects are compared with weights file, and the object label with the probability value greater than the threshold is returned along with the confidence score using multilabel suppression and class probability map and non-maxima suppression algorithms.

Step 5: Text to speech: The object labels returned are converted to address using pyttsx, a python package that converts simple text to speech.

Step 6: Performance Analysis: dataset Vs model.

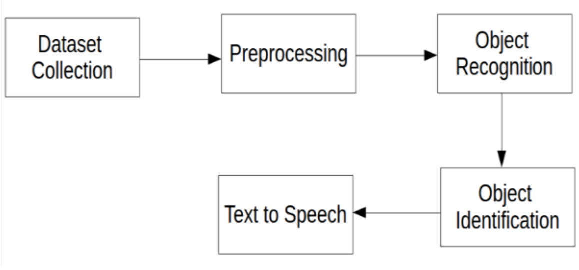


Fig. 1 System design

A. Dataset Description

The training dataset consists of images obtained from Pascal VOC. A subset of the large hand-labelled Pascal VOC dataset (60,000 labelled images depicting 80 classes) is used for training [6]. For testing, images from the COCO dataset, BROID challenge dataset and auto-detection challenge dataset are used. TABLE 1 explains the number of images and their classes for all datasets.

Table

Dataset	No. of images	No. of classes
BROID dataset	329	1
Pascal VOC	60000+	20
COCO dataset	123200+	80
Auto rickshaw	698	1

Table 1. Dataset description

B. Preprocessing

YOLO pre-trained weights are learned without any specific image preprocessing other than rescaling the image by running it through a tensor model. After rescaling the image to 7x7 grids, each grid in the image is compared with the pre-trained weights, and the results are obtained.

C. Object Recognition

The flow of the process to be carried out for object recognition using an overlapping problem and box prediction method is given in Figure 2.

Intersection over Union algorithm (IoU) algorithm [1] is used to evaluate the class prediction values of the bounding boxes by comparing the predicted bounding box with the ground-truth bounding box. If the intersection of the object is appropriate, then the probability value is increased. In reality, it is almost extremely unlikely that the x-y coordinates of the bounding boxes are exactly matching with the ground-truth coordinates value. The IoU value ranges between 0.0 to 1.0, depending on the intersection of the actual versus predicted bounding box. The higher intersections between the ground-truth box with the predicted box return the greater confidence value.

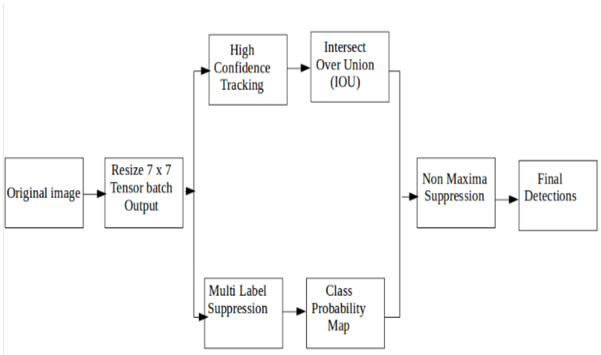


Fig. 2 Flow diagram for object recognition and identification

D. Object Detection

The recognized objects are then compared with weights file, and the object label with the probability value greater than the threshold is returned along with the confidence score [3].

a) Multilabel Suppression and Class Probability Map

In Multilabel Suppression, each grid's weight is compared with pre-trained weights, and a probability value of a grid belonging to a particular class is returned. The class with the highest probability value is chosen as the label for the grid. Based on the tags assigned, a class probability map is plotted, and the neighbouring similar class labels are merged to represent a single object. Now a bounding box is drawn around the objects identified. Class name and its respective confidence value is assigned and returned.

b) Non-Maxima Suppression Algorithm

Non-maxima algorithm [4] is used to overcome the overlapping problem. It first starts with the bounding box that has the highest score and removes any remaining bounding boxes that overlap it more than the given threshold amount (i.e. more than 50%). The same process is repeated until there are no more bounding boxes left. This removes any bounding boxes that overlap too much with other boxes that have a higher score. Finally, it stores the best bounding boxes.

E. Text-to-speech

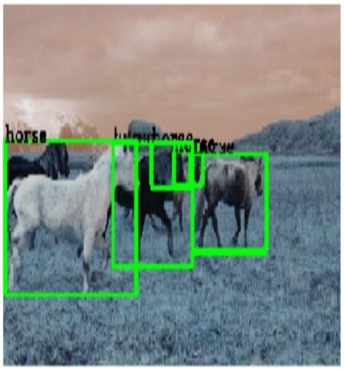
The object labels returned are converted to speech using pyttsx, a python package that converts simple text to speech.

IV. RESULTS AND DISCUSSION

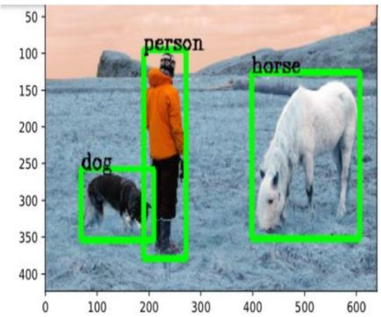
The proposed SSD and YOLO techniques for object detection are validated using four datasets, and the result of each stage are discussed and illustrated in this section. Finally, conclusions are compared and analyzed for each model and dataset in terms of speed and accuracy.

In the object identification phase, the detected objects are compared with the pre-trained model weight file, which has been trained with the Pascal VOC dataset. The intersection over union algorithm is then used to identify confidence value and class labels. The wrongly identified objects are removed since the confidence value remains lower than the threshold value.

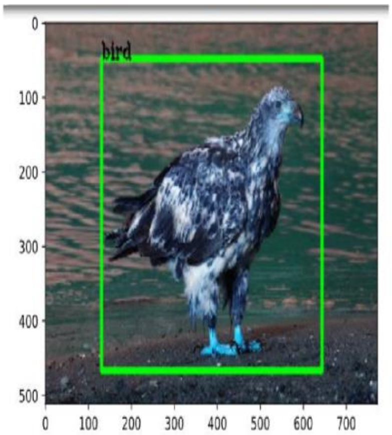
The results thus obtained are combined to obtain the class probability map, which specifies the object's location and its class label. Using multilabel suppression, the different labels and boxes identified with confidence values less than the mentioned threshold are neglected, and the final set of objects are identified.



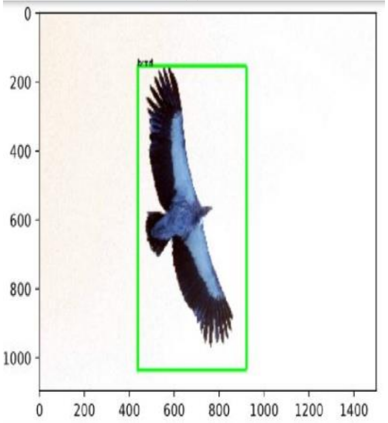
(a)



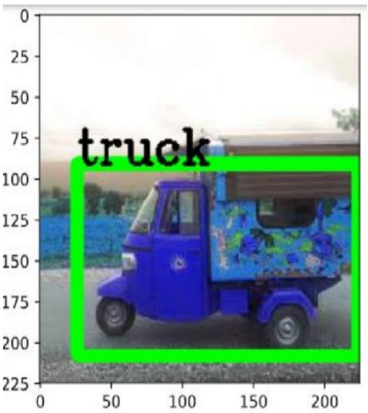
(b)



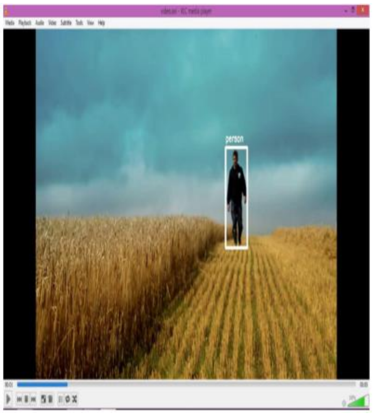
(c)



(d)



(e)

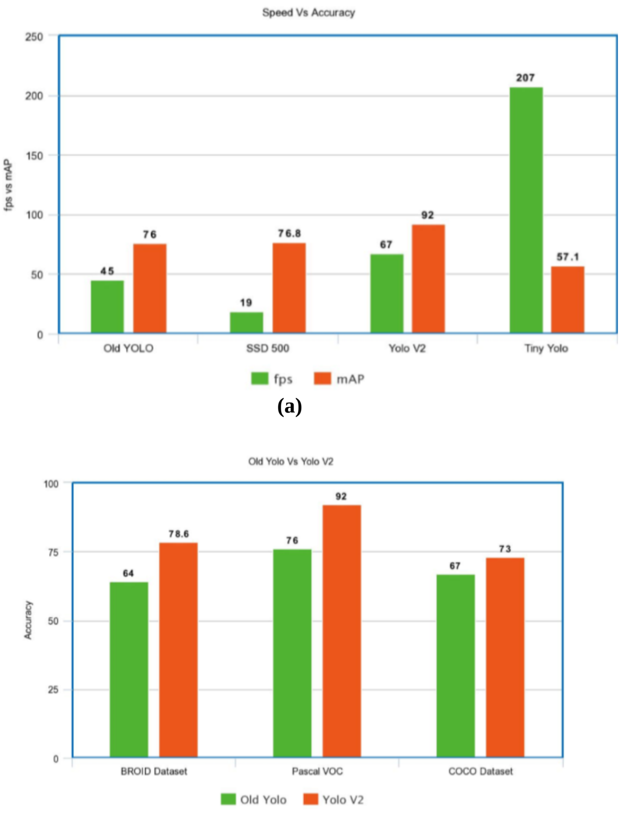


(f)

Fig. 3 (a-e) Object detection in four datasets (f) Text-to-speech conversion in video.

YOLO model is trained with the Pascal VOC dataset and tested for all four datasets. The multiple objects in the Pascal VOC dataset and COCO dataset in each image are detected with a bounding box, and the same is shown in Figure 3.a and 3.b, respectively. The YOLO model is evaluated using the BROID challenge dataset, and the bird is bounded accurately, and the result is shown in Figures 3.c and 3.d. But in the Auto-Rickshaw detection challenge dataset, some of the autos are labelled as the trunk, and the results are illustrated in Figure 3.e. From the effects, it has been inferred that for the Pascal VOC dataset and COCO dataset, multiple objects in a frame are recognized correctly with a high confidence score. In challenge datasets, the bird is recognized correctly, whereas some of the auto-rickshaws are known as the trunk. Finally, the text is converted to speech and tested for video using the Pyttsx tool, as shown in Figure 3. f.

The overall inference shows that Tiny YOLO has better speed and YOLO V2 has better accuracy, as shown in Figure 4. a. Among the proposed models, old YOLO and YOLO V2 are compared for each dataset, showing that Pascal VOC attains better accuracy for both models, and the results are illustrated in Figure 4. b.



(a)

(b)

Fig. 4 Performance analysis (a) Based on models (b) Based on Datasets

V. CONCLUSION AND FUTURE WORK

This research work is proposed for multiple object detection using the pre-trained model, namely You Only Look Once (YOLO), which has been implemented using darkflow. For testing the model, images from four different datasets such as Pascal VOC, COCO dataset, BROID challenge dataset and Auto Rickshaw detection challenge dataset are used. The input images are rescaled to a 7X7 grid by passing it to a tensorflow model. Then for object detection, bounding boxes are generated around the objects using selective search, and it is compared with the ground truth bounding boxes using Intersection over Union algorithm for prediction of the object along with confidence value. During prediction, the non-maxima suppression algorithm eliminates the overlapping problem by comparing confidence costs of more than one object in that box. For object identification, each box is compared with the weights file, and the class label with maximum confidence is assigned. Finally, a threshold has been specified using trial and error, and objects below that threshold are removed using multilabel suppression. The detected object labels are then converted to speech using the Pyttsx engine. The performance analysis is done on two different aspects: (i) based on the dataset, (ii) based on the model, using test images. From the analysis, it has been inferred that the YOLO v2 model has derived better accuracy and frames per second for the PASCAL VOC dataset since the model has been trained with that dataset only. YOLO imposes strong spatial constraints on bounding box predictions since each grid cell predicts two boxes and can only have one class. This spatial constraint limits the prediction of a number of nearby objects. Also, the small objects that appear in groups, such as flocks of birds, are not identified.

[11]

In future work, this research work can be developed as an android application for real-time object detection and identification. Furthermore, we would like to extend the research work for Google glass.

ACKNOWLEDGEMENT

We would like to thank the Department of CSE, Sri Sivasubramaniya Nadar College of Engineering, for the GPU facilities provided in the High-Performance Computing Lab to support our research work.

REFERENCES

[1] R. Girshick, Fast R-CNN, IEEE International Conference on Computer Vision (ICCV), Santiago, (2015). 1440-1448. doi: 10.1109/ICCV.2015.169.

[2] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence,39(6)(2017)1137–1149.
doi: 10.1109/TPAMI.2016.2577031

[3] Liu W. et al., SSD: Single Shot MultiBox Detector, In Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016, Lecture Notes in Computer Science, 9905(2016)21-37.doi: 10.1007/978-3-319-46448-0_2

[4] R. Joseph and F. Ali, YOLO9000: Better, Faster, Stronger, Conference on Computer Vision and Pattern Recognition (CVPR),(2016) 6517–6525.

[5] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, (2016)779-788, doi: 10.1109/CVPR.2016.91.

[6] M. Everingham, S.M.A. Eslami, L. Van Gool, C.K.L. Williams, J. Winn and Zisserman, The PASCAL visual object classes challenge, International Journal of Computer Vision, 88(2) (2010)303–338.

[7] X. Zhou, W. Gong, W. Fu and F. Du, Application of deep learning in object detection, IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), Wuhan, (2017)631-634, doi: 10.1109/ICIS.2017.7960069.

[8] Darkflow installation in Ubuntu. https://github.com/KleinYuan/ easy-Yolo, (2018).

[9] Darkflow installation in Windows. https://github.com/thtrieu/ dark flow, (4)(2018).

[10] Deep learning models. http://cv-tricks.com/object-detection/ faster-r-CNN-Yolo-SSD, (2018).