International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

BLIND ASSISTANCE WITH VOICE MODULE OUTPUT

Bobby Prathikshana M.*[*1], Nivetha V.[*2], Rinubha P.[*3], Dr. J. B. Jona[*4], Mr. S. A. Gunasekaran[*5]

*[*1,2,3]M.Sc Decision and Computing Science Dept. of Computing, Coimbatore Institute of Technology, India.*
*[*4] Guide, Associate Professor, Dept. Of Computer Applications, Coimbatore Institute of Technology, India*
*[*5] Guide, Assistant Professor, Dept. of Computing, Coimbatore Institute of Technology, India*

ABSTRACT

Multiple difficulties are being faced by visually challenged people. It becomes an uphill battle for them to perform their daily activities even in their familiar environments without constant support. Vision, one of the crucial senses, is important in perceiving the surrounding environment. Deep learning object detection frameworks aid in the process of restoring the core function, by identifying the objects just using simple camera devices. The proposed system uses a Computational neural network called Faster RCNN that detects objects and gives relevant audio output along with the distance of the object using depth estimation through speaker or headphone. The system is aimed at helping the visually impaired with a substitution system to assist them in everyday tasks.

Keywords*: Faster RCNN, visually impaired, object detection, depth estimation, neural network.*

1. INTRODUCTION

Vision impairment is a rising concern worldwide. It is identified by WHO that at least 2200 million people are visually impairment. They face constant challenges in navigation. They are in permanent need of assistance to guide them in day to day activities. It becomes quite a challenging work and the technical solution for vision impairment opens a new world to them. In computer vision, development of visual aids for blind is a very active research project. Deep Learning models help them in identifying day-to-day objects and prompt voice outputs and calculates distance which warns the person if they go near an object. The proposed system captures real-time image. The images are then pre-processed by resizing, creating labels etc. The pre-trained Faster RCNN model is applied on the preprocessed images resulting in feature extraction. The attributes are compared to the known object labels created in the pre-processing procedure to identify the objects. The name of the object is delivered to the user as and when the object is recognized using text to speech with Python.

2. RELATED WORK

Many research works attempt to develop visual aiding devices that recognize the surrounding input from the user, extract data about the input, and provide output through as human understandable audio information. A research paper titled ‘Smart Obstacle Detector for Blind Person’ proposed a system that aims at providing knowledge about a huge range of classes of objects that can be seen in the day to day environment , and analyses the size and distance of the object from the person and output is provided with a vibration using a vibrating motor connected to an ultrasonic sensor. MATLAB Software is used in Signal processing and videos are recorded using a camcorder and processed for feeding into the system. Many such research works that are currently in use come with a few disadvantages such as the repetition of objects under various differences including change of view, change of lighting and the change of size.

3. OVERVIEW OF THE SYSTEM

The model is a vision imaprover specific module for the visually impaired. The system is designed in such a way in which the blind person can take the help of an application which in turn sends real time frames to the laptop-based system. It works on real-time object detection using faster rcnn inception v2 algorithm and tensorflow APIs. Approximate distance of objects are calculated and voice module output is generated using pyttsx package in python.It makes the work of blind easy,efficient and reliable by sending wireless voice based feedback whether the particular object is either too close to him or is at a safer distance.

Resizing Data:

An image is resized by changing the dimensions along the height and width. The Open CV cv2.resize() function enables the system to resize the image and maintain uniformity.

Generating Labels for Images:

An image might contain multiple objects. It becomes an necessity to label the specific objects as to what category each of the objects belong to. The dimensions of the specific object in the image is tagged to its corresponding label. Our system is designed in such a way that almost 100 objects are distinguished and labelled. Therefore, nearly 100 different objects can be detected.

Bounding Box:

Bounding boxes are the close dimensions along which the object appears in an image. The better will be the accuracy of the model with precise bounding dimensions around the objects of interest without cutting off any portion of the object.

CSV File Generation:

First, we will load the required libraries into the python file (NumPy, OpenCV, etc.).Create an empty CSV file with the column name only if there is no existing CSV file(Height, Width, Channels, colors, etc.). If the file data.csv does not exist then a new file will be created by the else statement.Now we will use argparse() function to get the directory path of the images . It is obtained from the user in the command prompt.The image.shape function is used in order to find out the Height, Width, Channels of the image. Then the average red, average blue, green of the image is calculated. Then we will write the outputs to the csv file using writerow() function.

TF Records as Input

The TFRecord format is a simple format for storing a sequence of binary records.TFRecords are opaque: as serialized files, one cannot easily open them in a text editor to inspect their contents. Advantages of TFRecords As binary file formats, a TFRecord file takes up less disk space, which means every read/write operation that needs to be performed is faster. TFRecord files are optimized to handle component parts of a larger dataset.That is if a given dataset requires a space more than the memory, streaming a subset of the dataset is easily done. This is exactly what happens when training on a single batch of data: the machine is using a subset of the overall data.The major advantage of TFRecord files is that they are optimized for storing sequenced data which are essential for word or time sequences.

Faster RCNN Model:

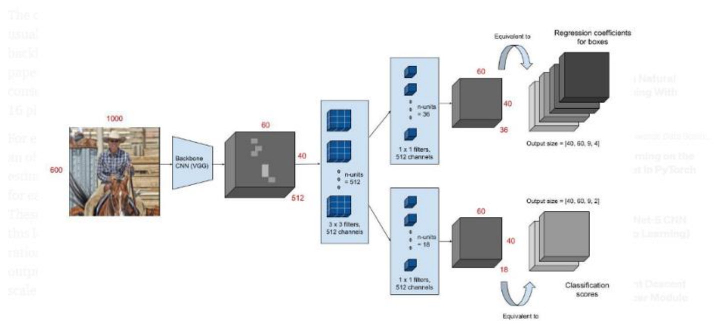
1000
600
Backbone CNN (VGG)
60
40
512
n-units = 512
3 x 3 filters, 512 channels
n-units = 36
1 x 1 filters, 512 channels
n-units = 18
1 x 1 filters, 512 channels
60
40
36
60
40
18
Equivalent to
Regression coefficients for boxes
Output size = [40, 60, 9, 4]
Output size = [40, 60, 9, 2]
Classification scores
Equivalent to

Figure 1 RPN model for input image

The model which has been used for object detection is the Faster Regional Convolutional Neural Network. The very first step of the algorithm is the region proposal network (RPN). with the help of RPN, we will try to find out the area where the object can be possibly detected. Once you get the area of object presence, we will label that area as a foreground class. The rest of the area is labelled as background class. The object which is detected as the foreground class will move to the next stage of the algorithm which starts with the input image being fed into the backbone convolutional neural network. The first stage in RPN is to generate anchor boxes. Then the input image is resized with 600px as its shortest side and the longer side not exceeding 1000px. The next step is to find out the IOU that is Intersection Over Union. In simple words, we need to find out the region of overlapping the object. The output features of the backbone network (indicated by H x W) are smaller than the input image. It depends on the stride of the background network. For both the possible background networks used (VGG, ZF-Net). Here the network stride is 16. This means that two consecutive pixels in the background relate to two points 16 pixels excluding the input image. For every point in the output feature map, the network should learn to find an object presence in the input image at its corresponding location and find its size. This is done by marking a set of “Anchors” on the input image for each location on the output feature map. The architecture shows 9 possible anchors in 3 different aspect ratios and 3 different sizes placed on the input image for a point A on the output feature map. The anchors used have 3 scales of box area 128^2, 256^2, 512^2 and 3 aspect ratios of 1:1, 1:2 and 2:1. First, a 3 x 3 convolution with 512 units is applied to the backbone feature map to give a 512-d feature map for every location. Now, this layer is followed by two similar layers:

- 1 x 1 convolution layer with 18 units for object classification,
- 1 x 1 convolution with 36 units for bounding box regression.

The 18 units in the classification layer gives an output of size (H, W, 18). This output gives 4 regression coefficients with each of the 9 anchors for every point in the background feature map. These regression coefficients are used to improve the coordinates of the bounding boxes that contain objects.

Training and Loss functions:

The output feature map includes 40 x 60 locations, in accordance to 40x60x09 equivalent to a total of 20000 anchors. While training, the anchors that go beyond the boundary are omitted thereby to improve the accuracy. This leaves us with around 6000 anchors for each image. A positive sample, has to gratify one of the two conditions — i) It must have the maximum Intersection Over Union, with a ground truth box or ii) It must have an IoU more than 0.7 with any ground truth box. One ground truth box can generate numerous anchors to have assigned positive labels. An anchor is labeled negative when its IoU with all ground truth boxes is below 0.3. The rest of the anchors (not positive or negative) are eliminated for RPN training.

Each mini-batch for training the RPN is taken from one single image. The training loss for the RPN is also a multi-task loss, given by:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*).$$

The i here is the anchor index of the mini-batch. The classification loss $L_{cls}(p_i, p_i*)$ is the log loss of whether it is an object or not an object. p_i is the output score that is calculated from the classification branch for anchor i, and p_i* is the ground truth label of either 1 or 0. The regression loss $L_{re}(t_i, t_i*)$ is activated only when the anchor actually contains an object, that is while the ground truth value denotes to 1. The term t_i is the predicted output of the regression layer which contains 4 variables [t_x, t_y, tw, t_h]. The regression target t_i* is calculated using

$$t_x = (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a, \quad t_w = \log(w/w_a), \quad t_h = \log(h/h_a)$$

Test time details:

While testing, the 20000 anchors from every image are put through a series of post-processing phases to send in the object proposal bounding boxes. The regression coefficients are pertained to the anchors for optimal localization. This gives accurate bounding boxes. All the boxes are arranged based on their cls scores. Then, a non-maximum suppression (NMS) is pertained with a threshold of 0.7. From the top down, each of the bounding boxes having an IoU of more than 0.7 with other bounding boxes are disregarded. Therefore, the bounding box, with the maximum score is retained for a group of overlapping boxes. This gives around 2000 proposals for every image. The cross-boundary bounding boxes are retained and clipped to the image boundary. While using these object proposals to train the Fast R-CNN detection pipeline, all 2000 proposals from the RPN are used. While testing for Fast R-CNN detection, the Top N proposals from the RPN are chosen.

Dimensionality:

Mean average precision (mAP) is the metric used in evaluating object detection models such as faster rcnn , yolo, fast rcnn etc The map is calculated over recall values ranging between 0 and 1. The mAP is used in comparing the ground-truth bounding box positioned around the object in a video or image input, to the box detected by the system. Then the model returns a score for the comparison.The accuracy of the object detection model is determined by the score returned. When the score is higher, the accuracy of the model is evaluated to be better. Stride is one of the components of convolutional neural networks, used for the compression of images and video data. Stride is a filter parameter of the neural network that modifies the amount of movement over the image or video. Consider an example where stride is set to 5, it means the filter will move 5 pixels at a time.

Softmax Activation Function:

The softmax function is used in assigning probabilities in decimals for each of the classes while considering a multi class problem. It turns a n valued vector to its sum between 0 and 1. Thus a softmax function obtains its input as a positive or negative or zero valued vector and converts it into a probability lying between 0 and 1. Therefore , the function generates a probability closer to 1 when large numbers are given as input and generates a probability closer to 0 when a small or negligible valued input is provided. This means that it gives the likelihood of the object in an image belonging to a particular class. It should also be noted that the sum of all the probabilities will be equal to one. If the softmax function is used for a multi-classification model it returns the probabilities of each class and the target class will have a high probability.

Momentum optimizer:

Momentum optimization algorithm allows the search to build inertia. It is built in a direction in the search space and overcomes the oscillations of noisy gradients and coast across flat spots of the search space. Here, we define a parameter that accumulates a weighted average of the past gradients. Then we use this parameter in our update rule. Therefore, we add a weighted average of previous updates to our current update. The value of the momentum parameter determines the weight of the previous updates. As a result of accumulating the gradients, the updates get larger when the algorithm repeatedly gets gradients having similar direction. It is most useful in optimization where its objective function has a large amount of curvature, meaning that the gradient might differ a lot over small regions of the search space.

Max Pooling:

Max pooling is a type of operation that is added to neural networks followed by the convolutional layers. When max pooling is added to the RCNN model, it reduces the image's dimensionality by reducing the number of pixels in the output, which is distinguished from the previous convolutional layer. It is used to calculate the maximum or the largest value in the feature map. It is based on the discretization process. In max pooling, an n by n region is defined as a corresponding filter. Here, the filter size is 2x2. Then we define a stride of 1 and padding of 1 which will lead to a 3x3 dimension image. This indicates how many pixels do we want our filter to move as it slides across the image. Therefore, the output after max-pooling layer would be a feature map containing the features of the previous feature map. The max pooling filter is applied to each cell in the grid to return a single value. All values from all cells in the grid are applied with the max pooling filter. In order to detect the objects precisely, max pooling is used.

Depth Estimation:

Depth estimation or extraction is an estimation of visual representation which is vitally used to interpret the three-dimensional structure of an image by quantifying the degree of close and distant from the photographing systems such as camera and the image.In other words, it is used to calculate distance between two real time objects. Our main objective is to assist the blind people in providing warning to the blind people about the hurdles coming their way. In order to do this, we need to find how much distance the obstacle and person are located in any real time situation. As soon as the object is detected, a rectangular box known as a bounding box is framed around that object. If that object fits the frame completely, then with some constraints an approximate distance of the object from the particular person is calculated. Following code is used to recognize objects and to return the information for the locations and confidence: (boxes, scores, classes, num_detections) = sess.run([boxes, scores, classes,num_detections],feed_dict={image_tensor: image_np_expanded})

4. RESULTS

In this section, the performance evaluation of our proposed system is described with the real time results obtained from using the system in a live environment. This involves running the system in the laptop to allow it to detect the objects that a person might come across while walking ina a room which includes the detection of objects such as water bottle, book, remote, spoon, chair, person, cat , dog etc. The performance of using object detection model, and the performance of the discrimination between different objects i different lighting conditions have been verified. In order to check the accuracy of detections of our object detection model under several different scenarios, the system has been taken to detect objects in various locations, angles, and times of day and night in the live camera. Also, images taken at various mentioned scenarios are fed into the system apart from the image data that was used with the training model. Rest of the objects were from verification data that includes live data and the images in testing were taken elsewhere and are different from the ones used during the training process.All of the images used for testing the system were sent by the camera module as the live environment was successfully recognized in real time. Bounding boxes were accurately positioned around the images in the videos and images. Not only the position of the obstacles but also the coordinates were accurately recognized. Initially the Faster R-CNN model is trained over the data of images that correspond to different class labels. The accuracy and speed of the faster rcnn model is calculated using the COCO mean average precision (mAP). Since we have utilised the pretrained model , the accuracy and the speed of the Faster R-CNN was great and the COCO mAP was smaller. For the verification of live data where data taken from different locations, sizes and lighting (daytime and nighttime) were used, Faster R-CNN had a higher recognition rate, accurately detecting 94% of obstacles. The live environment was considered for the verification during different times of the day. The algorithm confirmed if the blind person is safe to cross a place or could get hit by an obstacle. The accuracy of detection is relatively lesser when the environment has poor lighting conditions. Most crucial part of the system is the prediction and monitoring of the state of obstacles, because any incorrect guidance could impair the safety of the blind person using the system and cause harm.

5. CONCLUSION

It is a simple, economical, configurable electronic guidance system which is easy to handle and it is proposed to provide constructive assistance and support for blind and visually impaired persons. The system has been designed and implemented. The results of the real-time system are encouraging. The results show that the system is efficient and unique in detecting the distance and identifying the object that the blind person may encounter. Future work on this system will be focused to enhance the performance of the system and reduce the load on the user and to incorporate additional features thereby making it a more efficient system.

REFERENCES

[1] https://github.com/tensorflow/models/

[2] Peter Harrington, "Machine Learning in Action, pp."by Manning Publications-1st edition.

[3] https://www.tensorflow.org/datasets/catalog/coco

[4] [4] Zraqou, Jamal & Alkhadour, Wissam & Siam, Mohammad. (2017). "Real-Time Objects Recognition Approach for Assisting Blind People." Multimedia Systems Department, Electrical Engineering Department, Isra University, Amman-Jordan Accepted 30 Jan 2017, Available online 31 Jan 2017, Vol.7, No.1

[5] A. Dionisi, E. Sardini and M. Serpelloni, "Wearable object detection system for the blind," 2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings, Graz, 2012, pp. 1255-1258, doi: 10.1109/I2MTC.2012.6229180

[6] Daniyal, Daniyal & Ahmed, Faheem & Ahmed, Habib & Shaikh, Engr & Shamshad, Aamir. (2014). “Smart Obstacle Detector for Blind Person.” Journal of Biomedical Engineering and Medical Imaging. 1. 31-40. 10.14738/jbemi.13.245.

[7] Christian Szegedy Alexander Toshev Dumitru Erhan, “Deep Neural Networks for Object Detection.”

[8] N.Saranya, M.Nandinipriya, U.Priya,”Real Time Object Detection for Blind People”,Bannari Amman Institute of Technology, Sathyamangalam, Erode.(India).

[9] Rui (Forest) Jiang,Qian Lin,Shuhui Qu,”Let Blind People See: RealTime Visual Recognition with Results Converted to 3D Audio”,Stanford,2018.