A Research Using Object-Detection with a Microscopic Viewpoint to Identify Human Activities for Visually Impaired People

Sameer Sadman Chowdhury
dept. of CSE
BRAC University
Dhaka,Bangladesh
sameer.sadman.chowdhury@g.bracu.ac.bd

Nishat Zerin
dept. of CSE
BRAC University
Dhaka,Bangladesh
nishat.zerin@g.bracu.ac.bd

Muntasir Ahmed Ador
dept. of CSE
BRAC University
Dhaka,Bangladesh
muntasir.ahmed.ador@g.bracu.ac.bd

Rushayed Ali Faiaz
dept. of CSE
BRAC University
Dhaka,Bangladesh
rushayed.ali.faiaz@g.bracu.ac.bd

Mehnaz Ara Fazal
dept. of CSE
BRAC University
Dhaka,Bangladesh
mehnaz.ara.fazal@g.bracu.ac.bd

Adib Muhammad Amit
dept. of CSE
BRAC University
Dhaka,Bangladesh
adib.muhammad.amit@g.bracu.ac.bd

Annajiat Alim Rasel
dept. of CSE
BRAC University
Dhaka,Bangladesh
annajiat@gmail.com

***Abstract*—Human activity recognition is currently an incredibly active research field in the development of advanced computing technology. In this project using HAR, the goal is to aid visually impaired individuals by allowing them to recognize human activities. Integrating object detection into Human Activity Recognition (HAR) poses the difficulty of precisely and effectively detecting and tracking pertinent objects within the intricate dynamics of human actions. This aims to enhance the comprehension of activities by discerning the interaction and manipulation of objects within a specific environment. However, the implementation of object detection in assisting the visually impaired is quite unexplored. Our research demonstrates that Object Detection is a viable method to help the visually impaired in an urgent situation where the individual needs to understand the environment and adapt accordingly.**

***Index Terms*—pertinent, style, styling, insert**

I. Background and Motivation

Object detection has been used in the case of aiding the visually impaired. Various models like YOLO, SDD, and Mobilenet have been used in this field in various applications from real-time detection of pedestrians to real live detection using mobile devices. However, the field is mostly focused on a live feed that supports the visually impaired person whereas the urgent support system of the visually impaired is left barren. Our research primarily delves into the topic of helping a disoriented person take a picture of the time of disorientation such that this can be implemented into other systems or is outside of the scope of our research. Wang et al. (2018) state the use of mobile devices makes it feasible for our scope of research, Sharma et al.(2022) compared various models to create a real-time object detector such that we can observe with real-time perspective, but there is staggering like of time specific image processing in urgent situations in the literature even though there are an array of papers consolidating the field of HAR research.

II. Related Work

In the realm of object detection, several seminal works have significantly contributed to the advancement of real-time and accurate detection methodologies. The groundbreaking "You Only Look Once: Unified, Real-Time Object Detection" by Joseph Redmon et al. introduces the YOLO model, leveraging a single neural network for simultaneous bounding box prediction and class probability estimation, achieving remarkable processing speed. Zhong-Qiu Zhao, et al.'s "Object Detection With Deep Learning: A Review" provides an overview of deep learning-based object detection techniques, contextualizing the evolution of the field and offering valuable insights for researchers and practitioners. Naman Jain et al.'s "Performance Analysis of Object Detection and Tracking Algorithms for Traffic Surveillance Applications using Neural Networks" delves into the application of convolution layers and neural networks for object detection in traffic surveillance, emphasizing the necessity of intelligent systems. Furthermore, the exploration of YOLOv3 and YOLOv4 in "YOLOv3 and YOLOv4: Multiple Object Detection for Surveillance Applications" showcases their effectiveness in multiple object detection scenarios, offering insights into their implementation and potential deployment. Nehashree M R, et al.'s "Simulation and Performance Analysis of Feature Extraction and Matching Algorithms for Image Processing Applications" contributes a comprehensive examination of feature extraction and matching algorithms, providing algorithm comparisons and practical methodologies. Lastly, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" by Shaoqing Ren et al. presents a significant advancement in real-time

object detection, introducing the Region Proposal Network as a key innovation to improve efficiency and accuracy. Collectively, these works contribute to the rich landscape of object detection research, offering diverse perspectives, methodologies, and insights for advancing the field.

III. Methodology

The proposed model by an implementation that is used for the scope of the research is:

A. YOLO

YOLO, or "You Only Look Once," is one of the crucial objects that works as a detection algorithm. It was created by Joseph Redmon Santosh Divvala, Ross Girshick, and Ali Farhadi, YOLO is known for its quick and effective object detection and classification by using only one neural network pass. It creates a grid using an image then gives each grid cell a job, where it estimates bounding boxes and class probabilities for included items in itself. Then the algorithm applies a class probability by distributing it to each bounding box and then predicts multiple bounding boxes in each cell with corresponding confidence scores. Using an all-inclusive loss function that also adds up localization, confidence and classification losses helps YOLO to achieve precise localization and classification with the help of anchor boxes. YOLO can be used in real-time such as for autonomous vehicles, robotics, and surveillance that requires accurate and quick object recognition for example YOLOv3 known as one of the most popular versions.

B. OpenCV(Open Source Computer Vision Library)

It is an open-source computer vision and machine learning software library. OpenCV is designed to provide a common infrastructure for computer vision applications and accelerate the use of machine perception in commercial products. As an Apache 2 licensed product, OpenCV allows businesses to easily use and modify the code. OpenCV is open-source, meaning it's freely available and can be used, modified, and distributed by anyone. It offers a vast library of computer vision and image processing functions, making it suitable for various applications. OpenCV is compatible with multiple platforms, including Windows, Linux, macOS, Android, and iOS. It has a large and active community, providing support, tutorials, and resources for users. OpenCV is written in C/C++ and optimized for performance, making it efficient for real-time and resource-intensive tasks.

C. Faster (R-CNN)

Faster R-CNN, or Region-based Convolutional Neural Network, is a popular and influential object detection algorithm in the field of computer vision.

Fast R-CNN is known for its high accuracy in object detection tasks. Achieve state-of-the-art results on a variety of benchmark datasets. Two-stage detection: Faster R-CNN uses a two-stage approach that separates region proposal generation from object classification. This allows the model to focus more on potential object areas, improving detection accuracy. Region Proposal Network (RPN): The introduction of RPN to Faster R-CNN helps to efficiently propose candidate object regions, thereby increasing the number of regions that the model needs to consider. The number of areas is reduced. This increases both speed and accuracy. Flexibility: Faster R-CNN is flexible and can be used for object detection, instance segmentation, keypoint detection, etc. By appropriately adjusting the network structure and loss function. This can be applied to a variety of tasks. End-to-end training: The model can be trained end-to-end, allowing the entire system, including the region proposal network and object detection network, to be optimized together

D. RetinaNet

RetinaNet is a popular object detection model that overcomes some of the limitations of previous models, especially in handling object detection tasks with large numbers of classes and tackling small object detection challenges.

Here are some advantages and potential disadvantages of RetinaNet: Focus loss for handling unbalanced data: RetinaNet is a class leader in object recognition. We introduce a focal loss to address the imbalance problem. Loss of focus increases the importance of hard-to-classify examples, such as rare classes or difficult instances, and helps the model focus on difficult cases during training. Anchor Box and Feature Pyramid Network (FPN): RetinaNet uses an Anchor Box and Feature Pyramid Network (FPN) architecture. Anchor boxes help your model efficiently handle objects of different sizes and aspect ratios. FPN allows the network to capture and utilize features at multiple scales, making it easier to detect objects at different resolutions. High accuracy across all scales: RetinaNet's design allows for high accuracy across a wide range of object scales, from small to large. This is important for detecting objects of different sizes in images. Single-shot object detection: RetinaNet is a single-shot object detection model. This means that you can predict object classes and bounding boxes in one forward pass through the network. This improves computational efficiency compared to two-stage detectors such as Faster R-CNN. RetinaNet demonstrated state-of-the-art performance on various benchmark datasets and demonstrated its effectiveness in accurate and efficient object detection.

E. Histogram of Oriented Gradients (HOG)

In computer vision and image processing, the Histogram of Oriented Gradients (HOG) is a well-liked feature descriptor method. To characterize an object's form and appearance, it examines the distribution of edge orientations inside it. The HOG approach divides an image into tiny cells by first calculating the gradient magnitude and direction for each pixel in the picture.

Apart from edge detection, the HOG descriptor may also determine the direction of edges in an image. HOG breaks the image up into smaller parts and calculates the gradients and orientations of each one individually rather than evaluating the

full image at once. This specialized method offers comprehensive data. After that, for every segment, histograms are produced. These serve as bar charts that represent the frequency of various edge directions, enabling HOG to comprehend the composition and geometry of objects in various areas of the picture.

HOG is sensitive to changes in light conditions since it depends on gradient information. Furthermore, because its primary focus is on edge orientation distribution, it may perform less well in situations where texture is crucial. In addition, it may be difficult for the method's fixed-size analysis zones to adjust to these changes, which would restrict its capacity to precisely describe and identify things with intricate forms and appearances.

F. MASK R-CNN

Mask R-CNN is an advanced deep-learning model used in computer vision for things like segmentation. By extending the Faster R-CNN architecture with a mask header, we can predict pixel-level segmentation masks for individual objects, making it particularly effective for tasks that require precise object localization and segmentation.

Mask R-CNN performs better compared to Faster R-CNN, especially for tasks that require detailed instance segmentation. It provides pixel-level masks, enabling precise object boundaries, a key feature in applications such as medical imaging. This model seamlessly combines object detection, classification, and segmentation tasks to simplify workflows and enable comprehensive instance-level detection. Its versatility extends to various fields such as robotics, self-driving cars, and image analysis. However, when choosing a model based on specific task requirements, it is important to balance these benefits against the high computational costs associated with pixel-level mask prediction.

Mask R-CNN has certain drawbacks, such as large computational costs, which prevent real-time applications even though it works well for instance segmentation. In Addition, because of its complex design and reliance on huge datasets with pixel-level annotations, the model is not well suited for usage in domains with sparse labeled data. When determining if deep neural networks are appropriate for a given job, one should take into account the difficulties with scale sensitivity, generalization to new classes, and interpretability that come with them.

IV. ACCURACY

Accuracy assesses overall correctness by calculating the ratio of correct predictions to the total instances.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (1)$$

Support: Support represents the actual instances in each class, giving insight into class distribution and potential imbalances.

Fig. 1. OpenCV

V. EXPERIMENTAL DATA AND FINDINGS

OpenCV: Using open CV we get the following image which shows an image detection that has an accuracy of 60 percent

Fig. 2. Faster R-CNN

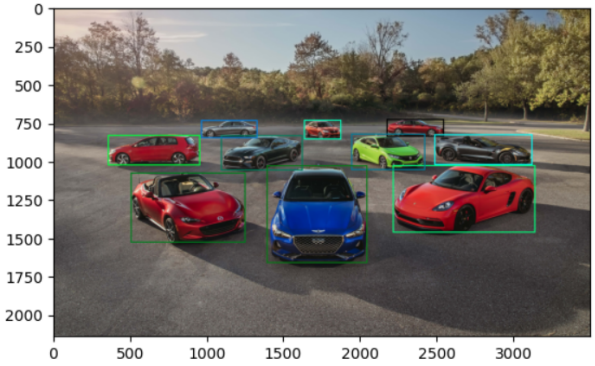

Fig. 3. RetinaNet

Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%
all	4	17	0.621	0.833	0.888	0.63
person	4	10	0.721	0.5	0.519	0.269
dog	4	1	0.37	1	0.995	0.597
horse	4	2	0.751	1	0.995	0.631
elephant	4	2	0.505	0.5	0.828	0.394
umbrella	4	1	0.564	1	0.995	0.995
potted plant	4	1	0.814	1	0.995	0.895

Fig. 4. Validation of model accuracy on the COCO dataset's test splits

```
Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
  1/3        0G      1.408      3.273      1.584         37        640: 100% 1/1 [00:03<00:00,  3.60s/it]
             Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:01<00:00,  1.50s/it]
               all          4         17      0.901      0.523      0.721      0.511

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
  2/3        0G      1.255      3.035      1.529         39        640: 100% 1/1 [00:03<00:00,  3.73s/it]
             Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:01<00:00,  1.49s/it]
               all          4         17      0.906      0.532      0.743      0.518

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
  3/3        0G      1.291      3.948      1.649         18        640: 100% 1/1 [00:02<00:00,  2.95s/it]
             Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:01<00:00,  1.20s/it]
               all          4         17      0.908      0.537      0.752      0.514
```

Fig. 5. Train the model on COCO8 dataset for 3 epochs

VI. Limitation

Firstly we only conducted the experiment for 3 epochs due to computational restrictions. There is the obvious scope of evaluating the model for a greater number of epochs which may provide a more reliable set of results. Secondly the experiment was carried out on the COCO dataset but the accuracy evaluation was not conducted for other available datasets. The model may produce a variation of results for alternate datasets.

VII. Conclusion and Future Work

In Conclusion, Object detection helping the visually impaired is a viable support system that allows for a more safe and secure lifestyle. This is a viable avenue to increase injury prevention and allow for more provisions in impasses of life. From our findings, we can see that this avenue of taking pictures in strenuous situations for the visually impaired is not only feasible but also practical and ensures a new safeguard for them. Our models demonstrate that the results are well supported and can be implemented in various applications such as a smartwatch mobile device or a pair of glasses.

There is immense scope for further experimentation and extension since only 3 epochs were used for the YOLOv8 model. Also it can be brought into question as to whether the model is more scalable or not. We only ran the models for the COCO8 dataset so it can be investigated whether the model will produce accurate results for other datasets which questions the generalization of the YOLOv8.

Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%
all	4	17	0.906	0.532	0.743	0.517
person	4	10	0.939	0.3	0.53	0.24
dog	4	1	1	0	0.249	0.129
horse	4	2	1	0.89	0.995	0.748
elephant	4	2	1	0	0.695	0.197
umbrella	4	1	0.755	1	0.995	0.895
potted plant	4	1	0.74	1	0.995	0.895

Fig. 6. The model's accuracy is then further validated

```
Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
  1/3        0G      1.166      1.461      1.243        294        640: 100%|██████████| 8/8 [01:58<00:00, 14.76s/it]
             Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:39<00:00,  9.88s/it]
               all        128        929      0.629      0.549      0.596      0.443

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
  2/3        0G       1.22      1.464       1.27        251        640: 100%|██████████| 8/8 [01:56<00:00, 14.61s/it]
             Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:40<00:00, 10.07s/it]
               all        128        929      0.613      0.578      0.614      0.461

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
  3/3        0G      1.127      1.385      1.218        158        640: 100%|██████████| 8/8 [01:53<00:00, 14.14s/it]
             Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:37<00:00,  9.32s/it]
               all        128        929      0.625      0.587      0.632       0.47
```

Fig. 7. The YOLOv8 is again trained and the performance is evaluated on the validation set for 3 epochs

References

[1] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.

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

[3] C. Kumar B., R. Punitha and Mohana, "YOLOv3 and YOLOv4: Multiple Object Detection for Surveillance Applications," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2020, pp. 1316-1321, doi: 10.1109/ICSSIT48917.2020.9214094.

[4] Z. -Q. Zhao, P. Zheng, S. -T. Xu and X. Wu, "Object Detection With Deep Learning: A Review," in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212-3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.

[5] M. R. Nehashree, P. Raj S and Mohana, "Simulation and Performance Analysis of Feature Extraction and Matching Algorithms for Image Processing Applications," 2019 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 2019, pp. 594-598, doi: 10.1109/ISS1.2019.8907990.

[6] N. Jain, S. Yerragolla, T. Guha and Mohana, "Performance Analysis of Object Detection and Tracking Algorithms for Traffic Surveillance Applications using Neural Networks," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2019, pp. 690-696, doi: 10.1109/I-SMAC47947.2019.9032502.

[7] S. H. Unger, "Pattern Detection and Recognition," in Proceedings of the IRE, vol. 47, no. 10, pp. 1737-1752, Oct. 1959, doi: 10.1109/JRPROC.1959.287109.