Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/57/20241306

GlassOnly: Transparent object dataset for object detection

Xinwei Mai

University of California, Santa Barbara

xmai@ucsb.edu

Abstract. Although datasets like ImageNet have a variety of classes for object detection, there are not many samples of transparents objects like glass walls, which are fully implemented in shopping malls and houses. The ignorance of transparent objects in object detection may cause potential danger to humans as the machines would not consider glasses as obstacles in path planning. Therefore, GlassOnly collected samples from malls and apartments and built a dataset for glass walls only. The dataset sample simulates a robot walking in human living environments from a perspective of the machine itself, providing data for studying detecting transparents objects.

Keywords: Deep Learning, Object detection, Transparent Objects, ImageNet, Open Images, COCO, GlassOnly.

1. Introduction

Developments of huge models and deeper networks facilitate deployments of robotics in different areas. From the application of auto-driving to service robots wandering in lobbies and halls, robotics with computer vision can save manpower from tedious work. Yet, with more advanced models and robotic technique, robotics have the potential to complete tasks in realms requiring precision and absolute focus, like acupuncture [1]. As human urges for more qualitative lives grows, robotics will be the central tool for substituting important manpower while doing a decent job.

Path planning is unavoidable in any robotic tasks, especially for autonomous land vehicles [2] like swiping machines in halls and lobbies as they walk in an open and complicated area with lots of walls and pillars. The distance between two points always comes with unwalkable obstacles in between. Once the robot has the local map in its memory (Simultaneous localization and mapping, or SLAM), it starts to predict the distance between the robot and available nodes [3]. Based on these predictions, it then calculates an efficient path to walk and avoid any obstacles that are scanned or detected [4]. If a robot does not have a function of object detection, it would be impossible to be deployed in real life situations for it might bring potential dangers to surrounding lives that are important to human society.

In general, Object detection has two parts: localization and classification [5]. Traditional methods use handcrafted features to localize objects within a picture, which is computationally expensive. Since deep convolutional neural networks (CNNs) have a strong ability to extract features, it can be used to overcome the weakness of traditional methods [5]. Models that do localization and classification separately are called two-stage detectors [5, 6, 7, 8]. The first two-stage detector is Region-based convolutional neural network (RCNN). It uses selective search to divide an input image into around 2,000 regions, then send those regions into a CNN to extract features. Based on the extracted features,

DOI: 10.54254/2755-2721/57/20241306

the support vector machine (SVM) classifier checks if any object is presented within each region. Finally, RCNN generates a more accurate boundary box for each identified object [7, 9]. However, RCNN's training is still costly and too slow for real life application. In order to improve the training cost and speed, researchers proposed new networks, such as spatial pyramid pooling and feature pyramids, and new structures, such as Faster RCNN and Mask RCNN [6, 7]. However, even with the newest feature pyramid network and mask RCNN, the detection speed is still slow for real time application.

On the other hand, detectors named one-stage detectors sacrifice accuracy but increase detection speed significantly by combining localization and classification into one task [5, 6, 7, 8]. For example, You Only Look Once (YOLO) divides an input image into an S x S grid. Then, a confidence score will be given to each grid after prediction. Every class has this score based on the predicted probability of existence, and the ones with high probability have a higher priority in drawing the boundary box. Then based on the size of the predicted object, YOLO determines the width and height of the boundary box within the grid [5, 10]. For the overlapping boxes, only the one with the highest IOU remains. With YOLO v5 proposed in 2020, one-stage detectors now have a detect speed in real-time while keeping a great accuracy [5]. However, for detecting low-resolution or small objects, even YOLO v5 can not outperform Faster RCNN [5]. Overall, if a task is focused on accuracy instead of speed, it would be better to use a two-stage object detector; otherwise, it would be better to use one of the YOLO versions based on the task difficulty and the size of datasets.

Backbone networks are the foundation of a model, regardless of the detector's type. They have evolved along with the development of object detection as well. Before the introduction of Residual Neural Networks in 2015, backbones like AlexNet and Visual Geometry Group Very Deep Convolutional Networks were shallow networks [7, 11, 12]. Since ResNets give a solution to the vanishing gradient problem [13], researchers started to make deeper networks and further increase the accuracy of detection. However, as the depth increases, networks are getting more complex and require more memory and time to train.

Most object detection research focuses on detecting non-transparent objects like people, plants, animals, or boxes that light can not go through. With proper architecture and datasets, current models can easily segment any tractable objects and value their importance to the path planning task. However, transparent objects like glass walls are also crucial for path planning as many shops chose glass walls to enclose their space while displaying their goods in malls. They can mislead a robot when doing path planning, especially objects like glass walls that separate two spaces but are hard to notice visually. It will become a problem for a machine to be implemented in a real life environment. To simplify the problem and start the investigation, GlassOnly provides a dataset that only focuses on glass walls, which will be defined as transparent objects that humans will call a piece of glass wall.

2. Background

Datasets for object detection have evolved correspondingly with the complexity of models, from PASCAL VOC 2007 with 2,501 training images and 6,301 annotated objects to Open Images Detection in 2018 containing 1,910,000 images with 15,440,000 annotated bounding boxes on 600 categories [6, 7]. Both datasets and models have fully developed in 2-D image classification, but their detection ability on 3-D objects or objects within a video is limited. Moreover, with the consideration of implementing object detection in an open-world situation, researchers are working on the ability to detect unknown categories that are not contained in the training sets and the speed of detection.

2.1. ImageNet

ImageNet contains around 14M images with annotations that follow WordNet hierarchy [14]. It has been a benchmark for image classification and object detection since 2010. Its annotations have two categories. One has an image-level annotation, describing both the existence and absence of a pair of object classes; the other has an object-level annotation, having a boundary box indicating object position within an image with one class label. Here is an example of the WordNet hierarchy within the dataset:

DOI: 10.54254/2755-2721/57/20241306

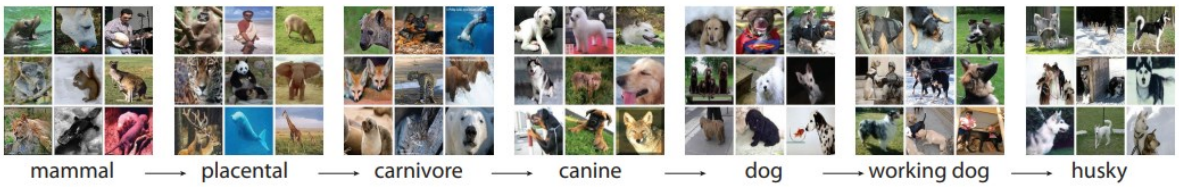

Figure 1. Example shows a hierarchy from the root (mammal) to the leaf [14].

2.2. Open Images-v7

Open Images originally contains around 9M images and 601 categories with image-level labels, object bounding boxes, and object segmentation masks (https://storage.googleapis.com/openimages/web/factsfigures_v7.html). As the dataset develops, the research team adds localized narratives, like audio and text, visual relationships, and point-level labels to enhance the learning ability of models by introducing more relevant information other than the object itself. Here is an example from the dataset (from embedded FiftyOne Dataset Zoo on pytorch):

<Detection: {
'id': '64f5ad5e1080c52dc0596ccc',
'attributes': {},
'tags': [],
'label': 'Building',
'bounding_box': [0.0, 0.903333, 0.054409, 0.09583399999999997],
'mask': None,
'confidence': None,
'index': None,
'IsOccluded': True,
'IsTruncated': True,
'IsGroupOf': False,
'IsDepiction': False,
'IsInside': False,

Figure 2. Example training image and its corresponding detection label (one out of six).

2.3. COCO 2020 Object Detection Task

COCO has different annotations for different tasks. For object detection, COCO values the importance of contextual information when detecting an object, so it uses either boundary boxing or object segmentation instead of image-level labels [15]. When constructing the dataset, the team wanted to focus on the most frequently discerned objects (like dogs, cats, cars, etc.) and therefore chose 80 selective categories and collected 0.2M images [15]. Here is an example from the dataset:

Figure 3. Example COCO segmentation [4].

DOI: 10.54254/2755-2721/57/20241306

3. Proposed dataset

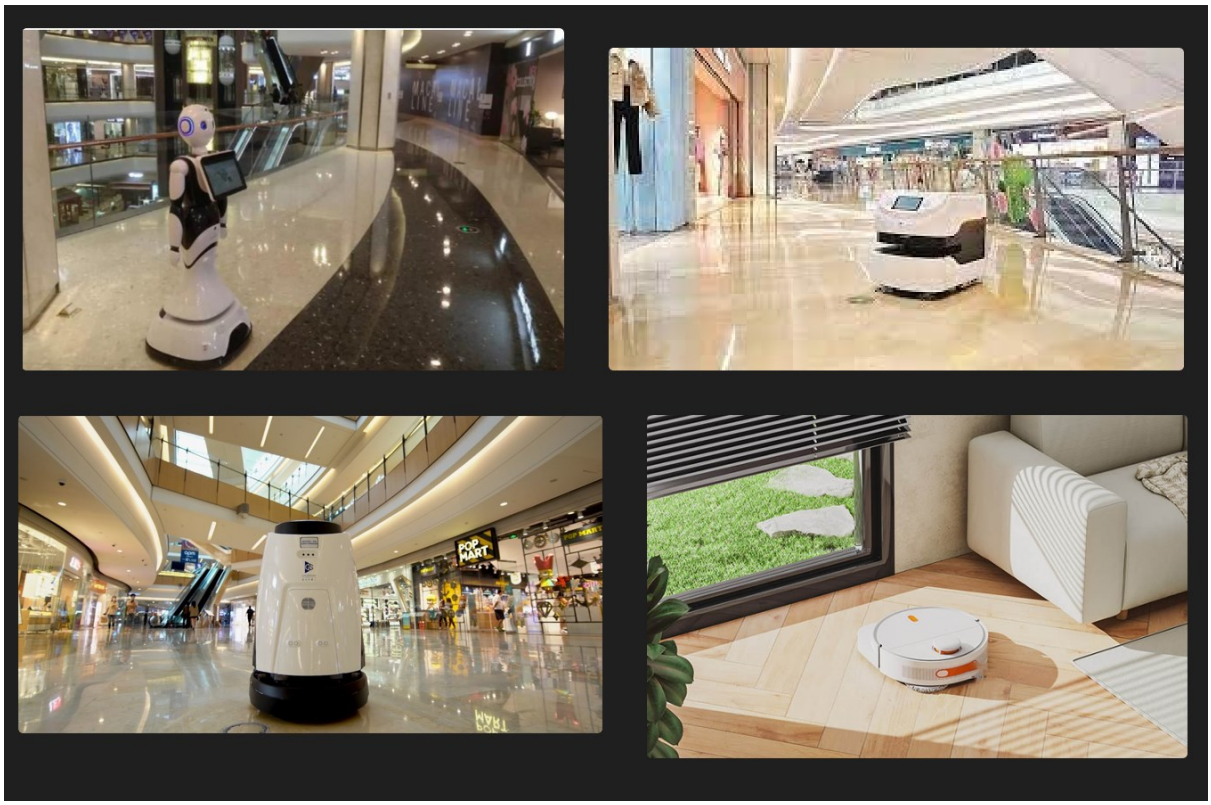

Figure 4. Glass walls are widely used in malls and houses.

3.1. Motivation

As Figure 4 has shown, glass walls are unavoidable objects in indoor environments. Without the ability to detect glass walls, robots may run straight into the glass wall because there are no obstacles in its path planning. Without frames, stains, and cracks, it will be difficult, even for humans, to discern whether a glass wall is in front of us or not. Still, one can detect transparent objects after a collision, but this does not coincide with the principle of causing the least potential harm to human society. The breaking of glass walls will cause more problems than directly crushing on people, including a possible bleeding accident which will not be tolerated by society. To avoid such collisions, a robot must learn to discern the existence of a transparent object and change its path planning based on the distance between them. The current samples within major datasets do not meet the necessity of learning discerning transparent objects. More datasets and models for transparent object detections are needed in order to deploy more robots more widely into daily lives. Therefore, this study proposed a dataset (GlassOnly) that focuses on only glass walls to trigger interest in this area.

3.2. Set up

The dataset samples are taken from a camera (vivo Y33s) on a cart that is approximately 58 cm above the ground and 43 cm away from the head of the cart. Here is a presentation of the cart:

DOI: 10.54254/2755-2721/57/20241306

Figure 5. Schematic diagram of sample taking cart (left) and the sample taking cart (right).

3.3. Dataset

GlassOnly(GO) has a total of 1074 pictures divided into three categories: 290 with glass wall within a meter, 270 with glass wall within three meters but beyond a meter, and 514 with glass wall beyond three meters or there are no glass walls. GO's annotations focus on the existence of glass walls within different ranges. Distance is the major concern here because of the importance of path planning in robotic missions, especially machines like swiping robots. This study wants to focus on glass walls that are close enough to have an impact on transportation only for a practical reason of reducing calculation power. It is not necessary for a robot to consider distant glass walls that it will never collide with. Here is an example of input and it annotation:

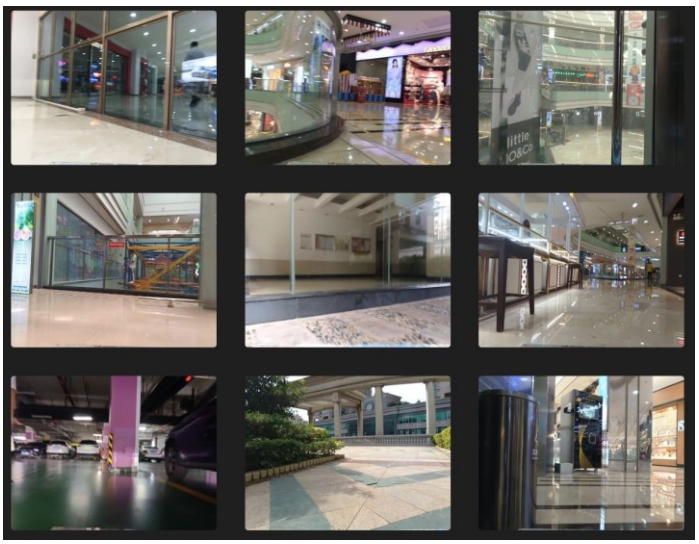

Figure 6. Example images. Sample's annotation consists of three parts: id, glass_existence, and distance. For glass_existence, 1 represents there is a glass wall that will have an impact on robot's path planning; on the other hand, 0 represents there are no glass walls that will have such an effect. As for distance, 1 represents a glass wall within one meter from the vision point; 3 represents it is within three meters but beyond one meter; 0 represents it is either beyond three meters or does not exist at all. In either case of 0 distance, glass walls will not have an impact on the robot's path planning.

3.4. Discussion

Due to the lack of manpower, here are some drawbacks of this dataset: 1. the types of environment that had been recorded are limited; 2. the distance between the viewing point and glass walls is ambiguous. There are three types of environments: indoor shopping mall, indoor apartment, and outdoor garden in a housing area. Moreover, these are typical Chinese fashion. In order to make a complete study on glass walls, the dataset should have included a worldwide range of both indoor and outdoor environments. The ambiguity of distance is a result of time constraint and a lack of technology. However, this

ambiguity only exists in some samples that are within three meters and beyond one meter. Most glass walls that are within one meter are clearly labeled. Overall, it does not undermine the goal of discerning glass walls that are close enough since all the existence of such walls are clearly annotated.

One good thing about this dataset is that the pictures were taken from a robotic perspective, which is normally much lower than a human perspective. The cart was monitoring the motion of a possible service robot and all the pictures were taken by the same camera, so the edge of vision is steady. Some samples were blurred due to the vibration while the cart was walking on an unsteady path, but this would be a good simulation of a robot actually walking.

4. Conclusion

GlassOnly samples simulate the movement of a land walking robot. The goal of it is to distinguish glass walls that are close enough to affect the machine's path planning ability. However, GlassOnly needs future advancements on both varieties of environments and sample quantity. Moreover, beyond the realm of transparent objects, highly reflective walls like mirrors that reflect light and form a clear image on their surface can also mislead a model to discern available working rooms. Overall, this study intends to bring up the problem of detecting such misleading obstacles and hopes there will be more well-established datasets in this area.

References

[1] T. W. Chan, C. Zhang, W. H. Ip and A. W. Choy, "A Combined Deep Learning and Anatomical Inch Measurement Approach to Robotic Acupuncture Points Positioning," 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Mexico, 2021, pp. 2597-2600, doi: 10.1109/EMBC46164.2021.9629761.

[2] E. Shang, B. Dai, Y. Nie, Q. Zhu, L. Xiao and D. Zhao, "A Guide-line and Key-point based A-star Path Planning Algorithm For Autonomous Land Vehicles," 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 2020, pp. 1-7, doi: 10.1109/ITSC45102.2020.9294336.

[3] S. Zhao and S. -H. Hwang 2021, "Path planning of ROS autonomous robot based on 2D lidar-based SLAM," 2021 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, pp. 1870-1872, doi: 10.1109/ICTC52510.2021.9620783.

[4] S. Klemm et al. 2015, "RRT*-Connect: Faster, asymptotically optimal motion planning," 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China, pp. 1670-1677, doi: 10.1109/ROBIO.2015.7419012.

[5] Ravpreet Kaur, Sarbjeet Singh, "A comprehensive review of object detection with deep learning," Digital Signal Processing, Volume 132, 2023, 103812, ISSN 1051-2004, https://doi.org/10.1016/j.dsp.2022.103812. (https://www.sciencedirect.com/science/article/pii/S1051200422004298)

[3] Z. Zou, K. Chen, Z. Shi, Y. Guo and J. Ye, "Object Detection in 20 Years: A Survey," in Proceedings of the IEEE, vol. 111, no. 3, pp. 257-276, March 2023, doi: 10.1109/JPROC.2023.3238524.

[5] A. B. Amjoud and M. Amrouch, "Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review," in IEEE Access, vol. 11, pp. 35479-35516, 2023, doi: 10.1109/ACCESS.2023.3266093.

[8] B. Fan, Y. Chen, J. Qu, Y. Chai, C. Xiao and P. Huang 2019, "FFBNet : Lightweight Backbone for Object Detection Based Feature Fusion Block," 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, pp. 3920-3924, doi: 10.1109/ICIP.2019.8803683.

[9] Girshick R, Donahue J, Darrell T, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

DOI: 10.54254/2755-2721/57/20241306

[10] Redmon J, Divvala S, Girshick R, et al. “You only look once: Unified, real-time object detection.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
[11] Simonyan, Karen, and Andrew Zisserman 2014. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556.
[12] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton 2012. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems 25.
[13] He, Kaiming, et al 2016. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition.
[14] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei 2009"ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.
[15] Tsung-Yi Lin and Michael Maire and Serge Belongie and Lubomir Bourdev and Ross Girshick and James Hays and Pietro Perona and Deva Ramanan and C. Lawrence Zitnick and Piotr Dollár, Microsoft COCO: Common Objects in Context, arXiv, 2015, lin2015microsof
[14] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei 2009"ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.
[15] Tsung-Yi Lin and Michael Maire and Serge Belongie and Lubomir Bourdev and Ross Girshick and James Hays and Pietro Perona and Deva Ramanan and C. Lawrence Zitnick and Piotr Dollár, Microsoft COCO: Common Objects in Context, arXiv, 2015, lin2015microsof