Obstacle Detection and Monocular Distance Estimation on a Mobile Phone for the Visually Impaired and Blind

Aaron Raymond See[1*], Monching Desierto[2], John Jefferson Sison[2], Chris Jordan Aliac[2]

ABSTRACT

In recent years there have been developments of assistive devices for the visually impaired and blind that uses computer vision. However, carrying extra devices can be a burden. Thus, we propose the use of a mobile phone application to detect obstacles and also determine the distance of these objects. The object detection model is instantiated from the MobileNet SDD v2 architecture which is designed for low-performance devices. On the other hand, we derived our depth estimation from the pinhole camera model to estimate the distance of the objects and achieve monocular depth estimation. The overall model returned a satisfying result of 0.19 m RMSE. Furthermore, object detection has an accuracy of approximately 73%. In addition, the said model is designed enough to handle depth estimation on real-time images and sudden camera tilt when the user is moving.

Keywords: object detection, distance measurement, machine learning, mobile phone application

I. INTRODUCTION

Blindness, vision loss, low vision, and people with visual impairment are terms generally used for people with visual disabilities that can affect a person's life. The World Health Organization states that 2.2 billion people around the world have vision impairment [1]. The impairment is not limited to any individual, given that a healthy person may lose their eyesight due to an accident or any unfortunate events that may lead to vision loss. To compensate for the loss of a person's vision, they seek assistance from others or rely on their other senses, such as hearing and touching. Assistive technologies are designed to assist people with disabilities to promote independence and improve the person's lifestyle.

*Corresponding Author: Aaron Raymond See (E-mail: aaronsee@stust.edu.tw).
[1]Department of Electrical Engineering, Southern Taiwan University of Science and Technology, 1, Nantai St., Tainan City, 710301, Taiwan
[2]College of Computer Studies, Cebu Institute of Technology - University, Natalio Bacalso Ave., Cebu City, 6000, Cebu City, Philippines

Many assistive technologies exist that helps and assists the blind in day-to-day events. The researchers propose an idea that helps the visually impaired navigate the surroundings by detecting objects and informing the distances of the objects via a smartphone. The combination of machine learning and mathematical models are used to achieve the said proposal.

Images captured by smartphone cameras are helpful for blind navigation. Given the object is captured by the camera, specifically the foot of the object, the distance is possible to be calculated if the angle of view and height of the camera from the ground is known. The smartphone can be hung on the neck of the user for easier use. The following are the contribution of the proposed idea:

1. Object-depth estimation designed for smartphone devices to assist the blind in navigation.
2. Two-way sequential process: object detection and depth estimation.

II. RELATED WORKS

In this section we will present models that are used in object-depth estimation and their functionalities. This also serves as a guide in choosing the right model to reach the desired results.

A. Object Detection: Object detection is a branch of computer vision that deals with localization and identification of an object [2]. The following are various object detection methods using artificial intelligence. First, TensorFlow Model Zoo is a collection of pre-trained object detection architectures, the model architecture included CenterNet, a deep Convolutional Neural Network (CNN) that is trained to detect each object as a triplet [3]. EfficientDet is a model that is built to scale up efficiency in computer vision [4]. Then there is MobileNet, a CNN architecture designed for mobile and embedded vision applications [5]. Another type is the RetinaNet which is a Feature Pyramid Network that generates a multi-scale convolutional feature pyramid on a feed-forward ResNet architecture [6]. The R-CNN is a 2-staged object detection architecture that sends region proposal down the pipeline for object classification and bounding box regression on the first stage [7]. ExtremeNet is an object detection framework that detects four extreme points of an object by

detecting four multi-peak heatmaps for each object category [8]. There are 2 versions of MobileNet SSD and their main difference is their convolutional layers that determines how expensive or costly they can be for programmers [9]. The first version has a depthwise convolutional layer which filters the inputs and followed by 1x1 pointwise convolutional layer that combines these filtered inputs [9]. The second version has 3 convolutional layers, instead of having the pointwise convolutional layer where it keeps the channels the same or, it doubles the channels, the second version has a layer called the projection convolutional layer where it makes the number of channels smaller, where it projects data with high numbers of channels into a tensor with a much lower number of channels [9]. The newest layer for the latest version of MobileNet SSD is the expansion layer. This expansion layer expands the number of channels in the data before it goes to the depthwise convolution, then the new 1x1 projection layer [9].

B. Pinhole Camera Model: The pinhole camera model is used to determine the coordinates of pixels to approximate 2D images from a 3D environment [10]. The model is composed of 3 parts: the 3D object, the pinhole or camera center, and the image. The perpendicular distance between the image and the pinhole is the focal length while the line itself is called the optical axis [11]. The model is usually represented as a closed box with a single tiny hole called aperture where the light enters and hit the photosensitive surface inside the box [10, 11]. Using this model will capture an image inversely in terms of x and y position in 2D plane.

C. Depth Estimation using Monocular Image: There has been two ways of estimating the depth of the objects inside the monocular images: using mathematical equations [12, 13, 14] or using artificial neural networks [13, 15, 16]. Mathematical equations consider the characteristics of the camera such as field of view and sensor. On the other hand, artificial neural networks learn how depth is the object using a large amount of image datasets to predict a new set of images. [12] uses mathematical equation to estimate distance using the motion of the bounding boxes of an object. [2] stores default bounding box sizes per object class. Together with the bounding boxes of the new image, [2] can estimates the bounding box of the new image using an artificial neural network. Griffin and Corso (2021) demonstrated good results in depth estimation using artificial neural networks [15]. In addition, there are monocular cues that can be used to estimate the depth of the object like texture variations and gradients, defocus, and color or haze [13]. If the monocular cues are used, the process requires consideration of the global structure of the image. Pixel size of the object can also be used as a cue which is similar to the model of [2].

III. METHODOLOGIES

A major step to blind navigation is to detect the objects around and then determine the distances of each object. The proposed model uses a smartphone device for assisting the blinds. The overall model is divided into two sequential processes: object detection and depth estimation. Object detection uses the camera of the device to capture a real-time image in front of the user and outputs the bounding boxes of the objects together with the labels. On the other hand, depth estimation needs the tilt angle of the device, the height of the camera lens from the floor, and the position of the objects in the captured image to estimate the depth of each object. Depth estimation relies on the vertical position of the objects from the image. Thus, the horizon line [17] plays a vital role in the estimation. Figure 1 shows real-time detection and estimation using a mobile phone.

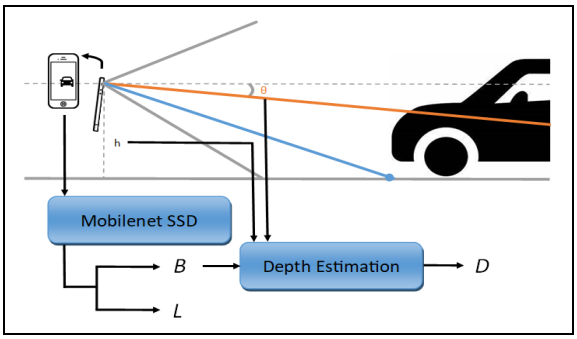


Figure 1. The smartphone captures a real-time image and inputs the image to the object detection model to generate bounding boxes B and labels L. Using the position of the objects in the image together with the camera height h and angle of tilt θ, distance D can be estimated.

A. Object Detection

The object detection uses SSD MobileNet architecture as it is designed to work on low-performance devices compared to desktop computers. The architecture is a CNN that accepts an image as the input and outputs the bounding boxes and labels of the detected objects. The bounding box composes two image coordinates: x_1, y_1, x_2, and y_2, where the first coordinate points to the top left edge of the box or rectangle while the second points to the bottom right edge. An instanced model will be created and trained and then converted to the TensorFlow Lite version for the model to be compatible with mobile devices.

B. Depth Estimation

Depth estimation is done mathematically using trigonometry and the pinhole camera model. The angle of view and the height from the floor of the camera are needed to calculate the basis of measuring the depths of the objects detected from the real-time image from the base of the camera. In addition, the horizon line also plays a vital role in the model. The closer the objects to the horizon line, the farther the objects are in the real world.

The model is designed to determine the depth of the objects on the floor, therefore, the objects which feet are below the horizon line are only to be estimated. Horizon lines are expected to be at the center of the image when the camera is facing parallel to the floor. Hence, half of the image can be measured as one image unit and can be

focused on calculation. When objects are detected in the images, the distance of each object in the real world is inversely proportional to the distance of the object from the horizon line. To generate the formula, the depth of the closest object must be determined first using

$$D_0 = H/\tan(A/2),$$

where D_0 is the depth of the closest object, which is at the lower edge of the image, H is the height of the camera from the floor, and A is the vertical angle of view of the camera. D_0 is also equivalent to the focal length of the pinhole camera model when H is 1 distance unit. The image-focal and H-D_0 pattern forms 2 similar triangles, thus, if the distance of the object to the horizontal line is a image unit, then the depth of the object D is D_0H/a.

The formula above only works if the horizontal line is at the middle of the image. If the smartphone is tilted, the horizon line must also be shifted. Likewise, when the camera is facing parallel to the ground, the horizon line is expected to be at the middle of the image, but when the camera tilts, the horizon line changes its position. Figure 2 shows a sample of the model to determine distance and figure 3 displays the image of the object that shifts the horizon when tilted. The shift in the position of the horizon line a_1 can be calculated as $D_0\tan(\theta)$, where θ is the angle of tilt. Tilting the camera downward shifts the horizon line higher in the image. Recall that starting from the center of the image to the below half is measured as 1 image unit, therefore, the height of the image is 2 image units and the topmost can be labeled as -1 while the lowermost as 1 giving 0 at the center. Accordingly, the new position of the horizontal line M would be $-a_1$. On the other hand, the new distance of the object to the horizontal line a_2 is now y_2-M, where y_2 is the vertical position of the foot of the image, thus the end formula is either of the following:

$$D = D_0H/a_2, D = D_0H/y_2-M, \text{ or } D = D_0H/y_2+a_1.$$

The limitation of the proposed model is that the model cannot accurately estimate the depth of the object when the foot is not captured, hence, objects that are behind another object, floating or not on the ground, or objects that are on the different floor of the smartphone are expected to have some discrepancies in measurement.

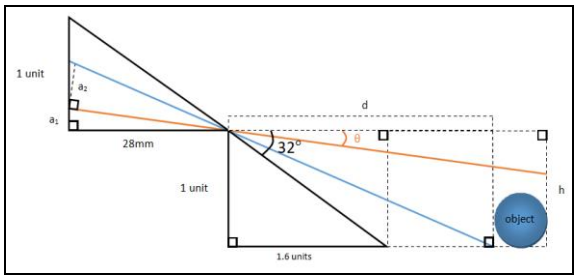


Figure 2. Derived pinhole camera model with 28mm of focal length. Half the angle of view is 32˚and generates 1.6 units of distance (d) if the height (h) is 1 unit. The pitch of the mobile phone (θ) is also a factor in determining the distance of the foot of the object on the ground.

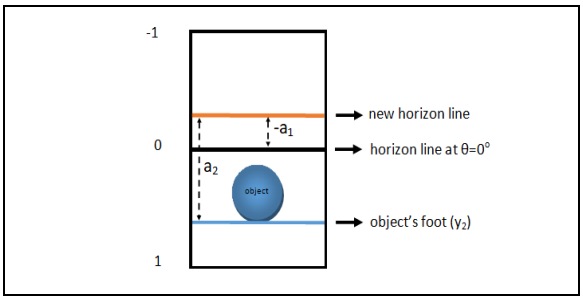


Figure 3. Whole image from the pinhole camera. The middle value goes up when the camera tilts downward. The bigger the gap between the middle and the object's foot, the nearer the object is.

IV. EXPERIMENT AND ANALYSIS

A. Setup

The object detection model is instantiated from a trained model and is retrained in Google Colaboratory as the environment. The model is retrained to filter unnecessary objects and to focus on the objects that are to be detected. Furthermore, the instantiated architecture is the second version of MobileNet SSD for the reason that the architecture requires lesser multiply-accumulate operations on every image and uses fewer parameters which gives more speed to low-end devices compared to the first version. The model is trained until the least error is evaluated and is then converted into a tflite model together with the metadata for it to be compatible with mobile devices.

The depth estimation is to be tested on the smartphone device with an Android platform. Furthremore, Tensorflow has already some templates for tflite model deployment. The proposed depth estimation model is just inserted into the source code and customized the output to show the bounding boxes and distances of the object. The device used in testing is a VIVO Y20s [G] phone with its rear camera. Finding the angle of view of the camera is $A = 2\arctan(d/2f)$, where d represents the size of the camera sensor and f is the focal length. In addition, the angle of tilt needed to find the value of a_1 is not needed anymore. Android devices have gravity sensors that can directly determine the value of a_1, in particular, the gravity on the y and z-axis. Y-axis points to the top of the device while the z-axis points in front. Using the ratio of similar triangles as shown in figure 3, vectors z and y are directly proportional to a_1 and D_0. When the device is perpendicular to the ground, y is 9.8 m/s^2 and z is 0 m/s^2, then a_1 is equal to 0 image units. Figure 4 exhibits the relationship of to calculate the various axes in the equation.

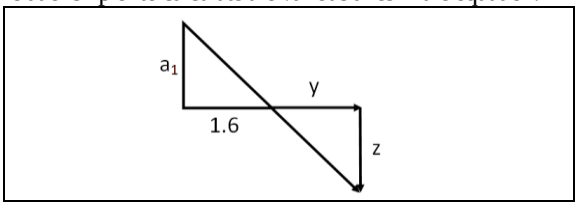


Figure 4. Direct computation of a1 with a D0 of 1.6 units using the gravity acceleration at the y-axis and z-axis of the mobile phone.

B. Dataset

The dataset for the object detection model was gathered manually. Images were taken in real-time, and some were gathered in Google Image and then resized to 640x480 pixel size in preparation for training. Overall, 200 images were annotated with 16 labels in total. The labels include persons/people, vehicles such as cars and buses, pedestrian lanes, traffic lights, tree and tree branches, plants, boxes, dogs, walls, bikes, cans & bottles, carts, and poles. A small test dataset is also prepared that contains 22 images.

C. Results

The object detector model was trained until the combination of regression and classification loss is 1.94. The objects were accurately labeled with over 70% when tested with the test dataset. The model will be improved in subsequent applications.

After converting the object detector into a tflite model together with the metadata, the Android application was built together with the depth estimation model. The application was tested on the neighborhood with a person and a car as the objects. The measurement used was in feet (ft), 0.3048 in meters (m), for the reason that the metric is more suitable for certain applications. Data was gathered in a combination of the camera height, tilt, real and estimated distance, and a boolean value if the foot the object is captured or not. Capturing the foot of the object can tell that the estimation is accurate. The gathered data is limit to 15ft or 4.572m and two tilt angles, 0 and -45 degrees. The gathered data is then analyzed by using the root mean squared error (RMSE) to measure how spread the residuals or errors are and by comparing how accurate the estimations are. Sample results are shown in figure 6.

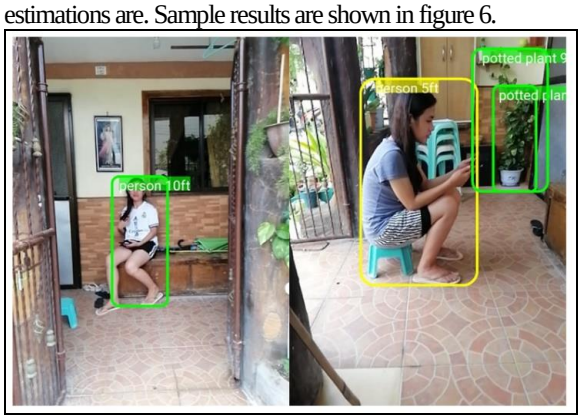


Figure 6: Sample output of the Android application.

The raw gathered data was to be filtered first before calculating the RMSE. Data on which feet of the objects are not captured are not needed since depth estimation relies on where the position of the foot in the image is. Out of 22 filtered data, the analysis gave a result of 0.64ft or 0.19m overall RMSE. Test results showed 73% detection accuracy. Furthermore, it was observed that the gathered data with 2-ft distances were less accurate because of the limitation of the angle of view of the camera. The camera needs to be tilted at least 45 degrees downward for the camera to capture the foot of the object given that the camera is at least 3ft above the ground

V. CONCLUSIONS AND FUTURE WORK

The mobile phone based object detection and monocular depth estimation for blind navigation assistance was successfully implemented using a smartphone device. The model was able to handle depth estimation using real-time images and sudden camera tilt while the user is moving. Object detection is instantiated from the MobileNet SSD v2 architecture and got 70% accuracy in detecting objects. Subsequently, depth estimation is derived from the pinhole camera model. The angle of view of the camera and the height from the ground of the camera are also needed to estimate the depth of the objects in the image. The overall model gave a satisfying result of 0.64ft or 0.19m RMSE and 73% accuracy at distances upto15ft or 4.572m. The result is good enough to assist the blind in navigation that can detect objects and estimate their distances. Although the proposed model has shown success, there is still room for improvement for depth estimation. The limitation of the model relies on the angle of view and height of the camera. Therefore, the model cannot estimate objects that are not wholly captured especially the feet of the objects. A new major process must be added to handle the limitations of depth estimation.

ACKNOWLEDGMENTS

This work was supported in part by the Ministry of Science and Technology (MOST), Taiwan, under Grant MOST 109-2222-E-218-001–MY2 and Ministry of Education, Taiwan, under grant MOE 1300-108P097.

REFERENCES

[1] WHO, “Blindness and Vision Impairment,” The World Health Organization, 2021. https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[2] O. Elisha, “Real-time object detection using SSD MobileNet V2 on Video Streams,” in Heartbeat, Exploring the intersection of mobile development and machine learning. Sponsored by Fritz AI, 2020.

[3] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “CenterNet: Keypoint Triplets for Object Detection”, in proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, South Korea, 2019, pp. 6568-6577.

[4] M. Tan, R. Pang, and Q. Le, “Efficient Det: Scalable and Efficient Object Detection,” in proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Washington, USA, 2020, pp. 10778-10787.

[5] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” in arXiv, 2017, pp. 1-9.

[6] T. -Y. Lin, P. Goyal, R. Girshick, K. He and P.

Dollár, "Focal Loss for Dense Object Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 318-327, Feb. 2020.

[7] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, Ohio, USA, 2014, pp. 580-587.

[8] X. Zhou, J. Zhuo and P. Krähenbühl, "Bottom-Up Object Detection by Grouping Extreme and Center Points," in the proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, California, USA, 2019, pp. 850-859.

[9] M. Hollemans, "MobileNet Version 2," in MachineThink, 2018. Accessed: Aug. 5, 2021. [Online]. Available: https://machinethink.net/blog/mobilenet-v2/

[10] P. Sturm, "Pinhole Camera Model," in Computer Vision: A Reference Guide, K. Ikeuchi, Ed. Boston, MA: Springer US, 2014, pp. 610–613.

[11] K. Hata and S. Savarese, "CS231A Course Notes 1: Camera Models", 2017. Accessed: Aug. 10, 2021. [Online]. Available: https://web.stanford.edu/class/cs231a/course_notes/01-camera-models.pdf

[12] B. A. Griffin and J. J. Corso, "Depth From Camera Motion and Object Detection," in the proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 2021, pp. 1397–1406.

[13] K.K. Tiwari, "A Polynomial Based Based Depth Estimation from a Single Image", Cornell University, 2010. Accessed: Aug. 10, 2021. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1011/1011.5694.pdf

[14] F. M. Dirgantara, A. S. Rohman, and L. Yulianti, "Object Distance Measurement System Using Monocular Camera on Vehicle," in proceedings of the 2019 6th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Bandung, Indonesia, Sep. 2019, pp. 122–127.

[15] J. Zhu and Y. Fang, "Learning Object-Specific Distance from a Monocular Image," in the proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), , Seoul, South Korea, 2019, pp. 3838-3847.

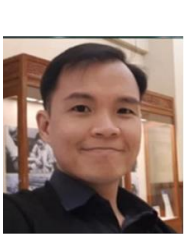
[16] M.A. Haseeb, J. Guan, D. Ristić-Durrant, and A. Gräser, "DisNet: a novel method for distance estimation from monocular camera," in the proceedings of the 10th Planning, Perception and Navigation for Intelligent Vehicles (PPNIV18), IROS (2018), Madrid, Spain, 2018, .

[17] T. Ahmad, G. Bebis, M. Nicolescu, A. Nefian and T. Fong, "An Edge-Less Approach to Horizon Line Detection," in the proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, Florida, USA, 2015, pp. 1095-1102.

Dr. Aaron Raymond See is an associate professor in the Department of Electrical Engineering, Southern Taiwan University of Science and Technology. His research interests are in assistive device design and development, biomedical image and signal processing, and engineering education.

Monching Desierto is a student at Cebu Institute of Technology – University taking Bachelor of Science in Computer Science. His research interests are in computer simulation and image processing.

John Jefferson Sison is a Bachelor of Science in Computer Science in Cebu Institute of Technology – University. His research interests are in artificial intelligence, game studies, and medical research software.



Dr. Chris Jordan Aliac is a professor at Cebu Institute of Technology – University as well as the manager of the University's Makerspace and an ICT security consultant. His research interests are in artificial intelligence, distributed systems, and ICT security.