HIGH SCHOOL EDITION
Journal of Student Research
Volume 10 Issue 4 (2021)

GuideDogNet: A Deep Learning Model for Guiding the Blind in Walking Environments

Yunseo Hwang[1], Taeseon Yoon[#], and Kyuyong Park[#]

[1]Hankuk Academy of Foreign Studies, South Korea
[#]Advisor

ABSTRACT

A guide dog is a critical companion for the blind, which enables independent travel of the blind. However, due to the costly and time-consuming training process, only 1.7% of the blind who wish to adopt a guide dog can take it. In order to alleviate this social problem, previous studies have suggested several blind guiding systems heavily based on hardware devices, such as GPS(Global Positioning System), RFID(Radio-Frequency Identification), and ultrasonic devices. However, those techniques lack administrative feasibility to use in real-world environments. Moreover, those techniques are deficient in warning of obstacles, which makes the system non-user-friendly. To guide the blind universally and provide accurate information about the obstacles without cumbersome devices, we propose a novel deep learning-based blind guiding system, GuideDogNet. The proposed system consists of an object detection network, depth prediction network, and post-processing module. To provide user-friendly outputs for the blind, we propose a rule-based post-processing module that outputs the label, direction, and distance of the obstacles by combining the results of the object detection network and the depth prediction network. We achieved an mAP of 67.8 on the AI Hub Sidewalks dataset which is publicly available. To the best of our knowledge, this is the first attempt at a deep learning-based blind guiding system.

The code will be available on https://github.com/Yunseo-Hwang/AI-GuideDog

Introduction

A guide dog is a critical companion for the blind as it provides mobility and enables the independent life of the blind. There are 37,000 people with severe visual impairment in Korea, and 3,400 people expect guide dogs' help. However, only 1.7% of the blind who wish to adopt a guide dog can take it. Guide dog training involves challenges that remain to be addressed: (1) costly and time-consuming training process, (2) limited guidable sphere, and (3) difficult management for the blind.

The guide dog training process lengthens up to 18 months and costs up to $42,000 for each dog. [1] Even after a costly and time-consuming training process, only 30% of the dogs qualify as guide dogs. The lack of guide dogs is a great challenge for the blind, as guide dogs work as the eyes for them. In addition, the limitation of guide dog competence is another challenge for the blind, restricting their living sphere. Guide dogs can only guide limited pre-trained areas. To take a new path, one needs an assistant who can newly train the dog. This hinders the independent life of the blind. The primary breed for the guide dog is Labrador Retriever. The Labrador Retriever weighs up to 36 kilograms, which is difficult for the blind to take care of.

So far, guide dogs and canes have become the eyes for the blind. However, due to the remaining challenges of guide dogs, several techniques were introduced in recent years on behalf of guide dogs and canes. Recent techniques guiding the blind are (1) GPS(Global Positioning System), (2) RFID(Radio Frequency Identification), and (3) Ultrasonic devices.

The GPS-based technique [2] uses location information to guide the user to his destination. It can guide the user to any destination with location information without training the route in advance. However, it cannot

detect destinations or obstacles without location information. Extensive facilities with location information, such as bus stops, crosswalks, and buildings, are recognized, but small obstacles such as people, bicycles, and benches are not identified. When a blind face a bench or a post without location information while walking, the GPS method does not take a detour or give any warning. Also, it fails to guide indoor destinations without location information. The RFID-based technique [3] consists of a smart floor and a portable terminal unit. The smart floor has a passive RFID tag built into it, which translates unique ID numbers. A portable terminal unit functions as an RFID reader and guides the user to the destination. However, the RFID method is not universally available anywhere. It is only available in places with RFID tags and preinstalled maps. Building infrastructures for the RFID method is expensive and inefficient. The ultrasonic sensor method [4] detects any obstacle within a specified range. It warns of dangers with voice or vibration. However, it cannot identify the type of obstacle, provides limited information about the obstacle.

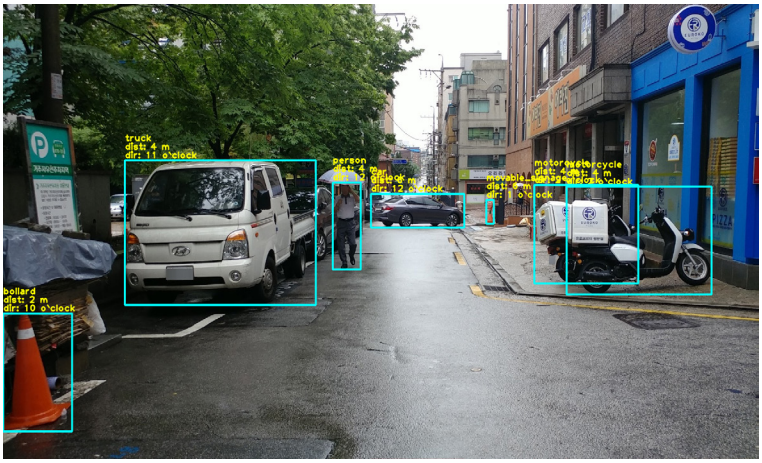

Figure 1. Result of GuideDogNet. The proposed system shows the label, distance, and direction of the obstacles from the RGB image.

To address these problems, we propose a novel deep learning-based blind guiding system, GuideDogNet, that is not only hardware-free but also user-friendly. The proposed system provides various information, including the label, distance, and direction of obstacles to the users. To develop the system, we exploit two recently released studies DETR [5] and DPT [6] for object detection and depth prediction, respectively. In addition, we propose a post-processing module that outputs the label, distance, and direction of obstacle objects as shown in Fig. 1.
Contributions of this paper are summarized as:

1. We proposed a novel deep learning-based blind guiding system. To the best of our knowledge, this is the first study to combine object detection and depth prediction results to make the system user-friendly.
2. We proposed the post-processing module that outputs various information; directions, distances, and labels of the obstacles.

Related Work

Deep learning-based algorithms have shown very successful results in many computer vision problems. In this paper, we apply two well-known deep learning tasks, object detection, and depth prediction, to develop the proposed system.

2.1 Object Detection

Object detection performs classification and localization of an object and outputs labels and binding boxes as a result. Object detection has two major approaches: two-stage detection and one-stage detection. Two-stage detection methods perform classification and localization sequentially. Due to their architectures, there is a bottleneck between the states and it makes the processing time very slow. R-CNN [7] and Fast R-CNN [8] are major two-stage detectors. On the other hand, one-stage detectors such as YOLO [9] and SSD [10] perform classification and localization simultaneously. These methods perform relatively faster but yield poor results compared to the two-stage methods.

Recently, there have been many studies about applying transformer architectures to computer vision problems. Inspired by this, Xizhou Zhu et al. proposed DETR(Detection Transformer) [5], an object detection network with a transformer structure. DETR dramatically reduced the processing time as it removed the hairy hand-craft post-processing by solving the object detection as a set prediction problem. In the proposed system, we exploit DETR to detect obstacle objects in the given RGB images.

2.2 Depth Prediction

Depth prediction, also known as 3D reconstruction, estimates the depth map from the input RGB images. Traditionally, depth prediction networks are mainly based on CNN [11][12]. However, conducting depth prediction networks using CNN requires deep neural layer depths, which increases computation volume. Recently, the transformer structure is also applied to the depth prediction problem. René Ranftl et al. proposed DPT(Dense Prediction Transformer) [6]. They experimentally proved that DPT effectively decreased computational cost while preserving comparable depth prediction accuracy. In the proposed system, we exploit DPT to generate the depth map from the input images.

Methods

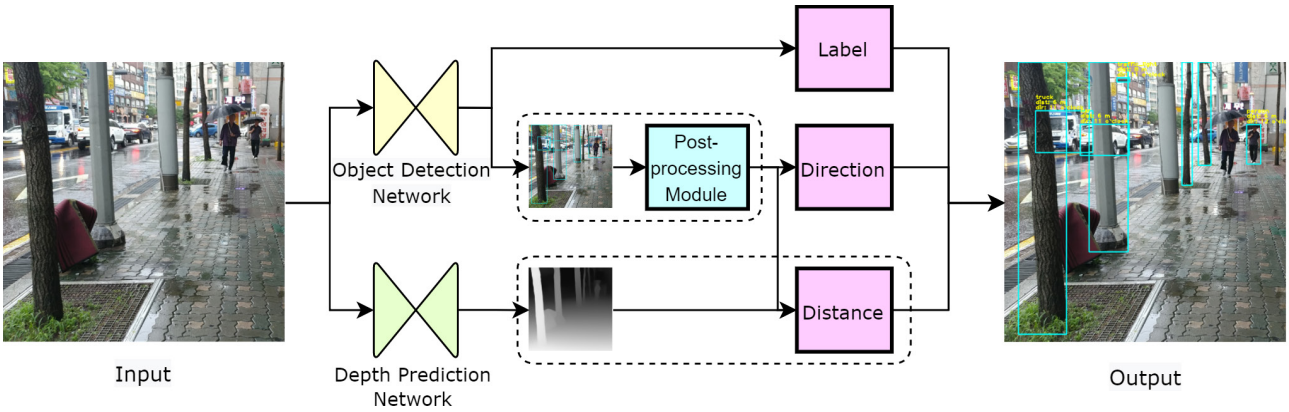

Figure 2. Architecture of GuideDogNet. The proposed system consists of an object detection network, depth prediction network, and post-processing module. It outputs the label, distance, and direction of the obstacles.

Fig. 2 shows the architecture of the proposed blind guiding system, GuideDogNet. The proposed system consists of an object detection network, depth prediction network, and post-processing module. The object detection network detects obstacle objects and the depth prediction network predicts a depth map in given RGB

images. The post-processing module takes the detection results and depth map as input and outputs the label, direction, and distance of the obstacles.

3.1 Object Detection Network

DETR(DEtection TRansformer) [5] is the state-of-the-art object detection network using transformer structure, generally composed of a backbone, encoder, decoder, and prediction heads. DETR considers object detection as a set prediction problem that does not require a hand-craft bounding box regression module and duplicate removal process which often causes a bottleneck. In this paper, we exploit the DETR structure to develop the object detection network as it has a relatively low computational cost while preserving comparable accuracy.

The backbone network takes input samples and produces 2D image features as output. They are converted into 1D vectors and then fed to the transformer encoder after concatenated with positional encodings. The transformer decoder takes the output of the encoder with additional learnable queries and produces the bounding boxes and their object labels. In this paper, we train the network with 30 classes, including 29 kinds of objects and backgrounds. Eq (1) and Eq (2) shows the loss function used to train the network.

Equation 1: Cross-Entropy Loss function:

$$L_{CE} = -\sum_{i=1}^{n} t_i \log(p_i) \ \ for\ n\ class$$

$$where\ t_i\ is\ the\ truth\ label\ and\ p_i\ is\ the\ Softmax\ probability\ for\ the\ i^{th}\ class.$$

Equation 2: L1 Loss function:

$$L1LossFunction = \sum_{i=1}^{n} |y_{true} - y_{predicted}|$$

3.2 Depth Prediction Network

Generally, the existing depth prediction methods need many neural network layers to produce accurate prediction results. However, this essentially makes the trained network have a high computational cost.

DPT(Dense Prediction Transformer) [6] is a depth prediction network built on a transformer structure. As it has a relatively lower computational cost while preserving comparable accuracy, we exploit the method to predict the depth map of the input images. DPT splits 2D images into small image patches and each patch is flattened into a 1D shape and then fed into the transformer encoder and transformer decoder. The decoder predicts the final depth map and it is used to predict the distance of the obstacle objects detected in the object detection module.

3.3 Post-processing module: location-aware depth and direction prediction module

The proposed post-processing module combines the detected object and depth map information and produces the label, distance, and direction of the objects. The distance and direction of the obstacles are calculated from the bounding box coordinates and the depth map.

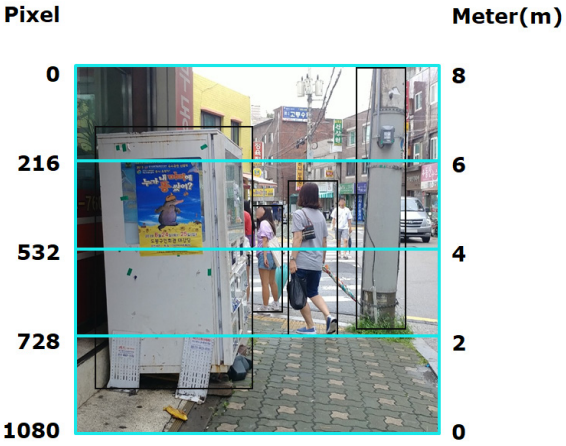

Figure 3. Overview of post-processing module calculating the distance of the obstacle from the bounding box coordinates

Fig. 3 shows the overview of the post-processing module calculating the distance of the obstacle, get distance function. The post-processing network uses the bounding box coordinates to output the direction and distance of the obstacles. The get distance function outputs the distance of obstacles in meters. Y-coordinate of the bottom-right point of the bounding box is the input value. Depending on the value of the y-coordinate, the distance of the obstacle is output to 2 m, 4 m, 6 m, and 8 m. The distance is verified if the distance obtained from the bounding box coordinate is consistent with the distance calculated from the depth map.

Table 1. Pseudocode of get distance function.

Input: y-coordinate of the bottom-right point of the bounding box, ybr

Output: distance in meters

if ybr < 216 **then** 8 meters

else if ybr < 532 **then** 6 meters

else if ybr < 728 **then** 4 meters

else 2 meters

end if

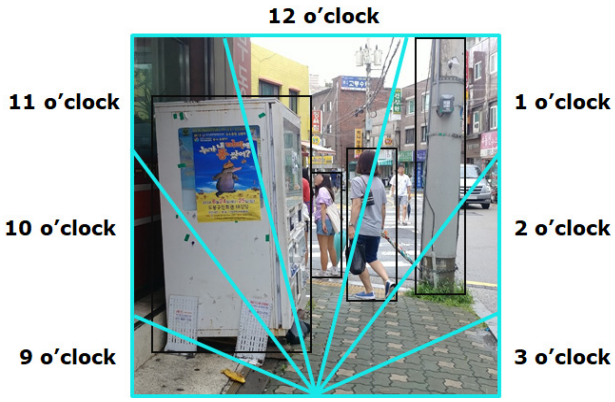

Figure 4. Overview of post-processing module calculating the direction of the obstacle from the bounding box coordinates

Fig.4 shows the overview of the post-processing module calculating the direction of the obstacle, get direction function. The get direction function outputs the direction of an obstacle in clock position. The bounding box coordinates are the input, and the direction is the output. The direction of the obstacles is displayed at 9 to 12 and 1 to 3 o'clock, depending on the location of the bounding box. The coordinates of the bounding boxes are normalized. The origin is the location of the user and at (0.5, 1). Using the slope between the origin and the center of the bounding box, the position of the obstacles is indicated.

Table 2. Pseudocode of get direction function

Input: xtl, ytl, xbr, ybr xtl: x-coordinate of the top-left point of the bounding box ytl: y-coordinate of the top-left point of the bounding box xbr: x-coordinate of the bottom-right point of the bounding box ybr: y-coordinate of the bottom-right point of the bounding box **Output**: direction in clock position Normalize coordinates of the bounding box xtl /= 1920 xbr /= 1920 ytl /= 1080	**if** slope <= start_12 and slope >= end_12: **return** 12 **elif** slope > end_11 and slope <= start_11: **return** 11 **elif** slope > end_1 and slope <= start_1: **return** 1 **elif** slope > end_10 and slope <= start_10:

ybr /= 1080

Coordinate of the origin

origin_x = 0.5

origin_y = 1.

Coordinate of the center of the bounding box

x = (xtl + xbr) * 0.5

y = (ytl + ybr) * 0.5

Slope between the origin and the center

slope = (x-origin_x) / (y-origin_y)

Slope of starting and ending point of each direction

12: start_12 = 0.2 end_12 = -1 * start_12

11: start_11 = 0.5 end_11 = start_12

1: start_1 = end_12 end_1 = -1 * start_11

10: start_10 = 2.5 end_10 = start_11

2: start_2 = end_1 end_2 = -1 * start_10

9: end_9 = start_10

3: start3 = end_2

```
        return 10

elif slope > end_2 and slope <= start_2:

        return 2

elif slope <= start3:

        return 3

else:

        return 9
```

Experiment

4.1 Dataset

Figure 5. Sample images of sidewalk dataset from AI Hub [13].

Dataset used in this study is obtained from AI Hub[13], organized by the Ministry of Science and ICT and supported by Korea Intelligence Information Society Promotion Agency. The dataset is specifically obtained for the use of artificial intelligence research to improve the quality of life of the blind. It meets the purpose of this study to provide a system guiding the blind.

The dataset consists of the images obtained in various real-world pedestrian environments that contain movable and static obstacles on the sidewalk with collision risks. It contains class labels and bounding boxes of 29 types of objects such as bicycles, movable signage, benches, and traffic lights. The dataset has 352,810 samples, divided into 10% of the test dataset and 90% of the training dataset.

4.2 Implementation Details

To train the proposed system, we perform a total of 200 Epochs using Adam (beta=0.9, beta2=0.99) [14]. We apply MultiStepLR with a decreasing factor of 0.1 from the 80th and 120th epoch.

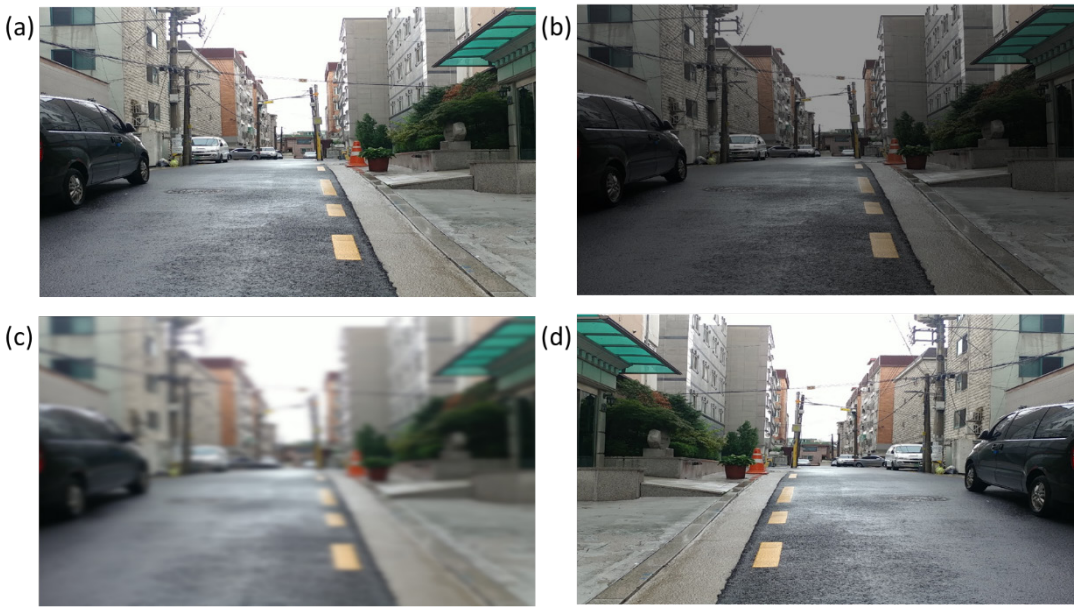

Figure 6. Visual comparisons between data augmentation techniques. (a) is the original image. (b) is the image with color gitter, (c) is the image with gaussian blur, and (d) is the image with a random flip.

For data augmentation, we use color jitter, gaussian blur, and random flip as shown in Fig. 6. Color Jitter calibrates the brightness of the samples. The samples in the dataset are photographed only in bright situations, so it differs from the real-world environments. To achieve rich probabilistic features, we apply color jitter data augmentation to mimic dark or cloudy environments. The images obtained in real-world environments also tend to be blurry since the camera attached to the wearable device oscillates as the user walks. Hence we randomly apply the gaussian blur in the training sample to enforce the trained model to perform well in this situation. Additionally, we apply Random Horizontal Flip which is often used to leverage out the accuracy and generalization power of the trained model.

4.3 Evaluation

The evaluation follows the same protocol explained in [5]

- The Intersection over Union (IoU) is the predicted bounding box and the actual bounding box intersection divided by the union of the two bounding boxes. If the IoU is greater than the threshold, it is considered true positive (TP), if it is less than the threshold, false positive (FP).
- The mean average precision (mAP) is the average area of the Precision-Recall curve for each class, which measures the performance of the object detection algorithm.

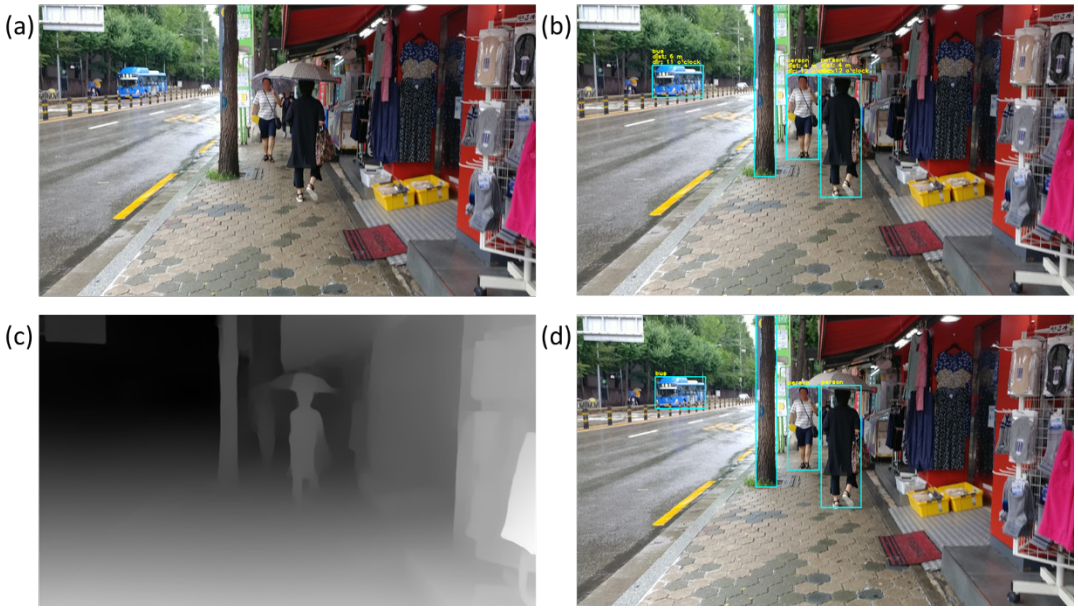

Figure 7. Visual comparison between results of networks of GuideDogNet.

(a) input image, (b) result of object detection network, (c) result of depth prediction network, and (d) final result of the proposed post-processing module.

Table 3. Ablation study result.

Model	mAP
baseline model	67.8
model with swine positional encoding	68.2

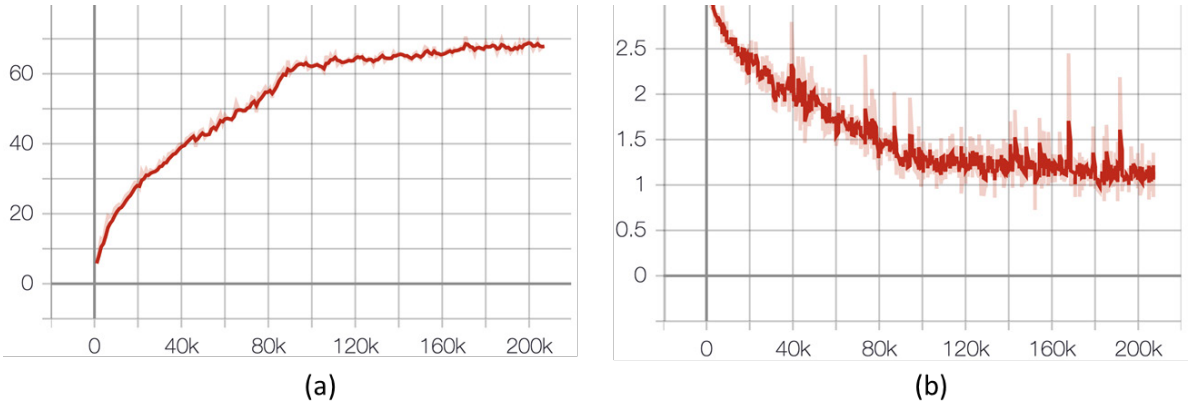

Figure 8. (a) is a training loss graph. (b) is a test accuracy graph.

Fig 7. shows the result images in each step in the proposed system; (a) is the input image, (b) shows labels and bounding boxes of obstacles detected from the object detection network, (c) is a depth map predicted from the depth prediction network, and (d) is the final result of the proposed system. To provide user-friendly outputs for the blind, we propose a rule-based post-processing module that outputs the label, direction, and distance of the obstacles by combining the results of the object detection network and the depth prediction network.

Additionally, we heuristically found that the positional encoding module affects the final accuracy. Through many experiments, we chose to apply the positional encoding method used in Swine [15]. It helped to yield better results in the proposed system as shown in Table 3. The baseline model achieves an mAP of 67.8. The model trained using the swine positional encoding method achieves a slightly better result of an mAP of 68.2. The performance was enhanced by 0.4.

Fig 8. shows the training loss and the accuracy of the test set. We found that the accuracy of the test set starts to saturate at around 180K training iteration thus we terminate the training process at the point.

Conclusion

In this paper, we proposed a novel deep learning-based blind guiding system, GuideDogNet, that outputs user-friendly information containing the label, direction, and distance of the obstacles. The proposed system consists of an object detection network, depth prediction network, and post-processing module. The proposed system achieved an mAP of 67.8. Additionally, we replaced the existing positional encoding module with the state-of-the-art method to produce better results. In conclusion, the final model achieved an mAP of 68.2. The proposed system can be easily applied to smartphones or wearable devices with cameras. Users can simply hold the camera to get guidance for the obstacles while walking. Yet, the system has a relatively high computational cost. In future research, we will develop a lighter model while preserving comparable accuracy.

Acknowledgments

We would like to thank Mr. Taeseon Yoon and Dr. Kyuyong Park at the Hankuk Academy of Foreign Studies for their guidance in this project.

References

[1] Wei, Yuanlong, Xiangxin Kou, and Min Cheol Lee. "A new vision and navigation research for a guide-dog robot system in urban system." 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. IEEE, 2014.
[2] Hapsari, Gita Indah, Giva Andriana Mutiara, and Dicky Tiara Kusumah. "Smart cane location guide for blind Using GPS." 2017 5th International Conference on Information and Communication Technology (ICoIC7). IEEE, 2017.
[3] Na, Jongwhoa. "The blind interactive guide system using RFID-based indoor positioning system." International Conference on Computers for Handicapped Persons. Springer, Berlin, Heidelberg, 2006.
[4] Harsur, Anushree, and M. Chitra. "Voice based navigation system for blind people using ultrasonic sensor." IJRITCC 3 (2017): 4117-4122.
[5] DETR (Detection Transformer): Zhu, Xizhou, et al. "Deformable DETR: Deformable Transformers for End-to-End Object Detection." arXiv preprint arXiv:2010.04159 (2020).
[6] DPT (Dense Prediction Transformer): Ranftl, René, Alexey Bochkovskiy, and Vladlen Koltun. "Vision transformers for dense prediction." arXiv preprint arXiv:2103.13413 (2021).
[7] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
[8] Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015.
[9] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[10] Liu, Wei, et al. "Ssd: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016.
[11] Garg, Ravi, et al. "Unsupervised cnn for single view depth estimation: Geometry to the rescue." European conference on computer vision. Springer, Cham, 2016.
[12] Liu, Fayao, Chunhua Shen, and Guosheng Lin. "Deep convolutional neural fields for depth estimation from a single image." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
[13] https://aihub.or.kr/
[14] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
[15] Liu, Ze, et al. "Swin transformer: Hierarchical vision transformer using shifted windows." arXiv preprint arXiv:2103.14030 (2021).