2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th Intl. Conference on Data Science and Systems

A Deep Learning Approach to Sensory Navigation Device for Blind Guidance

Josh Jia-Ching Ying, Chen-Yu Li, Guan-Wei Wu, Jian-Xing Li, Wei-Jheng Chen, and Don-Lin Yang

Department of Computer Science and Information Engineering, Feng Chia University, Taiwan, ROC

e-mail: jashying@gmail.com, st712123@gmail.com, canis850223@gmail.com, zz0978628963@gmail.com, davis10231@gmail.com, dlyang.tw@gmail.com

***Abstract*—Sensory navigation device is an important trend in the field of machine learning and data science. Nowadays, more and more sensory navigation devices are built for blind people. The core of such sensory navigation devices for blind people usually is implemented by an Image Recognition Method. To build an image recognition model, many tools and online machine learning platforms are proposed. However, these tools or platforms are not able to completely satisfy the requirements for sensory navigation device. To build a sensory navigation device with satisfying requirements for blind people, an ability of reducing the cost of model training and a capability of user-centric image recognition are the two main issues. Therefore, to address the above issues, we propose a novel approach, namely, DLSNF (Deep-Learning-based Sensory Navigation Framework). Our proposed DLSNF is built based on the YOLO architecture to deal with the reducing cost of model training and NVIDIA Jetson TX2 to take the user-centric image recognition into account. Based on our proposed DLSNF, the real-time image recognition can be trained well and conduct a sensory navigation to help blind people. At the same time, the train model is embedded in NVIDIA Jetson TX2 which is the fastest, most power-efficient embedded AI computing device. For the experiments, we evaluated our proposed DLSNF with a real-world dataset consisting of 4,570 images collected by part-time workers. The extensive experimental results show that our proposed DLSNF more effectively and efficiently beyond the existing baselines.**

Keywords—Sensory Navigation Device, Deep Learning, Residual Convolution Neural Network, Blind Guidance.

I. INTRODUCTION

In blind people's daily life, guidance tools can be realized in many ways, such as guide dog, guidance tiles, and Tactile sticks. However, the road conditions are always changed such that the guidance tools might make some irreparable mistakes. Take figure 1 as an example. The two pictures show the street view in different dates. If the left picture shows the usual road conditions, the blind who use the conventional guidance tools might consider the road conditions is very safe. When the blind walk through the street and found the real road conditions had become as shown in the right picture, he might be damaged by passing around the obstacles. This phenomena is called *time-variate dynamics*. As the result, hug number of blind people suffer terribly with these time-variate dynamics problem. To protect blind people against potential losses caused by time-variate dynamics, many real-time road condition recognition mechanisms have been proposed, whereas such mechanisms for blind guidance are still in its infancy stage. Currently, all of the real-time road condition recognition mechanisms for blind guidance more or less rely on the offline modeling manner. The system would collect street view via users' wearable sensors, and the recognition model would be trained on a server. However, the blind people still use the out-of-date model to detect obstacles before the model is updated. In other words, such offline modeling manner still has the time-variate dynamics problem.

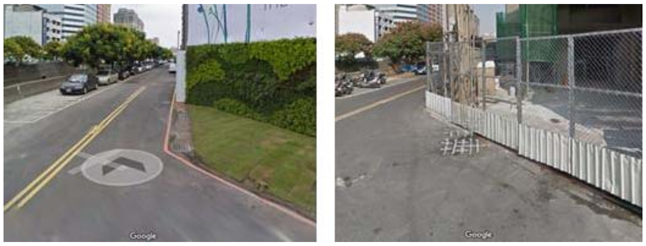


Figure 1. An Example of a Dialogue Robot.

Sensory Navigation Device [7] is an important trend in the field of guidance tool for blind people. The core of the sensory navigation device is a computer program that can lead blind and visually impaired people around obstacles through voice with artificial intelligence, data mining, or customized rules. Nowadays, there are three main groups of Sensory Navigation Devices, according to their working principle: radar, global positioning and stereovision. Meanwhile, the most widely known are the Sensory Navigation Devices based on the radar principle [18] [19] [20] . These devices emit laser or ultrasonic beams. When a beam strikes the object surface, it is reflected. Then, the distance between the user and the object can be calculated as the time difference between the emitted and received beam. A second type of Sensory Navigation Devices includes devices based on the Global Positioning System (GPS) [14] [15] [16] . These devices aim to guide the blind user through a previously selected route; also, it provides user location such as street number, street crossing, etc. Unfortunately, although the Sensory Navigation Devices based on the radar principle or the Global Positioning System have widely been applied to guidance tool for blind people, these two types of Sensory Navigation Devices are not able to deal with the time-variate dynamics problem.

With the development of the webcam, many researchers [5] [6] [7] proposed the application of stereovision to develop new

978-1-5386-6614-2/18/$31.00 ©2018 IEEE
DOI 10.1109/HPCC/SmartCity/DSS.2018.00201

techniques for representation of the surrounding environment. As the result, the Sensory Navigation Devices based on the stereovision principle which intend to represent the surrounding environment through acoustic signals has been proposed. Unlike the other two types of Sensory Navigation Devices, the Sensory Navigation Devices based on the stereovision principle can real-timely be updated. Accordingly, the time-variate dynamics problem could be partially solved. Although most sensory navigation devices based on the stereovision principle have been significantly improved with rapid growth of technology of artificial intelligence, the core of conventional sensory navigation devices is still constructed based on the offline modeling manner. Therefore, it is not realistic to directly adopt the conventional sensory navigation device as the guidance tool. Based on our observation, applying sensory navigation device with satisfying requirements for blind people has two main issues: 1) ability of reducing the cost of model training and 2) capability of user-centric image recognition.

As mentioned earlier, the reason why the offline training manner can not deal with the time-variate dynamics problem is the blind would be damaged because the model is out-of-date. The ability of reducing the cost of model training plays crucial role for speeding up the newer rate. The capability of user-centric image recognition can improve precision of sensory navigation device such that the blind can prevent some emergency accidents. To build a sensory navigation device with satisfying above-mentioned requirements, we propose a novel approach, DLSNF (Deep-Learning-based Sensory Navigation Framework). Meanwhile, to address the above-mentioned issues, DLSNF was built based on the YOLO [18] [19] to deal with the ability of reducing the cost of model training issue and NVIDIA Jetson TX2[1] to take the user-centric image recognition issue into account. Based on our proposed DLSNF, the user-centric image recognition model can be trained well and deploy on a sensory navigation device for blind guidance. At the same time, combined with real-world images crawled through our device, it can holistically enhance the users' usage experience.

The contributions of our research are four-fold:

- We propose the Deep-Learning-based Sensory Navigation Framework (DLSNF), a novel approach for blind guidance tool. The problems and ideas in *DLSNF* have not been explored previously in the research community.
- We develop the Residual-CNN based on YOLO architecture to deal with the ability of reducing the cost of model training issue in *DLCF*.
- We deploy the trained model on NVIDIA Jetson TX2 to take the user-centric image recognition issue into account.
- We evaluated our sensory navigation device with a real-world dataset collected by staffs. The dataset consists of 4,570 images. The extensive experimental results show that our sensory navigation device more effectively and efficiently beyond the existing baselines.

The remainder of this paper is organized as follows: Section 2 presents a review of related research. Section 3 details our proposed Deep-Learning-based Sensory Navigation Framework (DLSNF). Evaluations of the proposed system are presented in Section 4, and we present a case study in Section 5. Finally, the work is concluded in Section 6.

II. Related Works

In this section, we briefly introduce most popular methods which can be utilized for constructing a blind guidance tool. According to the type of core idea behind the methods, we categorize these methods into two categories, Conventional Image Recognition and Deep-Learning-based Image Recognition.

A. Conventional Image Recognition

OpenCV. Decade ago, OpenCV [3] is originally introduced by Intel for image and video analysis, originally introduced more than. In [26] , Xie *et al.* introduce a method of image edge detection based on OpenCV with rich computer vision and image processing algorithms and functions. Meanwhile, the detection model determines the exact number of the copper core in the tiny wire. In [8] , Emami *et al.* utilize OpenCV and .NET framework to build an application that would allow user access to a particular machine based on an in-depth analysis of a person's facial features.

SIFT. Scale Invariant Feature Transform (SIFT) is a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene [17] . Lowe also describes an approach to using these features for object recognition. The recognition proceeds by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform to identify clusters belonging to a single object, and finally performing verification through least-squares solution for consistent pose parameters.

SURF. In [1] , Bay *et al.* present a novel scale- and rotation-invariant interest point detector and descriptor, coined SURF (Speeded Up Robust Features). It approximates or even outperforms previously proposed schemes with respect to repeatability, distinctiveness, and robustness, yet can be computed and compared much faster.

BRIEF. In [2] , Calonder *et al.* propose to use binary strings as an efficient feature point descriptor, which they call BRIEF. They show that it is highly discriminative even when using relatively few bits and can be computed using simple intensity difference tests. Furthermore, the descriptor similarity can be evaluated using the Hamming distance, which is very efficient to compute, instead of the L2 norm as is usually done.

HOG. In [4] , Dalal *et al.* study the question of feature sets for robust visual object recognition, adopting linear SVM based human detection as a test case. The result shows experimentally that grids of Histograms of Oriented Gradient (HOG) descriptors significantly outperform existing feature sets for human detection.

[1] https://developer.nvidia.com/embedded/buy/jetson-tx2

B. Deep-Learning-based Image Recognition

CNN. Convolutional Neural Networks (CNN). In [13] , Krizhevsky *et al.* trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, they achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. In [11] , Goodfellow *et al.* propose an unified approach that use a deep convolutional neural network to recognize arbitrary multi-character text in unconstrained natural photographs. They evaluate this approach on the publicly available SVHN dataset and achieve over 96% accuracy in recognizing complete street numbers.

R-CNN. In [10] , Girshick *et al.* propose an approach combines two key insights: (1) one can apply high-capacity convolutional neural networks (CNNs) to bottom-up region proposals in order to localize and segment objects and (2) when labeled training data is scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, yields a significant performance boost. In [9] , Girshick proposes a fast R-CNN which can speed up the training process of deep convolutional networks. Compared to R-CNN, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy

VGG nets. In [24] , Simonyan *et al.* investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Their main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3 × 3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers.

GoogLeNet. In [25] , Szegedy *et al.* propose a deep convolutional neural network architecture codenamed Inception that achieves the new state of the art for classification and detection. The main hallmark of this architecture is the improved utilization of the computing resources inside the network.

Residual-Net. In [12] , He *et al.* present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. They explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions.

YOLO. In [21] , Redmon *et al.* developed a fast single-shot detection method named you only look once (YOLO). YOLO is to predict multiclass bounding box candidates directly from the grids in the full input images. The combination of the class probabilities and bounding box confidence provides the resulting detection. The input images are divided into 7 × 7 grids. Thus, each grid predicts classification probabilities for class and candidate bounding boxes with the confidence score. Each bounding box contains five position indicators, including the box coordinates (x, y, w, h) and the position confidence. In [22] , Redmon *et al.* proposed YOLOv2, a faster and more accurate detector has been proposed. Redmon *et al.* pool a variety of ideas from past work with our own novel concepts to improve YOLO's performance. In [23] , Redmon *et al.* continue to improve the YOLO's performance so that YOLOv3 has been proposed in 2018.

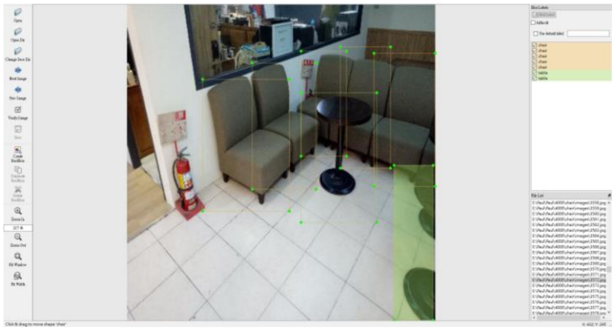


Figure 2. An Illustration of Object Annotation of an Image.

III. Our Proposed Method

To build the sensory navigation device for blind guidance, we first collect images through a webcam and annotate the objects shown in the images. Then we utilize the annotated images to train an image recognition model based on the YOLOv3 architecture. Finally, we deploy the trained model on the NVIDIA Jetson TX2.

A. Data Preprocess

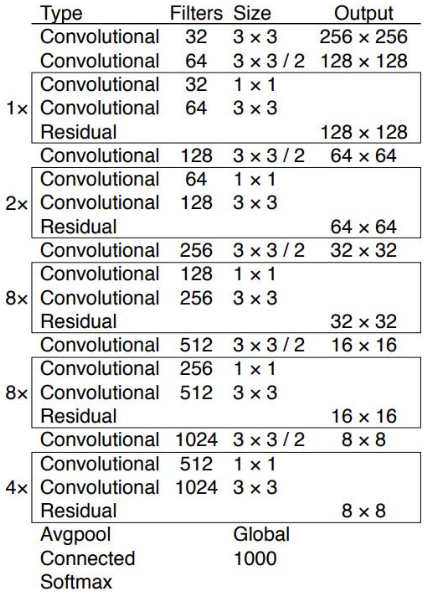
As mentioned earlier, the idea of our framework is listed as follows: 1) YOLO architecture can be used to deal with the ability of reducing the cost of model training, and 2) NVIDIA Jetson TX2 can take the capability of user-centric image recognition into account. Therefore, transforming annotated images into the formulation which is compatible for the training data of model building plays crucial role for building our sensory navigation device.

To do so, we first hire several staffs to collect images through a webcam. All objects shown in the collected images would be annotated by the staffs. Here, we adopt the an open source image labeling tool, LabelImg [27] , to annotate the objects which is critical for blind guidance. Figure 2 shows an example of LabelImg. The image shows several chairs and one table. Thus, the staff annotate these object and LabelImg would output a text file which records the boundary of these object and their labels.

B. Sensory Navigation Device Building

As mentioned earlier, we have already collect and annotate images. In this subsection, we then detail how we build a model for building an object detection model which can help blind to pass by skirted the obstacles. In order to producing the object detection model, we utilize the YOLOv3 architecture, which is one of the popular types of Residual-CNNs. Figure 3 shows the structure of a neuron of the YOLOv3. Meanwhile, it has 53 convolutional layers which uses successive 3 × 3 and 1 × 1 convolutional layers but now has some shortcut connections as well and is significantly larger.

Accordingly, we can realize that the model is too large to be trained within a reasonable time. Fortunately, the "Residual Neuron Network" inherently has some shortcut connections, i.e., Residual layer in Figure 3. Such shortcut connections can



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 3. An Illustration of YOLOv3 Architecture.

make the model much simpler if the characteristics of the training data is simple enough. For example, if our detection job is simplified as detecting "chair" and "desk", it only requires thousands images to train the model, and the trained model may only need lower 16 convolution layers. Thus, all residual layer would be activated such that all repeat block would not be performed. The remaining problem is how to use this characteristics to speed up training process. The ideal way is to divide the training dataset into small mini-batch, and to update the parameters based on accumulation of the loss produced from a mini-batch. The example of the mini-batch is shown in Figure 4. As the result, we utilize the mini-batch manner to build the object detection model within reasonable learning time

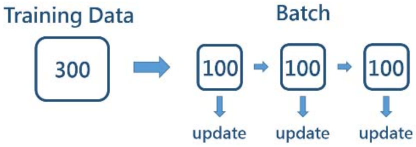


Figure 4. Illustration of Model Training Based on Mini-Batch Manner

IV. Experiments

In this section, we present the results from a series of experiments and evaluate the performance of our model. All the experiments are implemented in Python 3.5 on a NVIDIA Tesla K40 GPU machine with 64 GB of memory running Ubuntu Linux 16.04 LTS. We first describe the preparation of the datasets then, present and discuss our experimental results.

A. *Dataset Description & Performance Metrics*

We collected table and chair images, as shown in Figure 5 The data set used in the experiments consists of the chair and table images captured from various indoor sciences such as classrooms, library, and conference room, in which 2,084 chair images, 301 table images, 2,185 images contain both chair and table. Finally, the dataset was collected 4,570 images. All images were resized to 448 x 448. To build the training set, we manually draw the bounding boxes and assign the labels of 4,570 images.

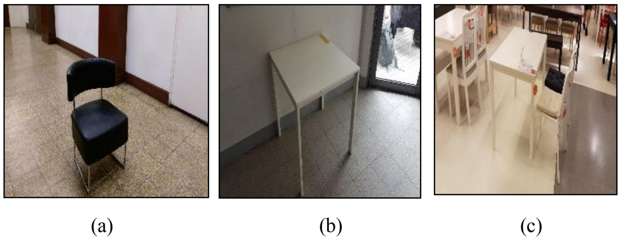

(a) (b) (c)

Figure 5. Three type image samples in our database, (a) is a chair sample, (b is a table sample and (c) is a image contains table and chair sample.

YOLO's loss function must simultaneously solve the object detection and object classification tasks. This function simultaneously penalizes incorrect object detections as well as considers what the best possible classification would be. We implement the following loss function as Equations (1), where $\mathbb{1}_i^{obj}$ denotes if object appears in cell i and $\mathbb{1}_{ij}^{obj}$ denotes that the jth bounding box predictor in cell i is "responsible" for that prediction.

$$\begin{aligned}
&\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2
\end{aligned} \tag{1}$$

Note that the loss function only penalizes classification error if an object is present in that grid cell. It also only penalizes bounding box coordinate error if that predictor is "responsible" for the ground truth box.

B. *Experimental Results*

In this section, we trained our model in set of 4,000 images, the testing images are collected from a different scenes but are tested under the same computation environment as training. The proposed method displays good results in localizing the chairs and tables. Figure 6 shows several visualized detection

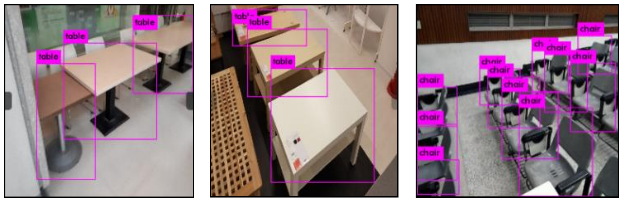
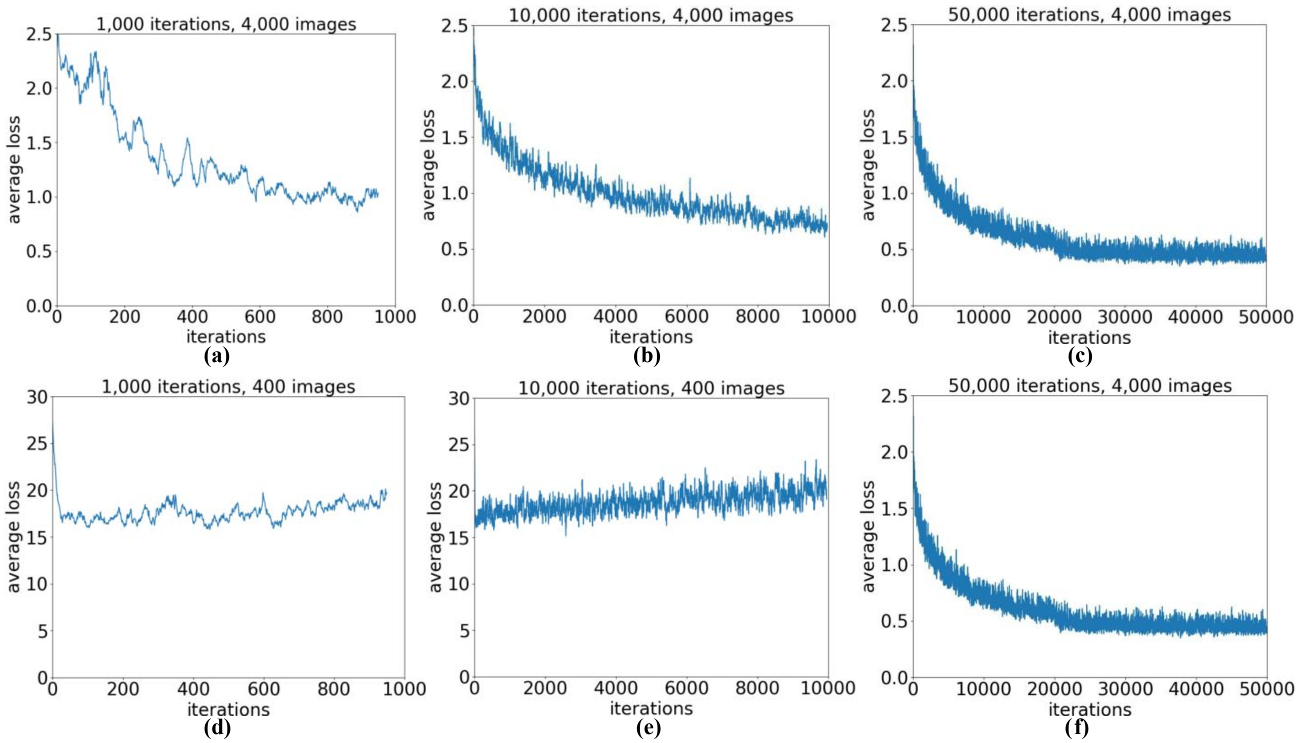


Figure 6. Most of the chairs and tables can be recognized and dectected



(a) (b) (c)

(d) (e) (f)

Figure 7. Average Loss under Various Hyperparameter Settings

examples and results.We also compared different iterations in terms of loss score. We do the normal training and fine-tuning both using batchsize = 64 to see the result of image preprocessing.

Figure 7(a)-(c) shows the comparison of different iterations with 4,000 images including tables and chairs in terms of loss score. It can be seen from Figure 7 (a) that 1,000 iterations is unstable and loss is high in end of loss score. However, according to Figure 9 (c), The loss decreases stably in 50,000 iterations. Therefore, we can observe that training through more multiple iterations to achieve the good result.

Figure 7(d)-(f) shows the training data for 400 chairs in terms of loss score. It can be seen that loss score does not drop and fluctuation is very unstable. The reason might be that too little training data, it means training data are not rich. Therefore, we can say that a small amount of data sets results in poor performance.

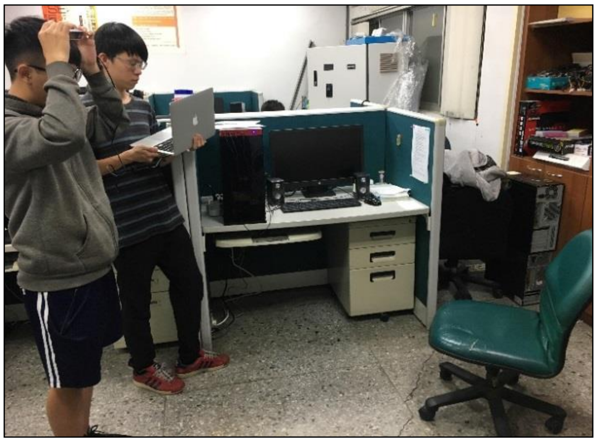


Figure 8. Test a chair about one meter away.

V. Case Study

In this section, we use our model for object detection and distance experiment. To prove the adaptability of the model, we experimented with chair detection and distance measurement indoors. Figure 8 shows we simulate the situation of blind people using camera indoors. We stand at a distance of one meter in front of the chair, and using the camera to detect the chair and its distance.

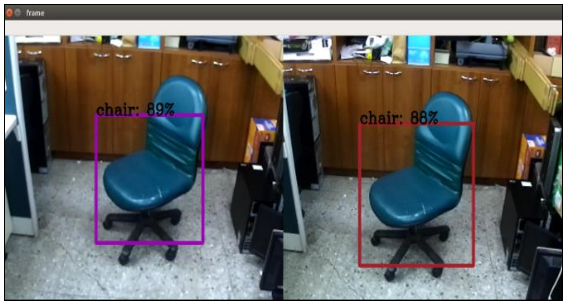
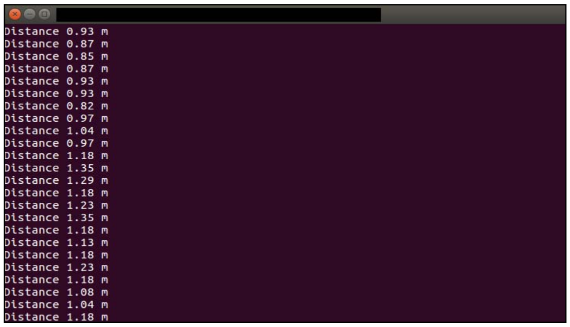


Figure 9. Detection of the chair by our model. Left screen output is left camera shot, and right screen output is right camera shot.

The proposed method displays good results in localizing the chairs. Figure 9 shows detection examples and results. It can be seen that both of the screen have good performance, and the proposed YOLO offers a speedup. It sometimes generates bounding boxes of different sizes because of different angles from left and right camera shot, but a slight gap of the false prediction of the bounding boxes is allowed to some extent.


[redacted]

```
Distance 0.93 m
Distance 0.87 m
Distance 0.85 m
Distance 0.87 m
Distance 0.93 m
Distance 0.93 m
Distance 0.82 m
Distance 0.97 m
Distance 1.04 m
Distance 0.97 m
Distance 1.18 m
Distance 1.35 m
Distance 1.29 m
Distance 1.18 m
Distance 1.23 m
Distance 1.35 m
Distance 1.18 m
Distance 1.13 m
Distance 1.18 m
Distance 1.23 m
Distance 1.18 m
Distance 1.08 m
Distance 1.04 m
Distance 1.18 m
```

Figure 10. Output of the chair's distance

Figure 10 shows the output of the chair's distance on command, it determines the distance based on the different angles of the two screens. According to the bounding box float will output the distance, it can be seen from command that the distance is constantly output. There may be some slight distance errors, but a slight gap of the distance will not affect users. We can see that the output of the chair's distance is about 1 meter, and the output is the same as the actual distance.

VI. Conclusions

In this paper, we propose a novel Deep-learning-based Sensory Navigation Framework (DLCF) to build a Sensory Navigation Device for blind guidance. We also tackled the problem of object detection, which is a crucial prerequisite for blind guidance tool. The core task of model learning is conveniently transformed to the problem of object detection model learning. We develop the Residual-CNN architecture to detect object shown in a snapshot catch by a webcam. Through a series of experiments using a dataset crawled by staffs, we have validated the Residual-CNN for building an object detection model and shown that it has excellent performance under various conditions. In future work, we plan to design more sophisticated methods and compare it with state-of-the-art methods.

References

This research was partially supported by Ministry of Science and Technology, Taiwan, R.O.C. under grant no. MOST 106-2218-E-126-001 and MOST 106-2221-E-035-094.

References

[1] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," Computer Vision and Image Understanding (CVIU), 110(3):346-359, 2008

[2] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. "BRIEF: Binary Robust Independent Elementary Features," In Proceedings of the European Conference on Computer Vision (ECCV), 2010

[3] I. Culjak, "A brief introduction to OpenCV," Proceedings of the 35th International MIPRO Convention, IEEE (2013), pp. 2142-2147, 2012

[4] N. Dalal, and B. Triggs, "Histograms of Oriented Gradients for Human Detection," Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference, 2005

[5] L. Dunai, G. P. Fajarnes, V. S. Praderas, and B. D. Garcia, "Electronic Travel Aid systems for visually impaired people." in Proceedings of DRT4ALL 2011 Conference, IV Congreso Internacional de Diseño, Redes de Investigación y Tecnología para Todos, Madrid, Spain, 2011.

[6] L. Dunai, G. P. Fajarnes, V. S. Praderas, B. D. Garcia, and I. Lengua, "RealTime assistance prototype – a new navigation aid for blind people." In Proceedings of IEEE Industrial Electronics Society Conference (IECON 2010), Phoenix, Arizona. 1173–1178, 2010.

[7] L. Dunai, G. P. Fajarnes, V. S. Praderas, B. D. Garcia, and I. Lengua, "EYE2021-Acoustical cognitive system for navigation." AEGIS 2nd International Conference, Brussels, 2011.

[8] S. Emami, and V. P. Suciu, "Facial Recognition using OpenCV," Journal of Mobile,Embedded and Distributed Systems, 4(1), 38-43, 2012

[9] R. Girshick, "Fast r-cnn," IEEE international conference on computer vision, 2015

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," IEEE conference on computer vision and pattern recognition, 2014.

[11] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks," arXiv:1312.6082, 2014

[12] K. He, X. Zhang, S. Ren, and Ji Sun, "Deep residual learning for image recognition," arXiv:1512.03385, 2015.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," In: Advances in neural information processing systems, pp. 1097-1105, 2012.

[14] R. Kuc, "Binaural Sonar Electronic Travel Aid Provides Vibrotactile Cues for Landmark, Reflector Motion and Surface Texture Classification." IEEE Transactions on Biomedical Engineering, 49, 1173–1180, 2002.

[15] J. M. Loomis, R. G. Golledge, and R. L. Klatzky, "GPS-Based Navigation Systems for the Visually Impaired." Fundamentals of wearable computers and augmented reality, W. Barfield and T. Caudell, Eds., 429–446, Mahwah, NJ: Lawrence Erlbaum Associates, 2001.

[16] J. Loomis and R. Golledge, "Personal Guidance System using GPS, GIS, and VR technologies." In Proceedings, CSUN Conference on Virtual Reality and Person with Disabilities, San Francisco, 2003.

[17] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," IJCV, 60 (2), pp. 91-110, 2004

[18] R. W. Mann, "Mobility aids for the blind – An argument for a computer-based, man-device environment, interactive, simulation system." In Proceedings of Conference on Evaluation of Mobility Aids for the Blind, Washington, DC: Com. On Interplay of Engineering With Biology and Medicine, National Academy of Engineering, 101–116, 1970.

[19] D. L. Morrissette, G. L. Goddrich, and J. J. Henesey, "A follow-up-study of the Mowat sensors applications, frequency of use and maintenance reliability." Journal of Visual Impairment and Blindness, 75, 244–247, 1981.

[20] L. Russell, "Travel Path Sounder." In Proceedings of Rotterdam Mobility Research Conference, New York: American Foundation for the Blind, 1965.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," arXiv:1506.02640, 2015.

[22] J. Redmon, and A. Farhadi, "YOLO9000: Better, Faster, Stronger," Computer Vision and Pattern Recognition (CVPR), 2017.

[23] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv:1804.02767v1, 2018.

[24] K. Simonyan, and A. Zisserman. "Very deep convolutional networks for large-scale image recognition." In ICLR, 2015.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions." In CVPR, 2015.

[26] G. Xie, and W. Lu, "Image Edge Detection Based on OpenCV." International Journal of Electronics and Electrical Engineering 1 (2): 104-6, 2013

[27] https://github.com/tzutalin/labelImg