ICIC Express Letters
Part B: Applications
Volume **10**, Number **3**, March **2019**

ICIC International ©2019 ISSN 2185-2766
pp. **219–226**

SMARTPHONE APPLICATION PROGRAM OF OBSTACLE DETECTION FOR VISUALLY IMPAIRED PEOPLE

Manabu Shimakawa[1], Issei Taguchi[2], Chiharu Okuma[1] and Kimiyasu Kiyota[1]

[1]Department of Human-Oriented Information Systems Engineering
[2]Electronics and Information Systems Engineering Course
National Institute of Technology, Kumamoto College
2659-2 Suya, Koshi, Kumamoto 861-1102, Japan
{ shimakawa; chiharu; kkiyota }@kumamoto-nct.ac.jp; ae16taguchi@g.kumamoto-nct.ac.jp

Received August 2018; accepted November 2018

Abstract. *This study deals with obstacle detection to help visually impaired people walk. Our previous studies had taken an approach of using some image processing techniques to detect the steps of staircase on depth images by RGB-D camera. However, it was difficult to carry the RGB-D camera and not practical. This study applied CNN (Convolutional Neural Network) which is a kind of deep learning techniques to obstacle detection and developed an application program working on smartphone. It can classify the taken images into some obstacles or surrounding situations. This paper shows experimental results and discusses about the effectiveness of them.*

Keywords: Obstacle detection, Visually impaired, Machine learning, Deep learning, CNN, TensorFlow

1. **Introduction.** According to the WHO (World Health Organization) fact sheets [1], an estimated 253 million people live with vision impairment: 36 million are blind and 217 million have moderate to severe vision impairment. The number of people is continuing to increase. The visually impaired has a high need to go out; however, there are so many situations outdoors where they feel danger. In fact, a survey report [2] about accidents of visually impaired pedestrian said that 42% of respondents experienced accidents during their walking.

Researches and developments of support tools for the walking of visually impaired people have been doing for a long time. The smart electronic white cane [3] is one of walking support devices commercialized. It has ultra-sonic sensors, alerts the danger with vibration when it detects obstacles above and/or front position. There is also research to detect staircase and pedestrian traffic signals using images of RGB-D camera [4]. Our previous studies [5,6] had taken an approach of using some image processing techniques to detect the steps of staircase on depth images by RGB-D camera. However, the RGB-D camera is large and heavy, and it is difficult to carry, so we aimed to implement it on a light-weight smartphone with RGB camera. We applied machine learning techniques to detecting stairs instead of ordinal image processing techniques to using RGB images because it can implement into smartphone [7,8].

This study used CNN (Convolutional Neural Network) which is a kind of deep learning techniques to obstacle detection and developed an application program working on smartphone [9]. It can classify the taken images into some obstacles or surrounding situations and notice it to user to help his/her safety walking. This paper shows experimental results and discusses about the effectiveness of them.

DOI: 10.24507/icicelb.10.03.219

2. **System Overview.** Figure 1 shows the diagram of this system flow. A visually impaired wears a smartphone equipped with a camera to take RGB images of forward. This system works as an application program on the smartphone. The taken RGB images are used as input data for CNN, and this system classifies some objects on the images. When the object is recognized as an obstacle, this system notifies it to the user by sound and vibration.

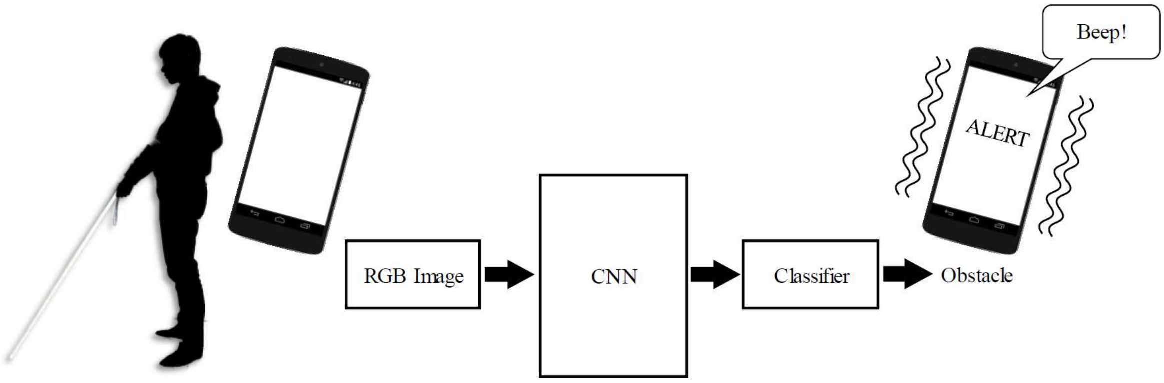


FIGURE 1. System flow diagram

3. **Obstacle Detection by CNN.**

3.1. **CNN (Convolutional Neural Network).** CNN is a feed-forward artificial neural network consisting of a repetition of convolution layers and pooling layers. In general, it is said to have superior performance in visual recognition problem, it is well applied to extracting features of images and to classifying them. There are learning phase and recognition phase in the object detection using CNN. Figure 2 shows a typical CNN architecture.

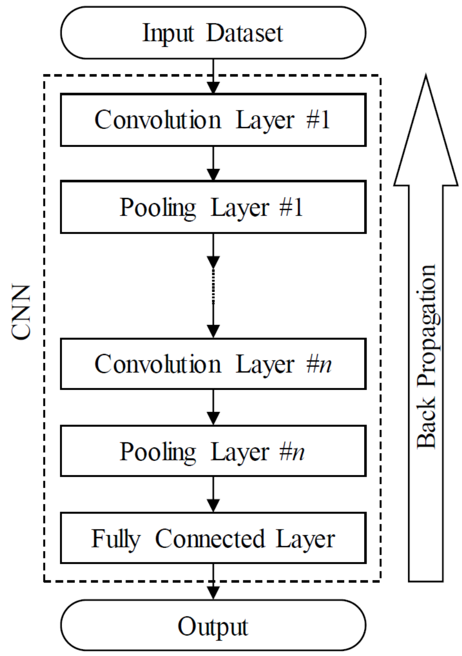


FIGURE 2. Typical CNN architecture

3.1.1. *Convolution layer.* This layer performs a convolution operation and extracts the features of the image. The convolution operation is similar to the filter operation often used for image processing such as Sobel filter and Gaussian filter. In the convolution operation, a product-sum operation is performed while sliding the filter region to create a feature map. Filter size and parameters are target to adjustment during the learning phase.

3.1.2. *Pooling layer.* Pooling performs calculation for obtaining the maximum value for each divided region. An example of pooling is shown in Figure 3. When pooling is carried out, even if the image is moving or rotating by several pixels, it does not substantially affect the output result. Therefore, correct classification is possible regardless of the position of the obstacle in the image.

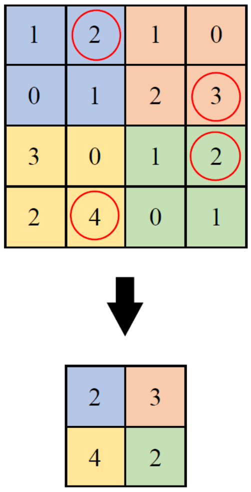


FIGURE 3. Pooling layer operation

3.1.3. *Fully connected layer.* This layer classifies the features that have been extracted through the convolution layer and the pooling layer. It is basically the same as the multilayer perceptron as shown in Figure 4. This layer regards data output x_i from the last pooling layer as input nodes, calculates a weighted sum with connection factors w_{ij} as weight, and adds a bias b_j by the following equation:

$$y_j = f\left(\sum_{i=1}^{m} w_{ij}x_i + b_j\right) \tag{1}$$

where y_i is an output node, and $f(\cdot)$ is an activation function. Generally, the probability of each output node is obtained by Softmax function, and the input data of CNN is classified into the candidate indicated by the output node with the highest probability.

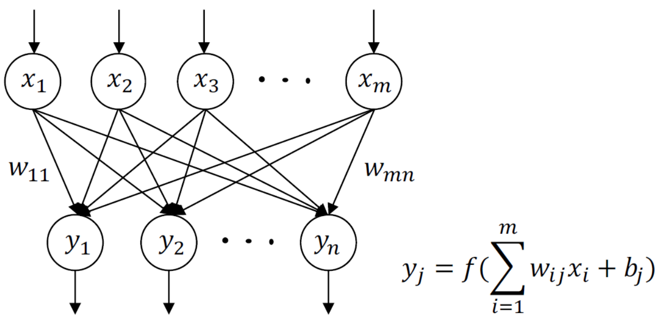


FIGURE 4. Basic structure of fully connected layer

3.2. **Learning phase.** The process to build a CNN is called learning phase. The learning phase needs a large amount of computation, so this phase is required by high performance computer. The dataset of obstacle images is given into the CNN network. The input images are classified through three processes of convolution layer, pooling layer and fully connected layer. And the error from the original output value is calculated and the weight and bias are corrected from the error. This learning method is called the back-propagation method. The above four steps are iterated, and the weight of the classification network and the bias optimum.

3.3. **Recognition phase.** RGB image taken by camera is given to the CNN that is constructed in the learning phase. The CNN classifies it into some obstacles or situations as an output at the fully connected layer after repeating the convolution layers and the pooling layers. Since the recognition phase is a lightweight process, it can be performed on a smartphone.

4. Obstacle Detection System.

4.1. **Building a CNN with TensorFlow.** This system uses TensorFlow which is an open-source framework of machine learning developed by Google. TensorFlow has the advantage of easy building CNN and is able to use Python. In building a CNN, it is necessary to prepare image data including obstacles for the classification. It needs huge amount of image data to improve classification accuracy; however, it is difficult to gather them. Therefore, this study uses Inception-v3 [10] which is already learned CNN model. Inception-v3 was also developed by Google and was learned to classify images into 1,000 classes for the image identification task of ILSVRC, and can perform very accurate image identification.

In this study, 637 images including obstacles such as *Stairs*, and *Bicycle*, and 393 images of the scenes such as *Rail Tracks*, *Sidewalk*, and *Pedestrian Crossing* were prepared for training. The Inception-v3 was re-learned using these images and increased the types that can be identified. Figure 5 shows a layer structure of the re-learned Inception-v3. It consists of very complex structure with many convolution layers and pooling layers.

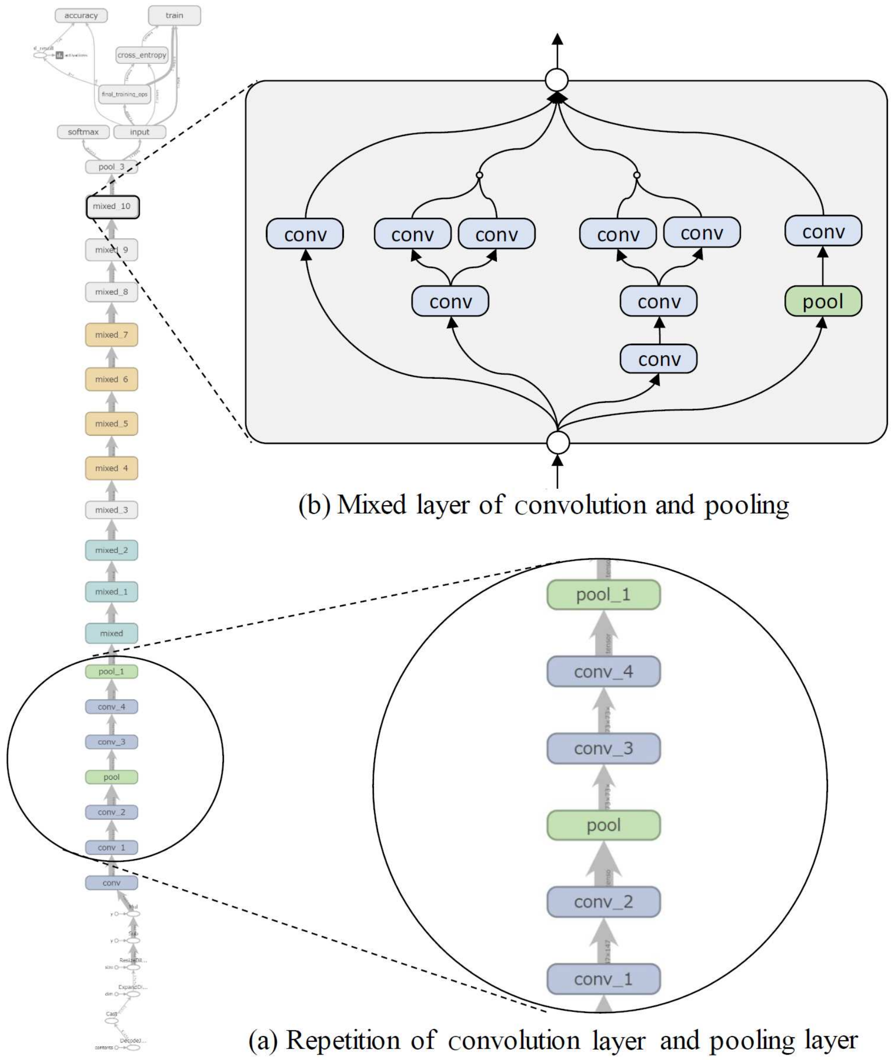
4.2. **Application program on a smartphone.** We developed an application program that works on Android smartphone. This program executes the recognition phase using the CNN created in the learning phase. The data of the CNN is expressed in a protocol buffer format file.

Then, CNN computes according to the layer structure and derives the probability for each output candidate by the Softmax function of the last layer. The candidate having the highest probability value will be selected. When the probability is 0.8 or more and it is in obstacle class, the program notifies by sound and vibration. When the probability is less than 0.8, the program displays it, but no notification.

5. Experimental Results.

5.1. **Experiments of classification and obstacle detection.** This section shows about experiments of classification and obstacle detection. The experiments were conducted during the day in city area. The experimenter held a smartphone at about 140 cm height from the ground, and did not fix the angle of it.

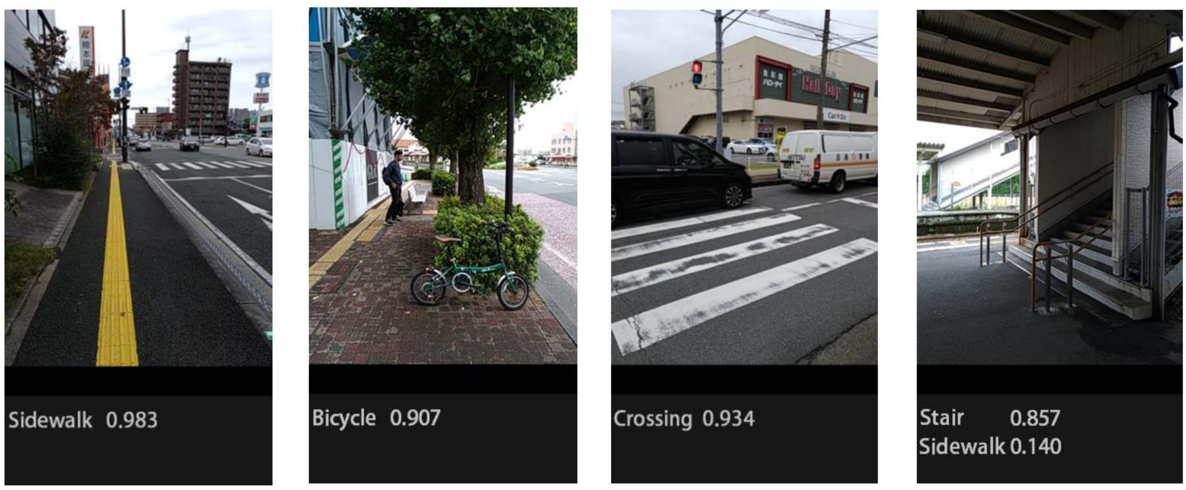
Figure 6 shows screenshots of situations in which the classification succeeded. All of these situations, (a) sidewalk, (b) bicycle, (c) pedestrian crossing and (d) stairs, were classified successfully with high probability values. On the other hand, Figure 7 are showing examples of fail classification. Figure 7(a) mistook a fence as a staircase, and Figure 7(b) shows a misidentification of a pedestrian crossing. Table 1 summarized the accuracy of classification. Although the number of samples is small, good results are shown.



(b) Mixed layer of convolution and pooling

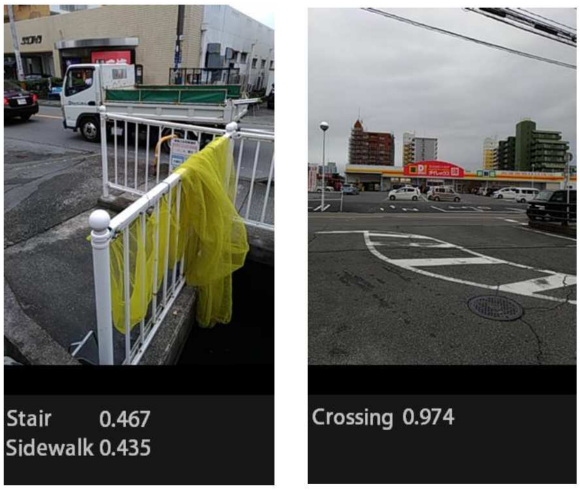
(a) Repetition of convolution layer and pooling layer

FIGURE 5. Layer structure of re-learned inception-v3



(a) Sidewalk (b) Bicycle (c) Pedestrian crossing (d) Stairs

FIGURE 6. Screenshots of classified situations

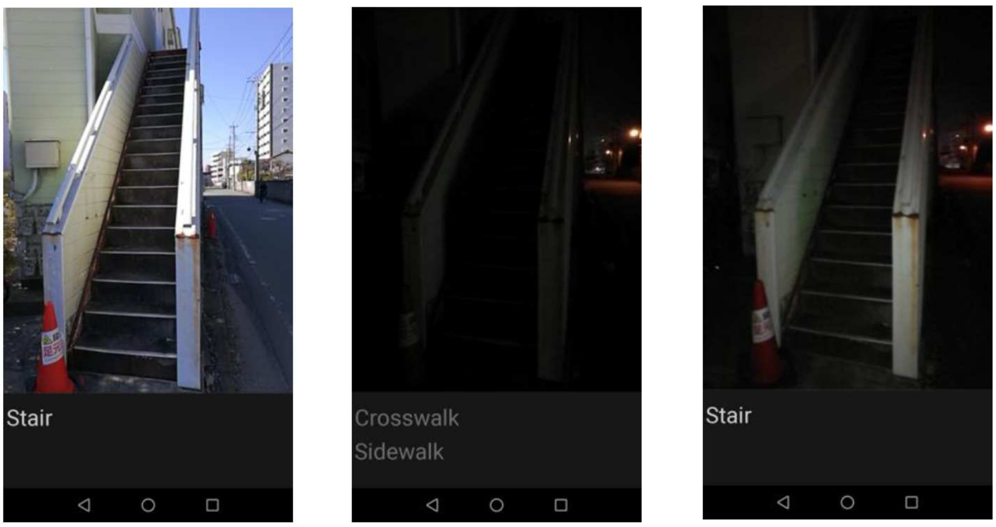


(a) Stairs (b) Pedestrian crossing

FIGURE 7. Examples of fail classification

TABLE 1. Accuracy of classification

	Scenes	Correct	Ratio
Sidewalk	39	36	92.3%
Pedestrian crossing	11	10	90.9%
Bicycle	14	14	100.0%
Stairs	37	36	97.3%
Total	101	96	95.0%



(a) Daytime (b) Nighttime without LED (c) Nighttime with LED

FIGURE 8. Screenshots of classified situations on daytime and nighttime

5.2. **Experiments on daytime and nighttime.** Classification accuracy deteriorates when the illuminance decreases. In order to investigate its effect, we conducted experiments during daytime and nighttime, and compared them. Figure 8 shows experimental results on the stair detection situation. Figure 8(a) shows a result of the experiment during the daytime, and the stairs were correctly detected. On the other hand, as shown in Figure 8(b), in the case of nighttime, it was dark and could not be detected. However, by turning on the LED light of the smartphone, the illuminance increased, and it was able to detect the stairs.

Table 2 shows comparison results of accuracy between daytime and nighttime. Even without using LED light at nighttime, the detection rate of 68.8% is shown, but these are all cases when there was a streetlight nearby.

Table 2. Comparison of accuracy between daytime and nighttime

		Scenes	Correct	Ratio
Daytime		16	16	100.0%
Nighttime	without LED	16	11	68.8%
	with LED	16	15	93.8%

5.3. **Power consumption.** These experiments measured power consumption during the application works. The smartphone was set airplane mode, display brightness 20%, the power saving setting "smart". Under such conditions, when using this application program continuously for about 90 minutes, the average current was 680 mA, and the battery level fell from 100% full to 64%.

6. **Conclusions.** This paper has proposed an obstacle detection system that can work on smartphone using RGB image. This system uses CNN which is one of deep learning techniques. The construction of CNN in the learning phase requires a huge amount of computation, but once CNN can be constructed, the amount of computation in the recognition phase that uses it is not so much needed. Moreover, in this study, we built a CNN by re-learning based on the Inception-v3 which is already learned CNN model. It makes possible to reduce the computational amount while keeping the accuracy of the CNN model.

The experimental results showed that it is possible to detect obstacles with a high accuracy of about 95%. In addition, it became clear that the detection rate can be greatly improved by turning on the LED light embedded in the smartphone even at nighttime. Although it can detect obstacles with high accuracy, it is limited to stairs, bicycles, rail tracks, sidewalks and pedestrian crossings that can be detected for the moment. Therefore, it is necessary to strengthen the learning so as to cope with more types of obstacles. Moreover, this system cannot inform the distance to the detected obstacle. It is desirable to be able to provide distance information that can distinguish whether close or far away. By improving the system to be able to handle such distance information, it becomes a more practical system.

From the result of the battery consumption experiments, it could be calculated that continuous operation can be performed for about 250 minutes (about 4 hours) with 100% remaining battery power. In order to extend the continuous usable time, it will be also necessary to consider constructing a low-load system.

REFERENCES

[1] *World Health Organization: Blindness and Visual Impairment*, Fact Sheets, 11 Oct. 2017, http://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment (accessed 05/06/2018).

[2] N. Abe and N. Hashimoto, Walking accident national survey to maintain visually handicapped persons walking environment, *The Bulletin of Hachinohe Institute of Technology*, no.24, pp.81-92, 2005.

[3] API Co. Ltd., *Smart Electronics White Cane System*, http://www.api-kk.com/denshi-hakujo/ (accessed 05/06/2018).

[4] S. Wang, H. Pan, C. Zhang et al., RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs, *Journal of Visual Communication Image Representation*, no.25, pp.263-272, 2014.

[5] S. Murakami, M. Shimakawa, K. Kiyota and T. Kato, Study on stairs detection using RGB-depth images, *Joint Conference of the 7th SCIS and the 15th ISIS (SCIS&ISIS2014)*, TP5-3-3-(1), 2014.

[6] M. Shimakawa, S. Murakami and K. Kiyota, Study on stairs detection using RGB-depth images, *The 31st Fuzzy System Symposium*, FB2-1, pp.699-702, 2015.

[7] R. Akutagawa, M. Shimakawa and K. Kiyota, A study on staircase detection by machine learning using OpenCV, *The 32nd Fuzzy System Symposium*, WF1-4, pp.83-86, 2016.

[8] M. Shimakawa, R. Akutagawa, K. Kiyota and M. Nakano, A study on staircase detection for visually impaired person by machine learning using RGB-D images, *Proc. of the 5th IIAE International Conference on Intelligent Systems and Image Processing (ICISIP2017)*, pp.440-443, 2017.

[9] I. Taguchi, K. Watanabe, M. Shimakawa and K. Kiyota, Obstacle detection by machine learning using RGB images for visually impaired, *International Conference on Innovative Engineering (ISIE2018)*, pp.78-83, 2018.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, *Rethinking the Inception Architecture for Computer Vision*, arXiv:1512.00567, 2015.