Convolutional Neural Network Based Zebra Crossing and Pedestrian Traffic Light Recognition

Shinji Eto[1,a,*], Yasuhiro Wada[2,b], and Chikamune Wada[1,c]

[1]Human Intelligence and Technology, Graduation School of Life Science and System Engineering, Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu-ku, Kitakyushu-shi, Fukuoka, Japan

[2]maris Co. Ltd., 3-6-10 Kyobashi, Chuo, Tokyo

*Corresponding author

[a]<eto.shinji786@email.kyutech.jp>, [b]<yasuhiro.wada@maris001.onmicrosoft.com>, [c]<wada@brain.kyutech.ac.jp>

Keywords: image processing, deep neural networks, visually impaired, blind, convolutional neural network (CNN)

Abstract. The study proposes a convolutional neural network (CNN) method for providing guided assistance for the visually impaired and blind (VIB) at zebra crossings with pedestrian traffic lights (PTL). It seeks to improve existing methods by providing locational orientation as well as navigation guidance in travel direction for the VIB at the crossing. Images of zebra crossings with a PTL were taken with a camera under sunny conditions. However, classifying PTL labels is difficult due to image background assimilation. Nevertheless, this could be resolved by using high resolution images or applying preprocessing to improve image contrast for highlighting the PTL outline. Although the study CNN outperforms state-of-the-art methods in accuracy and calculation time, a limitation of the model performance is that it would decrease when the input image is taken in different environments from the dataset. This problem can be addressed by collecting images under various environmental circumstances.

1. Introduction

When walking alone, one of the riskiest activities for the visually impaired and blind (VIB) is traversing the crosswalk. Due to a lack of visual information, the VIB are unable to identify signals on the pedestrian traffic lights (PTL) and the travel direction. As a result, they may cross at the incorrect time or walk toward the wrong direction.

Methods for assisting the VIB using only image processing which enable accurate zebra crossing detection and PTL recognition have been proposed [1, 2, 3]. However, there are many different types of crosswalks, and there is no uniform method that can be applied to all these. Although it is possible to use several different methods depending on the crosswalk, this is not a realistic approach because it requires countless conditional branches in the system.

Recently, methods combined with deep learning techniques have been proposed which achieve superior generalization performance. LytNetV2 [4] is an excellent convolutional neural network (CNN) that estimates the zebra crossing midline and recognizes PTL in real-time with high accuracy. The researchers used the zebra crossing midline to estimate the VIB travel direction. LytNetV2 was trained with a dataset which included several crosswalk images and ground truth of PTL label and midline, called PTLR dataset [5]. However, LytNetV2 cannot provide the VIB point of orientation because the assistive system or application could not activate itself at an appropriate timing without the VIB positional information. Therefore, the system must obtain the VIB positional information using other methods or additional sensors such as GPS. The implementation of such methods would either increase memory consumption or decrease processing speed. Moreover, ground truth in the

PTLR dataset may contain ambiguous information of the midline. This is because the midline ground truth was obtained by connecting the top line midpoint (the white line to start crossing) to the bottom midpoint (the white line to finish crossing) of the zebra crossing without measuring the actual midpoint itself. Therefore, the plotting to the image is done instinctively. In addition, some images do not capture the entire crosswalk. As a result, the differences of plotting positions among annotators become more significant. Therefore, it is necessary to define other ways to estimate the VIB travel direction.

The purpose of this study is to propose a highly accurate and small model that outputs PTL labels, VIB position labels, and the zebra crossing angle, instead of the zebra crossing midline, to reduce the ambiguity of ground truth. PTL labels and VIB position labels will be explained in detail in section 3. In addition to the proposed model, a dataset with consistent ground truth against annotators is created. The optimal model architecture and hyper parameter settings are decided using this dataset. Comparison experiments with state-of-the-art methods show that the proposed model achieves higher accuracy in less computation time.

2. Related work

Navigating through a crosswalk is one of the most dangerous daily activities for the VIB. To mitigate the danger, detecting zebra crossing and feeding back a safe route to the VIB is necessary. There are several attempts using Hough transform to detect zebra crossing [1,2,3]. These methods detect zebra crossing by using a combination of Hough transform and edge detection such as Canny Edge detection and Sobel edge detection. Uddin et al. [6] proposed a recognizing method by featuring the characteristic factors such as black-and-white pattern, zebra crossing width, orientation, and number of bands. Meanwhile, Wang et al. [7] proposed an algorithm to recognize zebra crossing and staircase using RGB-D information. RGB information was used to detect parallel lines from the image, while depth information was used to classify zebra crossings and stairs. In another study, Asami et al. [8] created an algorithm to detect zebra crossings by applying template matching using normalized cross-correlation to binarized input images.

While image processing based methods are capable of highly accurate detection and recognition, these suffer from generalization performance problems. To achieve highly accurate detection and recognition at a variety of real-world crosswalks, it is necessary to combine image processing and deep learning technology.

Wu et al. [9] proposed an algorithm to recognize zebra crossings using parallel line angles and cumulative scores to locate zebra crossings. In contrast to previous image processing based methods, zebra crossings were detected by applying a regression approach using CNN [10]. Images cropped by a fixed-size sliding window were identified as zebra crossings by a logistic regression model while directional predictions of zebra crossing were made by CNN. Meanwhile, Yu et al. [5] proposed a deep learning model called LytNet to predict the PTL labels and the midline of zebra crossings. Using these two important sources of information in assisting the VIB with crosswalk, their study built an application that provides guidance on where to stand, which direction to go, and when to cross the street.

Two related studies by Wu et al. [9,10] estimate the safe area on zebra crossing and enables the VIB to course through a safe direction. However, their method must be used in conjunction with other methods to recognize the PTL colors. As a solution, LytNet is an excellent CNN that can simultaneously detect zebra crossing and predict the PTL colors. To obtain locational information on the VIB, however, it is necessary to use additional methods or sensors such as GPS. Considering the memory consumption and processing speed, it is desirable to predict not only PTL colors and the midline of a zebra crossing but also the VIB locational information simultaneously.

3. Proposed method

3.1 GhostNet

An ordinary CNN needs a large amount of parameter and floating point operations (FLOPs) to obtain enough accuracy on classification and regression tasks. In recent years, high accuracy CNN that can be run on mobile devices have been proposed due to reductions in the number of parameters and FLOPs along with significant improvements in memory efficiency.

Han et al. [11] proposed Ghost module, Ghost bottleneck, and GhostNet. GhostNet is created by combining Ghost bottlenecks with convolutional and fully-connected layers. Meanwhile, Ghost bottleneck is composed of Ghost module which is lighter and faster than the usual convolution layer.

In this study, GhostNet [11], one of the lightest and most accurate state-of-the-art models, is used as a baseline to predict essential information that assists the VIB at the crosswalk in real-time.

3.1.1 Ghost Module

Past research [11] has indicated that well-trained deep neural networks have many similar pairs of feature maps, that is, like a ghost of each other. Therefore, authors called them ghost feature maps. In Ghost module, ghost feature maps are created by applying a series of linear transformation with cheap calculation cost to a handful of intrinsic feature maps. Outputs of Ghost module are obtained by concatenated intrinsic and ghost feature maps. Assuming that the number of intrinsic feature maps and linear operation for generating ghost feature maps are m and s, respectively, the number of ghost feature and output of Ghost module are $m \cdot (s-1)$ and $m \cdot s$, respectively. Theoretically, ordinary convolution with Ghost module is s times faster and lighter than without Ghost module.

3.1.2 Ghost Bottlenecks

Ghost Bottleneck has almost the same architecture as basic residual block in ResNet [12] which consists of two Ghost modules, depthwise convolution, and skip connection. The first Ghost module increases the number of input channels while the second decreases this to match to skip connection. The output of Ghost bottlenecks are obtained by concatenating the second Ghost module and inputs. Similar to the study by Sandler et al. [13], only batch normalization is applied after the second Ghost module, while batch normalization and ReLU are applied after the other layers.

3.2 Proposed Convolutional Neural Network

The original GhostNet had excess performance for this study because GhostNet was designed for datasets which have more images and classes [14, 15]. However, in this study, the CNN requires a smaller number of outputs. Therefore, a model smaller than the basic GhostNet was designed.

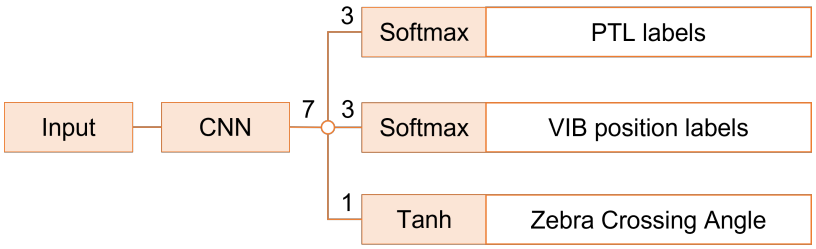

Fig. 1. Overall view of the proposed CNN. The softmax function is applied to both the PTL and VIB position labels while the Tanh function is applied to the zebra crossing angle. The number of CNN outputs is 7. The numbers of both PTL and VIB position labels are 3. Meanwhile, Zebra crossing angle is a single numeric output.

Table 1 shows the CNN architecture, where output from the CNN is separated, and the appropriate activation function applied to each output (Fig. 1).

Table 1. The proposed CNN architecture. **G-bneck** denotes Ghost Bottleneck. **k** denotes kernel size. **exp** denotes expansion size. **c** denotes channel size. **SE** denotes Squeeze-And-Excite. A check mark (✓) indicates when **SE** is used. **Stride** denotes stride.

Input	Operator	k	exp	c	SE	Stride
640 × 480 × 3	Conv2d	3	-	16	-	2
320 × 240 × 16	maxpool	2	-	-	-	2
160 × 120 × 16	G-bneck	3	16	16	-	1
160 × 120 × 16	G-bneck	3	64	24	-	2
80 × 60 × 24	G-bneck	3	72	24	-	1
80 × 60 × 24	G-bneck	5	72	40	✓	2
40 × 30 × 40	G-bneck	5	120	40	✓	1
40 × 30 × 40	G-bneck	3	240	80	-	2
20 × 15 × 80	G-bneck	3	200	80	-	1
20 × 15 × 80	G-bneck	3	480	112	✓	1
20 × 15 × 112	G-bneck	5	672	160	✓	2
10 × 8 × 160	G-bneck	5	960	160	✓	1
10 × 8 × 160	G-bneck	3	960	320	-	1
10 × 8 × 160	Conv2d	1	-	960	-	1
10 × 8 × 960	avgpool	-	-	-	-	-
960	FC	-	-	1280	-	1
1280	FC	-	-	7	-	-

3.3 Proposed Dataset

3.3.1 Label of Classification

Fig. 2 shows samples from the dataset. The study dataset consists of a total of 5640 crosswalk images captured in Fukuoka, Japan. Images were created by cropping one image every five frames from videos with a resolution of 640×480 and 30 FPS. The videos were shot during a sunny day at a crosswalk with a PTL and tactile paving's.

Because the camera shutter speed and the PTL frequency caused disappearing PTL, the PTL labels were defined as: red, green, and none (Fig. 2). A label of "none" includes images without the PTL. The VIB position labels was categorized into three labels: approaching, crossing, and others.

(a) the PTL label is "green" and the VIB position label is "crossing."

(b) the PTL label is "red" and the VIB position label is "approaching."

(c) the PTL is "none" and the location information is "others."

(d) PTL appears to be unlit due to PTL frequency and camera shutter speed.

Fig. 2. Dataset sample images.

3.3.2 Ground Truth of Regression

Yu et al. [4,5] utilized the zebra crossing midline to estimate the direction of travel for the VIB. However, as pointed out in the first section, plotting two midpoints at the top line and bottom line of the zebra crossing was done instinctively. To eliminate dataset ambiguity, the zebra crossing angle is utilized to estimate the VIB direction of travel because of its obvious definition. By plotting two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ on the top-middle and bottom-middle of zebra crossing, the angle θ is calculated as follows:

$$\theta = arctan\frac{y_1 - y_2}{x_2 - x_1} \tag{1}$$

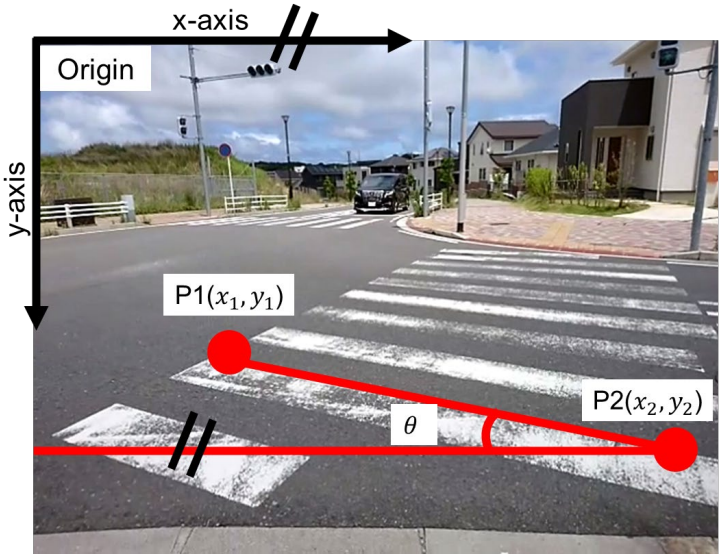

Fig. 3. Illustration on where the two points are plotted. The origin is in the upper left corner. The point closer to the origin is P_1 and the point farther from the origin is P_2. θ is the zebra crossing angle.

3.4 Training Setting

The proposed CNN outputs the PTL labels, VIB position labels, and angle of zebra crossing. Also, the CNN was trained with a custom loss function which consists of classification and regression loss function. Cross-entropy was used to define the classification loss function:

$$L_{PTL}(p, q) = -\sum_{n=1}^{N}\sum_{m=1}^{M_c} p_{n,m} \log q_{n,m} \tag{2}$$

$$L_{position}(p, q) = -\sum_{n=1}^{N}\sum_{m=1}^{M_p} p_{n,m} \log q_{n,m} \tag{3}$$

where p is true label, q is prediction, n is index of data, m is class of data, and L_{PTL}, $L_{position}$ are the loss functions for PTL labels and VIB position labels, respectively. Since estimating the angle of zebra crossing is a regression task, mean squared error was used for the angle loss function:

$$L_{angle}(p,q) = \frac{1}{n}\sum_{n=1}^{N}(q_n - p_n)^2 \quad (4)$$

where $L_{angle}$is loss function for the zebra crossing angle. By combining three loss functions, the custom loss function is defined as follows:

$$L(p,q) = \alpha L_{PTL}(p,q) + \beta L_{position}(p,q) + \gamma L_{angle}(p,q) \quad (5)$$

where α, β, and γ are adjustable hyperparameters. When training the CNN, α=1e^{-4}, β=1, and γ=1e^{-1} were used. The CNN was trained with NVIDIA RTX A2000 GPU. The batch size was 16 and the optimization function was Adam with default parameters. Also, the learning rate was a fixed value of $1e^{-4}$.

Table 2 shows the details of study dataset. Of the dataset, 70[%] was used for training. Of the remaining images, 55[%] were used for validation and 45[%] for testing. Finally, 3947 images were used as training, 931 images as validation, and 762 images as testing.

To prevent the neural network from over-fitting, data augmentation was performed on the dataset. Online data augmentation was applied to each mini-batch during the training. Left-right flipping and rotation was applied while luminance, contrast, and saturation were changed randomly.

Table 2: Dataset details. **Training**, **Validation** and **Testing** denote training, validation, and testing data details respectively. **Red**, **Green** and **None** denote the numbers of data in each PTL label. **Crossing**, **Approaching** and **Others** denote the numbers of data in each VIB position label.

Label	PTL			VIB position			Sum
	Red	Green	None	Crossing	Approaching	Others	
Training	1291	1117	1539	3156	355	436	3947
Validation	308	269	354	704	109	118	931
Testing	259	222	281	617	63	82	762
Sum	1858	1608	2174	4477	527	636	5640

4. Experiments

Optimal values of tunable hyper parameters were searched to obtain higher accuracy. The accuracy and computational speed of the tuned CNN was compared with other methods [4]. Since the outputs of the other methods and the proposed CNN are different, the CNN outputs were modified to match the outputs of other methods and was trained on the same dataset by Yu et al. [4]. All experiments conducted in this section were performed with Intel Core i7-11700K 3.6 GHz CPU and 32 GB RAM.

4.1 Tuning our CNN

Past research [11] suggested that the number of channels in each layer can be customized by factor α which is called width multiplier, and allows control of model size and the computational cost quadratically. By adjusting α, the trade-off between model accuracy and computational speed can be investigated. We used α = 0.35, 0.5, 0.75, and 1.0.

Table 3: Comparison of the proposed CNN trained on the study dataset for each width multiplier. **PTL** denotes the average classification accuracies of PTL labels. **Position** denotes the average classification accuracies of VIB position labels. **Angle** denotes the average regression errors of the angle of zebra crossing. **CT** denotes calculation times.

Width Multiplier	PTL[%]	Position[%]	Angle[degrees]	CT[ms]
0.35	82.02	97.51	12.65	**16.25**
0.5	85.43	98.29	11.65	18.67
0.75	84.91	98.34	8.98	23.06
1.0	**88.19**	**98.58**	**8.03**	29.31

Table 3 shows that as width multiplier increases, classification accuracy and regression error improve and calculation time increases. We decided width multiplier to 1.0 since we could obtain good performance with enough fast calculation time. After we decided the optimal value of width multiplier, we examined the usefulness of Maxpool layer.

Yu et al. [4] used Maxpool layer to improve calculation time. The addition of the Maxpool layer allows the model to reduce the number of parameters and FLOPs while preserving decreasing model performance. The impact of adding the Maxpool layer was explored in the study model. Specifically, predicting accuracy of PTL labels, VIB position label, and zebra crossing angle between models was compared with and without a Maxpool layer. Estimation accuracy was measured while varying α. Only the result of classifying PTL labels is shown because the results for VIB position classification and angle estimation have the same trend.

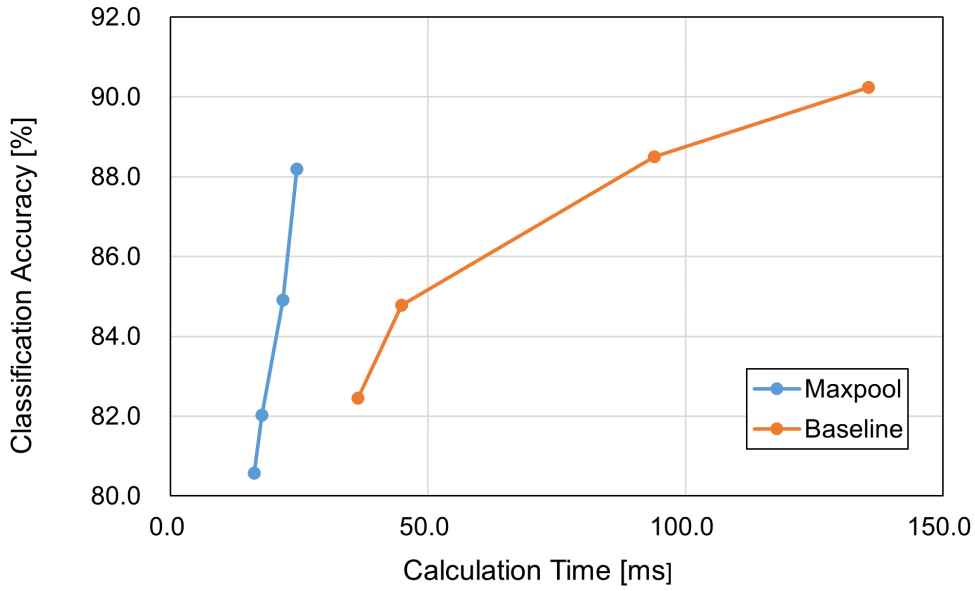

Fig. 4. Model performance comparison with and without Maxpool layer. The horizontal axis is the computation time, and the vertical axis is the accuracy of classifying PTL labels. We used α = 0.35, 0.5, 0.75, and 1.0.

Fig. 4 shows a model performance comparison on the classification accuracy of PTL labels. This figure shows accuracy of PTL labels only because the classification accuracy of VIB position labels and regression error of zebra crossing angle have a similar trend. When α was low, classification accuracy was low and calculation time was fast, and vice versa. Fig. 5 indicates that the study model can achieve better performance with faster calculation time due to the Maxpool layer insertion. Therefore, a Maxpool layer was added to the proposed CNN.

4.2 Comparison with State-of-the-art CNN

The study CNN was compared to the state-of-the-art model which is called LytNetV2 [26]. LytNetV2 is designed to output PTL labels as well as the start and end points of a zebra crossing midline. Because of the different output, the number of output layers was only modified in the study CNNs. In addition, the proposed CNN was trained on the PTLR dataset under the settings by Yu et al. [4] because LytNetV2 is trained on PTLR dataset.

Table 4 shows a model size comparison between the study CNN and other methods. It indicates that the CNN is smaller than LytNetV2. Therefore, the study model memory consumption will be less than LytNetV2.

Table 4. Model size comparison

Model	Size [MB]
LytNetV2	14.43
Ours	**12.51**

Table 5 shows a comparison between the study CNN and LytNetV2 in terms of classification accuracy, regression error, and computation time. The CNN obtains better accuracy and less regression error with less computation time.

Table 5. Comparison of accuracy, error, and calculation speed. **A-err** means average angular error. **Sp-err** means average starting point error. **Ep-err** means average end point error. **Ct** means average calculation time.

Model	Accuracy [%]	A-err [degrees]	Sp-err [−]	Ep-err [−]	CT [ms]
LytNetV2	92.01	6.98	0.096	**0.081**	69.93
Ours	**93.51**	**6.96**	**0.094**	0.083	**66.12**

5. Discussion

Although study dataset has almost same number of images in each PTL label, the number of VIP position label is uneven. This is because study dataset is created by clipped videos which is captured while crossing the crosswalk. When the PTL turned red while we captured video, we continued capturing before the crosswalk. However, videos categorized as "approaching" and "none" can be captured more easily than ones categorized as "crossing". This is because it is not necessary to cross a crosswalk while we capture the video and can be captured before the crosswalk or on the sidewalk. Therefore, disproportionateness of study dataset would be able to be solved easily.

The study observed classification accuracy of PTL labels was relatively small compared to the classification accuracy of VIB position labels (Table 3). This is because classifying PTL labels is more difficult than for VIB position labels. Since the study dataset is composed of images clipped from videos, it includes images in which the PTL is missing. Moreover, when the background was dark and assimilated with the PTL outline, the PTL was not visible in the image (Fig. 5). This problem can be resolved by using high resolution images or applying preprocessing to improve image contrast for highlighting the PTL outline.

Fig. 5. Image in which the PTL outline, and background are assimilated.

Difference between study model and original GhostNet is the Maxpool layer in the shallow layer. The Maxpool layer reduces the size of the feature map by extracting the maximum value in the local region, resulting in reduced computational cost. Moreover, Maxpool layer contributes to the dimensionality reduction by inserting it in shallow layer of study model. High dimension input would make optimization of the parameters difficult. Since Maxpool layer reduced computational cost and input dimension, study model with Maxpool layer could be optimized easily and achieved good performance with fast calculation time.

Comparing study model with Maxpool layer and without Maxpool layer, the latter achieved better classification accuracy. However, the increase in calculation time was greater than the increase in classification accuracy (Fig. 4). This is because high dimension input would make optimization of parameters in study model difficult.

6. Conclusion

In this study, a CNN with higher accuracy and faster calculation time was proposed. Furthermore, a dataset with consistent ground truth against annotators was created. The effects of width multiplier and the Maxpool layer were verified in experiments. Results of the experiments indicate that the Maxpool layer significantly improves the accuracy and calculation time trade-off. The classification accuracy of PTL labels is 88.19[%] and VIB position labels is 98.58[%] while the regression error of zebra crossing angle is 8.03[degrees] with fast calculation speed when the width multiplier is set to 1. The study CNN has enough real-time performance. Although the classification accuracy of PTL labels is relatively low, better performance is achieved by increasing the width multiplier.

Furthermore, the study CNN was compared with state-of-the-art methods using PTLR dataset. It was found that the CNN outperforms state-of-the-art methods in both accuracy and calculation time. The classification accuracy is 93.51[%] and angle error is 6.96[degrees] with faster calculation speed.

A limitation of this study is that the model performance would decrease when the input image is taken in different environments from the dataset. The study obtains images at crosswalks with PTL and tactile paving's on a sunny day. Since the study uses deep learning, predicting for inputs with different environments from the study is difficult. This problem can be improved by collecting images under various environmental circumstances.

References

[1] S. Se, "Zebra crossing detection for the partially sighted", ***Proceedings of CVPR2000*** (Hilton Head, SC, USA), June 2000.

[2] R. Cheng, K. Wang, K. Yang, N. Long, W. Hu, H. Chen, B. Jian and D. Liu, "Crossing navigation for people with visual impairments on a wearable device", ***Journal of Electronic Imaging***, Vol.26, No.5, p.053025, 2017.

[3] M.S. Uddin and T. Shioyama, "Robust zebra-crossing detection using bipolarity and projective invariant", ***Proceedings of the Eighth International Symposium on Signal Processing and Its Applications*** (Sydney, Australia), August 2005.

[4] S. Yu, H. Lee and J. Kim, "Street crossing aid using light-weight CNNs for the visually impaired", ***Proceedings of CCVW2019*** (Zagreb, Croatia), October 2019.

[5] S. Yu, H. Lee and J. Kim, "Lytnet: A convolutional neural network for real-time pedestrian traffic lights and zebra crossing recognition for the visually impaired", ***Proceedings of CAIP2019*** (Salerno, Italy), September 2019.

[6] M.S. Uddin, and T. Shioyama. "Detection of pedestrian crossing using bipolarity feature-an image-based technique", ***IEEE Transactions on Intelligent Transportation Systems***, Vol.6, No.4, pp.439-445, 2005.

[7] S. Wang, H. Pan, C.Zhang, and Y. Tian, "RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs", ***Journal of Visual Communication and Image Representation***, Vol.25, No.2, pp.263-272, 2014.

[8] T. Asami and K. Ohnishi, "Crosswalk location, direction and pedestrian signal state extraction system for assisting the expedition of person with impaired vision", ***Proceedings of MECATRONICS2014*** (Tokyo, Japan), November 2014.

[9] X. Wu, R. Hu and Y. Bao, "Block-based Hough transform for recognition of zebra crossing in natural scene images", ***IEEE Access***, Vol.7, pp.59895-59902, 2019.

[10] X.H. Wu, R. Hu and Y. Q. Bao, "A regression approach to zebra crossing detection based on convolutional neural networks", ***Journal of IET Cyber-Systems and Robotics***, Vol.3, No.1, pp.44-52, 2021.

[11] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu and C. Xu, "Ghostnet: More features from cheap operations", ***Proceedings of CVPR2020*** (Seoul, Korea), June 2020.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", ***Proceedings of CVPR2018*** (Las Vegas, America), June 2018.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks", ***Proceedings of CVPR2018*** (Salt Lake, USA), June 2018.

[14] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar and C. L. Zitnick, "Microsoft coco: Common objects in context", ***Proceedings of ECCV2014*** (Zurich, Switzerland), September 2014.

[15] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database" ***Proceedings of CVPR2009*** (Miami, USA), June 2009.