International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 14 (2022) pp. 100-109

Received: 1 October, 2021; Accepted: 2 February, 2022; Publish: April 21, 2022

Pedestrian Crossing Detection and Recognition Based on Two Connected Point and Uniform Local Binary Pattern

Md. Khaliluzzaman

Dept. of Computer Science and Engineering,
International Islamic University Chittagong (IIUC), Chattogram-4318, Bangladesh
khalil@iiuc.ac.bd

Abstract: **A geometrical feature base pedestrian crossing (PC) region detection and recognition framework is proposed in this work. A unique feature is that each end point of the horizontal strip edges of pedestrian crossings is intersected with a vertical stripe width edge. That makes up two connected point (2CP). Another feature is that the edges of the PC stripe are formed in ascending parallel order. By utilizing these two features the PC candidate region is detected from the PC image. Where, the PC region is validated and justified by using 2CP and arranging the horizontal parallel edge segment in sorted order respectively. Finally, the potential PC region is confirmed by a classifier i.e., support vector machine (SVM). Here, the features of the candidate area are extracted using a rotationally invariant uniform Local Binary Pattern (LBP). The proposed method is tested with our own dataset and results reveal significant improvement with respect to the existing works.**

Keywords: Pedestrian crossing, two connected point, rotational invariant, uniform LBP, SVM

I. Introduction

Detection of a specific object from an input image or a video is the fundamental steps in the field of computer vision (CV). Pedestrian crossing (PC) detection and recognition from pedestrian crossing image is also an important task in CV. This task is important for the autonomous navigation system and visually impaired people to navigate automatically in an unknown environment safely. It is a vital issue for advanced visual impaired assistance system (AVIAS) to warn impaired people during pedestrian crossing navigation. The system is also crucial for avoiding a vehicle-pedestrian collision during crossing the road.

In the practical environment, the pedestrian crossing detection is difficult because of its different shapes, various color shading, and viewpoints. To overcome these limitations as well as to satisfy the demand of autonomous navigation systems, the detection and recognition process of pedestrian crossing should be evaluated as fast as possible in different environmental conditions.

For satisfying the above criteria and increasing the pedestrian crossing safety, a pedestrian crossing region detection and recognition framework need to be developed. So that, the developed framework may recovers the existing methods problems and works at the different environmental conditions with various viewpoints to increase the detection and recognition accuracy. For this regard, a framework is presented in this research paper by utilizing pedestrian crossing's geometrical features. One feature is a pedestrian crossing (PC) stripe's horizontal edge (HE) ending points intersect with stripe's vertical edge (VE) end points. This intersection point makes up two connected point (2CP). This 2CP is as shown in Figure 1(a). Another feature is PC's HEs are arranged in concurrent sorted order. The sorted HEs feature is shown in Figure 1(b).

Figure 1 Geometrical property of pedestrian: a) Two connected point (2CP), and b) increasing HEs in parallel order

These two features are shown to the human according to the human vision if the human observes the PC from its front side through a litter distance. For that initially, the input PC image is filtered through the Gabor filters to reduce the different noisy illuminations and resolute the PC's horizontal edges. After that, from this image, the non-potential edges information is removed by utilizing a non-interest edge eliminating procedure. Furthermore, an edge linking process is applied to extract the longest horizontal edge segment. Finally, the aforementioned geometrical features are utilized on the extracted longest horizontal edges to detect the PC's region of interest (ROI), which is the key contribution of this paper. Here, pedestrian crossing horizontal edges are validated through the 2CP geometrical feature. The second geometrical feature i.e., arranging the horizontal parallel edge segment in sorted order is used to justify the validate pedestrian crossing horizontal edges. The extracted pedestrian crossing region is recognizing through the SVM. For that, the features are extracted by utilizing the rotational invariant uniform LBP. However, the main contribution of this paper is listed below.

- Unique Geometrical feature based pedestrian crossing detection and recognition framework is proposed.
- Gabor filters are used to eliminate the nose and shadows effect from image.
- Edge linking and tracking process is introduced to detect the potential horizontal edges.

- Developing a pedestrian crossing dataset which contains different oriented pedestrian crossing images with various environmental and elimination conditions.

In this paper, the state of the arts, i.e., related work is described in Section II. The step by step explanation of the proposed method is demonstrated in Section III. The results of the proposed model are demonstrated in Section IV. Finally, the paper concludes in Section V.

II. Related Works

In a few decades, researchers and scientist have developed many frameworks to elevate the perfection of the PC detection and recognition. This section describes some effective existing methods that are used to enhance the effectiveness of the PC detection and recognition. Such as, a coarse to fine based pedestrain crossing framework is introduced in [1]. Here, the pedestrain in detected from probe vehicle video. To detect the exact location of the pedestrain the vehicle video is combined with the GPS traces. some techniques of image processing are used for detecting the pedestrain crossing is introduced in [2]. The techiques used in this work are adaptive-histogram equalization, Flood-fill operation, and Hough transformation. These techniques are used to extract the PC region of interest (ROI). The detected ROI is recognized by using support vector machine (SVM). A vehicle-based Mobile Mapping System is introduced for automatic pedestrian crossing detection with monocular vision in [3]. Here, contour information is analyzed and crossing features are trained to detect and recognize the pedestrian crossing.

In [4], the authors introduce a method to identify the conflict of the vehicles and pedestrian at the pedestrian crossing area, where, drivers and pedestrians have not maintained the rules of the traffic. Here, the method is established based on the Probabilistic Navigation Function. In this model, at the pedestrian crossing area, the pedestrians establish a virtual risk map for a collision with the vehicles with the probability. The action of the driver and pedestrians are selected based on their perceived probability of collision. In [5], the authors propose a method to localize the crosswalk from the low-resolution satellite images, where, the images are captured through the Smartphone from the Google map. In [6], authors introduce a zebra-crossing detection system based on an on-board monocular camera. Here, the zebra crossing region is detected by applying the horizontal projection based integral method.

A zebra-crossing detection method based on a geometrical aspect of a zebra-crossing was presented in an earlier edition of this paper [7]. The main feature utilized in this paper is arranging the increasing horizontal edges in sorted order. Finally, the potential ROI is classified through the vertical vanishing point (VP) [27]. However, the VP is not suitable in all the environments to verify the pedestrian crossing region. That is because of stair region also shows this same property as like pedestrian crossing [21].

In [8], the authors presented a pedestrian crossing (PC) behavioral method based on the statistical analysis with and without vehicles. This method detects and tracks the PC from the video, so that, the method can track the pedestrians those are passing through the PC and reduce the collision between the pedestrian and vehicle. A depth information based pedestrian crosswalk method is proposed in [9], where depth information is extracted from the RGB-D image. Where, the crosswalk region is indentified by extracting the parallel crosswalk strip's lines and SVM classifier is used to recognize. The features used in this work are depth feature. In [10], the authors proposed a method to detect the PC in an urban area for pedestrian safety. In this method, the authors use Fourier and Hough transformation to extract the lines and angels form PC. Then ROI is detected by augmented bipolarity. Finally, classify the pedestrian crossing by the SVM classifier, where, the template matching process is utilized for feature extraction. An analysis of natural movement of pedestrians is presented in [31].

A data-driven approach is introduced in [11] to extract the zebra-crossing from aerial imagery. Here, geospatial data are used to examine the zebra-crossing region for learning an appearance model. Finally, HOG with LBPH features is utilized to classify the zebra-crossing through SVM. A generic context-based model is used in [12] to estimate the crossing behaviors of a pedestrian in the inner city. Local threshold segmentation based zebra-crossing detection method is proposed in [13], where, images are captured by Inverse Perspective Mapping. Finally, datum band information of zebra-crossing ROI is analyzed to identify the region of zebra-crossing.

For road management, in [15], a method is proposed to extract the region of zebra-crossing automatically from the mobile LiDAR data. In this method, zebra-crossing is detected based on the Hough Transform and logical constraints. An online-based PC detection procedure is presented in [25]. The procedure utilizes the pre-existing traffic-oriented video-sensors. In this work, the evidential based data fusion process is introduced to recover the missing primary sensor data from the second one and improve the accuracy of the proposed framework.

Earlier, a bipolar region based pedestrian crossing detection method is proposed in [14]. Before that, smartphone-based crosswalks detection method is introduced in [16]. A coherent structure of the crosswalk geometrical feature is used in [17] to detect the crosswalk. Vanishing point based pedestrian detection method is introduced in [7] [19]. Automatic pedestrian detection with dynamic traffic sign is introduced in [18]. This system is used to improve the safety of the pedestrian crossing.

From the above state of the arts analysis, it is observed that the scientist and researchers have been proposed many pedestrian crossing (PC) detection methods based on the geometrical feature [7], depth feature [9], the bipolarity theory [10][14], coherent structure [17], template matching [14] and so on. However, these methods are not performed better for their limitations. Such as, the geometrical feature used in the [7] is the horizontal edges are arranged in alphabetical order. This feature is extracted by using dynamic programming which required $O(n \log n)$ time. Moreover, the methods used the depth feature are not performed well at the outdoor environment due to the depth sensor's limitations. In noisy and dark environments, bipolarity approaches do not perform better.

The template matching methods required the much time to identify the candidate region of interest. Furthermore, the indentified ROI is recognized by utilizing the different recognition method such as vanishing point [7] [19]. However, this vanishing point method shows the similar property as a pedestrian crossing for various similar pattern objects such as rail-line and stairways. In this work, a framework is proposed based on the pedestrian crossing's geometrical properties. The essential feature is that the PC stripe's horizontal edge (HE) ending points connect with the vertical edge (VE) finishing points. This issue is linear and takes $O(N)$ time to solve. Another feature is concurrent horizontal edges are arranged in sorted order. This problem is solved by comparing the horizontal edge's x coordinate values. By utilizing these features the PC region of interest is detected. Finally, the detected region of interest is recognized through the SVM classifier.

III. Proposed Framework

To detect the pedestrian crossing (PC) region from input pedestrian crossing image, the unique features of the PC have to be identified from the PC image. This framework explains the process of identifying key geometrical features of a PC from the PC image to detect the PC's ROI. Detecting the PC's ROI by utilizing the PC geometrical features, i.e., two connected point (2CP) and arranging the horizontal parallel edge segment in sorted order is the prime contribution of the proposed framework.

The PC region recognition process is also explained in this section. The suggested framework consists of six key phases. Those are: (1) filtering and extracting pedestrian crossing image edges, (2) removing non-potential edges, (3) detecting the longest horizontal edges, (4) finding two connected point, (5) arranging HEs in concurrent sorted order and detect the pedestrian crossing region of interest, and (6) finally, feature extraction and classification is performed for verifying the pedestrian crossing candidate region by SVM classifier. The pedestrian crossing detection framework is depicted in Figure 2.

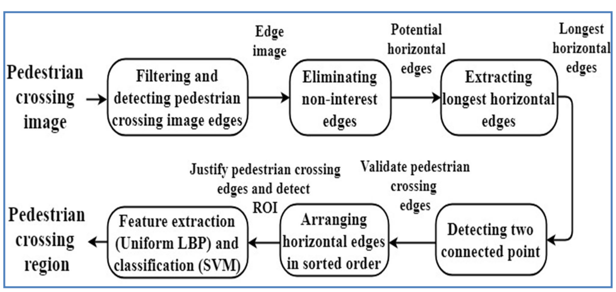


Figure 2 Proposed pedestrian crossing detection and recognition framework

A. Filtering and extracting pedestrian crossing edges

To extract the pedestrian crossing (PC) edge information from the pedestrian crossing input image which conveys the features of the PC, initially, the noise and shadows effect have to be eliminated from the PC input image. For that, in this work, the Gabor filters are used to reduce the illumination effect and resolute the pedestrian crossing's potential horizontal and vertical edges. Before applying the Gabor filter the input image is transformed to the gray-scale image to minimize the execution complexity. The Gabor's filter used in this work is presented in (1), (2), and (3) [20] [28].

$$G_{\lambda,\theta,\varphi,\sigma,\gamma}(x,y) = exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right) cos(2\pi \frac{x'}{\lambda} + \varphi) \quad (1)$$

$$x' = xcos\theta + y\, sin\theta \quad (2)$$

$$y' = -xsin\theta + y\, cos\theta \quad (3)$$

Here, (x,y), λ, and θ represent the image coordinate, the wavelength of cosign factor and angular orientation respectively. ψ denotes the filter phase offset. γ and σ represents the spatial aspect ratio and standard deviation respectively. By the Eq. (1) the Gabor filter works only on the x and y-axis. The Eq. 1 does not work diagonally. To work the Gabor filter in any desired orientation, in this work (2) and (3) are introduced. By these equations, Gabor filters can work at any orientation defined as θ at the coordinate of *(x', y')*. The response of the Gabor filter over the PC image is presented in Figure 3(b).

To validate and justify the geometrical features of the PC, the horizontal as well as vertical edge information is needed to be detected from the PC image. For that, Laplacian operator [29] based Canny edge operator [26][28] is utilized in this work. This operator is used because of its advantages such as smoothing and thresholding. Here, the smoothing is used to eliminate the Gaussian noise and the thresholding is used to detect the PC edges effectively. Furthermore, this operator has the ordinary attributes such as standard deviation and threshold values [30]. The extracted Canny edge image from the preprocessed filtered image is shown in 3(c).

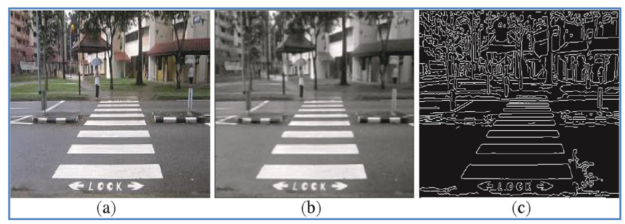


Figure 3 Examples of filtering and finding edges: a) input pedestrian-crossing image, b) result of Gabor filtered image, and c) result after applying the canny edge image

B. Removing non-potential edges

Basically, the Canny operator extracts the multi-directional edges. To validate and justify the geometrical features of a pedestrian crossing, the vertical and horizontal edges are most needed. For that, from the Canny edge image, the VEs are removed. As a result, the HEs may be largely present in the PC Canny edge image. The extracted HEs are depicted in Figure 4(a).

As seen in Figure 4(a), the pedestrian crossing (PC) edge image still contains numerous small edges. A suggested filtering procedure removes these tiny edges from the PC edge image. A statistical threshold value is used in the filtering procedure to remove tiny non-potential edges. Edges with a length larger than the statistical threshold value are kept. The threshold value is estimated by $\sqrt{\frac{\sum_{i=1}^{N}(edge(i)-edge\_mean)^2}{N}}$. Here, N is the total count of horizontal edges (HEs) that are present in Figure 4(a). The standard deviation of the edge length values shown in Figure 4(a) is the estimated threshold. The result of removing the small edges is as shown in Figure 4(b). As the small and vertical edges are eliminated from edge image, now in the edge image mainly remain the potential HEs. However, in the edge image still remain some edge segments which are not concurrent to the potential horizontal edges. These non-concurrent HEs are taken as non-interest edge. These non-interest horizontal edges are removed from the HE image.

The concurrent horizontal edges without non-potential edges are depicted in Figure 4(c).

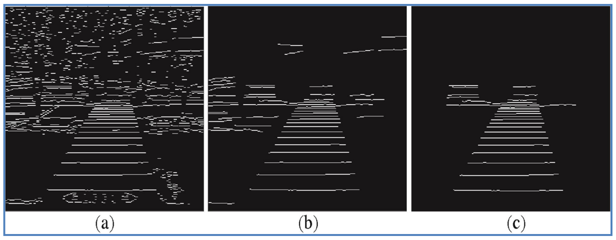


Figure 4 Examples of non-interest edge elimination: a) eliminating vertical edges, b) filtering small edges, and b) removing non-interest edges.

C. Extracting longest horizontal edges

In the previous section, the non-candidate edges are eliminated. So, at the present in the edge image mainly remains potential concurrent horizontal stripe lines. Since the potential HEs are extracted from the pedestrian crossing stripes and these stripes are presented in different colors. These colorful shapes in the stripes are not smooth all the time for different reasons. So, in the horizontal edges may have some breaks or gaps. To extract the longest horizontal edges from the potential horizontal edge image these edge discontinuities have to be filled up. A method to fill the edge discontinuities i.e., breaks or gaps is introduced in this work. According to the edge linking procedure, the discontinuities (breaks or gaps) of four consecutive pixels are filled up automatically. Otherwise, the edge linking process will be performed.

In Figure 5 explain the edge linking process. The discontinuities (breaks or gaps) of the HEs are linked up if the consecutive HEs satisfied the subsequent situations. According to the procedure depicted in Figure 5, the end point of $LineEdge(i)$ i.e., $(Lx_r(i)\ , Ly_r(i))$ is connected to the beginning point of $LineEdge(j)$ i.e., $(Lx_l(\mathrm{j}),\ Ly_l(j))$ if the two edge lines are fulfill the criteria of $Lx_l(j) > Lx_r(i)$ and $Ly_r(i) - Ly_l(j) < T_{height}$ respectively. In the same way, the end point of $LineEdge(j)$ is connected to the beginning point of $LineEdge(i)$ i.e., $(Lx_l(\mathrm{i}), Ly_l(i))$ if the two edge lines are fulfill the criteria of $Lx_r(j) < Lx_l(i)$ and $Ly_l(i) - Ly_r(j) < T_{height}$ respectively. Where, Ly_r, Ly_l and Lx_r, Lx_l are y-coordinate and x-coordinate of the right and left end points of horizontal $LineEdge(i)$ and horizontal $LineEdge(j)$ respectively. T_{height} is defined as the statistical threshold value. The effect of the edge linking process is demonstrated in Figure 6(b). From Figure 6(b) it is seen that the horizontal PC edge-image keeps only longest concurrent horizontal edges. In this stage assume that, after edge-linking the edge-image contains the total *N* number of longest HEs.

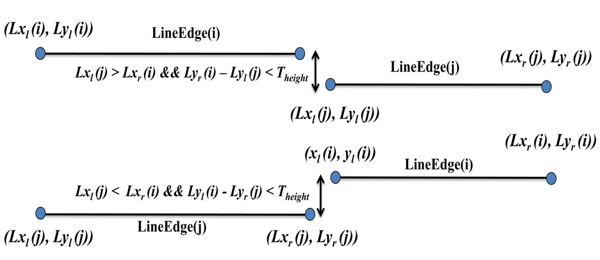


Figure 5 The edge linking procedure

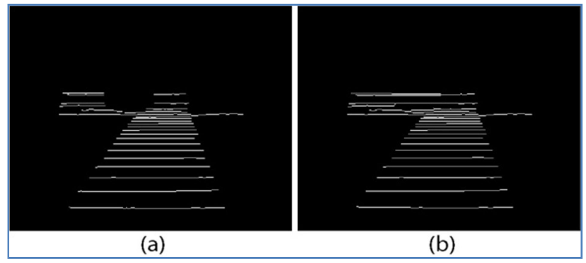


Figure 6 Processing example of edge-linking: a) an edge image before applying linking process, and b) extracting longest HEs through the edge linking.

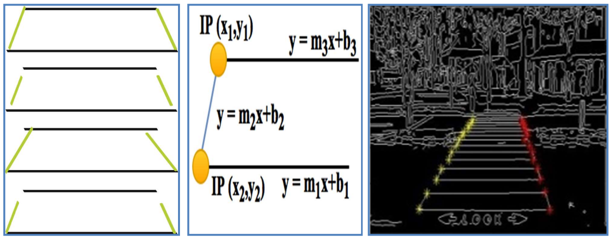
D. Finding two connected point

This section describes the process of finding two connected points (2CP) from the pedestrian crossing edge image which is the unique geometrical property of a pedestrian crossing. This 2CP is shaped at every pedestrian crossing stripe's HEs end points with pedestrian crossing stripe's height and width edge's intersection point. This feature is shown in Figure 1(a). This two connected point geometrical feature is utilized in this paper to validate the extracted pedestrian crossing HEs. For that, the Canny edge image is utilized. That is because of the edge image contains the pedestrian crossing horizontal and vertical edges, from where the N longest horizontal edges of the pedestrian crossing were extracted. According to the 2CP geometric feature, every pedestrian crossing stripe width edges intersect with pedestrian crossing HE's two end points.

For finding these 2CPs, pedestrian crossing stripe's width vertical edges (VEs) are finding at the ending points of the pedestrian crossing horizontal edges. However, the vertical edges do not exactly intersect at the horizontal edge's ending point due to the different illumination conditions as well as noise. The situations that would be arisen are depicted in Figure 7(a). To escape these types of situations, the two connected point (2CP) is estimated through a procedure. The procedure is depicted at Figure 7(b). In this, the two parallel horizontal edge-lines are $y = m_1x + b_1$ and $y = m_3x + b_3$ respectively, while the vertical edge-line is $y = m_2x + b_2$. The vertical and horizontal edge lines slop are represented by m_2, m_1, and m_3. The outcome of the two connected points with the process presented in Figure 7(b) is demonstrated in Figure 7(c).

According to the process of estimating the two connected point (2CP) at each ending points of the horizontal edges needed a constant time. Let, c is the constant time. For estimating 2CP at the two ends of each HE is required 2c time. Finding 2CPs for concurrent parallel *N* horizontal edges required 2*Nc* times. Hence, the time complexity claims for estimating 2CPs from *N* horizontal edges is about $O(N)$.

After estimating the 2CPs, the consecutive parallel pedestrian crossing stripes are evaluated. Where, N parallel horizontal pedestrian crossing edges were determined as candidate edge in the last section. The extracted consecutive horizontal parallel edges are though as pedestrian crossing stripe edge segment if the pedestrian crossing stripe with 2CP is 75% with respect to the N horizontal-edges. These N horizontal-edges that are validated by the 2CP are used in the next section to ensure and detect the pedestrian crossing region.



(a) (b) (c)

Figure 7 Processing experiment of 2CP: a) different situations of pedestrian crossing stripe's width edge alignment, b) procedure of estimating 2CP, and c) 2CPs in the edge image.

E. Arranging horizontal edges in concurrent shorted order and detect ROI

This section justifies the horizontal stripe edges that are validated with the 2CP. This justification is performed by arranging HEs in increasing sorted parallel order, which is the other key geometrical feature of the pedestrian crossing. To accomplish this, a comparison of the two contemporaneous successive parallel edge end points x coordinate values from the beginning of the pedestrian crossing stripe edges is used. As a result, , $edge\ (i)$ and $edge\ (j)$ are in the sequential form if they meet the following requirements i.e., $EL_x(j) > EL_x(i)$ and $ER_x(j) < ER_x(i)$. Here, $EL_x(i)$, $EL_x(j)$ and $ER_x(i)$, $ER_x(j)$are the coordinate points of the left and right sites respectively. In Figure 8(a), the results of sorting horizontal edges are presented.

The HEs that meet the distinct and natural features of a pedestrian crossing, i.e., two connected point and arranging HEs in sorted concurrent parallel order are considered as a pedestrian crossing. This region that holds these horizontal parallel edges are a thread as a PC region of interest (ROI). The ROI in pedestrian crossing input original image is presented in Figure 8(b). The ROI is presented in Figure 8(c). This ROI is verified through the linear SVM [23]. For that, the features are estimated from the detected pedestrian crossing ROI with rotational invariant uniform Local Binary Pattern (LBP)**.**

Figure 8 Examples of pedestrian crossing region detection: a) candidate increasing HE segment of a pedestrian crossing, b) detected ROI in the original PC image, and c) detected pedestrian crossing region.

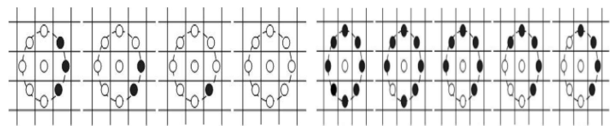
F. Extracting features and classification

The appropriate feature of prospective region's extraction is the critical issue in the recognition phase. For feature extraction, the candidate pedestrian crossing ROI is resized to 128 by 128 pixels. From the resized pedestrian crossing ROI the feature of uniform local binary pattern (LBP) are extracted with moment invariant [22] [24].

The consistent patterns are made possible by the rotational invariant uniform LBP for two transitions between 0-1 bit is 59 for eight-bit pattern and one radius neighborhood. These patterns are achieved by increasing number of one's from the seven set of codes. The patterns are 00000001, 00000011, 00000111, …, 011111111.

From the individual pattern eight set of pattern can be achieved, such as, from 00000001, eight combinations can be achieved. As a result the total output labels will be 8x7=56. Three additional combinations are for 11111111, 00000000, and one for all non-uniform patterns. So, the final desired labels will be 8x7+3 = 59, i.e., P x (P – 1) + 3.

For producing improved recognition results, the uniform LBP is utilized in this research work. The uniform LBP of an object is shown in Figure 9, where, the white and black circle is represented by 1 and 0 bit value respectively. The local primitives detected by the uniform LBP model include spots, flat areas, edges, lines, and corners. Where, edge, corners, and line are major uniform patterns. Finally, SVM [23] is used to classify the features.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 9 Uniform patterns of LBP eight bit pattern and one radius neighborhood: a) corner, b) line end, c) spot/flat, d) spot/flat, e) spot, f) spot, g) line end, h) corner, and i) edge.

IV. Experimental Results and Discussions

This section demonstrates the experimental result of the proposed pedestrian crossing (PC) detection and recognition framework. The framework is tested on a desktop with an Intel i3-E321220 3.10GHz CPU and 4GB RAM through MATLAB environment.

A. Pedestrian crossing dataset

The entire pedestrian crossing (PC) dataset images used in the experiment are considered 480x320 pixel sizes. This dataset's experimental pedestrian crossing samples are captured from outdoor locations with various environmental conditions, i.e., normal condition, noisy condition, as well as an uneven condition to consider different types of pedestrian crossing.

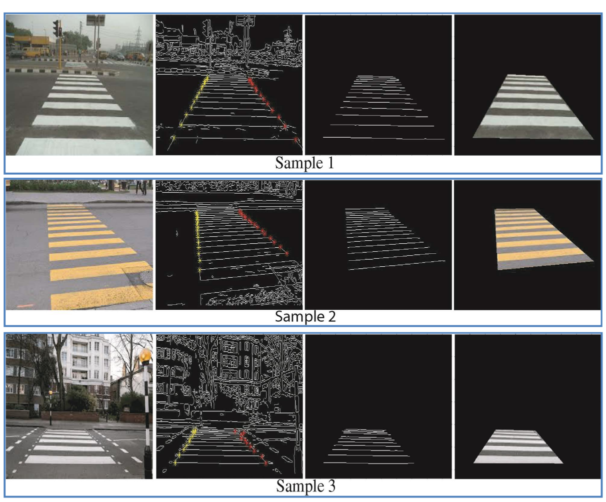
Some sample of the various types of positive pedestrian crossing images is shown in Figure 10. The used PC images are taken from the frontal side of the PC. The framework working well for the images whose angular orientation is not more than 30 degrees. Some negative pedestrian crossing images are shown in Figure 11. In the negative sample, the image dataset includes different types of PC similar images such as stair image and rail line image. These stair and rail line image have the features as like the pedestrian crossing image. The window, overdraw, bookshelves, and bipolar texture are looking similar to the pedestrian crossing. However, the pedestrian crossing has different key features by which the proposed method can detect the PC candidate region and the SVM classifier can differentiate PC from others. The ROI of the pedestrian crossing (PC) is resized to 128x128. In this paper, the main dataset contains 302 images that are divided into two datasets i.e., training and test dataset. In the training dataset, 185 pedestrian crossing images are considered, where, 117 images are considered for the test dataset.

Figure 10 Different types of pedestrian crossing images: a) with normal illumination, b) with the noisy background, c) with uneven illumination, and d) at night time with uneven illumination

Figure 11 Different types of negative pedestrian crossing images

B. Experimental results

The proposed framework's detection and recognition accuracy are determined through the different environmental experimental results. The processing example of the pedestrian crossing region of interest (ROI) detection of different environmental conditions is shown in Figure 12. In this figure, the input pedestrian crossing sample images' geometrical feature of two connected points (2CP), sorted HE segment, and pedestrian crossing ROI region are depicted in Figure 12(a), Figure 12(b), Figure 12(c), and Figure 12(d) respectively.



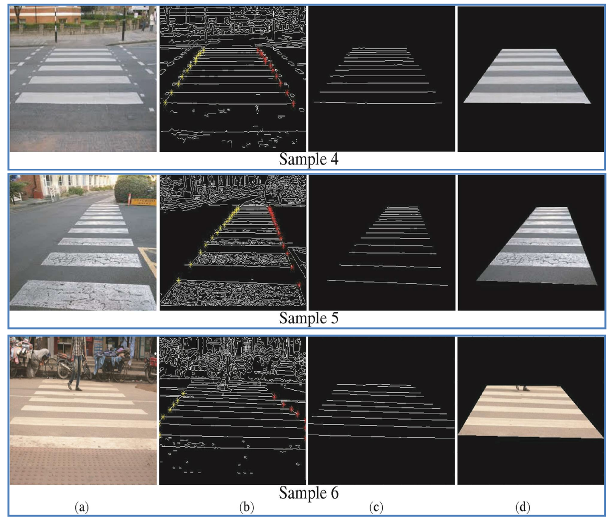


Figure 12 Processing example of pedestrian crossing region detection: a) input pedestrian crossing image, b) 2CP in edge image, c) arranging HEs in increasing sorted order, and d) detecting pedestrian crossing region

Sample image 1 and image 3 in Figure 12 are captured from the normal environment and noisy background. Where sample image 2 is captured from the uneven illumination condition with color strips. Here, Sample image 1 and Sample image 3 are taken from various heights and Sample image 2 is taken closely. Sample image 4 and 5 are captured from low illumination conditions, where samples white strips are not smooth and resolute as like another sample image white strips. Sample image 6 is captured from low illumination and noisy environment where human waking over the pedestrian crossing strip lines.

All the processing pedestrian crossing samples validate by the 2CP and justify by arranging horizontal edges in sorted order. The pedestrian crossing detection is challenging in the situation where the objects are on the pedestrian crossing strip lines as like Sample image 6 in Figure 12. That is because of horizontal strip edge lines are broken in the place of object location. However, the broken horizontal edge lines are linked in the phase of edge linking.

The number of 2CP, detection error rate with respect to false positive (FP) and false negative (FN), and respective Sample processing computation time are shown in Table 1. False positive (FP) indicates that the proposed framework detect PC candidate region, where the detected region is not from a PC. The false negative (FN) indicates that the proposed framework should detect the PC region correctly when the framework has not been.

Pedestrian crossing Sample	Total HEs	Total 2CP	2CP (%)	Detection rate		Executio n time (s)
				FP	FN	
Sample 1	14	13	92.86	0.00	1.75	0.070
Sample 2	19	16	84.21	0.00	5.15	0.069
Sample 3	12	10	83.33	0.25	0.50	0.068
Sample 4	12	11	91.67	0.00	0.00	0.070
Sample 5	16	13	81.25	0.00	3.25	0.069
Sample 6	13	10	76.92	0.00	9.25	0.072

* FP=False Positive FN=False Negative

Table 1. 2CP, detection rate and processing execution time of pedestrian crossing sample images.

In Figure13 shows the processing example of the pedestrian crossing at the rainy and under strip light's environment. From these experiments, it is revealed that the proposed framework detects the pedestrian crossing region effectively if the framework extracts the pedestrian crossing's strip line edges efficiently. The number of 2CP, stair region detection rate and respective processed image computation time are shown in Table 2.

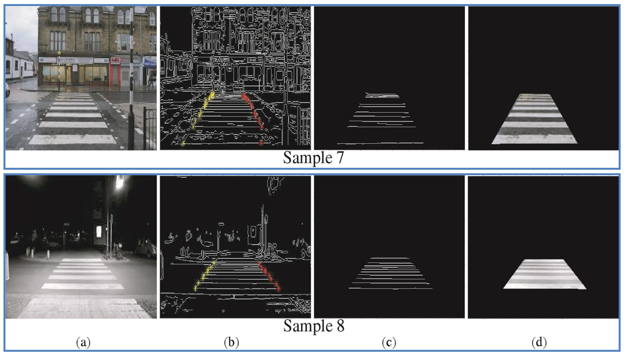


Figure 13 Processing example of detecting pedestrian crossing region in rainy and strip light environment: a) input image, b) two connected point, c) arranging increasing HEs in sorted order, and d) detecting pedestrian crossing region.

PC sample	**Total HEs**	**Total 2CP**	**2CP (%)**	**Detection rate**		**Execut-ion time (s)**
				FP	**FN**	
Sample 7	14	13	92.86	0	2.50	0.070
Sample 8	19	17	89.47	0	7.50	0.071

Table 2. 2CP, detection rate and processing execution time of pedestrian crossing sample images.

If any pedestrian crossing images are captured from the environments where different objects are located on the different portion of the pedestrian crossing images, for which pedestrian crossing strips are not shown properly as shown in Figure14. In that situation, it would be difficult to extract the pedestrian crossing's strips' edges properly as the proposed framework detect the pedestrian crossing region from an input image through the edge detection method, not from the input video frames.

The proposed system is able to detect the pedestrian crossing (PC) region from angular pedestrian crossing images. However, in that case, the angular orientation of the PC images should not be more than 30^0. The processing examples of angular PC region detection are shown in Figure15.

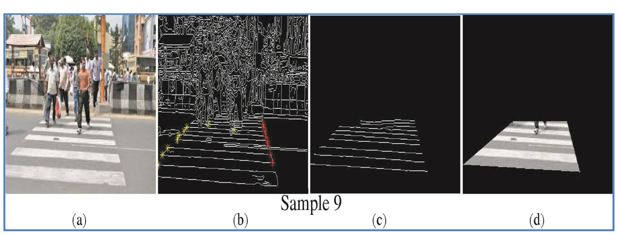


Figure 14 Processing example of detecting pedestrian crossing (PC) region where multiple humans are on the PC: a) input image, b) two connected point, c) arranging HEs in increasing sorted order, and d) detecting PC region

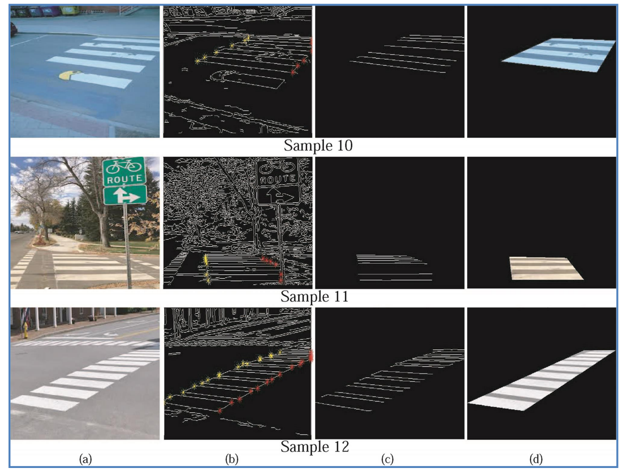


Figure 15 Processing example of detecting differently oriented pedestrian crossing (PC) region: a) input image, b) two connected point, c) arranging HEs in increasing sorted order, and d) detecting PC region.

From Figure15, it is seen that the proposed system efficiently detect the different oriented PC region from the angular PC images, where, Sample 10 and Sample 12 are captured in a right-oriented form and Sample 11 is captured in the left-oriented form with a road sign in front of it. In these samples, Sample 10's stripes are not smooth and some of the stripes are broken. For that, the framework is not able to satisfy the feature of arranging the horizontal edges in parallel sorted order. So that, the framework is not detected the first stripe of Sample 10. The remaining region of Sample 10 is detected efficiently as this region satisfies the key features of the proposed method. From the Sample 11 and Sample 12, the PC regions are detected efficiently as these samples satisfy the geometrical features of the proposed framework, although a road sign is placed in front of Sample 11. The number of 2CP, detection error rate, and respective processed image computation time are shown in Table 3.

PC sample	**Total HEs**	**Total 2CP**	**2CP (%)**	**Detection rate**		**Exec-ution time (s)**
				FP	**FN**	
Sample 10	10	8	80.00	0.00	17.5	0.053
Sample 11	10	9	90.00	0.00	9.50	0.067
Sample 12	18	17	94.44	0.50	2.50	0.072

Table 3. 2CP, detection rate and processing execution time of pedestrian crossing sample images.

If any PC's image is captured with more than 30^0 angular orientations, in that situations the proposed system is not performed better. That is because of in those situations the system is not able to extract the concurrent parallel edges efficiently. The reason behind that, the viewpoint of the PC's stripe edges will be changed and the structures of the vertical and horizontal edges do not remain the same as described in the proposed method. Some processing example to justify the problem arises due to the higher oriented PC image to detect PC region is shown in Figure16. From the samples shown in Figure16 it is revealed that, in the higher right or left orientation of PC images, the PC's stripes horizontal edges will be shifted to the vertical axis and stripe's height is shifted to the horizontal axis. As a result, the PC's strip horizontal edges that are shifted to the vertical axis will be eliminated according to the vertical edge elimination procedure described in 3.3 sections.

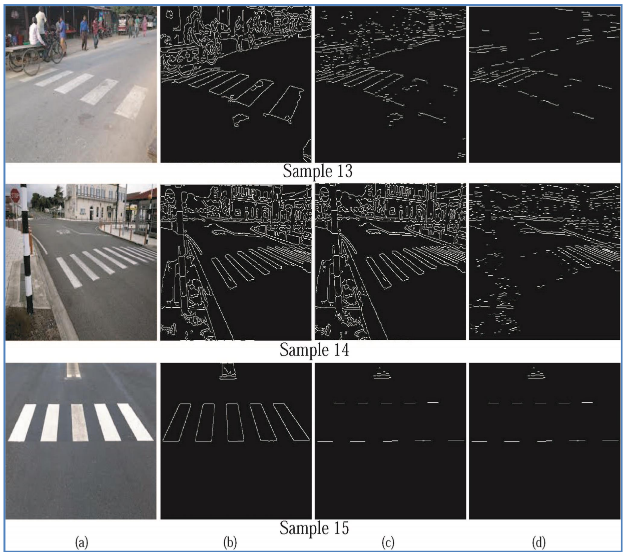


Figure 16 Processing example to justify the PC orientation problem: a) input image, b) Canny edge image, c) eliminating vertical edges, and d) after removing small non-interest edges.

The accuracy of the pedestrian crossing (PC) detection is estimated from various types of PC images with recall, precision, and F1-Score metrics. These metrics are measured based on the true positive (TP), false negative (FN) and false positive (FP) values. The TP indicates that the correct PC candidate region is detected by the proposed system. FP indicates that the PC candidate region is detected by the proposed framework, where the detected region is not from a PC. The false negative indicates that the proposed framework should detect the PC region correctly when the framework has not been.

The recall indicates the proposed framework's completeness. It refers the percentage of the right PC area discovered by the proposed framework. While precision indicates to the proposed framework's correctness. It represents the percentage of the PC area that is successfully identified by the proposed framework. Equations (4) and (5) define the recall and precision, respectively. The harmonic mean of recall and precision is used to get the F1-score. Equation (6) determines the F1-score.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FP}} \tag{4}$$

$$\text{Pecision} = \frac{\text{TP}}{\text{TP+}} \tag{5}$$

$$\text{F1} - \text{Score} = 2.\ \frac{\text{Recall . Precesion}}{\text{Recall + Precision}} \tag{6}$$

The detection accuracy of PC's region of interest (ROI) at the various environmental situations is presented in Table 4 with considering the false positive (FP) as well as false negative (FN) results. The pedestrian crossing ROI recognition accuracy through linear SVM at different environmental conditions is presented in Table 5. From Table 5, it is revealed that the system's PC region recognition accuracy is better at the normal illumination conditions, where, the system shows low accuracy at the noisy background.

Conditions	Total Image	Detected error (%)		Recall (%)
		FN	FP	
Normal illumination	51	0.00	0.00	100
Uneven Illumination	39	3.56	2.15	96.44
Noisy background	27	4.75	1.50	95.25
Avg.	**117**	**2.77**	**1.22**	**97.23**

Table 4(a). The detection result of pedestrian-crossing at various conditions.

Conditions	Precision (%)	F1-Score (%)	Detection accuracy (%)
Normal illumination	100	100	100.00
Uneven Illumination	97.85	97.14	97.15
Noisy background	98.50	96.85	96.88
Avg.	**98.78**	**98.00**	**98.01**

Table 4(b). The detection result of pedestrian-crossing at various conditions.

Pedestrian crossing type	Total image	Correctly detected	Recognition accuracy (%)
Normal-illumination	51	51	100
Uneven-Illumination	39	38	97.44
Noisy-background	27	26	96.30
	117	**115**	**98.29**

Table 5. Recognition accuracy of pedestrian crossing region through SVM at different environmental conditions.

The proposed framework of pedestrian crossing detection and recognition is compared with [7], [9], [10], and [14] with respect to our own dataset. The result is shown in Table 6. The curve of the receiver operating characteristics (ROC) of the proposed PC detection method with compare to the [7], [9], [10], and [14] methods is shown in Figure17.

Method	Detection accuracy
The proposed method	**98.01%**
[2]	97.95%
[7]	97.02%
[9]	78.90%
[10]	92.47%
[14]	96.20%

Table 6. Comparison of the proposed method and other reported methods for pedestrian crossing detection accuracy.

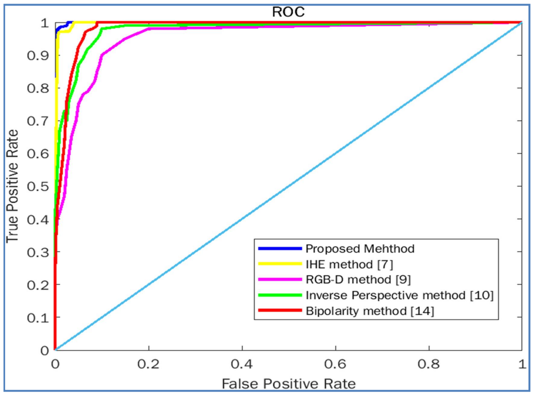


Figure 17 The ROC curve of the proposed method with compares with IHE method [7], RGB-D method [9], Inverse Perspective method [10] and Bipolarity method [14]

C. Discussion

According to the experimental findings, the proposed pedestrian crossing detection framework recognized the PC candidate zone from the diverse environments and illumination circumstances through the proposed key geometrical features.

The proposed framework performs better to detect the PC region in the case of computation and feature extraction with compare to the present state-of-the-art. The problems arise in the different state of the art that are compared with the proposed framework are listed here. Such as, in [7], the pedestrian crossing area is extracted through the feature of increasing horizontal edges (IHE) in sorted order. For this purpose, dynamic programming is used. This issue takes $O(n \log n)$ time to solve. In [9], the pedestrian crossing region is recognized by the depth feature information. However, the depth features cannot be extracted efficiently from all environments especially outdoor environments where sunshine is too bright due to the limitation of the Kinect sensor. In [14], the authors utilized the bipolarity theory idea. However, this idea is not performing well in noisy and dark white-stripes of the pedestrian crossing. In [10] introduces a method to reduce the problem occur in the [14] for bipolarity. However, for a template matching, this method used Inverse Perspective Mapping, which is time consuming and is not performed well in all the environmental conditions.

By the proposed method presented in this paper, the essential feature of 2CP detects the pedestrian crossing region. The 2CPs are estimated from the pedestrian crossing edge image that takes $O(n)$ time, which is linear. Here, n is the total horizontal parallel strip edge line at the pedestrian crossing. Moreover, the proposed system can detect and recognize the PC candidate zone in a variety of lighting and noise environments through filtering the PC input image with Gabor filters. The framework also detect and recognize the PC zones in the situation of objects are on the pedestrian crossing strip lines for which horizontal stripe edge lines are broken in the place of object location. However, the broken horizontal edge lines are linked in the phase of edge linking. This condition is also true for the broken stripe lines. Since the framework eliminated the non-interest edge segments through the non-interest edge elimination procedure before applying the key geometrical features. Hence the computation of this proposed framework is better as the framework utilized the key geometrical features in the longest horizontal parallel edge segment only. Finally, the framework efficiently recognized the PC candidate region through SVM, where, the rotational invariant uniform Local Binary Pattern is used to extract features.

D. Conclusions

This study proposes a natural geometrical feature-based pedestrian crossing detection and identification framework for automatically recognizing the pedestrian crossing region from a pedestrian crossing image. The proposed framework is vital for the visually impaired as well as the autonomous navigation system. To detect the pedestrian crossings region from the pedestrian crossing image some unique and natural geometrical features are used in this work. A group of pedestrian crossing images was tested successfully in this work to measure the proposed framework's efficiency and efficacy. The framework reveals the acceptable accuracy and runtime of 98.01% and 0.070 (s) respectively. Where, the recognition rate is revealed 97.29%. The pedestrian crossing detection framework detects and recognizes the traditional pedestrian crossing. This framework is not shown satisfactory performance for the unnatural shapes pedestrian crossing such as wheel pedestrian crossing. In the future, this framework will be improved for unnatural shapes pedestrian crossing and the pedestrian crossing region is detected by using deep-learning methods.

References

[1] Y. Fan, Z. Sun, and G. Zhao, "A coarse-to-fine framework for multiple pedestrian crossing detection," Sensors, vol. 20, no. 15, p.4144, January 2020.

[2] M. I. Meem, P. K. Dhar, M. Khaliluzzaman, and T. Shimamura, " Zebra-Crossing Detection and Recognition Based on Flood Fill Operation and Uniform Local Binary Pattern," Proc. 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), IEEE Press, pp. 1-6, February 2019.

[3] X. Liu, Y. Zhang, and Q. Li, "Automatic Pedestrian Crossing Detection and Impairment Based on Mobile Mapping System," ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, vol. 4, September 2017.

[4] S. Hacohen, N. Shvalb, and S. Shoval, "Dynamic model for pedestrian crossing in congested traffic based on probabilistic navigation function," Transportation research part C: emerging technologies, vol. 86, pp.78-96, January 2018.

[5] M.C. Ghilardi, J. J. Junior, and I. Manssour, "Crosswalk localization from low resolution satellite images to assist visually impaired people," IEEE computer graphics and applications, vol. 38, no. 1, pp. 30-46, May 2018.

[6] Z. S. Chen and D. F. Zhang, "An Effective Detection Algorithm of Zebra-Crossing," Proc. 3rd International Conference on Electrical and Information Technologies for Rail Transportation (EITRT 2017), pp. 809-816, October 2017.

[7] M. Khaliluzzaman and K. Deb, "Zebra-crossing Detection based on Geometric Feature and Vertical Vanishing Point," Proc. 3rd International Conference on

Electrical Engineering and Information & Communication Technology (iCEEiCT 2016), IEEE Press, pp. 1-6, September 2016.

[8] W. Czajewski, P. Mrówka and P. Olszewski, “Video Processing for Detection and Tracking of Pedestrians and Vehicles at Zebra-crossings,” Proc. International Conference on Transport Systems Telematics, pp. 34-44, April 2015.

[9] S. Wang, H. Pan, C. Zhang, C. 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, February 2014.

[10] S. Sichelschmidt, A. Haselhoff, A. Kummert, M. Roehder, B. Elias, and K. Berns, "Pedestrian crossing detecting as a part of an urban pedestrian safety system," Proc. IEEE Intelligent Vehicles Symposium, IEEE Press, pp. 840-844, June 2021.

[11] D. Koester, B. Lunt, and R. Stiefelhagen, “Zebra crossing detection from aerial imagery across countries,” Proc. International Conference on Computers Helping People with Special Needs, pp. 27-34, July 2016.

[12] S. Bonnin, T. H. Weisswange, F. Kummert and J. Schmuedderich, "Pedestrian crossing prediction using multiple context-based models," Proc. *IEEE Conference on Intelligent Transportation Systems (ITSC 2014)*, IEEE Press, pp. 378-385, October 2014.

[13] H. Li, M. Y. Feng, and X. Wang, “A Zebra-Crossing Detection Algorithm for Intelligent Vehicles,” Applied Mechanics and Materials, vol. 236, pp. 390-395, 2012.

[14] C. Yuzhen, C. Lushi, and J. Shuo, “An Image Based Detection of Pedestrian Crossing,” Proc. 2nd International Congress on Image and Signal Processing (CISP 09), IEEE Press, pp. 1-5, October 2009.

[15] B. Riveiro, H. González-Jorge, J. Martínez-Sánchez, L. Díaz-Vilariño, and P. Arias, “Automatic detection of zebra crossings from mobile LiDAR data,” Optics & Laser Technology, vol. 70, pp. 63-70, July 2015.

[16] V. Ivanchenko, J. Coughlan, and H. Shen, "Detecting and locating crosswalks using a camera phone," Proc. Computer Vision and Pattern Recognition Workshops (CVPRW 08), IEEE Press, pp. 1-8, June 2008.

[17] J. Coughlan and H. Shen, "A fast algorithm for finding crosswalks using Fig.-ground segmentation," Proc. 2nd Workshop on Applications of Computer Vision, in conjunction with ECCV, Vol. 5, May 2006.

[18] W. Czajewski, P. Dąbkowski, P. Olszewski, “Innovate solutions for improving safety at pedestrian crossings,” Arch. Transp. Syst. Telemat. vol. 6, no. 2, pp.16–22, 2013.

[19] S. Se, "Zebra-crossing detection for the partially sighted," Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Press, vol. 2, pp. 211-217, June 2000.

[20] C. A. Basca, B. Remus, and B. Lucian, “Texture Segmentation.Gabor Filter Bank Optimization Using Genetic Algoritnms,” Proc. International Conference on Computer Tool, IEEE Press, pp. 331-335, September 2007.

[21] M. Khaliluzzaman, and K. Deb, “Support vector machine for overcoming the problem of vanishing point during stairways detection,” In International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE 2016), IEEE Press, pp. 1-5, December 2016 .

[22] M. Pietikäinen , A. Hadid, G. Zhao, and T. Ahonen, “Computer vision using local binary patterns,” Springer Science & Business Media, Vol. 40, June 2011.

[23] J. C. Burges, "A tutorial on support vector machines for patten recognition,” Data Mining and Knowledge Discovery, Vol. 2, pp.121-167, 1998.

[24] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Transactions on pattern analysis and machine intelligence, vol. 24, no. 7, pp.971-987, August 2002.

[25] L. Boudet and S. Midenet, “Pedestrian crossing detection based on evidential fusion of video-sensors,” Transportation research part C: emerging technologies, vol. 17, no. 5, pp.484-497, October 2009.

[26] J. Canny, “A Computational Approach to Edge Detection,” IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 679–698, November 1986.

[27] G. F. McLean and D. Kotturi, “Vanishing point detection by line clustering,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no.11, pp. 1090–1095, November 1995.

[28] T. S. Lee, “Image representation using 2D Gabor wavelets”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 10, pp. 959-971, October 1996.

[29] A. Huertas and G. Medioni, “Detection of intensity changes with sub pixel accuracy using Laplacian-Gaussian masks,” IEEE Transactions On Pattern Analysis and Machine Intelligence, vol. 8, pp. 651–664, 1986.

[30] L. Chen, “Laplacian embedded regression for scalable manifold regularization,” IEEE transactions on neural networks and learning systems, vol. 23, no. 6, pp. 902–915, June 2012.

[31] S. Mejia, J. E. Lugo, R. Doti, and J. Faubert, “Pedestrian modeling using the least action principle with sequences obtained from thermal cameras in a real life scenario,” International Journal of Computer Information Systems and Industrial Management Applications, Vol. 9, 2017.

Author Biographies

Md. Khaliluzzaman received his BSc in CSE in 2007 from Khulna University of Engineering and Technology (KUET), Bangladesh. He received his MSc in CSE in 2017 from Chittagong University of Engineering and Technology (CUET), Bangladesh. Currently, he is working as an Assistant Professor in the Department of Computer Science and Engineering, International Islamic University Chittagong, Chittagong, Bangladesh. His research interests include intelligence system, computer vision, human computer interaction, pattern recognition and Gait analysis.