IIUC Studies, Vol.-20, Issue-1, June 2023
DOI: https://doi.org/10.3329/iiucs.v20i1.69050

A geometrical feature-based framework for pedestrian crossing recognition

Md. Khaliluzzaman
Department of Computer Science & Engineering
International Islamic University Chittagong (IIUC), Bangladesh

Abstract
In this work, a framework for the recognition of pedestrian crossing (PC) regions based on geometrical features is proposed. A distinctive feature of a pedestrian crossing (PC) is that each end point of the horizontal strip edges at pedestrian crossings intersects with a vertical stripe width edge, which comprises two connected points (2CP). Another unique feature of PC is that the PC stripe's edges are formed in ascending parallel order. These two features are utilized to identify the PC candidate region in the PC image. Where the 2CP and parallel edge segment in sorted order is used to validate and justify the PC region. Finally, classifier support vector machine (SVM) confirms the potential PC region. Here, the features of the candidate area are extracted using the uniform rotationally invariant Local Binary Pattern (LBP). The proposed framework is tested with our own dataset, and the results reveal significant improvement over previous work.

Keywords Pedestrian crossing, Two connected point, Rotational invariant, Uniform LBP, SVM

Paper type Research paper

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 images 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, 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

IIUC Studies
Vol.-20, Issue-1, June 2023
pp. 59-86

ISSN 2408-8544

recognition process of pedestrian crossing should be evaluated as fast as possible in different environmental conditions.

To satisfy the above criteria and increase pedestrian crossing safety, a pedestrian crossing region detection and recognition framework need to be developed. So, the developed framework may recover the existing methods problems and work 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 litte 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 edge 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 recognized 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.

- A unique Geometrical feature based pedestrian crossing recognition framework is proposed.
- Gabor filters are used to eliminate the nose and shadows effect from the 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 explanation of the proposed framework is demonstrated in Section III. Section IV presents the outcomes of the proposed framework. Section V serves as the paper's conclusion.

Related works

In a few decades, researchers and scientists have developed many frameworks to elevate the perfection of 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 pedestrian crossing framework is introduced in Fan, Sun, and Zhao (2020). Here, the pedestrian is detected from probe vehicle video. To detect the exact location of the pedestrian the vehicle video is combined with the GPS traces. Some techniques of image processing used for detecting the pedestrian crossing are introduced in Meem, Dhar, Khaliluzzaman, and Shimamura (2019). Hough transformation, Flood-fill operation, and adaptive-histogram equalization are the techniques employed in this paper. These techniques are used to extract the PC region of interest (ROI). Support vector machine (SVM) is used to identify the measured ROI. A vehicle-based Mobile Mapping System is introduced for automatic pedestrian crossing detection with monocular vision in Liu, Zhang, and Li (2017). Here, contour information is analyzed and crossing features are trained to detect and recognize the pedestrian crossing.

In Hacohen, Shvalb, and Shoval (2018), the authors introduce a method to identify the conflict of the vehicles and pedestrians at the pedestrian crossing area, where drivers and pedestrians do not maintain 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 Ghilardi, Jacques, and Manssour (2016), 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 Chen and Zhang (2018), authors introduce a zebra-crossing detection system based on an on-board monocular camera. Here, the zebra-crossing region is extracted 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 (Khaliluzzaman & Deb, 2016b). The main feature utilized in this paper is sorted order by the increasing horizontal edges. Finally, the potential ROI is classified through the vertical vanishing point (VP) (McLean & Kotturi 1995). However, the VP is not suitable in all the environments to verify the pedestrian crossing region. That is because the stair region also shows this same property as pedestrian crossing (Khaliluzzaman & Deb, 2016a).

In Czajewski, Mrówka, and Olszewski (2015), 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 that are passing through the PC and reduce the collision between the pedestrian and vehicle. A depth information based pedestrian crosswalk method is proposed in Wang, Pan, Zhang, and Tian (2014), where depth information is detected from the RGB-D image. Where, the crosswalk region is identified by extracting the parallel crosswalk strip's lines and SVM classifier is used to recognize. The features used in this work are depth features. In Sichelschmidt, Haselhoff, Kummert, Roehder, Elias, and Berns (2010), 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 angles from the 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 Mejia, Lugo, Doti, and Faubert, (2017).

A data-driven approach is introduced in Koester, Lunt, and Stiefelhagen (2016) 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 Bonnin, Weisswange, Kummert, and Schmüdderich (2014) to estimate the crossing behaviors of a pedestrian in the inner city. Local threshold segmentation based zebra-crossing detection method is proposed in Li, Feng, and Wang (2012), 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 Riveiro, González-Jorge, Martínez-Sánchez, Díaz-Vilariño, and Arias (2015), 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 Boudet and Midenet (2009). 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 was proposed in Cao, Chen, and Jia (2009). Before that, smartphone-based crosswalks detection method was introduced in Ivanchenko, Coughlan, and Shen (2008). A coherent structure of the crosswalk geometrical feature is used in Coughlan and Shen (2006) to detect the crosswalk. Vanishing point based pedestrian detection method is introduced in Khaliluzzaman and Deb (2016b) and Se (2000). Automatic pedestrian detection with dynamic traffic signs is introduced in Czajewski, Dąbkowski, and Olszewski (2013). 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 proposed many pedestrian crossing (PC) detection methods based on the geometrical feature (Khaliluzzaman & Deb, 2016b), depth feature (Wang, Pan, Zhang, & Tian, 2014), the bipolarity theory (Sichelschmidt, Haselhoff, Kummert, Roehder, Elias, & Berns, 2010; Ivanchenko, Coughlan, & Shen 2008), coherent structure (Coughlan & Shen 2006), template matching (Ivanchenko, Coughlan, & Shen, 2008) and so on. However, these methods do not perform better for their limitations. Such as, the geometrical feature used in Khaliluzzaman and Deb (2016b) 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 for the depth feature do not perform well in 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 much time to identify the candidate ROI. Furthermore, the identified ROI is acknowledged by utilizing the different recognition methods such as vanishing point (Khaliluzzaman & Deb, 2016b; Se, 2000). 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 presented 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 SVM classifier recognizes the identified region of interest.

Proposed framework

To detect the pedestrian crossing (PC) region from input pedestrian crossing images, 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.

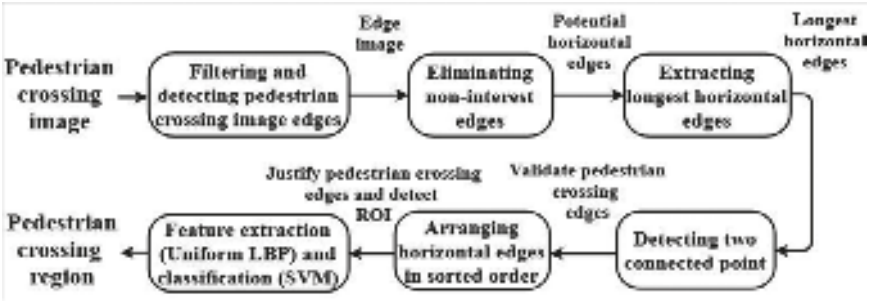


Figure 2
Proposed pedestrian crossing detection and recognition framework.

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) (Basca & Brad, 2007; Lee, 1996).

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

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

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

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 denote 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'). Figure 3(b) displays the Gabor filter's behavior to the PC image.

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 (Huertas & Medioni, 1986) based Canny edge operator (Canny, 1986; Lee, 1996) 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 (Chen, Tsang, & Xu 2012). The extracted Canny edge image from the preprocessed filtered image is shown in 3(c).

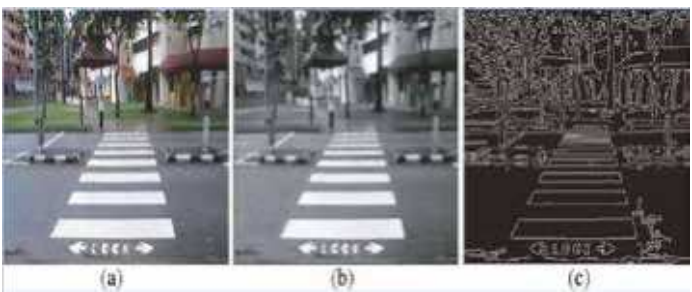


Figure 3
Examples of filtering and finding edges: a) input pedestrian-crossing image, b) result of Gabor filtered image, and c) result after applying the canny edge image.

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 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 edge length values standard deviation 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 there 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 HEs without non-potential edges are depicted in Figure 4(c).

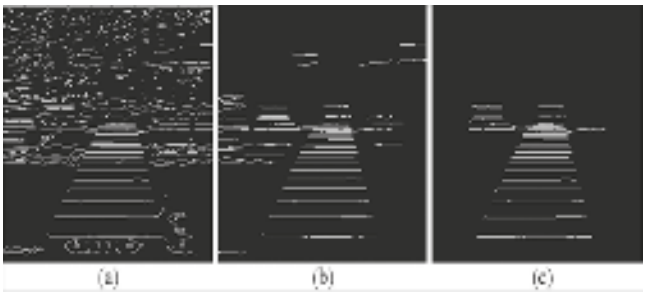


Figure 4

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

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., $(Lx_l(j), Ly_l(j))$ if the two edge lines 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., $(Lx_l(i), Ly_l(i))$ if the two edge lines 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 the longest concurrent horizontal edges. In this stage assume that,

after edge-linking the edge-image contains the total N number of longest HEs.

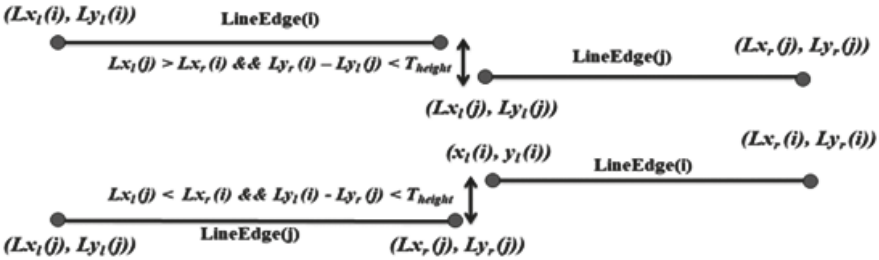


Figure 5
The edge linking procedure.

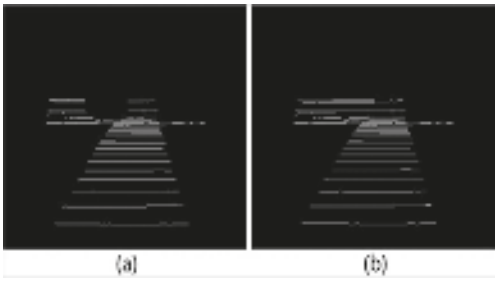


Figure 6
Processing example of edge-linking: a) an edge image before applying linking process, and b) extracting longest HEs through the edge linking.

Finding two connected point

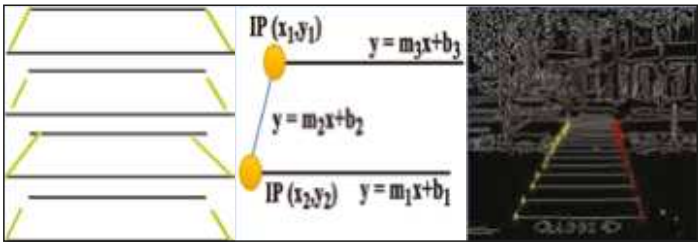
This section describes the process of finding two connected points (2CP) from the pedestrian crossing edge image which is the unique geometrical property of a pedestrian crossing. This 2CP is shaped at every pedestrian crossing stripe's HEs end points with pedestrian crossing stripe's height and width 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 the edge image contains the pedestrian crossing horizontal and vertical edges, from which 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 stripes' width vertical edges (VEs) are found 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 arise are depicted in Figure 7(a). To escape these types of situations, the two connected points (2CP) are 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 slope 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 points (2CP) at each ending point 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 2Nc 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 edges in the last section. The extracted consecutive horizontal parallel edges are thought as pedestrian crossing stripe edge segments 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.

Arranging horizontal edges in concurrent sorted 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 points and arranging HEs in sorted concurrent parallel order are considered as a pedestrian crossing. This region that holds these horizontal parallel edges is 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 (Burges, 1998). 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.

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. The feature of the uniform local binary pattern (LBP) is recovered using moment invariant techniques from the resized pedestrian crossing ROI (Pietikäinen, Hadid, Zhao, & Ahonen, 2011; Ojala, Pietikainen, & Maenpaa, 2002).

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 the

number of ones from the seven sets of codes. The patterns are 00000001, 00000011, 00000111, …, 011111111.

From the individual pattern eight sets of patterns 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 values respectively. The uniform LBP model can identify local primitives such as spots, flat surfaces, edges, lines, and corners. Where, the principal uniform patterns are edge, corners, and line. Finally, SVM (Burges, 1998) is used to classify the features.

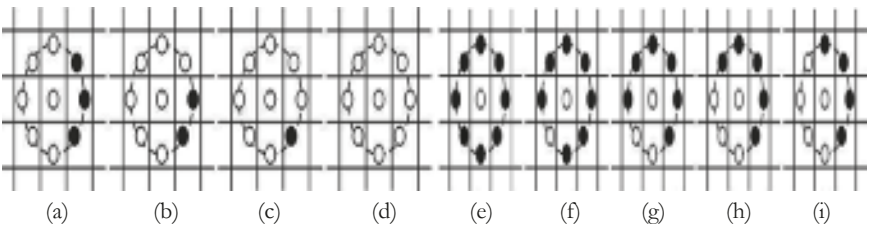


Figure 9

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.

Experimental results and discussions

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

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.

A 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 works 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 images have the features 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 framework can detect the PC candidate ROI and the SVM classifier can differentiate PCs 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.

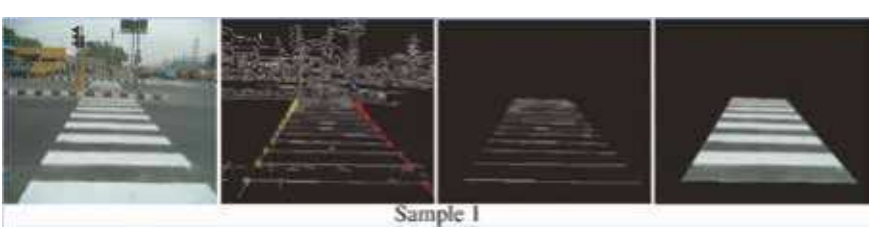
Figure 10

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.

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.



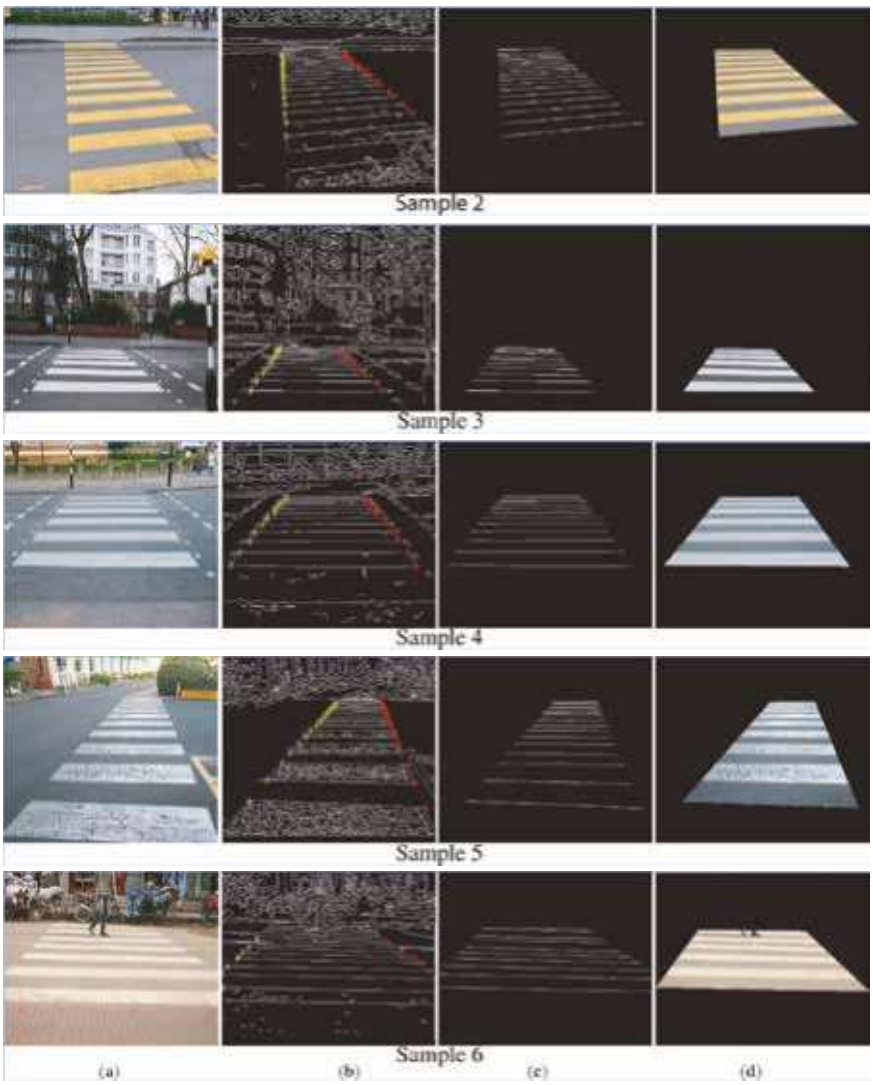


Figure 12

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 images 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 was taken in a noisy, low-light condition where humans were walking 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 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 detects the 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 through the framework could not do it.

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

Pedestrian crossing Sample	Total HEs	Total 2CP	2CP (%)	Detection rate		Execution 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

Figure 13 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.

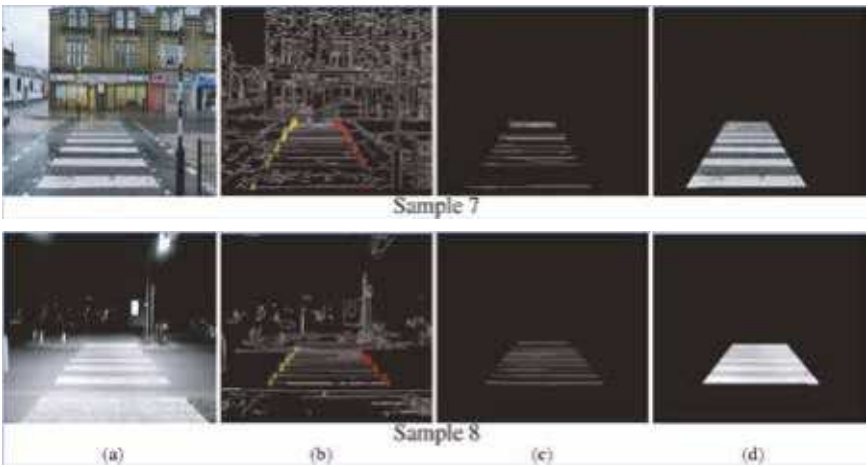


Figure 13

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

Table 2

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

PC sample	Total HEs	Total 2CP	2CP (%)	Detection rate		Execution 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

If any pedestrian crossing images are captured from the environments where different objects are located on the different portions 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 detects 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 300. The processing examples of angular PC region detection are shown in Figure15.

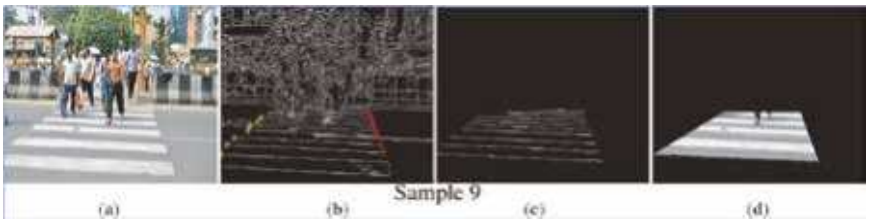


Figure 14

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

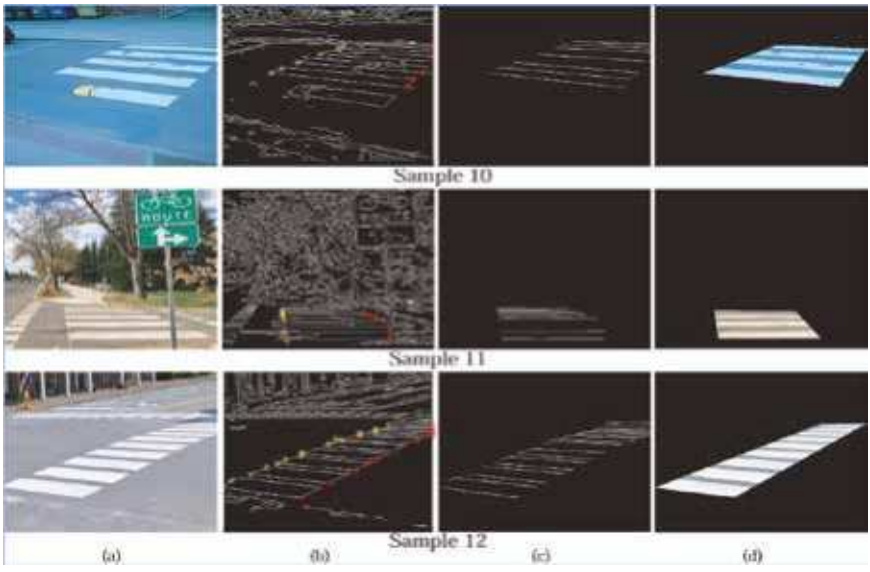


Figure 15

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

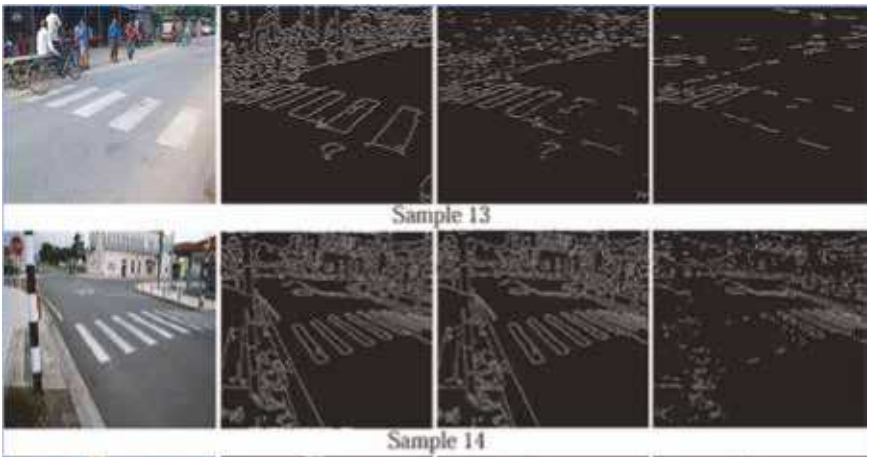
From Figure15, it is seen that the proposed system efficiently detects the different oriented PC region from the angular PC images, where, Sample 10 and Sample 12 are captured in a right-oriented form and Sample 11 is captured in the left-oriented form with a road sign in front of it. In these samples, 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, the framework does not detect 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.

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

PC sample	Total HEs	Total 2CP	2CP (%)	Detection rate		Execution 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

If any PC's image is captured with more than 30^0 angular orientations, in that situation the proposed system is not performed better. That is because 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 chang and the structures of the vertical and horizontal edges do not remain the same as described in the proposed method. Figure 16 shows some processing examples to justify the problem arise due to the higher oriented PC image to detect PC region. From the samples shown in Figure 16 it is revealed that, in the higher right or left orientation of PC images, the PC's stripes horizontal edges shift to the vertical axis and stripe's height shift 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.



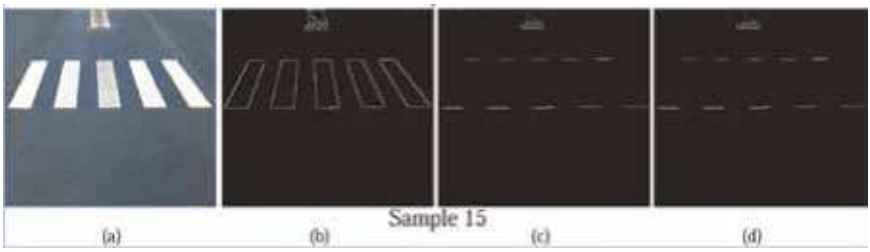


Figure 16

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 through the framework could not do it.

The recall indicates the proposed framework's completeness. It refers to the percentage of the right PC area discovered by the proposed framework. While precision indicates 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.

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

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

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

The detection accuracy of PC's 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.

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

Conditions	Total Image	Detected error (%)		Detection accuracy (%)
		FP	FN	
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(b)
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 5
Recognition accuracy of pedestrian crossing region through SVM at different environmental conditions

Pedestrian crossing type	Total images	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

The proposed framework of pedestrian crossing detection and recognition is compared with Khaliluzzaman and Deb (2016b), Wang, Pan, Zhang, and Tian (2014), Sichelschmidt, Haselhoff, Kummert, Roehder, Elias, and Berns (2010) and Cao, Chen, and Jia (2009) 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 Khaliluzzaman and Deb (2016b), Wang, Pan, Zhang, and Tian (2014), Sichelschmidt, Haselhoff, Kummert, Roehder, Elias, and Berns (2010) and Cao, Chen, and Jia (2009) methods is shown in Figure 17.

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

Method	Detection accuracy
The proposed method	98.01%
Meem, Dhar, Khaliluzzaman, and Shimamura (2019)	97.95%
Khaliluzzaman and Deb (2016b)	97.02%
Wang, Pan, Zhang, and Tian (2014)	78.90%
Sichelschmidt, Haselhoff, Kummert, Roehder, Elias, and Berns (2010)	92.47%
Cao, Chen, and Jia (2009)	96.20%

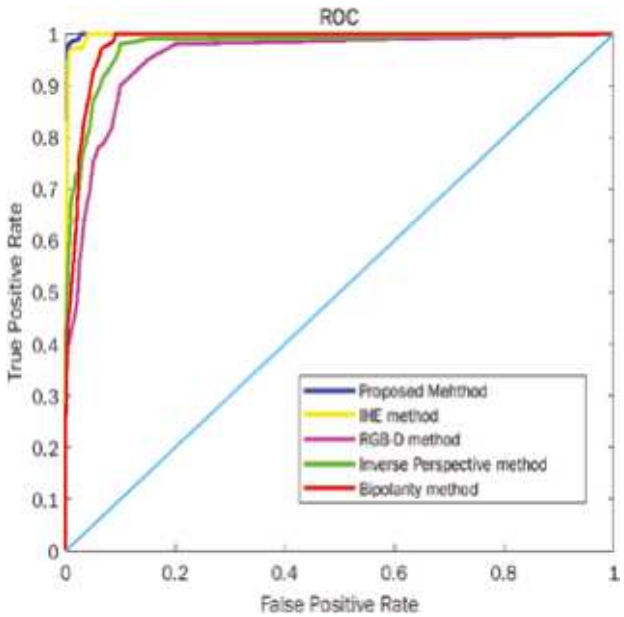


Figure 17

The ROC curve of the proposed method with compares with IHE method (Khaliluzzaman & Deb 2016b), RGB-D method (Wang, Pan, Zhang, & Tian 2014), Inverse Perspective method (Sichelschmidt, Haselhoff, Kummert, Roehder, Elias, & Berns 2010) and Bipolarity method (Cao, Chen, & Jia 2009).

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 compared to the present state-of-the-art. The problems that arise in the different state of the art that are compared with the proposed framework are listed here. Such as, in Khaliluzzaman and Deb (2016b), 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(nlog n) time to solve. In Wang, Pan, Zhang, and Tian (2014), 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 Cao, Chen, and Jia (2009), 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 Sichelschmidt, Haselhoff, Kummert, Roehder, Elias, and Berns (2010) introduces a method to reduce the problem occurring in the Cao, Chen, and Jia (2009) for bipolarity. However, for template matching, this method used Inverse Perspective Mapping, which is time consuming and does not perform 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 detects and recognizes the PC zones in the situation of objects 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 successfully identified the PC candidate region using SVM, where features were extracted from the rotationally invariant uniform Local Binary Pattern.

Conclusion

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 crossing 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 does not show 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 will be detected by using deep-learning methods.

References

Basca, C. A., & Brad, R. (2007, September 9-12). *Texture segmentation. Gabor filter bank optimization using genetic algorithms.* Paper presented at The International Conference on" Computer as a Tool", In EUROCON 2007- (pp. 331-335). IEEE.

Bonnin, S., Weisswange, T. H., Kummert, F., & Schmüdderich, J. (2014, October 8-11). *Pedestrian crossing prediction using multiple context-based models*. Paper presented at the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), China (pp. 378-385). IEEE.

Boudet , L., & Midenet, S. (2009). Pedestrian crossing detection based on evidential fusion of video-sensors. *Transportation research part C: emerging technologies, 17*(5), 484-497.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery, 2*(2), 121-167.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence,* (6), 679-698.

Cao, Y., Chen, L., & Jia, S. (2009, October 17-19). *An image based detection of pedestrian crossing.* Paper presented at the 2nd International Congress on Image and Signal Processing, China (pp. 1-5). IEEE.

Chen, L., Tsang, I. W., & Xu, D. (2012). Laplacian embedded regression for scalable manifold regularization. *IEEE transactions on neural networks and learning systems, 23*(6), 902-915.

Chen, Z. S., & Zhang, D. F. (2018). An Effective Detection Algorithm of Zebra-Crossing. In: Jia, L., Qin, Y., Suo, J., Feng, J., Diao, L., An, M. (eds) Proceedings of the 3rd International Conference on Electrical and Information Technologies for Rail Transportation (EITRT) 2017. EITRT 2017. Lecture Notes in Electrical Engineering, vol 482. Springer, Singapore. https://doi.org/10.1007/978-981-10-7986-3_81

Coughlan, J., & Shen, H. (2006). *A fast algorithm for finding crosswalks using figure-ground segmentation.* Paper presented at the 2nd Workshop on Applications of Computer Vision, in conjunction with ECCV (Vol. 5), Smith-Kettlewell Eye Research Institute.

Czajewski, W., Dąbkowski, P., & Olszewski, P. (2013). Innovative solutions for improving safety at pedestrian crossings. *Archives of Transport System Telematics, 6*(2), 16-22.

Czajewski, W., Mrówka, P., & Olszewski, P. (2015, April). *Video processing for detection and tracking of pedestrians and vehicles at zebra crossings.* Paper presented at the International Conference on Transport Systems Telematics (TST), Springer, Cham.

Fan, Y., Sun, Z., & Zhao, G. (2020). A coarse-to-fine framework for multiple pedestrian crossing detection. *Sensors, 20*(15), 4144.

Ghilardi, M. C., Jacques, J. C., & Manssour, I. (2016). Crosswalk localization from low resolution satellite images to assist visually impaired people. *IEEE computer graphics and applications, 38*(1), 30-46.

Hacohen, S., Shvalb, N., & Shoval, S. (2018). Dynamic model for pedestrian crossing in congested traffic based on probabilistic navigation function. *Transportation research part C: emerging technologies, 86,* 78-96.

Huertas, A., & Medioni, G. (1986). Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (5), 651-664.

Ivanchenko, V., Coughlan, J., & Shen, H. (2008, June 23-28). *Detecting and locating crosswalks using a camera phone.* Paper presented at the IEEE computer society conference on computer vision and pattern recognition workshops (pp. 1-8). IEEE.

Khaliluzzaman, M., & Deb, K. (2016a, December 8-10). *Support vector machine for overcoming the problem of vanishing point during stairways detection.* Paper presented at the 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE) (pp. 1-5). IEEE. doi: 10.1109/ICECTE.2016.7879637.

Khaliluzzaman, M., & Deb, K. (2016b, September). *Zebra-crossing detection based on geometric feature and vertical vanishing point.* Paper presented at the 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT) (pp. 1-6). IEEE. doi: 10.1109/CEEICT.2016.7873114.

Koester, D., Lunt, B., & Stiefelhagen, R. (2016). Zebra Crossing Detection from Aerial Imagery Across Countries. In: Miesenberger, K., Bühler, C., Penaz, P. (eds) Computers Helping People with Special Needs. ICCHP 2016. Lecture Notes in Computer Science, vol 9759. Springer, Cham. https://doi.org/10.1007/978-3-319-41267-2_5

Lee, T. S. (1996). Image representation using 2D Gabor wavelets. *IEEE Transactions on pattern analysis and machine intelligence, 18*(10), 959-971.

Li, H., Feng, M. Y., & Wang, X. (2012). A zebra-crossing detection algorithm for intelligent vehicles. In *Applied Mechanics and Materials 236,* 390-395.

Liu, X., Zhang, Y., & Li, Q. (2017). Automatic pedestrian crossing detection and impairment analysis based on mobile mapping system. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4*.

McLean, G. F., & Kotturi, D. (1995). Vanishing point detection by line clustering. *IEEE Transactions on pattern analysis and machine intelligence, 17*(11), 1090-1095.

Meem, M. I., Dhar, P. K., Khaliluzzaman, M., & Shimamura, T. (2019, February 7-9). *Zebra-Crossing Detection and Recognition Based on Flood Fill Operation and Uniform Local Binary Pattern.* Paper presented at the International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox'sBazar, Bangladesh, (pp. 1-6), IEEE.

Mejia, S., Lugo, J. E., Doti, R., & Faubert, J. (2017). 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 (IJCISIM), 9*, 145-152.

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence, 24*(7), 971-987.

Pietikäinen, M., Hadid, A., Zhao, G., & Ahonen, T. (2011). *Computer vision using local binary patterns* (Vol. 40). New York, USA: Springer Science & Business Media.

Riveiro, B., González-Jorge, H., Martínez-Sánchez, J., Díaz-Vilariño, L., & Arias, P. (2015). Automatic detection of zebra crossings from mobile LiDAR data. *Optics & Laser Technology, 70*, 63-70.

Se, S. (2000, June 15). *Zebra-crossing detection for the partially sighted.* Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition. (CVPR), South Carolina, USA, (Cat. No. PR00662) (Vol. 2, pp. 211-217). IEEE

Sichelschmidt, S., Haselhoff, A., Kummert, A., Roehder, M., Elias, B., & Berns, K. (2010, June 21-24). *Pedestrian crossing detecting as a part of an urban pedestrian safety system.* Paper presented at the 2010 IEEE Intelligent Vehicles Symposium, La Jolla, CA, USA, (pp. 840-844). IEEE.

Wang, S., Pan, H., Zhang, C., & Tian, Y. (2014). RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs. *Journal of Visual Communication and Image Representation, 25*(2), 263-272.

Corresponding author
Md. Khaliluzzaman can be contacted at: khalil@iiuc.ac.bd