Improved Candidate Generation for Pedestrian Detection using Background Modeling in Connected Vehicles

Ghaith Al-refai[1], Osamah A. Rawashdeh[2]
School of Electrical and Computer Engineering
Oakland University, Rochester, Michigan 48309

***Abstract*—Pedestrian detection is widely used in today's vehicle safety applications to avoid vehicle-pedestrian accidents. The current technology of pedestrian detection utilizes onboard sensors such as cameras, radars, and Lidars to detect pedestrians, then information is used in a safety feature like Automatic Emergency Braking (AEB). This paper proposes pedestrian detection system using vehicle connectivity, image processing and computer vision algorithms. In the proposed model, vehicles collect image frames using on-vehicle cameras, then frames are transferred to the Infrastructure database using Vehicle to Infrastructure communication (V2I). Image processing and machine learning algorithms are used to process the infrastructure images for pedestrian detection. Background modeling is used to extract the foreground regions in an image to identify regions of interest for candidate generation. This paper explains the algorithms of the infrastructure pedestrian detection system, which includes image registration, background modeling, image filtering, candidate generation, feature extraction, and classification. The paper explains the MATLAB implementation of the algorithm with a road-collected dataset and provides analysis for the detection results with respect to detection accuracy and runtime. The algorithm implementation results show an improvement in the detection performance and algorithm runtime.**

***Keywords*—Pedestrian detection; computer vision; image processing; machine learning; vehicle safety**

I. Introduction

Between 2010 and 2013 the number of registered vehicles increased by 16% [1]. This causes a significant increase in the number of road accidents and road fatalities. The number of worldwide deaths because of road accidents was 1.25 million in 2015 [1]. Many safety solutions have been introduced in vehicles to improve road safety: Advanced Driver Assistance Systems (ADAS) is one of them. ADAS technology utilizes on-vehicle sensors to detect surrounding objects and then analyze detection results to avoid accidents and drive safely. Radar, Lidar, and ultrasonic sensors are examples of sensors that are used in ADAS. Cameras are a widely used sensor in ADAS due to the low cost and the rich information they provide. Image processing and machine learning are used to detect objects of interest in image frames, and the results are used in many safety features. A basic vision-based object detection system includes the following processes: image acquisition using a camera, candidate generation for the object of interest, feature extraction to describe the candidates, and finally, a trained classifier to classify candidates.

The candidate generation process is a very critical step in the detection system and it has a direct impact on the detection accuracy and the processing requirements. There are many approaches for pedestrian candidate generation. The basic approach is the multiple size image scanning, where the whole image is scanned by a sliding window at multiple sizes to detect pedestrians at different sizes and distances. Papageorgiou and Broggi used a window of 64x128 for pedestrian detection and image sizing between 0.2 to 2 of its original size with a step of 0.1 [2]. The flat world approach for candidate generation assumes the world is flat, and it generates the candidates from the ground plane level [3]. This approach provides inaccurate results when the camera location changes with respect to the ground because of vehicle dynamics and road slope. Many solutions introduced to stabilize the images using horizontal edges histogram [4] and features matching [5], but they are computationally expensive. The stereo vision is another approach for candidate generation, where a constructed 3-D map is used to identify the regions of interest to generate the candidates [6]. This approach is expensive since it requires two cameras for the 3-D map construction and it requires a lot of computations.

The current on-board candidate generation approaches can't distinguish between static and moving objects in an image. This leads to the generation of many unnecessary candidates, which can cause false detection and increases the algorithm runtime. An example of this is generating candidates for trees and buildings in an image and misclassifying them as pedestrians.

This paper introduces a new model for candidate generation using connected vehicles and background modeling. The model suggests that images of roads are collected by on-vehicle cameras and the frames are transferred to the infrastructure using V2I. Images that belong to the same location are processed together to generate a background model and improve candidate generation and then pedestrian detection system.

According to a study done by National Highway Safety Admiration (NHTSA), V2V can address 79% of all vehicle crashes while V2I can handle 28% of traffic light accidents [7]. Because of connected vehicles potential in road safety, many researches have been aiming to extend connected

vehicles, capabilities in images and video sharing. Video sharing using V2V was experimentally implemented in [8]. Vehicle connectivity for video sharing using 5G network was proved in [9].

This paper focuses on the image processing and machine learning algorithms that needs to be implemented in the infrastructure for accurate detection results. The second section of this paper provides an overview for the infrastructure pedestrian detection system. The third section explains the algorithms of the infrastructure detection system. The fourth section explains the infrastructure implementation in MATLAB. The fifth section shows the algorithm results and compares them to a reference on-board detection algorithm. The sixth section summarizes the conclusions of this research.

II. Infrastructure System Overview

Implementing a pedestrian detection system in the connected vehicles needs special requirements in the vehicle, V2I communication channel, and the infrastructure system. This section provides an overview of the infrastructure background modeling for pedestrian detection.

A. Vehicle Components

The system requires a vehicle with a forward-looking camera for video collection. V2I transceiver is also required to transfer image frames from vehicle to infrastructure. Other information such as GPS data and vehicle dynamics shall be transferred along with the images for registration.

B. V2I Communication

The image frames and their associated data are transferred via V2I channel. The channel shall have enough bandwidth for image transfer. The channel shall have acceptable latency for real time detection. The channel shall meet other communication specifications such as data encryption and data security.

C. Infrastructure Database

The image frames and their associated data are stored in the infrastructure database. The database is real time maintained with every passing vehicle. Image frames that belong to the same location are grouped together.

D. Infrastructure Pedestrian Detection System

The infrastructure has the history images of a location that was collected by the passed vehicles. History image availability makes pedestrian detection in the infrastructure different from onboard approaches. The infrastructure pedestrian detection system includes following processes:

1) Image registration: Vehicle cameras have different specifications such as resolution, field of view and orientation. Therefore, image registration is required to match images together to be processed as a group. There are many registration techniques to handle this challenge. Vitoza and Flusser provided a review for image registration approaches that can be utilized in this step [10]. Harris -Stephen approach is used in the pedestrian detection system for images alignment and registration.

2) Background modeling: Image frames belonging to the same location are used for background modeling. The background model is used for foreground pixels extraction from the current frame. There are many approaches for background modeling. The used background modeling shall have the ability to handle dynamic changes in background images, such as removing and inserting objects. The background model shall be real time maintained to have the latest updates of road conditions.

3) Foreground regions extraction and candidate generation: The background model is compared to the current image frame to extract the foreground pixels. Image filtering is required to remove the noise and construct the shape of the moving regions. Finally, candidates are generated only from the foreground regions by applying image thresholding.

4) Feature extraction and classification: Features such as edges, corners, and colors are extracted from the candidates for better object description. The feature vectors of the candidates are passed to a trained classifier to classify them as pedestrians or non-pedestrians. Gerónimo and López provided a review for the different approaches of feature extraction and classification in pedestrian detection [11]. Fig. 1 provides the block diagram of the infrastructure improved candidate generation in pedestrian detection using background modeling. Al-refai, Horani and Rawashdeh provided a detailed system architecture and specifications of the infrastructure pedestrian detection system [12].

III. Infrastructure Pedestrian Detection System Algorithms

This section introduces and explains the algorithms to implement the infrastructure pedestrian detection system. The proposed system includes image registration, background modeling, foreground regions extraction, image filtering, candidate generation, feature extraction, and classification.

Harris-Stephens approach for corner detection is used for image registration and matching. The Gaussian Mixture Model (GMM) is used for background modeling and maintenance. The foreground regions in images are extracted using the GMM model. The foreground digital mask is filtered using morphological filters. Candidates are generated from the moving regions in the foreground digital mask. Finally, Histogram Oriented Gradient (HOG) and Support Vector Machine (SVM) are used for candidate feature extraction and classification.

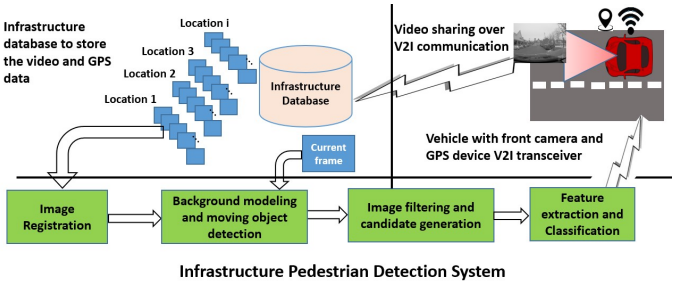

Fig. 1. Background modeling for improved candidate generation in pedestrian detection using V2I block diagram

The following subsections explain the algorithms and the mathematical model for each process.

A. Harris-Stephens Corner Detection for Image Registration

Images collected by vehicle's front cameras don't have the same specifications. They have different rotations, field of view and resolution. In this step, the images in the database for a certain location will be registered and aligned to a reference image. The reference image is selected to be the first image captured by a vehicle for the location. Reference image shall be updated every certain amount of time to include the latest updates in the scene. In our collected dataset, the first image in the sequence is selected to be the reference image.
Image registration includes three steps: Feature extraction from the reference image and the target image, Feature mapping and image transformation.

Harris-Stephen proposed an algorithm for corner detection [13]. This algorithm is used in our system for image registration. Corner features are selected as a control point in the registration for the following reasons:

- Corners are common features in roads
- Corners invariant to geometric changes
- Corners are invariant to resolution change
- Corners are Partially invariant to intensity values

Corners are detected by measuring the change in the intensity values of the pixels in the x and y directions. If the change in the intensity values are large in both directions, then it is considered as a corner. More information about the algorithm implementation can be found in [13].
The next step is to match the features in the reference image and the target image.One of the best algorithms for feature matching is the nearest neighbor distance ratio (NNDR) [14]. The NNDR algorithm works as following:

- Compute the distance between the corners vector in the reference image f_r and the nearest neighbor corners vector in the target image f_{t1} using the sum of square root differences (SSD).

$$d_1 = \sum_{i=1}^{n} (f_{t1} - f_r)^2 \quad (1)$$

where
L: The length of the feature vector i
f_r: A feature vector in the reference image
f_{t1}: The nearest neighbor vector in the target image

- Compute the distance between the reference image feature vector and the second nearest neighbor in the target image

$$d_2 = \sum_{i=1}^{n} (f_{t2} - f_r)^2 \quad (2)$$

- If the ratio between the two distances d_1/d_2 is low, then it is a good match. If the ration is greater than the threshold "MaxThrshld", then the algorithm eliminates the matched as ambiguous.

The last step of the registration is the image transformation. In this step the transformation factors are predicted. this includes image rotation in the pitch, yaw and roll directions, image translation and scaling. The transformation matrix is 3 x 3 with eight unknowns, so the minimum required matching points between the reference image and the target image shall be four. Fig. 2 shows the block diagram of the image registration block.

Fig. 3 shows an example of corner detection and feature matching for a rotated image. Registered images are passed to GMM for background modeling as explained in the next section.

B. GMM for Background Modeling and Foreground Extraction

This part explains GMM for background modeling and feature extraction and compares GMM to the mean filter for background modeling to highlight the advantages of GMM over the basic approaches for background modeling and foreground extraction.

Background modeling and foreground pixel extractions are generally done in three steps: background modeling, background maintenance, and foreground detection. The background modeling step uses the previous image frames to create a model of the background. The background model can

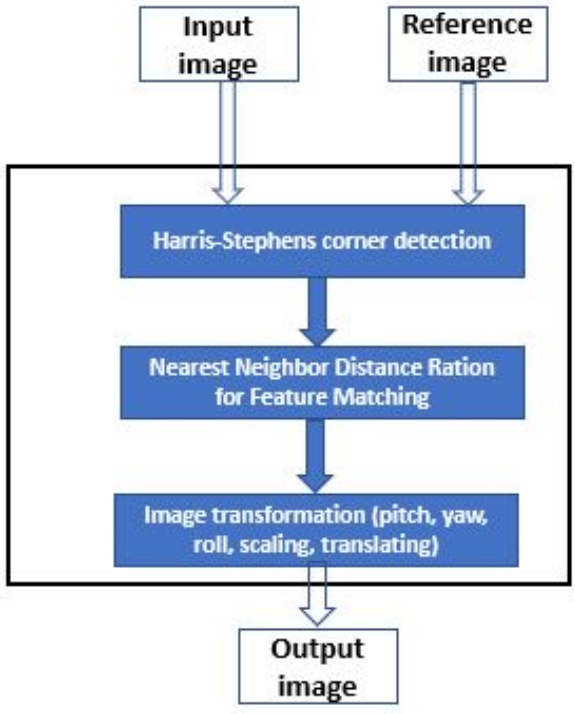

Fig. 2. The registration system block diagram

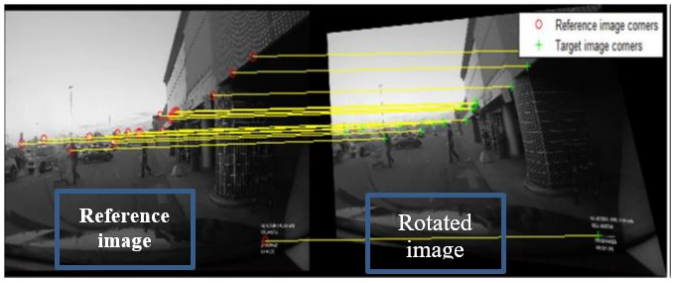

Fig. 3. Image registration using Harris-Stephens corner detection and nearest neighbor ratio for feature matching

be an image or mathematical function such as a probability density function.

Many changes occur in images for a location over time. As objects move, objects are removed from the background and others are inserted. Background maintenance is needed as a mechanism to adapt the background to the latest changes. Many approaches were developed for background maintenance, and they are generally categorized as a blind maintenance and a selective maintenance.

The maintained background model is used to extract the foreground pixels by comparing the current image to the background. The simplest approach to extract the foreground regions is to subtract the current frame from the background model. Other approaches use statistical modeling for background estimation.

One of the basic ways for background modeling is the mean filter [15], which is given by:

$$B(x, y, t) = \frac{1}{n} \sum_{i=1}^{n} I(x, y, t-1), \tag{3}$$

where B(x,y,t) is the background model at time t, I(x,y,t) is the image frame with (x,y) pixels at time t, and n is the total number of image frames. Then foreground pixels are determined by:

$$F(x, y, t) = |I(x, y, t) - B(x, y, t)| > T \tag{4}$$

where T is a fixed threshold value. Median filter is also used for background modeling [16]. The background is maintained by adding a portion of the current image to the background model:

$$B(x, y, t+1) = (1-\alpha)B(x, y, t) + \alpha I(x, y, t), \tag{5}$$

where α is the learning rate which is a constant in [0,1], usually it is 0.05.

Basic approaches have many problems in handling the dynamic changes in the background, such as light variations and shadowing. Also, the basic models require a large memory. Statistical approaches were introduced to handle the dynamic changes in the background. In the statistical approaches, the intensity values of the pixels are modeled in a probability density functions (PDF). Then the PDFs are used to estimate the current pixel as belonging to the background or not. Background modeling using a single Gaussian function is proposed in [17]. However, one PDF for each pixel is insufficient to model the background in a dynamic environment. To solve the problem, a mixture of Gaussians is used to model the background [18]. It is also called Gaussian Mixture Model (GMM). GMM solves many issues for background modeling such as removed background objects and inserted background objects. The memory requirement of GMM is less than the basic approaches. More details about background modeling approaches and foreground detection can be found [19] and [20].

GMM is used in our proposed model as introduced by Stauffer and Grimson [18]. A simplified explanation of GMM mathematical model is provided below:

At any time t, what is known about a particular pixel is its intensity history values. A recent history of each pixel $\{P_1,.....,P_t\}$ is modeled by a mixture of K Gaussian distributions. The probability of observing the current pixel value (C_t) is:

$$P(C_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \sigma i, t), \tag{6}$$

where

K: the number of Gaussian functions to model the pixel

$\omega_{i,t}$:the estimated weight of the ith Gaussian in the mixture at time t

$\mu_{i,t}$: the mean value of the ith Gaussian in the mixture at time t

$\sigma i, t$: the standard deviation of the ith Gaussian in the mixture at time t, and

$$\eta(x_t, \mu, \sigma) = \frac{1}{(2\pi)^{n/2}|\sigma|^{1/2}} e^{\frac{-1}{2}(x_t-\mu_t)^T \sigma^{-1}(x_t-\mu_t)}, \tag{7}$$

Every new pixel value, P_t, is checked against the existing K Gaussian distributions until a match is found. A match is defined as a pixel value within 2.5 of the standard deviation σ of a distribution.

The maintenance of the model is done with a new pixel based on the pixel to GMM match. There are two cases for the maintenance as following:

Case one: If none of the K distributions match the current pixel value, then the least probable distribution is replaced by a distribution with the current pixel value as its mean value, an initially high variance, and low prior weight

$$\mu_t = P_t \tag{8}$$

$$\omega_{i,t} = \alpha \tag{9}$$

where α is the learning rate of the GMM

Case two: If one or more distribution functions match the new pixel value, then the matched functions' parameters are updated as following:

$$\mu_t = (1-\rho)\mu_{t-1} + \rho P_t \tag{10}$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(P_t-\mu_t)^T(P_t-\mu_t) \tag{11}$$

where

$$\rho = \alpha\eta(P_t|\mu_k,\sigma_k) \tag{12}$$

The weights of the K distribution are updated as following:

$$\omega_{t,k} = (1-\alpha)\omega_{k,t-1} + \alpha M_{k,t} \tag{13}$$

where $M_{k,t}$ is 1 for models that are matched and 0 for the remaining distributions. The $M_{k,t}$ and σ parameters for the unmatched distributions remain the same.

To estimate the foreground pixels of the current frame using the GMM model, the Gaussian distributions for each pixel are sorted in descending order based on the value ω/σ. This value increases as a distribution gains more evidence to represent a background pixel. The most likely background distributions remain on top and the less probable transient background distributions gravitate towards the bottom and are eventually replaced by new distributions. The algorithm selects the first B distributions that counts for a predefined fraction of the evidence T.

$$B = argmin_b\{\sum_{i=1}^{b}\omega_i > T\} \tag{14}$$

where

B: the distributions that represent the background model

ω_i: the weight of the distribution i

T: a threshold for the minimum background ratio to the image, usually 0.7.

The output of the GMM is a digital image with values of zero or one. Ones represents the foreground pixel and appears in white color. Zeroes represents the background pixels and appears in black color. GMM shows a very good result for foreground extraction when there are statically moving objects, such as moving trees due to a wind. It also shows a good maintenance for the background with removed and inserted background objects. The output image of the GMM is called foreground digital mask.

Fig. 4 shows an example compares between the mean filter and the GMM in foreground pixels extraction. 70 images were captured for a road intersection at different time stamps and used for the background modeling, the time separation between the frames is 10 sec. The left image shows a vehicle that was inserted in the background in the last 30 image frames, this car shall be categorized as background object. The mean filter has detected the car as foreground, while GMM adapted to the inserted object (the car) quickly and categorized it as background.

C. Morphological Filtering

The background model using GMM may have false positives in some regions of the image due to statically moving objects, and objects that were removed or added to the background. It can also miss-detect foreground pixels due to the similarity of the foreground pixels and the background. Morphological filtering removes the noise in the foreground digital mask by connecting the neighbor foreground regions to construct the shape of the objects. It disconnects the small and the outlier foreground regions that doesn't belong to the same object. It also closes the small holes in the foreground digital mask.

Morphological image processing is suitable for binary image processing since it depends only on the relative ordering of pixel values, and not on their numerical values. Morphological operations are a collection of non-linear operations related to the shape or morphology of features in an image. More details about morphological filtering can be found in [21].

There are two fundamental operations for morphological filtering, erosion and dilation. Also, there are compound operations by mixing the erosion and the dilation. Opening filter is erosion followed by dilation. closing filter is a dilation followed by an erosion. Fig. 5 shows an example of a binary image filtered with opening filter and closing filter. As shown in the in the figure, the opening filter (the center image) connects the close foreground regions together, which helps constructing the shape pf the foreground object. The closing filter (the right image)removes the small foreground regions, which helps in removing the small false foreground extractions. After trying many morphological filters with many sizes and structures, observations showed that filtering an image with a 10x10 square closing filter followed by a 3x3 opening filter provides the best result to remove the noise from the foreground digital mask and connect the foreground regions. The closing filter constructs the shape of the moving regions by connecting them together. The opening filter removes the small holes in the image. Fig. 6 shows examples of the foreground digital mask filtering using a square closing filter with size of 10x10 followed by an opening filter with size of 3x3.

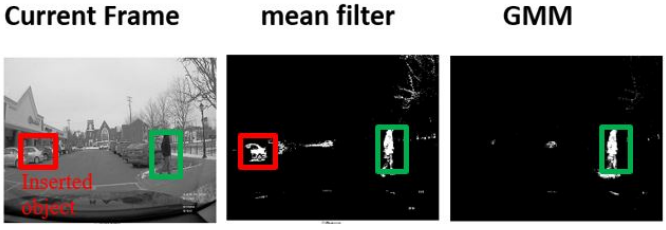

Fig. 4. The first image shows the image frames, the second image shows the foreground digital mask using the mean filter, and the right image shows the foreground digital mask using GMM. The true positives are highlighted in green, while the false positives are highlighted in red

Fig. 5. The first image is the foreground digital mask of a moving object using GMM, the second image shows the output image after applying a closing filter, and the third image is the output of implementing the opening filter

D. Foreground Digital Mask Thresholding for Candidate Generation

A typical candidate generation scans the whole image or a large portion of it to generate the candidates. In our infrastructure system, the generation of the candidates focuses on the foreground regions only and excludes the background objects.

The candidate generation in the infrastructure algorithm applies a threshold to the foreground digital mask. The digital mask is scanned by a sliding window of a 64x128. The mean of the window is calculated; if the mean is higher than the threshold, the same window in the corresponding image is passed to the next step, and, if not, the region is excluded from being a candidate and the scanning window moves to the next region in the digital mask.

Fig. 7 shows the flowchart of the candidate generation. The image is scanned at multiple sizes of its original size. The scanning is applied on the images while resolution is varied from 0.5 to 1.3 with a step of 0.1. Fig. 8 shows an example of how the candidate generation approach is applied on an image. The main advantage of the candidate generation using infrastructure background modeling is to reduce the number of the candidates from the static regions. This reduction is reflected in the performance of the detection algorithm as shown in the system evaluation section.

Fig. 6. The first column shows three input images, the second column is the foreground digital mask using GMM, and the third column is the filtered mask using a square closing filter of 10x10 followed by a square opening filter of 3x3

E. HOG and SVM for Feature Extraction and Candidate Generation

Pedestrians are one of the most complex objects to detect because they can appear in different sizes, poses, and colors. The shape of the pedestrian may change while carrying different objects. The change in the outdoor light conditions is another challenge. To go over these challenges, unique features of the object are extracted to provide a robust description of pedestrians. These features can be textures, contours, and edges. The features of the object should be very similar under different view conditions.

Histogram Oriented Gradient (HOG) feature extraction is considered as one of the most successful approaches for pedestrian detection when it is used with Support Vector Machine (SVM) classifier. This model was introduced by Dalal and Triggs in 2005 [22]. The main advantages of

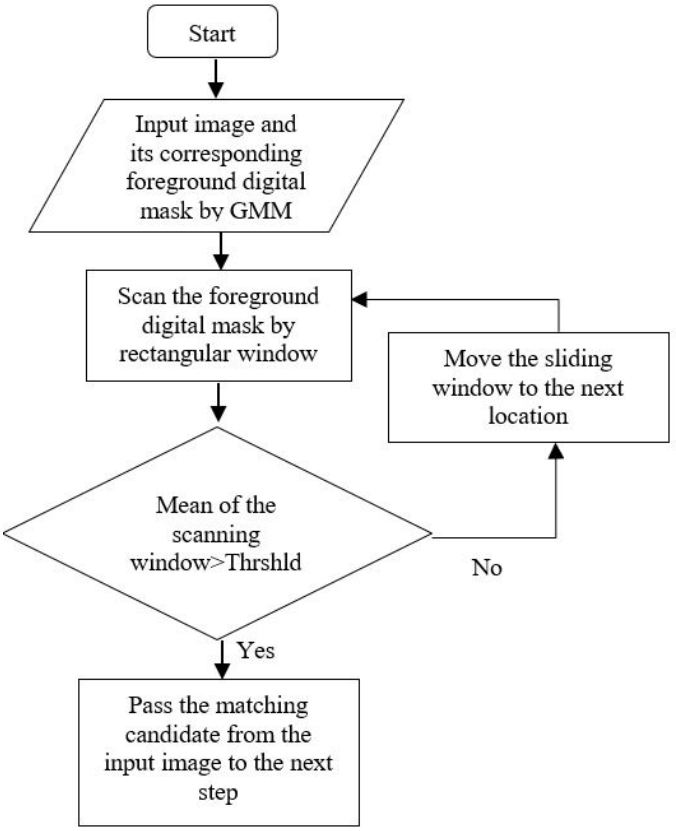

Fig. 7. Candidate generation flow chart

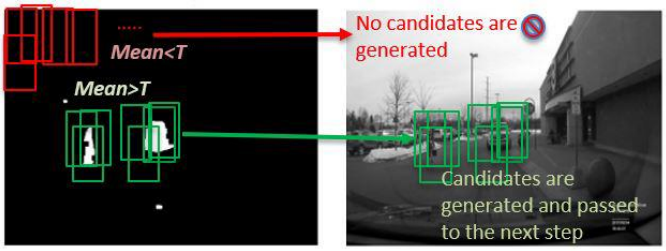

Fig. 8. The left image shows the candidate generation, the right image comparing the mean of the scanning window to a threshold to make the decision for candidate generation

HOG are the induced robustness against the global and the local illumination changes, the moderation of pedestrian pose differences, and algorithm runtime. HOG with SVM are used for pedestrian detection in our infrastructure module due to its accurate detection result.

HOG is calculated by computing the first order image gradients. It captures the object contours and the texture information. Features are collected in a vector and passed to the classifier. Dalal and Triggs explained the mathematical model of the algorithm and analyzed the detection results using many human datasets [22].

SVM is a learning model that analyzes the training data and build a set of rules to classify similar observations that haven't been seen before. SVM requires training data for each class. In our case, the classes are pedestrian and non-pedestrian. HOG vectors are passed to the SVM to develop the classification rules. More information about the SVM model can be found in [23]. One of the main advantages of the SVM is the ability to use Kernel functions to transfer the data to a higher dimensional domain to provide an accurate classification for the non-linearly separable data. More information about the training data and the used SVM parameters are listed in the implementation section. Fig. 9 shows the block diagram for the HOG with the SVM for pedestrian detection. Fig. 10 shows the block diagram of the infrastructure pedestrian detection algorithms including the image background modeling and moving object detection, image filtering, image thresholding and candidate generation, HOG, and SVM for pedestrian classification.

IV. The Infrastructure Pedestrian Detection System Implementation

A. Testing and Training Datasets

To implement the infrastructure system, a labeled dataset is required for SVM. A test dataset is also needed to verify the system results. There are many pedestrian datasets available online, such as INRIA and MIT. However, none of these datasets can be used to implement the infrastructure system since multiple images for the same location at different time stamps are required for background modeling.

A vehicle was setup with a front windshield camera for video collection. The camera is equipped with external memory to store the videos. The used camera is a 3.2 MP CMOS sensor with a 135 degrees field of view. The output video resolution is1280x720 with 60 fps.

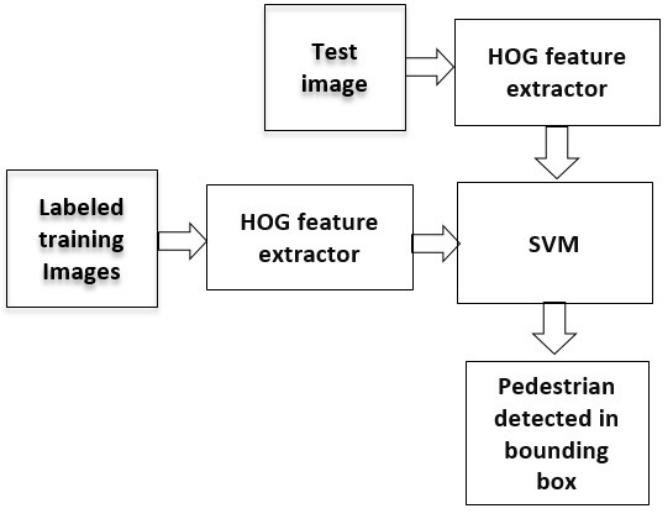

Fig. 9. Histogram Oriented Gradient (HOG) and Support Vector Machine (SVM) for pedestrian detection block diagram

1) The training dataset: Videos were collected from locations with pedestrian traffic, such as downtowns, shopping centers, and school campuses. Videos were sampled to frames, and the "Training Image Labeler" MATLAB tool was used to label pedestrians in the images. The training data include 1066 positive samples and 1600 negative samples. Fig. 11 shows an example of positive samples for pedestrians, and negative samples like trees and buildings.

2) The testing dataset: Testing videos were collected while the vehicle was stationary to capture multiple images for the

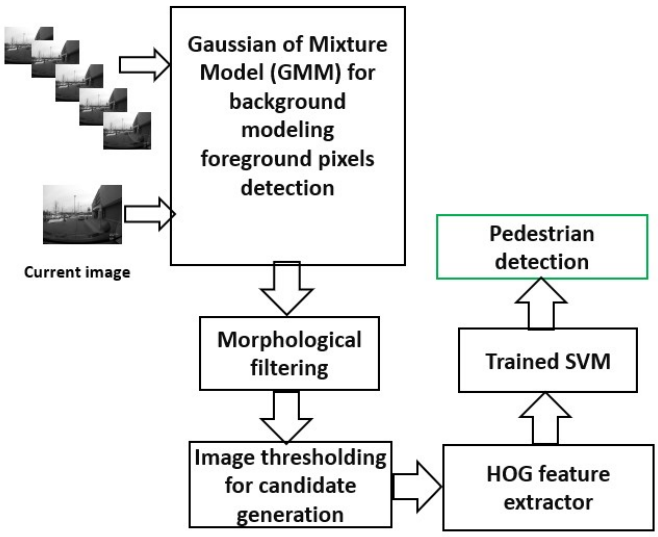

Fig. 10. Infrastructure image processing module algorithms for pedestrian detection

same location. The testing data were collected from different locations with vehicle and pedestrian traffic. By sampling the collected videos for each location to frames, many image frames for the same location at different times are available for background modeling. The test dataset specifications are as following:

- The testing data collected for 100 different locations
- The video length for each location is 330 sec
- Each video is sampled to frames with rate of 1 fps
- Pedestrians are labeled in the last image frame for each location

The last image frame represents the current image frame of the passing vehicle, where the algorithm shall detect the pedestrians in it. The separation time between the frames can be selected by the algorithm. For example, if the algorithm reads one frame every 10 seconds, this represents a vehicle passing from the location every 10 seconds. The separation time between the frames represents the road traffic. The time separation impact in the detection is studied later in this section. The first frame represents the reference frame for image registration. Fig. 12 shows an example of testing image frames for a location; it shows the previous frames and the current image frame with the labeled pedestrian.

B. MATLAB Implementation

The infrastructure pedestrian detection system is implemented using MATLAB. The implementation is divided in blocks as following:

- Testing image frames read:

  In this block, the image frames for a location is imported to MATLAB. The time separation between the frames (T sep) can be selected by the algorithm to simulate the different traffic conditions. This time represents the time between the passing vehicles.

- Image registration:
  Images for each location were registered to the reference frame using Harris-Stephens approach as explained above.

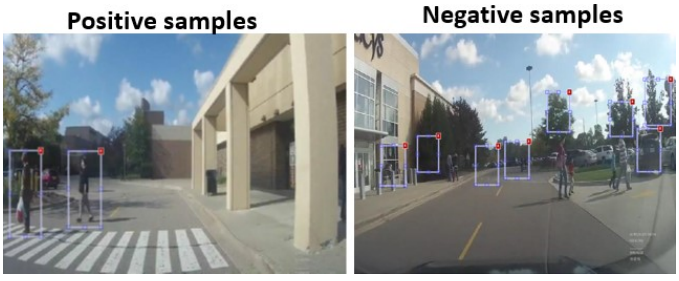

Fig. 11. The left image shows labeled positive samples and the right image shows labeled negative samples. The images were labeled using the MATLAB toolbox "Training Image Labeler"

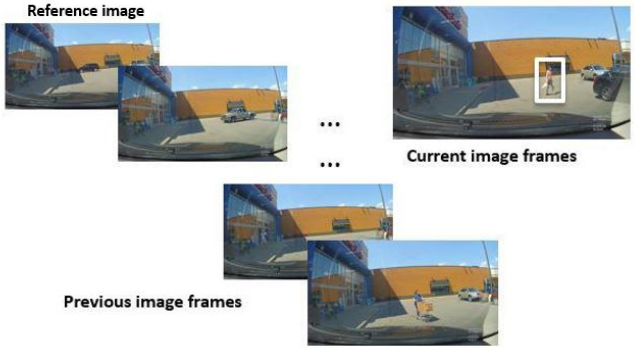

Fig. 12. Testing images for a location, it includes the previous image frames and the current image with a labeled pedestrian

- GMM for foreground extraction:
  The imported images for a location are passed to the GMM for background modeling and foreground pixels extraction. Table I summarizes the GMM parameters and their nominal values used in the implementation.

- Morphological filtering:

  The foreground digital mask is filtered using a 10x10 square closing filter to connect the foreground regions followed by a 3x3 square opening filter to close the small holes in the foreground mask.

- Candidate generation:

  The foreground digital mask is scanned by a 64x128 window at multiple sizes from 0.5 to 1.3 with a step of 0.1. If the mean of the window is greater than a threshold, the candidate is passed to HOG. The threshold value impact in the detection result is studied in the next section.

- HOG with SVM:

  HOG is used to extract the features of the candidates. Each candidate produces 3780 features. SVM is trained using 1066 positive samples and 1600 negative samples. The implemented HOG main parameters are shown in Table II.

TABLE I. THE NOMINAL VALUES FOR GMM PARAMETERS IN THE MATLAB IMPLEMENTATION

GMM parameters	Nominal values
Learning rate (α)	0.005
Maximum background ratio	0.7
Initial variance	900
Number of Gaussians	5

C. Frames Separation Time Impact on the Detection (T_{sep})

The time separation between the frames is an important factor that affects the system, as it specifies in which traffic situations the system can be implemented. Urban areas such as downtowns and shopping centers usually have a high vehicle traffic, so the time separation is short, while it is longer in rural areas with low traffic.

The time separation factor (T_{sep}) is studied for the following values: 5 seconds, 10 seconds, 20 seconds, 30 seconds, and 40 seconds. Fig. 13 shows the Receiver Operating Characteristic (ROC) of the precision and the recall for each time separation value; the performance of the system is very similar for all the (T_{sep}).

The result shows that the system provides a good detection result under many traffic conditions. However, in low traffic conditions, there is more chance to miss some changes in the background. This will result in false foreground detection, which means generating more candidates from background regions.

V. Infrastructure System Evaluation

For better understanding of the infrastructure pedestrian detection system results, a comparison of the detection results is done with a reference approach of a traditional on-vehicle pedestrian detection system. Detection results were compared with respect to the detection accuracy by counting false positives and false negatives reported by each system. The runtime of the reference algorithm was also compared to the infrastructure system to show the effect of the improved candidate generation of the infrastructure system in the processing time of the detection.

TABLE II. The nominal values for the HOG parameters in the MATLAB implementation

HOG parameters	Nominal values
Cell size	8x8 pixels
Block size	2x2 cells
Block overlapping	50%
Number of histogram bins	9

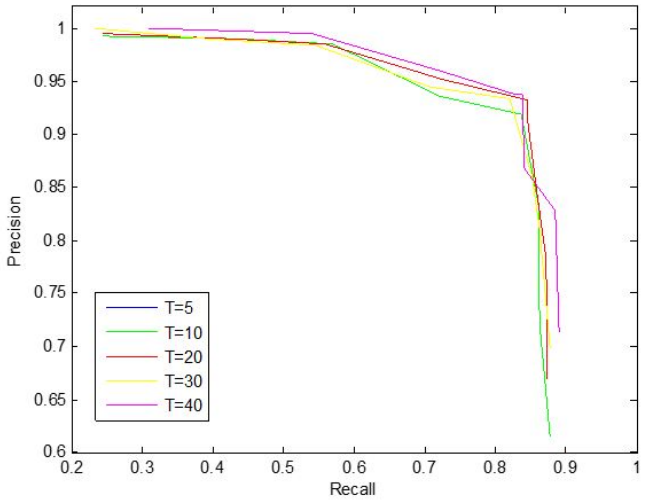

Fig. 13. Precision vs. Recall ROC plot for different frame separation time (T_{sep})

In the reference algorithm, the candidates were generated using multiple size image scanning. The input image is scanned with a fixed scanning window of 64x128. The scanning included the whole image except the top of the images that includes the sky. Images were scanned between 0.5 to 1.3 of their size, with a step of 0.1.

Candidates were passed to HOG for feature extraction, and then a trained SVM was used for classification. Fig. 14 shows the block diagram of the reference pedestrian detection system and the proposed infrastructure pedestrian detection system. The blocks colored in green are common between the reference and the infrastructure system. The blue blocks are related to the on-vehicle detection system, while the yellow ones are related to the proposed infrastructure system. That means any improvement in the detection result in the infrastructure system is related to the improved candidate generation using background modeling and foreground pixels extraction.

A. Detection Results

The testing dataset for the 100 locations were passed to the infrastructure system for pedestrian detection. The labeled testing frames were also passed to the reference algorithm. No previous images were used in the reference algorithm since there is no background modeling.

The number of the generated candidates by the reference algorithm using multiple size image scanning was 42900 for the whole dataset. The total number of the generated candidates using the infrastructure algorithm was 28750. The first advantage of the infrastructure system is the reduction in the number of the candidates by 33% when compared to the reference algorithm.

The infrastructure system reported 24 false positives. The reference algorithm reported 98 false positives. This shows a 75.5% reduction in false positives in the infrastructure system. This significant improvement is due to the reduction in the number of candidates that is generated from the background region that may cause more false positives in the reference algorithm.

The infrastructure system showed better results in false negatives compared to the reference algorithm. The total number of false negatives reported by the infrastructure model is 19, as compared to 24 for the reference algorithm. The reason for the reduction in the false negatives in the infrastructure is that candidate generation is focused in the foreground pixels, which increases the possibility of capturing candidates for a pedestrian in different poses and angles, thereby increasing

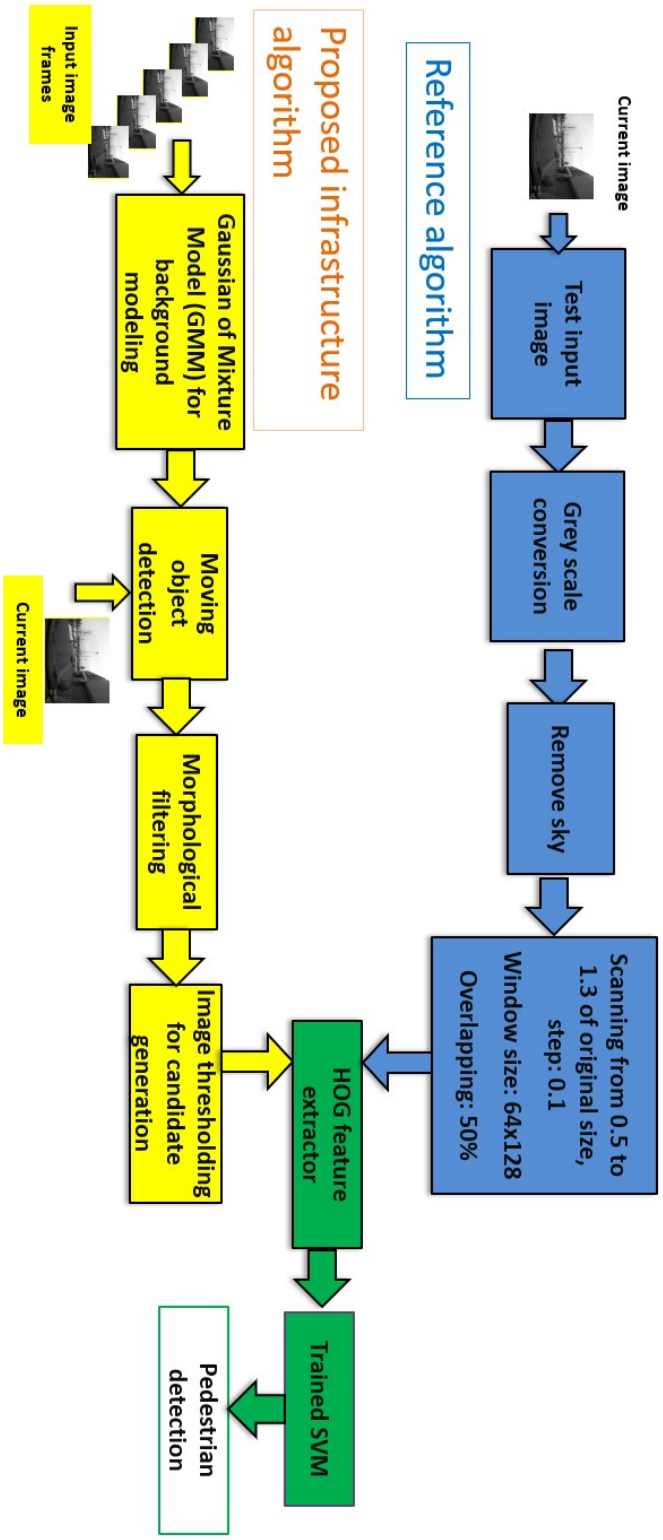

Fig. 14. The infrastructure Pedestrian Detection system and the reference algorithm block diagram

the chance to classify the candidates correctly. Table III summarizes the detection results of the infrastructure system and the reference system.

Fig. 15 shows the Receiver Operating Characteristic (ROC) curve of the precision and the recall for the infrastructure system and the reference system. The infrastructure system shows very high precision values when compared to the reference algorithm. It also shows a better recall at many SVM operating points.

Fig. 16 shows an example of the reference algorithm detection compared to the infrastructure algorithm. The reference algorithm showed a false positive for a background object highlighted in red, while the infrastructure system didn't report the same false positive. Fig. 17 shows another example of a pedestrian miss-detection in the reference algorithm, while it is detected in the infrastructure algorithm.

B. Algorithm Runtime

One of the main advantages of the infrastructure algorithm is the reduction in the number of the candidates, which reduces the runtime of the detection system. The runtime of the infrastructure system was compared to the reference detection system by computing the time to process and classify the testing dataset. The computer specifications used for the runtime study are listed below:

- Processor: Intel(R) Core(TM) i7-8550U CPU @ 1.8

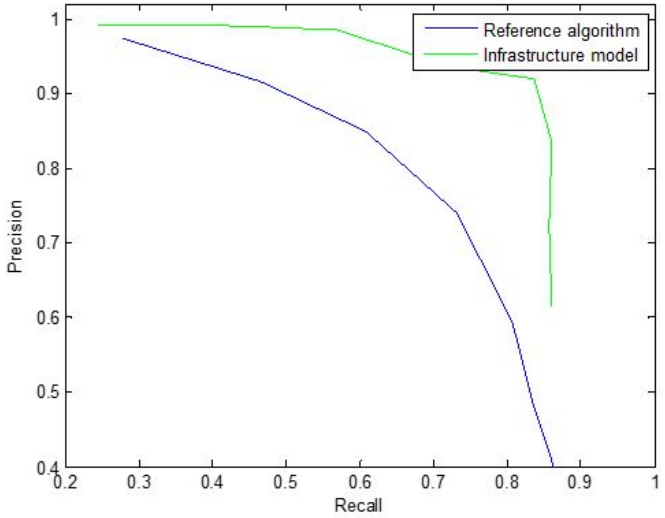

Fig. 15. Precision vs. Recall ROC for the infrastructure system and the reference system

Fig. 16. Left image shows a false positive for the stop sign reported by the reference algorithm, the right image shows the detection result for the same image with no false positive using the infrastructure model

TABLE III. THE DETECTION RESULTS SUMMARY FOR THE INFRASTRUCTURE MODEL AND THE REFERENCE ALGORITHM

Method	Total candidates	False negative	False positive	Recall	Precision
Infrastructure system	28750	19	24	0.837	0.894
The reference system	42900	24	98	0.807	0.593

Fig. 17. The left image shows pedestrian miss-detection using the reference algorithm, the infrastructure model detected the pedestrian in the same image

GHz

- RAM: 12 GB
- System type: 64-bit operating system

The runtime of the reference system for an image frame is given by:

$$RunTime_{ref} = Numberofcandidates * T_{HOG/SVM} \quad (15)$$

where
$RunTime_{ref}$ is the runtime for one image frame using the reference algorithm measured in sec/frame.

$T_{HOG/SVM}$ is the reference algorithm runtime to classify one candidate in sec/candidate.

$T_{HOG/SVM}$ equals 0.0558 sec/candidate. the number of candidates generated by the reference algorithm is 429 per frame. By applying Equation (13), the runtime for the reference algorithm is 23.938 sec / frame.

The runtime for the infrastructure system for an image is given by:

$$RunTime_{inf} = T_{GMM} + Numberofcandidates * T_{HOG/SVM} \quad (16)$$

where

$Runtime_{infrastructure}$ is the runtime of the infrastructure algorithm for one candidate measured in sec/frame

T_{GMM} is the time to extract the foreground pixels from the current image frame and maintain the background model, the time is measured in sec/frame.

$T_{HOG/SVM}$ is the reference algorithm runtime to classify one candidate in sec/candidate.

$T_{HOG/SVM}$ pf the infrastructure system has the same value in the reference algorithm because HOG and SVM are common processes in the two approaches.

T_{GMM} equals 0.0484 sec/frame in both approaches. The total number of candidates in the infrastructure algorithm is reduced by 33% when compared to the reference algorithm. Therefore, the total number of candidates per image frame using the infrastructure algorithm equals to 287.43 candidate/frame.

By applying Equation (14), the runtime for the infrastructure algorithm equals 16.086 sec/frame. The analysis shows that the runtime of the infrastructure algorithm is reduced by 32.7% when compared to the reference algorithm. Table IV summarizes the runtime analysis for the infrastructure algorithm and the reference algorithm.

VI. CONCLUSIONS

This paper proposed a system to improve the candidate generation process for pedestrian detection in connected vehicles. The system registers the collected images for a location to a reference image. Harris-Stephens approach for corner detection, Nearest Neighbor Distance Ratio (NNDR) for feature mapping and image transformation are used in the registration step.

Gaussian Mixture Model (GMM) is used to model the background of a location using the registered images stored in the infrastructure database. The foreground pixels in the images extracted using the GMM model. Candidates are generated through scanning the foreground regions by a rectangular box. Finally, Histogram Oriented Gradient (HOG) and Support Vector Machine (SVM) were used to classify candidates as pedestrians or non-pedestrians.

A data-set is collected for algorithm training and test. A reference algorithm is implemented to highlight the

TABLE IV. THE RUNTIME FOR THE INFRASTRUCTURE MODEL AND THE REFERENCE ALGORITHM

Detection system	Algorithm runtime (sec/frame)
Infrastructure system	16.086
Reference system	23.938

improvements achieved in the proposed system.

The infrastructure pedestrian detection system showed a huge improvement in detection performance when compared to the reference algorithm that represents a typical on-board detection approach. The infrastructure algorithm significantly reduced the number of the generated candidates when compared to the reference algorithm. The generated candidates in the proposed infrastructure system is reduced by 33%. Also, the false positives are reduced by 75% in the infrastructure system compared to the reference algorithm. Since the infrastructure system classifies less candidates, the runtime of the algorithm is improved by 67% when compared to the reference algorithm.

REFERENCES

[1] World Health Organization. Global status report on road safety 2015. World Health Organization, 2015.

[2] Papageorgiou, Constantine, and Tomaso Poggio. "A trainable system for object detection." International journal of computer vision 38, no. 1 (2000): 15-33.

[3] Broggi, Alberto, Massimo Bertozzi, Alessandra Fascioli, and Massimiliano Sechi. "Shape-based pedestrian detection." In Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE, pp. 215-220. IEEE, 2000.

[4] Broggi, Alberto, Paolo Grisleri, Thorsten Graf, and M. Meinecke. "A software video stabilization system for automotive oriented applications." In Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st, vol. 5, pp. 2760-2764. IEEE, 2005.

[5] Bombini, Luca, Pietro Cerri, Paolo Grisleri, Simone Scaffardi, and Paolo Zani. "An evaluation of monocular image stabilization algorithms for automotive applications." Intel. Transp. Syst (2006).

[6] Llorca, D. F., M. A. Sotelo, A. M. Hellín, A. Orellana, M. Gavilán, I. G. Daza, and A. G. Lorente. "Stereo regions-of-interest selection for pedestrian protection: A survey." Transportation research part C: emerging technologies 25 (2012): 226-237.

[7] Najm, Wassim G., Jonathan Koopmann, John D. Smith, and John Brewer. Frequency of target crashes for intellidrive safety systems. No. DOT HS 811 381. United States. National Highway Traffic Safety Administration, 2010.

[8] Belyaev, Evgeny, Alexey Vinel, Adam Surak, Moncef Gabbouj, Magnus Jonsson, and Karen Egiazarian. "Robust vehicle-to-infrastructure video transmission for road surveillance applications." IEEE Transactions on Vehicular Technology 64, no. 7 (2015): 2991-3003.

[9] Pervez, Farhan, Abdulkareem Adinoyi, and Halim Yanikomeroglu. "Efficient resource allocation for video streaming for 5G network-to-vehicle communications." In Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on, pp. 1-6. IEEE, 2017.

[10] Zitova, Barbara, and Jan Flusser. "Image registration methods: a survey." Image and vision computing 21, no. 11 (2003): 977-1000.

[11] Gerónimo, David, and Antonio M. López. Vision-based pedestrian protection systems for intelligent vehicles. New York, NY, USA:: Springer, 2014.

[12] Al-Refai, Ghaith, Modar Horani, and Osamah Rawashdeh. A Framework for Background Modeling Using Vehicle-to-Infrastructure Communication for Improved Candidate Generation in Pedestrian Detection. 17th Annual IEEE International Conference on Electro Information Technology EIT, 2018.

[13] Harris, Chris, and Mike Stephens. "A combined corner and edge detector." In Alvey vision conference, vol. 15, no. 50, pp. 10-5244. 1988.

[14] Li, Qiaoliang, Guoyou Wang, Jianguo Liu, and Shaobo Chen. "Robust scale-invariant feature matching for remote sensing image registration." IEEE Geoscience and Remote Sensing Letters 6, no. 2 (2009): 287-291.

[15] 14. Lee, B., and M. Hedley. "Background estimation for video surveillance." (2002).

[16] McFarlane, Nigel JB, and C. Paddy Schofield. "Segmentation and tracking of piglets in images." Machine vision and applications 8, no. 3 (1995): 187-193.

[17] Wren, Christopher Richard, et al. "Pfinder: Real-time tracking of the human body." IEEE Transactions on pattern analysis and machine intelligence 19.7 (1997): 780-785.

[18] Stauffer, Chris, and W. Eric L. Grimson. "Adaptive background mixture models for real-time tracking." In cvpr, p. 2246. IEEE, 1999.

[19] Bouwmans, Thierry. "Traditional and recent approaches in background modeling for foreground detection: An overview." Computer Science Review 11 (2014): 31-66.

[20] Bouwmans, Thierry. "Recent advanced statistical background modeling for foreground detection-a systematic survey." Recent Patents on Computer Science 4, no. 3 (2011): 147-176.

[21] Serra, Jean. "Morphological filtering: an overview." Signal processing 38, no. 1 (1994): 3-11.]

[22] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 886-893. IEEE, 2005.

[23] Burges, Christopher JC. "A tutorial on support vector machines for pattern recognition." Data mining and knowledge discovery 2, no. 2 (1998): 121-167.