smart cities

MDPI

Article

Fast Object Detection Using Dimensional Based Features for Public Street Environments

Ivan Matveev [1,2,*,†], Kirill Karpov [1,3,*,†], Ingo Chmielewski [1] and Eduard Siemens [1] and Aleksey Yurchenko [2]

[1] Department of Electrical, Mechanical and Industrial Engineering, Anhalt University of Applied Sciences, Bernburger Str. 55, 06366 Köthen, Germany; ingo.chmielewski@hs-anhalt.de (I.C.); eduard.siemens@hs-anhalt.de (E.S.)

[2] Department of Physics of High Energy Processes, Tomsk Polytechnic University, Lenin Str. 30, 634050 Tomsk, Russia; niipp@inbox.ru

[3] Department of Transmission of Discrete Data and Metrology, Siberian State University of Telecommunications and Information Sciences, Kirova Str. 86, 630102 Novosibirsk, Russia

* Correspondence: ivan.matveev@hs-anhalt.de (I.M.); kirill.karpov@hs-anhalt.de (K.K.)

† These authors contributed equally to this work.

Received: 16 January 2020; Accepted: 26 February 2020; Published: 3 March 2020

Abstract: Modern object recognition algorithms have very high precision. At the same time, they require high computational power. Thus, widely used low-power IoT devices, which gather a substantial amount of data, cannot directly apply the corresponding machine learning algorithms to process it due to the lack of local computational resources. A method for fast detection and classification of moving objects for low-power single-board computers is shown in this paper. The developed algorithm uses geometric parameters of an object as well as scene-related parameters as features for classification. The extraction and classification of these features is a relatively simple process which can be executed by low-power IoT devices. The algorithm aims to recognize the most common objects in the street environment, e.g., pedestrians, cyclists, and cars. The algorithm can be applied in the dark environment by processing images from a near-infrared camera. The method has been tested on both synthetic virtual scenes and real-world data. The research showed that a low-performance computing system, such as a Raspberry Pi 3, is able to classify objects with acceptable frame rate and accuracy.

Keywords: detection; object recognition; background subtraction; smart-city; pinhole camera model; single-board computer; machine learning

1. Introduction

A smart city concept is directly related to gathering data by sensors of the Internet of Things (IoT). Some of these sensors are camera-based and used in services of a smart city, such as surveillance or traffic monitoring. The cameras produce a lot of data which is processed centrally by powerful computers that are equipped with GPUs [1]. The services using camera as a sensor might need to process data decentralized, locally on IoT devices, which is primarily caused by the implementation of the Internet connection. Transmitting video data from numerous IoT devices through the wireless Internet or mesh network might be an inappropriate way due to delays and link reliability. These factors influence the workability of the services which need to react to events in real world immediately. At the same time, single-board computers are getting cheaper and more powerful. They provide a possibility to process video data autonomously. However, this is a challenging task due to their limited processing capabilities.

Conventional street lights are constantly switched on during the dark time of the day. Pedestrian and vehicle traffic is very limited in suburban and rural areas during this time. Therefore, the reduction of energy consumption for lighting is desirable; consequently that leads to reduction of costs for street lighting. The method presented in this work has been developed within the context of smart lighting systems; in particular, the SmartLighting project [2] of the Future Internet Lab Anhalt. The project is aimed at energy saving with intelligent lamps control, depending on the actual objects' motions, especially pedestrians, bicyclists and vehicles. The SmartLighting system is a decentralized system in which lamps communicate through a wireless network. The system calculates the maximum likelihood vector of an object and provides dynamic lighting to the object based on detection information from multiple lamps. Thus, the role of the detection method in such a system is essential since it influences its overall effectiveness and workability.

The task of the outdoor object detection has been a subject of a study for a variety of detection methods and techniques. To detect moving objects, conventional methods exploit their different physical properties and effects, such as infrared radiation, Doppler shift, etc. However, they have a lot of disadvantages detecting moving objects; in particular, low detection range, high detection error rate and high price. The detection reliability of such methods highly depends on the operating conditions. In addition, they have limitations regarding the type and speed of the object to be detected. Computer vision methods recently reached significant success in object detection. However, these methods require high computational power of the processing unit. It makes impossible to achieve a sufficient frame rate (about 15 FPS) using low-performance single-boards computers such as the Raspberry Pi [3], even if they are supported by hardware acceleration units [4]. Therefore, finding and developing new efficient and low-complexity algorithms for object classification becomes a crucial task for the outdoor applications, especially in smart city approaches.

The main objective of this paper is a demonstration of the designed method which satisfies several requirements which are critical for street lighting systems:

- low computational requirements of the algorithm so it can be operated on inexpensive, low-performance and low-energy consumption devices (e.g., Raspberry Pi, Orange Pi and alike)
- ability to detect objects moving with the speeds up to 60 km/h (default speed limit within the urban area [5]) with ranges up to 25 m; the detection range requirement is justified by expected roads width in the suburban area: 2–4 vehicle's lanes and two sidewalks [6,7]
- less than 5% false positive error rate
- less than 30% false negative error rate
- detection of object types which are in need of illumination and expected in the street environment; the system uses information about object type to adjust illumination area, e.g., to provide longer visibility range for vehicles or dime the light according to the object class
- ability to use low-resolution frames (not less than 320 × 240 pixels) captured in gray-scale by an inexpensive camera with infrared illumination

The false positive detection error mainly impacts the energy efficiency of the SmartLighting system. When a false positive error occurs, the street light is switched on while the target object is not present. The reduction of such errors increases overall energy efficiency. The error is considered to be a false negative when the system ignores the desired object. The reduction of false negative errors improves lighting system precision to provide timely illumination. The system is more tolerant to false negative errors than to false positive ones since it is compensated by true positive detection on the subsequent frames.

The main contribution and scientific novelty of this work is a mechanism of computationally inexpensive object feature extraction which uses methods of projective geometry and 3D reconstruction. Another contribution is a fast detection pipeline which meets the requirements for the limited camera resolution and computational resources of processing device for classification. The developed method proposes an idea of estimating the geometrical properties of an object to perform object classification based on these properties. The logistic regression has been chosen as a classification model due to its simplicity from a perspective of computational load [8]. Thus, the developed method is an attempt to close the gap between computationally expensive algorithms and low-performance hardware.

The remainder of this paper is structured as follows. The existing methods of object detection are discussed in Section 2. Section 3 is devoted to the general description of the proposed method stages. The feature extraction method is described in Section 4. The section shows how the algorithm evaluates geometrical parameters of moving objects, e.g., distance, object width and height. The classification method is explained in Section 5. The section contains a detailed description of features extracted from photos, target classes of the objects and their properties, data acquisition techniques and classification model. Interpretation and demonstration of the experimental results can be found in Section 6. Section 7 is devoted to the conclusion, where the performance of the algorithm is discussed. Plans on future work are described in Section 8.

2. Related Work and State of the Art

Conventional technologies for moving object detection include passive infrared (PIR), ultrasonic (US) and radio wave (RW) detection approaches [9]. Devices which are based on these technologies can operate in conditions of limited visibility, moreover, time of day does not affect their performance. PIR sensors are widely used in various indoor and outdoor detection applications. Such sensors are low-power, inexpensive and easy to use [10]. However, their efficiency is dependent on the ambient temperature and brightness level. The most critical limitation of commercial PIR sensors is the relatively small operating range which rarely exceeds 12 m [11,12]. US sensors are able to provide object distance information and therefore are often used in robotics applications [13]. A significant disadvantage of US sensors is a quite narrow horizontal angle of operation of about 30–40° and high sensitivity which often leads to false triggering. In addition, environmental conditions impact sound speed, leading to significant bias in object detection accuracy [14]. RW sensors are widely used for object detection in security or surveillance systems. They have high sensitivity [9] which causes frequent false positive triggering due to vibrations or negligible movements [15]. RW sensors have a detection range, comparable to PIR sensors. The hybrid systems based on combinations of RW-PIR [16] or US-PIR [17] are designed for indoor object detection, and they are hardly applicable to street scenarios.

Computer vision methods also provide many approaches for object detection. Among the conventional detection methods, Viola-Jones [18] and Histograms of Oriented Gradients (HOG) [19] are widely used detection approaches. Viola-Jones method is aimed at fast object detection with accuracy comparable to relevant existing methods. The method uses a sliding window over an image to calculate Haar-like features under different location and scales which afterwards are used by a classifier. They achieved 15 FPS on 700 Mhz Pentium III detecting 77.8% objects with five false positives on MIT dataset containing faces. The preliminary condition for this method is training the classifier using the data which is close to operating scenes. Dalal and Triggs in their work [19] studied parameters of HOG features to detect pedestrians. They showed that HOG features reduce false positive rate in comparison to Haar-based detectors on more than one order of magnitude. For a long time it was one of the most accurate approaches [20]. However, HOG requires high computational efforts due to exhaustive search that makes the method not applicable for the real-time deployment on modern low-performance Systems on Chips (SoCs). Beside the above mentioned methods, there are several techniques which can be used for the given task, but they all experience difficulties on low resolution frames [21].

In recent years, machine learning (ML) methods gained wide popularity due to the increase of GPU computing power. The most representative ML methods in the field of object classification and recognition are Regional-based Convolutional Neural Networks (RCNN) and You Look Only Once (YOLO). RCNN exploits a concept of image segmentation to extract regions of interest (ROI) which are passed subsequently to the CNN [22]. The RCNN method the introduced ROI computation technique, which accelerated the object classification by narrowing down a set of image regions to be processed by

CNN. Their selective search algorithm can successfully segment 96.7% objects in the Pascal VOC 2007 dataset. The RCNN became the basis for Fast RCNN and Faster RCNN methods which attempt to solve the problem of slow RCNN operation. In contrast to the sliding window and ROI proposal techniques, YOLO has brought the concept of processing the entire image by a neural network. The neural network processes each region simultaneously taking into account the contextual information of an object [23]. YOLO achieved a mean average precision (mAP) of 57.9% on VOC 2012 dataset that is comparable to RCNN. Moreover, YOLO can process images at 45 FPS using GPU that is unattainable by RCNN. Further modifications YOLOv2 and YOLOv3 are designed to improve the detection accuracy. These methods have impressive performance, while keeping high accuracy. However, they remain too slow for real-time operation on embedded devices [4]. Some implementations that attempt to combine deep-learning approaches with Background Subtraction (BS) [1]. Despite this, they require a GPU module for real-time detection.

Thermal imagery in conjunction with computer vision offers some methods for pedestrian detection. Object candidates are extracted using signal from thermal imager only or by fusing data from camera and thermal imager. Such approaches are relatively accurate; however, this solution is sensitive to lighting changes and require an expensive (in comparison to a camera) thermal imager [24–27].

3. Method Description

The detection process of the proposed method can be divided into stages as shown in Figure 1.

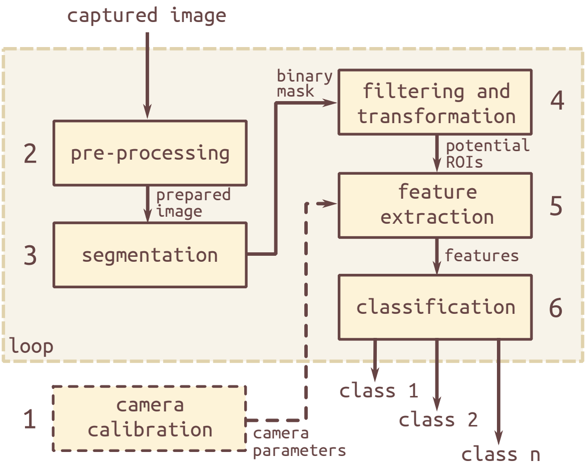


Figure 1. Block-diagram for the proposed detection algorithm.

1. As a result of camera calibration, the preliminary data, such as internal and external camera parameters as well as lens distortion coefficients are obtained. The calibration can be done using well-known methods, such as the algorithm by Roger Y. Tsai [28].
2. The *pre-processing* block receives images/video captured using a camera mounted on a static object and directed at the control area. Adaptive histogram equalization of contrast is applied to the input images to improve the results of the following background subtraction in nighttime scenes. The contrast improvement technique and its parameters can be adjusted depending on the concrete exploitation scenario. Lens distortions can be corrected either at the current stage, applying correction to the entire image, or at the stage of *feature extraction* to pixels of interest only to improve processing speed. The correction is performed using coefficients obtained via camera calibration.
3. The taken image is fed to a *background subtraction* method for the purpose of object segmentation resulting in a binary mask containing moving objects. The *background subtraction* is chosen due to its relatively low requirements for computational power [29]. The method has some limitations, such as the inability to segment static objects, which however, do not contradict imposed requirements. Background subtraction methods build an average background model, which is regularly updated

to adapt to the gradual changing of lighting and scene conditions [30]. The commonly used methods are based on Gaussian Mixture Models, such as GMG, KNN, MOG and MOG2 [31–35]. The MOG2 method provided by the *OpenCV library* was used in the software implementation of the proposed detection method.

4. At the stage of *filtering and transformation*, the filtering is performed via a morphological opening on the binary image that includes procedures of erosion and dilation following each other with the same kernel. Erosion removes noises caused by insignificant fluctuations and lighting changes, while the dilation increases an area of ROIs to compensate erosion shrinking [36]. Additional dilation can be performed in order to join the sub-regions which were divided by the segmentation that happened due to insufficient illumination. Primary features such as a contour, bounding rectangle and image coordinates of ROI are extracted via standard methods of *OpenCV*, which are subsequently computed through the Green's theorem [37]. The objects with frame borders' intersections are ignored and not considered in the following stages due to misleading geometrical form which is the defining criterion of detection.
5. Information about the camera, obtained during calibration, allows transforming initial ROI's primary parameters into features *estimated width*, *height* and *contour area* which represent the object's geometrical form in the real world (3D scene) independent of object distance and scene parameters. The detailed explanation of the stage *feature extraction* is provided in Section 4.
6. The trained logistic regression classifier finally makes the decision whether the candidate belongs to a specific group based on obtained features. The stage of *classification* is discussed in Section 5.

4. Proposed Feature Extraction Method

Geometrical parameters of an object in the image plane (2D image) are defined by the perspective transformation which takes into account camera parameters, coordinates and dimensions of the object in the real world. The camera parameters incorporate such scene properties as height and angles of camera installation as well as some internal camera specifications. The object projection is also affected by perspective distortion which depends on object coordinates in the 3D scene. The *feature extraction* method exploits camera parameters and real-world object coordinates to approximately estimate dimensions of a candidate in the 3D scene.

It is important to note that the feature extraction method is not aimed at the precise measurement of object dimensions, rather at finding unique features which make the differentiation of object types possible on the classification stage. The obtained geometrical parameters are considered *estimated* since they can differ from the dimensions of an object in the real world.

4.1. Object Distance Obtaining

Object distance obtaining is a primary step for the estimation of the candidate's dimensions. The method transforms the image plane coordinates into a real-world distance to an object. For that, a number of pre-requirements must be met:

- an object intersects a ground surface in real world in at least one point, and
- the lowest object point in image lies on a ground surface in real world
- known intrinsic (internal) camera parameters such as equivalent focal length, physical dimensions of the sensor and image resolution
- known extrinsic (external) parameters which include tilt angle and real-world coordinates of the camera relative to the horizontal surface
- known distortion coefficients of the camera lenses

To compensate for the camera lens distortion, the correction procedure is performed as a first step. The pixels of interest are corrected taking into account radial and tangential lens distortion factors using Equation (1). Each pixel has coordinates u, v in image plane along horizontal and vertical axis correspondingly.

$$\begin{aligned} r^2 &= u'^2 + v'^2 \\ u &= u'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 u' v' + p_2(r^2 + 2u'^2) \\ v &= v'(1 + k_1 r^2 + k_2 r^4) + p_1(r^2 + 2v'^2) + 2p_2 u' v' \end{aligned} \tag{1}$$

where u', v' — coordinates of the distorted point in image plane along horizontal and vertical axis in pixels
k_1–k_3 — radial distortion coefficients
p_1, p_2 — tangential distortion coefficients

According to Figure 2, one of the bottom pixels of an object O is a point p_b which has some coordinates u_r, v_b in the image plane with resolution $u_{max} \times v_{max}$. An image principal point C has coordinates u_c, v_c which usually correspond to a central pixel of an image.

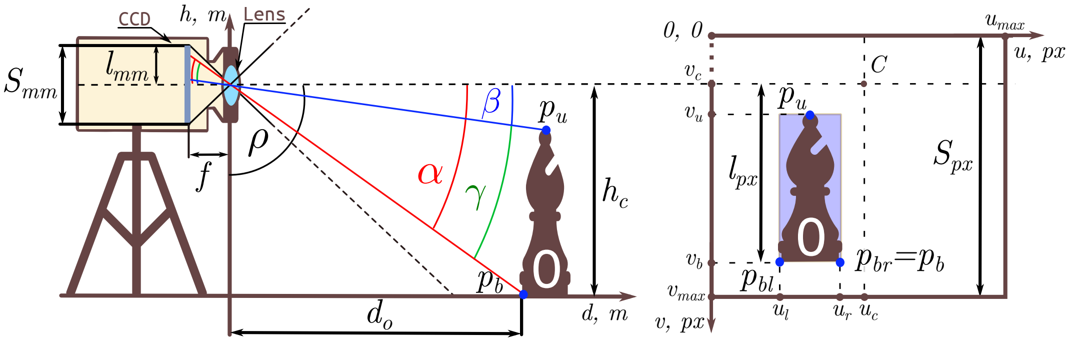


Figure 2. Object distance estimation scheme: real-world (**Left**) and image plane (**Right**) scenes.

With a distance l_{px} between coordinates v_b and v_c, it is possible to find this length in millimeters:

$$l_{mm} = S_{mm} \cdot \left(\frac{v_b}{S_{px}} - \frac{1}{2} \right) \tag{2}$$

where v_b — bottom coordinate of the object in image plane in pixels
S_{mm} — height of a camera sensor in mm
S_{px} — total amount of pixels along the v-axis, $S_{px} = v_{max}$

Using l_{mm} from Equation (2), as a first cathetus and focal length f as a second one, the angle α is calculated as:

$$\alpha = \arctan\left(\frac{l_{mm}}{f}\right) + 90^\circ - \rho = \arctan\left(\frac{S_{mm}}{f} \cdot \left(\frac{v_b}{S_{px}} - \frac{1}{2}\right)\right) + 90^\circ - \rho \tag{3}$$

where f — focal length of a camera in mm
ρ — incline of a camera with respect to the ground in degrees

The *distance* to the object is obtained using the height of a camera h_c and calculated angle α between the lowest point of the object **O** and the horizon line of the camera. The distance is calculated through the following equation:

$$d_o = \frac{h_c}{\tan(\alpha)} \tag{4}$$

4.2. Object Height Estimation

The object height is estimated based on known camera parameters and distance to the object. As shown in Figure 2, the pixels p_b and p_u are the lowest and the highest points of the object O. The angle between these pixels is calculated through Equation (3).

$$\gamma = \alpha - \beta = \arctan\left(\frac{S_{mm}}{f} \cdot \left(\frac{v_b}{S_{px}} - \frac{1}{2}\right)\right) - \arctan\left(\frac{S_{mm}}{f} \cdot \left(\frac{v_u}{S_{px}} - \frac{1}{2}\right)\right) \tag{5}$$

where v_b, v_u — the lowest and the highest object coordinates along the v-axis in pixels

The angle η which is shown in Figure 3 can be found using Equation (5). Angle $\delta = \arctan\left(\frac{d_o}{h_c}\right)$, hence, $\varepsilon = 180° - \eta - \delta$. With a triangle's values of angles and length of one side, the *estimated height* of the object can be calculated as:

$$h_e = h_o + h_{err} = h_1 \cdot \frac{\sin(\eta)}{\sin(\varepsilon)} \tag{6}$$

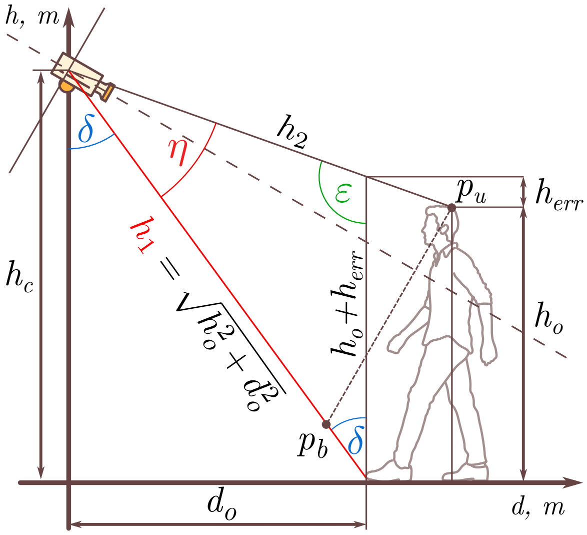


Figure 3. Object height estimation scheme.

The result of Equation (6) is a real height of the object and non-negative error of estimation caused by the geometrical form of the object and camera direction.

4.3. Object Width Estimation

Object width estimation is performed by applying inverse perspective transformation to reconstruct real-world coordinates from image plane points. Inverse perspective transformation to project real-world coordinates into the image plane can be found via the Pinhole Camera Model [38] which is described by Equation (7):

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \cdot [R|t] \cdot W = \begin{bmatrix} f_x & 0 & u_c \\ 0 & f_y & v_c \\ 0 & 0 & 1 \end{bmatrix} \cdot \left[\begin{array}{ccc|c} 1 & 0 & 0 & t_1 \\ 0 & \cos\theta & -\sin\theta & t_2 \\ 0 & \sin\theta & \cos\theta & t_3 \end{array}\right] \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{7}$$

where u, v — coordinates of projected point in image plane in pixels
K — an intrinsic camera matrix, represents relationship between camera coordinates and image plane coordinates. It includes focal lengths f_x, f_y in pixels and central coordinates of image plan u_c, v_c in pixels
$[R|t]$ — an extrinsic camera matrix, describes a camera location and direction in world and includes matrix R rotated about X axis by an angle of θ and translation vector t_1—t_3. The values t_1—t_3 can be equal to zero if camera translation is incorporated in coordinates X, Y, Z
W — real-world coordinates X, Y, Z of a point in meters which are expressed in a homogeneous form

The inverse perspective transformation can be expressed based on direct transformation (7) as:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = [R|t]^{-1} \cdot K^{-1} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cdot z_{cam} \tag{8}$$

where K^{-1} — an inverted intrinsic camera matrix
$[R|t]^{-1}$ — an inverted extrinsic camera matrix
z_{cam} — a scaling factor which is the distance to the object in camera coordinates in meters

Image plane point with coordinates u, v is supposed to be preliminary corrected via the expression 1. Being interested in z_{cam} only and considering rotation about X axis, the z_{cam} is found through simplification of $[R|t] \cdot W$, which results in:

$$z_{cam} = Y \cdot \sin(\theta) + Z \cdot \cos(\theta) \tag{9}$$

According to Figure 2, parts of the equation are the camera height $Y = h_c$, the object distance $Z = d_o$ and the camera rotation angle $\theta = 90^\circ - \rho$.

Thus, the inverse transformation (Equation (8)) allows estimating the coordinate X of the object point in 3D space which corresponds to image plane projection with coordinates u, v. The estimation is based on the assumption that the lowest point of the object lies on the ground surface.

To estimate the real-world object width, the most left $p_{bl}(u_l, v_b)$ and most right $p_{br}(u_r, v_b)$ bottom points of the object's bounding rectangle in the image are reconstructed in 3D scene. The reconstructed p_{bl} and p_{br} have X_l, Y_l, Z_l and X_r, Y_r, Z_r real-world coordinates correspondingly. The *estimated width* of the object is then found as the distance between these points. Since bottom coordinates Y and Z of object bounding rectangle are the same, there is only X coordinate presented in the calculation:

$$w_e = w_o + w_{err} = \sqrt{(X_r - X_l)^2} \tag{10}$$

5. Classification

The description of the classification process is presented in this section. A decision about lamp switching is performed for specific types of objects based on their extracted features. The type of an object is estimated by trained logistic regression classifier. The training is performed by using synthetically generated parameters of objects.

5.1. Features

The objects are quantified by several features which describe geometrical parameters of the object; in particular, *estimated width, height, contour area, distance* and environmental parameters of the scene—*camera angle* and *camera height*.

The *estimated contour area* (ca_e) of the object in the real-world scale is hereby calculated through the following relation:

$$\begin{gathered}\frac{h_{px} \cdot w_{px}}{ca_{px}} = \frac{h_e \cdot w_e}{ca_e} \\ ca_e = \frac{ca_{px} \cdot h_e \cdot w_e}{h_{px} \cdot w_{px}}\end{gathered} \tag{11}$$

where h_{px}, w_{px} — height and width of object's bounding rectangle in image plane in pixels
ca_{px} — a contour area of the object in pixels

The object *distance, camera height* and *angle* features are designated to compensate error values (h_{err}, w_{err}) from the feature obtaining stage (Equations (6) and (10)).

5.2. Object Classes

The classes of *pedestrian*, *group of pedestrians* (further referred as a *group*), *cyclist* and *vehicle* are expected in the frame of the street lighting systems. These classes can be also combined into a general *target group* class. Any other object, which does not belong to the classes of the *target group*, is treated as a class of *noise*. The general *target group* class is introduced for a binary classification between the *target group* and *noise*. The problem of false classification between classes of the *target group* is considered to be less significant than the *target group*/*noise* classification since all the classes of the *target group* are in need of illumination at nighttime.

In order to generate scenarios of the objects movement, the most probable minimal and maximal values of classes' dimensions in 3D space were chosen (Table 1).

Table 1. Assumed real-world dimensions of the *target group* classes [39–43].

Object Class	Target Group			
	Pedestrian	**Group**	**Cyclist**	**Vehicle**
Width, m	0.3–0.64	1.2–3	0.3–0.64	1.4–1.8
Height, m	1.15–2	1.5–2	1.5–1.9	1.4–2
Depth (step length), m	0.3–0.79	0.3–0.79	1.5–1.7	3.7–4.6

Environmental conditions and properties of the 3D scene, such as illumination, object material, camera point of view can distort geometrical parameters of objects being projected in the image. The distortions can be expressed in poor/false object segmentation. Furthermore, each class inserts inherent distortions.

Pedestrian class is influenced by non-expected behavior, e.g., jumping, squatting, etc. Due to non-rigid nature of *pedestrian*, it can change geometrical parameters during walking. The *group* class keeps all the traits of the *pedestrian*. Besides that, it is hard to highlight the thresholds due to varying numbers of members in the *group* class. In the current paper, between two and four pedestrians are assumed to be in the group class.

There are distortions which are common for *cyclist* and *vehicle*. In some nighttime scenes, an object and a spot produced by its lamps can be overlapped, therefore, the division of a candidate's region should be performed. The division method proposes splitting an object into parts based on horizontal and vertical region histograms [26], whereas the criteria of splitting are *extent*, *height to width ratio* and *estimated contour area*. The *feature extraction* is completed again after the division to update object's features.

The motion blur effect can appear due to low shutter speed and high object velocity. Detection of large vehicles can be problematic due to filtration of the objects crossing the frame border (described in Section 3). Also, large dimensional vehicles can cause an error of estimation under certain angles.

5.3. Data Acquisition

The classifier was trained using a combination of real-world and synthetic data. Synthetic data has been added to the training since it provides a simplistic way to generate a wide range of possible movement scenarios and scenes, which is much bigger than the existing empirical data can to cover. Such an approach allows generating any possible events with high authenticity without gathering redundant amounts of real-world data.

For each of the described classes, the 3D models (Figure 4) were generated based on geometrical parameters ranges provided in Table 1. The 3D models of the *pedestrian* are generated in various postures such as standing, walking and running. The instance of the *group* model is a combination of the *pedestrian* models located next to each other. The *cyclist* models are expressed in different postures of a driver, while the *vehicle* varies in car models.

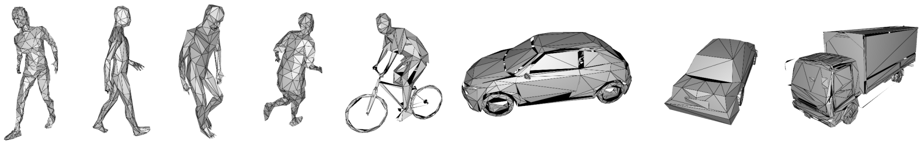
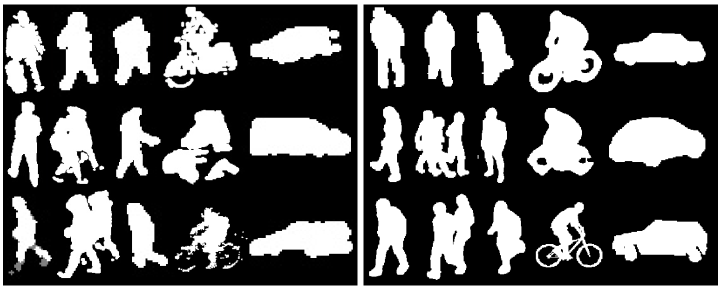


Figure 4. Examples of used 3D objects.

In order to gather features, the 3D objects are projected into a 2D image plane on the range of camera operation angles (0–80°) and heights (2–10 m) using the Pinhole Camera Model transformation. The 3D objects are rotated around to synthesize different movement angles. To increase the authenticity of synthetic data, a number of distortion techniques are applied, in particular, the motion blur effect has been added to simulate distortions of a real camera. The real-world data (Figure 5a) is used to validate the synthetic data. Obtained objects contours (Figure 5b) are processed by the feature extraction method.



(**a**) Real-world data (**b**) Generated data

Figure 5. Object contours.

The noises are generated numerically as vectors of features out of ranges declared in Table 1.

5.4. Model of Classifier

The logistic regression method is used as a statistical model for multi-class classification because this model does not require too many computational resources. It is easy to tune a target function, and it is easy to regularize [8]. To separate target classes from the noise which surrounds them, a circular decision boundary with the polynomial order of 2 has been used. The data used for the logistic regression model training is shown in Figure 6.

The data has been generated synthetically for the range of camera angles, heights (mentioned in Section 5.3) and target objects parameters, according to Table 1. The features of the *pedestrian*, *group* and *cyclist* classes are distributed densely which makes them hard to distinguish under the specific direction vectors that lead to misclassification. The geometrical parameters of the *vehicle* make this class easy to differentiate. As shown on the plots, the *vehicle* features are distanced from the other classes.

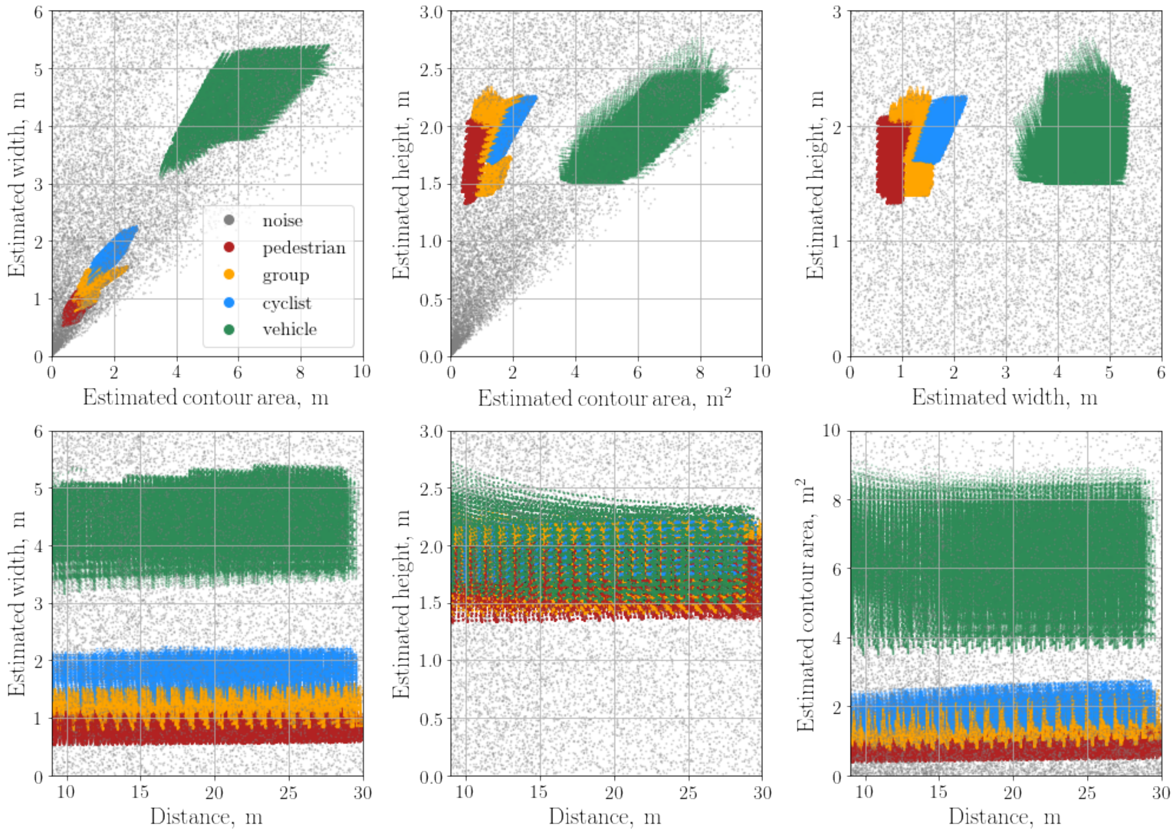


Figure 6. An example of synthetically generated features for real-world scene: angle—22°, height— 3.1 m.

6. Evaluation Results

The results of detection system tests are presented in the current section. It contains testbed description and evaluation of classifier performance on real-world datasets.

6.1. Testbed

Several testbeds with different camera installation parameters were built to collect real-world material. The parameters of the gathered datasets are shown in Table 2. Three cameras were used to prove that the feature extraction mechanism can handle different intrinsic camera parameters. The real-world data has been gathered in four street scenarios (Figure 7):

A **The nighttime parking area** includes moving objects on distances up to 25 m. The scene per image contains up to three pedestrians, two groups and one cyclist with and without a flashlight. The web-camera's IR-cut filter has been removed. The camera is equipped with an external IR emitter.

B Another **nighttime parking area** with up to six pedestrians, three groups, two cyclists with and without a flashlight and one vehicle are present on distances up to 25 m. The camera with removed IR-cut filter is equipped with an external IR emitter.

C **Nighttime city street** includes up to four pedestrians, one group, two cyclists with flashlight and one vehicle on distances up to 50 m. The camera has an embedded IR illumination.

D **Daytime city street** includes moving objects on distances up to 50 m, in particular, up to six pedestrians, three groups, three cyclists and two cars.

Table 2. Datasets parameters.

Scene	A	B	C	D
Times of day	Night			Day
Description	Parking 1	Parking 2	City street	City street
Camera	Logitech C920 HD	Microsoft HD-3000	Gadinan Full HD IP cam	
Height, m	3	3.1	5	
Angle, °	13	22	17	
Resolution	424 × 240	320 × 240	424 × 240	
Focal length, mm	3.67	0.73	3.6	
Sensor dim., mm	4.8 × 3.6	0.7 × 0.52	3.45 × 1.94	
Objects distance, m	up to 25	up to 25	up to 50	
Noises	4575	13,724	2474	2258
Pedestrians	1364	6705	326	1990
Groups	108	473	218	928
Cyclists	335	134	183	730
Vehicles	—	26	238	398

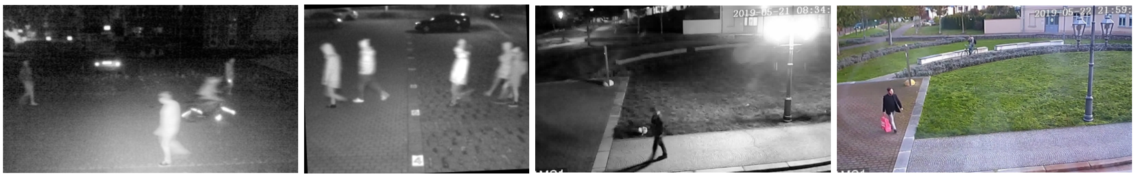

(**a**) (**b**) (**c**) (**d**)

Figure 7. Real-world scenes.

6.2. Classifier Precision

The datasets which are close to expected exploitation scenarios were used for classifier's precision evaluation.

The datasets described in Section 6.1 were annotated and processed by the proposed feature extraction method. The real-world features for all datasets are shown in Figure 8. The real-world features have similar distribution patterns to the synthetic ones (Figure 6); however, the *group* class is less distinguishable.

Annotated scenes were divided into nighttime and daytime which are classified by the trained logistic regression model. The result of classification can be summarized in corresponding confusion matrices.

Binary classification. According to Figures 9a,b (the darker color, the greater probability), the false negative rates for the *target group* are noticeably high at nighttime that at daytime (**0.24** vs. **0.04**). Such error rates can be explained by poor segmentation of the *target group* on long distances due to insufficient IR illumination. The false positive errors for the *target group* (**0.03** for nighttime and **0.05** for daytime scenes) are mainly caused by the interpretation of the object's lights as a target object, especially, by reflections of vehicle's lights from the ground and some vertical surfaces (i.e., windows). The daylight scene illumination provides a better BS segmentation, which resulted in relatively high true positive rates of **0.96** for the *target group*. The obtained false negative error rate of **0.24** (Figure 9a) and false positive error rate of **0.05** (Figure 9b) for binary classification do not exceed predefined requirements.

Multi-class classification. The problem of misclassification inside the *target group*, in particular, for *group* and *cyclist* remains relevant for both nighttime and daytime scenes, as shown in Figure 9c,d. Achieved true positive rates of **0.48** for the cyclist and **0.44** for the group at night show that the precision of the classifier is not sufficient to differentiate these classes. This problem appears when objects move under certain angles which do not allow obtaining distinguishable features. The presented classifier accuracy takes into consideration controversial scenes, where an object is partially present in the frame. A high false positive rate (**0.32**) for the *pedestrian* towards the *group* is related to intersecting parameters of classes which are shown in Table 1. The *vehicle* has the lowest false positive rates towards any other class of the *target group* due to infrequency of 2D projection, wherein features of the *vehicle* and any other class of the *target group* are overlapped. Unfortunately, the *vehicle* is interpreted as *noise* in some projections that result in a false positive rate of **0.25**. This rate can be reduced by extending the geometrical parameters of the *vehicle* from Table 1. However, this leads to precision reduction of the system in general. Since the *cyclist* and the *group* have similar ranges of geometrical parameters, their features can be misinterpreted by the classifier, as described in Section 5.4.

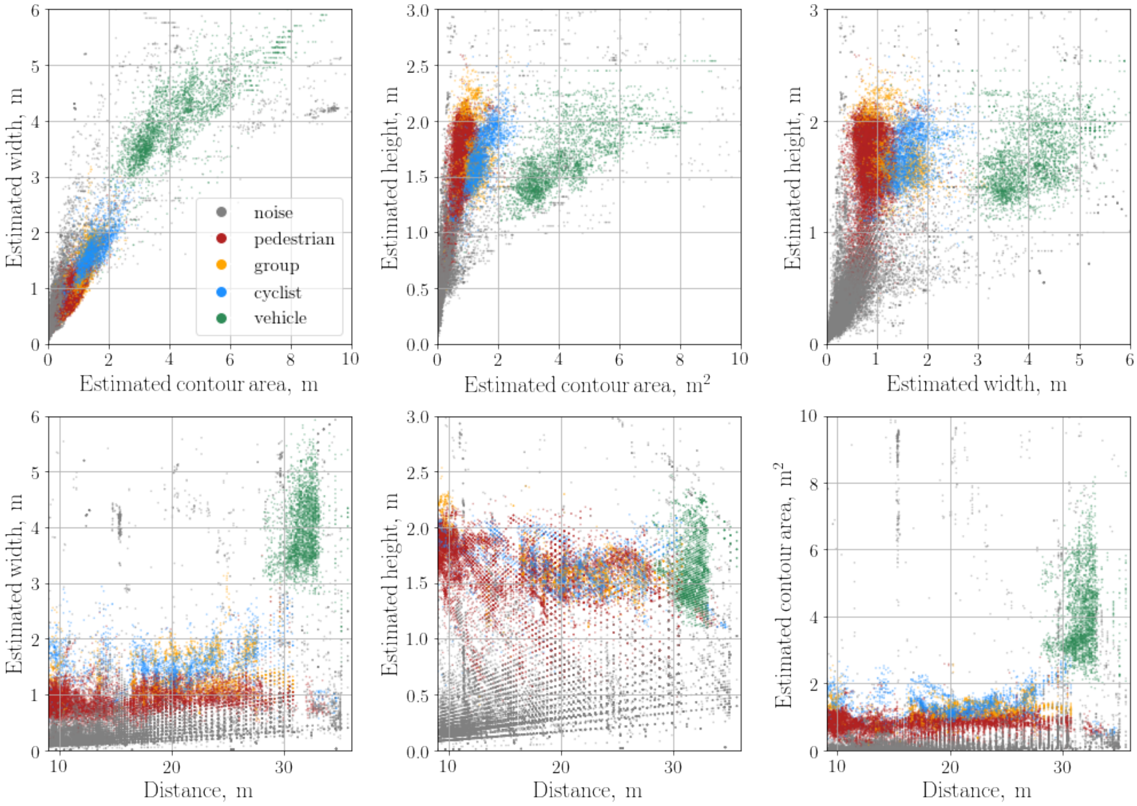
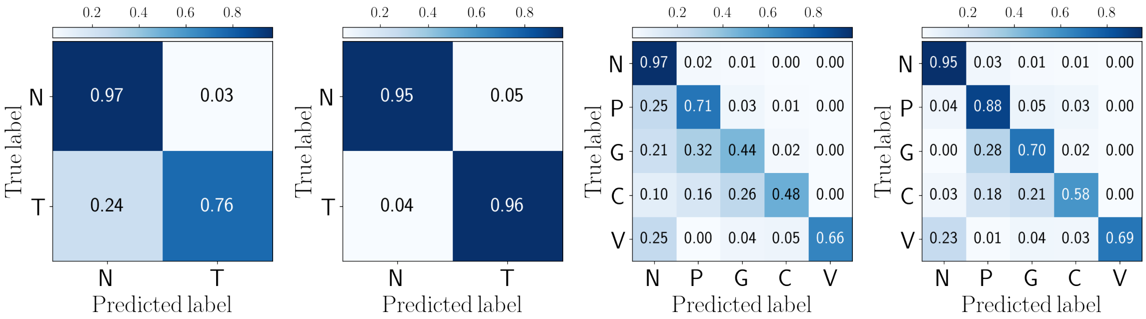


Figure 8. Real-world features for datasets.



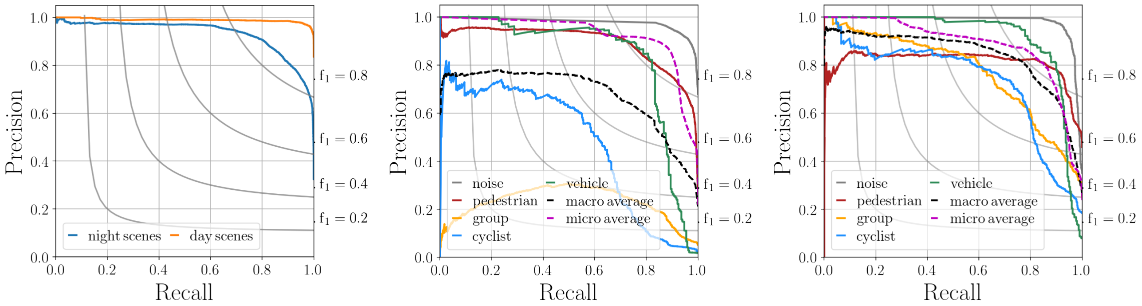
(**a**) Night binary (**b**) Day binary (**c**) Night multi-class (**d**) Day multi-class

Figure 9. Confusion matrices, where N—*noise*, P—*pedestrian*, G—*group*, C—*cyclist*, V—*vehicle*, T—*target group*.

Precision-Recall curves. Overall classifier performance can be evaluated through the precision-recall curves, shown in Figure 10. Considering a task of binary classification between *target group* and *noise* (Figure 10a), the curve for the daytime scene shows high average precision since the *noises* are less distinguishable than the *target objects*. The amount of *noises* is limited in daytime scenes and shapes of *noises* cannot produce features belonging to the *target group*. The mAP value for daytime is **0.99**, while the same parameter for the nighttime dataset is equal to **0.93** which is acceptable in the targeted application of SmartLighting.

The micro average precision-recall curve for nighttime scenes (Figure 10b) is too optimistic due to classes dis-balance, more precisely, because of numerous *noise* instances at night. Therefore, macro metrics were chosen for classifier evaluation. The macro mAP for all classes during the night equals to **0.67**. Considering precision-recall curves inside the *target group*, the highest mAP of **0.8** has been achieved by the *pedestrian* classifier, the lowest **0.36**-by the *group* classifier. Such a low rate of the *group* classifier is caused by the formless nature of the class.

During the day all precision-recall curves of the *target classes* are grouped (Figure 10c). Also, the micro and macro average precision-recall curves are close to each other due to the better balancing of classes. The macro mAP for all classes during the day is **0.84**.



(**a**) Binary (*target group*/*noise*) (**b**) Multi-class nighttime (**c**) Multi-class daytime

Figure 10. Precision-recall curves.

7. Conclusions

The developed detection technique satisfies the imposed requirements of the SmartLighting system for which it was designed. It is able to operate on low-cost computers, e.g., with Cortex A8 processor, due to its computational power efficacy. The detection method meets the established requirements for false positive/negative error rates.

The algorithm has been tested on different single-board computers using real-world datasets. The achieved frame rate is shown in Table 3. The algorithm is implemented as single-thread, therefore only a portion of the CPU resources is used on multi-core boards.

The detection algorithm showed a sufficient frame rate on Orange Pi Zero H5 and Raspberry Pi 3, satisfying the requirements of the SmartLighting system guaranteeing a reliable detection of fast-moving objects.

The applicability of the detection algorithm has been proven on real-world datasets which include low-resolution and gray-scale images. The elaborated framework for synthetic data generation allows obtaining a variety of scenes that quite accurately approximate real-world scenarios, minimizing time spending for data collection.

Despite of achieved efficiency, the current method has several challenges to overcome. It requires a preliminary camera calibration, specific object position with respect to the ground, and unsloped ground in the scene. Moreover, the method can be applied in environments with static background only. The system makes a decision based on the geometrical shape of an object that leads to the misclassification of candidates with similar geometrical features.

Table 3. Achieved FPS using different single-board computers.

Board	**FPS**			**CPU**	**SoC**
	Min	**Mean**	**Max**		
Beaglebone Black	4	5	7	Cortex-A8 1 GHz	AM3358
Orange Pi PC	10	13	15	4 × Cortex-A7 1.4 GHz	Allwinner H3
Orange Pi Zero H5	25	28	35	4 × Cortex-A53 1.4 GHz	Allwinner H5
Raspberry Pi 3 Model B+	30	35	40	4 × Cortex-A53 1.4 GHz	Broadcom BCM2837B0

The proposed algorithm outperforms existing approaches in terms of FPS and CPU usage that is shown in Table 4. The achieved false positive rate is comparable to analogies; however, the evaluation is performed using different datasets. Limited resources of embedded devices make other methods unusable to operate in real-time mode for the given application. Nevertheless, the accuracy of other methods is higher.

Table 4. Performance parameters of object detection algorithms on Raspberry Pi 3 B [4,44–47].

Algorithm	**Our Method**	**HOG**	**Viola-Jones**	**SSD GoogleNet**	**L-CNN**	**YOLO**
FPS	35	0.3	1.82	0.39	0.39	1.25
CPU, %	25	93	76.9	84.7	75.7	-

The performance of the detection system can be estimated in comparison to the other detection approaches which are applicable in the SmartLighting system. Among the computer vision techniques, one of the most straightforward alternatives to the presented method is the usage of BS with subsequent evaluating sizes of contour areas in the foreground mask. With this approach, the object is considered to be detected, when the size of its contour area exceeds a pre-defined threshold. Applying this method to the presented nighttime datasets, the selected threshold of 5%, from overall image resolution, has shown the true positive rate of 0.2. Such a true positive rate means that the lamp is switched on only for 20% target objects. The threshold of 0.5%, however, resulted in excessive sensitivity making the false positive rate equal to 0.9. That makes the lighting system energy inefficient. In contrast to trivial thresholding, the developed method takes into account distance and perspective distortions that allow detecting desired objects more precisely and therefore much more energy efficient.

8. Future Work

Though the proposed approach solves the problem of object detection in street environment, there is still much room to improve the described system. The system can be modified in terms of performance, accuracy, and simulation methods.

8.1. Dynamic Error Compensation

The binary classification decision can be made using a sequence of frames to improve the true positive detection rate. The true positive rate of an object detection $P(A)$ within n frames can be calculated using Equation (12) which is based on a Bernoulli scheme. This equation allows finding the probability of an event A that the object has been detected on at least $(n/2)+1$ frames, which is more than half.

$$P(A) = \sum_{k=\frac{n}{2}+1}^{n} \binom{n}{k} p^k q^{n-k} \tag{12}$$

where A — an event that the object has been detected on at least more than half of the frames
p — a true positive rate for object detection on a single frame
q — a false negative rate for object detection on a single frame
n — an amount of frames in the detection sequence
k — the least number of frames where the object is supposed to be detected

The $P(A)$ grows with an increase of n only when p is greater than 0.5. The accuracy of the system can be significantly increased through this property. For example, the per-frame true positive detection rate for the nighttime dataset of **0.76** (Figure 9a) increases to **0.90** within the time window, where the time window is one second that corresponds to five frames for BeagleBone Black from Table 3.

8.2. Tracking System

Approaches of dynamic error compensation require tracking an object between frames. To calculate $P(A)$, it is necessary to be confident that the frame-wise probability belongs to the same object which is present on multiple frames. Moreover, the tracking system can improve the true positive rate since it increases a prior probability of the object presence in the next frame. The tracking system is supposed to satisfy requirements for performance and reliability in the real-time domain. Such a system is an important subject for future work.

8.3. Segmentation Improvement

To compensate disadvantages of segmentation, a technique for the reunification of objects regions can be applied. The reunification of the object can be based on finding a skeleton structure of the object.

8.4. Simulation Framework

Another important objective is the development of a framework for synthetic data generation to construct dynamic virtual reality scenes. The framework is supposed to simulate real-world environments more precisely taking into account object deformation during movement, weather, illumination, undesired objects and distortion effects of the camera.

Author Contributions: Conceptualization, I.M., K.K., I.C. and E.S.; methodology, I.M. and K.K.; software, I.M. and K.K.; Writing—Original draft, I.M. and K.K.; Writing—Review and editing, I.C., E.S. and A.Y.; supervision, I.C. and E.S.; project administration, I.C. and E.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Scholarsip of the State of Saxonia Anhalt and Volkswagen Foundation grant number 90338 for trilateral partnership between scholars and scientists from Ukraine, Russia and Germany within the project CloudBDT: Algorithms and Methods for Big Data Transport in Cloud Environments.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
RCNN	Region Based CNN
ML	Machine Learning
HOG	Histograms of Oriented Gradients
SoC	System on Chip
PIR	Passive Infra-Red
GMM	Gaussian Mixture Model
KNN	K-Nearest-Neighbor
MOG	Mixture of Gaussians
YOLO	You Look Only Once
BS	Background Subtraction
GPU	Graphics Processing Unit
US	Ultra-Sonic
RW	Radio Wave
SoC	System on Chip
IOT	Internet of Things
IR	Infrared
FPS	Frames Per Second
mAP	mean Average Precision

References

1. Kim, C.; Lee, J.; Han, T.; Kim, Y.M. A hybrid framework combining background subtraction and deep neural networks for rapid person detection. *J. Big Data* **2018**, *5*, 22. [CrossRef]
2. Siemens, E. Method for lighting e.g., road, involves switching on the lamp on detection of movement of person, and sending message to neighboring lamps through communication unit. Germany Patent DE102010049121 A1, 2012.
3. Maolanon, P.; Sukvichai, K. Development of a Wearable Household Objects Finder and Localizer Device Using CNNs on Raspberry Pi 3. In Proceedings of the 2018 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), Chonburi, Thailand, 14–16 December 2018; pp. 25–28. [CrossRef]
4. Yang, L.W.; Su, C.Y. Low-Cost CNN Design for Intelligent Surveillance System. In Proceedings of the 2018 International Conference on System Science and Engineering (ICSSE), New Taipei, Taiwan, 28–30 June 2018; pp. 1–4, ISSN: 2325-0925, 2325-0909. [CrossRef]
5. *Speed and Speed Management*; Technical Report; European Commission, Directorate General for Transport: Brussel, Belgium, 2018.
6. *La Center Road Standards Ordinance. Chapter 12.10. Public and Private Road Standards*; Standard, La Center: Washington, DC, USA, 2009.
7. *Code of Practice (Part-1)*; Technical Report; Institute of Urban Transport: Delhi, India, 2012.
8. Lim, T.S.; Loh, W.Y.; Shih, Y.S. A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. *Mach. Learn.* **2000**, *40*, 203–228. [CrossRef]
9. Yavari, E.; Jou, H.; Lubecke, V.; Boric-Lubecke, O. Doppler radar sensor for occupancy monitoring. In Proceedings of the 2013 IEEE Topical Conference on Power Amplifiers for Wireless and Radio Applications, Santa Clara, CA, USA, 20 January 2013; pp. 145–147. [CrossRef]
10. Teixeira, T. A Survey of Human-Sensing: Methods for Detecting Presence, Count, Location, Track and Identity. *ACM Comput. Surv.* **2010**, *5*, 59–69.
11. Panasonic Corporation. *PIR Motion Sensor. EKMB/EKMC Series*; Panasonic Corporation: Kadoma, Japan, 2016.
12. Digital Security Controls (DSC). DSC PIR Motion Detector. Digital Bravo 3. 2005. Available online: https://objects.eanixter.com/PD487897.PDF (accessed on 1 December 2019).

13. Canali, C.; De Cicco, G.; Morten, B.; Prudenziati, M.; Taroni, A. A Temperature Compensated Ultrasonic Sensor Operating in Air for Distance and Proximity Measurements. *IEEE Trans. Ind. Electron.* **1982**, *IE-29*, 336–341. [CrossRef]
14. Mainetti, L.; Patrono, L.; Sergi, I. A survey on indoor positioning systems. In Proceedings of the 2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 17–19 September 2014; pp. 111–120. [CrossRef]
15. Matveev, I.; Siemens, E.; Dugaev, D.A.; Yurchenko, A. Development of the Detection Module for a SmartLighting System. In Proceedings of the 5th International Conference on Applied Innovations in IT, (ICAIIT), Koethen, Germany, 16 March 2017; Volume 5, pp. 87–94.
16. Pfeifer, T.; Elias, D. Commercial Hybrid IR/RF Local Positioning System. In Proceedings of the Kommunikation in Verteilten Systemen (KiVS), Leipzig, Germany, 26–28 February 2003; pp. 1–9.
17. Bai, Y.W.; Cheng, C.C.; Xie, Z.L. Use of ultrasonic signal coding and PIR sensors to enhance the sensing reliability of an embedded surveillance system. In Proceedings of the 2013 IEEE International Systems Conference (SysCon), Orlando, FL, USA, 15–18 April 2013; pp. 287–291. [CrossRef]
18. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 8–14 December 2001; Volume 1, pp. I–511–I–518. [CrossRef]
19. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893, ISSN: 1063-6919. [CrossRef]
20. Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *arXiv* **2019**, arXiv: 1905.05055.
21. Sager, D.H. Pedestrian Detection in Low Resolution Videos Using a Multi-Frame Hog-Based Detector. *Int. Res. J. Comput. Sci.* **2019**, *6*, 17.
22. Van de Sande, K.E.A.; Uijlings, J.R.R.; Gevers, T.; Smeulders, A.W.M. Segmentation as selective search for object recognition. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 1879–1886, ISSN: 2380-7504, 1550-5499, 1550-5499. [CrossRef]
23. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
24. Fernández-Caballero, A.; López, M.; Serrano-Cuerda, J. Thermal-Infrared Pedestrian ROI Extraction through Thermal and Motion Information Fusion. *Sensors* **2014**, *14*, 6666–6676. [CrossRef]
25. Bertozzi, M.; Broggi, A.; Fascioli, A.; Graf, T.; Meinecke, M. Pedestrian Detection for Driver Assistance Using Multiresolution Infrared Vision. *IEEE Trans. Veh. Technol.* **2004**, *53*, 1666–1678. [CrossRef]
26. Jeon, E.S.; Choi, J.S.; Lee, J.H.; Shin, K.Y.; Kim, Y.G.; Le, T.T.; Park, K.R. Human Detection Based on the Generation of a Background Image by Using a Far-Infrared Light Camera. *Sensors* **2015**, *15*, 6763–6788. [CrossRef] [PubMed]
27. Zhao, X.; He, Z.; Zhang, S.; Liang, D. Robust pedestrian detection in thermal infrared imagery using a shape distribution histogram feature and modified sparse representation classification. *Pattern Recognit.* **2015**, *48*, 1947–1960. [CrossRef]
28. Tsai, R. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE J. Robot. Autom.* **1987**, *3*, 323–344. [CrossRef]
29. Hardas, A.; Bade, D.; Wali, V. Moving Object Detection using Background Subtraction, Shadow Removal and Post Processing. *Int. J. Comput. Appl.* **2015**, *975*, 8887.
30. Piccardi, M. Background subtraction techniques: A review. In Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), Hague, The Netherlands, 10–13 October 2004; Volume 4, pp. 3099–3104. [CrossRef]
31. Trnovszký, T.; Sýkora, P.; Hudec, R. Comparison of Background Subtraction Methods on Near Infra-Red Spectrum Video Sequences. *Procedia Eng.* **2017**, *192*, 887–892. [CrossRef]
32. Godbehere, A.B.; Matsukawa, A.; Goldberg, K. Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. In Proceedings of the 2012 American Control Conference (ACC), Montreal, QC, Canada, 27–29 June 2012; pp. 4305–4312. [CrossRef]
33. Zivkovic, Z.; van der Heijden, F. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognit. Lett.* **2006**, *27*, 773–780. [CrossRef]

34. KaewTraKulPong, P.; Bowden, R. An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection. In *Video-Based Surveillance Systems: Computer Vision and Distributed Processing*; Remagnino, P., Jones, G.A., Paragios, N., Regazzoni, C.S., Eds.; Springe: Boston, MA, USA, 2002; pp. 135–144. [CrossRef]
35. Zivkovic, Z. Improved adaptive Gaussian mixture model for background subtraction. In Proceedings of the 17th International Conference on Pattern Recognition, 2004 (ICPR 2004), Cambridge, UK, 26 August 2004; Volume 2, pp. 28–31. [CrossRef]
36. Abbott, R.G.; Williams, L.R. Multiple target tracking with lazy background subtraction and connected components analysis. *Mach. Vis. Appl.* **2009**, *20*, 93–101. [CrossRef]
37. Strang, G.; Herman, E.J. Green's Theorem. In *Calculus*; OpenStax: Houston, TX, USA, 2016; Volume 3, Chapter 6.
38. Sturm, P. Pinhole Camera Model. In *Computer Vision: A Reference Guide*; Ikeuchi, K., Ed.; Springer: Boston, MA, USA, 2014; pp. 610–613. [CrossRef]
39. Ho, N.H.; Truong, P.H.; Jeong, G.M. Step-Detection and Adaptive Step-Length Estimation for Pedestrian Dead-Reckoning at Various Walking Speeds Using a Smartphone. *Sensors* **2016**, *16*, 1423. [CrossRef] [PubMed]
40. Buchmüller, S.; Weidmann, U. *Parameters of Pedestrians, Pedestrian Traffic and Walking Facilities*; ETH Zurich: Zürich, Switzerland, 2006. [CrossRef]
41. *Collision Mitigation System: Pedestrian Test Target, Final Design Report*; Technical Report; Mechanical Engineering Department, California Polytechnic State University: San Luis Obispo, CA, USA, 2017.
42. Blocken, B.; Druenen, T.V.; Toparlar, Y.; Malizia, F.; Mannion, P.; Andrianne, T.; Marchal, T.; Maas, G.J.; Diepens, J. Aerodynamic drag in cycling pelotons: New insights by CFD simulation and wind tunnel testing. *J. Wind Eng. Ind. Aerodyn.* **2018**, *179*, 319–337. [CrossRef]
43. Fintelman, D.; Hemida, H.; Sterling, M.; Li, F.X. CFD simulations of the flow around a cyclist subjected to crosswinds. *J. Wind Eng. Ind. Aerodyn.* **2015**, *144*, 31–41. [CrossRef]
44. Nikouei, S.Y.; Chen, Y.; Song, S.; Xu, R.; Choi, B.Y.; Faughnan, T.R. Real-Time Human Detection as an Edge Service Enabled by a Lightweight CNN. In Proceedings of the 2018 IEEE International Conference on Edge Computing (EDGE), San Francisco, CA, USA, 2–7 July 2018; pp. 125–129. [CrossRef]
45. Durr, O.; Pauchard, Y.; Browarnik, D.; Axthelm, R.; Loeser, M. Deep Learning on a Raspberry Pi for Real Time Face Recognition. In *Eurographics (Posters)*; The Eurographics Association: Geneve, Switzerland, 2015. pp. 11–12.
46. Vikram, K.; Padmavathi, S. Facial parts detection using Viola Jones algorithm. In Proceedings of the 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 January 2017; pp. 1–4. [CrossRef]
47. Noman, M.; Yousaf, M.H.; Velastin, S.A. An Optimized and Fast Scheme for Real-Time Human Detection Using Raspberry Pi. In Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Gold Coast, Australia, 30 November–2 December 2016; pp. 1–7. [CrossRef]