IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 August 2023, accepted 2 October 2023, date of publication 5 October 2023, date of current version 11 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3322199

APPLIED RESEARCH

A Binary-Directional Shape Descriptor for the Enhancement of Real-Time Automatic Headlight Control Function

PIOTR BOGACKI[1,2] **AND RAFAŁ DŁUGOSZ**[2,3]

[1]Faculty of Computer Science, Electronics and Telecommunications, Institute of Telecommunications, AGH University of Science and Technology, 30-059 Kraków, Poland
[2]Aptiv Services Poland S.A., 30-399 Kraków, Poland
[3]Department of Computer Science and Electrical Engineering, Faculty of Telecommunications, Bydgoszcz University of Science and Technology, 85-796 Bydgoszcz, Poland

Corresponding author: Piotr Bogacki (pbogacki@agh.edu.pl)

ABSTRACT Advanced Driver Assistance Systems (ADAS) are becoming more and more important in recent years. All relevant manufacturers equip their cars with various safety features and in some locations it is even mandatory to do so. One of such systems is Automatic Headlight Control (AHC) algorithm that prevents from accidental dazzling of other road users. Recognizing vehicle lights with high accuracy is crucial in this algorithm. Any improvement can result in the limitation of car accidents and dangerous situations on the road. The paper introduces a novel visual descriptor that describes previously binarized blob shapes of the analyzed objects. We also present our referenced AHC system that operates not only on the images but also on the set of other visual features. Adding new feature leads to the improvement of around 1.5 percentage point of vehicle lights classification accuracy (21% reduction of incorrectly classified light sources). It can also replace the already used features (four-time reduction from 32 to 8 feature values) and keep the classification accuracy at the similar level. The proposed solution has been tested with an extensive set of video sequences (total duration exceeding ten hours) under diverse weather conditions and from different types of roads. The proposed descriptor is shown to be far less computationally complex, and it is proved that it can outperform other state-of-the-art descriptors when analyzed individually.

INDEX TERMS Adaptive headlight control, automatic headlight control, computer vision, high beam control, night time vehicle detection, shape description.

I. INTRODUCTION

In recent years, in the automotive industry one can observe a dynamic development of active safety (AS) systems. For new vehicles offered to the market it became a standard to equip them with various functions belonging to ADAS systems. This trend is followed due to the fact that, as shown by numerous analyses, the human error is one of the main contributory factors in more than 90% of all motor vehicle crashes [1]. As a result, the EU Commission, for example, introduced regulations so that selected ADAS functions are mandatory equipment for all new cars sold in Europe [2]. Such solutions may, for example, warn the driver about unintentional lane departure or exceeding currently applicable speed limit. In more advanced cases, such systems may also take over the control over the vehicle. For instance, they can trigger immediate emergency braking or appropriately modify the car's trajectory in order to avoid an oncoming collision.

One of the examples when the car drivers may benefit from computer aid are systems capable of controlling headlights, while traveling at night. Such functions, denoted as AHC, are currently developed in many research centers around the world. It is caused by dynamic expansion of AS / ADAS sector.

Nowadays, the vehicles are equipped with various types of lights so that the road can be sufficiently illuminated under different weather and lighting conditions. Headlights,

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaosong Hu.

in general, operate in two modes. They can generate low and high beams. In some countries low beams are mandatory during the day. They are also useful at night when there is other traffic present. It is because they produce less illumination and thus do not generate a risk of blinding other road users. Otherwise, in case of the absence of other road users, the high beams should be activated. Unfortunately, a lot of drivers do not operate the lights properly. Some drivers use low beams for most of the time and turn on the high beams only occasionally. This leads to unsafe situations when the road is not lit enough or the high beams are switched on too late. As a consequence, the driver's visibility is reduced and the ability to react on time is decreased. This might happen, for example, in case of sudden dangerous events such as appearing pedestrians or obstacles on the road. On the other hand, some drivers keep the high beams on for too long or simply forget to turn them off. This leads to unintentional dazzling of other drivers with an excessive glare and may result in the distraction of driver's perception. In a worst case it might cause temporal blindness which can create hazard on the road.

In order to diminish such dangerous situations, an automatic control of the vehicle headlight intensity can be realized. This might decrease the chances that other drivers from oncoming and preceding cars will be subjected to excessive glare levels. The main task for such a system is to detect the vehicles ahead with high accuracy. It should be taken into account that the system is intended to operate mainly at night or under insufficient lighting conditions. Therefore, it is necessary to find special characteristics that can help with the detection of vehicles moving at night time. It is possible to analyze such conditions and additionally try to detect the whole vehicles with the use of deep learning methods [3], [4], [5]. However, this may be restricted due to the fact that typical appearance features such as color or shape are effective only during the day. Hence, the most reliable features for the detection of vehicles at night time are their lights. This covers the recognition of the headlights from the oncoming vehicles and the taillights of the preceding vehicles.

In this work, we present a new descriptor that can be used as one of the input features of the already used neural network [6] for application in high-beam control algorithms. The descriptor is able to characterize, in a relatively simple way, shapes of selected objects (e.g. light blobs) derived from the images captured by the camera mounted inside the vehicle. It was used in the task of distinguishing lights of other vehicles from lights of stationary objects around the road as well as various light reflections. Additionally, since the ADAS functions need to operate in real-time [7], we needed to optimize the proposed descriptor in terms of computational complexity.

In this paper, at the beginning we try to familiarize the reader with an overview of already existing approaches used in AHC algorithms. This includes both commercial and academic solutions. Then, we present the characteristics of our referenced AHC system which is based on two CNN classifiers. The input to these networks consists of images of two different dimensions (in order to focus on the light itself as well as on its surrounding) and a set of visual features calculated on the smaller images. However, the main accent is put on the new visual descriptor which was additionally provided as the input to the already used AHC classifier that improves the classification accuracy.

It is worth mentioning that for the evaluation of the proposed method, we use our own image dataset that covers various lighting and weather conditions. Our AHC system was designed in order to provide automatic high beam control in trucks. This use case justifies the need for the creation of our custom dataset. To the best of our knowledge, all publicly available datasets that could be used in vehicle lights detection contain images that were captured by cameras mounted in typical cars or traffic cameras being a part of an urban or highway infrastructure [8], [9]. Besides, the images from our dataset were acquired from the videos of total duration exceeding 10 hours which guarantees a good representation of potential light sources.

The main highlights and contributions presented in this paper consist of:

1) a novel Binary-Directional Shape descriptor:
   - it improves the vehicle lights classification accuracy by over 1 pp (percentage point) when added to the already used set of features
   - it can replace other state-of-the-art visual descriptors (8 values of the proposed descriptor instead of 32 features already used in the system) and keep the classification accuracy at a similar level
   - simple calculation of its values
   - low computational complexity
   - its elements might be stored as eight 8-bit values
2) a double CNN-based classifier that operates on the images of two dimensions and a set of visual features
3) the evaluation of the presented scheme was performed with the use of videos of total duration exceeding 10 hours covering various lighting and weather conditions

The paper is organized as follows. The following section is dedicated to the state-of-the-art, covering both existing AHC solutions and visual descriptors used in light spot classification. Section III presents the methodology and workflow for vehicle lights detection and classification. Section IV describes new type of visual shape descriptor. Section V contains the experimental results. Finally, Section VI presents the conclusions and potential future work.

II. STATE-OF-THE-ART STUDY—AHC FUNCTIONS AND DESCRIPTORS USED IN LIGHT SPOT CLASSIFICATION

A. SOLUTIONS FOR AHC FUNCTION

In this Section we present existing solutions for the AHC function seen either from scientific or commercial side. It may be somewhat artificial to strictly distinguish between these two groups. Commercial solutions are usually based on previous scientific investigations.

The problem of detecting vehicles at night can be addressed in two ways - by using image-based or sensor-based approach. The former is usually selected since camera-based solutions are much more effective and precise in comparison to other sensors.

1) IMAGE-BASED APPROACH

The most common approach is the use of a series of frames captured by a camera at night time. Such sequence of frames is then analyzed in order to detect the presence of vehicles. This can be realized by performing the following steps:

a: LIGHT SPOTS DETECTION

The light spots detection methods are typically based on a fixed thresholding [10], [11], [12] or adaptive thresholding [13].

b: PAIRING

If the lights are grouped in pairs, this usually can help in distinguishing vehicle lights from other light sources. As described, for example, in [14], [15], and [16], when certain rules are satisfied, one can state that the detected light spots are paired together. It is based on geometrical, statistical and motion information measures of vehicle lights [17], [18]. The general rules for pairing are the following:

- vertical and horizontal positions should be taken into account – e.g. the spots should have similar vertical coordinate and they should be positioned close to each other,
- spots should have similar size,
- the width of the bounding box that encloses two spots should be greater than its height,
- area (measured in pixels) of both spots ought to be similar,
- shape of both lights should be also similar
- a symmetry condition has to be satisfied.
- motion information from the tracker is consistent
- correlation level between histograms of vehicle lights is sufficient

It is not always possible to meet all of the above conditions at the same time. It happens that the illumination of both lamps differs for various reasons (damaged bulb, obstruction, etc.). Thus, it is necessary to find features that allow for an effective classification of individual light spots.

c: TRACKING

It provides motion-related data and therefore, it is useful in the identification of vehicles at night time [19]. In night time detection, headlights and taillights are tracked. The most common method relies on the use of Kalman filter [11], [13] or the nearest neighbor matching [20]. However, the use of these methods is limited to linear systems. For non-linear scenarios the Extended Kalman Filter which relies on linearization of non-linear functions can be applied [21]. Another alternative approach is using the Particle Filter which handles the non-linearity by representing the probability distribution of the state variables with a set of particles that can effectively approximate complex non-linear relationships in the system [22].

d: CLASSIFICATION

Currently, the most popular approach for classification is based on machine learning. The most common solutions use AdaBoost [10], [12], Support Vector Machine (SVM) [12], [13] and neural networks [9], [23], [24], [25].

Each type of classifier requires a specific input vector. Usually it consists of various types of features. The most common state-of-the-art features, used in such algorithms, may be divided into several categories [12]:

- Position features – they include the x and y coordinates of the center of gravity of a light spot. This is motivated by the fact that the lights from the vehicles (both oncoming and preceding) and from the street lamps are generally located in different regions of the input image,
- Brightness features – they include the average and the variance of intensity calculated for all the pixels within a light blob. Typically, the objects far away or the reflected objects are less bright than the ones which are nearby. Additionally, headlamps are generally brighter than taillamps,
- Shape features – they describe the characteristics of the blobs, such as their area (number of pixels), major and minor axes (or their aspect ratio), the relation between the blob's area and the size of its bounding box, an inclination angle of the blob, sizes of the, so called, halo and the proportion of the blob's halo to its area,
- Spatial relation features – they describe the spatial arrangement between light blobs. This is connected with the fact that headlights and taillights usually exist in pairs, while the street lights are normally situated along the side of the road. Thus, the horizontal and vertical positions of the neighboring blobs are analyzed,
- Color features – they can refer to the dominant hue, average saturation and average value calculated for each blob. It is observed that each type of the light source usually has its concrete dominant color – headlights are usually white, taillights are reddish and street lights might be green or yellow,
- Motion features – they describe the trajectory of each blob from frame to frame. It is observed that each type of light generally has a given movement direction. For instance, headlights move towards the observer, taillights can move away or get closer but usually they are more in front, while street lights generally move outwards.

In particular approaches reported in the literature, the features presented above are used in different way in the AI algorithms responsible for the classification of light spots.

In [10], except for the binary and intensity features, they also utilize color information and features that measure the ratios between monochrome and red pixels. Their image sensor provides 75% of monochrome pixels and 25% of red pixels.

Another method was presented in [11] where a single camera module is used but it operates in two different modes for frame capturing – low exposure and auto exposure. For low exposure images a fixed thresholding is applied, while Laplacian of Gaussian (LoG) filtering is applied to auto exposure images, so that relevant features of images can be selected. Besides, their system takes into account lane detection information that limits the area of interest for the auto exposure images. This prevents potential false positive detections.

In [13] the input vector is mainly composed of intensity and binary features. Besides, they observed that one specific feature of headlights and taillights is the halo effect. It does not occur in passive lamps, such as street lights, and nuisance artifacts. For the detection of the lights' halo, they proposed the black hat transformation. This transformation is defined as the difference between the morphological closing and the input image. The top-hat transformation is a powerful operator that enables the discerning of contrasted objects on non-uniform background.

Similarly, in [14], they try to consider the information about the blobs' position. The preliminary step is to filter out all non-vehicle light objects by rejecting the blobs located above the one-third of the vertical y-axis. However, such an assumption may lead to drastic reduction of the detection range distance and makes the system more prone to potential errors in hilly areas.

In contrast to the previously discussed solutions which operate on visible light images and focus on the detection of vehicle lights at night, recently also other approaches were proposed. The first group consists in an analysis of the scene captured by the infrared camera. In [26] the authors use a fast saliency map to select the areas where potential pedestrians or vehicles are present. Then, a multi-feature fusion algorithm combined with SVM classifier is used to determine whether the extracted target area really includes pedestrians or vehicles. In [27] the input images are preprocessed using various infrared-visible bimodal image fusion algorithms. In the next step, the four parameters (standard deviation, information entropy, mean gradient and spatial frequency) of the fused images are quantitatively analyzed. In [28] the authors combine image registration from visible light and near-infrared (NIR) band using SURF feature, pixel-level fusion based on YUV color space and the discrete wavelet transform (DWT) and transfer learning with YOLOv4 model. Another interesting approch entails the image enhancement thanks to Generative Adversarial Network (GAN). In [29] the GAN network is used for image translation from the nighttime image to the corresponding daytime image. Then, Faster R-CNN and YOLOv5 algorithms are used for vehicle detection. Similarly, in [30] a GAN-based network, called EnlightenGAN, is trained for enhancing the low-light image.

2) SENSOR-BASED APPROACH

An alternative way to detect vehicles at night and automatically control the vehicle's high beams is by utilizing a light sensor which provides the information about the light intensity. Based on its measurements, one can estimate whether some vehicles are present in the scene or not. The sensor is usually a photo sensitive device. In [31] and [32] LDR (Light Dependent Resistor) is used to analyze the light from oncoming vehicles. Then, they compare the measured light intensity with a predefined threshold value. In [31] if it exceeds this threshold, an excessive glare event is detected and a request to reduce the intensity of headlights is transferred via an XBee module to an oncoming vehicle. In contrast, in [32], the headlights are dimmed in the vehicle where the glaring event was detected.

For the completeness of the state-of-the-art presentation, it is also necessary to refer to the existing commercial solutions. Several automobile manufacturers have proposed their own solutions for the AHC functions, briefly described below.

3) SMARTBEAM

SmartBeam developed by Gentex [33] uses a camera mounted in the rear-view mirror together with automatic decision-making related to vehicle's high beam control. The system aims at the optimization of lights' usage with regard to the current traffic conditions so that they provide the best possible road illumination in glare-free manner. Specifically, the system detects oncoming or preceding vehicles and dynamically controls special shutters, that the headlamps are equipped with. This way some portions of assisted vehicle's light are blocked in order to prevent other road users from being dazzled. As a result, the road safety is increased and the sufficient forward illumination is provided.

4) INTELLIGENT HIGH-BEAM CONTROL

Intelligent High-Beam Control (IHC) developed by Mobileye [34] processes the frames captured from the camera mounted on the windshield. When the headlights from another vehicle are detected at a distance of less than 800 meters or there is a preceding vehicle in front of the host vehicle at a distance of less than 400 meters, the system switches off the high beams. The same happens when the vehicle enters a well-lit area. Only in case of the absence of street lights and no traffic detected, the high beams are automatically activated.

5) MULTIBEAM LED HEADLAMPS

Multibeam LED headlamps developed by Mercedes [35] utilize light gridding for high beam control. Each headlight is equipped with 84 independently activated high-performance LEDs which prevent other road users from being blinded and provide sufficient level of illumination of the road surface.

TABLE 1. Summary and comparison of other AHC-related methods and our referenced AHC system.

Algorithm	Brief methodology	Highlights	Limitations
Lopez et al. [10]	it is based on the image sensor that provides 75% of monochrome pixels and 25% of red pixels	the use of binary and intensity features, color information and features that measure the ratios between monochrome and red pixels	when the vehicle lights are further than 300 m, a series of frames needs to be analyzed to detect them
Eum et al. [11]	it is based on a single camera module that operates in two different modes for frame capturing – low exposure and auto exposure	the system takes into account lane detection information that limits the area of interest for the auto exposure images	potential false positives from reflections and misdetections of further vehicle lights
Alcantarilla et al. [13]	it is based on SVM classifier that takes as the input some intensity and binary features	the halo effect is also analyzed with the use of the black hat transformation	since the black-and-white camera is used, the taillights that are further than 200 m might not be detected
Chen [14]	it performs multilevel thresholding and connected-component analysis	the information about the blobs' position is taken into account	reduction of the detection range distance and potential errors in hilly areas
Bell et al. [9]	it is based on a series of foveal classifiers where each classifier is trained to provide the classification result for a particular image region	only point-based annotations are sufficient to train the grid of foveal classifiers which speeds up the creation of image databases	the localization performance depends on the number of classifiers and their distribution
Ewecker et al. [23]	it uses the detected light artifacts to control in a proactive way the glare-free high beam system	detection is based on observations of visual cues (e.g. shadows) the objects cause before they can be noticed	the method may not handle the taillights' detection
Liu et al. [24]	Attentive Generative Adversarial Network (GAN) is used to enhance the vehicle features while the classification is based on the Faster Region-based Convolutional Neural Network (R-CNN)	method achieves remarkable performance in extremely dark, occluded, and dazzling conditions	missed detection problems might occur at a distance
Huang et al. [25]	it is based on deep neural network model M-YOLO based on YOLOv3	the depth separable convolution can reduce the size of the model and the number of parameters	the frame rate per second does not exceed ten
Our AHC system	it is based on a double CNN-based classifier that operates on the images of two dimensions and a set of visual features	the use of a novel Binary-Directional Shape descriptor	the need for finding an optimal set of input features for the classifier

A grid light source allows for dynamic and adaptive control of the light distribution from left and right headlamps under given traffic conditions. This enables the illumination of separate areas of the road with high precision and brightness.

6) INTELLIGENT HEADLIGHT CONTROL

Intelligent headlight control developed by Bosch [36] analyzes the ambient brightness and estimates the distance to the vehicles in front based on the video sequence captured by the camera. It utilizes headlamps that are swivel-mounted vertically and horizontally or full LED headlights which enable the light to be distributed in particular segments in a controlled manner. This assures that other road users are not exposed to an excessive glare while the remaining area of the road can be optimally illuminated by the vehicle's high beams.

The system proposed by Bosch also uses a video camera. It analyzes the lighting conditions and when they are not sufficient, it activates the headlight control module. Headlights of oncoming vehicles and taillights of preceding vehicles are detected and based on this information, the headlights of the assisted vehicle are controlled in order to avoid dazzling other drivers.

B. EXISTING DESCRIPTORS AND USED APPROACHES

Since the topic of our work is related to descriptors used in the process of classifying light objects using neural networks, it is also necessary to present the state-of-the-art study in the area of the descriptors themselves.

1) HISTOGRAM OF ORIENTED GRADIENTS (HOG)

HOG [37] is a feature descriptor used in image processing for object detection tasks. In the HOG descriptor, the histograms of oriented gradients are used as features. The magnitude of gradients is large nearby edges and corners that carry much more information about the object shape than flat regions.

2) LOCAL BINARY PATTERN (LBP)

LBP [38] is a type of texture descriptor used in computer vision for image classification. LBP labels each pixel by comparing its gray component value with the neighboring pixels and then assigning a binary number. When all the pixels are labeled, the histogram of all LBP codes is constructed as LBP feature.

3) SIFT (SCALE-INVARIANT FEATURE TRANSFORM)

SIFT [39] is a feature extraction method broadly used in object recognition and point matching between different views of 3D objects. The SIFT descriptor is invariant to translation, rotation and scaling transformations and robust to illumination fluctuations, noise, partial occlusion and moderate viewpoint changes. In this method, the SIFT keypoints of objects are firstly extracted from an image. Then, a histogram of image gradients is calculated for each selected keypoint to describe the local image region around it.

4) SURF (SPEEDED UP ROBUST FEATURE)

SURF [40] is a local feature detector and descriptor. It is inspired by SIFT descriptor to some extent. As in case of SIFT descriptor, the points of interest are firstly selected and for each of them feature description is performed. SURF algorithm is several times faster than SIFT and it is also more robust against different image transformations. Both operations of keypoint detection and feature description are performed faster thanks to the use of integral images which reduces the computation complexity. SURF descriptor is based on the sum of Haar wavelet responses around the points of interest.

5) INTENSITY FEATURES

Intensity features [13] – this group covers all the measures related to intensity, for example moments, hat value which is an indicative parameter of the lights' halo based on the black hat transformation and other statistical measures like mean and maximum values or standard deviation.

6) COLOR INFORMATION

Color information [10] – this refers to various information related to color presence, dominance and distribution in the image, for example the ratio between the mean of monochrome level and the mean of red level.

III. METHODOLOGY AND WORKFLOW FOR VEHICLE LIGHTS DETECTION

A. BASICS

The referenced AHC system is based on two CNN classifiers, containing 1 and 3 convolution layers, respectively. The 12 × 12 pixel image and 32 feature values calculated for such a patch are the input for the first neural network. The 22 × 22 pixel patches and 60 values that come from one of intermediate layers of the first CNN are the input to the second network. The outputs from both networks are added together and form the final model output. Both neural networks were implemented with the use of PyTorch library in Python [41]. The small images carry the information about the light itself while the bigger images also hold some information about the surrounding of the light. This dual analysis enabled the improvement of the classification accuracy by a few percentage points. Similarly, the use of a set of visual features brought further improvement. Obviously, this entails a trade-off between higher computational complexity and more satisfying results. The detailed architecture of the referenced AHC classifier is presented in Fig. 1.

B. HYPERPARAMETERS SELECTION

Although the main emphasis in this paper is put on the visual features used as an input for the CNN, it is still worth mentioning other factors that can have some impact on the final classification accuracy of vehicle lights. Apart from the network architecture, in order to reproduce the results of our AHC scheme, one should be also aware of the right selection of hyperparameters. The most relevant hyperparameters that were used during training were presented in Table 2.

TABLE 2. Hyperparameters used for training of the models.

Hyperparameter	Value
Optimizer	Adam
Learning rate	0.01
Loss function	CrossEntropyLoss
Epochs	100
Batch size	128

C. SIGNAL PROCESSING FLOW IN THE PROPOSED AHC FUNCTION

The implemented AHC functionality consists of a few stages which are briefly presented below.

1) IMAGE PREPROCESSING

At this stage standard image processing techniques are used, such as image smoothing, noise cancellation, etc. An original camera image (I_1) is the input for this stage and an enhanced image (I_2) is produced as its output.

2) LIGHT SPOTS SEGMENTATION AND DETECTION IN CAMERA IMAGE

Algorithms used at this stage receive as an input the I_2 images, being an output from the previous stage. The aim of these algorithms is to localize the light sources and crop the part of the images where they were found. The result of the signal processing carried out at this stage is a set of images, $I_{3,k}$, with predefined size, e.g. 15 × 15 (odd version) or 16 × 16 (even version) pixels, which are resized parts of the I_2 image. As a result, in the $I_{3,k}$ images the center of the detected light blob more or less overlaps with the center of a given image $I_{3,k}$. The k index indicates consecutive light blobs found in the given image frame.

3) COMPUTATION OF DESCRIPTORS

At this stage, various image descriptors are calculated for $I_{3,k}$ images in order to extract specific features of particular light spots. The list of state-of-the-art descriptors that may be applicable in AHC functionality is presented in Section II-B.

4) CLASSIFICATION OF LIGHT SPOTS BASED ON AI ALGORITHMS

The feature values, together with the images they were calculated on at the previous stage, are taken as the input for a CNN classifier. The light spots can be assigned to one of the three classes: "headlights", "taillights" and "other".

The flowchart for the AHC system is shown in Fig. 2.

D. INPUT IMAGE DATASET

The image dataset has been divided into two subsets: a training dataset and a validation one. The training set contains over 3 million samples that were extracted from several

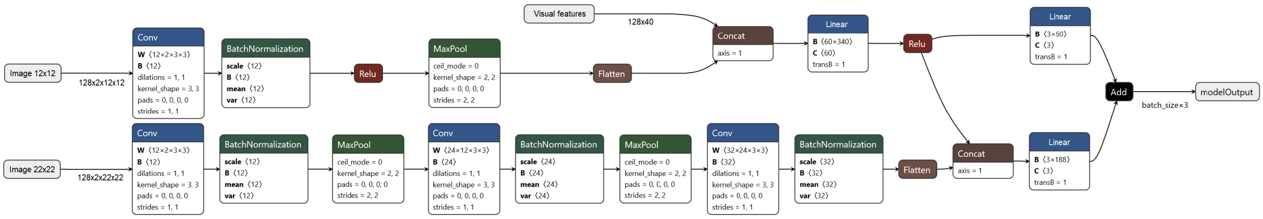

FIGURE 1. Architecture of the neural network used for AHC classification.

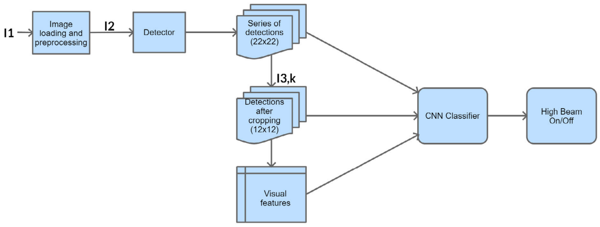
I1
Image loading and preprocessing
I2
Detector
Series of detections (22x22)
I3,k
Detections after cropping (12x12)
Visual features
CNN Classifier
High Beam On/Off

FIGURE 2. Architecture diagram of the referenced AHC scheme.

TABLE 3. Class representation in training and validation sets.

Dataset	Training Set	Validation Set
Headlight	170,240	20,710
Taillight	223,075	13,589
Other	2,681,426 (522,833)	164,238
Total	3,074,426 (916,148)	198,537

thousand video files of total duration of more than 10 hours. The validation dataset consists of almost 200,000 samples that were extracted from about 300 video logs of total length of 45 minutes. Such a large dataset is necessary to satisfy the safety requirements of our system. Videos were recorded in diverse weather conditions, including: clear sky, rain and snow with different intensities. In addition, three different types of roads were considered: highways, urban and rural roads. Such a variety of road and weather scenarios is highly required in both training and testing datasets in order to guarantee generalization that further leads to better reliability and robustness to varying conditions on the road [42]. In each video, all the light sources have been labeled with predefined classes. The exact numbers of samples for each class in both datasets are presented in Table 3.

In order to make the training set more balanced, undersampling of samples from the ''other'' class was performed. The resulting numbers are presented in brackets.

E. FEATURE DESCRIPTORS INCLUDED IN THE TRAINING

The features that had already been analyzed in the system can be grouped into three main categories: the intensity and the color features (INTCOL), e.g. the average and variance of intensity of all pixels within the blob, HOG feature, in which only some of its components are used and other various features (VAR), which include the measures that describe local surrounding of the detected light spots.

In order to improve the accuracy of the classifier some new types of features have been added and investigated:

- Gray and red blob properties – the imaging sensor from the referenced system produces two-channel images with gray and red components. As it might be seen in Fig. 3, in both gray and red channels, it is possible to observe characteristic features for all three classes of lights in the patches in which they are present. Therefore, we decided to analyze diverse features of light spots and check how this information affects the classification accuracy when it is provided as the input to the neural network. The image patches are binarized with the use of Otsu's method [43]. Among various blob properties, the following have been selected for further processing: area, major axis, minor axis, axis ratio, circularity, eccentricity, equivalent circular diameter, orientation and perimeter. The calculations of these properties were performed with the use of OpenCV library in Python [44].
- Binary Directional Shape (BDS) descriptor – it is a feature that consists of $2^N (N = 2, 3, \ldots)$ values that describe the shape of the binary blob. The algorithm is currently protected by US patent [45]. Some sample variants of the feature as well as its calculation method are presented in detail in the following section.

F. IMPLEMENTATION CHALLENGES

There are three implementation levels in our system:

I. Implementation of our new descriptor: no challenges, it is a simple and repeatable process (as shown in the following section)
II. CNN design: trade-off between the number of input features and the model performance
III. whole AHC system: ability to distinguish between vehicle lights and other lights

Especially, the third level sometimes might be tricky because it involves candidate lights detection, pairing, tracking and classification. However, the former three aspects are out of scope of this paper since we focused mainly on the classification step. Therefore, the main challenge is to select an optimal set of features to provide an acceptable computational complexity and satisfying model performance.

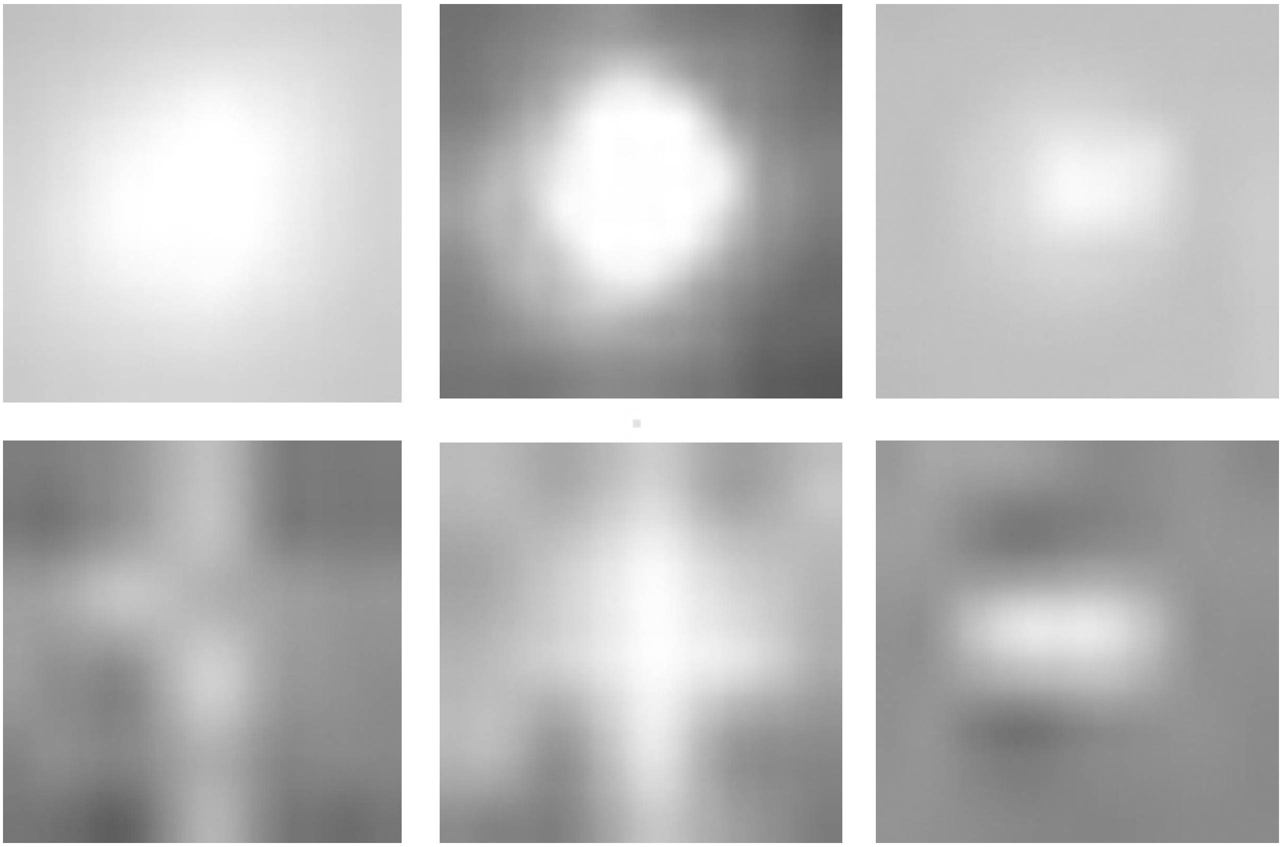

FIGURE 3. Patches of light sources. From left to right: headlights, taillights and other. Top row shows gray channel, bottom row shows red channel. Each column corresponds to a single light spot sample.

IV. PROPOSED DESCRIPTOR

A. AN OVERVIEW

Taking into account all the considerations and discussions presented in previous sections, in this paper, we propose a novel type of descriptor which, after being applied, leads to a significant improvement of light spot classification. This fact was verified during experimental tests of the referenced AHC function. The aim of the proposed solution was guaranteeing low computational power demand, which makes this method useful in the situation when the hardware resources are limited.

The carried out investigations indicate that the application of the proposed descriptor allows to replace or eliminate some number of remaining descriptors, while keeping or even improving the classification accuracy. Since the eliminated descriptors are more computationally complex, this translates into substantial reduction of the computation time.

B. ALGORITHMIC PROCEDURE

The proposed descriptor is applied at the 3^{rd} stage described in Section III-C3 above. The procedure may be divided into steps presented below.

Step 1: Centering the light spots in the $I_{3,k}$ images. The proposed descriptor already receives roughly centered spots, using other state-of-the-art methods.

Step 2: The processed $I_{3,k}$ images with the light spots are firstly binarized with an appropriate threshold, adaptively adjusted according to brightness distribution, e.g. using Otsu's method [43]. It is illustrated in Fig. 4, for selected typical situations. This operation helps to focus, in subsequent steps of the algorithm, only on the contours of the spot. The binarization is used to strongly reduce the amount of data necessary to unambiguously classify the spot to one of the three categories. After the binarization only one bit is required to store a single pixel. Additionally, we propose a solution, in which we skip a separate step, in which the overall image is binarized and then stored in the memory. Instead, we propose an approach, in which the binarization is performed ‘‘on the fly’’ (OTF) over only selected pixels. That simplifies the overall operation. We binarize only those pixels that are involved in the computation (counting operation) of new descriptors in the 4th step of the algorithm, as described below.

The binarization of the overall image requires mn iterations with a thresholding operation, where m and n are the sizes of the $I_{3,k}$ images. In the OTF approach, on the other hand, the binarization is performed only on those pixels in the $I_{3,k}$ images, which are overlapped by particular directions as described in Step 3 below. This is one of the advantages of the proposed OTF approach. This also

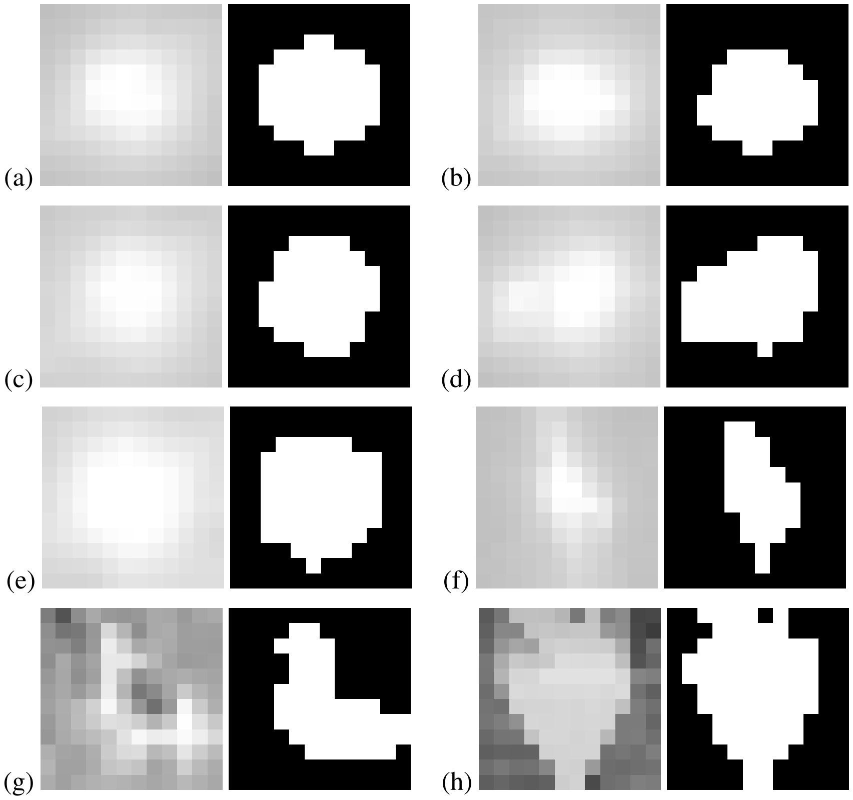

FIGURE 4. Selected light spots and their binary representation.

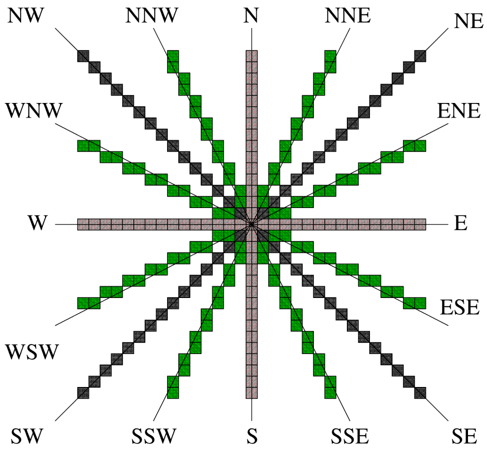

FIGURE 5. Directions used in the proposed descriptor.

reduces memory requirements that is an additional asset here.

Fig. 4 presents several typical cases that are encountered during the computations. Diagrams (a) and (b) present more regular spots usually associated with vehicles' lights. Diagram (c) shows a situation, in which a square spot is rotated to some degree. The (d) – (h) diagrams illustrate less regular spots. Such spots are usually associated with other light sources, however sometimes due to a fuzzy effect they may be also provided by car lights.

Step 3: Selection of the number of directions along which the light pixels are counted in the proposed descriptor. Directions in this case mean lines that concentrically diverge from the center of an image $I_{3,k}$ to its borders, as shown in Figs. 5 and 6, for a sample case of a square ($m = n = 15$). Fig. 5 illustrates potential directions in the whole frame, whereas Fig. 6 shows detailed options in a single quadrant.

The number of used directions may vary, depending on how precisely the spots need to be described. The higher the angular density of the directions is, the more details of the shape of the spots may be distinguished. Although, increasing the number of directions increases the memory usage, as well as the time needed to compute the descriptor, the increase in these values is not substantial.

A different number of directions may be used, depending on the level of details and parameters of the processed light spots. Fig. 6 presents several cases. To maintain symmetry, we consider cases with 4, 8, 16 and 32 directions. Corresponding binary descriptors are denoted as BDS4, BDS8, BDS16 and BDS32, respectively. Particular directions are presented in diagrams (b) – (f) for a single quadrant in Fig. 6. This allows to illustrate, in detail, which pixels are taken into account for all the variants. The same indexing approach is used in remaining three quadrants.

The BDS4 descriptor uses only four directions, two vertical (N, S) and two horizontal (W, E) ones. Diagram (b) presents selected (N and E) directions used in this case.

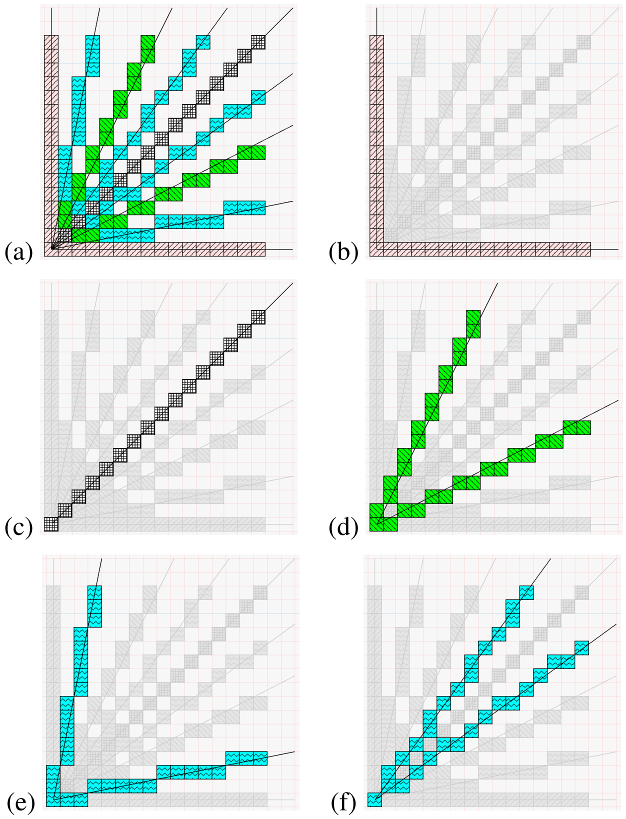

FIGURE 6. Selected directions used in proposed descriptors – one quadrant view: (a) directions for all described descriptors (BDS4, BDS8, BDS16 and BDS32), (b) directions used in BDS4 descriptor, (c) additional directions used in BDS8 descriptor, (d) additional directions used in BDS16 descriptor, (e, f) additional directions used in BDS32 descriptor.

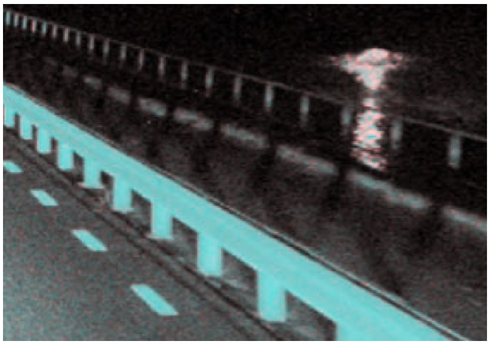

FIGURE 7. Problems with reflections, resulting in not regular shape of the spot – here a reflection from a water.

The BDS8 descriptor uses the same directions as BDS4, with additional four diagonal directions: NW, NE, SE, SW. It is illustrated in diagrams (a) and (c).

The BDS16 descriptor, in the comparison with the BDS8 one, additionally uses intermediate directions denoted as: NNE, ENE, ESE, SSE, SSW, WSW, WNW and NNW, as illustrated in diagrams (a) and (d).

In the BDS32 descriptor, 16 additional directions are additionally applied, as presented in diagrams (e) and (f).

Step 4: Counting the light pixels along particular directions, starting from the center of the spots (I_3 images). The counting may be performed using the Binary Coded Decimal (BCD). On the other hand, a direct binary representation (BIN) may be used which entails reading values 0 and 1 from the center outwards where the first value will be associated with the least significant bit of the resulting descriptor value. Each of these approaches offer some advantages. In the BCD code the memory usage is smaller, however we lose the information about the distribution of light pixels in particular directions. The BIN representation is more suitable for describing concave objects, in which discontinuities may appear in bright areas when looking in particular directions.

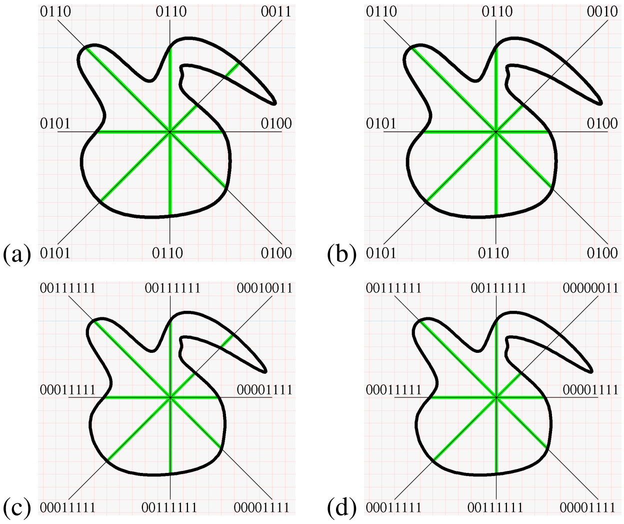

FIGURE 8. The way of describing particular directions in the proposed descriptors: (a) counting of the pixels (BCD approach) (b) counting of the pixels with detection of a discontinuity (BCD_CUT approach) (c) direct representation of pixels (BIN approach) (d) direct representation of pixels with detection of a discontinuity (BIN_CUT approach).

The counting operation may be performed also with an assumption, that it stops after reaching a first dark pixel in a given direction. This may allow to roughly filter out a noise, that, for example, may be caused by reflections of irregular surface, as seen in Fig. 7.

All the above-mentioned variants are illustrated in Fig. 8, for a sample concave spot, as follows:

Diagram (a) illustrates the BCD approach, in which all bright pixels in particular directions are counted. In the presented example, the result for the NE direction is 3, as the pixel inside an object visible in the upper right corner, stuck to the regular part of the spot, is also taken into account.

Diagram (b) illustrates the BCD_CUT approach, in which the pixels are counted in a given direction until a dark pixel is encountered. The remaining pixels are in this case cut-off, independently of their values.

Diagram (c) presents results obtained in direct BIN approach.

Diagram (d) illustrates the impact of detecting a discontinuity (a first dark pixel) in particular directions (BIN_CUT case).

Step 5: An issue is the way how the descriptor is stored in the memory. The sizes of the $I_{3,k}$ images are usually small, not exceeding 32 × 32. This results in maximum 16 pixels in each direction. In this case only several typical 32-bit integer variables are sufficient to describe the spot. The number of integer values equals the number of directions divided by 2, for example 2 for BDS4, 4 for BDS8, etc.

C. IMPLEMENTATION OF THE PROPOSED DESCRIPTORS

In the proposed descriptor, the number of directions may differ, as mentioned above and shown in Fig. 5. Every increase of the number of directions allows for more detailed description of the light spot. More complex variants of the proposed descriptor help in distinguishing other characteristics of the spot, as summarized below.

One of the issues is a way of efficient indexing over the two-dimensional input matrix (image $I_{3,k}$). To illustrate it, let us assume a square image $I_{3,k}$ with the size $m \times n$, $m = n = 15$ (an odd case). In the presented example, we assume that the coordinates of the top left corner of the image are $(0, 0)$, while of the bottom right corner are (M, M), where $M = m - 1$, in this case. The pixels to be considered in particular directions are selected based on the Bresenham's algorithm [46], as illustrated in Fig. 6 (b – f). In the proposed solution, the Bresenham's algorithm may be simply implemented in the proposed feature. Indexing over the matrix of pixels in particular directions is performed as presented below. In this sample case, the coordinates of the central point of the image are equal to: $c_x = M/2$; $c_y = M/2$;

1) BINARY DESCRIPTOR WITH FOUR DIRECTIONS (BDS4)

The BDS4 descriptor may be expressed as follows:

$$\begin{aligned}
&\mathit{Direction}:\\
&\quad N: ID_x = c_x; ID_y = c_y - [0, \ldots, M/2];\\
&\quad S: ID_x = c_x; ID_y = c_y + [0, \ldots, M/2];\\
&\quad W: ID_y = c_y; ID_x = c_x - [0, \ldots, M/2];\\
&\quad E: ID_y = c_y; ID_x = c_x + [0, \ldots, M/2]; \qquad (1)
\end{aligned}$$

where c_x, c_y - coordinates of the central point of the image ID_x, ID_y - coordinates of pixels for which the binary values should be read

The BDS4 descriptor does not allow to distinguish a circle from a square, and a rectangle from an ellipse. However, it allows to distinguish an ellipse from a circle, and a rectangle from a square. It does not allow to determine if the mentioned geometric figures are rotated either. While rotating a square object, in each of the four directions there are equal numbers of pixels. However, while rotating the object, the number of pixels in each direction equally increases, reaching their maximum values for rotations by 45, 135, 225, 315 degrees. In case of rotating an ellipse or a rectangle objects, the BDS4 descriptor allows to determine if the object is oriented vertically or horizontally. However the rotation angle is not possible to be determined.

2) BINARY DESCRIPTOR WITH EIGHT DIRECTIONS (BDS8)

The BDS8 descriptor uses the same directions as in BDS4 case, with additional diagonal ones:

$$\begin{aligned}
&\mathit{Direction}:\\
&\quad NW: && ID_x = c_x - [0, \ldots, M/2];\\
& && ID_y = c_y - [0, \ldots, M/2];\\
&\quad NE: && ID_x = c_x + [0, \ldots, M/2];\\
& && ID_y = c_y - [0, \ldots, M/2];\\
&\quad SW: && ID_x = c_x - [0, \ldots, M/2];\\
& && ID_y = c_y + [0, \ldots, M/2];\\
&\quad SE: && ID_x = c_x + [0, \ldots, M/2];\\
& && ID_y = c_y + [0, \ldots, M/2]; \qquad (2)
\end{aligned}$$

In case of the BDS8 descriptor, vertical, horizontal, as well as diagonal directions are used. It allows to distinguish such shapes as circles, squares, rectangles, ellipses and other more irregular objects. In this case it is also possible to say if a regular object is rotated.

3) BINARY DESCRIPTOR WITH SIXTEEN DIRECTIONS (BDS16)

The BDS16 descriptor uses the same directions as in BDS4 and BDS8 cases, with additional intermediate ones:

$$\begin{aligned}
&\mathit{Direction}:\\
&\quad NNE: && ID_x = c_x - [0, \ldots, M/2] \gg 1;\\
& && ID_y = c_y - [0, \ldots, M/2] \gg 0;\\
&\quad ENE: && ID_x = c_x + [0, \ldots, M/2] \gg 0;\\
& && ID_y = c_y - [0, \ldots, M/2] \gg 1;\\
&\quad ESE: && ID_x = c_x - [0, \ldots, M/2] \gg 0;\\
& && ID_y = c_y + [0, \ldots, M/2] \gg 1;\\
&\quad SSE: && ID_x = c_x + [0, \ldots, M/2] \gg 1;\\
& && ID_y = c_y + [0, \ldots, M/2] \gg 0;\\
&\quad SSW: && ID_x = c_x - [0, \ldots, M/2] \gg 1;\\
& && ID_y = c_y - [0, \ldots, M/2] \gg 0;\\
&\quad WSW: && ID_x = c_x + [0, \ldots, M/2] \gg 0;\\
& && ID_y = c_y - [0, \ldots, M/2] \gg 1;\\
&\quad WNW: && ID_x = c_x - [0, \ldots, M/2] \gg 0;\\
& && ID_y = c_y + [0, \ldots, M/2] \gg 1;\\
&\quad NNW: && ID_x = c_x + [0, \ldots, M/2] \gg 1;\\
& && ID_y = c_y + [0, \ldots, M/2] \gg 0; \qquad (3)
\end{aligned}$$

In case of the BDS16 it is possible to describe and distinguish the spots visible in Fig. 4. However, in many practical situations such detailed information is not required.

D. VERIFICATION–APPLICATION OF THE PROPOSED DESCRIPTORS IN AHC FUNCTION

In case of the AHC applications, there is a strong need to distinguish street lamps from vehicle lights. In this case the

BDS8 descriptor is usually sufficient. The AHC function requires also distinguishing head and tail vehicle lights.

The method after an additional adaptation may also be used in traffic sign recognition (TSR) functions to distinguish digits on traffic signs. In North America region, for example, there is a need to distinguish the number 0 and 5 at the second position in speed limit traffic signs (e.g. 20 and 25 mph). In this case BDS16 or BDS32 descriptors may be required.

V. EXPERIMENTAL RESULTS

When the classifier already provides high classification accuracy, it is more difficult to make a significant progress and obtain better results. However, it is possible to measure the strength of each feature, by using each group of features separately in training, without patch information. This way, one can determine how much valuable information each group carries. Except for the new features proposed in this paper, all three original groups of features: INTCOL, HOG and VAR have been tested. The results are shown in Table 4. This table compares the classification accuracy of the proposed descriptors with other state-of-the-art descriptors used in the same applications.

In addition, the information about the number of features in each group, the average calculation time of a single value and total memory occupancy for each group is provided.

T1 and T2 refer to two training repetitions, performed with particular groups of features. The mean accuracy is presented in the last column.

One can observe that features that describe the neighborhood of the detected light spots, as well as those related to intensity and color information provide satisfactory results. Gray Blob features may contribute to headlights detection while Red Blob features handle well in case of taillamps. Proposed BDS feature turns out to be beneficial for the detection of all categories of lights at the same time being computationally very cheap – taking into account both calculation time and memory usage.

Table 4 shows that in comparison with the proposed BDS descriptors, the INTCOL and VAR ones offer more accurate results. However, firstly, it is worth highlighting that VAR, INTCOL and Gray/Red Blob features are groups of descriptors that consists of 12, 14 and 9 values, respectively. Each value is calculated by using a different formula, which makes these four groups computationally much more complex. They require such operations like multiplications, divisions, square rooting, histogram calculation, etc. Secondly, when they are considered individually, they do not allow to achieve the performance of the classifier as good as in case of BDS solution. Moreover, in case of 16 × 16 patches, a single BDS descriptor value might be stored on 8-bit variable only. In turn, HOG features consist of only 6 floating point numbers but selected from 32, which need to be calculated for each data sample since they represent a histogram of gradient orientations in four image quadrants. Therefore, this group is also quite heavy computationally. It might be observed that Gray/Red Blob features provide similar results as these obtained by using the HOG feature. However, the proposed BDS feature leads to much better accuracy when analyzed separately.

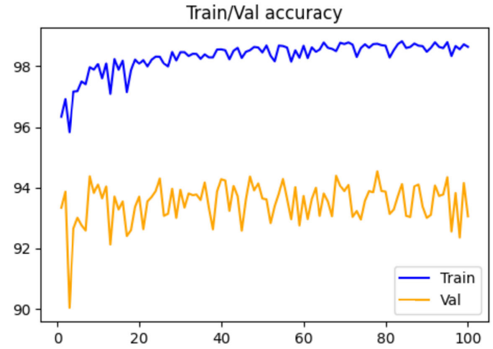

FIGURE 9. Training and validation accuracy vs. training epoch for the best performing model with BDS feature added to the input for the classifier.

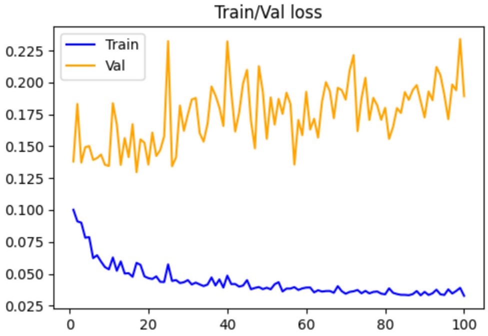

FIGURE 10. Training and validation loss vs. training epoch for the best performing model with BDS feature added to the input for the classifier.

The results shown in Table 4 for the proposed BDS descriptor (last four rows), are for all four variants: BDS4, BDS8, BDS16 and BDS32. Admittedly, BDS may contain a comparable number of values. However, the key factor here is the computational complexity for computing the values that belong to particular groups. In this context, the BDS feature is much simpler. It requires only some basic preprocessing of the input image and then its values are easily returned by following the steps described in Section IV. It can be noticed that BDS8 might be a reasonable choice in vehicle lights classification since using more complex variants (BDS16 and BDS32) does not lead to better results. It might be also observed that, for instance, in case of BDS8 variant, the average time to calculate a single feature value is approximately 37, 16, 31 and 4 times faster and occupy 3.25, 5.875, 3 and 3.5 times less memory than VAR, INTCOL, HOG and Gray/Red Blob groups, respectively.

Each proposed feature was considered independently by adding it to a list of features already used in the system and performing the training and testing phases from the beginning. Selected results are presented in Table 5. The first row shows the referenced accuracy for the original dataset including VAR, INTCOL and HOG. And the following six rows present the results obtained after adding Gray Blob, Red

TABLE 4. Comparison of the proposed solution with other state-of-the-art descriptors.

					Accuracy [%]			
Feature	No. values	Average calculation time [µs]	Total memory usage [b]	#T	Headlight	Taillight	Other	Mean acc.
VAR	12	1007.1	208	T1	73.83	62.63	95.10	77.19
				T2	70.39	65.02	95.31	76.91
INTCOL	14	427.3	376	T1	88.64	80.54	88.41	85.86
				T2	89.61	78.68	88.79	85.69
HOG	6	835.2	192	T1	15.27	18.56	89.19	41.01
				T2	12.16	18.27	90.32	40.25
Gray Blob	9	110.6	224	T1	30.6	3.8	92.5	42.3
				T2	20.0	1.0	94.9	38.63
Red Blob	9	111.2	224	T1	10.4	44.4	87.6	47.47
				T2	10.4	40.6	89.3	46.77
BDS4 (Proposed)	4	28.3	32	T1	35.60	52.06	76.09	54.58
				T2	34.20	46.48	79.97	53.55
BDS8 (Proposed)	8	27.0	64	T1	54.02	57.48	73.52	61.67
				T2	47.33	53.86	78.51	59.90
BDS16 (Proposed)	16	19.7	128	T1	44.11	61.50	77.88	61.17
				T2	47.53	49.08	82.67	59.76
BDS32 (Proposed)	32	15.7	256	T1	56.81	45.54	79.45	60.60
				T2	48.28	49.44	80.60	59.44

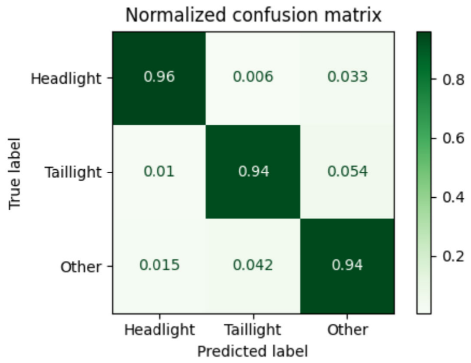

FIGURE 11. Confusion matrix for the best performing model with BDS feature added to the input for the classifier.

Blob and all four variants of BDS features, respectively. The best performing model is the one where the proposed BDS descriptor consisting of 32 values was added to the input data for the classifier.

Tables 4 and 5 show that BDS8 is a good trade-off between satisfying classification accuracy and the number of features needed to be calculated - BDS16 and BDS32 provide quite similar results (only 0.09 pp better in case of BDS32) but require additional calculation time: 216 µs (BDS8) vs. 315.2 µs (BDS16) and 502.4 µs (BDS32). That is why, BDS8 seems to be optimal variant in vehicle lights classification. For this case, in Fig. 9 the graph depicting the training and the validation accuracy is presented. In addition, the training and the validation loss is shown in Fig. 10. Furthermore, the confusion matrix that shows how well particular classes are recognized is depicted in Fig. 11.

The presented results demonstrate that adding new features bring slight enhancement in terms of classification performance. This is especially the case while applying BDS feature which gives an improvement of around 1.5 pp in classification accuracy. This result may seem to be a minor improvement, however, in this case, the system investigated was already achieving a performance close to 95%. In such a case, the improvement of over 1 pp in accuracy is noticeable.

TABLE 5. Classification accuracies after adding one feature.

Feature	Mean acc. [%]
ORIG	93.07
ORIG + Gray Blob	93.43
ORIG + Red Blob	93.13
ORIG + BDS4	94.34
ORIG + BDS8	94.55
ORIG + BDS16	94.55
ORIG + BDS32	94.64

One of the important aspects is to assess the stability in terms of complexity. In our work, the proposed feature operates on the 12 × 12 image patches. Calculation of particular values of the descriptor consists in binarization of the image (only in the selected directions) and then reading the pixel values. It's a simple and repeatable process. Thus, the complexity is low and stable when the processing of a single patch is taken into account. In contrast, regarding the referenced AHC system, the complexity and the workload of operations depends on the number of detections found by the system in particular frames. However, the processing of a single detection is always constant as mentioned above.

VI. CONCLUSION AND FUTURE WORK

In this work, we presented a new, computationally efficient, BDS descriptor that in a simple way enables a description of selected characteristics of light spots registered in video streams captured by vision sensors installed in the vehicle. The new descriptor can be used as a component of a training dataset for neural networks, the role of which is to classify light spots registered during night time driving. The motivation behind the investigations undertaken was an attempt to improve the effectiveness of the AHC algorithms. These algorithms belong to a group of modern ADAS algorithms.

The new feature descriptor was proved to contribute to the classification accuracy improvement in the task of vehicle lights recognition. It seems to be a beneficial complement for the already used CNN classifiers in the referenced AHC system. This is due to the fact that it provides satisfying results in comparison with solutions in which other types of descriptors are applied. Additionally, the proposed descriptor is computationally inexpensive, which is a paramount feature in real time systems operating on platforms with limited computational resources. Furthermore, it may become a promising solution in other ADAS algorithms, e.g. in Traffic Sign Recognition. Due to its flexibility and a range of variants proposed in this paper, the presented visual descriptor may be easily adjusted to various, even specific, tasks in order to describe the shape of the analyzed object with sufficient level of details.

It is worth highlighting that the referenced AHC system is a commercial solution and it was designed, tested and verified to satisfy a lot of strict requirements. Therefore, it is hard to indicate any relevant limitations. The main challenge which is presented in the paper is finding an optimal set of features that are provided as the input to the classifier. Eventually, it only serves to enhance the system even more and make it more robust.

The conducted research shows that the use of the proposed BDS descriptor increases the number of light spots correctly classified by the neural network by around 1.5 pp. This may seem to be a small number. However, it should be highlighted that these results are for the overall performance of the AHC algorithm with the accuracy at the level of 93 – 94%, i.e. in a situation where up to 7% of light spots are wrongly assigned to specific data classes. As a result, it can be said that the number of light spots incorrectly assigned to individual classes was reduced by about 21%. It is also worth adding that the AHC algorithm in which the new descriptor was used, was trained and then tested with a relatively large dataset containing more than 300 thousand light spot samples.

Further works in the presented topic will include an application of a modified BDS descriptor in the determination of the central point of the processed light blob. It could be an additional step, performed before feature calculation, that can lead to better description of light spots and in the end to increased classification accuracy of the CNN classifier.

REFERENCES

[1] *National Safety Council: Safety Topics*. Accessed: Sep. 17, 2023. [Online]. Available: https://www.nsc.org/road-safety/safety-topics

[2] *European Commission: Road Safety—Press Release*. Accessed: Aug. 17, 2023. [Online]. Available: https://ec.europa.eu/commission/presscorner/detail/en/IP_22_4312

[3] K.-F. Lee, X.-Z. Chen, C.-W. Yu, K.-Y. Chin, Y.-C. Wang, C.-Y. Hsiao, and Y.-L. Chen, "An intelligent driving assistance system based on lightweight deep learning models," *IEEE Access*, vol. 10, pp. 111888–111900, 2022.

[4] A. M. Ibrahim, R. M. Hassan, A. E. Tawfiles, T. Ismail, and M. S. Darweesh, "Real-time collision warning system based on computer vision using mono camera," in *Proc. 2nd Novel Intell. Lead. Emerg. Sci. Conf. (NILES)*, Oct. 2020, pp. 60–64.

[5] M. M. Faisal, M. S. Mohammed, A. M. Abduljabar, S. H. Abdulhussain, B. M. Mahmmod, W. Khan, and A. Hussain, "Object detection and distance measurement using AI," in *Proc. 14th Int. Conf. Develop. eSyst. Eng. (DeSE)*, Dec. 2021, pp. 559–565.

[6] P. Bogacki and R. Dlugosz, "Selected methods for increasing the accuracy of vehicle lights detection," in *Proc. 24th Int. Conf. Methods Models Autom. Robot. (MMAR)*, Aug. 2019, pp. 227–231.

[7] H. Gholamalinejad and H. Khosravi, "Vehicle classification using a real-time convolutional structure based on DWT pooling layer and SE blocks," *Expert Syst. Appl.*, vol. 183, Nov. 2021, Art. no. 115420.

[8] C. J. Rapson, B.-C. Seet, K. J. Lee, N. A. Naeem, M. Al-Sarayreh, and R. Klette, "Reducing the pain: A novel tool for efficient ground-truth labelling in images," in *Proc. Int. Conf. Image Vis. Comput. New Zealand (IVCNZ)*, Auckland, New Zealand, Nov. 2018, pp. 19–21.

[9] A. Bell, T. Mantecón, C. Díaz, C. R. Del-Blanco, F. Jaureguizar, and N. García, "A novel system for nighttime vehicle detection based on foveal classifiers with real-time performance," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5421–5433, Jun. 2022.

[10] P. Sevekar and S. B. Dhonde, "Nighttime vehicle detection for intelligent headlight control: A review," in *Proc. 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, Jul. 2016, pp. 188–190.

[11] S. Eum and H. G. Jung, "Enhancing light blob detection for intelligent headlight control using lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 1003–1011, Jun. 2013.

[12] Y. Li, N. Haas, and S. Pankanti, "Intelligent headlight control using learning-based approaches," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 722–727.

[13] P. F. Alcantarilla, L. M. Bergasa, P. Jiménez, I. Parra, D. F. Llorca, M. A. Sotelo, and S. S. Mayoral, "Automatic LightBeam controller for driver assistance," *Mach. Vis. Appl.*, vol. 22, no. 5, pp. 819–835, Sep. 2011.

[14] Y.-L. Chen, "Nighttime vehicle light detection on a moving vehicle using image segmentation and analysis techniques," *WSEAS Trans. Comput.*, vol. 8, pp. 506–515, Mar. 2009.

[15] S. Zhou, J. Li, Z. Shen, and L. Ying, "A night time application for a real-time vehicle detection algorithm based on computer vision," *Res. J. Appl. Sci., Eng. Technol.*, vol. 5, no. 10, pp. 3037–3043, Mar. 2013.

[16] J. Wang, X. Sun, and J. Guo, "A region tracking-based vehicle detection algorithm in nighttime traffic scenes," *Sensors*, vol. 13, no. 12, pp. 16474–16493, Dec. 2013.

[17] K. Gunasekaran and S. Padmavathi, "Night time vehicle detection for real time traffic monitoring systems: A review," *Int. J. Comput. Technol. Appl.*, vol. 5, pp. 451–456, Jan. 2014.

[18] T.-A. Pham and M. Yoo, "Nighttime vehicle detection and tracking with occlusion handling by pairing headlights and taillights," *Appl. Sci.*, vol. 10, no. 11, p. 3986, Jun. 2020.

[19] P. Sevekar, "Night-time vehicle detection for automatic headlight beam control," *Int. J. Comput. Appl.*, vol. 157, no. 7, pp. 8–12, Jan. 2017.

[20] A. Fossati, P. Schönmann, and P. Fua, "Real-time vehicle tracking for driving assistance," *Mach. Vis. Appl.*, vol. 22, no. 2, pp. 439–448, Mar. 2011.

[21] A. Jahantighy, H. Torabi, and F. Mohanna, "Multiple targets video tracking based on extended Kalman filter in combination with particle swarm optimization for intelligent applications," *Social Netw. Appl. Sci.*, vol. 5, no. 3, Mar. 2023. [Online]. Available: https://www.proquest.com/openview/ebf50a6ce1a9ef5b2eb2db64b25e42b6/1.pdf?pq-origsite=gscholar&cbl=5758472

[22] N. A. Valente, A. Sarrafi, Z. Mao, and C. Niezrecki, "Streamlined particle filtering of phase-based magnified videos for quantified operational deflection shapes," *Mech. Syst. Signal Process.*, vol. 177, Sep. 2022, Art. no. 109233.

[23] L. Ewecker, E. Asan, L. Ohnemus, and S. Saralajew, "Provident vehicle detection at night for advanced driver assistance systems," *Auto. Robots*, vol. 47, no. 3, pp. 313–335, Mar. 2023.

[24] Y. Liu, T. Qiu, J. Wang, and W. Qi, "A nighttime vehicle detection method with attentive GAN for accurate classification and regression," *Entropy*, vol. 23, no. 11, p. 1490, Nov. 2021.

[25] S. Huang, Y. He, and X.-A. Chen, "M-YOLO: A nighttime vehicle detection method combining MobileNet v2 and YOLO v3," *J. Phys., Conf. Ser.*, vol. 1883, no. 1, Apr. 2021, Art. no. 012094.

[26] T. Xue, Z. Zhang, W. Ma, Y. Li, A. Yang, and T. Ji, "Nighttime pedestrian and vehicle detection based on a fast saliency and multifeature fusion algorithm for infrared images," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16741–16751, Sep. 2022.

[27] Z. Wu, X. Miao, W. Li, and H. Yu, "Deep learning based nighttime target enhancement detection algorithm for intelligent vehicles," in *Proc. 6th CAA Int. Conf. Veh. Control Intell. (CVCI)*, Oct. 2022, pp. 1–6.

[28] L. Deng, M. Pan, R. Jin, and Z. Xie, "Night target detection approach based on near infrared image fusion on vehicles," in *Proc. 5th Int. Conf. Pattern Recognit. Artif. Intell. (PRAI)*, Aug. 2022, pp. 755–759.

[29] U. N. Nisha and G. Ranjani, "Deep learning based night time vehicle detection for autonomous cars using generative adversarial network," in *Proc. Int. Conf. Augmented Intell. Sustain. Syst. (ICAISS)*, Nov. 2022, pp. 336–341.

[30] C. Nie, S. Zhou, H. Zhang, and Z. Sun, "Monocular vision based perception system for nighttime driving," in *Proc. 8th Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2022, pp. 258–263.

[31] A. K. Tripathy, D. Kayande, J. George, J. John, and B. Jose, "Wi lights—A wireless solution to control headlight intensity," in *Proc. Int. Conf. Technol. Sustain. Develop. (ICTSD)*, Feb. 2015, pp. 1–5.

[32] M. Alsumady and S. Alboon, "Intelligent automatic high beam light controller," *J. Active Passive Electron. Devices*, pp. 1–8, 2013. [Online]. Available: https://www.researchgate.net/publication/296183467_Intelligent_Automatic_High_Beam_Light_Controller and https://www.ijcaonline.org/archives/volume157/number7/26841-2017912737 and https://iopscience.iop.org/article/10.1088/1742-6596/2325/1/012001/meta

[33] Gentex Corporation. *Smartbeam*. Accessed: Aug. 17, 2023. [Online]. Available: https://www.gentex.com/products-technology/automotive/digital-vision/

[34] *Mobileye: Intelligent High-Beam Control*. Accessed: Aug. 17, 2023. [Online]. Available: https://static.mobileye.com/manual/me8/

[35] *Mercedes-Benz: Multibeam LED*. Accessed: Aug. 17, 2023. [Online]. Available: https://media.mercedes-benz.com/article/6d30c3b6-9875-4663-8418-1a14adea6b91

[36] *Bosch: Intelligent Headlight Control*. Accessed: Aug. 17, 2023. [Online]. Available: https://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/driver-assistance-systems/intelligent-headlight-control/

[37] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2005, pp. 886–893.

[38] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[39] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Feb. 2001, pp. 1–12.

[40] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. 9th Eur. Conf. Comput. Vis.*, vol. 3951. May 2006, pp. 404–417.

[41] *PyTorch Library*. Accessed: Aug. 17, 2023. [Online]. Available: https://pytorch.org/

[42] J. J. Anaya, A. Ponz, F. García, and E. Talavera, "Motorcycle detection for ADAS through camera and V2V communication, a comparative analysis of two modern technologies," *Expert Syst. Appl.*, vol. 77, pp. 148–159, Jul. 2017.

[43] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[44] *OpenCV Library in Python*. Accessed: Aug. 17, 2023. [Online]. Available: https://pypi.org/project/opencv-python/

[45] P. Bogacki and R. Długosz, "Methods and systems for object detection," U.S. Patent 11 562 575 B2 , Feb. 17, 2021.

[46] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Syst. J.*, vol. 4, no. 1, pp. 25–30, 1965.

PIOTR BOGACKI was born in Kraków, Poland, in 1989. He received the B.E. and M.S. degrees in telecommunications engineering from the AGH University of Science and Technology, Kraków, in 2012 and 2014, respectively, where he is currently pursuing the industrial Ph.D. degree in computer science and telecommunications.

He participated in the Erasmus exchange programme, within which he spent the last academic year (2012 and 2013) with Universitat Politecnica de Valencia, Valencia, Spain. In 2014, he realized a professional internship with Capgemini Company, Valencia, where he developed database and business intelligence systems. From 2014 to 2018, he was a Software Engineer with Samsung R&D Poland, where he was responsible for the development of 4G and 5G systems. Since 2018, he has been an Expert Algorithm Development Engineer in automotive sector with Aptiv Poland, Kraków, where he has been responsible for the development of active safety vision features. Since 2011, he has been involved in seven research projects. His research interests include image processing and video analysis.

RAFAŁ DŁUGOSZ received the M.S. and Ph.D. degrees from the Poznan University of Technology, Poland, in 1996 and 2004, respectively, and the D.Sc. degree from the Technical University of Łódź, Poland, in 2016. Since 2014, he has been with Aptiv automotive company. He is currently a Professor with the Bydgoszcz University of Science and Technology, Poland. His research interests include design and optimization of low power microelectronic circuits and systems, including filters, analog-to-digital converters, artificial neural networks, and fuzzy systems implemented at the transistor level. He specializes in hardware algorithmic solutions, implemented as parallel and asynchronous application specific integrated circuits (ASIC) in the CMOS technology. He also specializes in signal processing algorithms for automotive industry (Aptiv company), proposing solutions for image and radar data processing, and data fusion. He has published more than 200 research papers, including international patents and book chapters in mentioned research areas. He was a fellow of several national and international scholarships, under the framework of which he spent over five years with the University of Alberta, Canada, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, and the Innovation for High Performance (IHP) Microelectronics Institute, Frankfurt Oder, Germany.