Soft Computing
https://doi.org/10.1007/s00500-020-04999-1

METHODOLOGIES AND APPLICATION

Enhanced pedestrian detection using optimized deep convolution neural network for smart building surveillance

Bubryur Kim[1] · N. Yuvaraj[1] · K. R. Sri Preethaa[2] · R. Santhosh[3] · A. Sabari[4]

Abstract
Pedestrian detection and tracking is a critical task in the area of smart building surveillance. Due to advancements in sensors, the architects concentrate in construction of smart buildings. Pedestrian detection in smart building is greatly challenged by the image noises by various external environmental parameters. Traditional filter-based techniques for image classification like histogram of oriented gradients filters and machine learning algorithms suffer to perform well for huge volume of pedestrian input images. The advancements in deep learning algorithms perform exponentially good in handling the huge volume of image data. The current study proposes a pedestrian detection model based on deep convolution neural network (CNN) for classification of pedestrians from the input images. Proposed optimized version of VGG-16 architecture is evaluated for pedestrian detection on the INRIA benchmarking dataset consisting of 227 × 227 pixel images. The proposed model achieves an accuracy of 98.5%. It was found that proposed model performs better than the other pretrained CNN architectures and other machine learning models. Pedestrians are reasonably detected and the performance of the proposed algorithm is validated.

Keywords Pedestrian detection · Deep learning · Convolution neural network · Machine learning

1 Introduction

In recent years, researches are focused on imparting artificial intelligence to machines for many activities that does not involve humans. It includes object detection, animal tracking, weather forecasting, surveillance, civil structural health monitoring and so on. Smart buildings surveillance includes identification of suspicious objects and activity. Smart buildings integrate sensor technology and the IoT processing to provide solutions in overall building management. The biggest automotive and IT companies draws attention in investing huge money on autonomous self-driving cars (Shen et al. 2019). They also involved in development of machines and applications for human–machine interaction. (Wu and Rehg 2011; Gall and Lempitsky 2013; Majaranta and Bulling 2014; Rautaray and Agrawal 2015; Boudjit and Larbes 2015; Rhodin et al. 2016).

In developing autonomous vehicles apart from designing machines, learning of environment is required. Recent studies have highlighted more than 5000 pedestrians were killed due to traffic crashes (Unies et al. 2015). It also added that the pedestrians are more likely to be killed than

Communicated by V. Loia.

✉ K. R. Sri Preethaa
sripreethaakr@gmail.com

Bubryur Kim
rlaqjqfuf@gmail.com

N. Yuvaraj
drnyuvaraj@gmail.com

R. Santhosh
santhoshrd@gmail.com

A. Sabari
drsabaria@gmail.com

[1] Department of Architectural Engineering, Kyungil University, Gyeongbuk, South Korea

[2] Department of CSE, KPR Institute of Engineering and Technology, Coimbatore, India

[3] Department of CSE, Faculty of Engineering, Karpagam Academy of Higher Education, Coimbatore, India

[4] Department of IT, K S Rangasamy College of Technology, Tiruchengode, India

Published online: 11 May 2020

the passenger victims in crashes (Wang et al. 2015; Tian et al. 2015; Beck et al. 2007; Felzenszwalb et al. 2010). It is important that autonomous vehicles should act according to the dynamic environments where there is high risk of any person in its range of action. For this reason, it is necessary to detect any person and track their activity.

Pedestrian detection is process of determining whether an image or a video contains a person and to identify his/her location and track the direction of activity (Li et al. 2016). Development in computer visions has extended its applications in the intelligent auxiliary driving, pedestrian detection and intelligent analysis robots. Pedestrian detection has given a number of applications in robotics, surveillance, in autonomous vehicular driving systems (Wang et al. 2017), road scene understanding systems (Cai et al. 2016) and so on. However, pedestrian detections are a challenging task due to the complexity of real-time environment, diversity in pedestrian posture and angle of capture of vision. Practically, the pedestrian detection system requires two main aspects of very high accuracy and real-time speed.

The pedestrian detectors should give fact and very accurate identification with systems of limited computing power (Angelova et al. 2015). Despite number of algorithms for pedestrian detection, reliable human shape detection is still under research. The challenges in detecting humans are due to a number of reasons like different poses a humans take, may be occluded by other humans or objects, the background may have objects in humanoid shape that leads to false detections. Many objects like fire extinguisher, chair or dolls in the range of detector may have same features that of humans are wrongly associated with pedestrians (Li et al. 2016; Rautaray and Agrawal 2015). The authors address both of these requirements by combining very accurate deep learning-based classifiers within very efficient cascade classifier frameworks (Krizhevsky et al. 2012).

Although deep learning has various unique advantages, it also has various shortcomings. Deep learning algorithms are largely black boxes and have low interpretability. In addition deep learning algorithms heavily reliant on data and computationally demanding (Zhang et al. 2019). The main challenge in dealing with deep learning algorithms is its hyper-parameters. Among the many hyper-parameters, finding their optimal configuration remains difficult. In convolution neural networks (CNN) it is necessary to configure the hyper-parameters for the number, shape, stride and filters. When the depth of deep learning algorithm increases, number of hyper-parameters associated with the architecture also increases exponentially.

The main contribution of this work concentrates on extracting the best hyper-parameters out of the deep learning model to enhance the accuracy of pedestrian detection. The working architecture of VGG-16, a convolution neural network (CNN)-based architecture is optimized by working on its hyper-parameters. By optimizing the hyper-parameters it is not only possible to enhance the accuracy of the deep learning models but also it helps in reducing the computational power used for execution.

The rest of the paper is discussed as follows. Related works in the literature are discussed in Sect. 2. Section 3 discusses about the experimental study conducted by implementing the proposed optimized VGG-16 (OVGG) method for pedestrian detection. The performance of the proposed model is compared with other machine learning (ML) and deep learning (DL) models, namely hybrid metaheuristic pedestrian detection (HMPD) and VGG-16 model. Proposed model for pedestrian detection in smart buildings is discussed in Sect. 4. The results are discussed in Sect. 5. The key observations of this particular work are concluded in Sect. 6.

2 Related work

Recent years, the artificial intelligence (AI) has started conquering the world of research. It has a variety of applications in many fields. AI systems are replacing humans in places where it is found fatal to employ humans. In such systems, object detection has become an inevitable in AI systems. In earlier researches, integral channel features (ICF), aggregated channel features (ACF) (Dollár et al. 2014) and deformal part models (DPM) (Felzenszwalb et al. 2010), etc., were used for feature extractions. Classifiers like support vector machines (SVMs) (Cortes and Vapnik 1995), adaptive boosting (AdaBoost) (Felzenszwalb and Huttenlocher 2000) and neural networks (Bishop 2006) used these features for object recognition (Kim et al. 2018). Enzweiler and Gavrila (2009) have surveyed on monocular pedestrian detection methodologies using regions of interest (ROI). Benenson et al. (2015) produced a survey report on computer vision-based pedestrian detection methodologies with 2D and 3D dataset using deep convolutional neural network (CNN).

Dalal and Triggs (2005) proposed a very effective feature histogram of gradients (HOG). In combination of HOG and simple linear support vector machine (SVM) on pedestrian database MIT test set, he achieved 100% result in detecting pedestrians. Introduction of HOG has promoted the development of pedestrian detection. Zhu et al. has improved the 70% of speed in pedestrian detection by using histogram technique to calculate HOG feature quicker. Further, Felzenswalb et al. (Urmson et al. 2008) improved the pedestrian detection accuracy by combining HOG with deformable part model (DPM) algorithm.

Papageorgiou et al. (Mateus et al. 2019) proposed learning at multiple stages using adaptive combination of classifiers (ACC) formulating a hierarchical architecture. ACC models are used to train features of the human body parts separately ensuring their geometric fixture. Viola et al. (Wang et al. 2017) worked on integrating image intensity and motion information from a video sequence for pedestrian detection system usage. Ronfard et al. (Xiao et al. 2017) has used first- and second-order Gaussian filters incorporating SVM-based limb classifiers in a dynamic programming framework in building articulated body detector. Similar type of framework is done by Huttenlocher et al. (Bayoumi et al. 2019) for modeling body shapes and physical motions.

Recent developments in neural networks have improved the accuracy beyond the usage of handcrafted features in recognition and detection (Kim et al. 2018). On the other hand, the recent development of convolutional neural network (CNN) using deep convolutional features has already improved accuracy far beyond the previous methods of using handcrafted features both in recognition and detection. Donahue et al. (Jia et al. 2014) has shown that CNN has ranked 5 in the classification techniques with error rate of 15.3%.

Zhang et al. (2019) have discussed the on the state-of-the-art in deep learning techniques with its potential in networking applications. Li et al. (2016) proposed a multilayer deep convolutional neural network for pedestrian detection. Brunetti et al. (2018) discussed CNN architectures such as AlexNet (Bartlett et al. 2003), VGG-16 (Levinson et al. 2011), InceptionV3 (Geronimo et al. 2010) and Resnet50 (Dollar et al. 2012) were typically trained on a large image dataset, e.g., ImageNet (Krizhevsky et al. 2012). He proved that spatial hierarchy of features learned by a deep CNN pretrained on a large and general original dataset is efficient for detection (Hinton et al. 2012).

3 Experimental study

3.1 Methodology

An optimized fully connected convolution neural network (FCNN) based on VGG-16 (Mateus et al. 2018; Mohan et al. 2001), is trained end to end for the task of classification which has the ability to classify an input image into pedestrian and non-pedestrian classification (Overett et al. 2008; Ronfard et al. 2002). First the proposed optimized model is developed by using deep learning libraries (Spencer et al. 2019; Tian et al. 2015). The experiments were conducted in this proposed work to evaluate the performance of proposed deep CNN model (OVGG-16) (Viola et al. 2003; Wagner et al. 2016) with a machine learning model (HMPD) (Wu et al. 2016a, b; Zeng et al. 2013) and pretrained deep learning model (VGG 16) (Zitnick and Dollar 2014).

3.2 Machine learning model for pedestrian detection

Before the advancements in deep learning, machine learning models plays a main role in the area of computer vision and image analysis (Becherer et al. 2019). In literature, histogram of oriented gradients (HOG) filter is used for quantifying (Benenson et al. 2012) and representing the texture and shape of the various objects present in the images (Dinakaran et al. 2020). HOG is a shape descriptor that counts the occurrences of gradient orientations in localized portions of an image (Ess et al. 2008). HOG descriptor gathers the gradients over the pixel of a small spatial region (Gavrila 1999) which makes the model to stuck between local maxima and local minima (Coelingh et al. 2010). This makes the model to depend on machine learning (ML) based models instead of HOG filters (Cong and Xiao 2014).

Among the machine learning models (He et al. 2016), support vector machine (SVM) shows massive performance in feature extraction (Hou et al. 2017). Comparing the performance of evolutionary algorithms (Huang et al. 2008, 2017), genetic algorithm (GA) is good in handling the noisy data (Ioffe and Forsyth 2001). In addition (Kim et al. 2017), GA has the good ability to make a global search and explore the entire search space with different kinds of crossover values (Lai and Teoh 2014). Since pedestrian dataset is shuffled with various external noises (Lavin and Gray 2015), a hybrid metaheuristic pedestrian detection (HMPD) algorithm is opted for PD (Li et al. 2018; Liu et al. 2015, 2019). The working of HMPD (Llorca et al. 2014) model is given in Fig. 1.

HMPD extracts the best out of SVM and genetic algorithm (GA). SVM is used to extract the healthy set of

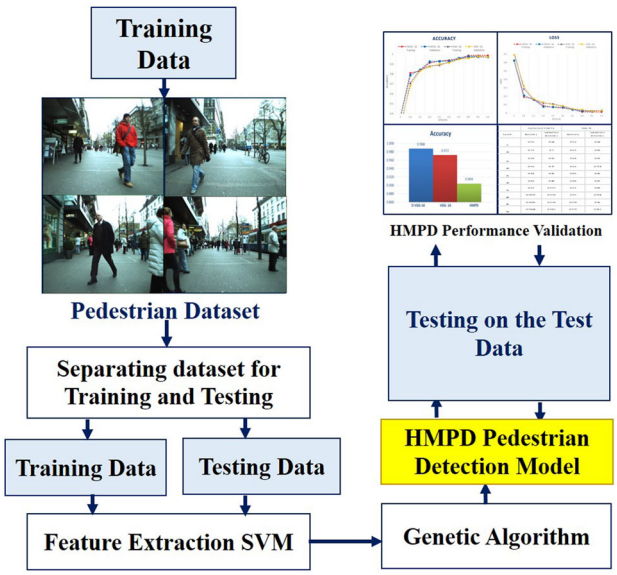

Fig. 1 HMPD model for pedestrian detection

features out of the available input features, and it is given as an input to the GA for further processing. The GA is used to obtain a set of features with large discrimination power. Attempts should be made to have multiple runs of the GA, to overcome the drawback of premature convergence in GA. It is necessary to have different values of crossover and mutation operators.

3.3 Pertained deep CNN model for pedestrian detection

A convolutional neural network (CNN) is a group of deep neural network architecture which is widely applied for analyzing the visual imagery (Dung and Anh 2019). A deep CNN architecture typically is made up of several convolutional layers and a connected layer for its end to end operation. Each convolutional block in deep CNN is composed of a convolutional layer, an activation unit and a pooling layer. Filters form the core part in the convolution layer that performs the convolution operation over the output of the preceding layers. Filters are used to extract the important features available over the large set of features. CNN's depute a collection of locally connected filters to identify the correlations between the different data regions (Zhang et al. 2019). The mathematical relationship for the standard convolution in the each location p_y of the output y performs the following operation that represents the receptive range of each neuron to inputs in a convolutional layer.

$$y(p_y) = \sum w(p_G) \cdot x(p_y + p_G) \tag{1}$$

where w represents the filter, G denotes the field in the convolution layer, p_G denotes the positions in the field G.

Here, the weights were shared across different locations of the input map. Figure 2 explains the operating principle of CNN architecture.

The operation of convolutional layer is shown in Fig. 2. The feature set from the input map passes through the convolution kernel/filter and maps with the output map. Specifically, the inputs of a 2D CNN layer are multiple 2D matrices with different channels each representing the different color. A convolutional layer employs multiple filters with different stride values, and it is shared across different locations of the input feature set to scan the inputs and produce output maps. The convolutional layer will require $M \times N$ filters to perform the convolution operation if the inputs and outputs are having M and N filters, respectively.

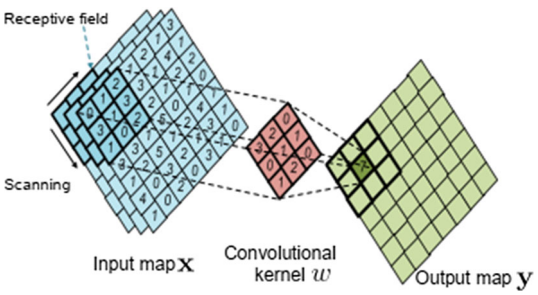

Fig. 2 Operating principle of convolution layer

There exist different configurations of CNN architectures. LeNet-5 is an early CNN architecture proposed for handwritten digit classification (Dung and Anh 2019). It is made up of two convolutional blocks. Other deeper CNN architectures include VGG-16, ResNet and AlexNet. These architectures improved the working efficiency by increasing the number of weight layers. ResNet is with 152 layer depth and considered to be very deep architecture which made it to won the first place in classification competitions including ILSVRC 2015 and ILSVRC & COCO 2015 (Felzenszwalb et al. 2010) VGG-16 is made up of 16 layers, and it is shallow when compared to ResNet, and it is a widely used convolutional architecture pretrained on ImageNet (Tian et al. 2015). The architecture of VGG-16 is given in Fig. 3.

VGG-16 is considered to be one of the excellent vision model architecture till date. Most unique thing about VGG-16 is that it focuses mainly on the convolution layers instead of working on a large number of hyper-parameters. All the convolution layers operates on the 3×3 filter with a stride 1 and always used same padding and maxpool layer of 2×2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. Finally, the spatial arrangement is made up of 2 FC (fully connected layers) followed by a softmax for output.

3.3.1 Optimized VGG-16 architecture for pedestrian detection

VGG-16 is deep architecture with 16 layers, and it is effective in handling the image data. The main challenge about VGG architecture is that it never concentrates to work on its hyper-parameters for optimization. Most of the deep learning architecture involves various hyper-parameters. Performance optimization of the deep architecture can be done by fine-tuning the hyper-parameters like depth, stride, padding and parameter sharing playing a main role in optimization of DL architecture. In this paper, a novel optimized VGG-16 (OVGG-16) architecture is proposed for enhancing the effectiveness of pedestrian detection. The spatial arrangement of OVGG-16 is optimized in comparison with the architectural arrangement of VGG-16.

The depth of the DL architecture remains as an important hyper-parameter. The proposed algorithm OVGG-16 makes use of the variable depths at each layer of

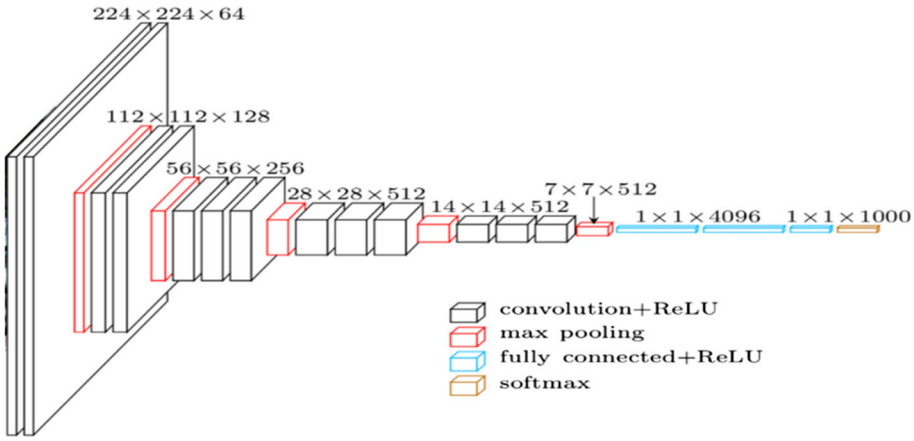

Fig. 3 VGG-16 architecture

convolution. The variable depths at each layer increase the nonlinearity of the input data which increase the learning rate of the architecture. The new architecture works with a stride value of 3. When the stride is increased, then filters jump at each convolution layer is also increased which results in producing the smaller output volumes spatially. In addition, the proposed OVGG-16 is controlled with zero padding to control the spatial size of the output volumes.

The main optimization parameter of the proposed algorithm is done by concentrating on parameter sharing. With this parameter sharing scheme, the convolution layer would have unique set of weights for a total value of input parameters. Parameter sharing enables the features to be learned in different spatial locations. Parameter sharing method is relatively reasonable. In a classification problem, the need to detect a horizontal edge is important at some location in the image, and it should intuitively be useful at some other location as well due to the translationally invariant structure of images. With the help of parameter sharing, there is no need to relearn to detect a horizontal edge at every one of the distinct locations in the

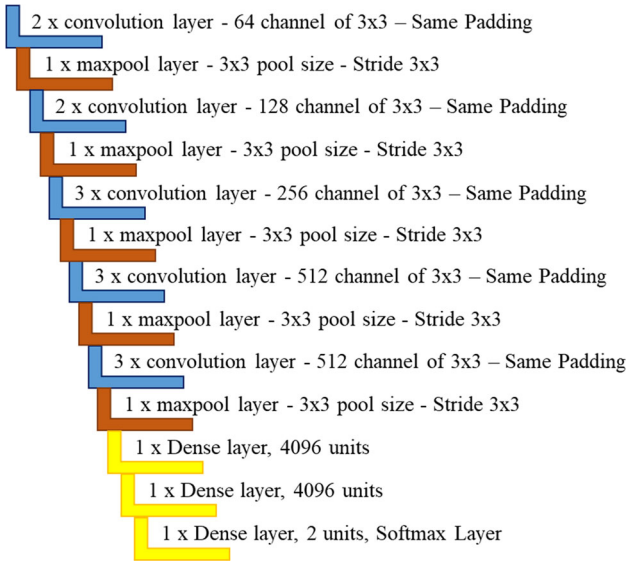

Fig. 5 OVGG-16 configuration procedure

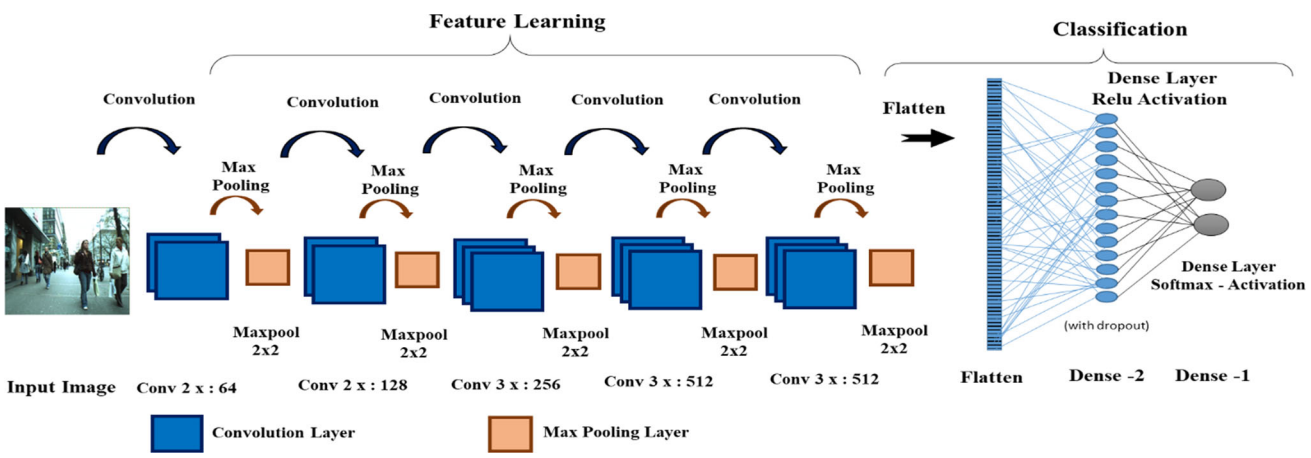

Fig. 4 OVGG-16 architecture

convolution layer output volume. The architecture of OVGG-16 is shown in Fig. 4.

To initialize the proposed model, the model needs to be configured with different layers representing OVGG-16. The implementation procedure of the OVGG-16 architecture is represented in Fig. 5.

The model is configured by placing the convolution layer and maxpool layer as mentioned in the configuration procedure. It is necessary to work on the mentioned hyperparameters while configuring the deep learning architecture, which results in generating the optimized model. The deep architecture is having three dense layers with a softmax layer at the end which supports for binary classification.

4 Optimized VGG-16 for pedestrian detection

In the present study, the performance of the proposed pedestrian detection model is compared with the hybrid machine learning model and pretrained VGG-16–CNN models. This section summarizes the training and validation procedure of OVGG-16 architecture for pedestrian detection. The details of the INRIA dataset used for training and validation of the proposed CNN architecture are represented in Table 1.

The dataset consists a total of 6817 images which includes 3578 images with pedestrian and 3239 non-pedestrian images. The size of the each image in the given dataset is 227 × 227. Among the total images in the dataset, 80% of the input data is used for training the model and 20% of the data is used for model validation. The architecture and the steps involved in the implementation of the proposed OVGG-16 are represented in Fig. 6.

Existing VGG-16 model is optimized with all the hyperparameter values to get an optimized VGG-16 architecture. The OVGG-16 pedestrian detection model represented in Fig. 6 is constructed as discussed in Sect. 3.3.1. The configured model splits the total input data into training set and validation set. The data available in the validation set is used to validate the model at each later of the construction. The developed OVGG-16 is capable of detecting the pedestrian from the new set of input images. The performance of the proposed model is verified with the new set of images available as the testing data (Fig. 7).

Table 1 Dataset details

Total images	Dataset details			Training—5453		Testing—1364	
	Image size	Pedestrian	Non pedestrian	Pedestrian	Non pedestrian	Pedestrian	Non pedestrian
6817	227 × 227	3578	3239	2872	2581	706	658

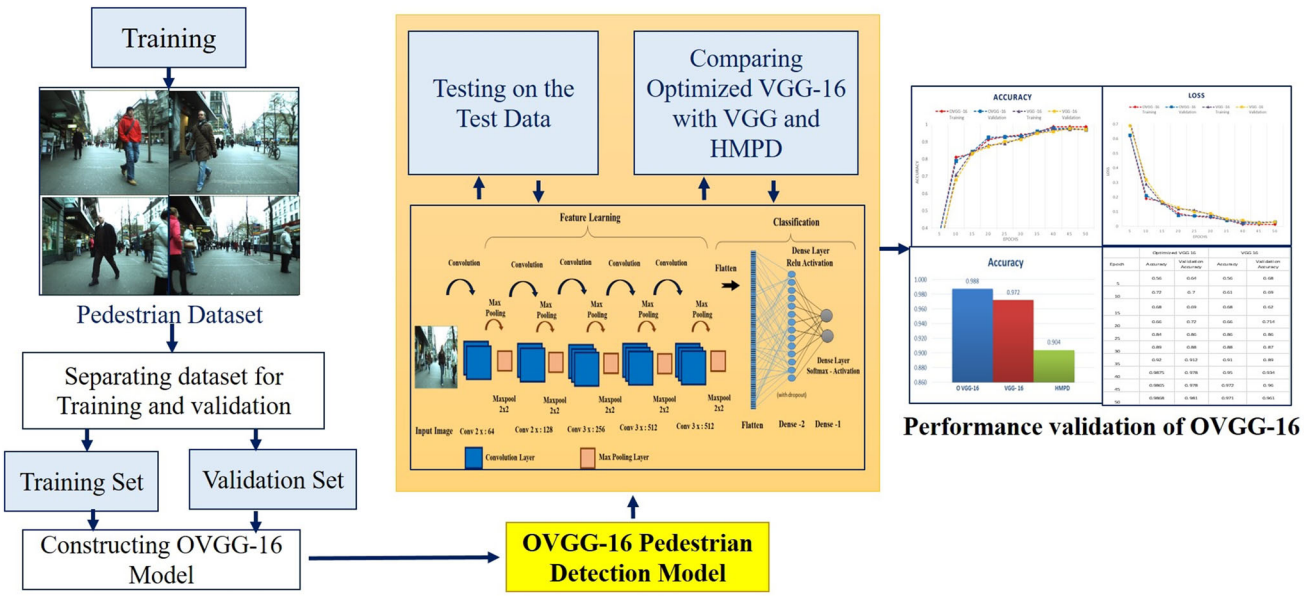

Fig. 6 Working architecture of proposed OVGG-16 pedestrian detection model

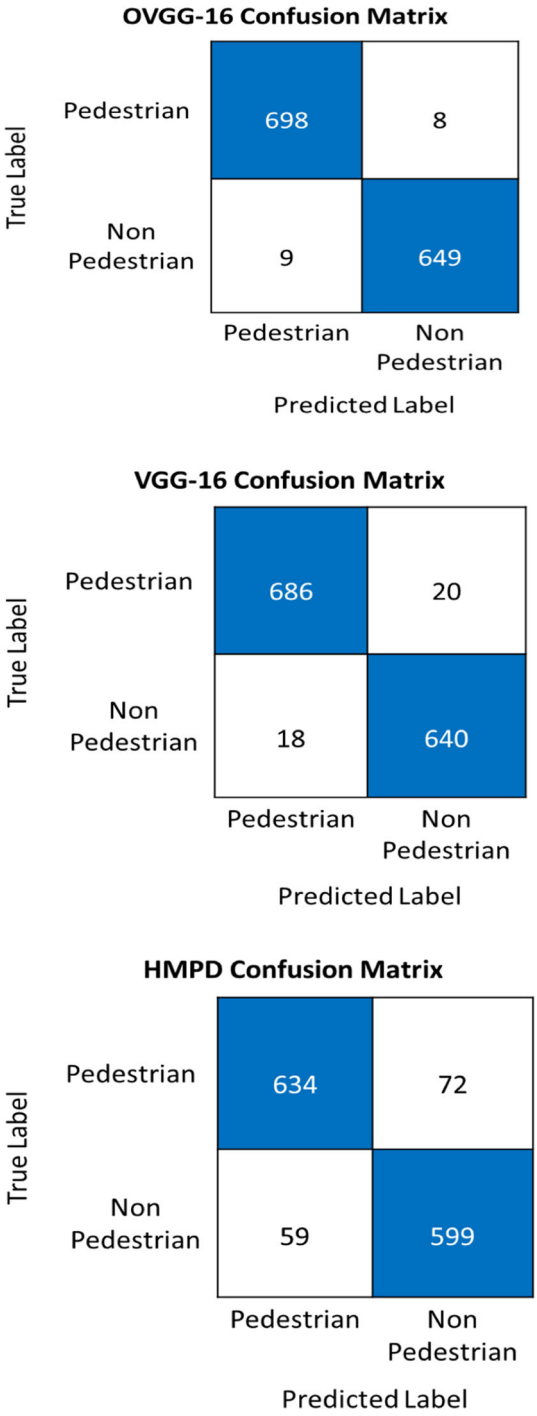

Fig. 7 Confusion matrices comparison

5 Results and discussion

Once the model is validated with the testing data, the performance of the proposed model is compared with existing machine learning and deep learning models. In this work, HMPD is used as a ML model, and VGG-16 is used

Table 2 Confusion matrix of OVGG-16, VGG-16 and HMPD

Confusion matrix					
Algorithm	TP	TN	FP	FN	Accuracy
OVGG-16	698	649	8	9	0.988
VGG-16	686	640	20	18	0.972
HMPD	634	599	72	59	0.904

Table 3 Performance measure

Algorithm	Accuracy	Precision	Recall	*F*-Measure
OVGG-16	0.988	0.989	0.987	0.988
VGG-16	0.972	0.972	0.974	0.973
HMPD	0.904	0.898	0.915	0.906

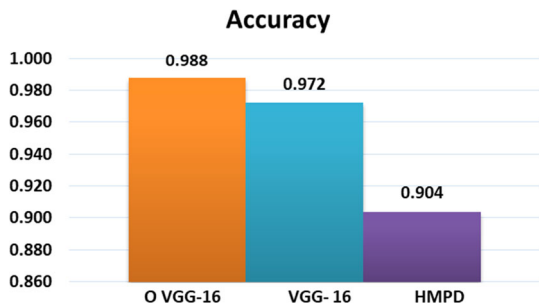

Fig. 8 Accuracy comparison

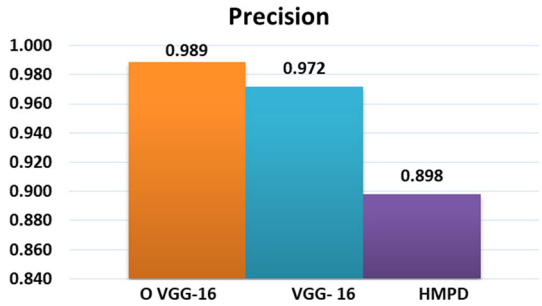

Fig. 9 Precision comparison

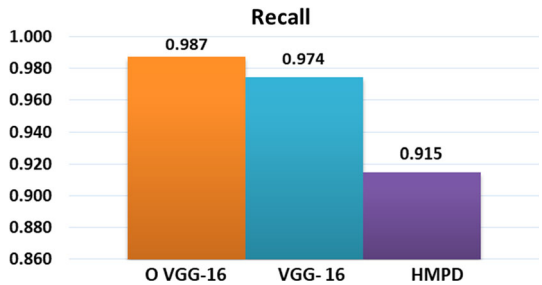

Fig. 10 Recall comparison

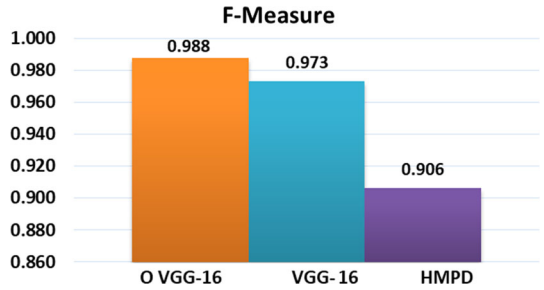

Fig. 11 *F*-Measure comparison

as a DL model for validating the performance of proposed OVGG-16 deep learning architecture.

5.1 Comparison of OVGG-16 with VGG-16 and HMPD

The contingency matrix comparing the performance of OVGG-16, VGG-16 and HMPD algorithm is represented in Table 2.

To analyze the performance of proposed OVGG-16 model with VGG-16 and HMPD Algorithm four different performance measures namely precision, recall, F1 score and accuracy were considered. The mentioned performance measures are calculated from the values given in the contingency matrix represented in Table 2 using Eqs. (2), (3), (4) and (5).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F1Score} = 2 * \frac{\text{Recall*Precision}}{(\text{Recall} + \text{Precision})} \tag{4}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

The performance measures derived by using the procedure are consolidated in Table 3. The visualization of all the performance measures is represented from Figs. 8, 9, 10 and 11.

OVGG-16 gives the overall accuracy of 98.8%, whereas VGG-16 produces the accuracy of 97.2% and HMPD produces the accuracy of 90.4%. In addition to accuracy,

Table 4 Performance analysis: OVGG-16 architecture

Optimized VGG-16				
Epoch	Training accuracy	Validation accuracy	Training loss	Validation loss
5	0.81	0.82	0.91	0.79
10	0.844	0.854	0.35	0.48
15	0.908	0.831	0.25	0.18
20	0.934	0.925	0.066	0.075
25	0.945	0.948	0.055	0.052
30	0.956	0.932	0.044	0.068
35	0.979	0.979	0.021	0.021
40	0.9874	0.983	0.0126	0.017
45	0.9875	0.986	0.0125	0.014
50	0.9873	0.979	0.0127	0.021

Table 5 Performance analysis: VGG-16 architecture

VGG-16				
Epoch	Training accuracy	Validation accuracy	Training loss	Validation loss
5	0.56	0.68	0.44	0.32
10	0.61	0.69	0.39	0.31
15	0.68	0.62	0.32	0.38
20	0.66	0.714	0.34	0.286
25	0.86	0.86	0.14	0.14
30	0.88	0.87	0.12	0.13
35	0.91	0.89	0.09	0.11
40	0.95	0.934	0.05	0.066
45	0.972	0.96	0.028	0.04
50	0.971	0.961	0.029	0.039

OVGG-16 performs better than VGG-16 and HMPD in terms of precision, recall and *F*-measure evaluation methods. From the mentioned statistics, it is clear that of OVGG-16 architecture holds better results in terms of all performance measures when compared with VGG-16 and HMPD algorithms. One more clear insight out of the evaluation measures is that deep learning models perform far better visual classification than the machine learning models and the hybrid models. Among the all deep learning architectures, CNN-based models are best suited for the visual recognition.

5.2 Performance comparison of OVGG-16 with VGG-16

VGG-16 is considered to be one of the excellent deep learning-based vision model architecture. Most unique thing about VGG-16 is that, it focuses on having convolution layers of 3 × 3 filter with a stride 1 and always used same padding and maxpool layer of 2 × 2 filter of stride 2. The proposed OVGG-16 optimizes the hypermeters in VGG-16 architecture to enhance the performance. Tables 4 and 5 consolidate the overall performance of OVGG-16 and VGG-16.

The performance of deep architectures are evaluated in terms of four different evaluation measures namely training accuracy, validation accuracy, training loss and validation loss. The performances of the algorithms are measured at each epoch value. Epoch is also a hyper-parameter value for deep learning architectures, which need to be defined before training a model. One epoch value is a measure, when an entire dataset is passed both forward and backward through the deep neural network only once. Since one epoch is too big to feed to the computer at once, epochs are divided in several smaller batches and then the performance is measured. The visualization of all the performance measures is shown in Figs. 12, 13, 14, 15, 16 and 17.

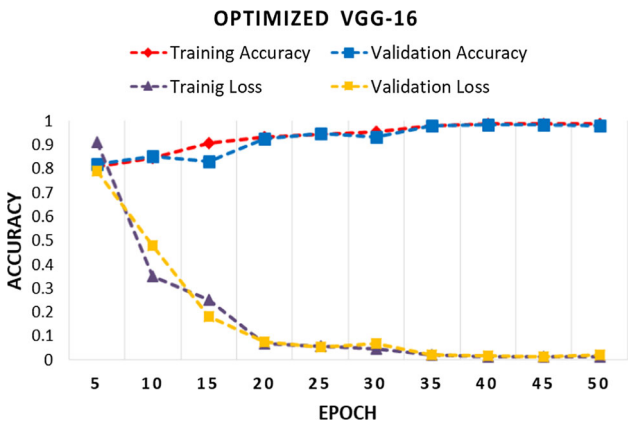

Fig. 12 OVGG-16 performance comparison

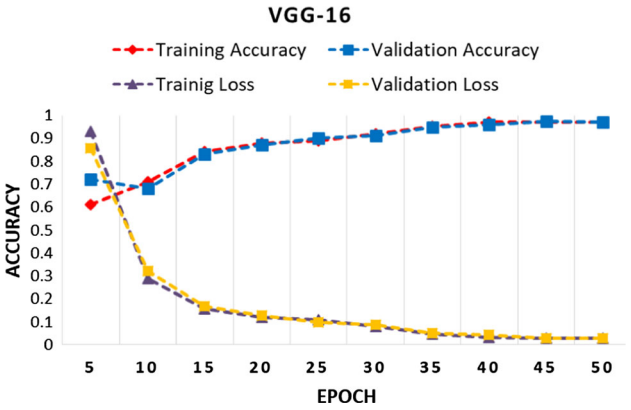

Fig. 13 VGG-16 performance comparison

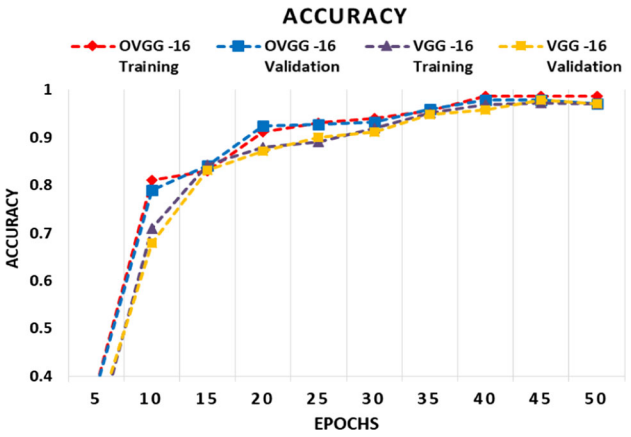

Fig. 14 Accuracy comparison

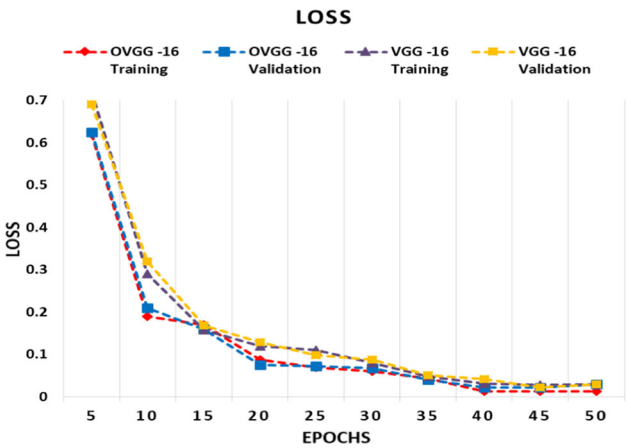

Fig. 15 Loss comparison

By optimizing the hyper-parameters, it is noted that the accuracy and loss values started stabilize at the minimum epoch values. The stabilization of accuracy and loss values of the deep architectures with respect to the epoch counts is

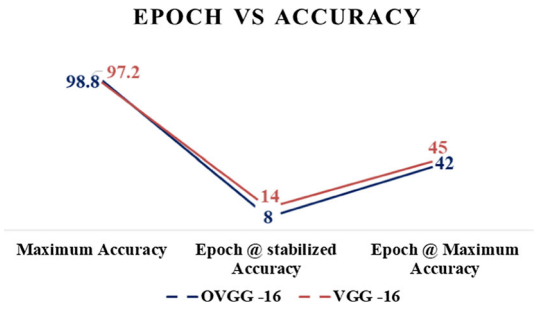

Fig. 16 Epoch versus accuracy comparison

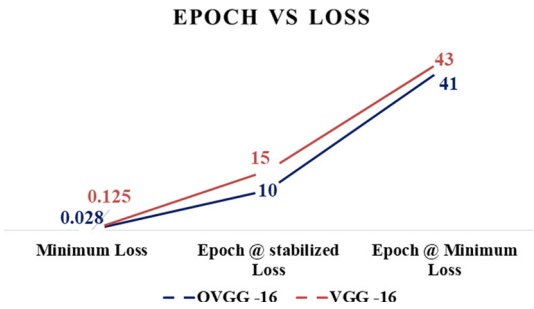

Fig. 17 Epoch versus loss comparison

graphically shown in Figs. 16 and 17. The accuracy values of OVGG-16 started to get stable at the epoch value of 8, whereas for VGG-16, it started to get stable at the epoch value of 14. Similarly, the loss values of OVGG-16 started to get stable at the epoch value of 10, whereas for VGG-16, it started to get stable at the epoch value of 15. From the above graphical representations, it is clear that optimized version of VGG-16 architecture produces the maximum accuracy value in detecting the pedestrians. The optimization of hyper-parameters plays a vital role in enhancing the efficiency of the deep architecture models.

6 Conclusion

In this present study, a vision-based model for pedestrian detection using CNN is proposed. Optimized version of VGG-16 (OVGG-16) is used as a core architecture for pedestrian detection which performed better than VGG-16 and hybrid machine learning models. The proposed pedestrian model is validated using a new set of pedestrian images and reached an average training and validation accuracy of 98.5%. The average training and validation loss of the proposed model is reduced to 0.015. The model is capable to identify the pedestrians in the given input frame. Although the proposed model can effectively identify the pedestrians, it is still challenging for the model to work on the frames which is accumulated with noise-like features. Therefore, the further studies should focus on training the model with noisy data which can enable the model to classify the input images more effectively.

Acknowledgements This work was supported by Korea Research Fellowship Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No. 2019H1D3A1A01101442). This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. 2019R1G1A1095215).

Data availability We used our own data.

Compliance with ethical standards

Conflict of interest The authors state that there is no conflict of interest.

Human and animal rights statement Humans and animals are not involved in this research work.

References

Angelova A, Krizhevsky A, Vanhoucke V, Ogale A, Ferguson D (2015) Real-time pedestrian detection with deep network cascades

Bartlett MS, Littlewort G, Fasel I, Movellan JR (2003) Real time face detection and facial expression recognition: development and applications to human computer interaction. In: Proceedings of the conference on computer vision and pattern recognition workshop (CVPRW), IEEE, vol 5, p 53

Bayoumi A, Karkowski P, Bennewitz M (2019) Speeding up person finding using hidden Markov models. Robot Auton Syst 115:40–48

Becherer N, Pecarina J, Nykl S, Hopkinson K (2019) Improving optimization of convolutional neural networks through parameter fine-tuning. Neural Comput Appl 31:3469–3479

Beck LF, Dellinger AM, O'Neil ME (2007) Motor vehicle crash injury rates by mode of travel, United States: using exposure-based methods to quantify differences. Am J Epidemiol 166(2):212–218

Benenson R, Mathias M, Timofte R, Van Gool L (2012) Pedestrian detection at 100 frames per second. In: Proceedings IEEE conference CVPR, pp 2903–2910

Benenson R, Omran M, Hosang J, Schiele B (2015) Ten years of pedestrian detection, what have we learned?. In: Agapito L, Bronstein M, Rother C (eds) Computer vision—ECCV 2014 workshops. Lecture notes in computer science, vol 8926. Springer, Cham

Bishop CM (2006) Pattern recognition and machine learning. Springer, New York

Boudjit K, Larbes C (2015) Detection and implementation autonomous target tracking with a quadrotor drone. In: Proceedings of the 12th international conference on informatics in control, automation and robotics (ICINCO), IEEE, pp 223–230

Brunetti A, Buongiorno D, Trotta GF, Bevilacqua V (2018) Computer vision and deep learning techniques for pedestrian detection and tracking: a survey. Neurocomputing 300:17–33

Cai Z, Fan Q, Feris RS, Vasconcelos N (2016) A unified multi-scale deep convolutional neural network for fast object detection. arXiv:1607.07155v1[cs.CV]

Coelingh E, Eidehall A, Bengtsson M (2010) Collision warning with full auto brake and pedestrian detection—a practical example of

automatic emergency braking. In: 13th international IEEE conference on intelligent transportation systems (ITSC)
Cong J, Xiao B (2014) Minimizing computation in convolutional neural networks. Artificial neural networks and machine learning. Springer, ICANN, pp 281–290
Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Computer vision and pattern recognition, IEEE computer society conference, vol 1, pp 886–893
Dinakaran RK, Easom P, Bouridane A, Zhang L, Jiang R, Mehboob F, Rauf A (2020) Deep learning based pedestrian detection at distance in smart cities. In: Bi Y, Bhatia R, Kapoor S (eds) Intelligent systems and applications. Springer, Cham
Dollar P, Wojek C, Schiele B, Perona P (2012) Pedestrian detection: an evaluation of the state of the art. IEEE Trans Pattern Anal Mach Intell 34(4):743–761
Dollár P, Appel R, Belongie S, Perona P (2014) Fast feature pyramids for object detection. IEEE Trans Pattern Anal Mach Intell 36:1532–1545
Dung CV, Anh LD (2019) Autonomous concrete crack detection using deep fully convolutional neural network. Autom Constr Elsevier 99:52–58
Enzweiler M, Gavrila DM (2009) Monocular pedestrian detection: survey and experiments. IEEE Trans Pattern Anal Mach Intell 31(12):2179–2195
Ess A, Leibe B, Schindler K, van Gool L (2008) A mobile vision system for robust multi-person tracking. In: CVPR
Felzenszwalb P, Huttenlocher D (2000) Efficient matching of pictorial structures. In: CVPR, Hilton Head Island, South Carolina, USA, pp 66–75
Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part based models. IEEE Trans Pattern Anal Mach Intell 32:1627–1645
Gall J, Lempitsky V (2013) Class-specific hough forests for object detection. In: Criminisi A, Shotton J (eds) Decision forests for computer vision and medical image analysis. Springer, London, pp 143–157
Gavrila DM (1999) The visual analysis of human movement: a survey. CVIU 73(1):82–98
Geronimo D, Lopez AM, Sappa AD, Graf T (2010) Survey of pedestrian detection for advanced driver assistance systems. IEEE Trans Pattern Anal Mach Intell 32(7):1239–1258
He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: CVPR
Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580
Hou Y-L, Song Y, Hao X, Shen Y, Qian M (2017) Multispectral pedestrian detection based on deep convolutional neural networks. In: Proceedings of IEEE international conference on signal processing, communications and computing, pp 1–4
Huang K, Wang L, Tan T, Maybank S (2008) A real-time object detecting and tracking system for outdoor night surveillance. Pattern Recognit 41:432–444
Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A et al (2017) Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 3296–3305
Ioffe S, Forsyth DA (2001) Probabilistic methods for finding people. IJCV 43(1):45–68
Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on multimedia, ACM, pp 675–678
Kim JH, Hong HG, Park KR (2017) Convolutional neural network-based human detection in nighttime images using visible light camera sensors. Sensors 17:1–26
Kim JH, Batchuluun G, Park KR (2018) Pedestrian detection based on faster R-CNN in nighttime by fusing, deep convolutional features of successive images. Expert Syst Appl 114:15–33
Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of the advances in neural information processing systems, pp 1097–1105
Lai CQ, Teoh SS (2014) A review on pedestrian detection techniques based on histogram of oriented gradient feature. In: Proceedings of IEEE conference on research and development (SCOReD), pp 1–6
Lavin A, Gray S (2015) Fast algorithms for convolutional neural networks. arXiv:1509.09308v2[cs.NE]
Levinson J, Askeland J, Becker J, Dolson J, Held D, Kammel S, Kolter JZ, Langer D, Pink O, Pratt V et al (2011) Towards fully autonomous driving: systems and algorithms. In: Proceedings of the intelligent vehicles symposium (IV), IEEE, pp 163–168
Li H, Wu Z, Zhang J (2016) Pedestrian detection based on deep learning model. In: 9th international congress on image and signal processing, biomedical engineering and informatics, IEEE, pp 796–800
Li J, Liang X, Shen S, Xu T, Feng J, Yan S (2018) Scale-aware fast R-CNN for pedestrian detection. IEEE Trans Multimed 20:985–996
Liu D, Wang X, Song J (2015) A robust pedestrian detection based on corner tracking. In: International conference on information, science and technology, pp 207–211
Liu W, Liao S, Hasan I (2019) Center and scale prediction: a box-free approach for object detection. arXiv:1904.02948v2[cs.CV]
Llorca DF, Parra I, Quintero R, Fernández C, Izquierdo R, Sotelo M (2014) Stereo-based pedestrian detection in crosswalks for pedestrian behavioural modelling assessment. In: International conference information in control, automation and robotics, pp 102–109
Majaranta P, Bulling A (2014) Eye tracking and eye-based human–computer interaction. In: Proceedings of the advances in physiological computing, Springer, pp 39–65
Mateus A, Ribeiro D, Miraldo P, Nascimento JC (2018) Efficient and robust pedestrian detection using deep learning for human-aware navigation. Robot Auton Syst. arXiv:1607.04441v3[cs.RO]
Mateus A, Ribeiro D, Miraldo P, Nascimento JC (2019) Efficient and robust pedestrian detection using deep learning for human-aware navigation. Robot Auton Syst 113:23–37
Mohan A, Papageorgiou C, Poggio T (2001) Detection in images by components. PAMI 23(4):349–361
Overett G, Petersson L, Brewer N, Andersson L, Pettersson N (2008) A new pedestrian dataset for supervised learning. In: Proceedings of IEEE intelligent vehicles symposium, pp 373–378
Rautaray SS, Agrawal A (2015) Vision based hand gesture recognition for human computer interaction: a survey. Artif Intell Revol 43:1–54
Rhodin H, Robertini N, Casas D, Richardt C, Seidel H-P, Theobalt C (2016) General automatic human shape and motion capture using volumetric contour cues. In: Proceedings of the European conference on computer vision, Springer, pp 509–526
Ronfard R, Schmid C, Triggs B (2002) Learning to parse pictures of people. In: The 7th ECCV, Copenhagen, Denmark, vol IV, pp 700–714
Shen J, Xiong X, Xue Z, Bian Y (2019) A convolutional neural-network-based pedestrian counting model for various crowded scenes. Comput Aided Civ Infrastruct Eng 34:897–914

Spencer BF Jr, Hoskere V, Narazaki Y (2019) Advances in computer vision-based civil infrastructure inspection and monitoring. Engineering 5:199–222

Tian Y, Luo P, Wang X, Tang X (2015) Deep learning strong parts for pedestrian detection. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1904–1912

Unies N, U.N.E.C. for Europe et al (2015) Statistics of road traffic accidents in Europe and North America. United Nations, New York

Urmson C et al (2008) Self-driving cars and the urban challenge. IEEE Intell Syst 23:66–68

Viola P, Jones MJ, Snow D (2003) Detecting pedestrians using patterns of motion and appearance. In: The 9th ICCV, Nice, France, vol 1, pp 734–741

Wagner J, Fischer V, Herman M, Behnke S (2016) Multispectral pedestrian detection using deep fusion convolutional neural networks. In: Proceedings of European symposium on artificial neural networks, computational intelligence, and machine learning, pp 509–514

Wang L, Ouyang W, Wang X, Lu H (2015) Visual tracking with fully convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 3119–3127

Wang Y, Pierard S, Song-Zhi S, Jodoin P-M (2017) Improving pedestrian detection using motion-guided filtering. Pattern Recognit Lett 96:106–112

Wu J, Rehg JM (2011) CENTRIST: a visual descriptor for scene categorization. IEEE Trans Pattern Anal Mach Intell 33:1489–1500

Wu Z, Yuan J, Zhang J, Huang H (2016a) A hierarchical face recognition algorithm based on humanoid nonlinear least-squares computation. J Ambient Intell Humaniz Comput 7:229–238

Wu Z, Yu Z, Yuan J, Zhang J (2016b) A twice face recognition algorithm. Soft Comput 20(3):1007–1019

Xiao T, Li S, Wang B, Lin L, Wang X (2017) Joint detection and identification feature learning for person search. arXiv:1604.01850v3[cs.CV]

Zeng X, Ouyang W, Wang X (2013) Multi-stage contextual deep learning for pedestrian detection. In: ICCV

Zhang C, Patras P, Haddadi H (2019) Deep learning in mobile and wireless networking: a survey. IEEE Commun Surv Tutor 21:2224–2287

Zitnick CL, Dollar P (2014) Edge boxes: locating object proposals from edges. In: ECCV