



A new hybrid mobile CNN approach for crosswalk recognition in autonomous vehicles

Gürkan Doğan¹ · Burhan Ergen²

Received: 9 October 2022 / Revised: 19 December 2023 / Accepted: 5 January 2024 /
Published online: 27 January 2024
© The Author(s) 2024

Abstract

While automobile transportation is increasing worldwide, it also negatively affects the safety of road users. Along with the neglect of traffic rules, pedestrians account for 22% of all highway traffic deaths. Millions of pedestrians suffer non-fatal injuries from these accidents. Most of these injuries and deaths occur at crosswalks, where the highway and pedestrians intersect. In this study, deep learning-based a new hybrid mobile CNN approaches are proposed to reduce injuries and deaths by automatically recognizing of crosswalks in autonomous vehicles. The first of these proposed approaches is the HMCNet approach, which is a hybrid model in which the MobileNetv3 and MNasNet CNN models are used together. This model achieves approximately 2% more accuracy than the peak performance of the lean used MobileNetv3 and MNasNet models. Another proposed approach is the FHMCNet approach, which increases the success of the HMCNet approach. In the FHMCNet approach, L SVC feature selection method and SVM classification method are used in addition to HMCNet. This approach increased the classification success of HMCNet by more than approximately 2%. Finally, the proposed FHMCNet offered approximately 3% more classification accuracy than state-of-the-art methods in the literature.

Keywords Crosswalk recognition · Intelligent transportation systems · Deep learning · Mobile CNN · FHMCNet · Computer vision

1 Introduction

As worldwide automobile transport continues to increase, it also severely impacts the safety of vulnerable road users (i.e. pedestrians, cyclists, and motorcyclists). Violation of traffic rules is the main source of all traffic accidents. According to the WHO, the

✉ Gürkan Doğan
gurkandogan@munzur.edu.tr
Burhan Ergen
bergen@firat.edu.tr

¹ Department of Computer Engineering, Faculty of Engineering, Munzur University, Tunceli, Turkey

² Department of Computer Engineering, Faculty of Engineering, Firat University, Elazig, Turkey

number of death in highway traffic reached 1.35 million in 2016 [1]. Each year, 270 thousand of these deaths are pedestrian deaths. Globally, pedestrian deaths are 22% of all road traffic deaths. In some countries, this rate is up to two-thirds. On the other hand, millions of pedestrians are non-fatally injured in traffic accidents [2]. Most of all these injuries and deaths occur at crosswalks, where vehicles and pedestrians intersect on the highway [3]. In this context, detecting and recognizing the crosswalk is an important research topic. Crosswalks can be detected from several perspectives: top view or satellite view, from behind the windshield or from the top of the vehicle, and pedestrian perspective. Each perspective has a different purpose. Although the crosswalk recognition problem can be addressed from all these perspectives, the scope of this study is limited only to the perspective of the cockpit and pedestrian view.

Although there are many studies in the literature based on convolutional neural networks [4–7], a few approaches focusing on pedestrian crossing recognition have been proposed. To assist visually impaired pedestrians, algorithms and approaches that can detect crosswalks for cellular and smart mobile phones are proposed in [8–11]. In another study, Herumurti et al. [12] proposed a new approach to detect zebra crosswalks from satellite images, combining the SURF method with a circle mask. Zhai et al. [13] present a new approach for crosswalk detection and localization based on MSER and ERANSAC. Foucher et al. [14] present a new algorithm to classify and detect crosswalks from urban images. Berriel et al. [15] proposed a system that can automatically train the images obtained from satellite images and annotations with deep learning models to classify zebra crossings. In another study, [16] proposed a classification system in which different datasets are used together to classify the crosswalk in real-time.

Along with the advancement of technology, the decrease in the cost of hardware, and the supply of mobility, the use of low-power devices in daily life has become quite common. In addition, the successful use of deep learning applications [7, 17–22] on these devices attracts the attention of researchers. MobileNetv3 [23] and MNasNet [24] are one of the CNN models that can be used on low-power devices.

In this study, a dataset combining real environment images obtained from automobile cockpit view and the pedestrian view is used to recognize zebra crosswalk in autonomous vehicles. Thus, this dataset has been made useful for automobile drivers and autonomous driving. A new hybrid model based on deep learning has been proposed to recognize the crosswalk. The proposed model consists of two different steps. Firstly, the intersection of the features obtained from the last convolution layers of the MobileNetv3 Small model (henceforth it will be called MobileNetv3) and the MNasNet model is taken. In order to do this, the number of filter output the last convolution layer of the MNasNet model was changed to 1024. Then, the intersecting features with the features in the last convolution layer of MobileNetv3 are added element-wise. Besides, training was carried out by adding the GAP layer and the Fully Connected Layer, respectively. Secondly, a new feature set was created by selecting features from the GAP layer by using the LSVC feature selection method. Finally, the new feature set is classified with the SVM classifier. According to the experimental results, the classification success seems to increase very well.

Other sections of this paper are organized as follows: In Section 2, contains information about the crosswalk dataset, the MobileNetv3 and MNASNET CNN models, the LSVC feature selection method, and the SVM classifier. In Section 3, the proposed methods are presented, while in Section 4, the experimental strategy and its results are presented. Finally, in Section 5, the general conclusions are given.

2 Materials and methods

2.1 Crosswalk dataset

IARA [16] and another crosswalk dataset [25] consisting of zebra crosswalk images were used to train and evaluate the proposed model. The IARA dataset created by Berriel et al. was obtained using the Intelligent Autonomous Robotic Automobile (IARA) developed by the High-Performance Computing Laboratory. To collect the data set, the camera, a Bumblebee XB3, was mounted on top of the car and facing forward. Since this data set was recorded during weekdays, it also includes ordinary traffic. The dataset contains a total of 12,441 images recorded in a real traffic environment in Vitória, the capital city of Espírito Santo. All the images were manually labeled with two different labels: crosswalk and non-crosswalk. Two subjective criteria were used for manual annotations: first, the crosswalk in the image frame should cover a reasonable area of the object, i.e. very distant crosswalks deemed too small should be removed, second, the crosswalk in the image frame should preferably be in the direction of traffic flow. The IARA data set created according to these criteria consists of 21.2% positive samples and 78.8% negative samples.

Another crosswalk dataset [25] created by Silva et al. contains 600 images to detect zebra crosswalks of visually impaired pedestrians. The image frames in this dataset are 1280×720 pixels in dimension. The dataset consists of images obtained from different angles horizontally and vertically in a real environment from Fortaleza-CE city of Brazil. In the original of this dataset, it has 4 different classes: 150 from the front, 150 from the right, 150 from the left crosswalk, and 150 non-crosswalk images. Since there are two different classes in the IARA dataset: crosswalk and non-crosswalk, the other crosswalk dataset was used in this study by setting it in two different classes: crosswalk (450 images) and non-crosswalk (150 images). Otherwise, the IARA dataset cannot be combined with the other crosswalk dataset.

In this study, due to hardware limitations, about 20% of the IARA dataset was used instead of the entire data set. 20% was randomly splitted using the *train_test_split()* function of the scikit-learn library. The random state parameter in this function is set to 7. After separating the IARA dataset, a total of 528 crosswalk images and 1961 non-crosswalk images remained to be used in the training and testing process of the proposed model. However, since the ratio of crosswalk images in the IARA dataset to non-crosswalk images is low and the crosswalk object in an image frame is far away, labeling it as non-crosswalk will reduce the accuracy of the proposed model, so a new dataset was created by including the images in the other crosswalk dataset [25] into IARA dataset. 450 crosswalk images in the other crosswalk dataset were included in the 528 crosswalk images in IARA, and 150 non-crosswalk images in the other crosswalk dataset [25] were included in the 1961 non-crosswalk images in IARA. The new dataset contains a total of 978 crosswalk images and 2111 non-crosswalk images. The dimensions of the images in this dataset are set to 224×224 pixels. To perform the training and testing of the models, this new data set was randomly divided into 80% training set and 20% test set, with the help of the *train_test_split()* function in the scikit-learn library and by setting the random state parameter to 7. Some sample images from this new dataset are given in Fig. 1.

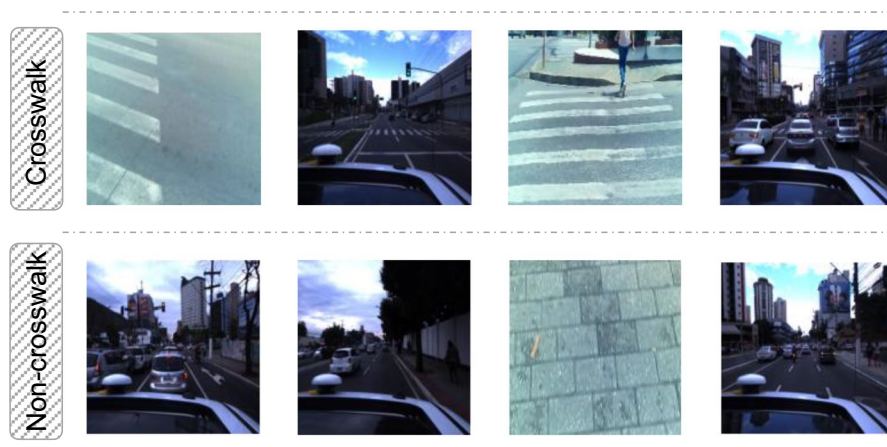


Fig. 1 Some sample images from the new crosswalk dataset

2.2 MobileNetv3 model

MobileNetv3 [23] is divided into two different models; Mobilenetv3-Small and MobileNetv3-Large. These models can be used in mobile devices according to low and high resource usage situations. The biggest difference between the original MobileNetv3-Small model and the MobileNetv3-Large model is that there are 4 fewer layers. Other than that, the structure of the general architecture of both models is almost the same. The models were built by implementing platform-aware NAS and NetAdapt for network searching and making some improvements. In this study, the MobileNetv3-Small model (henceforth it will be called MobileNetv3), which needs low hardware resources, is used. The MobileNetv3 model is built on the basis of some layers in the MobileNetv2 [26] and MNASNet [24] models. The specification of the MobileNetv3 CNN model is presented in Table 1. This table is important in terms of showing the size and number of images or features that each layer receives as input, kernel size and operator, expansion factor, number of output features, whether squeeze-and-excitation block is used, type of activation function, and stride size. The layer referred to as Bneck in this table is the abbreviated expression of Bottleneck. The bottleneck layer is formed by the succession of 1×1 expansion convolution, depthwise separable convolution and 1×1 projection convolution, respectively. This bottleneck is also called inverted residual block and it contains the squeeze-and-excitation block structure and residual connection. Depthwise separable convolution includes a depth convolutional kernel applied to each channel, a batch normalization layer (BN), and a 1×1 point convolutional kernel with ReLU or h-swish activation functions. The h-swish activation function, which reduces the number of parameters of the network model, can be defined as in Eq. (1):

$$h-swish(x) = x \cdot \frac{ReLU6(x+3)}{6} \quad (1)$$

Besides swish non-linearity, hard sigmoid is used to calculate squeeze-and-excitation status. The layers are also stacked with modified swish non-linearities. Depthwise separable convolution was preferred because it increases the computational speed by

Table 1 Specification for MobileNetv3-Small. Exp denotes expansion size. SE denotes whether there is a Squeeze-And-Excite in that block. NL denotes the type of nonlinearity used. Here, HS denotes h-swish and RE denotes ReLU. NBN denotes no batch normalization. S denotes stride

Input	Operator	Exp	Out	SE	NL	S
$224^2 \times 3$	Conv2d, 3×3	-	16	-	HS	2
$112^2 \times 16$	Bneck, 3×3	16	16	✓	RE	2
$56^2 \times 16$	Bneck, 3×3	72	24	-	RE	2
$28^2 \times 24$	Bneck, 3×3	88	24	-	RE	1
$28^2 \times 24$	Bneck, 5×5	96	40	✓	HS	2
$14^2 \times 40$	Bneck, 5×5	240	40	✓	HS	1
$14^2 \times 40$	Bneck, 5×5	240	40	✓	HS	1
$14^2 \times 40$	Bneck, 5×5	120	48	✓	HS	1
$14^2 \times 48$	Bneck, 5×5	144	48	✓	HS	1
$14^2 \times 48$	Bneck, 5×5	288	96	✓	HS	2
$7^2 \times 96$	Bneck, 5×5	576	96	✓	HS	1
$7^2 \times 96$	Bneck, 5×5	576	96	✓	HS	1
$7^2 \times 96$	Conv2d, 1×1	-	576	✓	HS	1
$7^2 \times 576$	Pool, 7×7	-	-	-	-	1
$1^2 \times 576$	Conv2d, 1×1 , NBN	-	1024	-	HS	1
$1^2 \times 1024$	Conv2d, 1×1 , NBN	-	k	-	-	1

reducing the computational cost. The squeeze-and-excitation (SE) block is used during the training process to draw more attention to relevant features in each channel.

2.3 MNASNet model

The MNASNet [24] CNN model is a convolutional neural network model that optimizes accuracy and latency on low-power devices using reinforcement learning. The CNN model has a much better accuracy-latency trade-off than layer diversity in other resource-constrained models. Table 2 shows the specification of MNASNet. As seen in this specification, images with 224×224 pixel size and RGB channel as model input are passed through the convolution layer with a 3×3 dimensional kernel. While this convolution process does not have an expansion factor, it has 32 output channels. Also, this convolution process repeats itself only once and the number of strides is 2. Then,

Table 2 Specification for MNASNet. Exp denotes expansion factor. Ch denotes channel number. N denotes the number of blocks and S denotes stride

Input	Operator	Exp	Ch	N	S
$224^2 \times 3$	Conv2d, 3×3	-	32	1	2
$112^2 \times 32$	SepConv, 3×3	1	16	1	1
$112^2 \times 16$	MBConv, 3×3	3	24	3	2,1,1
$56^2 \times 24$	MBConv, 5×5	3	40	3	2,1,1
$28^2 \times 40$	MBConv, 5×5	6	80	3	2,1,1
$14^2 \times 80$	MBConv, 3×3	6	96	2	1,1
$14^2 \times 96$	MBConv, 5×5	6	192	4	2,1,1,1
$7^2 \times 192$	MBConv, 3×3	6	320	1	1
$7^2 \times 320$	Pooling, FC	-	k	-	-

the batch in this layer is subjected to normalization and ReLU6 activation process. The ReLU6 activation function can be formulated as in Eq. (2):

$$ReLU6(x) = \begin{cases} x, & \text{if } x \geq 0 \text{ and } x \leq 6 \\ 0, & \text{if } x < 0 \\ 6, & \text{if } x > 6 \end{cases} \quad (2)$$

The second layer included in this specification is SepConv. The abbreviation for the separable convolution block is SepConv. This block structure includes two-dimensional depthwise convolution operation, batch normalization, ReLU6, and two-dimensional convolution, batch normalization, and ReLU6 activation functions, respectively. The expansion factor, the number of repeats or number of blocks and stride size of this block structure are 1. The third layer in this specification is MBConv. The abbreviation for mobile inverted bottleneck convolution is MBConv. MBConv block includes 1×1 kernel size convolution, batch normalization, ReLU6 operation and $k \times k$ kernel size depthwise convolution, batch normalization and ReLU6 operation and 1×1 kernel size convolution, batch normalization, ReLU6 operation, respectively. In addition, the expansion factor, the number of output channels, the number of repetitions and the size of strides of this block structure are 3, 24, 3, {2, 1, 1}, respectively. Here, the number of repetitions and the size of the stride are directly related. For example, in the 1st repetition, this block structure is stride 2, while in the 3rd repetition, the stride size is 1.

In addition, RMSProp optimizer, batch normalization, and dropout regularization techniques were used in the training process of the MNasNet model built according to this specification.

2.4 Feature selection and classification method

SVM method [27, 28] is a supervised machine learning algorithm that can be used for classification and regression problems. The SVM is a binary classification method that attempts to distinguish between positive and negative samples in the sample set. In this method, a hyperplane is sought that separates the negative samples from the positive samples and the margin between the nearest negative and positive samples is maximized. On the other hand, the choice of kernel used in SVM is very important for the success of classification. The main purpose of using different kernels is to classify data or features in the best possible way.

In this study, among the feature selection methods, chi-squared, ANOVA F-value, mutual information and model selection techniques, as well as techniques such as RidgeCV, SGD Classifier, Random Forest Classifier and linear SVC, were tried iteratively. Since the combination of linear SVC and model selection technique provides the highest accuracy among these feature selection methods, this combination was used in the proposed method. For feature selection, model selection technique (SelectFromModel), one of the scikit-learn functions, was used along with the linear SVM method. In order to make feature selection, first the relevant features are trained with the linear SVM technique and the weight of each feature is determined. These weights indicate the importance of the features in the decision function of the linear SVM technique. That is, features with larger weights are more effective in the classification decision. Secondly, the model selection technique selects important features using the weights of the linear SVM technique and creates a simpler and more effective feature set by eliminating others. The selection technique from the model selects

this selection according to the determined threshold value. Features with weights above the threshold value are selected, while those below the threshold are eliminated. Thus, the prediction speed of the proposed model is increased with fewer features and the accuracy of the proposed model is increased by discarding unimportant features.

3 Proposed approach

In the proposed model, MobileNetv3 and MNasNet CNN models, which are more successful than other network models in performance metrics such as number of parameters, execution time and accuracy rate, were used as feature extractors. The proposed method basically consists of two different stages. In the first stage, element-based addition was performed with the conv-2 layer by taking the intersection of $7 \times 7 \times 1024$ dimensional features obtained from the conv-2 (Conv_2) and conv-34 (conv2d_34) layers of the MobileNetv3 and MNasnet CNN models, respectively. To determine the intersecting features, the element-wise equality of the feature maps obtained in these two layers is checked, and a boolean matrix of the same size is obtained. Then, the features in conv-34 layer are multiplied element-wise with the boolean matrix to obtain the intersection. Thus, attention is drawn to important features from feature maps obtained from different network models. Then, these intersecting features are element-wise added to the features in the conv-2 layer. The result of this addition is given as input to the Global Average Pooling (GAP) Layer. As the last layer, a new hybrid model was created by adding the full-connected layer. This model is named HMCNet, which is an abbreviation for hybrid mobile crosswalk networks. In the second stage, the last layer of the hybrid model, the fully connected layer, was removed and a new feature set was created by eliminating the unimportant features from the GAP layer by using the LSVC method. The LSVC feature selection method increases the prediction speed and improves the accuracy performance of the proposed approach. Finally, the new feature set was classified by the SVM method. This proposed approach is named FHMCNet. The schematic of the proposed approach is given in Fig. 2. The reason for applying the second stage in addition to the hybrid model in the first stage is to further increase the classification success.

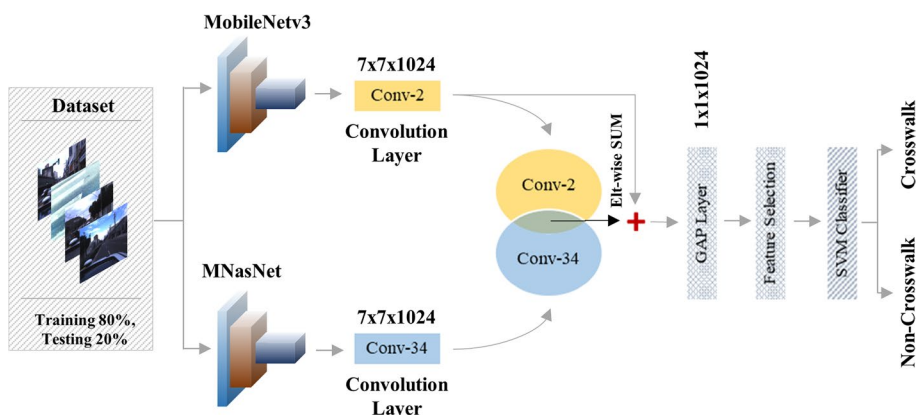


Fig. 2 Schematic of the proposed approach

4 Experiments

4.1 Experimental strategy

In this study, the batch size of MobileNetv3, MNasNet, and HMCNet models was set to 16, 8, and 8, respectively, considering the training process. Adam [29] optimization method was used, which gave the best results in all three models. Different learning rates were tried iteratively for all three models and 1×10^{-7} learning rate was used for MobileNetv3 and MNasNet models and 1.5×10^{-7} learning rate was used for the HMCNet model. Softmax is used as the output function for all three CNN models, while binary cross-entropy is used as the loss function. The training of all three models kept up to 750 epochs. Table 3 shows the hyperparameters used in the training process of CNN models associated with the proposed method. Dropout regularization technique was not used in these CNN models. On the other hand, feature selection and classification methods are also used in FHMCNet, which is the finally proposed approach. In the Linear SVM method used in the feature selection process, the value of 0.01 for the regularization parameter c , L1 (Lasso) for the penalty norm, False for the dual parameter, and 1×10^{-5} for the stop criterion tolerance are used. Except for these parameters, default values are applied. While the RBF kernel of SVM is used in classification, default parameter values are used in other parameters. On the other hand, the software and hardware features used in this study are as follows;

- Intel(R) Xeon(R) CPU @ 2.30 GHz,
- 24 GB RAM,
- Tesla P100-PCIE-16 GB,
- Python version 3.7.10,
- Tensorflow version 2.4.1,
- Keras version 2.4.3.

4.2 Experimental results and discussion

In the experimental study, four different classifications were made by using the MobileNetv3, MNasNet, HMCNet, and FHMCNet models/approaches and the crosswalk dataset, and the results were evaluated. The contribution of the LSVC feature selection method and SVM classification method was investigated in FHMCNet, which is the final proposed approach, unlike HMCNet.

Accuracy and loss graphs of training and validation sets are used to evaluate the performance of MobileNetv3, MNasNet, and HCMNet CNN models. When the MobileNetv3,

Table 3 Hyper parameters used by the CNN models in the training process in the proposed method

	Epoch	Batch	Optimizer	LR	Loss Function	Classification Function
MobileNetv3	750	16	Adam	1×10^{-7}	Binary Cross Entropy	Softmax
MNasNet		8				
HMCNet		8		1.5×10^{-7}		

MNasNet, and HMCNet CNN models are trained with the crosswalk dataset, the training and validation graphs are as in Fig. 3. The best training and validation harmony is seen in Fig. 3e, f, where the validation and loss graphs of the HMCNet CNN model. At the end of 750 epochs, the best loss value was obtained in MobileNetv3 and HMCNet models. The healthiest loss graph between these two models was obtained with HMCNet. As the epoch increases, it is seen that the margin increases in the training and validation loss graphs of MobileNetv3, while the loss graphs in HMCNet are almost the same.

Another way to compare the performance of MobileNetv3, MNasNet, and HMCNet CNN models is to evaluate the Receiver Operating Characteristic (ROC) curve and the area under the curve (AUC). Figure 4 presents the ROC curves of these CNN models. If we

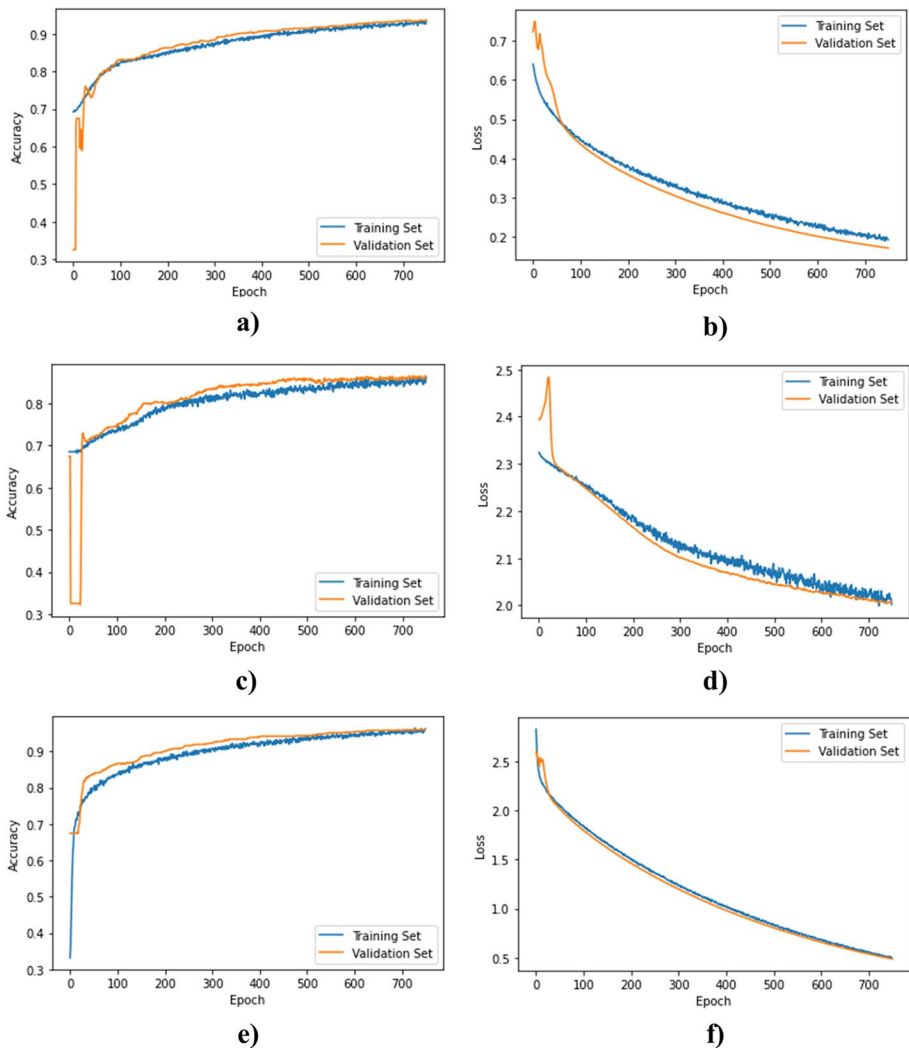
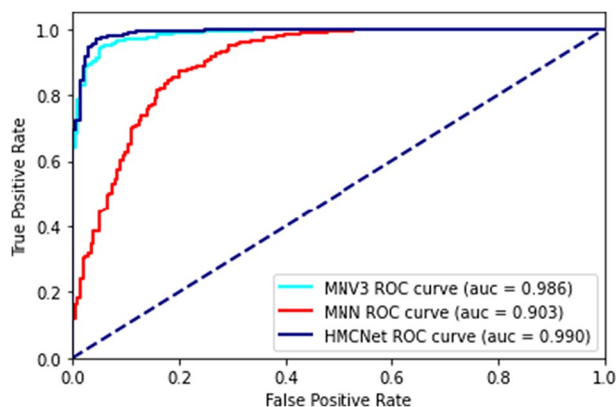


Fig. 3 Accuracy and loss plots of training and validation sets for cnn models; **a** MobileNetv3 Accuracy, **b** MobileNetv3 Loss, **c** MNasNet Accuracy, **d** MNasNet Loss, **e** HMCNet Accuracy, **f** HMCNet Loss

Fig. 4 ROC Curve for CNN models



pay attention to this graph, the worst AUC value (0.903) was obtained with the MNasNet model, while the best AUC value (0.99) was obtained with the HMCNet ROC curve.

Some of the metrics used to evaluate experimental results are also obtained from the complexity matrix. The values of precision (Prec.), recall (Rec.), f1-score (F1-Scr.), and accuracy (Acc.) are obtained from the following (3), (4), (5) ve (6) equations, respectively.

$$Prec = TP / (TP + FP) \quad (3)$$

$$Rec = TP / (TP + FN) \quad (4)$$

$$F1 - Scr = 2TP / (2TP + FP + FN) \quad (5)$$

$$Acc = (TP + TN) / (TP + FN + TN + FP) \quad (6)$$

TP used in these equations denotes true-positive, TN denotes true-negative, FP denotes false-positive, and FN denotes false-negative. The graphical representation of the obtained values of the complexity matrix metrics is given in Fig. 5. These graphs show the percentage values of precision, recall, f1-score, and overall accuracy metrics according to the methods. Besides, the crosswalk and non-crosswalk classes in the crosswalk dataset were used in the evaluation. The graph of the proposed FHMCNet approach is also included in this graph.

Another method of evaluating the performance of CNN models is by examining resource consumption. In Table 4, the execution time, classification accuracy, and model size of the CNN models are presented. The execution time of each model is the prediction time made by the model for a test image. The same test image was used for each model. The best execution time was achieved in the FHMCNet approach with 58 μsec . The least consumption of the storage resource was performed with the MobileNetv3 model.

When the relevant methods were evaluated according to the Precision chart (Fig. 5a), the highest success was achieved in the FHMCNet approach with 97.51%. The lowest success rate was 75.44% with the MNASNet model and crosswalk class. In the Recall (Fig. 5b) and F1-Score (Fig. 5c) charts, the highest success was obtained with FHMCNet for both classes, while the lowest success was obtained with the MNasNet method. In the overall accuracy chart (Fig. 5d), the highest accuracy rate was 97.73% with FHMCNet.

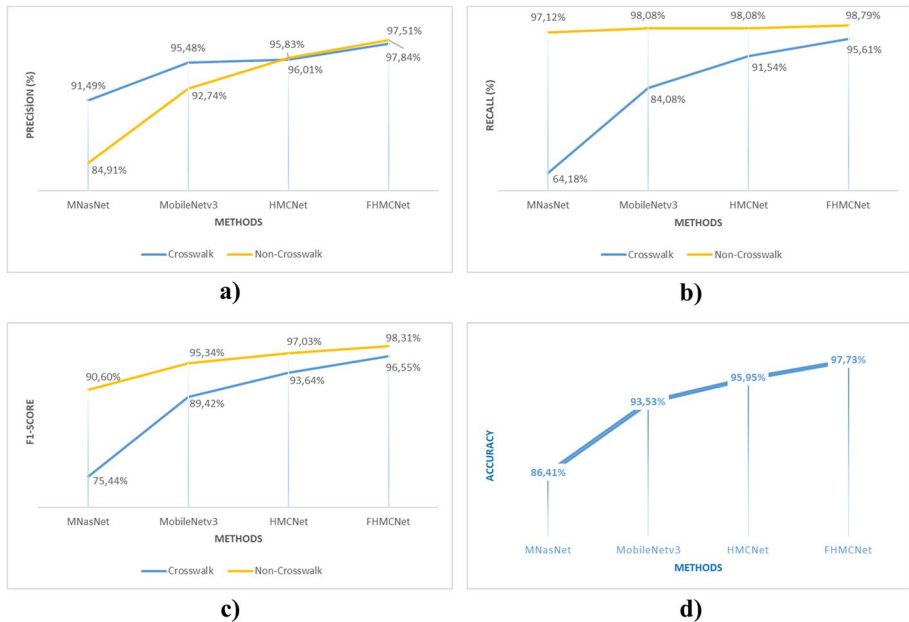


Fig. 5 Complexity matrix metrics according to the methods; **a** precision, **b** recall, **c** f1-score, **d** overall accuracy

Table 4 Performance comparison according to CNN methods

Models	Exe. time (ms)	Acc. (%)	Size (MB)
MobileNetv3	67.2	93.53	18
MNasNet	91.8	86.41	36
HMCNet	97.6	95.95	54
FHMCNet	0.58	97.73	-

Entries with bold characters indicate the best result obtained in the relevant metric

According to all these experimental results, the accuracy of HMCNet, which is the first of the proposed methods, has been increased by more than 2% according to MNASNet and MobileNetv3. The accuracy of FHMCNet, the second of the proposed methods, that is, the finally proposed approach, has been increased by more than 2% according to the HMCNet model. This result shows the positive effect of LSVC feature selection and SVM classifier applied to the HMCNet method. When the accuracy success of the MobileNet and MNASNet models is evaluated with the FHMCNet approach, which is finally proposed, it has achieved approximately 4% higher success than the MobileNetv3 CNN model, which provides the highest accuracy.

The comparison of the results of our study and other crosswalk studies in the literature is given in Table 5. Radványi et al. [30] are achieved a success rate of 89.2% when he trained and tested a private crosswalk dataset containing 83 images with the CNN method. Wang [10] achieved an average success of 93.9% when evaluating a private crosswalk dataset with 228 images using the SVM method. Berriel et al. [16] conducted a study using the deep CNN method and three different crosswalk datasets with different sizes. In this

Table 5 Comparing the results of other studies based on different datasets and parameters in the literature

Author	Year	Sample Size	Dataset	No. of Classes	Method	Accuracy (%)
Radványi [30]	2010	83	Private	2	CNN	89.2
Wang [10]	2014	228	Private	3	SVM	93.9
Berriel [16]	2017	657 K	GSV	2	Deep CNN	94.21
		12 K	IARA			93.06
		11 K	GOPRO			89.30
Tümen [31]	2020	296	Private	2	VP & CNN	94.73
Ni et al. [32]	2022	75 K	KITTI & Place365	5	Faster RCNN & Inception_v1 & Feature Fusion & Softmax	94.5
Wangsiripitak et al. [33]	2022	6 K	Private	2	DarkNet53 & PAFNet & YOLOv3	92.77
Bayat et al. [34]	2023	27 K	Private	24	ViT & Mask RCNN	95.0
HMCNet (Our)	-	3089	IARA & other [25]	2	Hybrid CNN	95.95
FHMCNet (Our)					Hybrid CNN & LSVC & SVM	97.73

Entries with bold characters indicate the best result obtained in the relevant metric

study, the highest success was obtained as 94.21% with the GSV dataset consisting of 657 thousand images. Tümen et al. [31] conducted a study using The Vanishing Point (VP) and CNN method together with a private crosswalk dataset consisting of 296 images. In this study, the average highest success rate for crosswalk recognition was 93.10%. Ni et al. [32] made improvements by integrating the residual connection module based on spatial attention into the Faster RCNN model. In addition, it has made an improvement to the Inception_v1 model by mixing ELU and Leaky ReLU functions. Then, the features obtained from the improved Faster RCNN are passed through some functions and local features are extracted. In parallel, the extraction of global features obtained from the improved Inception_v1 model is provided. Then, the global features and local features are combined and passed through the fully connected layer and softmax layer. Ni et al. created a new dataset by selecting images from the publicly available KITTI and Place365 datasets for the training and testing of their proposed method. While this data set includes scene categories such as crosswalk, gas station, parking lot, highway, and street, each category consists of 15 thousand images. The reason for making this choice is to increase the accuracy and efficiency of the method. They reached 94.5% accuracy for 5 categories, including the crosswalk category, with the method and data set they proposed. Wangsiripitak et al. [33] proposed a new model that they developed based on the DarkNet53 model and strengthened it with PANet and YOLOv3 models. Their proposed model uses YOLOv4 pre-trained weights with transfer learning with their own dataset. They took image frames of the crosswalk from a vehicular camera and created a dataset with a total of 6128 frames. When they trained and tested this dataset with the proposed method, they reached 92.77% accuracy. Bayat et al. [34] used the vision transformer (ViT) model, which uses the self-attention mechanism to capture the spatial dependencies of the input images and learn important features. Next, Mask RCNN was used to perform object classification. The dataset was video-collected via an RGB camera mounted on the chest at a 30-degree angle while walking on the main campus of the University of Central Florida at various times in both indoor and outdoor environments. Then, image frames were extracted from the collected videos and 27,867 image frames and 24 different object categories were obtained. As a result of the training and testing carried out with the proposed method and data set, 95.0% accuracy was achieved for 24 categories, including the crosswalk category. A total of 3089 images obtained from the IARA [16] and other crosswalk [25] datasets were used for both approaches we proposed. A success rate of 95.95% was achieved with the HCMNet approach, which is a hybrid CNN-based method we proposed. This result is the highest crosswalk recognition success among other studies in the literature ([10, 16, 30–34]). Another approach that we have finally proposed, FHMCNet, is a hybrid CNN-based approach in which the LSVC feature selection method and the SVM method are used together. In the study using this approach, a very high crosswalk recognition success rate of 97.73% was achieved. This result is the highest crosswalk recognition success among other studies in the literature ([10, 16, 30–34]) and the HCMNet study.

5 Conclusions and future works

In this study, a new hybrid mobile CNN method is proposed to recognize crosswalk in autonomous vehicles. The proposed method uses MobileNetv3 and MNasNet mobile CNN models as a feature extractor. The intersection of the features obtained from these CNN models is first taken, and then this intersection is added with the features obtained from

MobileNetv3. Then, these features are passed through the GAP layer and the fully connected layer and classified with the softmax function. This method is called HMCNet. The HMCNet method achieved approximately 1% higher accuracy than other state-of-the-art methods. To further increase the accuracy of this method, feature selection with L SVC technique and classification with SVM technique were applied after the GAP layer. This method is called FHMCNet. The proposed FHMCNet method achieved approximately 3% higher accuracy than state-of-the-art methods. With this proposed method, crosswalk detection of autonomous vehicles has increased significantly. This will significantly reduce accident rates by reducing the likelihood of collisions between pedestrians and vehicles on the crosswalk. In our future work, we plan to develop new mobile CNN models that are faster, consume less resources, and can be used in low-power devices for autonomous vehicles and other computer vision problems. Additionally, we plan to implement the FHMCNet approach we proposed in this study and test it in a real environment in the near future.

Acknowledgements We acknowledge support from TÜBİTAK, the Scientific and Technological Research Council of Turkey, under project number 122E623.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Data availability All data generated or analysed during this study are included in this published article.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. World Health Organization (2018) Global status report on road safety 2018: summary (No. WHO/NMH/NVI/18.20). World Health Organization
2. World Health Organization (2013) Pedestrian safety: a road safety manual for decision-makers and practitioners. World Health Organization
3. Pfortmueller CA, Marti M, Kunz M, Lindner G, Exadaktylos AK (2014) Injury severity and mortality of adult zebra crosswalk and non-zebra crosswalk road crossing accidents: A Cross-sectional analysis. *PLoS One* 9(3):1–6. <https://doi.org/10.1371/journal.pone.0090835>
4. Kaur M, Singh D, Kumar V, Lee HN (2023) MLNet: metaheuristics-based lightweight deep learning network for cervical cancer diagnosis. *IEEE J Biomed Health Inform* 27(10):5004–5014. <https://doi.org/10.1109/JBHI.2022.3223127>
5. Singh D, Kaur M, Alanazi JM, Alzubi AA, Lee HN (2023) Efficient evolving deep ensemble medical image captioning network. *IEEE J Biomed Health Inform* 27(2):1016–1025. <https://doi.org/10.1109/JBHI.2022.3223181>
6. Şener A, Doğan G, Ergen B (2023) A novel convolutional neural network model with hybrid attentional atrous convolution module for detecting the areas affected by the flood. *Earth Sci Inform.* <https://doi.org/10.1007/s12145-023-01155-9>

7. Doğan G, Ergen B (2022) “A new mobile convolutional neural network-based approach for pixel-wise road surface crack detection,” *Measurement (Lond)* 195. <https://doi.org/10.1016/j.measurement.2022.111119>
8. Ivanchenko V, Coughlan J, Shen H (2008) “Detecting and locating crosswalks using a camera phone,” *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*. <https://doi.org/10.1109/CVPRW.2008.4563143>
9. Murali VN, Coughlan JM (2013) “Smartphone-based crosswalk detection and localization for visually impaired pedestrians,” *Electronic Proceedings of the 2013 IEEE International Conference on Multi-media and Expo Workshops, ICMEW 2013*. <https://doi.org/10.1109/ICMEW.2013.6618432>
10. Wang S, Pan H, Zhang C, Tian Y (2014) RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs. *J Vis Commun Image Represent* 25(2):263–272. <https://doi.org/10.1016/j.jvcir.2013.11.005>
11. Ghilardi MC, Jacques Junior J, Manssour I (2018) Crosswalk localization from low resolution satellite images to assist visually impaired people. *IEEE Comput Graph Appl* 38(1):30–46. <https://doi.org/10.1109/MCG.2016.50>
12. Herumurti D, Uchimura K, Koutaki G, Uemura T (2013) “Urban road network extraction based on zebra crossing detection from a very high resolution RGB aerial image and DSM data,” *Proc - 2013 Int Conf Sig-Image Technol Internet-Based Syst, SITIS 2013* 79–84 <https://doi.org/10.1109/SITIS.2013.24>
13. Zhai Y, Cui G, Gu Q, Kong L (2015) “Crosswalk detection based on MSER and ERANSAC,” *IEEE Conf Intell Trans Syst, Proc, ITSC 2015*:2770–2775. <https://doi.org/10.1109/ITSC.2015.448>
14. Foucher P, Sebsadji Y, Tarel JP, Charbonnier P, Nicolle P (2011) “Detection and recognition of urban road markings using images,” *IEEE Conf Intell Trans Syst, Proc, ITSC 1747–1752*. <https://doi.org/10.1109/ITSC.2011.6082840>
15. Berriel RF, Lopes AT, de Souza AF, Oliveira-Santos T (2017) Deep learning based large-scale automatic satellite crosswalk classification. *ArXiv* 14(9):1513–1517
16. Berriel RF, Rossi FS, de Souza AF, Oliveira-Santos T (2017) Automatic large-scale data acquisition via crowdsourcing for crosswalk classification: A deep learning approach. *Comput Graphics (Pergamon)* 68:32–42. <https://doi.org/10.1016/j.cag.2017.08.004>
17. Aggarwal K et al (2022) Has the future started? The current growth of artificial intelligence, machine learning, and deep learning. *Iraqi J Comput Sci Math* 3(1):115–123. <https://doi.org/10.52866/ijcsm.2022.01.01.013>
18. Moura RS, Sanches SRR, Bugatti PH, Saito PTM (2022) Pedestrian traffic lights and crosswalk identification. *Multimed Tools Appl* 81(12):16497–16513. <https://doi.org/10.1007/s11042-022-12222-6>
19. Kaur J, Singh W (2023) A systematic review of object detection from images using deep learning. *Multimed Tools Appl*. <https://doi.org/10.1007/s11042-023-15981-y>
20. Kaur J, Singh W (2022) Tools, techniques, datasets and application areas for object detection in an image: a review. *Multimed Tools Appl* 81(27):38297–38351. <https://doi.org/10.1007/s11042-022-13153-y>
21. Sapkal A, Arti, Pawar D, Singh P (2023) “Lane detection techniques for self-driving vehicle: comprehensive review,” *Multimed Tools Appl*. <https://doi.org/10.1007/s11042-023-14446-6>
22. Doğan G, Ergen B (2023) A new approach based on convolutional neural network and feature selection for recognizing vehicle types. *Iran J Comput Sci* 6(2):95–105. <https://doi.org/10.1007/s42044-022-00125-6>
23. Howard A, et al (2019) “Searching for mobileNetV3,” *Proc IEEE IntConf Comput Vision 2019-Octob*:1314–1324. <https://doi.org/10.1109/ICCV.2019.00140>
24. Tan M, et al (2019) “Mnasnet: Platform-aware neural architecture search for mobile,” *Proc IEEE ComputSoc Conf Comput Vision Pattern Recog 2019-June*:2815–2823. <https://doi.org/10.1109/CVPR.2019.00293>
25. Silva ET, Sampaio F, da Silva LC, Medeiros DS, Correia GP (2020) A method for embedding a computer vision application into a wearable device. *Microprocess Microsyst* 76:103086. <https://doi.org/10.1016/j.micpro.2020.103086>
26. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 4510–4520
27. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297. <https://doi.org/10.1007/BF00994018>
28. Liu B, Yu X, Zhang P, Yu A, Fu Q, Wei X (2018) Supervised deep feature extraction for hyperspectral image classification. *IEEE Trans Geosci Remote Sens* 56(4):1909–1921. <https://doi.org/10.1109/TGRS.2017.2769673>

29. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. CoRR, abs/1412.6980
30. Radványi M, Varga B, Karacs K (2010) “Advanced crosswalk detection for the bionic eyeglass,” *2010 12th International Workshop on Cellular Nanoscale Networks and their Applications, CNNA 2010*. <https://doi.org/10.1109/cnna.2010.5430281>.
31. Tümen V, Ergen B (2020) Intersections and crosswalk detection using deep learning and image processing techniques. *Physica A: Stat Mech Appl* 543:123510. <https://doi.org/10.1016/j.physa.2019.123510>
32. Ni J, Shen K, Chen Y, Cao W, Yang SX (2022) An improved deep network-based scene classification method for self-driving cars. *IEEE Trans Instrum Meas* 71:1. <https://doi.org/10.1109/TIM.2022.3146923>
33. Wangsiripitak S, Hano K, Kuchii S (2022) “Traffic light and crosswalk detection and localization using vehicular camera,” in *KST 2022 - 2022 14th International Conference on Knowledge and Smart Technology*, Institute of Electrical and Electronics Engineers Inc., pp. 108–113. <https://doi.org/10.1109/KST53302.2022.9729066>
34. Bayat N et al (2023) Vision transformer customized for environment detection and collision prediction to assist the visually impaired. *J Imaging* 9(8):161. <https://doi.org/10.3390/jimaging9080161>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.