AN INFRARED VIDEO DETECTION AND CATEGORIZATION SYSTEM BASED ON MACHINE LEARNING

D. Švorc[*], *T. Tichý*[†], *M. Růžička*[*]

Abstract: The main aim of this paper is to present a new possibility for detection and recognition of different categories of electric and conventional (equipped with combustion engine) vehicles. These possibilities are provided by use of thermal and visual video cameras and two methods of machine learning. The used methods are Haar cascade classifier and convolutional neural network (CNN). The thermal images, obtained through an infrared thermography camera, were used for the training database. The thermal cameras can complement or substitute visible spectrum of video cameras and other conventional sensors and provide detailed recognition and classification data needed for vehicle type recognition. The first listed method was used as an object detector and serves for the localization of the vehicle on the road without any further classification. The second method was trained for vehicle recognition on the thermal image database and classifies a localized object according to one of the defined categories. The results confirmed that it is possible to use infrared thermography for vehicle drive categorization according to the thermal features of vehicle exteriors together with methods of machine learning for vehicle type recognition.

Key words: *convolution neural network, Haar cascade algorithm, thermal classification, detection and classification system*

Received: January 26, 2021 **DOI:** 10.14311/NNW.2021.31.014
Revised and accepted: August 30, 2021

1. Introduction

The goal of automated video monitoring systems is to substitute the need of human labour for solving simple vision based tasks, that can be fully performed by a computer or an automated system [1]. The applications of computer vision systems are applied in many public areas such as roads, airports and retail areas [1]. One of the application, which is widely used for the vision systems, is the monitoring of

[†]Tomáš Tichý; Department of Transport Telematics, Faculty of Transportation Sciences, Czech Technical University in Prague, Konviktska 20, 110 00 Czech Republic, E-mail: tichyto1@fd.cvut.cz

[*]David Švorc – Corresponding author; Miroslav Růžička; Department of Vehicles and Ground Transport, Faculty of Engineering, Czech University of Life Sciences Prague, Kamýcká 129, 165 00 Czech Republic, E-mail: svorcd@tf.czu.cz, ruzicka@tf.czu.cz

highways and city intersections. Effective real time traffic management is required in these localities for the appropriate reaction to changes in traffic characteristics in a timely manner. This will allow traffic dispatchers and other authorities to flexible respond to the traffic situation occurring on roads. Among the advantages that are offered by vision-based video monitoring systems are vehicle counting, vehicle classification, illegal U-turns etc. Vehicle classification is an important signal processing task that has wide application from ITS (Intelligent transportation systems) to the military [2]. The disadvantage of actual video detection systems is the use limited to visual detection cameras. Hence these detection systems are more easily influenced by weather conditions and light changes [3]. Infrared cameras are expected to provide continuous and independent detection of vehicles on any weather and light conditions in their surroundings [4]. Larger expansion of these cameras is still limited by their higher price and also lower resolution in comparison with ordinary visual cameras [5]. Due to the ongoing development and the growing number of produced thermal cameras it is expected future price reductions and their wider usage.

The novel method of the battery electric vehicles detection is proposed in this paper. Every single car generates heat during drive [6]. Even the car equipped with an electric motor generates heat but when it is compared with conventional vehicles, it is significantly less heat amount. The heat is emitted to the body of the vehicle and it is easily visible on the front mask and side of the vehicle. These features were used to get advantages of the infrared camera together with methods of machine learning. Principle of the built system is to detect object evaluated as a vehicle due to Haar Cascade classifier, then to classify vehicles into three types of vehicle (personal car, van, bus) with support of CNN and prepared vehicle thermal image database. Recognition of the battery electric vehicles in traffic flow is based on the measurement of thermal patterns of the vehicle exteriors (grilled area and front side). Result of measurements establishes the criteria for each category and differentiates electric and conventional vehicles. The extraction and calculation of individuals pixels was described in our previous works [7, 8].

2. Related work

The following review of related work is limited due to a small number of papers that are focused on the application of thermal cameras for vehicle detection with the use of machine learning methods. However many developers use visual cameras with the combination of machine learning in their research. Not many authors devote their work to the possibility of the use of the thermal property for object detection [6, 7], but few papers describe temperature maps of side view profile of vehicles [9] or detection of humans based on the body temperature [5]. Several numbers of methods are used for detection of the objects such as pedestrians, cyclists or vehicles. The authors of the papers [2, 9, 10] used principal component analysis (PCA). Other literature sources refer the principle together with support vector machine (SVM) for vehicle classification [11]. Iwasaki et al. took advantage of the horizontal and vertical vectors of the objects in the image recorded by the thermal camera to detect vehicles [12]. The other authors [3, 13] used the principle of the Viola-Jones detector firstly described in the article [14]. Another systems for vehicle

detection are based on day and night algorithm that uses as a target of detection the front and rear lights [14,15]. Paper [3] compares the video technologies, specifically conventional visual cameras and thermal cameras. Author [3] processed captured video using object tracker available video in open source project of computer vision "traffic intelligence" [17]. Individual pixels are firstly detected and tracked from frame to frame and saved as vehicle body trajectory using Kanade-Lucas-Tomasi tracking algorithm [18].

The authors used the SVM to learn the appearance of road users (cyclists, pedestrians) by using histograms of oriented gradients (HOG) [19]. As a result of this comparison, ordinary video is well functional under sunny or overcast weather conditions, but with significant weather changes in light conditions, shade etc., the thermal camera's performance is much higher [3].

The author [6] presents the possibilities of vehicle classification based on temperature signs. The focus is placed on the different thermal features of various vehicle categories and distinguish between them by statistical analysis of their thermal images, the detection area is the entire image of a given height and width. Only a part of the temperature histogram was taken into account and it was statistically evaluated as a representative part for the specific vehicle category. Author [18] uses the windshield and its surroundings as a detection area. However, this method proved to be less accurate under winter conditions due to similar windshield temperatures and ambient temperatures [20]. Authors [15] used both visual and thermal cameras for detection and focused on grille areas and headlight as the detection area.

3. Proposed method

The proposed system uses Haar cascade classifier as an object detector that detects object in the picture. The define object is marked as a car without any further classification. Vehicle image database was prepared by the use of Flir thermal camera E5 and built CNN is used for vehicle categorization. An ordinary visual camera [21] was used with high-resolution visual video stream due to lower resolution of the thermal camera and less quality video stream for vehicle detection by Haar cascade classifier. Training of CNN was performed by the prepared database of the thermal pictures of vehicles. The individual steps to train the classifier of proposed methods are described in the diagram see Tab. II. To verify the recognition functionality of the engine's thermal signature, image data was captured under two different temperature conditions specifically in the August and in December. The smallest difference between the atmospheric temperature and the temperature of the mask of the vehicle heated by the engine occurs in the summer months and it is therefore suitable for evaluation. Preliminary measurements carried out in winter proved that this system could meet with difficulties, especially at higher outdoor temperatures from May to September as usual in the Central Europe. Different thermal signs in summer and winter can be seen for comparison in the Fig. 1.

The thermal camera obtains two inputs linked with thermal energy. It is external temperature of a body of a vehicle and engine temperature reflected off the road surface. The values of temperature are contained in each pixel of the image are represented by the brightness that is indicated by the grey level. It should be noted that visible contrast of high value pixels is in the tires area in the summer season.

(a) (b)

Fig. 1 *Measurement of vehicles in traffic flow: (a) winter, and (b) summer measurement [22].*

The perceptible high value of pixels of the image in the mask area of the vehicle and emitted thermal energy under the vehicle is noticeable visible in the winter season. The marked right part of the diagram (see Fig. 2) describes the manual analysis of the thermal features of the vehicle images. It contains the calculation of individual pixel values and transforms them into the block of the thermal detector for further processing. This part is presented in our previous work [7,8]. The paper is particularly concerning on left side of diagram, especially the vehicle detection and classification. Following chapters explain individual parts of the diagram such

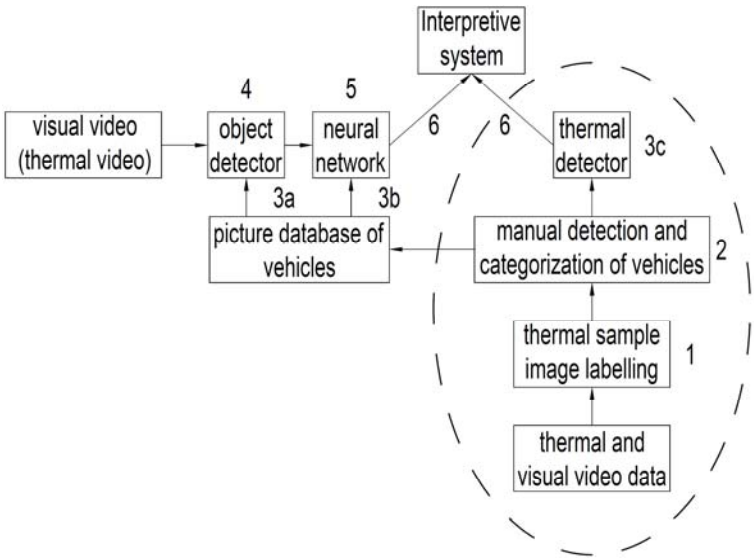

Fig. 2 *Diagram of the proposed system.*

as Haar cascade detection algorithm and CNN.

1. Manual label of thermal vehicle image sample and the preparation for step 2.
2. Manual vehicle categorization for vehicle category database.
3. Training of Haar cascade detector (3a) and CNN (3b) and manual analyse of thermal features of vehicles and computing of pixel values (3c).
4. Object detector sends information of detected vehicle based on database of vehicle to CNN.
5. CNN trained on vehicle category database select vehicle category for detected vehicle.
6. Composed information vehicle category and engine type (electric / combustion).

3.1 Haar cascade classifier

The Haar cascade algorithm is a method used to detect objects in an image based on the concept of characteristics proposed by Paul Viola and Michael Jones [14]. The algorithm consists of four steps:

- Selection of Haar characteristics.
- Integrated image representation.
- Using the AdaBoost algorithm to train the classifier.
- Compilation of the topology of the classifier, which takes the form of a degenerate decision tree.

3.2 Convolution neural network

The model of the used convolutional neural network is based on the concept proposed by the Simonyan and Zisserman [23]. Fig. 3 describes how architecture of CNN is constructed according to cited literature [23]. The chosen architecture of CNN was selected according to suitability to our task of research. The selected CNN offers quality performance computing compared to the shorter time and difficulty of the creation. We used deep learning frameworks Keras [24] to built the CNN, which consists of convolution layers together with pooling layers, perceptron layers and softmax function. The standard RGB image has a depth consisting of three channels: red, green and blue. Given these facts, we can imagine the image as a large matrix and a convolution matrix (also called a kernel) as a smaller dimensional matrix, which is used for blurring, sharpening, edge detection and other image manipulations [25].

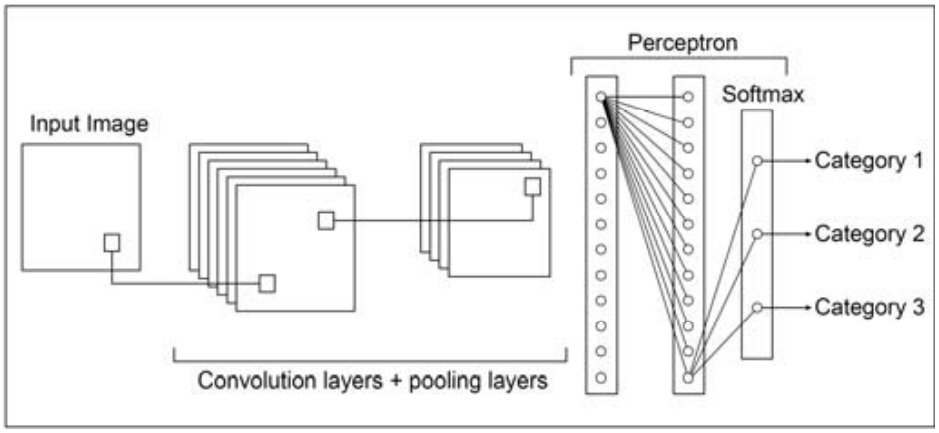

Fig. 3 *Network architecture [26].*

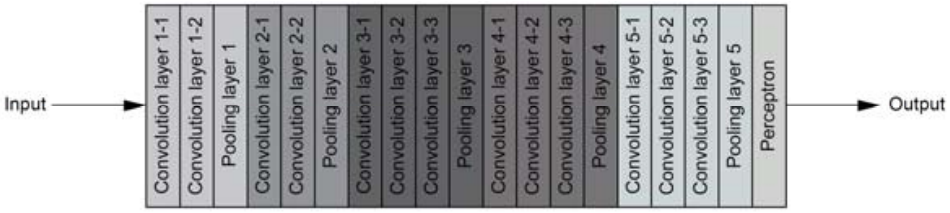

Fig. 4 *The scheme of the used layers in the CNN [26].*

Description of individual layers of the algorithm presented in the Fig. 4:

- an input – image in format 224 (height) × 224 (width) × 3 (depth – RGB model)
- 2 convolution layers containing 64 filters with a size of 3 × 3
- 1 × Maxpooling layer (poolsize 2 × 2)
- 2 convolution layers containing 128 filters with a size of 3 × 3
- 1 × Maxpooling layer (poolsize 2 × 2)
- 3 convolution layers containing 256 filters with a size of 3 × 3
- 1 × Maxpooling layer (poolsize 2 × 2)
- 3 convolution layers containing 512 filters with a size of 3 × 3
- 1 × Maxpooling layer (poolsize 2 × 2)
- 3 convolution layers containing 512 filters with a size of 3 × 3

- 1 × Maxpooling layer (poolsize 2 × 2)
- 1 × classic perceptron layer
- 1 × classic perceptron layer
- an output layer with a number of neurons equal to the number of categories
- softmax function

The convolution process is a process in which the convolution matrix moves from left to right and from top to bottom along the original image. For each coordinate (x, y) of the original image, the convolution matrix stops and convolves its surroundings in the range given by the size of the convolution matrix [27]. Convolution is simply the sum of the multiplication of matrices by elements, i.e. the elements of the convolution matrix and the elements of the matrix formed by the elements of the original image, in the range given by the size of the convolution matrix [28]. The output of the convolution is one value that is saved in the output image at coordinates identical to those occupied at that moment by the core of the convolution matrix in the original image. The general notation of a convolution matrix using a matrix called a convolution matrix can be written as follows (1) [26]

$$\begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} * \begin{bmatrix} y_{11} & \cdots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{m1} & \cdots & y_{mn} \end{bmatrix} = \\ = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} x_{(m-i)(n-j)} \cdot y_{(1+i)(1+j)}. \tag{1}$$

Each convolution layer in the convolutional neural network contains k convolution matrices applied to the matrix entering the convolution layer, see the Fig. 5.

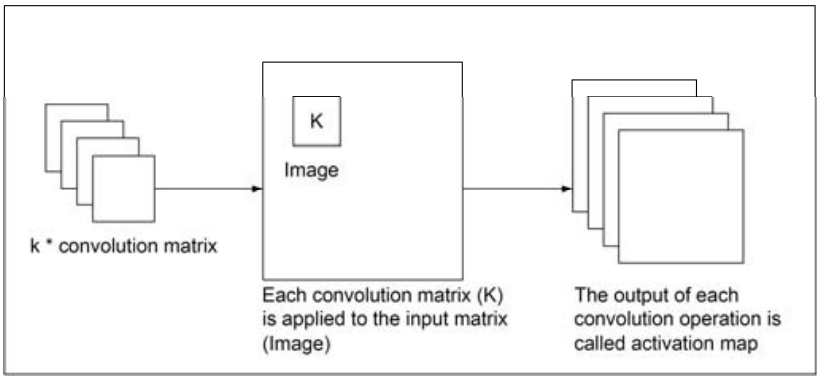

Fig. 5 *The convolution matrix [26].*

By applying all k filters to the input matrix, we obtain k 2D activation maps, which are placed one behind the other along the depth, coordinate and form the final output of the convolution process, see the Fig. 6.

By gradually applying convolution, the dimensions of the input image would begin to shrink rapidly, so "zero padding" [27] see Tab. I is used, which ensures that the dimensions are maintained.

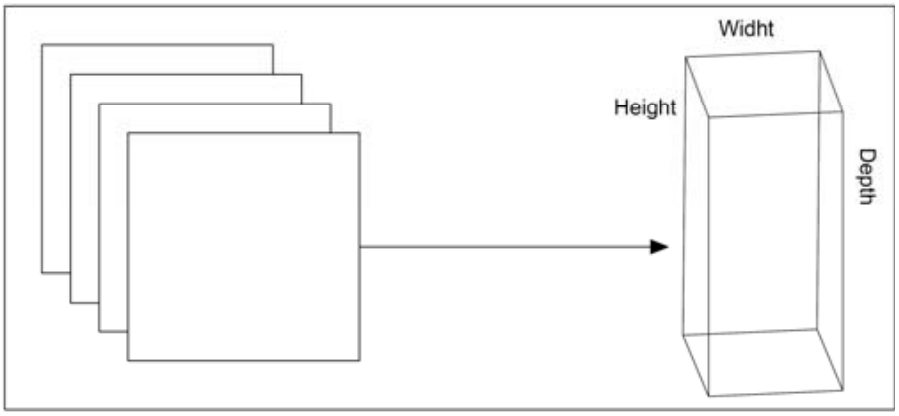

Fig. 6 *Final output of the CNN [23].*

0	0	0	0	0	0	0	0	0
0	x_{11}	x_{12}	–	–	–	–	x_{1n}	0
0	–	–	–	–	–	–	–	0
0	–	–	–	–	–	–	–	0
0	–	–	–	–	–	–	–	0
0	–	–	–	–	–	–	–	0
0	–	–	–	–	–	–	–	0
0	x_{m1}	x_{m2}	–	–	–	–	x_{mn}	0
0	0	0	0	0	0	0	0	0

Tab. I *Zero padding [23, 27].*

The size of the image after the application of the convolution process can be written using Eq. 2 according to [26, 27]

$$\left(\frac{W - F + 2P}{S}\right) + 1, \tag{2}$$

into which enter the dimensions of the input image W, the size of the convolution matrix F, the size of the "padding" P and the size of the displacement S.

Each convolution layer accepts enter about dimensions $(W_{\text{input}} \times H_{\text{input}} \times D_{\text{input}})$ and specification of these parameters:

- Number of the convolution matrices K (which indicate the depth of output from the convolution layer).
- The size of the convolution matrices F used for convolution is usually square, i.e. $F * F$.
- Shift of the convolution matrix S.
- Padding P.

The output of the convolution layer has dimensions ($W_{\text{output}} \times H_{\text{output}} \times D_{\text{output}}$) [28], where:

$$W_{\text{output}} = \left(\frac{W_{\text{input}} - F + 2P}{S}\right) + 1, \tag{3}$$

$$H_{\text{output}} = \left(\frac{H_{\text{input}} - F + 2P}{S}\right) + 1, \tag{4}$$

$$D_{\text{output}} = K. \tag{5}$$

3.3 Activation layer

After each convolution layer in the convolutional neural network is followed by a nonlinear activation function, in our case the Rectified Linear Unit (ReLu function) (Fig. 7) was chosen. Following the Krizhevsky et. al. [29] we chose this function due to the reason that convolutional neural networks with ReLUs train several times faster than their equivalents with tanh units. This function can be expressed as:

$$f(x) = x^{+} = \max(0, x). \tag{6}$$

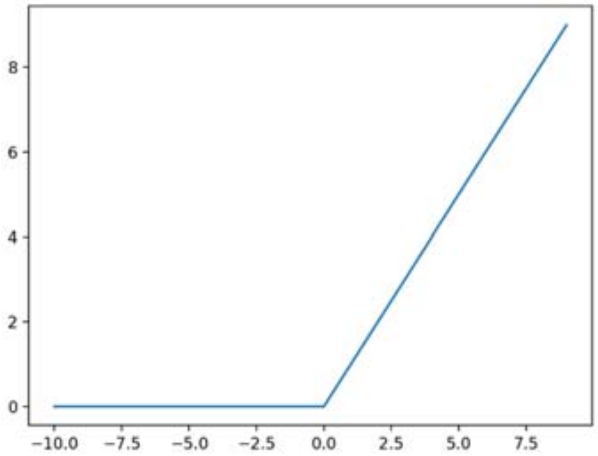

Fig. 7 *ReLu function [26, 28], where y axis is output,* $f(x) = \max(0, x)$ *and x axis is input.*

Activation layer accept input image of dimensions ($W_{\text{input}} \times H_{\text{input}} \times D_{\text{input}}$) [23].

$$W_{\text{input}} * H_{\text{input}} * D_{\text{input}} = W_{\text{output}} * H_{\text{output}} * D_{\text{output}}. \tag{7}$$

3.4 Pooling layer

Pooling layers are located between convolution layers. Their main function is to reduce the size of the input to the next convolution layer and at the same time help prevent network re-learning. The layer accepts input of size ($W_{\text{input}} * H_{\text{input}} * D_{\text{input}}$) and specification of these parameters [28]

- Size of the receptive field F.
- Shift of the receptive field S.

We get the output of dimension ($W_{\text{input}} * H_{\text{input}} * D_{\text{input}}$) [30] by application of pooling layers, where:

$$W_{\text{output}} = \left(\frac{W_{\text{input}} - F}{S}\right) + 1, \tag{8}$$

$$H_{\text{output}} = \left(\frac{H_{\text{input}} - F}{S}\right) + 1, \tag{9}$$

$$D_{\text{output}} = D_{\text{input}}. \tag{10}$$

Algorithm uses type of pooling, which takes maximal value from convolution area.

3.5 Perceptron

Rosenblat [31] defined a Perceptron as a system that learns using labeled examples of feature vectors (or raw pixel intensities), mapping these inputs to their corresponding output class labels. The perceptron is placed at the end of the convolutional neural network, before applying the softmax classifier. Following equations and expression were described in [26].

Perceptron – Forward signal propagation with activation function

Input layer: $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$

Hidden layer: $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{2n+1}$

Output layer: x_1

The activation function f is applied in the hidden layer and the output layer [28]. We demonstrate its application at the output of a hidden layer. The output from the hidden layer without using the activation function looks like this:

$$\mathbf{w}_j = a_{j,1} \cdot v_1 + a_{j,2} \cdot v_2 + \cdots + a_{j,n} \cdot v_n = \sum_{i=1}^{n} a_{j,i} \cdot v_i \text{ for } j \in \{1, \ldots, 2n+1\}. \tag{11}$$

The output from the hidden layer using the activation function looks like this:

$$\hat{\mathbf{w}}_j = f\left(a_{j,1}{\cdot}v_1{+}a_{j,2}{\cdot}v_2{+}\cdots{+}a_{j,n}{\cdot}v_n\right) = f\left(\sum\nolimits_{i=1}^{n} a_{j,i}{\cdot}v_i\right) \text{ for } j{\in}\{1,\ldots,2n{+}1\}. \tag{12}$$

Possible described as:

$$\hat{\mathbf{w}} = f(\mathbf{w}) = f(Av). \tag{13}$$

The vectors of the individual layers after applying the activation function, look as follows:

Input layer vector: $\mathbf{v} = (v_1, v_2, \ldots, v_n)^{\mathrm{T}}$

Hidden layer vector: $\hat{\mathbf{w}} = (\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_{2n+1})^{\mathrm{T}}$ gets as $\hat{\mathbf{w}} = f(\mathbf{w}) = f(\mathbf{A}\mathbf{v})$

Output layer vector: $\hat{\mathbf{x}} = (\hat{x}_1)$ gets as $\hat{\mathbf{x}} = f(\mathbf{x}) = f(\mathbf{B}\hat{w})$

Forward signal propagation three layers CNN can be described as:

$$\hat{\mathbf{x}} = f(\mathbf{B}f(\mathbf{A}\mathbf{v})). \tag{14}$$

3.6 Softmax classifier

The algorithm uses Softmax classifier desribted in literature [26] to normalize output of CNN, which is written as:

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_i}}, \tag{15}$$

where y_i is output of perceptron.

4. Analysis and results

4.1 Detection and classification of the vehicles

The first step of the object detector (Haar cascade classifier) is to indicate objects – vehicle in the video scene without further classification. Output is presented in Fig. 8. The average success of detecting of the Haar cascade classifier is 0.9793 (car: 0.9792, van: 0.9777, bus: 0.9810).

The object (car_object) detected by the cascade algorithm (OD) is after that passed to the convolutional neural network (NS) prediction module in the form of a data matrix of dimensions $224 * 224 * 3$ (height $*$ width $*$ depth).

NS learned for vehicle category recognition assigns one of the categories to the image. NS returns the output of the softmax function, in the form of a probability assigned to the individual vehicle categories. The most likely category is the classification for a specific case and is assigned to the object detected in the image. The detected and classified object in the image then has the following form see Fig. 9.

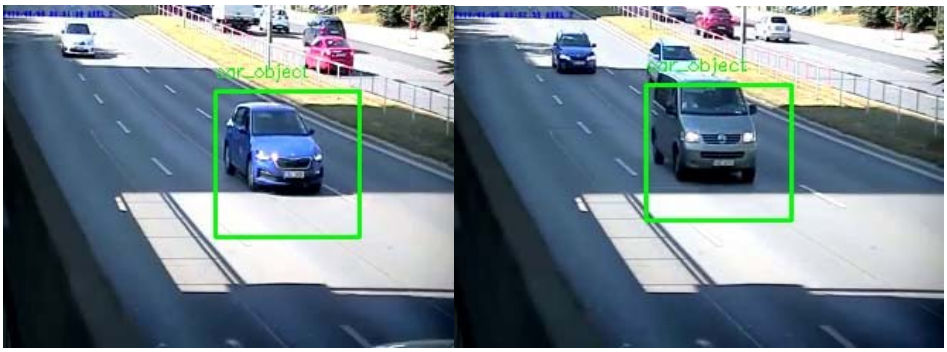

Fig. 8 *Result of the object detector [21].*

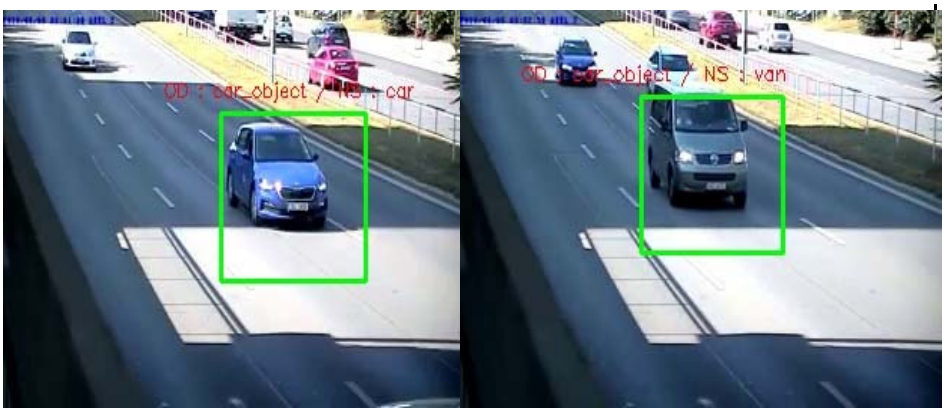

Fig. 9 *Detected and classified objects — car and van [21].*

4.2 Training of CNN – categorization of the vehicles

The training of CNN for vehicle categorization was performed through database of thermal pictures of the vehicles. The examples of sample pictures in the database see Fig. 10, Fig. 10, Fig. 12.

Proposed category of vehicles:

- personal car
- van
- bus

Dataset:

- personal car (1334 frames, training set: 1264, test set: 70)
- van (319 images, training images: 296, test set: 23)
- bus (32 images, training set: 26, test set: 6)

Example of sample of data:

- personal car

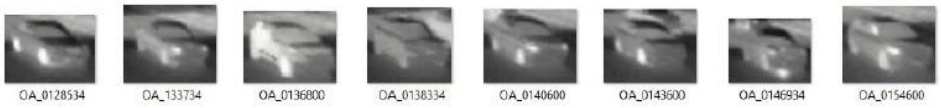

Fig. 10 *Example of sample personal vehicles images [22].*

- van

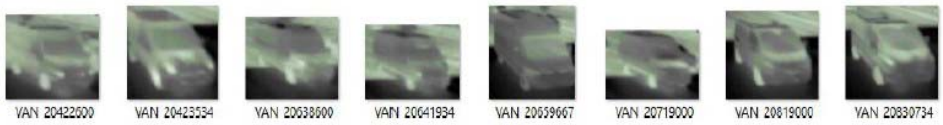

Fig. 11 *Example of sample van images [22].*

- bus

Fig. 12 *Example of sample bus images [22].*

Training of the network:

- time of training: 4.5 h
- number of epoch: 200
- learning coefficient: 0.0001
- momentum: 0.9

The most important metric from the above is the validation loss (val_loss), which is defined as:

$$\text{val loss} = \frac{1}{N} \cdot (\text{cross entropy}), \tag{16}$$

where cross entropy is calculated as

$$\text{cross entropy} = -\sum_i \mathbf{L}_i \cdot \log(S_i), \tag{17}$$

where S represents the softmax output for each record that is defined in Eq. 15 and $\mathbf{L}$ represents the vector of values for each record for each category.

Network evaluation, see Tab. II.

The classification algorithm correctly classified 99 % of thermal images of cars, 95 % of thermal images of vans and 100 % of thermal images of buses. The weighted accuracy of the algorithm (weighted average of the classification precisions for individual categories with respect to the number of records) is 98 %. The images used for the classification were not used in its training phase by the algorithm.

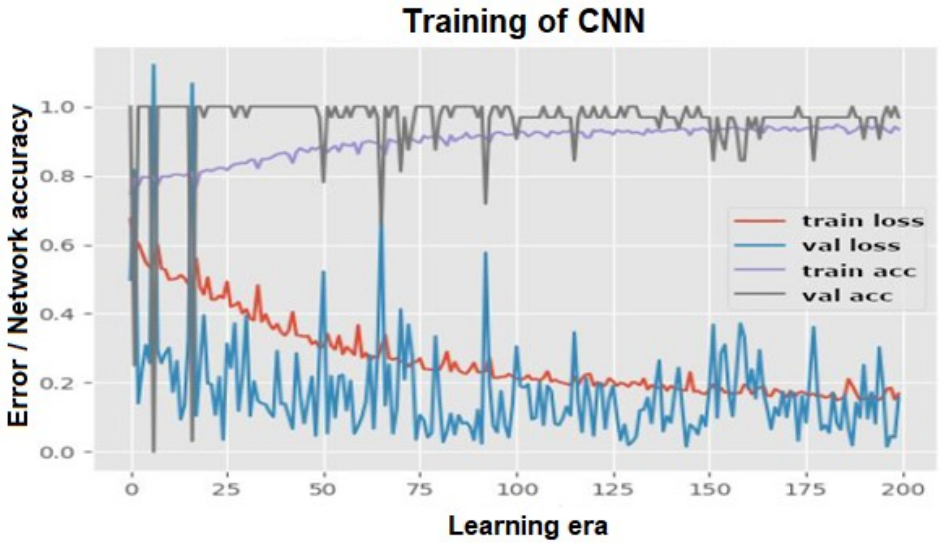

Fig. 13 *Result of the CNN training.*

	precision	recall	f1-score	support
car	0.99	0.96	0.97	70
van	0.95	0.96	0.90	23
bus	1.00	0.83	0.91	6

Tab. II *Network evaluation.*

5. Discussion

The proposed detection and classification system proved the possibility of the use of CNN together with the Haar cascade classifier. CNN was trained for vehicle classification through the thermal image database of the vehicles. The proposed system worked well under different conditions such as changes in atmospheric temperature and intensity of light. What is possible to conclude the system pitfalls

was in resolution of used thermal camera that was not sufficiently high for the detection of the object from the thermal video stream. Therefore, a high-resolution visual video stream was used to help with reliable detection by the Haar cascade classifier. The presented system and its functionality of cooperation between object detector and classification block of CNN suggests a very successfuly working method for vehicle detection and classification even with lower resolution quality of thermal cameras. The suggested system proved that thermal pictures can be used for vehicle classification and for reliable detection but it is necessary to use sharper images. Compared to standard feedforward neural networks with similarly sized layers, CNN have much fewer connections and parameters and so they are easier to train, while their theoretically best performance is likely to be only slightly worse. The completed presented system prospectively offers a robust approach for vehicle detection and categorization of conventional and electric vehicles. The following research must be focused on the tests of the Haar cascade classifier with the higher resolution thermal stream of higher quality thermal camera with include our experience of testing [8].

Acknowledgement

This work was supported by the Technology Agency of the Czech Republic under the research project "The impacts after low emission mobility implementation on the environmental effects reduction in urban agglomerations" No. TH04030403.

References

[1] ARINALDI A., PRADANA J.A., GURUSINGA A.A. Detection and classification of vehicles for traffic video analytics. *Procedia Computer Science* [online]. 2018, 144, pp. 259–268, doi: 10.1016/j.procs.2018.10.527.

[2] ZHANG C., CHEN X., CHEN W.B. A PCA-based vehicle classification framework. *ICDEW 2006 – Proceedings of the 22nd International Conference on Data Engineering Workshops*. 2006, doi: 10.1109/ICDEW.2006.16.

[3] FU T., STIPANCIC J., ZANGENEHPOUR S., MIRANDA-MORENO L., SAUNIER N. Automatic Traffic Data Collection under Varying Lighting and Temperature Conditions in Multimodal Environments: Thermal versus Visible Spectrum Video-Based Systems. *Journal of Advanced Transportation* [online]. 2017. 2017, pp. 1–15. doi: 10.1155/2017/5142732. Available at: https://www.hindawi.com/journals/jat/2017/5142732/.

[4] IWASAKI Y., KAWATA S., NAKAMIYA T. Vehicle detection even in poor visibility conditions using infrared thermal images and its application to road traffic flow monitoring. *Lecture Notes in Electrical Engineering*. 2011, 151, pp. 997–1009. doi: 10.1007/978-1-4614-3558-7_85.

[5] TEUTSCH M., MUELLER T., HUBER M., BEYERER J. Low resolution person detection with a moving thermal infrared camera by hot spot classification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 2014, pp. 209–216, doi: 10.1109/CVPRW.2014.40.

[6] SANGNOREE A., CHAMNONGTHAI K. Thermal-image processing and statistical analysis for vehicle category in nighttime traffic. *Journal of Visual Communication and Image Representation* [online]. October 2017. 48, pp. 88–109, doi: 10.1016/j.jvcir.2017.06.006.

[7] SVORC D., TICHY T., RUZICKA M. Detection of the electric vehicle using thermal characteristics. In : *2020 Smart City Symposium Prague (SCSP)* [online]. IEEE, June 2020. pp. 1–5. ISBN 978-1-7281-6821-0. Available at: https://ieeexplore.ieee.org/document/9133981/.

[8] TICHÝ T., ŠVORC D., RŮŽIČKA M., BĚLINOVÁ Z. Thermal Feature Detection of Vehicle Categories in the Urban Area. *Sustainability* [online]. 17 June 2021. 13(12), pp. 6873, doi: 10.3390/su13126873. Available at: https://www.mdpi.com/2071-1050/13/12/6873.

[9] SARIKAN S.S., OZBAYOGLU A.M., ZILCI O. Automated Vehicle Classification with Image Processing and Computational Intelligence. *Procedia Computer Science* [online]. 2017, 114, pp. 515–522, doi: 10.1016/j.procs.2017.09.022.

[10] WU J., ZHANG X. A PCA classifier and its application in vehicle detection. *Proceedings of the International Joint Conference on Neural Networks*. 2001, 1, pp. 600–604, doi: 10.1109/ijcnn.2001.939090.

[11] ZHOU J., GAO D., ZHANG D. Moving Vehicle Detection for Automatic Traffic Monitoring. *IEEE Transactions on Vehicular Technology* [online]. January 2007. 56(1), pp. 51–59, doi: 10.1109/TVT.2006.883735. Available at: http://ieeexplore.ieee.org/document/4067123/.

[12] YOICHIRO I. A method of robust moving vehicle detection for bad weather using an infrared thermography camera. In: *2008 International Conference on Wavelet Analysis and Pattern Recognition* [online]. IEEE, August 2008. pp. 86–90. ISBN 978-1-4244-2238-8. Available at: http://ieeexplore.ieee.org/document/4635755/.

[13] IWASAKI Y., MISUMI M., NAKAMIYA T. Robust Vehicle Detection under Various Environments to Realize Road Traffic Flow Surveillance Using an Infrared Thermal Camera. *The Scientific World Journal* [online]. 2015, pp. 1–11, doi: 10.1155/2015/947272. Available at: http://www.hindawi.com/journals/tswj/2015/947272/.

[14] VIOLA P., JONES M. Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2001, 1, doi: 10.1109/cvpr.2001.990517.

[15] NAM Y., NAM Y.-C. Vehicle classification based on images from visible light and thermal cameras. *EURASIP Journal on Image and Video Processing* [online]. 19 December 2018, 5(1), pp. 1–9, doi: 10.1186/s13640-018-0245-2. Available at: https://jivp-eurasipjournals.springeropen.com/articles/10.1186/s13640-018-0245-2.

[16] LEE D., PARK Y. Measurement of traffic parameters in image sequence using spatio-temporal information. *Measurement Science and Technology* [online]. 1 November 2008, 19(11), pp. 115503. [Accessed 24 November 2019]. doi: 10.1088/0957-0233/19/11/115503. Available at: http://stacks.iop.org/0957-0233/19/i$=$11/a$=$115503?key$=$crossref.1d6fec0e6da90eb9125c9ba82fa0089f.

[17] SAUNIER N., SAYED T. A feature-based tracking algorithm for vehicles in intersections. In: *Third Canadian Conference on Computer and Robot Vision, CRV 2006* [online]. IEEE, 2006, p. 59. ISBN 0769525423. Available at: http://ieeexplore.ieee.org/document/1640414/.

[18] JIANBO SHI, T. Good features to track. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94* [online]. IEEE Comput. Soc. Press, 1994. pp. 593–600. ISBN 0-8186-5825-8. Available at: http://ieeexplore.ieee.org/document/323794/.

[19] XU Y., YU G., WANG Y., WU X., MA Y. A hybrid vehicle detection method based on viola-jones and HOG + SVM from UAV images. *Sensors (Switzerland)*. 2016, 16(8), doi: 10.3390/s16081325.

[20] IWASAKI Y., MISUMI M., NAKAMIYA T. Robust Vehicle Detection under Various Environmental Conditions Using an Infrared Thermal Camera and Its Application to Road Traffic Flow Monitoring. *Sensors* [online]. 17 June 2013. 13(6), pp. 7756–7773. doi: 10.3390/s130607756. Available at: http://www.mdpi.com/1424-8220/13/6/7756.

[21] *HARDWARE INSTALLATION GUIDE VC58SMi-1.3 Megapixel Box Type IP Camera VC58SM3i-3.0 Megapixel Box Type IP Camera VC58EHi-Standard D1 Box Type IP Camera* [online]. [Accessed 25 May 2021]. Available at: http://www2.viakom.cz/stahuj/vision/manualy/VC58_Hardware_Installation_Guide_eng_20110922.pdf.

[22] FLIR E5 Infrared Camera with MSX® |FLIR Systems. [online]. [Accessed 1 February 2020]. Available at: https://www.flir.com/products/e5/.

[23] SIMONYAN K., ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *3rd International Conference on Learning Representations, ICLR 2015* [online]. 4 September 2014. pp. 1–14. Available at: http://arxiv.org/abs/1409.1556.

[24] CHOLLET F., Others. Keras: the Python deep learning API. *Keras: the Python deep learning API* [online]. 2020, [Accessed 24 March 2021]. Available at: https://keras.io/.

[25] GOODFELLOW I., BENGIO Y., COURVILLE A. *Deep Learning* [online]. MIT Press, 2016. [Accessed 22 May 2021]. Available at: https://www.deeplearningbook.org/contents/convnets.html.

[26] ROSEBROCK A. *Deep Learning for Computer Vision with Python |Adrian Rosebrock |download* [online]. 1. PyImageSearch, 2017. [Accessed 23 March 2021]. Available at: https://b-ok.cc/book/3559375/00e406.

[27] O'SHEAi K., NASH R. An Introduction to Convolutional Neural Networks. [online]. 26 November 2015. Available at: http://arxiv.org/abs/1511.08458.

[28] WU J. Introduction to Convolutional Neural Networks. *Introduction to Convolutional Neural Networks* [online]. 2017, pp. 1–31. Available at: https://web.archive.org/web/20180928011532/https://cs.nju.edu.cn/wujx/teaching/15_CNN.pdf.

[29] KRIZHEVSKY A., SUTSKEVER I., HINTON G.E. ImageNet classification with deep convolutional neural networks. GONZALEZ, Teofilo F. (ed.), *NIPS* [online]. 24 May 2012. 60(6), pp. 84–90. doi: 10.1145/3065386. Available at: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[30] KARPATHY A. Convolutional Neural Networks for Visual Recognition. [online]. [Accessed 23 May 2021]. Available at: https://cs231n.github.io/convolutional-networks/.

[31] ROSENBLATT F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*. November 1958. 65(6), pp. 386–408, doi: 10.1037/h0042519.