Journal of Computer Virology and Hacking Techniques
https://doi.org/10.1007/s11416-023-00499-6

ORIGINAL PAPER

Machine learning methods for the industrial robotic systems security

Dmitry Tsapin[1] · Kirill Pitelinskiy[1] · Stanislav Suvorov[1] · Aleksey Osipov[2] · Ekaterina Pleshakova[3] · Sergey Gataullin[2]

Received: 30 May 2023 / Accepted: 3 August 2023

Abstract
The trends in the introduction of industrial and logistics robots into the social sphere of activity in order to ensure the safety of civilian facilities, the current problems of the growth of crime in the Russian Federation, as well as the downward trend in the identification of persons who have committed crimes (taking into account the criminogenic situation in the Russian Federation) are discussed. The features of the application of BigData methods are considered, in particular, the use of an ensemble of computer vision algorithms and the mathematical apparatus of convolutional neural networks for the timely detection of emergency situations in parking lots by means of mobile robots. Implemented training of convolutional neural networks on the MobileNetV2, ResNet50 and DenseNet121 architectures with the addition of Squeeze-and-Excitation block (SE) to solve the problem of identifying semantic signs of vehicle damage in parking lots. The methods improved using the SE block made it possible to increase the accuracy by 2–3%, which amounted to 88%, 91% and 92%, respectively. At the second stage of work, when training neural networks, about 20% of the images obtained in difficult video shooting conditions (at dusk, during rain, snow, etc.) were used. The results of testing retrained neural networks on images obtained in difficult video conditions were compared with the method based on the HOG descriptor selected by us for similar conditions (a combined approach based on directional gradient histograms, the bag-of-visual-words method, and a neural network of inverse distribution). Compared with the comparable method, DenseNet121 + SE showed an accuracy of 86%, which is the same as the accuracy of the HOG-BoVW-BPNN method, but the speed of DenseNet121 + SE is 40% faster, which makes it a more attractive method for a car park computer vision system.

Keywords Information security · Security management · Security systems · Information processes · Mobile robots · Neural networks · Convolutions · Big data · Car parking

1 Introduction

Currently, there is a use of modern methods of solving crimes using information technology (hereinafter referred to as IT). In the operational-technical, criminal procedural methods of solving crimes, artificial intelligence is increasingly used. Systems focused on analyzing the behavior of subjects, including the timely detection of emergency situations, are becoming in demand, based on algorithms for analyzing video streaming data in real time.

Automation of such processes as identifying a car entering a parking lot and monitoring its safe presence in a parking space is a necessary aspect of ensuring the security of a protected area. However, the recognition of semantic objects is complicated not only because these objects occupy a far small part of the image, but also because of environmental conditions.

✉ Ekaterina Pleshakova
espleshakova@fa.ru

Dmitry Tsapin
aw3472@gmail.com

Kirill Pitelinskiy
Yekadath@gmail.com

Stanislav Suvorov
ssw1168@mail.ru

Aleksey Osipov
a.v.osipov@mtuci.ru

Sergey Gataullin
s.t.gataullin@mtuci.ru

[1] Moscow Polytechnic University, Moscow, Russian Federation

[2] Moscow Technical University of Communications and Informatics, Moscow, Russian Federation

[3] MIREA - Russian Technological University, 78 Vernadsky Avenue, Moscow, Russian Federation119454

Published online: 02 September 2023

Currently, parking lots play a big role in the life of a car owner, since without them it is difficult to imagine a well-functioning mechanism for storing vehicles in a modern urban environment, in which the number of these vehicles tends to grow every year [1–3].

In parallel with this, there is a problem of ensuring the safe location of vehicles on the territory of car parks, in particular:

- Detection and recognition of a car number at the entrance to the car park [4];
- Control of the presence of people near vehicles [5];
- Timely detection of damage to vehicles and subsequent notification of security personnel about this [6].

Based on the problems of the study, we can conclude that the purpose of this study is to develop effective tools for analyzing the video stream from security systems in order to identify negative impacts on vehicles. The scientific novelty consists in improving the existing methods of computer vision by using their ensembles, as well as modifications that make it possible to increase the stability of algorithms to affine transformations of input data. It is planned to modify existing architectures of computer vision algorithms to improve recognition accuracy, as well as to work correctly in different video shooting conditions.

The development of developments in the field of robotics originates in the twentieth century. It was then that the main paradigms were formed, indirectly or directly related to this subject area and representing the sciences and technologies of the post-industrial society, such as: computer science, electrical engineering, special chapters of mathematics, cybernetics, mechanical engineering, Big Data, artificial intelligence, etc. [7–10]. In the context of using information technologies to solve applied problems, significant computing power is required [11–13]. Recently, computing power has been increasing in a trending manner. Such a significant increase in the computing resources of automated systems made it possible to develop robotics in the social sphere. With the beginning of the use of robotic systems of various specializations, an extremely wide field of vision has opened up for end users for their future implementation in order to improve or optimize business processes in the information dynamic contour of an enterprise/organization [14].

Currently, ensuring the safety of civilian facilities is an urgent task. One of its possible solutions is the use of a robotic security system, which is a group (swarm) of mobile robots. This ensemble of technical means is able to provide protection for both mobile and stationary objects in protected areas (within the limits of permissible self-defense) [15].

The use of robotic systems to ensure the safety of civilian facilities involves ensuring the continuity of public order, since the lives of people and the state of other people's property depend on the performance and reliability/accuracy of the functioning of specialized algorithms in such systems [16]. This underlines the hypothesis that current trends are to introduce robotic devices into areas of activity that are in demand for automating business processes. Absolutely any object of civil use cannot do without providing a proper security system that could stop the actual threats of disturbing public order, as well as ensure the continuity of its functioning in terms of ongoing business processes in order to minimize the costs of possible violations in their functioning. At the same time, the costs can be both financial and reputational [17]. In the context of ensuring the safety of vehicles, there are also logistical risks that can manifest themselves in a violation of the continuity of logistical business processes [18].

The main hypothesis to substantiate the relevance of the ideas for the development of security robots, which are based on the use of the mathematical apparatus of neural networks and computer vision algorithms, is that when protecting a civilian object, situations related to the human factor may arise (for example, rare bypasses of the territories of protected objects in 24/7 mode). Based on this formulation of the problem, problems arise in processing streams of intensely incoming heterogeneous and semi-structured data, determining patterns in such arrays and optimizing the use of high-speed algorithms, methods and tools related to the Big Data paradigm. Even in the event of an emergency situation (for example, someone commits an act of vandalism, tries to illegally enter a room to which physical access is prohibited), the main duty of a security guard, according to his own job description, is to immediately report all abnormal / illegal actions to the relevant services or law enforcement agencies, and then act depending on the situation [19–23].

Having determined the criteria corresponding to an emergency situation at a protected facility for civil use, the security robot, as a rule, communicates with notifications to the security service of the protected facility, using, for example, warning templates. The use of Big Data technologies (computer vision algorithms, deep learning algorithms) allows you to automate processes analytical processing, management and visualization of results as data arrives, from a wide array of video recording devices [24].

From the point of view of the technologies used, this class of tasks is called video analytics. Video analytics is a combination of implementations of artificial intelligence and computer vision methods in order to obtain arrays of heterogeneous real-time data for subsequent analysis of video images from video recording devices and video recordings or photos stored in the archive. The basis of video analytics software is a set of computer vision algorithms that perform the task of monitoring and analyzing data received as a video stream without any human intervention. Currently, due to the increased crime situation and increased requirements for the security infrastructure and protection of civilian facilities

and access control to them, video analytics algorithms are promising [25–27].

Artificial intelligence in the context of video analytics covers a wide range of applied areas, such as: face recognition (smart entry into the system of modern smartphones, passport control through biometric authentication systems, etc.), video stream analysis to determine the behavior of objects, security video surveillance of territories etc. This is due to the development of technologies in the field of programming, in particular, with the spread of methods and technologies of neural networks [28–30].

Neural networks in the context of video analytics is a machine learning algorithm that allows you to form and train a mathematical model to solve the classification problem directly based on the input images, text or sound. The use of neural networks involves the use of raw images as input to a program that learns them based on high-fidelity neural network algorithms. Neural network classifiers and deep learning algorithms are able to perform image classification and recognition very accurately and fairly quickly [31, 32].

If we take into account that modern video recording systems record frames in real time in high resolution (Full HD, 2 K, 4 K), then the volume of such information can reach tens of terabytes in just one week, which requires high-performance computing systems. In turn, the number of different variants of images can be an endless number, if we take into account such factors as affine transformations, camera rotations, the presence of noise, different variants of positioned objects in images, etc. Such aspects make it possible to attribute the classes of tasks solved by means of video analytics to the sphere of Big Data. This is facilitated by the use of systems based on the mathematical apparatus of artificial intelligence and neural networks. Image recognition using high-precision neural networks will allow developers of intelligent systems to move to a new level that will replace traditional video analytics [33–35]. In this case, a number of problems arise, such as insufficient computer power, incorrect equipment settings, adverse weather conditions for video shooting, video shooting at dusk, etc.

The work of Rachel Blinatal is devoted to the detection of objects in adverse weather conditions [36] Its authors, in combination with classical teaching methods, namely DPM, HOG methods, use polarization-encoded images. The authors noted that polarimetry combined with deep learning can improve accuracy by about 20–50% when performing various detection tasks.

To improve the accuracy of object classification and detection, various preprocessing methods are used, for example, edge detection methods. Robert, Sobel and Prewitt classical edge detection methods work with pixels of neighboring areas and obtain a gradient with pattern approximation [37], which are relatively simple and easy to implement, have good real-time performance, but these operators are sensitive to noise [38]. The authors of [39] noted that the use of the Sobel filter to detect COVID-19 using X-ray images improves the performance of a convolutional neural network. They rated this combination of methods as the best of the wide range of options explored.

In addition to neural networks, other methods are used in image classification and detection. The BoVW model is actively used in image classification. Also, studies have shown that "bag of visual words" (BoVW) schemes for classifying histopathological images showed higher accuracy than classical convolutional neural networks, which was 96.50% [40].

The Viola-Jones algorithm is an object detector that is used to detect the human face, facial features, cars, etc., which, in combination with the support vector machine (SVM), provides a robust object detection and classification pipeline. The simplicity of the algorithm allows real-time classification, and the accuracy for classifying some types of objects can reach 99% [41].

2 Materials and methods

2.1 Description of the object of study

A car park is a structure designed to store vehicles (Fig. 1). The investigated object is equipped with a barrier with a camera that controls the authorized entry into the territory (with license plate recognition) and cameras located around the perimeter for the purpose of video surveillance (7 video surveillance cameras), a total of 8 video cameras. Near each parking space there is a container disguised as a dustbin, which houses a mobile robot equipped with a video surveillance camera and a secure data carrier, which, if a negative impact on the vehicle is detected via a secure communication channel, informs the security service of the facility about this. At the same time, at the entrance there is a sign warning about the functioning of a mobile security robot.

Data from all cameras is processed by a high-performance system located in the corporate network of the car park and enters the data center via secure communication channels (VPN) through a single entry point. The storage period for archived video information is 30 calendar days. In case of any emergency situations, the video recording is archived on a specially designated secure storage medium. The organization of communication with the data processing center is shown in Fig. 2.

2.2 Data sets and experimental protocol

For the experiment, a system of eight Hiwatch dome-shaped IP cameras connected to an 8-channel video recorder, as well as a swarm of mobile robots, was used. This ensemble of

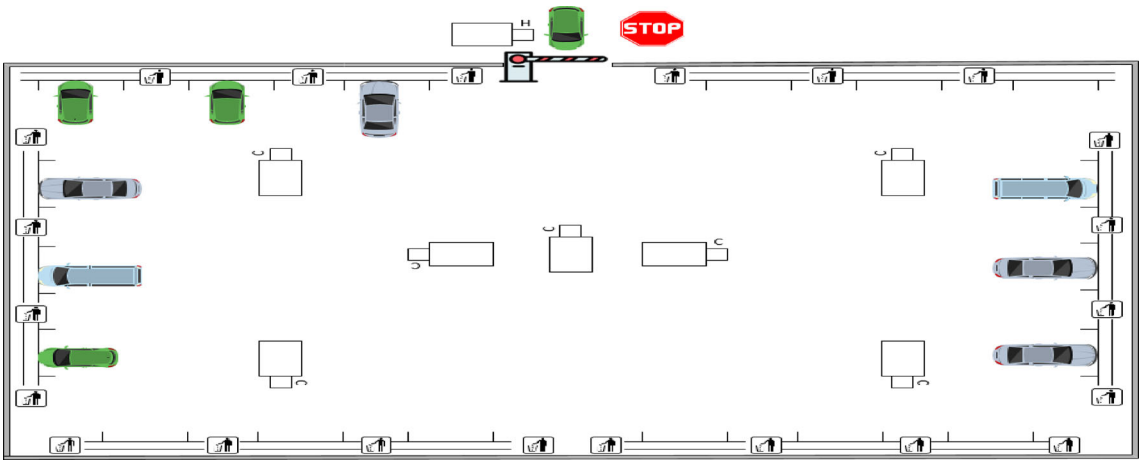

Fig. 1 Scheme of a guarded car park

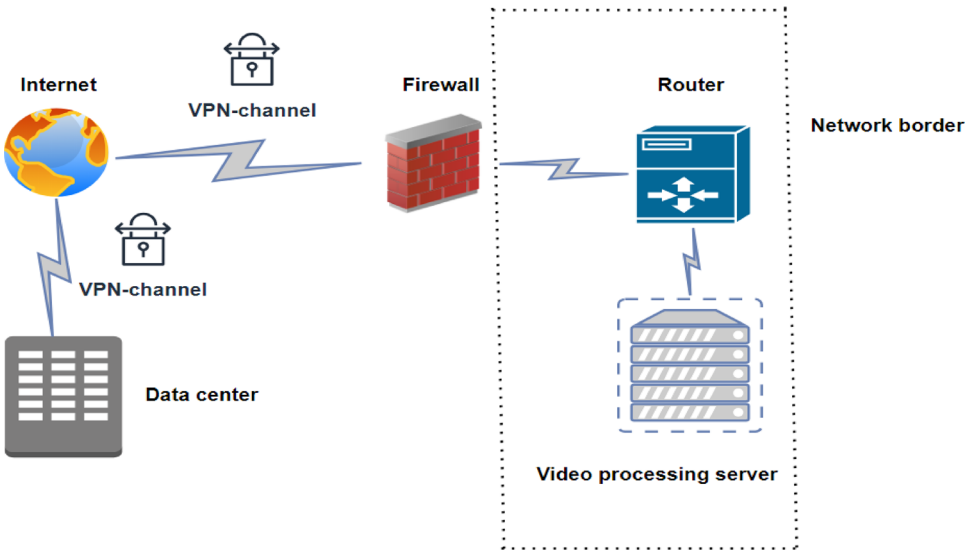

Fig. 2 Organization of communication with the data processing center

video stream measurement sources captures the situation in the parking lot in real time. We used video cameras equipped with a varifocal lens, the focal length is 2.8—12 mm, which made it possible to adjust the optical zoom of the camera at the installation site. Camera protection class—ip66.

Each IP camera runs at 30 frames per second (Fps). Since the ensemble of video cameras operates in 24/7 mode, it makes sense to adjust the frequency of processed frames from the original video stream, since the analysis of the initial set of the video stream would make the designed system based on the use of computer vision and artificial intelligence algorithms extremely slow and unsuitable for implementation on mobile phones. robotic systems. Thus, the number of processed frames per second depends on the location of each specific camera. It is worth noting that the presented cameras have the functions of automatically capturing data according to the planned time, according to the timer, according to the set events, as well as according to alarms.

At the entrance to the parking lot, the installed camera, specialized in license plate recognition, was set to a frame rate of up to 10 frames per second, and the cameras located along the perimeter of the protected object—3–5 frames per second. The dataset was collected on the basis of video recordings made in the territory of the car park equipped with the video surveillance system presented above for 6 months. The markings were made by employees of the parking lot, instructed on the subject of the experiment.

As a result, a sample was obtained consisting of 2000 images (1000 in the training set and 1000 in the validation set), divided into 2 classes—"cars without damage" and "cars with damage" [42]. In addition, registration numbers were extracted from the images [43].

2.3 Data processing tools

To develop the algorithm, the Python 3.9.13 programming language with the Tensorflow and Keras libraries was used. The experimental platform was configured with an Intel Core i7-9700 K processor, GeForce RTX 2070 GPU, and 64 GB of RAM.

2.4 License plate recognition

The Viola-Jones method uses Haar features [44] to classify objects in an image. These functions are similar to convolution kernels and represent rectangular areas consisting of several adjacent parts. The Viola-Jones method uses the AdaBoost algorithm to build a cascade classifier. When forming each new level, the AdaBoost algorithm selects the most informative features. The formation of the classifier ends when the specified target quality of the classifier is reached.

The Viola-Jones method was used to recognize registration numbers on images in real time.. Here, the original image M(x, y) is transformed into an integral form of information representation V(i, j), which consists in analyzing the brightness of pixels in certain areas of the image with the shapes of rectangular fragments of the image in different areas. Mathematically, themethodisdescribedbythefollowingformula:

$$M(x,\ y) = \sum_{i=0}^{x} \sum_{j=0}^{y} V(i,\ j), \tag{1}$$

where *M (x, y)* – original image represented as a matrix, *V (i, j)* – variant intensity of an image pixel as a matrix element, *i*– image height, *j* – image width.

This formula describes the formation of matrix elements, each element stores the sum of the intensity of the image pixels, which are then used to apply a cascade classifier and quickly cut off areas of the image that are not of interest.

The process of cutting off non-informative areas of the image can be described by the formula:

$$V(i,\ j) = \begin{cases} \sum\limits_{1 \le s \le N} \sum\limits_{1 \le t \le N} V(s,\ t),\ 1 \le s \le i and 1 \le t \le j \\ 0, \qquad\qquad otherwise \end{cases}, \tag{2}$$

where *s*– desired values of variance of pixels along the height of the image, *t* – desired values of pixel variance across the width of the image.

It should be noted that for the correct operation of the cascade classifier, pre-processing of the input image was used (translation to grayscale, noise reduction, drawing image contours using local binarization, as well as the use of methods of mathematical morphology) [45–47].

At the first stage of image preprocessing, it should be understood that the input image needs to be converted to grayscale, since in this way it is possible to get rid of unnecessary interactions with other color gradations and speed up the algorithm (see Fig. 3a). Also, the image needs to get rid of excess noise and smooth the corners. At this stage, the intensity of the pixels is determined by the weighted average value of the radius of the neighboring pixels, that is, the Gaussian filter is applied.

Using the methods of local image binarization, as well as drawing image contours, it is possible to perform threshold processing, which consists in analyzing the features by which it is possible to calculate the values of each individual pixel and compare the entire given set with the threshold value. Here, for local image binarization, a method based on a statistical analysis of neighboring pixels is used (see Fig. 3b).

The result of the operation of Haar cascades with preliminary processing of the input image is shown in Fig. 3c. Here, the Viola-Jones method can be applied as an additional functionality of the security system, which is an ensemble of neural network algorithms and computer vision algorithms. Thanks to license plate detection, it is easy to identify which vehicle has been affected.

2.5 Methods for vehicle damage recognition

2.5.1 Using the HOG descriptor

The features of using the descriptor in relation to images of cars obtained in difficult weather conditions for shooting are described in detail in the article by N. Vasilyev et al. [48]. The HOG method is based on the assumption that the type of distribution of image intensity gradients allows one to accurately determine the presence and shape of the objects present on it.

The image is divided into cells. Histograms are calculated in cells h_i directional gradients of interior points. They are merged into one histogram $h = f(h_1,\ldots, h_k)$, after which it is normalized in brightness. The normalization factor was used h_L:

$$h_L = \frac{h}{\sqrt{\|h\|_2^2 + \varepsilon^2}}, \tag{3}$$

where h_2 – norm used, ε – some small constant.

When calculating gradients, the image is convolved with kernels $[-1, 0, 1]$ and $[-1, 0, 1]^T$, resulting in two matrices D_x and D_y derivatives along the x and y axes, respectively. These matrices are used to calculate the angles and magnitudes (modules) of the gradients at each point in the image.

On Fig. 4 shows the result of applying the HOG method to images of the vehicle before (see Fig. 4a, b) and after the

Fig. 3 Results of applying the Viola-Jones method for license plate recognition **a** converting the original image to grayscale using a Gaussian filter, **b** adaptive image binarization, **c** detected license plate

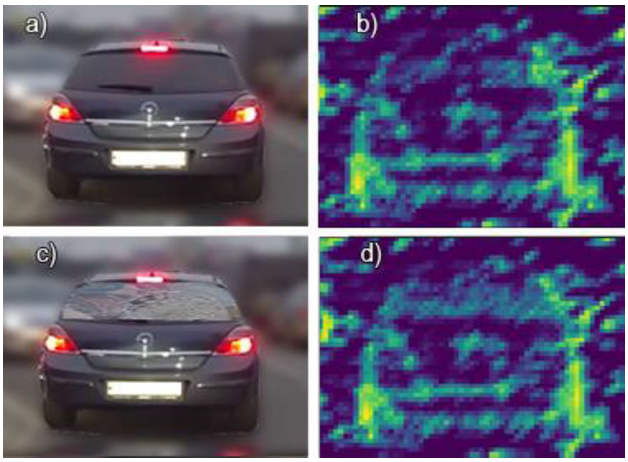

Fig. 4 The result of applying the HOG method to the images of the car before the misconduct (**a**, **b**) and after it (**c**, **d**)

misconduct (see Fig. 4c, d). Damage to the glass leads to an increase in the gradient in the corresponding HOG region. On Fig. 4d are the bright areas inside the histogram.

2.6 BOVW image classification

The Bag-of-visual-words (BOVW) method is used to improve the performance of descriptors. This approach treats blocks of an image as key parts, and the HOG of each block represents the local information of the corresponding part of the image. Then the HOGs of all blocks in the training sample are grouped into homogeneous groups using K-means, and the centers will be the averages of the HOGs of the blocks in the cluster (Fig. 5).

2.7 Convolutional neural networks

To solve the problem of recognizing a negative impact on a vehicle, we used modifications of convolutional artificial neural networks (ANNs). Convolutional Neural Networks Have Algorithm Resistant to Input Data Invariance [43].

The functioning of the ANN convolutional layer is described by the formula [49–51]:

$$x^{i-1} = f\left(x^i * c^i + b^i\right), \tag{4}$$

where x^{i-1} – feature map of the previous layer, $f()$ – activation function, b^i – parameter "bias" (displacement of the neuron), c^i – convolution kernel belonging to the layer$i$.

The functioning of the ANN subsample layer is described by the formula: (here the feature matrix is divided into new matrices of dimension nxn)

$$x_j^{i-1} = f\left(\sum_i x_j^i * c_j^i + b_j^i\right), \tag{5}$$

where x_j^{i-1} – feature map of the previous layer, $f()$ – activation function,b_j^i –"bias" parameter (neuron displacement) for the feature map, c_j^i – feature map convolution kernel j, belonging to the layer$i$.

For each ANN architecture under study, for the purpose of the purity of the experiment, the following hyperparameters were applied, which are tuned immediately before the start of training, in parallel with image preprocessing: 50 training epochs (epochs), batch-size of 32 units, adam optimizer, and a learning rate of $1*10^{-5}$.

Using the "transfer learning" technique, the ANN architectures discussed below were integrated into the Keras library. Then the parameters of the input data were adjusted (224 × 224 pixel images in RGB gradations, with all pixel values normalized in the range from 0 to 1). The input data of each pixel value is normalized in the range from 0 to 1, in order

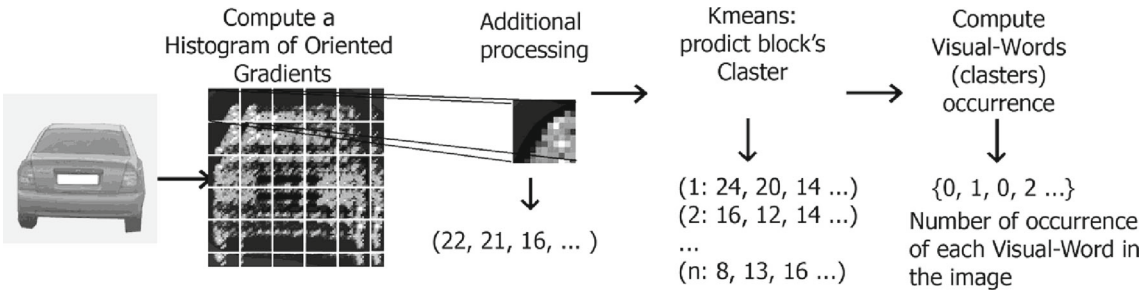

Fig. 5 Visualization of the BOVW method

to eliminate the influence of numbers greater than 1 on the learning process of the neural network:

$$f(i, min, \max) = \frac{i - min}{max - min}, \tag{6}$$

where f – normalization function, i – the value of a particular pixel, min – smallest pixel value, max – largest pixel value.

A fully connected ANN layer was also formed, which receives a feature vector as input. The fully connected layer consists of 128 neurons with two output neurons, the activation function of the convolutional and subsampling layer is ReLU (rectified linear function). Thefunctionisdescribed-bytheformula:

$$f(x) = max(0, x), \tag{7}$$

this function is used to cut off negative values, which makes it possible to simplify mathematical calculations and speed up the learning process of the neural network. The clipping of negative values is described by the formula:

$$f(x) = \begin{cases} 0, \ if \ x \leq 0 \\ x, \ if \ x > 0 \end{cases}, \tag{8}$$

The activation function of a fully connected layer is Softmax (returns the probability that an image belongs to a given class). Thefunctionisdescribedbytheformula:

$$f(x) = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}, \tag{9}$$

where f –activation function, e^{z_i} – standard exponential function for input vector,e^{z_k} –standard exponential function for output vector, K– number of classes.

The Keras library object "ImageDataGenerator" allows to perform additional data augmentation with a data set in order to expand the training and test sets of the ANN, since for the best results (in terms of generalizing non-standard input data), the set must have a high degree of data heterogeneity. Using the functionality of this object, the following data transformations were carried out:

- rotation_range – random rotation of the image (the value is set in degrees);
- zoom_range – random scaling of the image (the value is set by a floating point number);
- width_shift_range – random shift of the image in width (the value is set by a floating point number);
- height_shift_range – random shift of the image in height (the value is set by a floating point number);
- shear_range – random shift of image pixels (value is given by a floating point number);
- horizontal_flip – random image rotation by 180°about the axis y;
- fill_mode – filling image pixels that appear after applying a rotation or horizontal/vertical image shift.

In this study, the emphasis is on modifications to the connections between layers in order to improve the accuracy of existing ANN architectures. It is proposed to use a modified module for convolutional ANN architectures, called the Squeeze-and-Excitation block (SE), which makes it possible to enhance the generalizing ability of the ANN. The proposed mechanism makes it possible to amplify the feature map (the output of the convolutional layer), thereby the ANN generates a global feature map in order to more effectively cut off non-informative features and obtain a wider set of informative features). The input data for the modified mechanism is the feature vector of the convolutional layer, and the mechanism itself consists of two fully connected layers. The activation function of the first fully connected layer is ReLu, the activation function of the second fully connected layer is sigmoid. Here, the first fully connected layer performs the function of reducing the dimension of the feature vector, and the second layer performs the function of recalibrating the dimension to the original size. This block is integrated using Keras tools and is located in front of a fully connected ANN layer, which solves the classification problem. The structure of the SE block, which was integrated into subsequent architectures of convolutional ANNs, is shown in Fig. 6. The parameters of the modified SE-module are presented in Table 1.

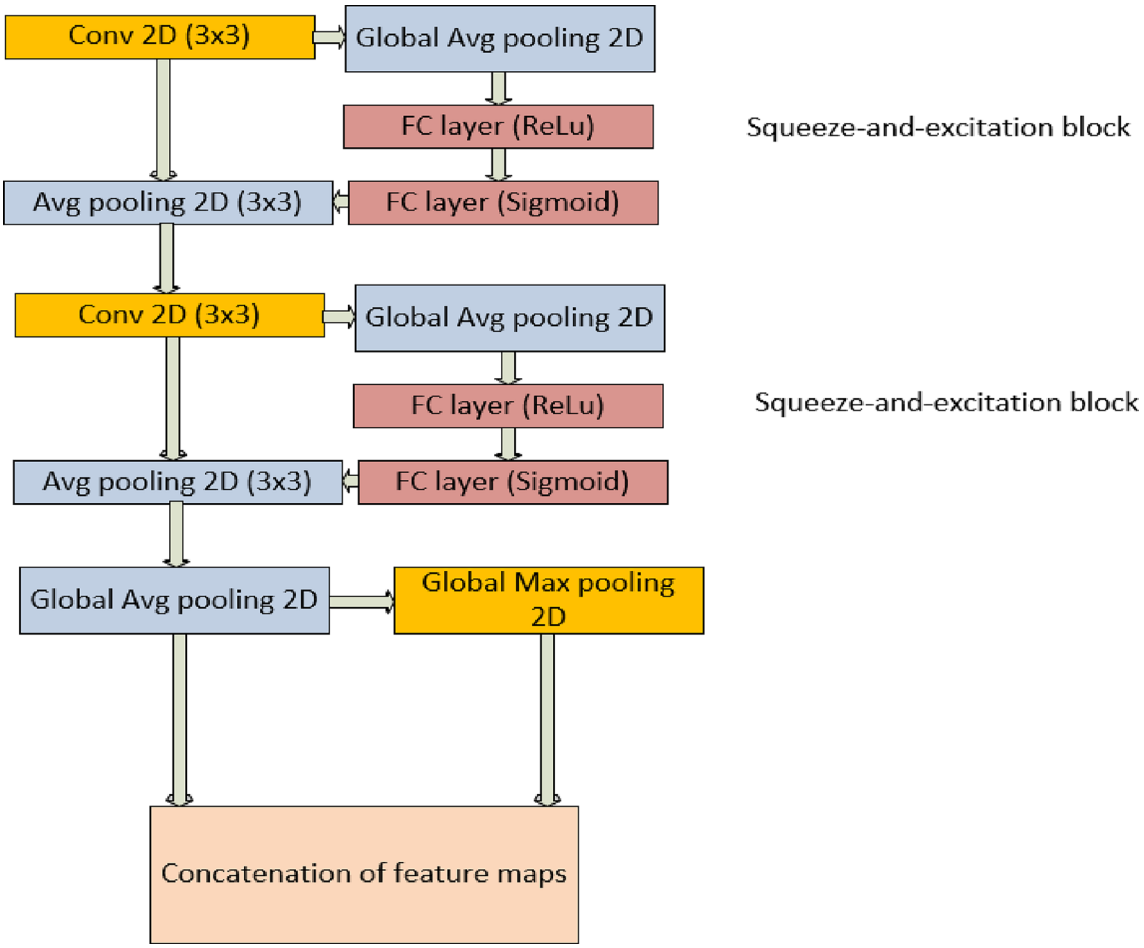

Fig. 6 Structure of the modified SE module

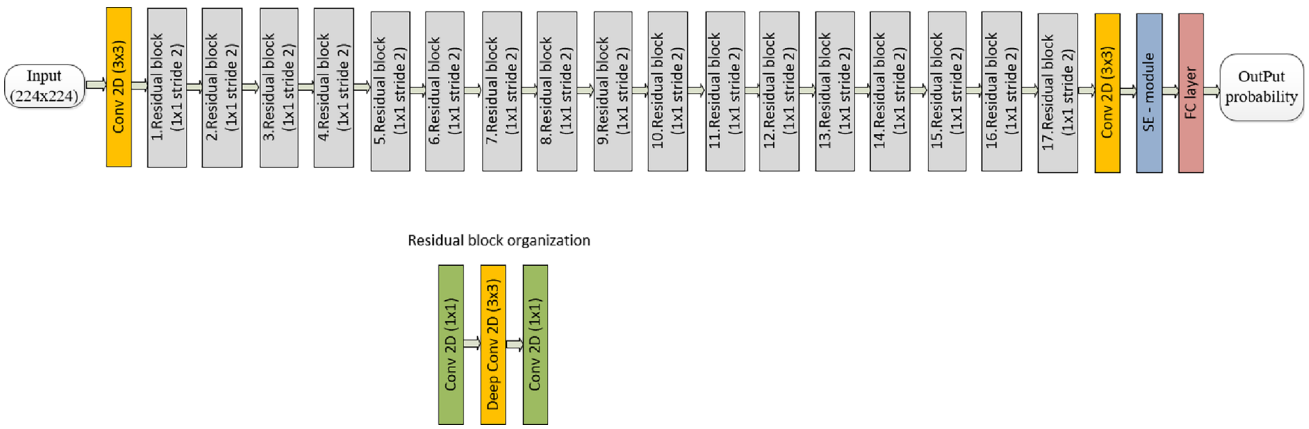

Fig. 7 Architecture of ANN "MobileNetV2 + SE"

MobileNetV2 [52] was chosen as the first architecture for research (see Fig. 7), since it is optimized in terms of memory consumption and can be integrated into mobile robotic systems.

A feature of this ANN architecture is that after a convolutional layer with a kernel size of 3 × 3 (which produces a feature map), there is another similar layer of 1 × 1 size with a convolution step of 2, which also analyzes the feature map of the previous layer. This operation is equivalent to the fact that the original image was analyzed by a 3 × 3 kernel, but in this case, in such a bundle, the total number of weights will be

Table 1 Settingsofthemodified SE-module

Nos	Layer name	Filter + Kernel size	Output size
1	Conv2D	32 × 3 × 3	7 × 7 × 32
2	Global Avg pooling	–	32
3	Dense	–	1 × 2
4	Dense_1	–	1 × 32
5	Multiply()	–	3 × 3 × 32
6	Avg pooling	32 × 3 × 3	7 × 7 × 32
7	Conv2D	32 × 3 × 3	7 × 7 × 32
8	Global Avg pooling	–	32
9	Dense_2	–	1 × 2
10	Dense_3	–	1 × 32
11	Multiply()	–	3 × 3 × 32
12	Avg pooling	32 × 3 × 3	7 × 7 × 32
13	Global Avg pooling	–	32
14	Global Ma × pooling	–	32
15	Concatenation	–	64

less than that of a 3 × 3 filter. All this optimizes the operation of the ANN and reduces memory consumption [53].

The second ANN architecture for the study was ResNet50 [54] (see Fig. 8), the architecture is a 50-layer convolutional ANN, which consists of 48 convolution layers, from 1 subsample layer by the maximum value (max pooling), and also from 1 subsample layer by average value (average pooling). This architecture of the ANN was created due to the fact that currently there is a problem of tenden- tial attenuation of gradients with an increase in the number of ANN layers. The solution to this problem was the introduction of ANN architectures based on the formation of residual links for a certain sequence of convolutional layers. Residual connections are expressed in the fact that the ANN is subdivided into a set of separate blocks that produce a non-linear signal transformation by skipping certain layers. Thus, when training an ANN, the objective function is optimized by creating labels of connections between layers, that is, not a complete, but a residual approximation of the objective function (differences in output values) occurs, which solves the problem of a vanishing gradient [55].

The first convolution layer has a 3 × 3 kernel, and subsequent convolution layers have a 1 × 1 kernel with a stride of 2. Both subsample layers have 3 × 3 kernels. Before the fully connected layer, there is a subsample layer by the mean value with a kernel size of 3 × 3. The fully connected layer in this ANN was generated manually, it, as in the case of the MobileNetV2 ANN, consists of 128 neurons with two output neurons, the activation function of the convolutional and subsample layers is ReLU (rectified linear function), the activation function of the fully connected layer is Softmax (returns the probability of belonging to the image to the given class).

The third ANN architecture for the study was DenseNet121 [56] (see Fig. 9), which continues the development of the ANN paradigm with residual connections to solve the problem of gradient attenuation with a large number of layers. The main feature of this ANN architecture is the concept of "dense blocks", which implies a set of 1 × 1 convolutional layers with a convolution step of 2, where the input of each subsequent layer is the concatenation of feature maps that were formed by the previous layers. This concept of "dense blocks" solves the problem of fading gradients [57].

The first convolution layer has a 7 × 7 kernel, then there is a subsampling layer by the maximum value (max pooling) with a 3 × 3 kernel. Then there are three bundles “dense block—convolutional layer—subsampling layer”, and on the fourth sheaf after the fourth dense block, a subsampling layer is used, which performs a global average selection operation in that it averages the values of a multivariate feature map into a 1 × 1 vector. The fully connected layer in this ANN was generated manually, as in the case of the MobileNetV2 and ResNet50 ANNs, it consists of 128 neurons with two output neurons, the activation function of the convolutional and subsample layers is ReLU (rectified linear function), the activation function of the fully connected layer is Softmax (returns the probability of membership images to a given class).

2.8 Research results

The training of the MobileNetV2 and MobileNetV2 + SE ANNs was performed over 50 epochs. The recognition accuracy for the first model was 85%, and for the second 88% (see Fig. 10). Binary crossentropy was chosen as the loss function, since the number of classes is two. At the beginning of training, the error in the first model was 2.25 and at the 50th epoch it decreased to 0.5, and in the second—0.7 and 0.25, respectively (see Fig. 11).

In addition to graphical visualization of ANN training results, the “classification_report” method, used on the basis of the “model.predict” method, allows you to generate a report in tabular form on the results of applying the trained ANN on previously unknown data. The report consists of the results of testing the NN for each given class according to the following metrics: accuracy, recall and the f1-score metric (harmonic mean between accuracy and recall), and these

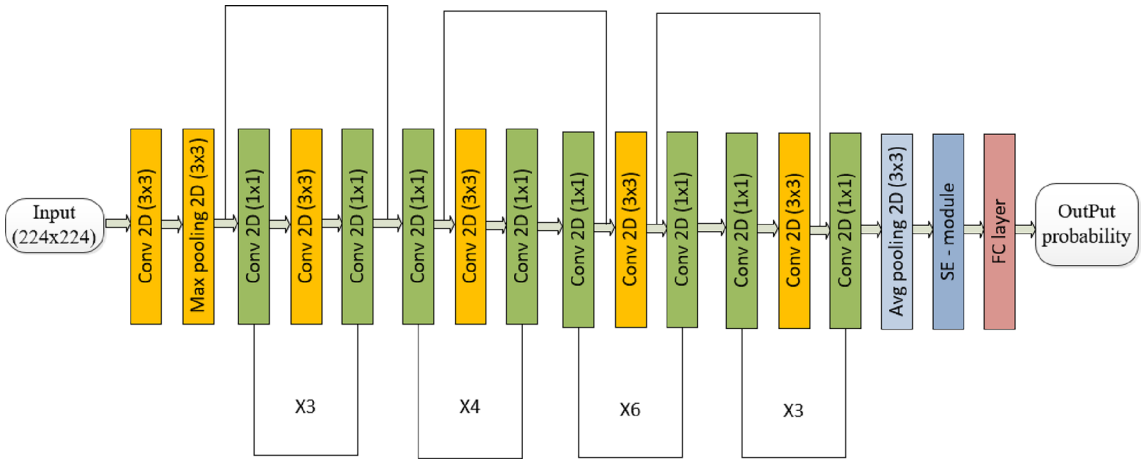

Fig. 8 Architecture of the ANN "ResNet50 +SE"

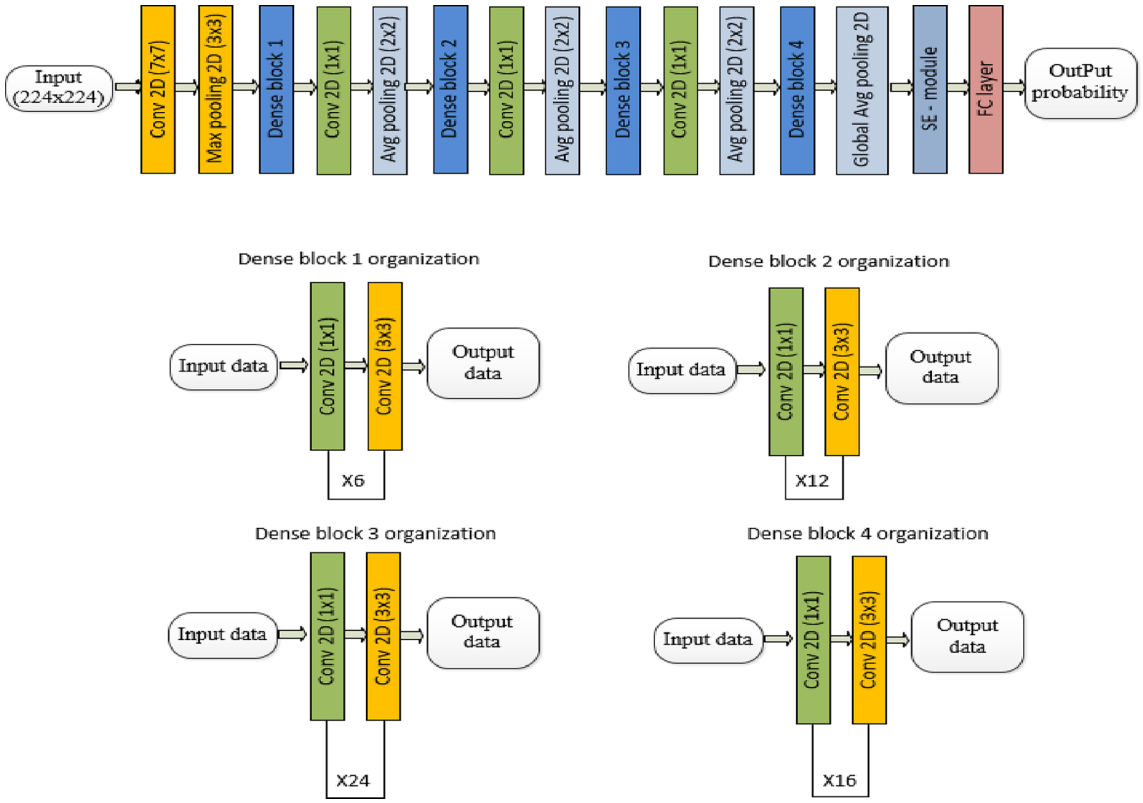

Fig. 9 Architecture of ANN "DenseNet121 +SE"

metrics are used to evaluate the model by the weighted average (weighted avg), as well as macro averaging (macro avg). To assess the quality of the trained models, the sklearn library and its specialized metrics module were used. The results of evaluating models on a test (unknown) data set for ANN are given in Table 2.

ResNet50 and ResNet50 + SE ANNs were trained over 50 epochs. The recognition accuracy for the first model was 90%, and for the second 91% (Fig. 12). Binary crossentropy was chosen as the loss function, since the number of classes is two. At the beginning of training, the error in the first model was 0.82, and at the 50th epoch it decreased to 0.27, and in the second—0.7 and 0.21, respectively (Fig. 13). The results of model evaluation on a test (unknown) data set for ANN are given in Table 3.

The DenseNet121 and DenseNet121 + SE ANNs were trained over 50 epochs. The recognition accuracy for the first model was 86%, and for the second—92% (Fig. 14). Binary

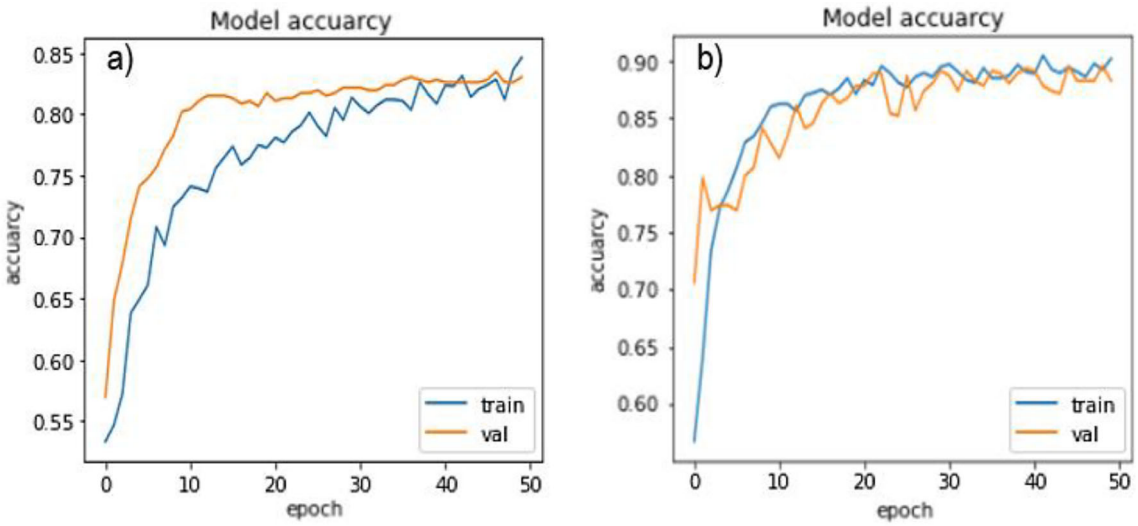

Fig. 10 Graph of the accuracy of the trained ANN MobileNetV2 (left) and MobileNetV2 + SE (right) depending on the training epoch

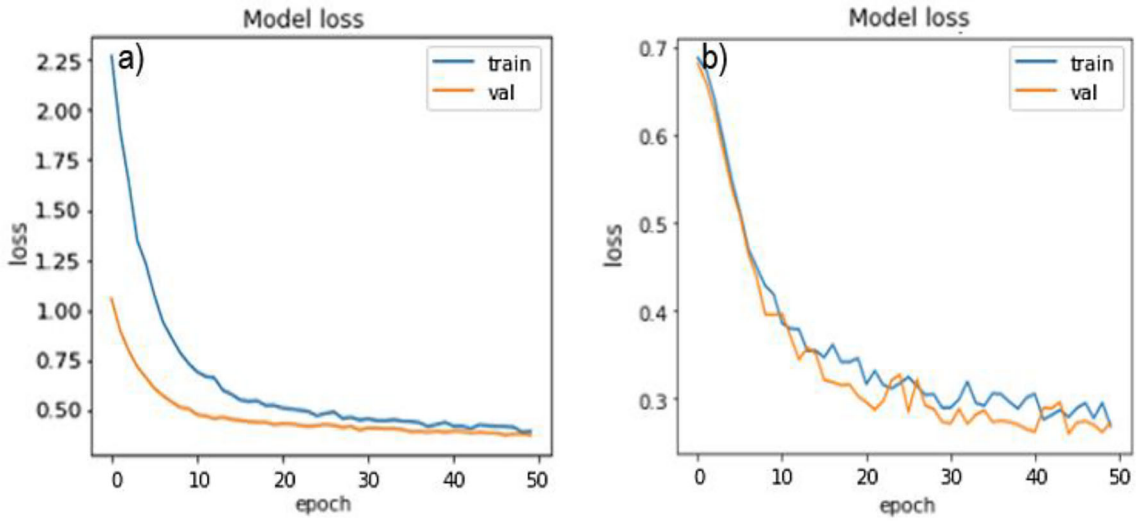

Fig. 11 Graph of the change in the error of the MobileNetV2 (left) and MobileNetV2 + SE (right) ANNs depending on the training epoch

Table 2 Evaluation of the trained ANN MobileNetV2 and MobileNetV2 + SE on the test data set, indicators with SE are on the right

Metrics	Accuracy		Recall		F1 score	
	MobileNetV2	MobileNetV2 + SE	MobileNetV2	MobileNetV2 + SE	MobileNetV2	MobileNetV2 + SE
Damagedcars	0.94	0.94	0.71	0.81	0.81	0.87
Carswithoutdamage	0.77	0.84	0.95	0.95	0.85	0.89
Macroaveraging	0.85	0.89	0.83	0.88	0.83	0.88
Weightedaverage	0.83	0.89	0.83	0.88	0.83	0.88

crossentropy was chosen as the loss function, since the number of classes is two. At the beginning of training, the error in the first model was 1.09, and at the 50th epoch it decreased to 0.36, and in the second—0.7 and 0.17, respectively (Fig. 15).

The results of model evaluation on a test (unknown) data set for ANN are given in Table 4.

Testing of trained architectures of convolutional ANNs on data unknown to them was carried out using the OpenCV computer vision library included in the Python programming

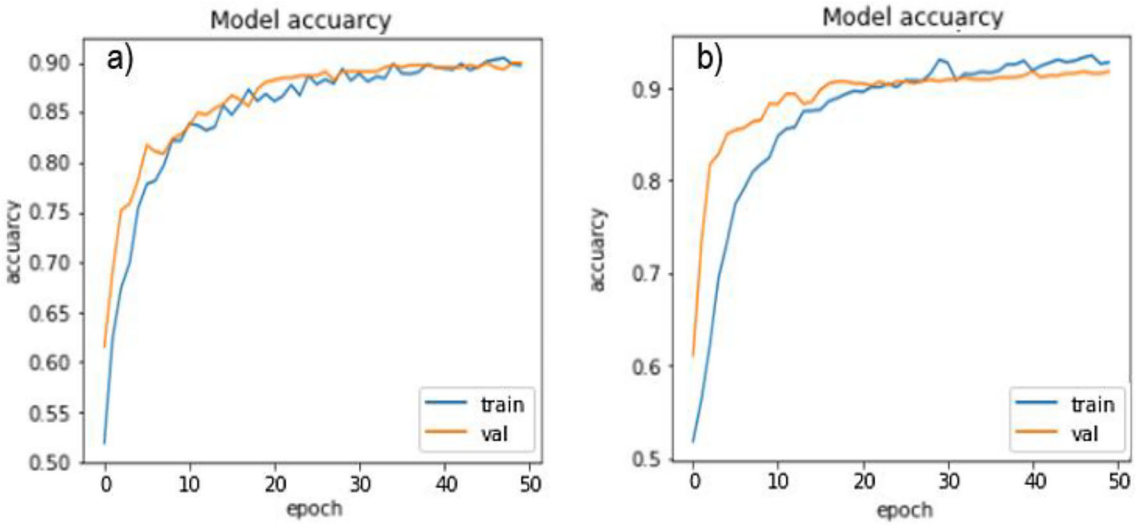

Fig. 12 Graph of the accuracy of the trained ANN ResNet50 (left) and ResNet50 + SE (right) depending on the training epoch

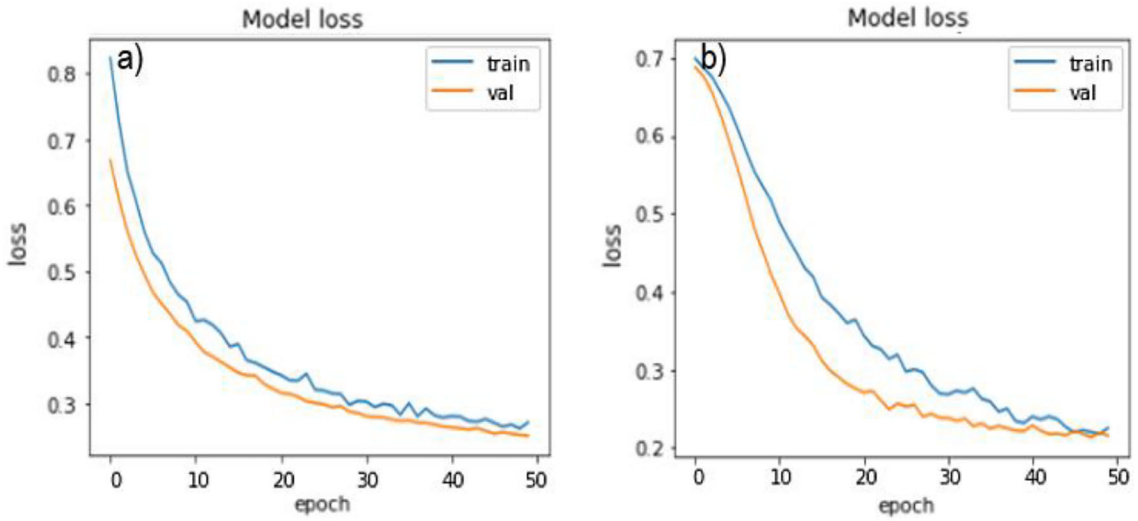

Fig. 13 Graph of the change in the error of the ANN ResNet50 (left) and ResNet50 + SE (right) depending on the training epoch

Table 3 Evaluation of the trained ANN ResNet50 and ResNet50 + SE on the test data set

Metrics	Accuracy		Recall		F1 score	
	ResNet50	ResNet50 + SE	ResNet50	ResNet50 + SE	ResNet50	ResNet50 + SE
Damagedcars	0.91	0.91	0.88	0.91	0.90	0.91
Carswithoutdamage	0.89	0.91	0.92	0.91	0.90	0.91
Macroaveraging	0.90	0.91	0.90	0.91	0.90	0.91
Weightedaverage	0.90	0.91	0.90	0.91	0.90	0.91

language environment. For testing, all images were compressed to a size of 224 × 224, then converted to a tensor of the format 224,224,3, and then the values of all pixels in the images were divided by 255 so that they were located in the range from 0 to 1. Examples of damage detection on vehicles are shown in Fig. 16.

As can be seen from Fig. 16, the presented ANN architectures did a good job of recognizing such semantic features as broken windows and dismantling of wheels. On Fig. 17. An

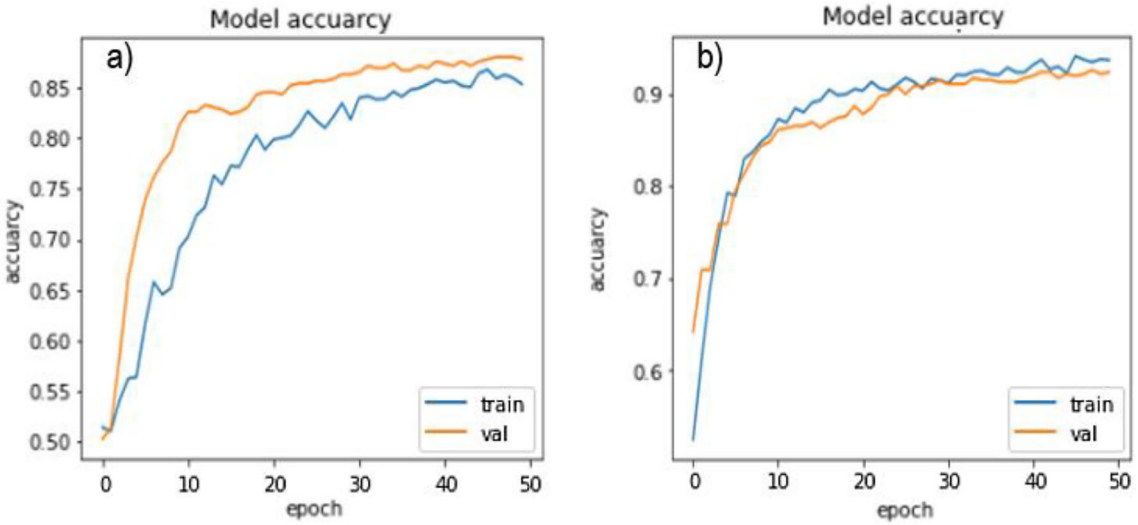

Fig. 14 Graph of the accuracy of the trained ANN DenseNet121 (left) and DenseNet121 + SE (right) depending on the training epoch

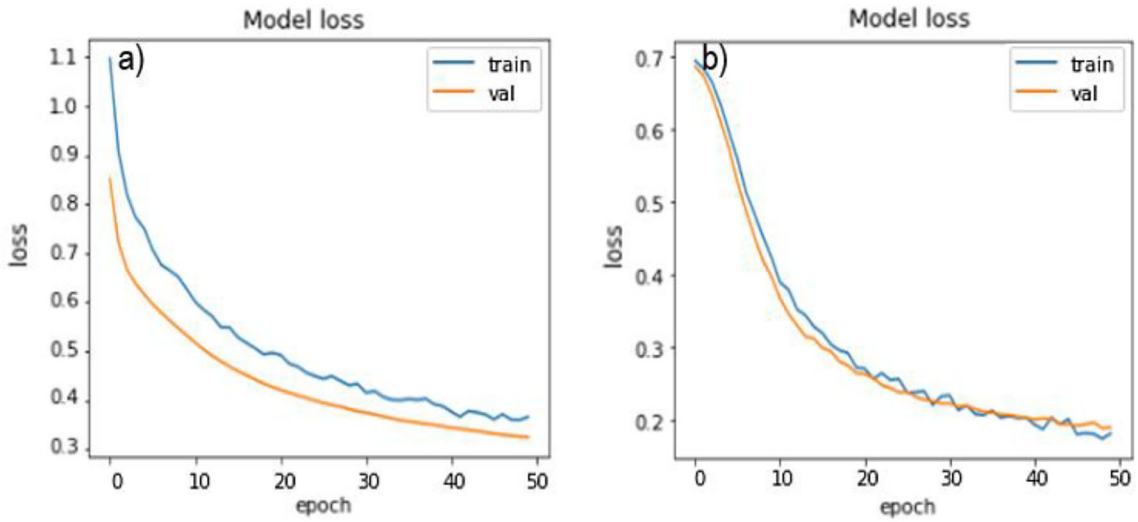

Fig. 15 Graph of the change in the error of the ANN DenseNet121 (left) and DenseNet121 + SE (right) depending on the training epoch

Table 4 Evaluation of the trained ANN DenseNet121 and DenseNet121 + SE on the test data set

Metrics	Accuracy		Recall		F1 score	
	DenseNet121	DenseNet121 + SE	DenseNet121	DenseNet121 + SE	DenseNet121	DenseNet121 + SE
Damagedcars	0.91	0.91	0.84	0.94	0.87	0.93
Carswithoutdamage	0.85	0.94	0.92	0.91	0.88	0.92
Macroaveraging	0.88	0.92	0.88	0.92	0.88	0.92
Weightedaverage	0.88	0.92	0.88	0.92	0.88	0.92

example of detection of scratches on the car body by ANN is given (Fig. 18).

Similarly, the presented ANNs give an answer if there is no damage to the machine (see Fig. 18).

2.9 Checking the performance of methods in difficult conditions for video filming

Difficult video shooting conditions include video filming during twilight, in rain, snow, etc. To test the performance of the

Fig. 16 Detection of car damage in images

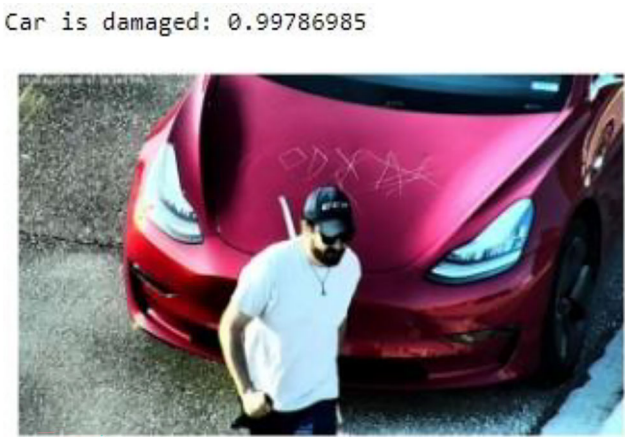

Fig. 17 Detection of scratches on the car body

presented ANN architectures, we used images in the dark, as well as images with the blur effect applied (here, the situation was simulated when the camera lens was dirty and, as a result, images were fed to the input of the ANN with defects). Images were about 20% of their total number.

It is supposed to take the HOG-BoVW-BPNN method as a standard for forming an estimate of the efficiency of the ANN on images taken in conditions that are difficult for video filming, which showed a fairly good invariance to the input data, which means it has a good ability to generalize data with varying degrees of affine image transformations, which in the context of performance assessment are various kinds of pixel intensities that occur with various defects in the input image caused by the external environment (but the method has a serious drawback—low speed).

The assessment of the accuracy of methods in difficult video recording conditions, for the purity of the experiment, was carried out according to the metric "Score F1", since here the use of this metric is due to the fact that in practice it is almost impossible to achieve maximum results simultaneously in terms of accuracy and recall metrics, therefore, in this case, it is necessary to find the average harmonic of these two metrics.

The results of supplying images to the input of the presented ANN architectures in the dark time of the day are shown in Fig. 19., and the results of feeding blurry images

Fig. 18 INS conclusions made if there is no damage to the car

Fig. 19 The results of the input of the INS images taken at night

Fig. 20 Results of feeding blurry images to the ANN input due to bad weather conditions, as well as when the camera lens is clogged

to the input presented by the ANN (both due to bad weather conditions and due to contamination of the camera lens) are shown in Fig. 20.

Thus, thanks to the use of the ImageDataGenerator functionality, the trained ANNs have a good ability to generalize the input data, which will only tend to increase with an increase in the number of images in the presented dataset. 19 and 20, trained ANNs have a good ability to generalize input data, which means that they are able to highlight key features even in images that have a high degree of affine transformations. Table 5 gives an assessment of the accuracy of the methods in difficult conditions of video filming by the metric « F1 score».

As can be seen from Table 5, the DenseNet121 + SE ANN according to the "F1 Score" metric showed identical results compared to the reference HOG-BoVW-BPNN method in

Table 5 Evaluation of the accuracy of methods in difficult conditions of video filming by metric "F1 score"

Metrics	F1 score			
	HOG-BoVW-BPNN	MobileNetV2 + SE	ResNet50 + SE	DenseNet121 + SE
Damagedcars	0.86	0.79	0.83	0.86
Carswithoutdamage	0.86	0.81	0.83	0.86
Macroaveraging	0.86	0.80	0.83	0.86
Weightedaverage	0.86	0.80	0.83	0.86

this experiment, which is an extremely worthy result, similarly, it should be noted that the DenseNet121 + SE ANN is faster in speed than the HOG- BoVW-BPNN by 40%, which makes it extremely promising for implementation in computer vision systems that are specialized in ensuring the safe operation of car parks.

3 Discussion

The use of ensembles of computer vision algorithms and the mathematical apparatus of ANNs in the framework of the problem of pattern recognition has already been touched upon earlier in scientific research. For example, work [58] describes the development of a mathematical model for the rapid and efficient detection of medical masks in public places based on Haar cascades and convolutional ANNs. The authors explained such an ensemble of algorithms in the mathematical model by the fact that the object of interest can be easily implemented by the Haar cascade, because it can serve as a unique attribute of the image, and convolutional ANNs were used to detect anomalies in the images (lack of a medical mask on the face). This work was published in 2021, when COVID-19 had a significant impact on all areas of life. This emphasizes the relevance and practical significance of using ensembles of computer vision algorithms and the mathematical apparatus of ANN.

In [59], the paradigm of applying computer vision and deep learning methods was used to solve the problems of detecting objects in a video stream with various kinds of defects (when measuring data with video recorders in poor visibility conditions). Images from the video stream obtained from video recorders were measured using a lens that was covered with drops of water or dirt, which introduced defects into the original data and served as the objects of study. In the article, the authors proposed a method for improving the accuracy of object detection in images with a complex background by using the Canny method, which excludes defective areas from image analysis by capturing smoothed areas. Only those areas of the image where the Canny method detected deterministic contours of objects fall into further analysis. To classify these contours, selected at the time of the Canny method, the authors proposed the use of histograms of directional gradients (HOG—descriptors), the "bag-of-words" method and the training of a multilayer perceptron, which is a backpropagation ANN. Such an ensemble of machine learning algorithms in the context of solving the problem of binary image classification showed a significant advantage over the use of convolutional ANNs (accuracy was 79% versus 65% for convolutional ANNs). Based on the results of the research, the authors came to the conclusion that the use of such an ensemble as computer vision methods in combination with the deep learning paradigm is promising for modern scientific research and can significantly improve the quality of object recognition in images.

4 Conclusion

The practically significant problem of classifying images of vehicles according to various semantic features corresponding to both their normal state and various types of damage has been solved. A description of ANN models based on the MobileNetV2, ResNet50 and DenseNet121 architectures, algorithmically implemented in the Python programming language environment (using the functionality of the Keras machine learning library) is given. Test studies of each of the models showed a good detection of semantic signs of damage caused by the attacker's vandalism, and also demonstrated good values for the quality metrics of machine learning models. The proposed mathematical models demonstrate a new approach to ensuring the security of objects in protected areas (parking lots), and are also suitable for implementation on the platform of mobile robotic systems. The implemented ANN models will be used to control a mobile security robot.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Suthir, S., Harshavardhanan, P., Subramani, K., Senthil, P., Veena, T., Faith, S.J., Nivethitha, V.: Conceptual approach on smart car

parking system for industry 4.0 internet of things assisted networks. Meas. Sens. **24**, 1–6 (2022)
2. Parygin, D.: Implementation of exoactive management model for urbanized area: real-time monitoring and proactive planning. In: Proceedings of the 2019 8th International Conference on System Modeling and Advancement in Research Trends, pp. 310–316. https://doi.org/10.1109/SMART46866.2019.9117298
3. Parygin, D., Usov, A., Burov, S., Sadovnikova, N., Ostroukhov, P., Pyannikova, A.: Multi-agent approach to modeling the dynamics of urban processes (on the example of urban movements). Commun. Comput. Inf. Sci., 2020, 243–257
4. Abdellatif, M.M., Elshabasy, N.H., Elashmawy, A.E., AbdelRaheem, M.: A low cost IoT-based Arabic license plate recognition model for smart parking systems. Ain Shams Eng. J. **14**, 1–6 (2023)
5. Kumagai, H., Kawaguchi, K., Sawatari, H., Kiyohara, Y., Hayashi, M., Shiomi, T.: Dashcam video footage-based analysis of microsleep-related behaviors in truck collisions attributed to falling asleep at the wheel. Accid. Anal. Prev. **187**, 1–9 (2023)
6. Kanan, R., Arbess, H.: An IoT-based intelligent system for real-time parking monitoring and automatic billing. In: 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT). IEEE, pp. 622–626
7. Finogeev, A., Finogeev, A., Fionova, L., Lyapin, A., Lychagin, K.: Intelligent monitoring system for smart road environment. J. Ind. Inf. Integr.
8. Andriyanov, N., Khasanshin, I., Utkin, D., Gataullin, T., Ignar, S., Shumaev, V., Soloviev, V.: Intelligent system for estimation of the spatial position of apples based on YOLOv3 and real sense depth camera D415. Symmetry **14**, 148 (2022). https://doi.org/10.3390/sym14010148
9. Ivanyuk, V.: Forecasting of digital financial crimes in Russia based on machine learning methods. J. Comput. Virol. Hack. Tech. (2023). https://doi.org/10.1007/s11416-023-00480-3
10. Boltachev, E.: Potential cyber threats of adversarial attacks on autonomous driving models. J. Comput. Virol. Hack. Tech. (2023). https://doi.org/10.1007/s11416-023-00486-x
11. Sergi, B.S., Popkova, E.G.: Towards a 'wide' role for venture capital in OECD countries' industry 4.0. Heliyon **8**, e08700 (2022)
12. Mhlanga, D.: Artificial intelligence in the industry 4.0, and its impact on poverty, innovation, infrastructure development, and the sustainable development goals: Lessons from emerging economies? Sustainability **13**, 5788 (2021). https://doi.org/10.3390/su13115788
13. Olah, J., Aburumman, N., Popp, J., Asif Khan, M., Haddad, H., Kitukutha, N.: Impact of industry 4.0 on environmental sustainability. Sustainability **12**, 4674 (2021). https://doi.org/10.3390/su12114674
14. Boyar-Sozonovitch, A.S., Buikin, A.Y., Pitelinskiy, K.V.: Features of enterprise risk management associated with operational risks. Amazonia Investiga **10**(46), 9–19 (2021). https://doi.org/10.34069/AI/2021.46.10.1
15. Macea, L.F., Serrano, I., Carcache-Guas, C.: A reservation-based parking behavioral model for parking demand management in urban areas. Socio-Econ. Sci. **86**, 1–15 (2023)
16. Hollerer, S., Fischer, C., Brenner, B., Papa, M., Schlund, S., Kastner, W., Fabini, J., Zseby, T.: Cobot attack: a security assessment exemplified by a specific collaborative robot. Procedia Manuf. **54**, 191–196 (2021)
17. Patwal, A., Diwakar, M., Tripathi, V., Singh, P.: An investigation of videos for abnormal behavior detection. Procedia Comput. Sci. **218**, 2264–2272 (2023)
18. Boyar-Sozonovitch, A.S., Pitelinskiy, K.V., Ermolatiy, D.A.: Innovation economy: aspects of economic and information security in logistics innovation. Amazonia Investiga **8**(21), 6–13 (2019)
19. Shah, N., Bhagat, N., Shah, M.: Crime forecasting: a machine learning and computer vision approach to crime prediction and prevention. Vis. Comput. Ind. Biomed. Art **4**, 9 (2021). https://doi.org/10.1186/s42492-021-00075-z
20. Wibowo A.H., Oesman T.I.: The comparative analysis on the accuracy of k-NN, naive Bayes, and decision tree algorithms in predicting crimes and criminal actions in Sleman regency. J. Phys. Conf. Ser. **1450**, 012076 (2020). https://doi.org/10.1088/1742-6596/1450/1/012076
21. Hossain, S., Abtahee, A., Kashem, I., Hoque, M., Sarker, I.H.: Crime prediction using spatio-temporal data. arXiv preprint arXiv:2003.09322 (2020). https://doi.org/10.1007/978-981-15-6648-6_22
22. Bandekar, S.R., Vijayalakshmi, C.: Design and analysis of machine learning algorithms for the reduction of crime rates in India. Procedia Comput. Sci. **172**, 122–127 (2020). https://doi.org/10.1016/j.procs.2020.05.018
23. Chen, Y., Ping, Y., Zhang, Z., Wang, B., He, S.: Privacy-preserving image multi-classification deep learning model in robot system of industrial IoT. Neural Comput. Appl. **33**, 4677–4694 (2021)
24. Natsui, S., Goto, Y., Takahashi, J.-I., Nogami, H.: Pattern analysis of the combustions of various copper concentrate tablets using high-speed microscopy and video-based deep learning. Chem. Eng. Sci. **276**, 1–12 (2023)
25. Prithi, S., Aravindan, S., Anusuya, E., Kumar, A.M.: GUI based prediction of crime rate using machine learning approach. Int. J. Comput. Sci. Mob. Comput. **9**(3), 221–229 (2020)
26. Khan, M., Tanveer, H., Sung, W.B.: Efficient CNN based summarization of surveillance videos for resource-constrained devices. Pattern Recognit. Lett. (2020). https://doi.org/10.1016/j.patrec.2018.08.003
27. Qasim, M., Verdu, E.: Video anomaly detection system using deep convolutional and recurrent models. Results Eng. **18**, 1–9 (2023)
28. Asif, M., Tiwana, M.I., Khan, U.S., Ahmad, M.W., Qureshi, W.S., Iqbal, J.: Human gait recognition subject to different covariate factors in a multi-view environment. Results Eng. **15**, 100556 (2022)
29. Gandapur, M.Q.: E2E-VSDL: end-to-end video surveillance-based deep learning model to detect and prevent criminal activities. Image Vis. Comput. **123**, 104467 (2022)
30. Socha, R., Kogut, B.: Urban video surveillance as a tool to improve security in public spaces. Sustainability **12**(15), 6210 (2020)
31. Rezaee, K., Rezakhani, S.M., Khosravi, M.R., Moghimi, M.K.: A survey on deep learning-based real-time crowd anomaly detection for secure distributed video surveillance. Personal Ubiquitous Comput. 1–17 (2021)
32. Zhang, H., Li, P., Du, Z., Dou, W.: Risk entropy modeling of surveillance camera for public security application. IEEE Access **8**, 45343–45355 (2020)
33. Yamashkina, E.O., Yamashkin, S.A., Platonova, O.V., Kovalenko, S.M.: Development of a neural network model for spatial data analysis. Russ. Technol. J. **10**(5), 28–37 (2022). https://doi.org/10.32362/2500-316X-2022-10-5-28-37.
34. Han, S.-Y., Lee, H.-W.: Deep reinforcement learning based edge computing for video processing. ICT Express (2022). https://doi.org/10.1016/j.icte.2022.05.001
35. Ullah, W., Ullah, A., Hussain, T., Muhammad, K., Heidari, A.A., Del Ser, J., WookBaik, S.C., De Albuquerque, V.H.: Artificial intelligence of things-assisted two-stream neural network for anomaly detection in surveillance big video data. Future Gen. Comput. Syst. **129**, 286–297 (2022). https://doi.org/10.1016/j.future.2021.10.033
36. Blin, R., Ainouz, S., Canu, S., Meriaudeau, F.: Road scenes analysis in adverse weather conditions by polarization-encoded images and adapted deep learning. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 27–32. https://doi.org/10.1109/ITSC.2019.8916853
37. Sharifrazi, D.: Fusion of convolution neural network, support vector machine and Sobel filter for accurate detection of COVID-19

patients using X-ray images. Biomed. Signal Process. Control **68**, 102622 (2021). https://doi.org/10.1016/j.bspc.2021.102622
38. Pawar, K.B., Nalbalwar, S.L.: Distributed canny edge detection algorithm using morphological filter. In: Recent Trends in Electronics Information & Communication Technology (RTEICT) IEEE International Conference, 2016, pp. 1523–1527
39. Kumar, M.D., Babaie, M., Zhu, S., Kalra, S., Tizhoosh, H.R.: A comparative study of CNN, BoVW and LBP for classification of histopathological images. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 2017, pp. 1–7. https://doi.org/10.1109/SSCI.2017.8285162
40. Deng, C.-X., Gui-Bin Wang, G.-B., Yang, X.-R.: Image edge detection algorithm based on improved Canny operator. In: 2013 International Conference on Wavelet Analysis and Pattern Recognition, 2013, pp. 168–172. https://doi.org/10.1109/ICWAPR.2013.6599311
41. www.kaggle.com. Cardamagedetection [Электронный ресурс], URL: https://www.kaggle.com/datasets/anujms/car-damage-detection (дата обращения - 03.04.2023)
42. Alghamdi, A.S., Saeed, A., Kamran, M., Mursi, K.T., Almukadi, W.S.: Vehicle classification using deep feature fusion and genetic algorithms. Electronics **12**(280), 1–14 (2023)
43. Bie, M., Liu, Y., Li, G., Hong, J., Li, J.: Real-time vehicle detection algorithm based on a lightweight You-Only-Look-Once (YOLOv5n-L) approach. Expert Syst. Appl. **213B**, 119108 (2023). https://doi.org/10.1016/j.eswa.2022.119108
44. Soleimanipour, A., Chegini, G.R.: A vision-based hybrid approach for identification of Anthurium flower cultivars. Comput. Electron. Agric. **174**, 05460 (2020)
45. Chen, J., Cai, Z., Heidari, A.A., Chen, H., He, Q., Escorcia-Gutierrez, J., Romany, F.M.: Multi-threshold image segmentation based on an improved differential evolution: case study of thyroid papillary carcinoma. Biomed. Signal Process. Control **85**, 104893 (2023). https://doi.org/10.1016/j.bspc.2023.104893
46. Liu, H., Yang, Z., Zhang, H., Cailing, W.: Edge detection with attention: from global view to local focus. Pattern Recognit. Lett. **154**, 99–109 (2022). https://doi.org/10.1016/j.patrec.2022.01.006
47. Yang, D., Peng, B., Al-Huda, Z., Malik, A., Zhai, D.: An overview of edge and object contour detection. Neurocomputing **488**, 470–493 (2022). https://doi.org/10.1016/j.neucom.2022.02.079
48. Poornima, E., Muthu, B., Agrawal, R., Kumar, S.P., Dhingra, M., Asaad, R.R., Jumani, A.K.: Fog robotics-based intelligence transportation system using line-of-sight intelligent transportation. Multimedia Tools Appl., 1–29 (2023)
49. Park, J., Jun, M.B.G., Yun, H.: Development of robotic bin picking platform with cluttered objects using human guidance and convolutional neural network (CNN). J. Manuf. Syst. **63**, 539–549 (2022). https://doi.org/10.1016/j.jmsy.2022.05.011
50. RetnoKinasih, F.M.T., Machbub, C., Yulianti, L., Rohman, A.S.: Two-stage multiple object detection using CNN and correlative filter for accuracy improvement. Heliyon **9**(1), e12716 (2023). https://doi.org/10.1016/j.heliyon.2022.e12716
51. Zhang, H., Feng, L., Zhang, X., Yang, Y., Li, J.: Necessary conditions for convergence of CNNs and initialization of convolution kernels. Digit. Signal Process. **123**, 1–12 (2022)
52. www.tensorflow.org/. MobileNetV2 [Электронный ресурс]. https://www.tensorflow.org/api_docs/python/tf/keras/applications/mobilenet_v2/MobileNetV2 (дата обращения - 12.05.2023)
53. Shamrat, F.M.J.M., Azam, S., Karim, A., Ahmed, K., Bui, F.M., De Boer, F.: High-precision multiclass classification of lung disease through customized MobileNetV2 from chest X-ray images. Comput. Biol. Med., **155**, 1–14 (2023)
54. www.tensorflow.org/. ResNet50 [Электронный ресурс], URL:https://www.tensorflow.org/api_docs/python/tf/keras/applications/resnet50/ResNet50 (дата обращения - 12.05.2023)
55. Md. Hossain, U., Md. Rahman, A., Md. Manowarul, I., Akhter A., Md. Uddin, A., Bikash Kumar, P.: Automatic driver distraction detection using deep convolutional neural networks. Intell. Syst. Appl. **14**, 1–12 (2022)
56. www.tensorflow.org/. DenseNet121 [Электронный ресурс]. https://www.tensorflow.org/api_docs/python/tf/keras/applications/densenet/DenseNet121(дата обращения - 12.05.2023)
57. SalamaW, M., Aly, M.H., Abouelseoud, Y.: Deep learning-based spam image filtering. Alex. Eng. J. **68**, 461–468 (2023)
58. Sai, B., Yalla, L., Kaushik, P.: Face mask detection in images using Haar cascade classifier. Int. Res. J. Mod. Eng. Technol. Sci. **3**(6), 3366–3372 (2021)
59. Osipov, A., Pleshakova, E., Gataullin, S., Korchagin, S., Ivanov, M., Finogeev, A., Yadav, V.: Deep learning method for recognition and classification of images from video recorders in difficult weather conditions. Sustainability **14**, 2420 (2022). https://doi.org/10.3390/su14042420