Scientific Research Publishing

Journal of Data Analysis and Information Processing, 2022, 10, 127-141
https://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

Detection and Selection of Moving Objects in Video Images Based on Impulse and Recurrent Neural Networks

Ihar Yeuseyenka[1]*, Ihar Melnikau[2], Ihar Yemelyanov[3]

[1]Research Center of Automotive Power Train and Electronic Control, College of Automotive Engineering, Hubei University of Automotive Technology, Shiyan, China
[2]Department of Asset Performance and Information Management, AVEVA Solutions Ltd., Cambridge, The United Kingdom
[3]Laboratory of Mathematical Cybernetics, United Institute of Informatics Problems, Minsk, Belarus
Email: *yeuseyenkaihar@gmail.com

How to cite this paper: Yeuseyenka, I., Melnikau, I. and Yemelyanov, I. (2022) Detection and Selection of Moving Objects in Video Images Based on Impulse and Recurrent Neural Networks. *Journal of Data Analysis and Information Processing*, **10**, 127-141.
https://doi.org/10.4236/jdaip.2022.102008

Received: February 1, 2022
Accepted: April 26, 2022
Published: April 29, 2022

Abstract

The purpose of the article is to develop a methodology for automating the detection and selection of moving objects. The detection and separation of moving objects based on impulse and recurrence neural networks simulation. The result of the work is a developed motion detector based on impulse and recurrence neural networks and an automated system developed on the basis of this detector for detecting and separating moving objects and is ready for practical application. The feasibility of integrating the developed motion detector with Emgu CV (OpenCV) image processing package, multimedia framework functions, and DirectShow application programming interface were investigated. The proposed approach and software for the detection and separating of moving objects in video images using neural networks can be integrated into more sophisticated specialized computer-aided video surveillance systems, IoT (Internet of Things), IoV (Internet of Vehicles), etc.

Keywords

Automated System, Video Image, Pixel, Neuron, Neural Network

1. Introduction

The continuous population growth, coupled with the automotive industry development made it necessary for the implementation of Intelligent Transport Systems (ITS). ITS is a core part of intelligent logistics and IoV (Internet of Vehicles). Previous approaches, such as widening roads, and increasing their number, have become useless. This is especially evident in the large cities of the

United States of America (USA) and China [1]. The efficient service of the existing road networks and their optimal development required the introduction of modern high-tech solutions. The integrated implementation of technologies and automation tools focused on providing optimal transport services to traffic participants has become increasingly used in the world. ITSs are widely implemented today and continue to develop in Korea, Japan, the USA, Germany, Australia, and other countries [1] [2].

Recently, much attention has been paid to safety in transport systems. For example perimeter defender (protecting the perimeter of the object), aggression detection, gunshot sounds, glass breakage, smoke and fire detection, inspection and response to incidents in real-time, etc. Such systems can be used independently and as a part of the IoT (Internet of Things)/IoV. The latter can significantly improve the efficiency of traffic management with a rapidly growing vehicles fleet.

Currently, a large number of automated transport systems are in use that allows traffic control. These systems make it possible to effectively manage traffic, but they need continuous information about the traffic density in real-time and its identification (for example, the composition of traffic: trucks, minibusses, cars, motorcycles, bicycles, pedestrians, etc.). Information is collected mainly based on infrared and radar detectors, which have certain disadvantages. At the same time, more and more video surveillance systems appear on the roads, which are used mainly for violations of video recording. However, the potential of such systems is much greater: they can be integrated with smart city applications and IoV or used for automated analysis of the traffic situation with statistical information collecting, which is very necessary for rapidly growing cities. Therefore, there is a need to ensure continuous moving objects monitoring/control on various road sections, regardless of weather conditions and time of day, which is also a far from solved problem in many cities.

Before you can directly recognize moving objects, you must extract them from static background. For this, there are a number of deterministic foreground extraction methods: background subtraction methods, time difference methods, optical flow methods, etc. The choice of method greatly affects the efficiency of the entire recognition system [1]-[7]. And the more efficient the method is, the more complex it is, and requires more resources.

The purpose of this work is to present a fundamentally different approach to digital video processing, in particular to the separation of moving objects based on neural networks. This will make it possible to create an effective motion detector that can be an alternative to detectors developed on the basis of deterministic methods, including an improved way of parallel processing of video segments.

Automated AI-based selection and recognition of moving objects is a promising area of research in digital video processing and pattern recognition [8] [9]. It makes it possible, without human intervention, to control the technological process, record violations in the field of road traffic, and carry out automated information collection and analysis, etc.

The relevance of research into the use of artificial neural networks in pattern recognition [10] is related to the fact that how the human brain processes information is fundamentally different from that employed by conventional digital computers [11]. The brain is an extremely complex, non-linear, parallel computer (an information processing system). It has the ability to organize its structural components, called neurons so that they can perform specific tasks (such as pattern recognition, sensory signal processing, motor functions) many times faster than the fastest modern computers can afford. Ordinary vision is an example of this task of information processing. The function of the visual system is to create an image of the environment in a way that enables interaction with that world. More precisely, the brain successively performs a number of recognition tasks (such as recognizing a familiar face in an unfamiliar environment). It takes about 100 - 200 milliseconds to do this, while even smaller tasks on a computer can take several days.

The objective of this work is to develop a methodology for automating the detection and selection of moving objects, a motion detector based on impulse and recurrence neural networks and an automated system developed on the basis of this detector in order to integrate it with conventional video surveillance systems used for violations video recording.

2. Model of Neuron Impulse Network to Select Moving Objects

The general structure of the impulse neural network used to isolate moving objects and used in the detector being developed is shown in **Figure 1** [12] [13]. The input layer of neurons is an analog of the retinal photoreceptor layer, so what we're going to call the first layer neurons receptors. Each pixel (x, y) of the input frame of the video image corresponds to its receptor $N_r(x, y)$. The hidden layer is an analog of the retinal interlayer of the eye. It consists of two independent arrays of neurons N_1 and N_2. They are the same size as the first layer and are connected by synaptic connections to both the input layer of neurons (receptors) N_r and the output layer of neurons N_{out}.

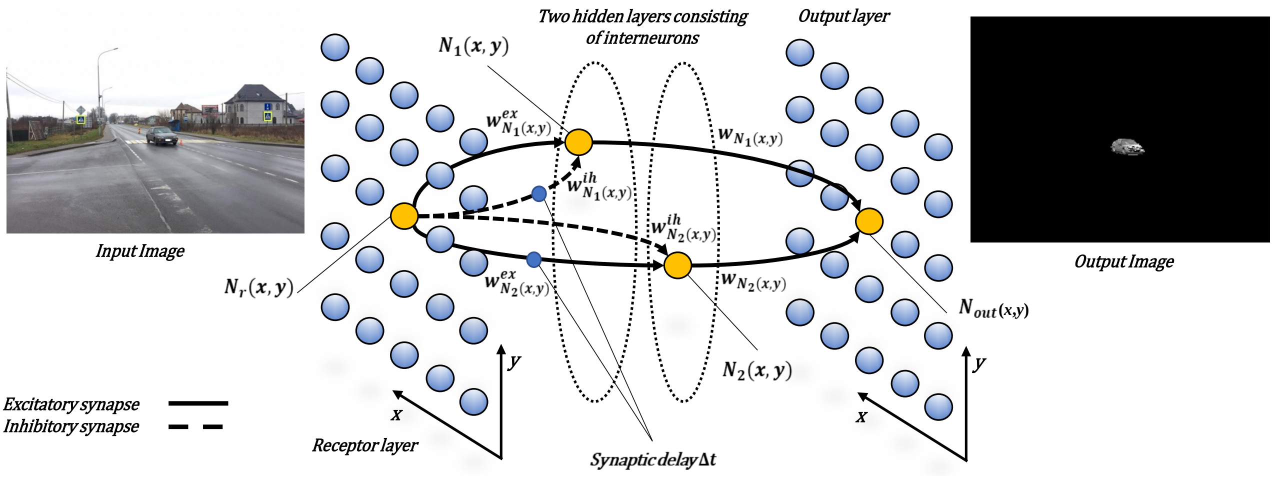
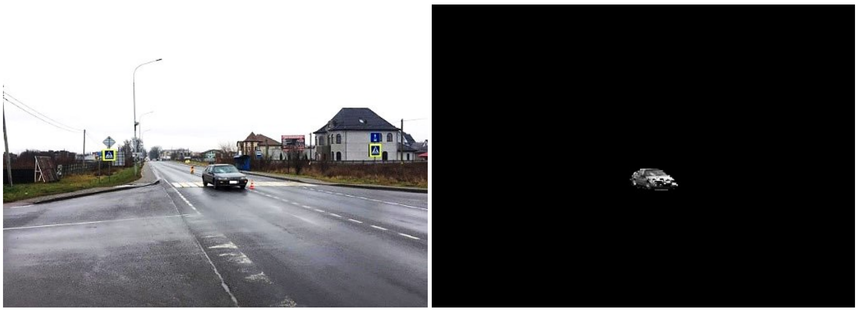


Figure 1. Neural impulse network model of Motion detector.

Each receptor $N_r(x,y)$ is connected to the interneuron $N_1(x,y)$ through an excitable synapse that transmits signals without delay and an inhibiting synapse that transmits signals with synaptic delay Δt. Each receptor $N_r(x,y)$ is also connected to the interneuron $N_2(x,y)$ by means of an excitating synapse transmitting signals with synaptic delay Δt, and an inhibiting synapse transmitting signals without delay. Let's suppose that $I_{nr}(x,y,t)$—the current flowing from the receptor $N_r(x,y)$ at a time *t*. If the current flowing from the receptor $N_r(x,y)$ is stable, *i.e.* $I_{nr}(x,y,t)=I_{nr}(x,y,t-\Delta t)$, and the excitatory and inhibitory inputs of the interneuron $N_1(x,y)$ are balanced by adjusting the weights $w^{ex}_{N_1(x,y)}$ and $w^{ih}_{N_1(x,y)}$ accordingly, the interneuron $N_1(x,y)$ will be at rest. The situation is similar for interneuron $N_2(x,y)$. If the current of the receptor $N_r(x,y)$ increases, *i.e.* $I_{nr}(x,y,t)>I_{nr}(x,y,t-\Delta t)$, the balance is disturbed since the signal coming from the excitatory synapse is stronger than the signal delayed for a period Δt coming from the inhibitory synapse. The interneuron $N_1(x,y)$ begins to generate impulses (spikes). If the current of the receptor $N_r(x,y)$ decreases, *i.e.* $I_{nr}(x,y,t)<I_{nr}(x,y,t-\Delta t)$, the interneuron $N_1(x,y)$ does not react, but interneuron $N_2(x,y)$ starts to generate impulses, since the delayed signal (for a period Δt), flowing from the excitatory synapse is stronger than the signal from the inhibiting synapse. In other words, the neural network begins to react to the variation in pixel brightness that can be caused by the passage of objects moving over a static background.

The result of the work/simulation of the detector based on the impulse network model is given in **Figure 2**. The output layer N_{out} of the neural network has the same dimensions as the input layer and the hidden layer. Each neuron of a given layer $N_{out}(x,y)$ corresponds to each pixel (x,y) of the output frame of the video image. The interneurons $N_1(x,y)$ and $N_2(x,y)$ are connected by excitable synapses without delay to the output neuron $N_{out}(x,y)$. It produces signals only when it receives impulses from the interneuron $N_1(x,y)$ or $N_2(x,y)$, otherwise, it is at rest. The magnitude of the grayscale of each pixel (x,y) of the output video frame is proportional to the frequency of pulse generation by the output neuron $N_{out}(x,y)$ and has a value of 0 (black) if the output neuron $N_{out}(x,y)$ does not generate any signals over a certain time period *T*. Otherwise, the brightness of the pixel (x,y) will be above 0 (**Figure 2**).



(a) Input Image (b) Output Image

Figure 2. Selecting moving objects by impulse neural network.

3. Recurrent Neural Network Model

Recurrent neural networks are a class of neural networks with feedback between different layers of neurons [14]. The special feature of such networks is the transmission of signals from the output (hidden) layer to the input layer. The particular interests are the multi-layer recurrence networks, which are the development of single-directional perceptron networks through the addition of feedback links. On each layer of such a network, there is a unit delay element that allows the input flow to be considered as unidirectional. Such a recurrent neural network functions as a single-directional perceptron network. The learning algorithm of such a network is more complex due to the dependence of signals at time *t* on their value in previous moments [15].

There are different models of recurrent neurons: Recurrent Multilayer Perceptron, Real-Time Recurrent Network, etc. The most interesting is the RTRN network (Real-Time Recurrent Network) proposed by R. Williams and D. Zipser and intended for real-time signal processing [15]. The RTRN network is a special case of the Elman network. The structure of the RTRN network is shown in **Figure 3**.

The network contains N input nodes, K hidden neurons, and K corresponding nodes of the context layer. Of the K hidden neurons, only M forms the output of the network. All output neuron signals are connected as neural network inputs through the $z - 1$ delay elements. Denote the weighted sum of the i-th neuron of the hidden layer u_i, and the output of this neuron is y_i. The vector $x(t)$ and the deflected (delayed) vector $y(t - 1)$ form an extended activation vector $x(t)$ that excites the network neurons:

$$x(t) = \left[1, x_1(t), x_2(t), \cdots, x_N(t), y_1(t-1), \cdots, y_K(t-1)\right]^{\mathrm{T}} \tag{1}$$

After describing the input vector of the network at t, you can determine the state of all neurons according to the dependencies:

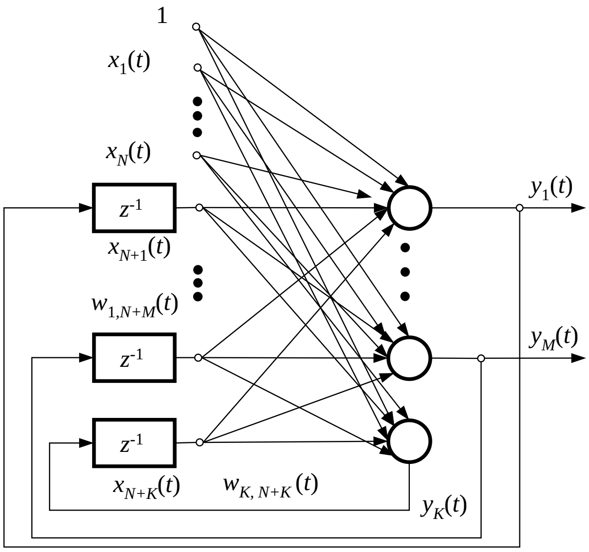


Figure 3. RTRN network structure.

$$u_i = \sum\nolimits_{k=0}^{N+K} w_{ik} x_k \tag{2}$$

$$y_i(t) = f(u_i(t)) \tag{3}$$

Furthermore $f()$ denotes a continuous neuron activation function (sigmoidal). We also determined $w_i = [w_{i0}, \cdots, w_{i,N+K}]$ as a vector of weights of the i-th neuron and $W = [w_1, \cdots, w_K]$ - the matrix of weights of neurons. The William-Zipser learning algorithm is used to train the RTRN network, namely to minimize the following criterion:

$$J(t) = \frac{1}{2} \sum\nolimits_{i=1}^{K} \left[d_i(t) - f(x(t) w_i(t)) \right]^2 , \tag{4}$$

where $\varepsilon_i(t)$ is defined by expression (5)

$$\varepsilon_i(t) = \begin{cases} d_i(t) - y_i(t) \mid i = 1, \cdots, M; \\ 0 \mid i > M. \end{cases} \tag{5}$$

After minimizing the weight criterion (4) $w_{\alpha\beta}$ (matrix weight W) can be defined as follows:

$$w_{\alpha\beta}(t+1) = w_{\alpha\beta}(t) - \eta \sum\nolimits_{i=1}^{K} \varepsilon_i(t) \frac{\mathrm{d}y_i(t)}{\mathrm{d}w_{\alpha\beta}} \tag{6}$$

The proposed model of a recurrent neural network can be used for detecting moving objects in a video image. The video image is a sequence of frames, and each frame is a set of pixels defined by the corresponding parameters. We will assume that the input of the neural network receives a sequence of frames, and the output is expected to receive a processed sequence of frames on which the moving object is selected from the general image.

The set of pixels of the input frame of the video images will correspond to the input flow of signals for the input nodes of the neural network. Thus, take the number of input nodes of the N network equal to the number of pixels per frame. Since the output of the neural network is expected to receive a processed frame, the number of neurons in the hidden layer $K = N$ is determined. The output of the hidden layer of neurons is the output of the neural network, and also the input signals for the delay elements $z - 1$ (contextual layer).

Each hidden layer neuron receives a signal from each input node and each context layer delay element. Thus, each neuron of the hidden layer at time t has information about pixels of the current frame and the processed (previous) frame at time $t - 1$. Processing pixel information of the current and previous (processed) frames hidden neuron frame emits a signal.

At the beginning of time t_0, when the delay elements do not have information about the processed frame, the first frame of the video image enters the network. At the same time, a learning process (the installation of neural network weights) is taking place, which is aimed at putting the network in an equilibrium position. Equilibrium is achieved when input and processed frames are identical, *i.e.* there is no movement in the video image.

When motion appears in a video image (there is a difference in two sequential frames), the configured neural network will go out of equilibrium, which will be displayed on the processed frame. In the next step, the neural network will attempt to enter a state of equilibrium based on the processed frame. This process will continue as long as movement is present in the video.

One of the variants of the development (modernization) of the proposed model of a recurrent neural network for a motion detector is the reduction of redundancy in the connections between the hidden layer of neurons and the input layer of signals, as well as the context layer. This redundancy in communications makes it difficult to train the network and process high-resolution video images. It is proposed to reduce the number of connections and to bind each neuron of the hidden layer to the corresponding input element of the input layer (pixel) and elements adjacent to this pixel. The adjacent pixels can be 4 or 8 adjacent pixels with 4 or 8 connections (**Figure 4**).

The proposed option to reduce redundancy would decrease the learning time of the network. When the number of connections is reduced, it is necessary to carry out a smaller number of calculation operations, which will decrease the demands on the computational resources of the motion detector. The processing time of each frame and the time of the sensor reaction to movement will also be reduced, which will make it possible to use the proposed technique for high-frequency video images.

4. Structure and Operation Algorithm of the Automated System

The design and development of the automated system used the principle of modularity, *i.e.* the system consists of several software modules that are relatively independent and easily integrated with each other. This allows changes to be made to one software module without affecting the other.

The structure of the automated system is presented in **Figure 5**.

The first module is a graphical interface for setting the system parameters and displaying the results. This module was created on the Microsoft NET Core 3.1 platform using Windows Form technology for user interface and interaction with data models. The user selects and changes the source of the video sequence, and specifies the format of the data output (output to the screen or write to the file). As a result, this module forms a set of parameters that are used in other modules of the software complex.

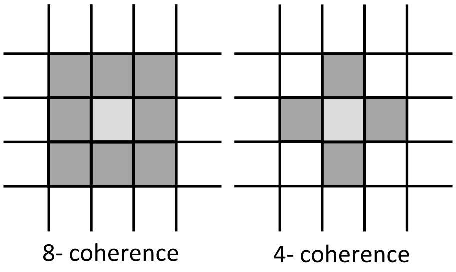


Figure 4. Adjacent pixels 8 and 4 connected.

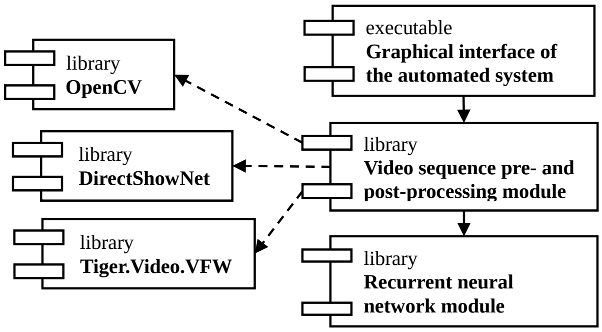

library
OpenCV
library
DirectShowNet
library
Tiger.Video.VFW
executable
Graphical interface of the automated system
library
Video sequence pre- and post-processing module
library
Recurrent neural network module

Figure 5. Structure of the automated system for the selection of moving objects in a video sequence based on a recurrent neural network.

The second module is a module for pre- and post-processing the video sequence. This module interacts closely with the other modules of the system. The pre-processing of a video sequence consists of obtaining a video image from a source and in preparing for the subsequent transmission to the module of processing a recurrent neural network. Postwork consists of displaying processed data to a user screen or writing to a file.

The third module is a module for processing a recurrence neural network. This module is an implementation of a recurrence neural network. The input of the given network receives a sequence of frames of the original video. The result of the work of the module is processed frames of the initial video with an image of moving objects.

As can be seen from **Figure 5**, the system uses the Emgu CV (OpenCV) package libraries extensively [16]. This is due to the fact that this computer vision package provides a fairly wide range of well-developed and well-organized image processing classes and methods. It also uses the DirectShowNet library, which allows working with graphics input devices and the Tiger.Video.VFW library to work with video files.

5. Recurrent Neural Network Module

The recurrent neural network module is one of the components of the automated system being developed. Its task is to process the incoming video sequence to select moving objects. Processing is performed by a recurrent neural network [14] [16].

The result of the work/simulation of the detector based on the recurrent network model is given in **Figure 6** and **Figure 8**. Consider the work of this module by example. Let the module pre- and post-work produce a sequence of frames in greyscale. Frames examples are provided in **Figure 6**.

The resultant images are then delivered to the input of the recursive neural network.

The recurrent neural network processes each incoming frame according to the algorithm presented in **Figure 7**.

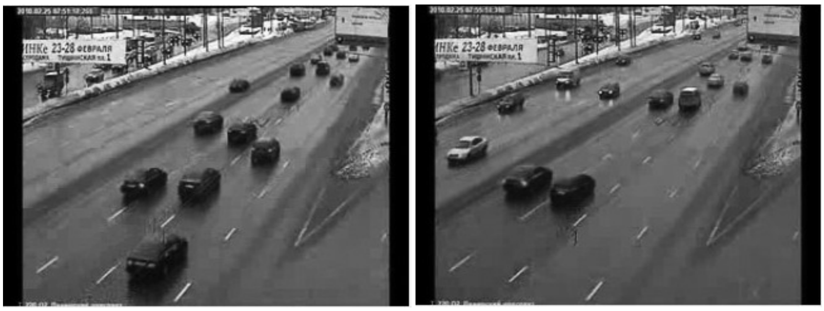


Figure 6. Examples of incoming video sequence frames.

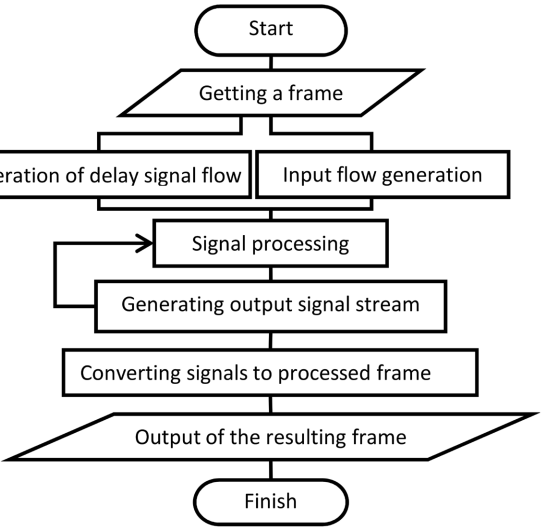

Start
Getting a frame
Generation of delay signal flow
Input flow generation
Signal processing
Generating output signal stream
Converting signals to processed frame
Output of the resulting frame
Finish

Figure 7. Operation algorithm of the recurrent neural network module.

After the frame is received, a signal flow is formed on the basis of information about the brightness of pixels of the input frame.

The processing of the input flow of signals takes place after a delay signal is received indicating that all signals are received and can be processed. Input flow signals and output stream signals received during the processing of the previous sequence frame are accepted for processing. At an initial time, the output flow is equal to the input stream of the signals.

During processing, the recurrent neural network forms an output stream of signals, which is sent to the input of the network and transformed into an output frame. The converted frame is transferred to the pre-work and post-work module for further use.

The input frame is received by a module for pre-processing and post-processing the video sequence, and the module receives an output frame after processing by a recurring neural network.

The input and output frame is an object of type System.Drawing.Bitmap. All modules are written in C# programming language in Microsoft Visual Studio Enterprise 2019.

The result of the module is an output frame sequence on which only the selected moving objects are located. An example of the results of the module is presented in Figure 8.

6. Video Sequence Pre- and Post-Processing Module

This module is intended for obtaining an input video sequence from different sources (camera, file). The DirectShow Multimedia Framework and Application Programming Interface (Figure 9) functions are used to obtain a video sequence from a capture device (camera).

DirectShow allows Windows applications to manage a wide range of audio/video input/output devices, including DV and web cameras, DVD devices, TV tuners, etc.

Capture filters are used to obtain a video sequence from the capture devices. Capture Filters are designed to inject multimedia data into the program stream from various physical devices. The role of the device can be both various video devices (portable video cameras, webcams, TV-tuners) and audio devices (microphone, modem line) as well as data can be obtained from files (AVI, MPEG, MP3).

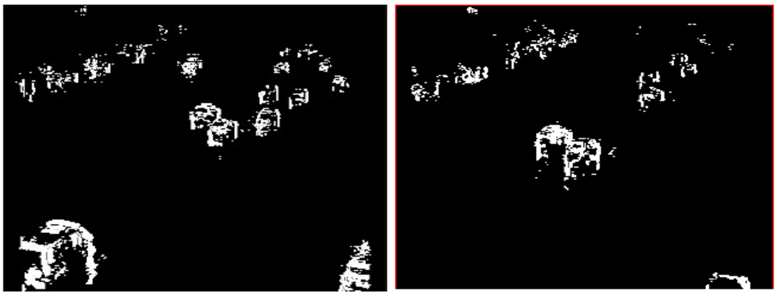


Figure 8. Results of the automated system operation.

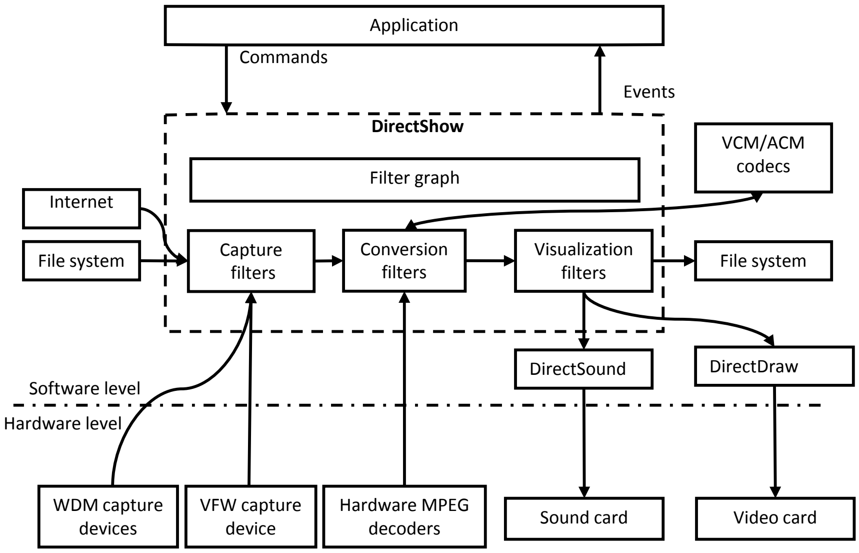


Figure 9. DirectShow structure scheme. MPEG: Motion Pictures Expert Group; VFW: Video for Windows; WDM: Wavelength-Division Multiplexing; VCM: Variable Coding and Modulation; ACM: Adaptive Coding and Modulation;

DirectShow allows the simultaneous use of multiple capture filters, for example: for simultaneous capture of video from the webcam and sound from the microphone. The number of simultaneously used capture filters is limited only by the power of the computer used.

If a capture device is selected as a video sequence source, this information is transmitted via API functions to the DirectShow package filter core, so this package must be preinstalled on the computer where the developed system is used.

If an existing file is selected as a video sequence source, this information is transmitted via API functions to the Tiger.Video.VFW package. The Tiger.Video.VFW package is intended for reading and writing AVI files. The package has a set of features implemented through low-level access to the Windows file system. This makes it possible to quickly work with video files as a source of a video sequence for further processing by the system.

7. Conclusions

The paper describes promising research directions in the field of artificial intelligence motion detection using neural networks.

The models of impulse and recurrence neural networks for the detection and selection of moving objects in a video image are proposed. The novelty of the proposed model of a recurrent neural network for a motion detector is the reduction of redundancy in the connections between the hidden layer of neurons and the input layer of signals, as well as the context layer. This excess in interconnections makes it complicated to train the neural network and process high-resolution video images. It is suggested to decrease the number of interconnections and to bundle each neuron of the hidden layer to the corresponding input element of the input layer (pixel) and elements neighbouring to this pixel. The bordering pixels can be 4 or 8 adjacent pixels with 4 or 8 connections (**Figure 4**).

The proposed modification of the recurrent neural network model would decrease the learning time of the network. If the number of connections is reduced, it leads to a decrease in the number of calculation operations. As the result, it mitigates the demands on the computational resources of the motion detector. The processing time of each frame and the time of the detector reaction to motion will also be decreased, which contributes to the application of the proposed detector and methodology improvement of automation of the detection and selection of moving objects for high-frequency video images.

The suggested approach for detecting and separating moving objects is an attempt to simulate the ability of the human eye to quickly isolate moving objects and surpass existing deterministic methods in terms of speed of selection of moving objects and economy computing resources. And the motion detector being developed on the basis of this approach, as a software module, will be able to find a suitable application in the field of digital video processing. It is intended to use this detector in automated traffic management systems [17] [18] [19] as an alternative to existing detectors, even taking into account the possible improvement of the latter by the use of parallel computations for simultaneous processing of segments of the video image and the selection of moving objects within each of them [20].

The disadvantage of the proposed approach for the detection of motion in video is that as the image resolution increases, the number of neurons in the network increases dramatically, resulting in noticeable delays in the processing of the signal flow of neurons.

Impulse neural network elements can be implemented in hardware [21], or software with the use of modern parallel computing technologies based on graphics processors. This can significantly accelerate the selection of moving objects in a video image.

As a result of the work carried out, an automated system for separating moving objects on a video sequence based on a recurrent neural network has been created and is ready for practical use. The automated system consists of three independent modules (a graphical interface of the automated system, a recurrent neural network, and a pre- and post-processing video sequence). The structure of the automated system follows the principle of modularity, which allows for the development of modules independently from each other and allows for future changes in any of the modules without affecting the other.

The paper explored the possibility of integrating programs written in object-oriented programming languages (in particular C#) with the Emgu CV image processing package and the multimedia framework and programming interface functions of DirectShow, selected the most efficient Emgu CV package tools for image processing.

The developed automated system of moving objects selection uses the graphical core of video stream processing based on the package Emgu CV, namely the functions of transforming images into semitone (in grayscale shades). However, the disadvantage of this approach is the need to have Emgu CV package libraries on the computer where the proposed automated system will be used [16].

The proposed automated system is a software product that makes it possible to separate moving objects in a video stream based on recurring neural networks. This software provides the opportunity to upgrade in order to improve the detection of moving objects by modifying the module of the recurrence neural network.

The proposed approach and software for the detection of moving objects in video images using neural networks can be incorporated into more sophisticated specialized computer-aided video surveillance systems, IoT, IoV, etc.

Future work directions may include extending the proposed approach and motion detector for global traffic management, including the optimization of next tasks: providing city services with up-to-date information about the current state of the transport infrastructure by creating a measurement system based on IoT, Industrial Internet of Things (IIoT), and IoV technologies; adaptive control systems implementation for traffic lights, bollards, and other road infrastructure devices based on information about traffic congestion in order to redirect and synchronize traffic flows, prevent the traffic jams formation; intelligent systems implementation for: forecasting the transport situation in the city, costs optimization of the transport infrastructure development by predicting possible results of decision-making; the use of intelligent systems for modeling the transport situation in the design of roads, intersections, crossings, traffic lights, and other tasks; monitoring systems implementation in order to detect the location and current state of urban transport, creation of "smart" stops; informing citizens systems creation about the current roads congestion and the condition of parking lots (free places availability), including mobile applications; transport flows synchronization of various types (buses, subways, trams, etc.), to reduce the time that passengers spend on transfers; introduction of unmanned road transport and railway systems, including the metro; automation of management processes and accounting of the work of contractors engaged in snow removal on roads, crossings, and stops, adaptive management of cleaning taking into account the current weather and transport situation; the use of unmanned aerial vehicles/drones for traffic monitoring, the snow removal quality on roads and crossings, the quality of road surface, etc.; the transition to the use of wireless technologies (communications, power supply).

Acknowledgements

The authors gratefully acknowledge the facilities and support provided by the Director and all other staff members of the School of Information Engineering, Development and Popularization Center for Industry-Academy-Research Projects, Xi'an Eurasia University. The research was partly supported by the Xi'an Eurasia University within the framework of the Long-term Innovation Project OYGJS-2021002 Application Research and Development of a Distributed Photovoltaic Power Generation System in order to develop renewable energy training equipment and its intelligent control system for video cameras' power supply and to integrate them with IoT.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] U.S. Department of Transportation (n.d.) About ITS/RITA. http://www.its.dot.gov/its_program/about_its.htm

[2] Japan Society of Civil Engineers (n.d.) "ITS Introduction Guide": ACECC TC-16 (ITS-Based Solutions for Urban Traffic Problems in Asia). https://www.jsce-int.org/search/node/its%20introduction%20guide.

[3] Hilmani, A., Maizate A. and Hassouni, L. (2020) Automated Real-Time Intelligent Traffic Control System for Smart Cities Using Wireless Sensor Networks. *Wireless*

Communications and Mobile Computing, **2020**, Article ID: 8841893, 28 p. https://doi.org/10.1155/2020/8841893

[4] SoulPage (2020, January 8) Smart Traffic Monitoring and Management System. https://soulpageit.com/smart-traffic-monitoring-and-management-system/

[5] Zaatouri, K., Jeridi, M.H. and Ezzedine, T. (2018) Adaptive Traffic Light Control System Based on WSN: Algorithm Optimization and Hardware Design. 2018 26*th International Conference on Software*, *Telecommunications and Computer Networks* (*SoftCOM*), Split, Croatia, 13-15 September 2018, 1-6. https://doi.org/10.23919/SOFTCOM.2018.8555802

[6] Automatic Traffic Management System. https://www.parknsecure.com/automatic-traffic-management-system

[7] Lewandowski, M., Płaczek, B., Bernas, M. and Szymała, P. (2018) Road Traffic Monitoring System Based on Mobile Devices and Bluetooth Low Energy Beacons. *Wireless Communications and Mobile Computing*, **2018**, Article ID: 3251598, 12 p. https://doi.org/10.1155/2018/3251598

[8] Chakrabarti, I., Batta, K.N.S. and Chatterjee, S.K. (2015) Motion Estimation for Video Coding. Springer Int'l Publishing, Cham. https://doi.org/10.1007/978-3-319-14376-7

[9] Beyerer, J., Puente Leon, F. and Frese, C. (2016) Machine Vision—Automated Visual Inspection: Theory, Practice, and Applications. Springer-Verlag, Berlin, New York. https://doi.org/10.1007/978-3-662-47794-6

[10] Menter, Z., Tee, W. and Dave, R. (2021) A Study of Machine Learning Based Pattern Recognition in IoT Devices. *Proceedings of the* 3*rd International Conference on Communication and Computational Technologies*, *Algorithms for Intelligent Systems*, Jaipur, 27-28 February 2021, 669-689.

[11] Nielsen, M.A. (2015) Neural Networks and Deep Learning. Determination Press. http://neuralnetworksanddeeplearning.com/index.html

[12] Wu, Q. (2008) Motion Detection Using Spiking Neural Network Model. *Proceedings of the* 4th *International Conference on Intelligent Computing* (ICIC'08), Shanghai, 15-18 September 2008, 76-83. https://doi.org/10.1007/978-3-540-85984-0_10

[13] Gerstner, W. and Kistler, W.M. (2002) Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511815706

[14] Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S. and Razaque, A. (2020) Deep Recurrent Neural Network for IoT Intrusion Detection System. *Simulation Modelling Practice and Theory*, **101**, Article ID: 102031. https://doi.org/10.1016/j.simpat.2019.102031

[15] LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep Learning. *Nature*, **521**, 436-444. https://doi.org/10.1038/nature14539

[16] Gonzalez, R.C. and Woods, R.E. (2018) Digital Image Processing. 4th Edition, Pearson Education, New York, 1022 p.

[17] Sadhukhan, P. and Gazi, F. (2018) An IoT Based Intelligent Traffic Congestion Control System for Road Crossings. 2018 *International Conference on Communication, Computing and Internet of Things* (IC3IoT), Chennai, 15-17 February 2018, 371-375. https://doi.org/10.1109/IC3IoT.2018.8668131

[18] Jha, S., Seo, C., Yang, E. and Prasad Joshi, G. (2021) Real-Time Object Detection and Tracking System for Video Surveillance System. *Multimedia Tools and Applications*, **80**, 3981-3996. https://doi.org/10.1007/s11042-020-09749-x

[19] Kim, K., Lee, J., Lim, H. and Han, Y. (2021) Deep RNN-Based Network Traffic Classification Scheme in Edge Computing System. *Computer Science and Information Systems*, **19**, 165-184. https://doi.org/10.2298/CSIS200424038K

[20] Fujita, K., Okuno, S. and Kashimori, Y. (2018) Evaluation of the Computational Efficacy in GPU-Accelerated Simulations of Spiking Neurons. *Computing*, **100**, 907-926. https://doi.org/10.1007/s00607-018-0590-0

[21] Dinu, A., Cirstea, M.N. and Cirstea, S.E. (2010) Direct Neural-Network Hardware-Implementation Algorithm. *IEEE Transactions on Industrial Electronics*, **57**, 1845-1848. https://doi.org/10.1109/TIE.2009.2033097