# An Application Development for Recognizing of View in Order to Control the Mouse Pointer

Petro Zdebskyi[0000-0002-0478-2308][1], Victoria Vysotska[0000-0001-6417-3689][2],
Roman Peleshchak[0000-0002-0536-3252][3], Ivan Peleshchak[0000-0002-7481-8628][4],
Andriy Demchuk[0000-0001-5710-9347][5], Maksym Krylyshyn[0000-0002-2964-9546][6]

[1-2, 4-6]Lviv Polytechnic National University, Lviv, Ukraine
[3]Ivan Franko Drohobych State Pedagogical University, Drohobych, Ukraine
petrozd@gmail.com[1], victoria.a.vysotska@lpnu.ua[2],
rpeleshchak@ukr.net[3], peleshchakivan@gmail.com[4],
andriydemchuk@gmail.com[5], maksum99@gmail.com[6]

**Abstract.** The purpose of this article is to develop an application for recognizing the user's point of view in order to control the mouse pointer. In the course of the task, an analysis of the subject area and technologies for implementation of the application was conducted. The algorithms of machine learning for solving the problem were considered. The input data of system are 50 coordinates placed on the face, which include the contours of the face, eyebrows, eyes, nose and coordinates of pupils. Finding the required coordinates occur without use of special devices, but only with webcam used for recognition. The application is implemented in the form of two modules, one of which is responsible for training system for the recognition of the view; the other one is responsible for controlling the mouse cursor with a view. The product can be operated on any operational system: the main requirement is the presence of an interpreter for the Python programming language, which can be downloaded for free from the official site.

**Keywords.** Control the Mouse Pointer, User's Point, Machine Learning, Neural Network, Python, eye tracking, mouse pointer, human eye, third party device, artificial neural network, data analyzation, real time, user perspective, machine learning algorithm, sequence action response, web camera, trademark office, mouse cursor, system component, functional requirement, eye recognition, image recognition, image processing, convolutional neural network, component called viewpointdata, data stream mining, demand knowledge, facial feature, weight coefficient, open source

## 1 Introduction

The purpose of article is to develop an application for controlling mouse cursor with view. The application will not use third-party devices, webcam will be used for recognition. Eye tracking has many possible ways to be used. It can be applied as a new way of communication with computer, as well as providing it will be especially useful for people with disabilities. The recognition of eyes was investigated to determine the behavior of users on the Internet [1]. Understanding what people are paying attention to when browsing a web page can be important for improving advertising, developing a better web interface, optimizing websites to improve interoperability and ease the use of computer as a whole. The easiest way to track view is human's interaction with a computer. The desire is to increase the ability to interact with the computer. Usage of view can accelerate and make the interaction more intuitive and natural, compared with a computer mouse. It can also help a limited user with limited access to control other interaction devices. Existing systems that detect and track view often use infrared light, stereo cameras, and often require special equipment. This results in cumbersome third-party devices that worsen usability of the system. In addition, infrared radiation may not be completely safe. Infrared radiation affects the human eye, because it can not completely reflect infrared radiation from the retina through the nature of human eye. [2, 4, 5]. Transferring video in real time with a limited bandwidth is desirable, but difficult to get a high quality image. In order to improve the perception of image quality, you can compress different parts of the image more or less than others. For example, transferring the regions where user is looking with a higher resolution and compress those that are outside of the user's perspective. This can be implemented using information from the view recognition system [3]. Therefore, it was decided to implement an application for recognizing the user's point of view to control the mouse pointer. The application must only use a webcam and require no additional devices, making the system more user-friendly and accessible.

## 2 Use of Eye Tracking to Control Mouse Pointer

### 2.1 Description of the problem of tracking the eye

View - this is the direction in which person is looking. Gaze recognition means detecting and tracking this direction. The task of this work is to develop an algorithm that carries out a view tracking using a webcam. In other words, the algorithm of the sequence of images, on which the face is a glance. The goal is to create a real-time functional system. In the process of viewing something, such as an image or icons on a computer monitor, the eyes point in a direction that indicates what they are focused on. Because of physiological limitations, people can observe only a small area around the place where their eyes are concentrated. This property of the human eye can be used to create a system for recognizing the look. Eye tracking can be used both for monitoring and responding based on the user's perspective [4].

During normal use of a computer, user often makes head movements in all directions. Thus, the point at which the user focuses depends not only on the direction of the user's eyes, but also on the inclining of his face, the distance between face and monitor, as well as many other parameters [4]. Due to the fact that there are many factors that affect the recognition of the look, as well as the presence of noise in images, usage of artificial neural networks for this task will be most suitable.

There are certain limitations regarding the speed of the system, because for easy control of the mouse cursor, the minimum delay is required. The task requires huge amount of computing power, since image processing is a resource-intensive process.

Although in the future perspective the speed may increase, as the computing power of equipment increases, so in the future, such systems may become widespread use.

The need in additional equipment is still a problem for ordinary users, since they do not want to buy additional equipment. In a perfect scenario, the user will only need to install the software and use the webcam to interact with computer. Therefore, it is obvious that tracking views needs further development.

## 2.2 Eye tracking usage of user

This section provides some examples of applications and research done using similar systems to show benefits of eye tracking. Eye tracking has a large number of possible applications for the community. There are many obvious benefits, such as helping the disabled, or a new form of human-computer interaction. In his 1996 research, Tok was trying to track the position of eyelids while he was in a vehicle. This system could determine when a driver becomes drowsy and thereby reduce the number of accidents caused by driver fatigue and potentially save lives. [5].

**Eye tracking in test usage.** This area has a key value to the study of websites and applications. Not only for marketing purposes, but also for studying user interaction with the application. Due to this we can check:

1. Trajectory of the eyeball movement and time of focusing on a particular object.
2. In which sequence objects attract the attention of respondents.
3. Distracting elements.
4. Elements that are often considered.
5. Correctness of displaying objects on a website, or their sequence is the same as the one in which they are viewed.

**Use of eye tracking to study ergonomics.** From customer's point of view, ergonomics is a key when it comes to choosing a product. Here it is measured how simple and otherwise understandable the product is in use. Aesthetic research on ergonomics is used in testing of mobile phones, audio and video devices, home appliances. Due to this research we can check:

1. Elements accuracy.
6. What distracted respondents.
7. Which elements were not found and why.
8. Whether their use and behavior were predictable.
9. Were the elements in the places where the respondent expected them?

**Use of eye tracking in psychology.** Eye tracking research is increasingly providing new opportunities in many areas of psychology. This is due to technological development and increased availability of equipment, as well as increased analytical capabilities. Eye tracking can be used in the following areas:

1. Cognitive psychology and cognitive science are the study of perception of visual stimuli, the relationship between the types of information and how it is perceived, human behavior in vehicles while they move and the interaction between man and computer.

10. Psychology of development is research on the development of eye coordination, attention distribution skills, relationship between the control system of motion and understanding of text and the study of autism.
11. Experimental psychology is study of blind perception and recognition, visual perception of plots and images and differences in visual perception of healthy people and people with disorders of the nervous system.
12. Psycholinguistics is the study of relationship between visual perception and ability to read.
13. Neuropsychology and disturbance of mental health is research of perception of images by healthy and sick people, studying of mechanisms of observation.
14. Ophthalmology is diagnosis of musculoskeletal movement of the eyeball and evaluation of the effect of surgical treatment.

**Use of eye tracking in advertisements.** First seconds of customers contact with advertisement are important. Eye tracking shows which elements have attracted attention and which remained invisible. Therefore, a complete picture of effectiveness of each of them is made. Using eye tracking, you can learn the following types of promotional products:

1. Press advertising.
2. Video clips.
3. Internet advertising.
4. Outdoor advertisements
5. Electronic mailing.

**Use of eye tracking while observing shelves in stores.** Eye tracking is used in modern commerce, where modern methods do not provide satisfying results. This applies to both macro level and micro level navigation. The macro level of navigation refers to the location of product categories throughout the store. Research of eye tracking is directed to allow customers to move freely around the store and at the same time have all the information elements within their reach. Navigation at the micro level is to create an optimal location of individual products on the shelf. This is aimed to adapt the mall to make the location of products as effective as possible.

**Research of eye tracking, directed for testing of public information systems.** Thanks to them, our environment becomes more and more understandable, that is, we can easily find our way to the metro station, cash desk, or a certain store in mall. Eye tracking is also used when evacuation and warning signs need to be evaluated [7].

## 2.3 Review of analogues

There are a lot of eye tracking systems. They give us these features: control with mouse, data receiving of user eyes and data analyzation. System analogs:

1. xLabs Gaze Tracking is created as extension for web-browser Google Chrome. This application is a result of 2-year research and development by 4 cofounders.

   *Advantages*:

   (a) Easy to install.
   (b) Easy to use.

(c) Can be used in different platforms.

*Disadvantages*:

(a) Works only with web cameras (lower accuracy).
(b) There are no easy ways of getting data.
(c) No function of data analyzation.
(d) No support.

15. GazePointer is program that is easy to install and works on Windows, what makes it one of the "easy to get" programs in this list.

*Advantages*:

(e) Easy to install.

*Disadvantages*:

(a) Works only with web cameras (lower accuracy).
(b) No support.
(c) No function of data analyzation.

16. Ogama – This software is with opencode, developed in Free Univsersity of Berlin.

*Advantages*:

(d) Provides main parameters of data analyzation.

*Disadvantages*:

(a) No support.
(b) No updates for last 2 years.

17. OpenEyes – This software with opencode allows to track eyes with usage of infrared and visible lighting, using Matlab.

*Advantages*:

(c) Can be used with web-cameras and infrared eyes trackers.

*Disadvantages*:

(a) Need to install Matlab (commercial program) and knowledge of working with Matlab.
(b) No support.
(c) No function of data analyzation.

18. PyGaze – This software works on Python and was published in 2014 by 3 researchers (Oxford university, Aix-Marseille University and Utrecht university.

*Advantages*:

(d) Data analyzation (Need knowledge of Python).

*Disadvantages*:

(a) Needs strong knowledge of Python.
(b) Limited support.

19. OpenGazer is used in order to increase accessibility of using computer and was supported by Samsung and Charity fund Gatsby.

*Advantages*:

(c) Potentially compatible with OS Apple (demands knowledge of programming).

*Disadvantages*:

(a) Needs Linux (demands knowledge of using Linux).
(b) No support.

20. TurkerGaze is a program developed by researchers in Princeton. Systems works on Linux and depends from couple other Linux programs for its functioning.

Advantages:

(c) Provides main functions of data analyzation.
*Disadvantages*:
(a) Is better to be used with headrest.
(b) Needs Linux (demands knowledge of using Linux).

21. GazeParser – This software consists of 2 components: GazeParser (data convertation and analyzation) and SimpleGazeTracker (used for recording eyes) with Python usage.
    *Advantages*:
    (c) Can analyze data (Demands knowledge of Python)
    *Disadvantages*:
    (a) Needs movement tracking camera or camera of machine vision.
    (b) Needs headrest to limit head movement.
    (c) Needs strong knowledge of Python
22. ITU Gaze Tracker – Program created by Gaze in Copenhagen University, ITU Gaze Tracked is a platform with open code, used for increasing technology access.
    *Advantages*:
    (d) Easy to install
    Disadvantages:
    (a) Needs to create your own infrared eyes tracker (if web-camera is not used)
    (b) No support

During the analysis of the software for recognition of user's perspective, the basic principles of the work of such applications were considered. Analogues were analyzed, namely, the system of recognition of the view, their advantages and disadvantages were determined. There are a large number of systems that solve problems, but they have certain limitations, such as knowledge of additional technologies, or the use of third-party devices. Another disadvantage is that applications analogs were developed long time ago, and many of them are no longer supported.

## 3 Problem Formulation

In this bachelor's work, the task was to develop a system for recognizing the user's point of view to control mouse pointer. This system allows users to control mouse pointer with a glance. The main priority of the application is that it will only use a regular webcam without use of third-party devices. Accordingly, the following tasks were set:

1. Analysis of applications-analogues, that have a similar functional.
23. Analysis of technologies and environments for the development of image recognition systems.
24. Explore and write an image processing algorithm that should be resistant to noises in the image and should work with low-quality images.
25. According to the set of tasks, program the mouse cursor control system.
26. Testing the finished system.

The task can be divided in two parts: image analysis and machine learning. An image analysis involves identifying and isolating features that are sufficient to determine the look. Machine learning is creation and training of the neural network using these features as input and determines location of the mouse pointer using the output, or the result of calculations of the neural network. This paper examines the creation of a mouse eye view control system that uses cheap hardware, but is quite effective to achieve the desired accuracy. A large number of computers users already have webcams and they are relatively cheap. The hardware that will be needed for this system will be a simple webcam. The ultimate goal of this project is to find out if you can create a view tracking system using a regular webcam. The application will use open-source OpenCV and Dlib computer vision libraries to recognize the image.

## 4 Specification of Software Product Requirements

The purpose of this project is to create a system for recognizing the user's perspective to control the mouse pointer without the use of third-party devices, only with the use of a webcam. Products-analogues: xLabs Gaze Tracking, GazePointer, Ogama, OpenEyes, PyGaze, OpenGazer, TurkerGaze, GazeParser/SimpleGazeTracker, ITU Gaze Tracker. The following key characteristics can be distinguished:

1. Low quality image recognition
2. Noise processing in images
3. High speed, which is sufficient for real-time operation
4. High recognition accuracy, which is sufficient to recognize the look.

The system will have only one class: application users. The app is intended for people who need to use the look for mouse cursor, as well as for people with disabilities. System requirements is Interpreter for Python 3.0

**System characteristics**

*System training*

1. Description and priority. Priority is high. Ability to train the eye tracking system.
   (a) Sequence action-response
   (b) User opens application ViewpointData.
   (c) User executes a “train” command in the console.
   (d) System checks whether the training was successful.
   (e) If training was unsuccessful, the system reports about it.
   (f) System reports result of training.
2. Functional requirements
   (a) REQ 1. Informative notice that the training process begins.
   (b) REQ 2. Give the opportunity to cancel the workout.
   (c) REQ 3. Provide the opportunity to rehearse the system.

*Data storage*

1. Description and priority. Priority is average. Ability to save data about coordinates of facial features.
2. Sequence action-response
   (a) User opens ViewpointData and executes the “save” command in the console.

   (b) If the save failed, the system notifies about it.
   (c) System reports the result of the save.
3. Functional requirements
   (a) REQ 1. Informative notice that the process of saving begins.
   (b) REQ 2. Give the opportunity to cancel saving.

*Revision of visual representation of the coordinates of characteristics of face*

1. Description and priority. Priority is low. Ability to visually represent the coordinates of facial features.
2. Sequence action-response
   (a) User opens ViewpointData and executes the "show" command in console.
   (b) System displays a window with coordinates of characteristics of face.
3. Functional requirements: REQ 1. Provide the ability to interrupt display of coordinates at any time.

*Tracking the user's point of view*

1. Description and priority. Priority is high. Ability to track user's point of view to control the mouse pointer.
2. Sequence action-response
   (a) User opens application Viewpoint.
   (b) Mouse cursor moves towards the user's perspective.
   (c) Functional requirements: REQ 1. Recognition accuracy should be high enough to be able to track the look.

**Requirements of external interfaces**

1. *User interfaces.* User can interact with the system using a personal computer that has a *webcam*.
2. *Hardware interfaces.* Current system will not use any hardware interfaces.
3. *Software Interfaces:* OpenCV, Dlib and Keras.

**Other non-functional requirements.** The system must react quickly to change user's pupils' positions in order to work in real time (all information should be processed within a few milliseconds). Personal data is confidential information and can not be transferred to third parties. This can be done by making this system with open source. Software product attributes of quality:

1. Comfort of usage.
2. Reliability.
3. Convenience of the escort.

Permissions required for the application to work - Access to webcam. In this section, the task setting and the justification of the selected technologies in order to solve the task were made. A specification of requirements was made. The choice of ways and means for developing a user's view control application for mouse control was described, and the advantages of the chosen direction for solving tasks associated with the development of the application were described.

# 5 Eye-Tracking Application Design

The goal of the project is to create a mouse tracking system using simple techniques, webcam, OpenCV and Dlib libraries. Since the logical design of most view tracking systems is quite standard, the overall design of this system is very similar to existing systems. However, the tools used in these systems have a high degree of variation. This section will start with the details of some design decisions, and then the motivations of using a specific webcam and programming language, and finally move on to the logical and actual design specifics.

## 5.1 Analysis of machine learning algorithms

**Decision making tree.** The advantages of this method is that it is stable for data gaps (when the object is skipped, the object is skipped to the left and right subtree), and it allows various types of data (data does not need to be normalized, because it is a logical classifier). The disadvantage is that it is sensitive to noise and changes in data, a small change in data can significantly change the tree. A re-learning option is also possible, which is solved by means of pruning and tree compositions. Another drawback is that it is a greedy algorithm, because in each vertex a predicate is chosen which the best locally is.

**Nearest neighbors method.** The advantage of this algorithm is simplicity, because the main parameter for optimization is only the number of neighbors. It also fits when an informative metric of distance is easier to find than signs, such as Levenstein distance for a sequence of characters. The disadvantage is that it is resource-intensive, since basic computations are conducted during the evaluation, rather than learning through. This drawback is due to the fact that the method of the closest neighbors has the so-called lazy teaching [17].

**Linear regression.** Linear regression has many advantages, one of which is rapid learning due to the fact that there are few parameters, because the number of parameters is equal to the number of signs. This algorithm can be used for large data through the optimization of a stochastic gradient descent, which takes objects alternately. Another advantage is that it is easy to interpret, the greater the weight, the more important this sign is. It is also suitable for sparse data. However, the algorithm has certain limitations. Linear regression can not achieve very high quality, for example, in random forest. It is mandatory to pre-process data. Signs need to be scaled up, noise and emissions have to be eliminated, what in turn requires additional resources.

**The method of supporting vectors.** In the reference vector method, the loss function always has a single minimum, which is global, that is, there are no problems with getting stuck in local minimums. A useful consequence of this algorithm is the allocation of supporting objects. It usually works well on small and clean data. The disadvantage is that the reference objects can be emissions or noise, so data needs to be cleared before training the classifier. Also it does not work well on large data, because with a large amount of training data this can take a lot of time [18].

**Random forest.** This algorithm restores complex dependencies, and does not retrain, because adding new trees to the composition only improves quality. It does not require pre-processing of data, since it is a logical classifier. One of the disadvantages is that he trains a long time and builds foresight for a long time through the depth of the trees. Also, the random forest does not work well with sparse data, due to the fact that the trees are not adapted for this purpose [19-25].

**Gradient boosting.** This algorithm restores complex dependencies and does not require pre-processing data. And also it is faster than a random forest. One of the drawbacks is that during training of model a reprogramming is possible and it does not work well with sparse data [26-31].

**Neural network.** The neural network is a composition of linear classifiers, therefore partially overtaking their advantages and disadvantages. The advantages are that the gradient descent is used in neural networks to optimize the model can effectively work with large data. Also, neural networks can recover complex dependencies, because it is a composition of algorithms and can have many neurons. The so-called deep neural networks, which have recently surpassed all other methods of machine learning in complex tasks, such as image recognition, text, etc., are especially effective for complex tasks. Regarding to the disadvantages of this approach, one of the drawbacks is that as the model grows, the complexity of its optimization increases and the time for which the result of the calculations is obtained. And also the tendency to retraining, which is solved by different methods of regularization [17, 32-37].

**Choice of machine learning algorithm.** After analyzing the algorithms, an artificial neural network was selected as an algorithm for finding the direction of the user's perspective. This method is chosen because the problem is non-linear and rather complicated for approximation [38-46].

## 5.2 Artificial Neural Networks

Part of the work associated with neural networks means that the system is not ready for use at first, so it needs to be trained. Training is carried out by updating parameters that improve the behavior of the system. During training with teacher monitoring of learning process using the educational examples of input-output is happening. Typical applications of the neural network are classification and pattern recognition [2]. The neural network approximates the function of the input and output data dependence [38-42].

**Perceptron.** Model of neuron in artificial neural networks perceptron (Fig. 1). Perceptron (English perceptron from the Latin perceptio - perception) is a mathematical or computer model of information perception of the brain (a cybernetic model of the brain) proposed by Frank Rosenblatt in 1957 and implemented as an electronic machine "Mark-1" in 1960. Perceptron became one of the first models of neural networks, and Mark-1 - the world's first neurocomputer. Despite its simplicity, the perceptron is capable of learning and solving quite complex tasks. The basic mathematical problem with which he is able to handle is the linear separation of arbitrary nonlinear sets, the so-called provision of linear separability.

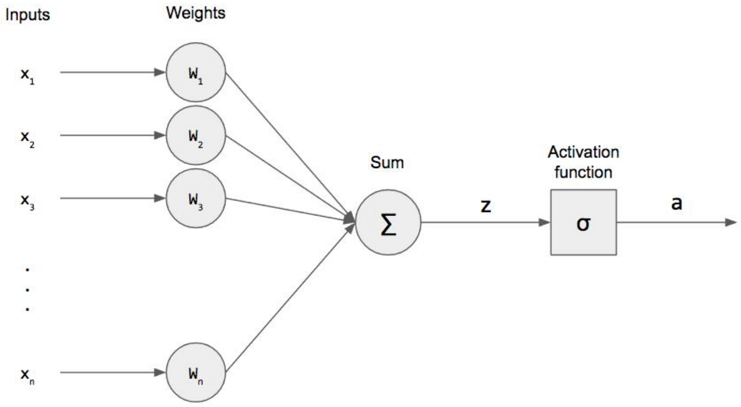


**Fig. 1.** Schematic representation of the fundamental part of the neural network

The perceptron consists of three types of elements, namely: the signals received from the sensors are transmitted to the associative elements, and hence to the responders. Thus, perceptron's allow the creation of a set of "associations" between input stimuli and the necessary response to the exit. In biological terms, this corresponds to the transformation of, for example, visual information into the physiological response of motor neurons. Against the background of the growing popularity of neural networks in 1969, the book Marvin Minsky and Seymour Paperner, which showed the principle of perceptron restriction. This led to a shift in the interest of researchers in artificial intelligence to the opposite of neural networks area of symbolic computing. In addition, due to the complexity of mathematical analysis of percutrines, as well as the lack of common terminology, various inaccuracies and errors occurred. Subsequently, interest in neural networks, and in particular, the work of Rosenblatt, was updated. For example, nowadays Biocomputing is rapidly developing, which in its theoretical basis of computing, in particular, is based on neural networks, and the perceptron is reproduced on the basis of bacteriorhodopsin-containing films.

An elementary perceptron consists of elements of three types: S-elements, A-elements and one R-element. S-elements are layers of sensors or receptors. In physical embodiment, they correspond, for example, to photosensitive retinal cells or camera matrix photoresistors. Each receptor can be in one of two states - rest or excitation, and only in the latter case it transmits a single signal to the next layer, associative elements. A-elements are called associative, because each such element, as a rule, corresponds to the whole set (association) of S-elements. The A-element is activated as soon as the number of signals from the S-elements at its input exceeds a certain value. The signals from the excited A-elements, in turn, are transmitted to the adder R, and the signal from the i-th associative element is transmitted with the weight coefficient. As well as A-elements, the R-element counts the sum of the values of input signals multiplied by the weights (linear form). The R-element, and with it the elementary perceptron, gives "1" if the linear form exceeds the threshold, otherwise the output will be "-1". Mathematically, the function implementing the R element can be written as follows: $f(x) = sign(\sum_{i=1}^{n} w_i \cdot x_i - \theta)$, where $x_i$ is incoming data, $w_i$ is weight coefficients, $\theta$ is trigger value of activation.

The training of the elementary perceptron consists in changing the weight coefficients of the A-R bonds. S-A bonds (which can take values (-1; 0; 1)) and threshold values of A-elements are randomly selected at the beginning and then do not change. After training, the perceptron is ready to work in recognition mode, or generalization. In this mode, perceptrons are presented to previously unknown objects, and he must establish to which class they belong. The work of the perceptron consists in the following: when the object is presented, the excited A-elements transmit the signal of the R-element equal to the sum of the corresponding weight coefficients. If this amount is positive, then the decision is made that this object belongs to the first class, and if it is negative - then to the second [10].

**The method of backpropagation.** Back propagation method is a multilayer perceptron training method. This is an iterative gradient algorithm that is used to minimize the operation error of a multilayer perceptron and obtain the desired output. The basic idea of this method is to spread the error signals from the outputs of the network to its inputs, in the direction of the reverse direct propagation of signals in the normal mode of operation. Barz and Okhonin proposed the general method ("duality principle"), which can be applied to a wider class of systems, including delayed systems, distributed systems, and so on. For the possibility of using the method of reverse error propagation, the activation function of the neurons should be differentiated [11].

## 5.3 Choice of system design

**System components.** A mouse point-of-view user recognition system will consist of two components:

1. ViewpointData - which will be able to recognize the position of the faces and store this data. It will also be able to train the neural network with that data.
2. Viewpoint - A mouse point-of-view recognition system that uses a trained neural network.

**The task of recognizing the gaze.** The task of recognizing the view can be divided into two parts:

1. Image analysis is identification of facial features that will be sufficient to recognize the look, using image analysis methods.
2. Machine learning is the creation and training of a machine learning algorithm using the signs found during the analysis of the image.

**General description of the algorithm.** The first step is based on getting a single frame from the camera. Once the frame has been received, face recognition begins. Finding the parts of the face separate regions of the eyes that are searching for the pupils. The method of finding pupils is to find a circle, but for this, you need to clear the image first. Threshold technology is useful for reducing noise in the image. Detecting contours of the image also improves the accuracy of finding the desired objects. After searching for pupils in regions where the eyes are located, filters can be searched. After finding the coordinates of the pupils and other facial parts, the neural network training starts with usage of these coordinates.

The task setting and reasonably chosen technologies for solving the task are carried out. This section presents all the main technical aspects of solving the task, describes the proposed solutions for the development of algorithmic, functional and programmatic components of the user's eye recognition application for mouse pointing. The algorithms of machine learning are analyzed, their advantages and disadvantages are determined. A detailed application model has been designed.

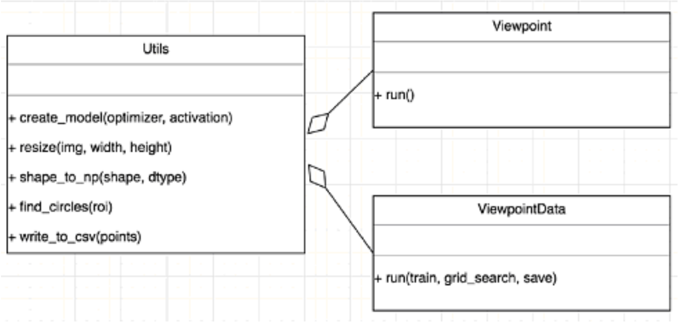

Utils
+ create_model(optimizer, activation)
+ resize(img, width, height)
+ shape_to_np(shape, dtype)
+ find_circles(roi)
+ write_to_csv(points)
Viewpoint
+ run()
ViewpointData
+ run(train, grid_search, save)

**Fig. 2.** Class Diagram

# 6 Realization and Testing of a Viewpoint Development Application for Mouse Pointer Control by View of User

## 6.1 Eye recognition

**Signs of Haar.** Haar signs are signs of a digital image used in pattern recognition. Signs of Haar were used in the first detector of people working in real time. Historically, algorithms that work only with the intensity of the image have a high computational complexity. Viola and Jones adapted the idea of using Haar wavelets and developed what was called Haar signs (Fig. 3). The sign of Haar consists of adjacent rectangular domains. They are positioned in the image, then the intensity of the pixels in the regions are added, then the difference between the amounts is calculated. This difference will be the value of a certain attribute, a certain size, in some way with positioned on the image [12].

**Histogram of oriented gradients.** Histogram of Oriented Gradients (HOG) is a descriptor of features used in the computer vision and image processing to recognize objects. The method calculates the gradient directions in the local points of the image. It is calculated in a dense network of evenly spaced cells and used to improve the accuracy of the local normalization of contrast [13].

**Choosing a method for eye recognition.** To recognize the image, the image was converted from a color image into black and white, which greatly increases the speed of operation, without losing most of the image information. For face and eye recognition, the Violi-Jones method, which in turn uses signs similar to those of Haar, was first used. Finding an eye occurs after a person is found, that is, the first one is looking for a rectangular area in which the face is located, then it searches for the eye, it is done in order to discard a large number of images and search only in the region in which the face is located (Fig. 4).

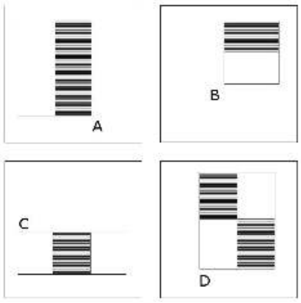


**Fig. 3.** Types of attributes used in the Viola-Jones algorithm

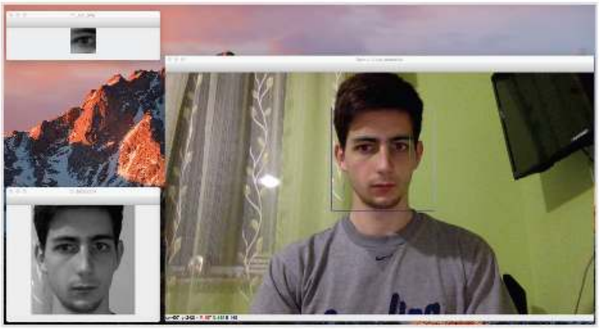


**Fig. 4.** Face and Eye Recognition

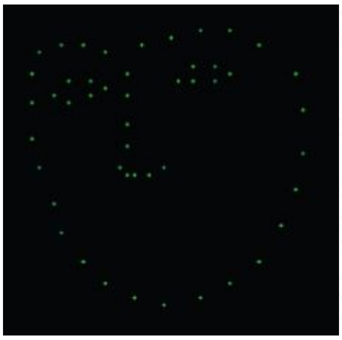


**Fig. 5.** Coordinates of face, found using a stratified gradient histogram

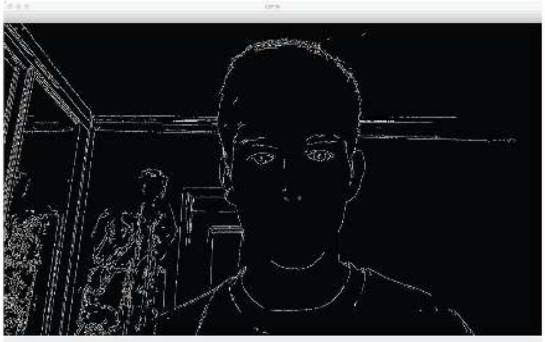


**Fig. 6.** A filter used by Kenny's operator

After the eye regions were found using the Viola-Jones method, the face detection tool that comes with the Dlib library was used. It uses a directed gradient histogram. The trained detector was used to obtain 68 coordinates located on the face, which include the coordinates of the contours of the eyes, mouth, nose, and the like. It was decided to discard the set of coordinates of the contours of the mouth, and leave only 48 coordinates that are necessary for the recognition of the eye, namely the coordinates of the contours of the eyes, eyebrows, nose and facial contours (Fig. 5). After analyzing the accuracy of the Viola-Jones algorithm and comparing it with the method using the directed gradient histogram, it was decided to use the latter, through more precise and more reliable recognition.

## 6.2 Recognition of pupils

After finding rectangular areas in which the eyes are found, program searches for pupils in these regions. To find the pupils, Haf's transformation is used to search for round objects in these regions. OpenCV provides a HoughCircles class that uses the Kenny operator to find the boundaries of the image (Fig. 6), and then searches for round objects using the Haf transform.

## 6.3 Data collection for training the neural network

To collect data and train the neural network, a separate system module was implemented that enabled them to track the coordinates and record them in a csv format file. The data consisted of fifty two-dimensional facial coordinates (forty-eight coordinates of faces and two coordinates of pupils) and position coordinates of the cursor. That is, in general for training the network used fifty one coordinates, fifty input parameters and one output (Table 1). Data collection lasted for fifteen minutes and the total number of rows in the table was about thirteen thousand (Table 1).

**Table 1.** Part of the data used to train the neural network

| | | | | | |
|---|---|---|---|---|---|
| (480, 246) | (480, 282) | … | (35.5, 17.5) | (41.5, 18.5) | (869.7578125,482.58203125) |
| (474, 258) | (480, 294) | … | (35.5, 15.5) | (41.5, 14.5) | (838.81640625,444.55078125) |
| (474, 258) | (480, 288) | … | (35.5, 15.5) | (41.5, 20.5) | (837.7734375,442.69140625) |
| (480, 252) | (480, 288) | … | (35.5, 14.5) | (41.5, 15.5) | (835.83203125,439.546875) |
| (474, 252) | (480, 288) | … | (35.5, 15.5) | (41.5, 15.5) | (833.5703125,436.6796875) |
| … | … | … | … | … | … |

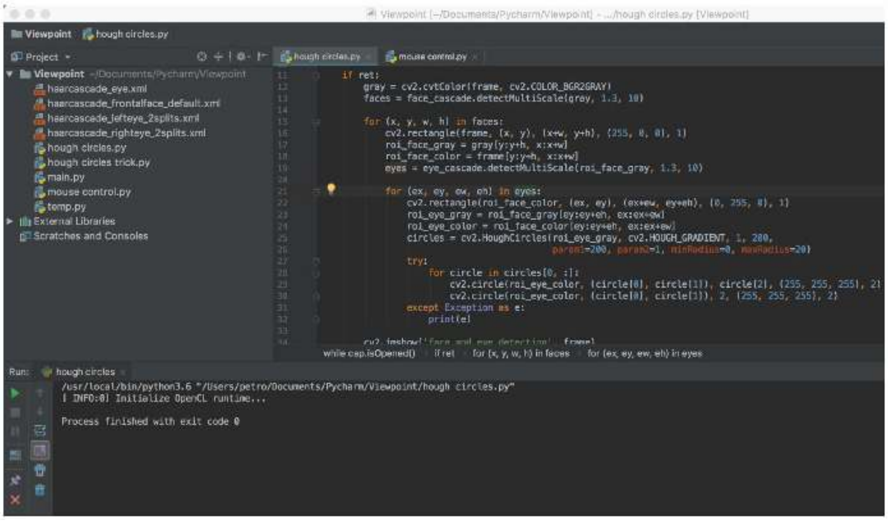
## 6.4 Hyperparameters of the neural network

Data obtained during the recognition of facial and pupil coordinates were used to train the neural network. The selection of hyperparameters was performed using the Grid Search method, which is a complete overview of hyperparameter combinations from the set of hyperparameters given manually. Using this method, it was determined that the smallest error in this task has a neural network that has five hundred twelve neurons in the first layer, sixty-four neurons in the second, and the number of objects that will be taken simultaneously during the training of sixty-four. For regularization during training, one-fifth of all neurons in the inner layers were nullified. As an activation function, the corrected linear Relu function was used. Adam's algorithm was chosen to optimize the model. In order to evaluate the quality of the algorithm, the mean square deviation was selected.

## 6.5 Used tools

From the well-known IDE (Integrated development environment), PyCharm was used (Fig. 7). This development environment has been applied because it is currently the most popular software development program in the Python programming language.

PyCharm has a user-friendly interface, which provides the ability to quickly navigate through files and code, as well as intelligent tips that increase the speed of writing code. OpenCV was used to process images. The library is developed by Intel and is now supported by Willow Garage and Itseez. The crimson code of the library is written in C ++ and distributed under the BSD license. Bindings are prepared for various programming languages such as Python, Java, Ruby, Matlab, Lua and others. Can be freely used for academic and commercial purposes [14].



Viewpoint [~/Documents/Pycharm/Viewpoint] - .../hough circles.py [Viewpoint]
Viewpoint
hough circles.py
Project
hough circles.py
mouse control.py
Viewpoint ~/Documents/Pycharm/Viewpoint
haarcascade_eye.xml
haarcascade_frontalface_default.xml
haarcascade_lefteye_2splits.xml
haarcascade_righteye_2splits.xml
hough circles.py
hough circles trick.py
main.py
mouse control.py
temp.py
External Libraries
Scratches and Consoles
while cap.isOpened()
if ret
for (x, y, w, h) in faces
for (ex, ey, ew, eh) in eyes
Run:
hough circles

```
    if ret:
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, 1.3, 10)

        for (x, y, w, h) in faces:
            cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 1)
            roi_face_gray = gray[y:y+h, x:x+w]
            roi_face_color = frame[y:y+h, x:x+w]
            eyes = eye_cascade.detectMultiScale(roi_face_gray, 1.3, 10)

            for (ex, ey, ew, eh) in eyes:
                cv2.rectangle(roi_face_color, (ex, ey), (ex+ew, ey+eh), (0, 255, 0), 1)
                roi_eye_gray = roi_face_gray[ey:ey+eh, ex:ex+ew]
                roi_eye_color = roi_face_color[ey:ey+eh, ex:ex+ew]
                circles = cv2.HoughCircles(roi_eye_gray, cv2.HOUGH_GRADIENT, 1, 200,
                                           param1=200, param2=1, minRadius=0, maxRadius=20)
                try:
                    for circle in circles[0, :]:
                        cv2.circle(roi_eye_color, (circle[0], circle[1]), circle[2], (255, 255, 255), 2)
                        cv2.circle(roi_eye_color, (circle[0], circle[1]), 2, (255, 255, 255), 2)
                except Exception as e:
                    print(e)

        cv2.imshow('face and eye detection', frame)
```

```
/usr/local/bin/python3.6 "/Users/petro/Documents/Pycharm/Viewpoint/hough circles.py"
[ INFO:0] Initialize OpenCL runtime...

Process finished with exit code 0
```

**Fig. 7.** PyCharm development environment

The Dlib library was used to find faces. Keras was used to develop the neural network. Keras is an open library written in Python. It is capable of working on top of Deeplearning4j, TensorFlow and Theano. Designed to enable rapid experiments with deep learning networks, it is focuses on being minimal, modular and extensible. It was created as part of the research effort of the ONEIROS project, and its main author and sponsor is Francois Schollie, a Google engineer. Although the Google TensorFlow team decided to support Keras in the TensorFlow main library, Scholl said that Microsoft is working on adding to Keras and the back of CNTK [16].

## 6.6 Testing the application

During the test, it was found that the average square deviation is equal to 17 thousand, which in turn means that the average error is about one hundred pixels. About speed, the average delay is one second. After testing the system, we can conclude that the error is too large to be used for convenient and accurate control of the mouse pointer. The accuracy of the system is heavily dependent on the lighting and quality of the webcam. Due to the complexity of the function approximated, a neural network with a large number of neurons was used, which in turn slowed down the system due to the large number of calculations to be done. Another reason for the delay is the analysis of the image, which is a rather resource-intensive task. The program implementation of project decisions is described, the application testing report is submitted. In order to ensure the effective use of the developed software, a user manual is provided.

## 6.7 User Manual

**Software components.** The package is developed with usage of the Python 3 programming language in PyCharm development environment. During the design of subsystems, a combination of object-oriented approach to programming with procedurally-oriented. All classes were documented. The AutoPy, Keras, OpenCV and Dlib libraries were used. For correct operation of the package, you need a custom machine with a processor of at least 400 MHz, RAM of at least 512 Mb. To use package, you need to install Python 3 programming language interpreter and have all the necessary library files and settings available. The list of necessary files, explanation of their purpose and information about belonging to a specific project is given in the Table 2.

**Table 2.** A set of files for correct operation of PPP "Clusterization"

| File | Appointment |
|---|---|
| ViewpointData.py | A file for running components for data collection and training of the neural network |
| ViewPoint.py | A file to launch components for tracking the look of the mouse pointer |
| Points.csv | File with data set for training neural network |
| Model.h5 | Neural Network Model |

**Software installation.** So the package could work, you must install the Python interpreter.

**Setting software.** No additional configuration is required for the system.

**Base function of software.** To save the facial features coordinates, run the system component called ViewpointData with the -save command. For visual representation of the coordinates, run the system component called ViewpointData with the -show command. The system displays a window with coordinates on the face. To display only coordinates, execute the remove_background command. To train the neural network, run the system component called ViewpointData with the -train command. To control the mouse pointer, run the system component Viewpoint.

**Error analysis.** When the message "Failed to capture a frame from camera" occurs, check the camera's operation.

# 7 Conclusion

The results of error analysis obtained during test indicate that the system is ineffective. During image recognition, there are numbers of limitations that impair the accuracy of the recognition. These limitations are computing power of computers, as well as quality of webcam shooting. There are several ways to improve the accuracy of the system. One is to get more data. This, in turn, will enable the algorithm to adapt to different conditions. But with this increase it is likely to increase the size of the neural network, which in turn will lead to further delays in work. Not all coordinates with which the system works may be necessary to recognize the look, because some of them can be linearly dependent. This, in turn, often leads to getting stuck in the local minimum, due to the problem of multicollinearity. Therefore, one should study the importance of each coordinate and leave only those that are sufficient for the recognition of the view. You can accelerate the system by improving computations using a graphics processor rather than a CPU. Another way to improve the results to look for is the roller neural network which shows high accuracy of image recognition. By effectively using a convolutional neural network, better recognition accuracy can be achieved. Using a convolutional neural network can also be used to find necessary coordinates, which can then be transmitted to another machine learning algorithm to find coordinates of mouse pointer and to recognize coordinates of mouse immediately using only convolutional neural network, which will directly work with processed and filtered image. After analyzing the work result was obtained, which is the content of further research in this area, due to many possibilities of improvement of algorithm of recognition of user's perspective, as well as through the rapid development of technology related to neural networks.

## References

1. B. Georg What Do You See When You’re Surfing? Using Eye Tracking to Predict Salient Regions of Web Pages / B. Georg, C. Edward, R. M. Meredith., 2007. – 10 c.
2. Back D. Neural Network Gaze Tracking using Web Camera / David Back., 2003. – 52 c.
3. Joakim R. Perception-based second generation image coding using variable resolution. Master’s thesis, Department of Electrical Engineering / Joakim Rydell. – Linköping, Sweden, 2003. – 54 c.
4. Nischal M. Piratla A neural network based real-time gaze tracker / Nischal M. Piratla, Anura P. Jayasumanay., 2001. – 196 c.
5. Wild D. J. Gaze Tracking Using A Regular Web Camera, Rhodes University / David Jonathan Wild., 2012. – 52 c.
6. Sewell W. Real-Time Eye Gaze Tracking With an Unmodified Commodity Webcam Employing a Neural Network / W. Sewell, O. Komogortsev., 2010. – 6 c.
7. Wąsikowska B. The Application of Eye Tracking in Business / Barbara Wąsikowska., 2014. – 244 c.
8. Segaran T. Programming Collective Intelligence / Toby Segaran., 2008. – 368 c.
9. Mitchell T. Machine Learning / Tom Mitchell., 1997. – 432 c.
10. Nascimento, G., Ribeiro, M., Cerf, L., Cesário, N., Kaytoue, M., Raïssi, C., Meira, W.: Modeling and analyzing the video game live-streaming community. In: Latin American Web Congress. 1-9. (2014).
11. Obermaier, J., Hutle, M.: Analyzing the security and privacy of cloud-based video surveillance systems. In: Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security, 22-28. (2016).
12. Xu, D., Wang, R., & Shi, Y. Q.: Data hiding in encrypted H. 264/AVC video streams by codeword substitution. In: IEEE transactions on information forensics and security, vol. 9(4), 596-606. (2014).
13. Saxena, M., Sharan, U., Fahmy, S.: Analyzing video services in web 2.0: a global perspective. In: Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video, pp. 39-44. (2008).

14. Brône, G., Oben, B., Goedemé, T.: Towards a more effective method for analyzing mobile eye-tracking data: integrating gaze data with object recognition algorithms. In: Proceedings of the 1st international workshop on pervasive eye tracking & mobile eye-based interaction, 53-56. (2011).
15. Reibman, A. R., Sen, S., Van der Merwe, J.: Analyzing the spatial quality of internet streaming video. In: in Proceedings of International Workshop on Video Processing and Quality Metrics for Consumer Electronics. (2005)
16. Perniss, P. Collecting and analyzing sign language data: Video requirements and use of annotation software. In: Research methods in sign language studies, 56-73. (2015).
17. Tran, B. Q.: U.S. Patent No. 8,849,659. Washington, DC: U.S. Patent and Trademark Office. (2014)
18. Badawy, W., Gomaa, H.: U.S. Patent No. 9,014,429. Washington, DC: U.S. Patent and Trademark Office. (2015)
19. Badawy, W., Gomaa, H.: U.S. Patent No. 8,630,497. Washington, DC: U.S. Patent and Trademark Office. (2014)
20. Golan, O., Dudovich, B., Daliyot, S., Horovitz, I., Kiro, S.: U.S. Patent No. 8,885,047. Washington, DC: U.S. Patent and Trademark Office. (2014)
21. Chambers, C. A., Gagvani, N., Robertson, P., Shepro, H. E.: U.S. Patent No. 8,204,273. Washington, DC: U.S. Patent and Trademark Office. (2012).
22. Maes, S. H.: U.S. Patent No. 7,917,612. Washington, DC: U.S. Patent Office. (2011).
23. Peleshko, D., Ivanov, Y., Sharov, B., Izonin, I., Borzov, Y.: Design and implementation of visitors queue density analysis and registration method for retail videosurveillance purposes. In: Data Stream Mining & Processing (DSMP), 159-162. (2016)
24. Maksymiv, O., Rak, T., Peleshko, D.: Video-based Flame Detection using LBP-based Descriptor: Influences of Classifiers Variety on Detection Efficiency. In: International Journal of Intelligent Systems and Applications, 9(2), 42-48 (2017)
25. Rusyn, B., Lutsyk, O., Lysak, O., Lukeniuk, A., Pohreliuk, L.: Lossless Image Compression in the Remote Sensing Applications. In: Int. Conf. on Data Stream Mining & Processing (DSMP), 195-198 (2016)
26. Kravets, P.: The control agent with fuzzy logic. In: Perspective Technologies and Methods in MEMS Design, MEMSTECH'2010, 40-41 (2010)
27. Babichev, S., Gozhyj, A., Kornelyuk A., Litvinenko, V.: Objective clustering inductive technology of gene expression profiles based on SOTA clustering algorithm. In: Biopolymers and Cell, 33(5), 379–392. (2017)
28. Nazarkevych, M., Klyujnyk, I., Nazarkevych, H.: Investigation the Ateb-Gabor Filter in Biometric Security Systems. In: Data Stream Mining & Processing, 580-583. (2018).
29. Lytvyn, V., Sharonova, N., Hamon, T., Vysotska, V., Grabar, N., Kowalska-Styczen, A.: Computational linguistics and intelligent systems. In: CEUR Workshop Proceedings, Vol-2136 (2018)
30. Vysotska, V., Fernandes, V.B., Emmerich, M.: Web content support method in electronic business systems. In: CEUR Workshop Proceedings, Vol-2136, 20-41 (2018)
31. Lytvyn, V., Peleshchak, I., Vysotska, V., Peleshchak, R.: Satellite spectral information recognition based on the synthesis of modified dynamic neural networks and holographic data processing techniques. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), 330-334 (2018)
32. Lytvyn, V., Vysotska, V., Peleshchak, I., Rishnyak, I., Peleshchak, R.: Time Dependence of the Output Signal Morphology for Nonlinear Oscillator Neuron Based on Van der Pol Model. In: International Journal of Intelligent Systems and Applications, 10, 8-17 (2018)

33. Su, J., Sachenko, A., Lytvyn, V., Vysotska, V., Dosyn, D.: Model of Touristic Information Resources Integration According to User Needs. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies, 113-116 (2018)
34. Vysotska, V., Hasko, R., Kuchkovskiy, V.: Process analysis in electronic content commerce system. In: International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT), 120-123 (2015)
35. Lytvyn, V., Vysotska, V., Veres, O., Rishnyak, I., Rishnyak, H.: The Risk Management Modelling in Multi Project Environment. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 32-35 (2017)
36. Shakhovska, N., Vysotska, V., Chyrun, L.: Features of E-Learning Realization Using Virtual Research Laboratory. In: Computer Science and Information Technologies, 143–148 (2016)
37. Shakhovska, N., Vysotska V., Chyrun, L.: Intelligent Systems Design of Distance Learning Realization for Modern Youth Promotion and Involvement in Independent Scientific Researches. In: Advances in Intelligent Systems and Computing 512. Springer International Publishing AG, 175–198 (2017)
38. Lytvyn, V., Vysotska, V., Demchuk, A., Demkiv, I., Ukhanska, O., Hladun, V., Kovalchuk, R., Petruchenko, O., Dzyubyk, L., Sokulska, N.: Design of the architecture of an intelligent system for distributing commercial content in the internet space based on SEO-technologies, neural networks, and Machine Learning. In: Eastern-European Journal of Enterprise Technologies, 2(2-98), 15-34 (2019)
39. Gozhyj, A., Kalinina, I., Vysotska, V., Gozhyj, V.: The method of web-resources management under conditions of uncertainty based on fuzzy logic. In: IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018 – Proceedings 1, 343-346 (2018)
40. Rusyn, B., Vysotska, V., Pohreliuk, L.: Model and architecture for virtual library information system. In: IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, 37-41 (2018)
41. Kowalik, D., Rusyn, B.: Innovative Vocation Didactics Aimed at the Preparation of Staff According to Indastry 4.0 and Europe 2020. In: Int. Conf. on Education Reform and Modern Managment, 12-17 (2017)
42. Kowalik, D.: New Vocation Qualification Course as a Tool for the Increase in the Mobility on the European Labour Market. In: Int. Conf. on Education Reform and Modern Managment, 80-85 (2016)
43. Mukalov, P., Zelinskyi, O., Levkovych, R., (...), Pylyp, A., Shakhovska, N.: Development of system for auto-tagging articles, based on neural network. In: CEUR Workshop Proceedings, Vol-2362, 116-125. (2019)
44. Shakhovska, N. B., Noha, R. Y.: Methods and tools for text analysis of publications to study the functioning of scientific schools. In: Journal of Automation and Information Sciences, 47(12). DOI: 10.1615/JAutomatInfScien.v47.i12.30 (2015).
45. Shakhovska, N., Shvorob, I.: The method for detecting plagiarism in a collection of documents. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies"(CSIT), 142-145. (2015)
46. Arzubov, M., Shakhovska, N., Lipinski, P.: Analyzing ways of building user profile based on web surf history. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), Vol. 1, 377-380. (2017)