A Mobile Facial Recognition System Based on a Set of Raspberry Technical Tools

Nikolay Kiktev [1,2], Taras Lendiel [1] and Oleksandr Korol [1]

[1] *National University of Life and Environmental Sciences of Ukraine, Heroiv Oborony str., Kyiv, 03041, Ukraine*

[2] *Taras Shevchenko National University of Kyiv, 64/13, Volodymyrs'ka str., Kyiv, 01601, Ukraine*

Abstract

Most machine vision systems use complex and expensive hardware. The work proposes to consider Raspberry hardware for the implementation of mobile machine vision systems, which will allow the implementation of the specified systems as mobile pattern recognition stations. The hardware and software of the mobile machine vision system, which reads and stores information about the recognized image, has been implemented. This article develops a mobile machine vision system that allows you to recognize a face and compare it with one stored in a database, after which the control element generates a control action on the output ports. The method of histograms of oriented gradients is used to recognize faces during the operation of the machine vision system.

Keywords [1]

machine vision, pattern recognition, automation systems, histogram of oriented gradients, neural network, learning, image.

1. Introduction

Automated machine vision systems for identifying objects by image belong to the type of pattern recognition systems [1, 2]. The development of pattern recognition systems has reached a scale when these systems can work with one specific pattern recognition algorithm, and neural networks are used to increase the recognition speed [3]. Machine vision systems for pattern recognition, in addition to work with the detection of static images, are also used to determine the movements of a specified object or a group of them [4].

2. Literature review and problem statement

In the work of V. Martsenyuk et al. [5] proposed an algorithm that allows you to correctly identify images with different characteristics for identifying a person in a video stream. It includes anisotropic diffusion as an image preprocessing method, Gabor wavelet transform as an image processing method, histogram of oriented gradients (HOG), and one-dimensional local binary patterns (1DLBP) as methods for extracting feature vectors from images. The study used three facial image databases: The Database of Faces, Facial Recognition Technology (FERET) database, and Surveillance Cameras Face Database (SCface). The operation of the algorithm gave different results of identification accuracy with an average difference of 20%. The authors conducted experiments with different degrees of image compression, differences in image resolutions and areas of facial areas covering the images.

In the article by the authors O. Bychkov et al. [6] proposes a study of methods based on local texture descriptors in comparison with methods based on neural networks. In particular, an approach to personal identification is proposed, based on local texture descriptors of facial images, eliminating the disadvantages of algorithms based on neural networks. As a result of the experimental study, the identification accuracy of the proposed algorithm increased by more than 10% compared to the use of neural networks under conditions of different positions of the subject's head.

The article by N. Korshun et al. [7] proposed a methodology that describes the stages of integration of artificial intelligence and machine learning, including data input, system state analysis, decision making, optimization actions and continuous monitoring.

Researchers from Saudi Arabia Alonazi, M. et al. [8] proposed to perform facial emotion recognition using the Pelican optimization algorithm with a deep convolutional neural network. The authors optimized the performance of the CapsNet model by tuning the hyperparameters using the Pelican Optimization Algorithm (POA). This ensured that the model was fine-tuned to detect a wide range of emotions and generalize effectively across different populations and scenarios. Detection and classification of different types of facial emotions occurs using a bidirectional long-short-term memory (BiLSTM) network. The simulation analysis of the AFER-POADCNN system was tested on the FER benchmark dataset.

American researchers Zheng, Y. and Blasch, E. [9] performed facial microexpression recognition using score fusion and a hybrid model from a convolutional LSTM and Vision Transformer. The next step in human-machine interaction is to be able to detect facial emotions (for example, a humanoid robot). Allowing systems to recognize microexpressions allows the machine to delve deeper into a person's true feelings, allowing human emotions to be taken into account when making optimal decisions. Such machines will be able to detect dangerous situations, alert caregivers to problems and provide appropriate responses (for example, at an airport). The authors proposed a new hybrid neural network (NN) model capable of recognizing microexpressions in real-time applications. This study first compares several NN models and then creates a hybrid model by combining a convolutional neural network (CNN), a recurrent neural network (RNN, it can be long short-term memory (LSTM)) and a vision transformer.

There is also an interesting study on this topic by Italian scientists Raccanello, D. and Burro, R. [10]. This work examined the suitability of the Drawing Set for the Assessment of Children's Achievement Emotions (DS-AEAL). The authors considered the theory of control-value analysis as the main theoretical basis. Specifically, a set of 10 drawings of faces representing pleasure, pride, hope, relief, relaxation, anxiety, anger, shame, sadness, and boredom was developed using 259 adults as raters. The authors also administered a matching task and a labeling task to 89 adults. The results confirmed the suitability of the correspondence between DS-AEAL and verbal labels. Overall, recognition and recall were better for primary emotions compared to secondary ones. Despite their preliminary nature, the results of these studies support the suitability of the DS-AEAL for assessing achievement emotions in a variety of learning contexts along with associated verbal labels.

A simulated dataset generator for testing data analysis methods is presented in the article by Toliupa, S. et al. [11].

2. The purpose and objectives of the research

The purpose of the research is to develop software and hardware for a mobile machine vision system for image recognition, which will allow monitoring of the video image and recording of the recognized image from the input video stream.

3. Research materials and methods

It is proposed to develop a mobile machine vision system based on a set of Raspberry technical tools. Raspberry is the hardware of embedded devices for creating automated control systems [12]. The specified hardware is also used to implement automated systems based on Internet of Things technology [13].

A Raspberry Model Pi3 mini-computer was chosen as the controlling element of the mobile machine vision system. The control element receives video images from the electrically connected camera module OV5647 [14]. The structure of the proposed system is shown in fig. 1.
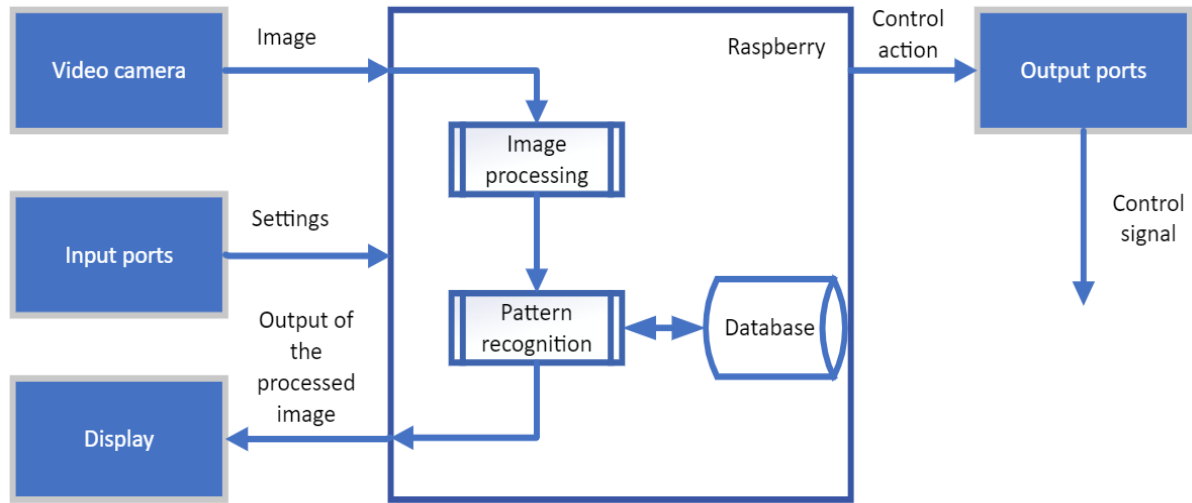

Figure 1: Structural diagram of the functioning of the mobile machine vision system

The pattern recognition algorithm is based on the histogram of oriented gradients (HOG) method. The HOG (Histogram of Oriented Gradients) method is one of the most effective for this task. The main idea is to use pixel intensity gradients to construct feature vectors that describe the image structure [15, 16].

During the execution of the HOG method, the image goes through a number of processing stages [17, 18]. First, the image is divided into small sectors (for example, 8*8 pixels). The next stage is the calculation of gradients, where horizontal (Gx) and vertical (Gy) intensity gradients are calculated for each pixel in the sector:

$$G_x = I(x + 1,y) - I(x - 1,y)$$

$$G_y = I(x,y + 1) - I(x,y - 1)$$

where I(x,y) is the pixel intensity in coordinates (x,y) in the image.

The calculation of intensity gradients is used to determine changes in pixel intensity in horizontal and vertical directions, which is the basis for edge detection in an image.

During the execution stages of the method, the important characteristics are the direction of the gradient θ and the magnitude of the gradient $\left|G\right|$, which are determined at each point of the image:

The complete data set consists of 18 csv files (Fig. 2).

$$\theta = arctan(G_y / G_x)$$

$$\left|G\right| = \sqrt{G_x^2 + G_y^2}$$

The direction of the gradient indicates the direction of the fastest change in intensity, and the magnitude of the gradient indicates how rapidly the pixel value is changing.

A histogram of gradient directions is created for each sector. Gradient angles are divided into bins, for example, with an interval of 20° (0°, 20°, 40°, ..., 180°). Each pixel contributes to the corresponding bin depending on the direction of the gradient. The weight of the contribution is determined by the magnitude of the gradient.

To detect the image of a face in the processed image, a recognition method is used, where regression based on gradient lifting is used to accurately place control points. This approach uses a regression calculation based on gradient ascent to precisely place control points:

$$p = p_0 + \sum_{i=1}^{n} \alpha_i d_i$$

where p_0 is the initial position of control points,

α_i is weighting factors,

d_i is base vectors.

The face image of an individual is stored in the memory of a mini-computer.

In order to compare the face of the saved person with other faces, encoding into feature vectors was performed. The recognition algorithm will be executed by the developed neural network, which will execute the specified image [19, 20].

A neural network consists of several layers: convolutional layers, pooling layers, and fully connected layers. Convolutional layers extract features, subsampling layers reduce dimensionality, and fully connected layers perform classification [19, 21].

The network is trained on a large dataset of face images using a loss function (F) to optimize parameters [19, 21]:

$$F = \sum_{i,j} (y_{i,j} \cdot \|f(x_i) - f(x_j)\|^2 + (1 - y_{i,j}) \cdot \max(0, m - \|f(x_i) - f(x_j)\|^2))$$

where $y_{i,j}$ are labels of pairs of coordinates x_i, x_j;

$f(x)$ is a function of a neural network, which maps the input data x into the feature space;

m is the limit of the set of statistical features and results of the neural network model;

$\| f(xi) - f(xj) \|$ is the Euclidean distance between the vector representations xi and xj in the feature space.

The last stage is the comparison of faces. For this, the Euclidean distance between feature vectors is used [19-21]:

$$d = \sqrt{\sum_{i=1}^{128} (f_i - g_i)^2}$$

where f_i and g_i are the components of feature vectors of two faces.

If the distance is less than the specified threshold, the faces are considered the same.

Hardware and software of the system.

The software was implemented in the Python programming language and tested on a laboratory model (Fig. 2).

Using a camera and display from Adafruit, students will begin to work with machine vision technologies, face recognition, and a desktop graphical interface for system management.

To prepare the stand, we used:

- Raspberry Pi 3
- Camera for Raspberry Pi
- Adafruit display
- LED
- Plastic case

Programming was done using my Python using face_recognition libraries for robot recognition I want to use tkinter to create a graphical interface.

The working algorithm of the developed layout is shown in Fig. 3. During the operation of the control system, it is assumed that certain information is recorded with a time counter. The recording of the specified information will be performed with the assignment of the time counter value "i".

The screen of the interface of the developed software is shown in Fig. 4. Esp8266 program code is given in the Appendix.

Stages of working with the program.

At launch, we see a window of our program with an image from the camera and a button. The face is currently marked as Unknown.

• Press the button and then enter the name.

• We take a series of pictures in different positions of the head.

• Then we see a message about successfully added photos.

• Now, when the app recognizes the added faces, the LED turns on, if it failed to recognize the added person, it turns off after 5 seconds.

After successful testing of the system, a case was made for more convenient and safe use (Fig. 5). In the future, this stand can be used to teach students how to work with machine vision and face recognition. Students will be able to familiarize themselves with the principle of work thanks to a detailed article written on this topic and work with the stand.

Practical implementation.

• Access automation: Creation of access control systems at enterprises or offices using facial recognition.

• Development of IoT solutions: Integrating cameras, microcontrollers and machine learning algorithms to create smart solutions for home and business.

• Working with data and security: Use of encryption and data validation methods to ensure information security in access systems.
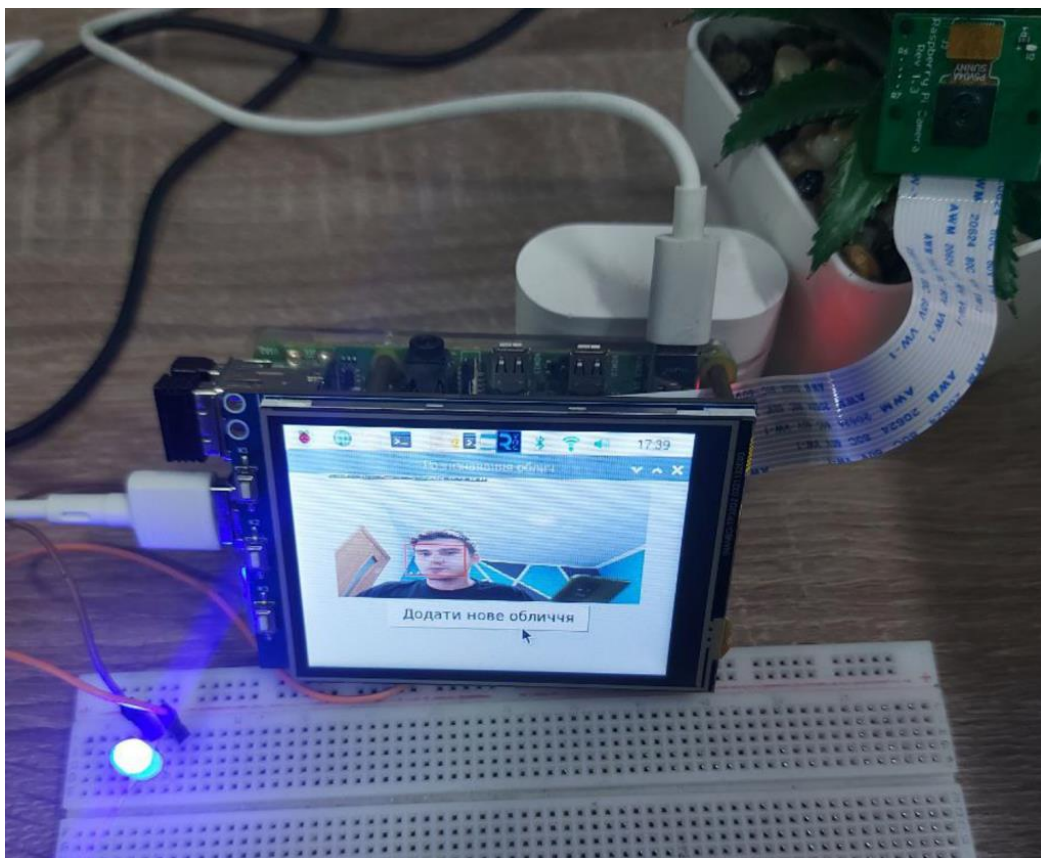

Figure 2: Laboratory model of machine vision

4. Discussion and prospects

This article is a combination of two directions in IT - image recognition using neural networks and Internet of Things technology for managing technological objects. Internet of things technology has become widespread in many industries nowadays.

The article of Berestov, D. et al. [22] analyzes the application of big data in agriculture based on Internet of things (IoT) technology. An IoT information collection platform for agricultural land is proposed, which can help forecast and analyze the weather for agriculture.

The research conducted in this article can be further used in our research in the automation of agricultural production: for plant productivity management [23], phytomonitoring [2], in robotic complexes, including in horticulture and greenhouse farming [24].
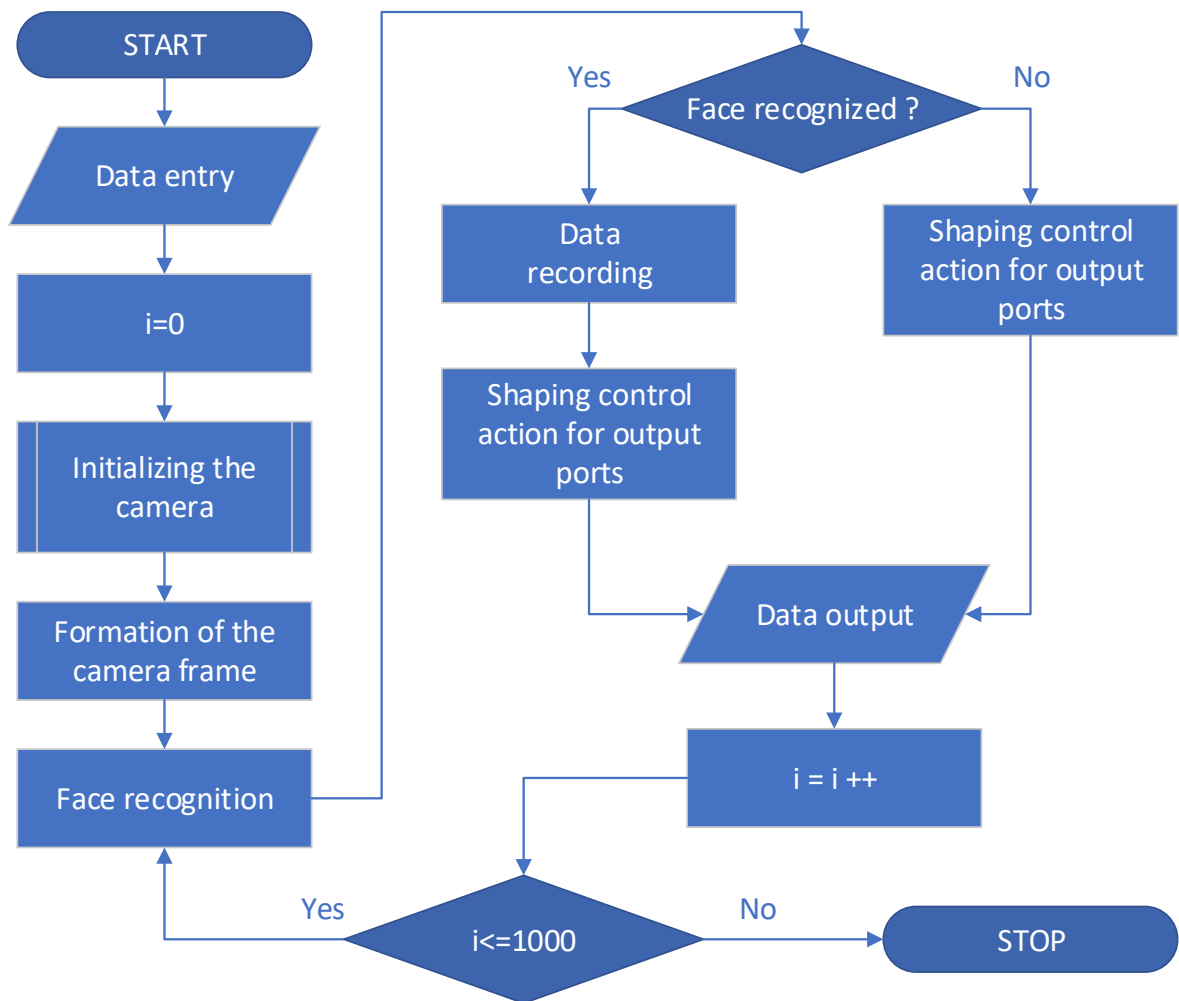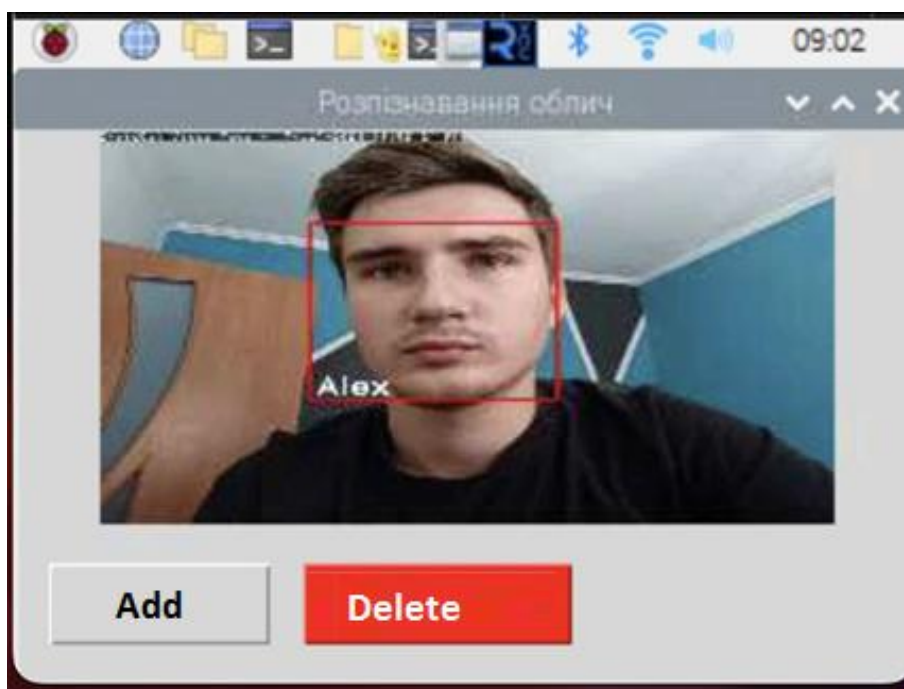
Figure 3. Algorithm flowchart

Figure 4. Appearance of the interface

Figure 5. Laboratory stand for teaching students how to work with machine vision and face recognition

5. Conclusions

A mobile machine vision system based on a set of Raspberry technical tools has been developed. Using the histogram method of oriented gradients, face recognition is performed during the operation of the machine vision system. By using the neural network tool, training was performed to recognize the stored face specified in the database. The mobile machine vision system allows you to recognize the face and compare it with the one stored in the database, after which the control element forms a control action on the output ports. The specified technical implementation allows the use of a mobile machine vision system in automated control systems and Internet of Things technology systems.

6. References

[1] J. Nayak and S. B. Kaje, "Fast Image Convolution and Pattern Recognition using Vedic Mathematics on Field Programmable Gate Arrays (FPGAs)," 2022 OITS International Conference on Information Technology (OCIT), Bhubaneswar, India, 2022, pp. 569-573, doi: 10.1109/OCIT56763.2022.00111.
[2] F. Ren, X. Zhang and L. Wang, "A new method of the image pattern recognition based on neural networks," Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, Harbin, China, 2011, pp. 3840-3843, doi: 10.1109/EMEIT.2011.6023833.
[3] Weiming Hu, Xuejuan Xiao, Zhouyu Fu, D. Xie, Tieniu Tan and S. Maybank, "A system for learning statistical motion patterns," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 9, pp. 1450-1464, Sept. 2006, doi: 10.1109/TPAMI.2006.176.
[4] Lysenko, V., Zhyltsov, A., Bolbot, I., Lendiel, T., & Nalyvaiko, V. (2020). Phytomonitoring in the phytometrics of the plants. In E3S Web of Conferences (Vol. 154, p. 07012). EDP Sciences.
[5] V. Martsenyuk, O. Bychkov, K. Merkulova and Y. Zhabska, "Exploring Image Unified Space for Improving Information Technology for Person Identification," in *IEEE Access*, vol. 11, pp. 76347-76358, 2023, doi: 10.1109/ACCESS.2023.3297488
[6] Bychkov, O., Zhabska, Y., Merkulova, K., Merkulov, M. Research and Comparative Analysis of Person Identification Information Technology. *C*EUR Workshop Proceedings*,* 2023, 3538, pp. 54–64. https://ceur-ws.org/Vol-3538/Paper_6.pdf
[7] Korshun, N., Myshko, I., Tkachenko, O. Automation and Management in Operating Systems: The Role of Artificial Intelligence and Machine Learning. *C*EUR Workshop Proceedings*,* 2023, 3687, pp. 59–68. https://ceur-ws.org/Vol-3687/Paper_6.pdf

[8] Alonazi, M.; Alshahrani, H.J.; Alotaibi, F.A.; Maray, M.; Alghamdi, M.; Sayed, A. Automated Facial Emotion Recognition Using the Pelican Optimization Algorithm with a Deep Convolutional Neural Network. *Electronics* 2023, *12*, 4608. https://doi.org/10.3390/electronics12224608

[9] Zheng, Y.; Blasch, E. Facial Micro-Expression Recognition Enhanced by Score Fusion and a Hybrid Model from Convolutional LSTM and Vision Transformer. *Sensors* 2023, *23*, 5650. https://doi.org/10.3390/s23125650

[10] Raccanello, D.; Burro, R. Development of a Drawing Set for the Achievement Emotions Adjective List (DS-AEAL): Preliminary Data on a Pictorial Instrument for Children. *Educ. Sci.* 2024, *14*, 756. https://doi.org/10.3390/educsci14070756

[11] Toliupa, S., Pylypenko, A., Tymchuk, O., Kohut, O. Simulated Datasets Generator for Testing Data Analytics Methods. C*EUR Workshop Proceedings*, 2023, 3687, pp. 11–24. https://ceur-ws.org/Vol-3687/Paper_2.pdf

[12] M. J. Al Hamdi, W. Mumtaz, N. Albar, T. A. Mifta, J. Ridha and K. Muchtar, "A Low-cost Raspberry Pi and Deep Learning System for Customer Analysis," 2023 IEEE 12th Global Conference on Consumer Electronics (GCCE), Nara, Japan, 2023, pp. 328-329, doi: 10.1109/GCCE59613.2023.10315605.

[13] Dominique Dom Guinard; Vlad M. Trifa, Building the Web of Things: With examples in Node.js and Raspberry Pi , Manning, 2016.

[14] Waveshare camera with optical stabilizer OV5647-70 5MP OIS Camera. Electronic resource: access mode - https://evo.net.ua/kamera-waveshare-z-optychnym-stabilizatorom-ov5647-70-5mp-ois-camera-22980/ (date of application 01.07.2024 p.).

[15] A. Ade-Ibijola and K. Aruleba, "Automatic Attendance Capturing Using Histogram of Oriented Gradients on Facial Images," 2018 IST-Africa Week Conference (IST-Africa), Gaborone, Botswana, 2018, pp. Page 1 of 8-Page 8 of 8.

[16] C. -P. Huang, C. -H. Hsieh, K. -T. Lai and W. -Y. Huang, "Human Action Recognition Using Histogram of Oriented Gradient of Motion History Image," 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control, Beijing, China, 2011, pp. 353-356, doi: 10.1109/IMCCC.2011.95.

[17] C. -S. Fahn, C. -P. Lee and Y. -S. Yeh, "A real-time pedestrian legs detection and tracking system used for autonomous mobile robots," 2017 International Conference on Applied System Innovation (ICASI), Sapporo, Japan, 2017, pp. 1122-1125, doi: 10.1109/ICASI.2017.7988208.

[18] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 886-893). Ieee.

[19] D. Ciregan, U. Meier and J. Schmidhuber, "Multi-column deep neural networks for image classification," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 3642-3649, doi: 10.1109/CVPR.2012.6248110.

[20] Y. Shi and Z. Yu, "Multi-Column Convolution Neural Network Model Based on Adaptive Enhancement," 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Xiamen, China, 2018, pp. 681-685, doi: 10.1109/ICITBS.2018.00177.

[21] Dan Cireşan, Ueli Meier, Jonathan Masci, Jürgen Schmidhuber, Multi-column deep neural network for traffic sign classification, Neural Networks, Volume 32, 2012, Pages 333-338, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2012.02.023.

[22] Berestov, D., Kurchenko, O., Zubyk, L., Kulibaba, S., Mazur, N. Assessment of Weather Risks for Agriculture using Big Data and Industrial Internet of Things Technologies. *CEUR Workshop Proceedings*, 2023, 3550, pp. 1–13. https://ceur-ws.org/Vol-3550/paper1.pdf

[23] Nykyforova, L., Kiktev, N., Lendiel, T., Pavlov, S., & Mazurchuk, P. (2023). Computer-integrated control system for electrophysical methods of increasing plant productivity. Machinery & Energetics, 14(2), 34-45. doi: 10.31548/ machinery/2.2023.34.

[24] Khort, D.; Kutyrev, A.; Kiktev, N.; Hutsol, T.; Glowacki, S.; Kuboń, M.; Nurek, T.; Rud, A.; Gródek-Szostak, Z. Automated Mobile Hot Mist Generator: A Quest for Effectiveness in Fruit Horticulture. *Sensors* 2022, *22*, 3164. https://doi.org/10.3390/s22093164

7. Appendix

Program code Esp8266

```cpp
#include <ESP8266WiFi.h>
// Enter your Wi-Fi SSID and password
const char* ssid = "TP-Link_F43C";
const char* password = "80818458";


// Specify the GPIO pin for the LED (eg D4)
const int ledPin = D7;
const int ledPin_0 = D5;
// A variable to store the current brightness value
int brightness = 0;
WiFiServer server(80);


void setup() {
 Serial.begin(115200);
 delay(10);


 // Setting the foam for the LED
 pinMode(ledPin, OUTPUT);
   pinMode(ledPin_0, OUTPUT);
 analogWrite(ledPin, brightness);


 // Connecting to a Wi-Fi network
 Serial.println();
 Serial.print("Connecting to ");
 Serial.println(ssid);
 WiFi.begin(ssid, password);


 while (WiFi.status() != WL_CONNECTED) {
   delay(500);
   Serial.print(".");
 }
 Serial.println("");
 Serial.println("WiFi connected");
 Serial.println("IP address: ");
 Serial.println(WiFi.localIP());
 // Starting the web server
 server.begin();
```

```cpp
    Serial.println("HTTP server started");
  }
  void loop() {
   WiFiClient client = server.available();
   if (!client) {
     return;
   }
   Serial.println("New client");
   while (!client.available()) {
     delay(1);
   }
   String request = client.readStringUntil('\r');
   Serial.println(request);
   client.flush();
   // Checking requests to change the brightness of the LED
   if (request.indexOf("/slider") != -1) {
     brightness = request.substring(request.indexOf("=") + 1).toInt();
     brightness = constrain(brightness, 0, 1023); // limiting the
brightness value from 0 to 1023
      analogWrite(ledPin, brightness);
   }
  // Checking requests to control the LED
   if (request.indexOf("/LED=ON") != -1) {
     digitalWrite(ledPin_0, HIGH);
   }
   if (request.indexOf("/LED=OFF") != -1) {
     digitalWrite(ledPin_0, LOW);
   }
   if (request.indexOf("/getBrightness") != -1) {
    float sl = analogRead(A0);
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: application/json");
    client.println("");
    client.print("{\"value\": ");
    client.print(String(sl));
    client.println("}");
   }
   delay(1);
   Serial.println("Client disconnected");
   Serial.println("");
  }
```