

# ДОДАТОК А

## АПРОБАЦІЯ РЕЗУЛЬТАТІВ РОБОТИ

Міністерство освіти і науки України



### ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2024

(Випуск 2)

[електронне видання]



<http://nare.ua/departament/kafedra-komp-yuterno-integrovanih-tehnologiy-avtomatizatsiyi-ta-mehatroniki-kitap>



<http://itcz.znu.edu.ua/>



<http://kafedra.kdu.edu.ua>

Харків 2024

UDC 004.896

### OVERVIEW AT MODERN MINE DETECTING ROBOTS

**A. Karpenko**  
Kharkiv National University of Radio Electronics  
Ukraine, 61166, Kharkiv, pr. Nauki, 14  
E-mail: [andrii.karpenko2@nure.ua](mailto:andrii.karpenko2@nure.ua)

**Abstract:** In this article, main robotic solutions for mine detection and demining were reviewed using such criteria, as approaches to detection and demining, types of utility tools and mechanical parts. As result, the general view at types of mine detecting robots and distribution of their use case amounts was gathered

**Keywords:** BOZENA, explosives, GPR, demining, landmine, mine, mine detection, robot.

### ОГЛЯД СУЧАСНИХ МІННОПОШУКОВИХ РОБОТІВ

**А.С.Карпенко**  
Харківський національний університет радіоелектроніки  
Україна, 61166, Харків, пр. Науки 14  
E-mail: [andrii.karpenko2@nure.ua](mailto:andrii.karpenko2@nure.ua)

**Анотація:** В даній роботі розглянуто різновиди роботів для знаходження та знешкодження мін на основі таких критеріїв, як способи розпізнавання мін, їх знешкодження, види допоміжного обладнання, а також види механічних частин. В результаті, було отримано загальне уявлення про види міннопошукових роботів та розподіл кількості їх використання.

**Ключові слова:** BOZENA, вибухові речовини, радар глибокого дослідження, розмінування, міна, пошук мін, робот.

**INTRODUCTION.** According to [1], Ukraine is ranked second among the amount of injured by mine explosions. It occurred due to the huge stashes of anti-infantry devices Russian Federation possess in combination with its eagerness to use them without regards to universal practices. Furthermore, it is a common practice to see explosives even far from the places of main battles: explosives can be placed in the toys and left on the streets of Ukrainian cities by infiltrates. As a result, Ukrainians must put their time and efforts in lowering the risk of civilians to be exploded. The process of mine demolition consists of three steps [2]: detection, neutralization and utilization. Usually, an expert is called to perform each step, however it is anyway dangerous to get into the area with a trap for human: the most attentive can get caught. Moreover, mines can be placed on vast distances, such as corners of crop field, or places inconvenient to reach, like dense forest. In such cases, better to risk a carcass of a machine instead of human being.

But what should the machine be like to perform its task the best? One way is to analyze the existing alternatives to either spot the ways to improve them or find the inspiration for a new one.

**TYPES OF EXPLOSIVES.** To understand the requirements for demining robots, common explosives to be seen must be observed. According to figure 1, most widespread explosives are in general metal, different in sizes: from aviation bombs to handheld grenades. The proportion between size and explosive power also vary: the devastation from fragmentation mines is bigger in comparison with a grenade of the close sizes.

**SENSING.** Initially, landmine must be detected before the removal. This process can be done in several ways, depending on the type of the explosive to locate and features of surrounding terrain. The most common approaches are represented on figure 2.

«AUTOMATION AND DEVELOPMENT OF ELECTRONIC DEVICES»  
ADED-2024 Part 2.

181

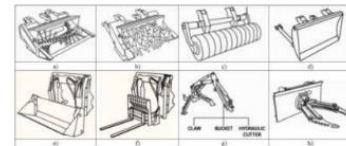


Figure 3 – Equipment of DOK-ING MV-4 demining system  
a) – flail; b) – tiller; c) – segmented roller; d) – dozer blade; e) – rear shovel; f) – rear forklift; g) – EOD arm; h) – rotational gripper [8]

According to [5], there are also other detection methods, however most of them are either analog to the mentioned systems (nuclear quadrupole resonance systems send radio waves to detect specifically  $^{14}N$  nuclei, X-ray backscatter systems send X-ray waves instead of radio ones, acoustic systems send out sound waves instead of electromagnetic, etc) or too specific to use in general cases (vapor sensors detect the mine only if a vapor of explosives gets to the sensor).

Despite sensing the landmine itself, robot should also get the data for movement planning. To do this, demining robots are usually equipped with cameras [5, 9] and LIDAR system.

**ACTUATION.** As it was shown on figure 3, demining robots can have parts of actuation, which are used either for landmine-based operations or area preparation activities. If robot is dedicated to demining in addition to mine detection, then it is usually equipped with flail or tiller for direct detonation (figure 3a-b) or manipulator (figure 3g) [10].

Other actuators are mainly used for terrain preparation: dozer blades are used for earth moving, rotational gripper – for obstacle removal, etc.

According to [11], most of demining robots are wheeled, either with tracks or without them (figure 4a)

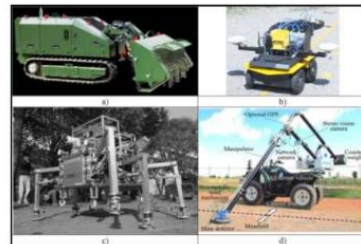


Figure 4 – Some types of demining robot structures  
a) – DOK-ING MV-2 [12]; b) – JACKAL by clearpath robotics [13]; c) COMET 1 [14]; d) – Gryphon [15]

Demining systems like Bozena [3] are most common to be seen among the robotic solutions [4] for demining services due to the efficiency of such machinery for demining vast open areas, such as crop fields. Such machinery possess tillers and flails (figure 3a-b), which both used to intentionally activate the mine, under the cover of hood. Despite their productivity, such robots cannot be used on non-flat areas or one with number of obstacles, like trees or buildings. Also, such systems are typically more expensive because of the required huge sizes and mass.

On the other hand, robots with mine detectors and ground penetrating radars (GRPs) can be both compact enough to pass in areas full of obstacles. Metal detector can find explosives with a help of (electro-)magnets, while GRPs send radio waves, that reflect only from the non-soil object, such as landmine. While metal detectors are usually enough to detect most types of explosives, GRPs can detect plastic one, which can be useful sometimes. Unfortunately, both systems are prone to false positive detections [5], which creates demand for alternative detection methods and advanced classification algorithms.



Figure 1 – Most common explosives, seen on the territory of Ukraine  
a) – mortar shells; b) – artillery shells; c) – hand grenades; d) – cluster munitions; e) – anti-tank remote control mines; f) – ballistic missiles; g) – shards of fragmentation grenades; h) – aircraft missiles; i) – anti-tank guided missiles; j) – jet missiles; k) – anti-personnel fragmentation mines; l) – anti-personnel landmines; m) – trap mines; n) – high explosive anti-tank mines; o) – aircraft bombs [6]

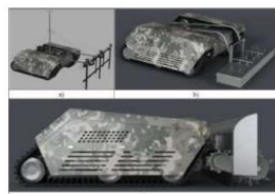


Figure 2 – Main types of landmine detection [7]  
a) – radiolocation; b) – magnetism; c) – detonation

«AUTOMATION AND DEVELOPMENT OF ELECTRONIC DEVICES»  
ADED-2024 Part 2.

182

«AUTOMATION AND DEVELOPMENT OF ELECTRONIC DEVICES»  
ADED-2024 Part 2.

183

Some other types of robots exist, but they either have significant drawbacks, that do not allow for them to be used on daily base (COMETs are huge and heavy), or just were not polished enough in comparison with competitors.

**CONCLUSION.** In the scientific realm, there are numbers of solutions for mine detection and disposal, however practically only some solutions are applied on a regular basis. Such robots are usually track wheeled devices, either of human height sizes, armed with direct detonation devices and utility tools for landscape modifications, or of compact dimensions, armed with mine detector like magnetic one or GRP, and sometimes with manipulator for demining. Even though other structures exist, they are implied seldom, if not only on the territory of the research field.

However, the first sappers were unlikely to imagine the ways their instruments would develop in decades, and so we can not to imagine completely the demining devices of future. Thus, new land clearance solutions should be tried, notwithstanding the amount and types of solutions, widespread nowadays.

#### REFERENCES

1. Протипіхотні міни: скільки жертв в Україні та світі. *Слово і Діло*. URL: <https://www.slovoidilo.ua/2023/11/16/infografika/suspilstvo/protypixotni-miny-skilky-zhertv-ukrayini-ta-sviti> (date of access: 17.11.2024).
2. Українські сервіси розмінування. *Українські сервіси розмінування*. URL: <https://uds.pro/> (date of access: 17.11.2024).
3. Нестехнічне обстеження | UDS. *Українські сервіси розмінування*. URL: <https://uds.pro/home/nto-1-1-1> (date of access: 20.11.2024).
4. Робот для розмінування | міношукачі україна. *Міношукачі Україна*. URL: <https://minoshukach.com.ua/wiki/tehnika-dlya-rozminuvannya/robot-dlya-rozminuvannya/> (date of access: 20.11.2024).
5. A review on design and characteristics of landmine detection robot / Vikas Singh Panwar та ін. *Evergreen*. 2024. Т. 11, № 2. С. 900–912. URL: <https://doi.org/10.5109/7183373> (date of access: 20.11.2024).
6. Розмінування України. *Розмінування України*. URL: <https://mine.dns.gov.ua/> (дата звернення: 19.11.2024).
7. TrackReitar CleanField - LeoTronics Robotics. *Autonomous mobile robots - LeoTronics Robotics*. URL: <https://leotronics.eu/en/tracked-robots/trackreitar-cleanfield> (date of access: 19.11.2024).
8. DOK-ING MV-4 | міношукачі україна. *Міношукачі Україна*. URL: <https://minoshukach.com.ua/wiki/tehnika-dlya-rozminuvannya/dok-ing/dok-ing-mv-4/> (date of access: 20.11.2024).
9. Robot model for mines searching development / O. Kuzmenko et al. *The multidisciplinary journal of science and technology*. 2024. Vol. 4, no. 6. P. 347–355. Ukinform.
10. Роботизовані комплекси допомагають саперам ДСНС розмінувати деокуповану територію, 2022. *YouTube*. URL: <https://www.youtube.com/watch?v=uDdui5T19-8> (date of access: 21.11.2024).
11. Неаглодов І. Ш., Андрусевич А. О., Свєтєв В. В., Новоселов С. П., Демська Н. П. Проективання мобільних маніпуляційних роботів: Монографія. – Х.: 2022. – 427 с.
12. DOK-ING MV-2 | міношукачі україна. *Міношукачі Україна*. URL: <https://minoshukach.com.ua/wiki/tehnika-dlya-rozminuvannya/dok-ing/dok-ing-mv-2/> (date of access: 22.11.2024).

«AUTOMATION AND DEVELOPMENT OF ELECTRONIC DEVICES»  
ADED-2024 Part 2.

184

«AUTOMATION AND DEVELOPMENT OF ELECTRONIC DEVICES»  
ADED-2024 Part 2.

185

UDC 004.89

#### DESIGN OF MINE-DETECTING ROBOT USING YOLOv8 OBJECT DETECTION MODEL

**A. Karpenko**  
Kharkiv National University of Radio Electronics  
Ukraine, 61166, Kharkiv, pr. Nauki, 14  
E-mail: [andrii.karpenko2@nure.ua](mailto:andrii.karpenko2@nure.ua)

**Abstract:** Methods of developing a demining robot, based on Android application control and YOLOv8 image detection algorithm, are uncovered in this article.

**Keywords:** computer vision, demining, explosives, landmine, mine, object detection, robot, robotics, Arduino, MIT App Inventor, YOLOv8.

#### РОЗРОБКА МІННОПОШУКОВОГО РОБОТА З ВИКОРИСТАННЯМ СИСТЕМИ РОЗПІЗНАВАННЯ ОБРАЗІВ МОДЕЛІ YOLOv8

**A.С.Карпенко**  
Харківський національний університет радіоелектроніки  
Україна, 61166, Харків, пр. Науки 14  
E-mail: [andrii.karpenko2@nure.ua](mailto:andrii.karpenko2@nure.ua)

**Анотація:** В даній статті описано спосіб побудови мінопошукового робота з системою комп'ютерного зору на базі моделі нейронної мережі YOLOv8 та системою керування від Android-додатку.

**Ключові слова:** комп'ютерний зір, розмінування, вибухівка, вибухонебезпечний предмет, міна, розпізнавання об'єктів, робот, робототехніка, Arduino, MIT App Inventor, YOLOv8.

**INTRODUCTION.** Previously [1], the significance of modern demining robot development, as well as demining solutions were mentioned. In short, due to violations of human rights, nowadays it is common to find landmines or other explosives on certain areas. This issue is especially important in Ukraine: country, that became the most mine-flooded in world due to russian military intervention. As a result, it is of great importance to clean up the landscape from explosives.

From the perspective of safety, robots are well suited for demining tasks: better to lose a machine rather than human life. However, robotic solutions have their own disadvantages, such as limited mobility and huge sizes. In this way, it is still possible to develop a better alternative to existing robotic solutions.

**HARDWARE SELECTION.** The main goal of robot is both move on a flat terrain and be capable to detect the explosives, and so the hardware selection was the main priority during development of new robot. In this way, the structure of robot was planned and developed according to figure 1.

The project does not involve sophisticated kinematics, like legged movement, precise allocation and other, so instead of building the carcasses and choosing the electronic parts from scratch, existing solution in a form of one of TSCINBUNY robots [2].

The schematic includes Arduino-like controller (figure 1.a), that can accept the 7.4V charge and reallocate it to servos (figure 1.c) via a specific shield (figure 1.b). The movement controls are transported via a Bluetooth module (figure 1.d), while the image from ESP-32CAM camera (figure 1.e) is sent to the laptop via webpage. Even though camera is physically connected to the controller, the only type of their communication is a set of messages for launching the camera during the launch of robot.

«AUTOMATION AND DEVELOPMENT OF ELECTRONIC DEVICES»  
ADED-2025 Part 1.

62

Міністерство освіти і науки України



**NURE**

Харківський національний університет  
радіоелектроніки

#### ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2025

(Випуск 1)

[електронне видання]



<http://nure.ua/department/kafedra-komp-yuterno-integrovanih-technologies-avtomatizatsiyi-ta-mehatroniki-kitap>



<http://iteez.zntu.edu.ua/>

Харків 2025

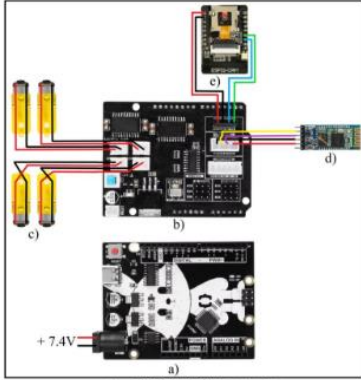


<http://kafedra.kdu.edu.ua>

**SOFTWARE DEVELOPMENT.** In this way, there are two subsystems to be programmed: movement and object detection ones.

To move the developed machine, an operator should have sent such a message to the Bluetooth module, that would correspond to one of the movement instructions. Furthermore, it is not enough to send only one message. This process is represented on figure 2: there, blue color represents the events in physical world, red one represents instructions, acquired by robot (M for movement, S for stop, Noth for nothing, when connection is lost), black cyclogram represents commands, executed by robot.

In similar projects [3,4,5], usually the movement of robot changes during the touching button up and down (figure 2.a). It is convenient for general projects because of simplicity in code, however it does not suit military-related applications due to lack of noise protection [6,7].



a) – TSCINBUNY controller;  
b) – McNamham wheel manipulator trolley shield;  
c) – gear motors;  
d) – HC-08 Bluetooth sensor;  
e) ESP32-CAM.

Figure 1 – Connection schematics of the developed robot

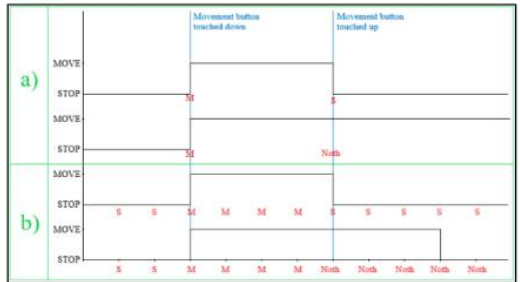


Figure 2 – Cyclograms for instruction execution for different robotic movement algorithms

If robot does not get the touch-up instruction (“stop”), then it will keep moving. Moreover, such behavior is unpreventable due to impossibility to distinguish the “Noth” instruction from user inactivity.

Thus, it is more reliable to perform instructions according to figure 2.b, where certain instruction is constantly sent on controller once in a particular time interim. Here, if no instructions get to the controller, then timer can be launched to make robot stop after certain time.

According to algorithm mentioned above, an Android application was developed to send instructions to controller (figure 3).



a) – layout; b) – code

Figure 3 – Android application for movement control

After getting the message via a virtual serial port, robot must process acquired instructions, which is done according to method, mentioned in [3]. Also, the movement stop after connection lost was considered. As a result, the code from figure 4 was uploaded into controller.

```
void setup() {
  Serial.begin(115200);
  pinMode(13, OUTPUT);
  pinMode(14, OUTPUT);
  pinMode(15, OUTPUT);
  pinMode(16, OUTPUT);
  pinMode(17, OUTPUT);
  pinMode(18, OUTPUT);
  pinMode(19, OUTPUT);
  pinMode(20, OUTPUT);
  pinMode(21, OUTPUT);
  pinMode(22, OUTPUT);
  pinMode(23, OUTPUT);
  pinMode(24, OUTPUT);
  pinMode(25, OUTPUT);
  pinMode(26, OUTPUT);
  pinMode(27, OUTPUT);
  pinMode(28, OUTPUT);
  pinMode(29, OUTPUT);
  pinMode(30, OUTPUT);
  pinMode(31, OUTPUT);
  pinMode(32, OUTPUT);
  pinMode(33, OUTPUT);
  pinMode(34, OUTPUT);
  pinMode(35, OUTPUT);
  pinMode(36, OUTPUT);
  pinMode(37, OUTPUT);
  pinMode(38, OUTPUT);
  pinMode(39, OUTPUT);
  pinMode(40, OUTPUT);
  pinMode(41, OUTPUT);
  pinMode(42, OUTPUT);
  pinMode(43, OUTPUT);
  pinMode(44, OUTPUT);
  pinMode(45, OUTPUT);
  pinMode(46, OUTPUT);
  pinMode(47, OUTPUT);
  pinMode(48, OUTPUT);
  pinMode(49, OUTPUT);
  pinMode(50, OUTPUT);
  pinMode(51, OUTPUT);
  pinMode(52, OUTPUT);
  pinMode(53, OUTPUT);
  pinMode(54, OUTPUT);
  pinMode(55, OUTPUT);
  pinMode(56, OUTPUT);
  pinMode(57, OUTPUT);
  pinMode(58, OUTPUT);
  pinMode(59, OUTPUT);
  pinMode(60, OUTPUT);
  pinMode(61, OUTPUT);
  pinMode(62, OUTPUT);
  pinMode(63, OUTPUT);
  pinMode(64, OUTPUT);
  pinMode(65, OUTPUT);
  pinMode(66, OUTPUT);
  pinMode(67, OUTPUT);
  pinMode(68, OUTPUT);
  pinMode(69, OUTPUT);
  pinMode(70, OUTPUT);
  pinMode(71, OUTPUT);
  pinMode(72, OUTPUT);
  pinMode(73, OUTPUT);
  pinMode(74, OUTPUT);
  pinMode(75, OUTPUT);
  pinMode(76, OUTPUT);
  pinMode(77, OUTPUT);
  pinMode(78, OUTPUT);
  pinMode(79, OUTPUT);
  pinMode(80, OUTPUT);
  pinMode(81, OUTPUT);
  pinMode(82, OUTPUT);
  pinMode(83, OUTPUT);
  pinMode(84, OUTPUT);
  pinMode(85, OUTPUT);
  pinMode(86, OUTPUT);
  pinMode(87, OUTPUT);
  pinMode(88, OUTPUT);
  pinMode(89, OUTPUT);
  pinMode(90, OUTPUT);
  pinMode(91, OUTPUT);
  pinMode(92, OUTPUT);
  pinMode(93, OUTPUT);
  pinMode(94, OUTPUT);
  pinMode(95, OUTPUT);
  pinMode(96, OUTPUT);
  pinMode(97, OUTPUT);
  pinMode(98, OUTPUT);
  pinMode(99, OUTPUT);
}
```

Figure 4 – Controller code for movement of robot, page 1

```
void loop() {
  if (Serial.available() > 0) {
    char c = Serial.read();
    if (c == 'M') {
      digitalWrite(13, HIGH);
      digitalWrite(14, HIGH);
      digitalWrite(15, HIGH);
      digitalWrite(16, HIGH);
      digitalWrite(17, HIGH);
      digitalWrite(18, HIGH);
      digitalWrite(19, HIGH);
      digitalWrite(20, HIGH);
      digitalWrite(21, HIGH);
      digitalWrite(22, HIGH);
      digitalWrite(23, HIGH);
      digitalWrite(24, HIGH);
      digitalWrite(25, HIGH);
      digitalWrite(26, HIGH);
      digitalWrite(27, HIGH);
      digitalWrite(28, HIGH);
      digitalWrite(29, HIGH);
      digitalWrite(30, HIGH);
      digitalWrite(31, HIGH);
      digitalWrite(32, HIGH);
      digitalWrite(33, HIGH);
      digitalWrite(34, HIGH);
      digitalWrite(35, HIGH);
      digitalWrite(36, HIGH);
      digitalWrite(37, HIGH);
      digitalWrite(38, HIGH);
      digitalWrite(39, HIGH);
      digitalWrite(40, HIGH);
      digitalWrite(41, HIGH);
      digitalWrite(42, HIGH);
      digitalWrite(43, HIGH);
      digitalWrite(44, HIGH);
      digitalWrite(45, HIGH);
      digitalWrite(46, HIGH);
      digitalWrite(47, HIGH);
      digitalWrite(48, HIGH);
      digitalWrite(49, HIGH);
      digitalWrite(50, HIGH);
      digitalWrite(51, HIGH);
      digitalWrite(52, HIGH);
      digitalWrite(53, HIGH);
      digitalWrite(54, HIGH);
      digitalWrite(55, HIGH);
      digitalWrite(56, HIGH);
      digitalWrite(57, HIGH);
      digitalWrite(58, HIGH);
      digitalWrite(59, HIGH);
      digitalWrite(60, HIGH);
      digitalWrite(61, HIGH);
      digitalWrite(62, HIGH);
      digitalWrite(63, HIGH);
      digitalWrite(64, HIGH);
      digitalWrite(65, HIGH);
      digitalWrite(66, HIGH);
      digitalWrite(67, HIGH);
      digitalWrite(68, HIGH);
      digitalWrite(69, HIGH);
      digitalWrite(70, HIGH);
      digitalWrite(71, HIGH);
      digitalWrite(72, HIGH);
      digitalWrite(73, HIGH);
      digitalWrite(74, HIGH);
      digitalWrite(75, HIGH);
      digitalWrite(76, HIGH);
      digitalWrite(77, HIGH);
      digitalWrite(78, HIGH);
      digitalWrite(79, HIGH);
      digitalWrite(80, HIGH);
      digitalWrite(81, HIGH);
      digitalWrite(82, HIGH);
      digitalWrite(83, HIGH);
      digitalWrite(84, HIGH);
      digitalWrite(85, HIGH);
      digitalWrite(86, HIGH);
      digitalWrite(87, HIGH);
      digitalWrite(88, HIGH);
      digitalWrite(89, HIGH);
      digitalWrite(90, HIGH);
      digitalWrite(91, HIGH);
      digitalWrite(92, HIGH);
      digitalWrite(93, HIGH);
      digitalWrite(94, HIGH);
      digitalWrite(95, HIGH);
      digitalWrite(96, HIGH);
      digitalWrite(97, HIGH);
      digitalWrite(98, HIGH);
      digitalWrite(99, HIGH);
    } else if (c == 'S') {
      digitalWrite(13, LOW);
      digitalWrite(14, LOW);
      digitalWrite(15, LOW);
      digitalWrite(16, LOW);
      digitalWrite(17, LOW);
      digitalWrite(18, LOW);
      digitalWrite(19, LOW);
      digitalWrite(20, LOW);
      digitalWrite(21, LOW);
      digitalWrite(22, LOW);
      digitalWrite(23, LOW);
      digitalWrite(24, LOW);
      digitalWrite(25, LOW);
      digitalWrite(26, LOW);
      digitalWrite(27, LOW);
      digitalWrite(28, LOW);
      digitalWrite(29, LOW);
      digitalWrite(30, LOW);
      digitalWrite(31, LOW);
      digitalWrite(32, LOW);
      digitalWrite(33, LOW);
      digitalWrite(34, LOW);
      digitalWrite(35, LOW);
      digitalWrite(36, LOW);
      digitalWrite(37, LOW);
      digitalWrite(38, LOW);
      digitalWrite(39, LOW);
      digitalWrite(40, LOW);
      digitalWrite(41, LOW);
      digitalWrite(42, LOW);
      digitalWrite(43, LOW);
      digitalWrite(44, LOW);
      digitalWrite(45, LOW);
      digitalWrite(46, LOW);
      digitalWrite(47, LOW);
      digitalWrite(48, LOW);
      digitalWrite(49, LOW);
      digitalWrite(50, LOW);
      digitalWrite(51, LOW);
      digitalWrite(52, LOW);
      digitalWrite(53, LOW);
      digitalWrite(54, LOW);
      digitalWrite(55, LOW);
      digitalWrite(56, LOW);
      digitalWrite(57, LOW);
      digitalWrite(58, LOW);
      digitalWrite(59, LOW);
      digitalWrite(60, LOW);
      digitalWrite(61, LOW);
      digitalWrite(62, LOW);
      digitalWrite(63, LOW);
      digitalWrite(64, LOW);
      digitalWrite(65, LOW);
      digitalWrite(66, LOW);
      digitalWrite(67, LOW);
      digitalWrite(68, LOW);
      digitalWrite(69, LOW);
      digitalWrite(70, LOW);
      digitalWrite(71, LOW);
      digitalWrite(72, LOW);
      digitalWrite(73, LOW);
      digitalWrite(74, LOW);
      digitalWrite(75, LOW);
      digitalWrite(76, LOW);
      digitalWrite(77, LOW);
      digitalWrite(78, LOW);
      digitalWrite(79, LOW);
      digitalWrite(80, LOW);
      digitalWrite(81, LOW);
      digitalWrite(82, LOW);
      digitalWrite(83, LOW);
      digitalWrite(84, LOW);
      digitalWrite(85, LOW);
      digitalWrite(86, LOW);
      digitalWrite(87, LOW);
      digitalWrite(88, LOW);
      digitalWrite(89, LOW);
      digitalWrite(90, LOW);
      digitalWrite(91, LOW);
      digitalWrite(92, LOW);
      digitalWrite(93, LOW);
      digitalWrite(94, LOW);
      digitalWrite(95, LOW);
      digitalWrite(96, LOW);
      digitalWrite(97, LOW);
      digitalWrite(98, LOW);
      digitalWrite(99, LOW);
    } else if (c == 'N') {
      // Do nothing
    }
  }
}
```

Figure 4 – Controller code for movement of robot, page 2

At the same time, robot must also transmit the video to computer for object detection tasks. To do it, a code from [8] was applied to transfer the image from ESP32-CAM to the local webpage. Then, the code was processed in PyCharm, where the image is processed with a pretrained YOLOv8 model.

To perform an object detection, a neural net must be trained and infused in a system [9,10]. This involves preparing training, cross validation and test data, followed by training processes itself and ending up with passing the image through image detection application. The dataset of explosives was taken from [11]. Then, model was trained in Google Collab according to [12]. Lastly, the video from web page was extracted according to [3], processed and displayed (figure 5 and 6).

```
import cv2
import urllib.request
import numpy as np
from ultralytics import YOLO
import supervision as sv

# Replace the URL with the IP camera's stream URL
url = "http://192.168.0.121:8181"
cv2.namedWindow("YOLOv8 Live Cam Testing", cv2.WINDOW_KEEPRATIO)

# Create a VideoCapture object
cap = cv2.VideoCapture(url)

# Check if the IP camera stream is opened successfully
if not cap.isOpened():
    print("Failed to open the IP camera stream")
    exit()

model = YOLO("C:/Users/ADMIN/AppData/Local/Programs/Python/Python311/Scripts/yolo11n.pt")
box_annotator = sv.BoxAnnotator()

threshold = 0.3

# Read and display video frames
while True:
    # Read a frame from the video stream
    img_resp = urllib.request.urlopen(url)
    imgnp = np.array(bytearray(img_resp.read()), dtype=np.uint8)
    im = cv2.imdecode(imgnp, -1)

    results = model(im)[0]

    for result in results.boxes.data.tolist():
        x1, y1, x2, y2, score, class_id = result

        if score > threshold:
            cv2.rectangle(im, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255, 0), 2)
            cv2.putText(im, f'{results.names[int(class_id)]} {score:.2f}', (int(x1), int(y1 - 10)),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 1)

    cv2.imshow("YOLOv8 Live Cam Testing", im)
    key = cv2.waitKey(1)
    if key == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Figure 5 – Python code for processing and displaying image



Figure 6 – Results of detection

**CONCLUSION.** Even though neural network successfully detects most of explosives, there are a lot of false negatives in the detections. Also, the developed robot lacks battery monitoring system and chassis for dust protection, so there are still ways of developing the project.

#### REFERENCES

1. Karpenko A. Overview at modern mine detecting robots. *Aedet-2024*. 2024. No. 2. P. 181–185. URL: [https://drive.google.com/file/d/1u\\_rEiXZx-TIR7n1sBhKl\\_OuFF5w7KOb1/view](https://drive.google.com/file/d/1u_rEiXZx-TIR7n1sBhKl_OuFF5w7KOb1/view) (date of access: 06.04.2025).
2. ESP32 camera robot for arduino uno starter kit *TSCINBUNY*. URL: <https://tscinbuny.com/products/tscinbuny-esp32-robot-for-arduino-uno-starter-kit-programmable-robot-educational-kit-4wd-60mm-omni-directional-wheel-chassis-with-wifi-app-obstacle-avoidance-line-tracking-smart-car-set> (date of access: 06.04.2025).
3. Viral Science - The home of Creativity. Arduino mecanum wheel robot car | all direction controlled, 2021. *YouTube*. URL: <https://www.youtube.com/watch?v=SYfCyF44h0> (date of access: 06.04.2025).
4. Srishti Robotics. Bluetooth controlled robot using arduino uno and MIT APP inventor, 2021. *YouTube*. URL: <https://www.youtube.com/watch?v=QC6Tlduhfg> (date of access: 06.04.2025).
5. Muhammad Ansar. Arduino Robot Car Control using HC-05 Bluetooth | mit app inventor for android apk, 2019. *YouTube*. URL: <https://www.youtube.com/watch?v=HBQIH98wmHs> (date of access: 06.04.2025).
6. Неялодов І. Ш., Андрусевич А. О., Сясеєв В. В., Новоселов С. П., Демська Н. П. Проектування мобільних маніпуляційних роботів: Монографія. – X. :, 2022. – 427 с.

7. Nevludov I., Tsybal O., Bronnikov A. Fuzzy decision-making for intelligent robotic system. *Optoelectronic devices in robotic systems*. Cham, 2022. P. 227–255. URL: [https://doi.org/10.1007/978-3-031-09791-1\\_9](https://doi.org/10.1007/978-3-031-09791-1_9) (date of access: 07.04.2025).
8. ESP32 CAM object detection & identification with opencv. *How To Electronics*. URL: <https://how2electronics.com/esp32-cam-based-object-detection-identification-with-opencv/> (date of access: 07.04.2025).
9. Queries classification using machine learning for implementation in intelligent manufacturing / V. Bortnikova et al. *Methods and tools in cad selected issues* / ed. by B. Butrylo. Bialystok, 2021. P. 63–74.
10. Yevsieiev V., Maksymova S., Alkhalaleh A. Mobilenetv2 neural network model for human recognition and identification in the working area of a collaborative robot. *Multidisciplinary journal of science and technology*. 2024. Vol. 4, no. 8. P. 5–12. URL: <https://www.mjstjournal.com/index.php/mjst/article/view/1783> (date of access: 07.04.2025).
11. Demining object detection dataset by trashsorting. *Roboflow*. URL: <https://universe.roboflow.com/trashsorting-gg1jy/demining-rpd9q> (date of access: 07.04.2025).
12. Computer vision engineer. Train Yolov8 object detection on a custom dataset | Step by step guide | Computer vision tutorial, 2023. *YouTube*. URL: <https://www.youtube.com/watch?v=m9H90Wn8YM> (date of access: 07.04.2025).
13. Chala, O., Yevsieiev, V., Maksymova, S., & Abu-Jassar, A. (2025). MATHEMATICAL MODEL BASED ON MULTI-AGENT REINFORCEMENT LEARNING (MARL) AND PARTIALLY OBSERVABLE MARKOV DECISION PROCESS (POMDP) FOR MODELING CARGO MOVEMENT FOR A MOBILE ROBOTS GROUP. *Multidisciplinary Journal of Science and Technology*, 5(4), 480–489.
14. Maksymova, S., Yevsieiev, V., & Abu-Jassar, A. (2025). MICROCHIP MARKING RECOGNITION AND IDENTIFICATION USING A COMPUTER VISION SYSTEM MATHEMATICAL MODEL. *Multidisciplinary Journal of Science and Technology*, 5(4), 321–330.
15. Chala, O., Yevsieiev, V., Maksymova, S., & Abu-Jassar, A. (2025). USING THE HUMAN FACE RECOGNITION METHOD BASED ON THE MOBILENETV2 NEURAL NETWORK IN AUTHENTICATION SYSTEMS. *Multidisciplinary Journal of Science and Technology*, 5(3), 882–895.
16. Неялодов, І. Ш., Сясеєв, В. В., & Гурін, Д. В. (2025). MODEL DEVELOPMENT OF DYNAMIC REPRESENTATION A MODEL DESCRIPTION PARAMETERS FOR THE ENVIRONMENT OF A COLLABORATIVE ROBOT MANIPULATOR WITHIN THE INDUSTRY 5.0 FRAMEWORK. *Системи управління, навігації та зв'язку. Збірник наукових праць*, 1(79), 42–48.
17. Abu-Jassar, A. T., Attar, H., Amer, A., Lyashenko, V., Yevsieiev, V., & Solyman, A. (2025). Development and Investigation of Vision System for a Small-Sized Mobile Humanoid Robot in a Smart Environment. *International Journal of Crowd Science*, 9(1), 29–43.
18. Yevsieiev, V., Maksymova, S., Alkhalaleh, A., & Gurin, D. (2025). Development of a program for processing 3d models of objects in a collaborative robot workspace using an HD camera. *ACUMEN: International journal of multidisciplinary research*, 2(1), 194–210.
19. Yevsieiev, V., Abu-Jassar, A., Maksymova, S., & Demaska, N. (2025). Development of a model for recognizing various objects and tools in a collaborative robot workspace. *ACUMEN: International journal of multidisciplinary research*, 2(1), 224–239.

**Scientific adviser:** Vladyslav Yevsieiev, professor of the Department of Computer-Integrated Technologies, Automation and Mechatronics, Kharkiv National University of Radio Electronics.

**ДОДАТОК Б**  
**ХАРАКТЕРИСТИКИ АПАРАТНИХ МОДУЛІВ**

Таблиця Б.1 – Характеристики камер

Камера	Характеристика					
	Якість зображення	Роздільна здатність	Кут огляду	Безпроводна передача даних (за наявності)	Ціна	Вимоги до моделі контролера
Raspberry Pi Camera Board (module 2)	8 мегапікселів	3280 × 2464	62.2° гор.; 48.8° верт.;	–	1629 грн	Raspberry Pi
Модуль ESP32-САМ	2 мегапікселі	1600 × 1200	66° гор.;	Wi-Fi/Bluetooth	400 грн	–
Arducam VO393	8 мегапікселів	3280 × 2464	65° гор.; 51° верт.; 77.6° діаг.;	–	1607 грн	Raspberry Pi
M5Stack ESP32 Timer Camera X	0.3 мегапікселя	2048 × 1536	66.5° гор.;	Wi-Fi	1097 грн	–
XIAO ESP32S3 sense seed studio	0.3 мегапікселя	1600 × 1200	66.5° гор.;	Wi-Fi/Bluetooth	1690 грн	–

Таблиця Б.2 – Характеристики електродвигунів

Двигун постійного струму	Характеристики							
	Крутний момент	Напруга живлення	Струм живлення	Швидкість (обертів/хв)	Діаметр валу	Передатне число редуктора (за наявності)	Маса (г)	Ціна (грн)
GA12-N20	0,2 кг/см	6В	0,1..0,16А	22..30	3 мм	—	10	167
				38..50				147
				76..100				149
				145..200				149
				220..300				149
				380..500				163
				440..600				155
				760..1000				155
DG01D	0,8 кг/см	6В	0,18..0,5А	140..230	5,5 мм	1:48	26	46
DC мотор 6В 162	1,2 кг/см	6В	0,05..0,6А	5800..11800	4 мм	—	68	84
RS133C- 10300-45	0,0033 кг/см	3В	0,04..0,12А	2930..4200	2 мм	—		32

Таблиця Б.3 – Характеристики Bluetooth датчиків

Датчик	Характеристики					
	Версія протоколу Bluetooth	Радіус охоплення	Швидкість передачі даних	Робоча частота	Напруга живлення	Ціна (грн)
НС-05	2.0	10 м	3 Мбіт/с	2402..2480 МГц	3,3 В	170
НС-06	2.0	10 м	3 Мбіт/с	2400 МГц	3,6..6В	160
НС-08	4.0	80 м	3 Мбіт/с	2400 МГц	3,3 В	263

Таблиця Б.4 – Характеристики серводвигунів

Серводвигун	Крутний момент (кг/см)	Напруга живлення (В)	Швидкість (секунд/60°)	Маса (г)	Ціна (грн)
MG90S V2	2,0..2,4	4,8..6	0,08..0,1	12	94
SG90	2	3,5..5	0,12	9	79
EMAX ES08A II	1,5..1,8	4,8..6,0	0,1..0,12	8.5	296
DS3218MG	18..21	4,8..6,8	0,14..0,16	68	628

Таблиця Б.5 – Характеристики плат розробки

Плата розробки	Характеристики					
	Мікро-контролер	Напруга живлення	Входи/виходи	Флеш-пам'ять програм	Оперативна пам'ять	Тактова частота
Arduino UNO R3	ATmega328	5В	Цифрові: 14 (з них 6 ШІМ) Аналогові входи: 6	32 Кб	2 Кб	16 МГц
Arduino UNO R4 Minima	Renesas RA4M1	5В	Цифрові: 14 (з них 6 ШІМ) Аналогові входи: 6	256 Кб	32 Кб	48 МГц
Arduino Mega 2560 R3 (CH340)	ATmega256 0	5В	Цифрові: 54 (з них 17 ШІМ) Аналогові входи: 16	256 Кб	8 Кб	16 МГц

## ДОДАТОК В

### ПОРІВНЯННЯ МОДЕЛЕЙ НЕЙРОННИХ МЕРЕЖ

Таблиця В.1 – Порівняння моделей нейронних мереж

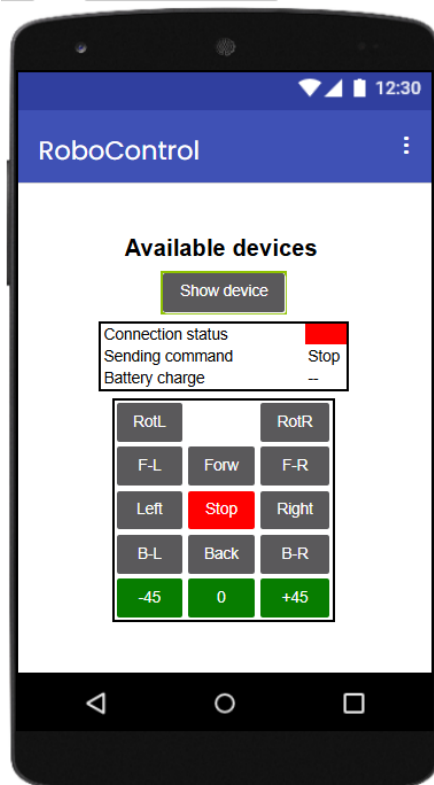
Модель мережі	Характеристики					
	Реального часу?	mAP <sub>50-90</sub> (COCO)	Пояснення структури	Наявність пакету для розгортання моделі	Ліцензія	Розмір моделі (Мб)
1	2	3	4	5	6	7
YOLOv8	Так	37.3 – 53.9	Ні	Так	AGPL-3.0	3.2 – 68.2
YOLOv7	Так	51.4 – 56.8	Так	Ні	GPL-3.0	36.9 – 151.7
YOLOv6	Так	37.5 – 57.2	Так	Ні	GPL-3.0	4.7 – 140.4
RMDet	Так	41.0 – 52.8	Так	Так	Apache 2.0	4.8 – 94.9
RT-DETR	Так	46.5 – 54.8	Так	Так	Apache 2.0	20.0 – 67.0
DETA	Ні	63.5	Так	Так	Apache 2.0	47
GroundingDINO	Так	52.5	Так	Так	Apache 2.0	172
Detectron2	Ні	44	Так	Так	Apache 2.0	200
EfficientDet	Так	52	Так	Так	Apache 2.0	17 – 50

Продовження таблиці В.1

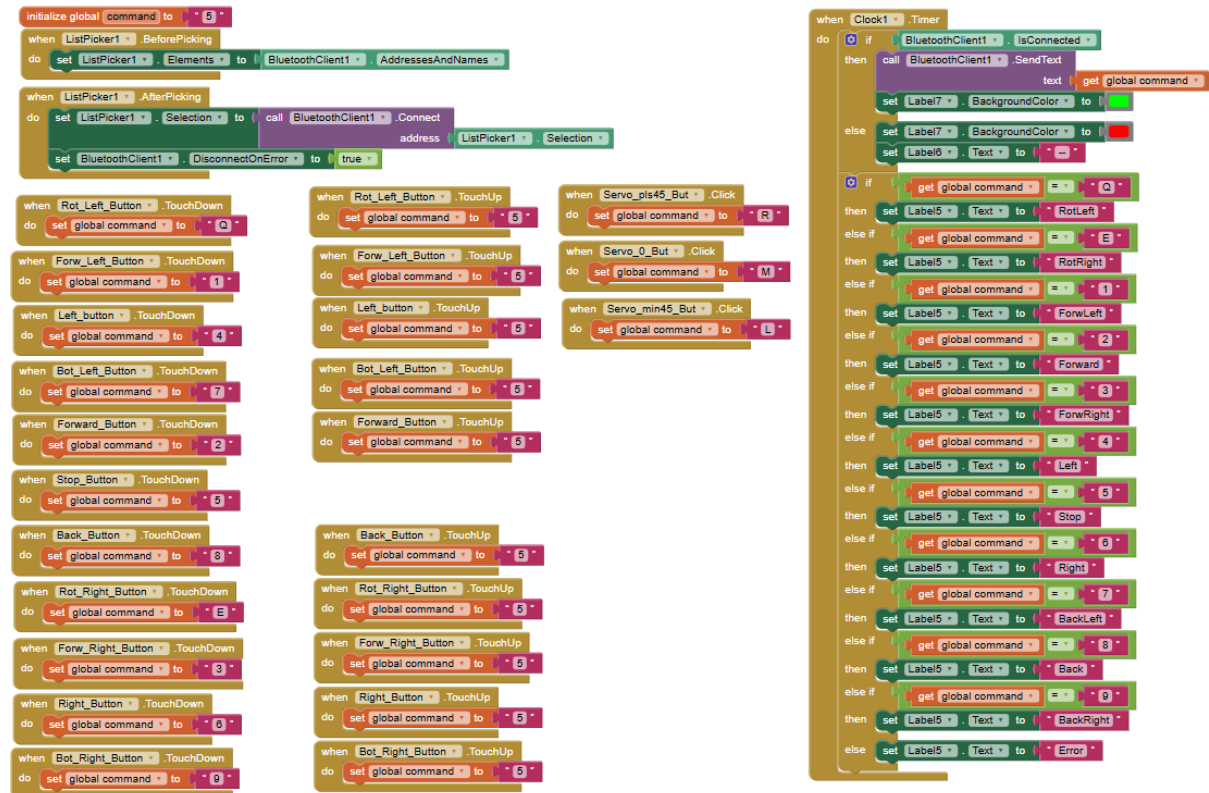
1	2	3	4	5	6	7
Single Shot Multibox detector	Так	43	Так	Так	Apache 2.0	100
Faster R-CNN	Hi	42	Так	Так	MIT	200

## ДОДАТОК Г

## ЛІСТИНГ ПРОГРАМИ КЕРУВАННЯ РУХОМ МОБІЛЬНОГО РОБОТА



a)



б)

а) макет інтерфейсу користувача; б) код роботи додатку

Рисунок Г.1 – Код мобільного додатку панелі керування мобільного робота

## ДОДАТОК Д ЛІСТИНГ

### ПРОГРАМИ КЕРУВАННЯ РУХОМ МОБІЛЬНОГО РОБОТА

```

1  #include<SoftwareSerial.h>
2  #include"SafeString.h"
3  #include<Servo.h>
4  // Контакти для керування ШІМ (до кожного драйвера)
5  #define PWM1_PIN      5
6  #define PWM2_PIN      6
7  // Контакти регістра зсуву 74НСТ595N
8  #define SHCP_PIN      2           // Таймер зчитування даних (із зсувом)
9  #define EN_PIN        7           // Ввімкнення/вимкнення передачі даних
10 #define DATA_PIN     8           // Дані для зчитування
11 #define STCP_PIN      4           // Таймер виведення даних
12
13 const int LeftUpBack   = 128;     // 10 00 00 00 Верхнє лїве колесо назад
14 const int LeftUpForw  = 64;      // 01 00 00 00 Верхнє лїве колесо вперед
15 const int LeftDownForw = 32;     // 00 10 00 00 Нижнє лїве колесо вперед
16 const int LeftDownBack = 16;    // 00 01 00 00 Нижнє лїве колесо назад
17 const int RightDownBack = 8;    // 00 00 10 00 Нижнє праве колесо назад
18 const int RightDownForw = 1;    // 00 00 00 01 Нижнє праве колесо вперед
19 const int RightUpForw  = 4;     // 00 00 01 00 Верхнє лїве колесо вперед
20 const int RightUpBack  = 2;     // 00 00 00 10 Верхнє лїве колесо назад
21
22
23 const int Forward     = LeftUpForw | LeftDownForw | RightDownForw | RightUpForw; // 01 10 01 01 Рух вперед
24 const int Backward    = LeftUpBack | LeftDownBack | RightDownBack | RightUpBack; // 10 01 10 10 Рух назад
25 const int Turn_Left   = LeftUpBack | LeftDownForw | RightUpForw | RightDownBack; // 10 10 11 00 Рух лїворуч
26 const int Turn_Right  = LeftUpForw | LeftDownBack | RightUpBack | RightDownForw; // 01 01 01 01 Рух праворуч
27 const int Top_Left    = LeftDownForw | RightUpForw; // 00 10 01 00 Рух вперед-влїво
28 const int Bottom_Left = LeftUpBack | RightDownBack; // 10 00 10 00 Рух вниз-влїво
29 const int Top_Right   = LeftUpForw | RightDownForw; // 01 00 00 01 Рух вперед-вправо
30 const int Bottom_Right = LeftDownBack | RightUpBack; // 00 01 00 10 Рух вниз-вправо
31 const int Stop        = 0; // 00 00 00 00 Зупинка
32 const int Contrarotate = LeftUpBack | LeftDownBack | RightDownForw | RightUpForw; // 10 01 01 01 Рух проти годин. стрїл.
33 const int Clockwise   = LeftUpForw | LeftDownForw | RightDownBack | RightUpBack; // 01 10 10 10 Рух за годин. стрїл

```

Рисунок Д.1 – Програма керування рухом мобільного робота, аркуш 1

```

35 SoftwareSerial bluet_com(12, 13); // RX та TX на контролері, TX та RX на модулі Bluetooth_модуль---Контролер
36 //SoftwareSerial battery_com(9,10); // RX та TX на контролері, TX та RX на модулі Модуль_контролю_живлення(МКЖ)---Контролер
37 char contr_letter = ' '; // Команда руху
38 //createSafeString(batterymes,10); // Контейнер для запису повідомлення з МКЖ
39 unsigned long currentmillis; // Змінна для запису поточного часу
40 unsigned long previousmillis; // Змінна для запису часу отримання попередньої команди з модулю Bluetooth
41 Servo servo; // Змінна для керування серводвигуном
42
43 void setup()
44 {
45     bluet_com.begin(9600); // Віртуальний послідовний порт для даних Bluetooth-модуля
46     pinMode(SHCP_PIN, OUTPUT); // Призначення контактів для регістра зсуву 74НСТ595N
47     pinMode(EN_PIN, OUTPUT);
48     pinMode(DATA_PIN, OUTPUT);
49     pinMode(STCP_PIN, OUTPUT);
50     pinMode(PWM1_PIN, OUTPUT); // Призначення контактів для ШІМ
51     pinMode(PWM2_PIN, OUTPUT);
52     servo.attach(11); // Призначення контакту для серводвигуна
53 }
54
55 void loop()
56 {
57     currentmillis = millis(); // Запис поточного часу
58     if(bluet_com.available()) // Якщо з модулю Bluetooth надходить будь-що...
59     {
60         contr_letter = bluet_com.read(); //... то зчитати ці дані...
61         previousmillis = currentmillis; //... записати час отримання повідомлення. Також, якщо...
62
63         if(contr_letter=='2') { //...дані збігаються з командою руху вперед,...
64             Motor(Forward, 250); //...увімкнути рух вперед
65         }
66         else if(contr_letter == '8'){ // Рух назад, аналогічно руху вперед
67             Motor(Backward, 250);
68         }
69     }
70 }

```

Рисунок Д.1, аркуш 2

```
69     else if(contr_letter == '4'){ // Рух вліво, аналогічно руху вперед
70         Motor(Turn_Left, 250);
71     }
72     else if(contr_letter == '6'){ // Рух вправо, аналогічно руху вперед
73         Motor(Turn_Right, 250);
74     }
75     else if(contr_letter == '1'){ // Рух вперед-вліво, аналогічно руху вперед
76         Motor(Top_Left, 250);
77     }
78     else if(contr_letter == '3'){ // Рух вперед-вправо, аналогічно руху вперед
79         Motor(Top_Right, 250);
80     }
81     else if(contr_letter == '7'){ // Рух вниз-вліво, аналогічно руху вперед
82         Motor(Bottom_Left , 250);
83     }
84     else if(contr_letter == '9'){ // Рух вниз-вправо, аналогічно руху вперед
85         Motor(Bottom_Right, 250);
86     }
87     else if(contr_letter == 'E'){ // Рух за год. стрілкою, аналогічно руху вперед
88         Motor(Clockwise, 250);
89     }
90     else if(contr_letter == 'Q'){ // Рух проти год. стрілки, аналогічно руху вперед
91         Motor(Contrarotate, 250);
92     }
93     else if(contr_letter == '5'){ // Зупинитися, аналогічно руху вперед
94         Motor(Stop, 250);
95     }
96     else if(contr_letter == 'L'){ // Серводвигун на 45 гр вліво, аналогічно руху вперед
97         servo.write(45);
98     }
99     else if(contr_letter == 'M'){ // Серводвигун прямо, аналогічно руху вперед
100        servo.write(90);
101    }
```

Рисунок Д.1, аркуш 3

```
102     else if(contr_letter == 'R'){ // Серводвигун на 45 гр вправо, аналогічно руху вперед
103         servo.write(135);
104     }
105 }
106
107 if ((currentmillis-previousmillis)>1000){ // Якщо датчик Bluetooth надіслав останнє повідомлення більше ніж 1с назад...
108     Motor(Stop, 250); // ...зупинити двигун
109 }
110
111 delay(100); //Затримка в 100 мс
112 }
113
114 void Motor(int Dir, int Speed) // Функція керування двигунами
115 {
116     digitalWrite(EN_PIN, LOW); // Розпочати передачу даних регістру 74НСТ595N
117     analogWrite(PWM1_PIN, Speed); // Надіслати ШІМ до першого драйвера двигунів
118     analogWrite(PWM2_PIN, Speed); // Надіслати ШІМ до другого драйвера двигунів
119
120     digitalWrite(STCP_PIN, LOW); // Зупинити виведення даних
121     shiftOut(DATA_PIN, SHCP_PIN, MSBFIRST, Dir); // Завантажити до регістру значення в Dir, старший біт перший
122     digitalWrite(STCP_PIN, HIGH); // Вивести дані з регістру
123 }
```

Рисунок Д.1, аркуш 4

## ДОДАТОК Е

## ЛІСТИНГ ПРОГРАМИ ПЕРЕДАЧІ ЗОБРАЖЕННЯ З КАМЕРИ ДО ВЕБ-СЕРВЕРА

```

1  #include <WebServer.h>
2  #include <WiFi.h>
3  #include <esp32cam.h>
4
5  const char* WIFI_SSID = "тут повинне бути ім'я Вашої мережі"; // Ім'я мережі Wi-Fi
6  const char* WIFI_PASS = "тут повинен бути пароль від мережі"; // Пароль від мережі Wi-Fi
7
8  WebServer server(80); //Створення веб-сервера на порту 80
9
10
11 static auto loRes = esp32cam::Resolution::find(320, 240); //Структури для збереження розд. здат., найближчих до записаних у функцію
12 static auto midRes = esp32cam::Resolution::find(350, 530);
13 static auto hiRes = esp32cam::Resolution::find(800, 600);
14
15 void serveJpg() // Захоплення зображення та його відправлення клієнту через веб-сервер
16 {
17     auto frame = esp32cam::capture(); // Захоплення кадру до frame
18     if (frame == nullptr) { // Якщо frame ніде не записано в пам'яті...
19         Serial.println("CAPTURE FAIL"); //... відправити повідомлення на послідовний порт...
20         server.send(503, "", ""); //... серверу -- помилку (503 -- сервіс тимчасово недоступний)...
21         return; // ... та вийти з функції serveJpg()...
22     }
23     Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame->getHeight(), //Інакше, записати до послідовного порту повідомлення про ...
24                 static_cast<int>(frame->size())); // ..успішне підключення (+шир, вис, розмір в байтах)
25
26     server.setContentLength(frame->size()); // Вказати серверу розмір даних до передачі...
27     server.send(200, "image/jpeg"); // ...повідомлення про успішну обробку запиту (200) та тип даних, що отримає клієнт (image/jpeg)...
28     WiFiClient client = server.client(); // ...отримує клієнта, що підключився до сервера...
29     frame->writeTo(client); // ...та надсилає йому зображення
30 }
31
32 void handleJpgLo() //Обробка помилок при змін розд. здат. на loRes
33 {
34     if (!esp32cam::Camera.changeResolution(loRes)) { // Якщо не вдається змінити роздільну здатність на loRes...
35         Serial.println("SET-LO-RES FAIL"); //...вивести повідомлення на послідовний порт...
36     }
37     serveJpg(); //...та знову спробувати захопити зображення
38 }

```

Рисунок Е.1 – Програма передачі зображення з камери до веб-сервера, аркуш 1

```

40 void handleJpgHi() //Обробка помилок при змін розд. здат. на hiRes, аналогічно handleJpgLo()
41 {
42     if (!esp32cam::Camera.changeResolution(hiRes)) {
43         Serial.println("SET-HI-RES FAIL");
44     }
45     serveJpg();
46 }
47
48 void handleJpgMid() //Обробка помилок при змін розд. здат. на hiRes, аналогічно handleJpgMid()
49 {
50     if (!esp32cam::Camera.changeResolution(midRes)) {
51         Serial.println("SET-MID-RES FAIL");
52     }
53     serveJpg();
54 }
55
56
57 void setup(){
58     Serial.begin(115200); // Розпочати передачу через послідовний порт
59     Serial.println(); // Новий рядок в послідовному порті
60     {
61         using namespace esp32cam; // Використовуючи простір імен esp32cam
62         Config cfg; // Створення об'єкту налаштування esp32cam:
63         cfg.setPins(pins::AiThinker); // -- піни збігаються з пінами esp32cam AiThinker
64         cfg.setResolution(hiRes); // -- роздільна здатність кадру -- висока
65         cfg.setBufferCount(2); // -- кількість буферів збереження зображення -- 2
66         cfg.setJpeg(80); // -- стиснення зображення -- 80 %
67
68         bool ok = Camera.begin(cfg); // Запуск камери із сформованим об'єктом налаштування
69         Serial.println(ok ? "CAMERA OK" : "CAMERA FAIL"); // І повідомити результат до послідовного порту
70     }
71     WiFi.persistent(false); // Не зберігати налаштування Wi-Fi в ПЗП
72     WiFi.mode(WIFI_STA); // Налаштування режиму Wi-Fi як клієнт
73     WiFi.begin(WIFI_SSID, WIFI_PASS); // Підключитися до Wi-Fi
74     while (WiFi.status() != WL_CONNECTED) { // Очікувати підключення до мережі
75         delay(500);
76     }

```

Рисунок Е.1, аркуш 2

```
77 Serial.print("http://");
78 Serial.println(WiFi.localIP());
79 Serial.println(" /cam-lo.jpg");
80 Serial.println(" /cam-hi.jpg");
81 Serial.println(" /cam-mid.jpg"); // Записати до послідовного порту повідомлення формату http://{IP}/cam-{qual}.jpg
82
83 server.on("/cam-lo.jpg", handleJpgLo); //Запити до /cam-lo.jpg обробляти функцією handleJpgLo
84 server.on("/cam-hi.jpg", handleJpgHi); //Аналогічно попередньому
85 server.on("/cam-mid.jpg", handleJpgMid); //Аналогічно попередньому
86
87 server.begin(); //Запуск веб-сервера
88 }
89
90 void loop()
91 {
92   server.handleClient(); //Обробка клієнтів веб-сервером
93 }
```

Рисунок Е.1, аркуш 3

## ДОДАТОК Ж

### ЛІСТИНГ ПРОГРАМИ ОБРОБКИ ПОТОКУ ВІДЕО НЕЙРОННОЮ МЕРЕЖЕЮ

```
1 import cv2
2 import urllib.request
3 import numpy as np
4 from ultralytics import YOLO
5
6
7 url = 'http://192.168.0.121/cam-hi.jpg' # Посилання на адресу веб-сторінки
8 cv2.namedWindow( winname: "live Cam Testing", cv2.WINDOW_AUTOSIZE) # Створити вікно для виведення зображення
9
10
11 cap = cv2.VideoCapture(url)# Створення об'єкта для зчитування зображень з відео
12
13
14 if not cap.isOpened(): # Якщо зчитування не вдається...
15     print("Failed to open the IP camera stream")#... вивести повідомлення до консолі...
16     exit() #... та припинити роботу застосунку
17
18 model = YOLO("C:/ENGINEERING/4cour_dipl/Neural_Net_code/Detector/last.pt")# Вказати розташування файлу коефіцієнтів нейронної мережі
19 threshold = 0.3 # Поріг ймовірності розпізнання (для позначення виключно деяких предметів на зображенні)
```

Рисунок Ж.1 – Програма обробки зображення нейронною мережею, аркуш 1

```

21 while True:
22
23     img_resp = urllib.request.urlopen(url) # Зчитати дані з веб-сторінки...
24     imgnp = np.array(bytearray(img_resp.read()), dtype=np.uint8)# ...перетворити їх на масив numpy uint8...
25     im = cv2.imdecode(imgnp, -1) # Автоматичне перетворення зображення в форматopencv
26
27     results = model(im)[0] #Застосувати коефіцієнти нейронної мережі до зображення
28
29     for result in results.bboxes.tolist():# Для кожного предмету, що виявлено на зображенні...
30         x1, y1, x2, y2, score, class_id = result# ...записати характеристики предмета (координати рамки, ймовірність збігу та номер класу)...
31
32         if score > threshold: # ... та якщо ймовірність збігу більша за граничну...
33             cv2.rectangle(im, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255, 0), 2)# ... обвести предмет прямокутною рамкою ...
34             cv2.putText(im, text=f'{results.names[int(class_id)]}, {score:.2f}', org=(int(x1), int(y1 - 10)),# ... та вставити над нею текст...
35                         cv2.FONT_HERSHEY_SIMPLEX, fontScale=0.5, color=(0, 255, 0), thickness=2)#... із зазначенням назви класу та ймовірності збігу
36
37     cv2.imshow( winname: 'live Cam Testing', im) # Вивести зображення на відеопотік
38     key = cv2.waitKey(5) # Якщо протягом 5 мс...
39     if key == ord('q'): # ...користувач натискає кнопку q...
40         break # ... вийти з нескінченного циклу...
41
42 cap.release() #... перестати зчитувати відео...
43 cv2.destroyAllWindows() #... та закрити всі вікна.

```

Рисунок Ж.1, аркуш 2

## ДОДАТОК И

### ЛІСТИНГ ПРОГРАМИ ТЕСТУВАННЯ МОДЕЛІ НА ЗОБРАЖЕННЯХ РІЗНОЇ ЯСКРАВОСТІ

```

1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from ultralytics import YOLO
5
6 def apply_gamma(channel, gamma):          # Функція перетворення каналу яскравості для гамма- корекції
7     channel = channel / 255.0             # Перетворення значення матриці з 0..255 на 0..1
8     corrected = np.power(*args: channel, gamma) # Введення значень матриці до степені gamma
9     return np.uint8(corrected * 255)      # Повернення модифікованого каналу
10 def neuralnet_the_im(image): # Функція обробки зображення нейронною мережею (збігається із зазначеною в програмі обробки відеопотоку)
11     results = model(image)[0]
12     confidences = []                    # Масив ймовірностей розпізнавання для кожного предмета на зображенні
13
14     for result in results.boxes.data.tolist():
15         x1, y1, x2, y2, score, class_id = result
16         confidences.append(score)
17         if score > threshold:
18             cv2.rectangle(image, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255, 0), 2)
19             cv2.putText(image, text=f'{results.names[int(class_id)]}, {score:.2f}', org=(int(x1), int(y1 - 10)),
20                 cv2.FONT_HERSHEY_SIMPLEX, fontScale=0.5, color=(0, 255, 0), thickness=2)
21     return image, confidences
22
23 model = YOLO("C:/ENGINEERING/4cour_dipl/Neural_Net_code/Detector/last.pt") # Вказати розташування файлу коефіцієнтів нейронної мережі
24 threshold = 0.3                    # Поріг ймовірності розпізнавання (для позначення виключно деяких предметів на зображенні)
25
26 img = cv2.imread("C:/ENGINEERING/4cour_dipl/TestImg.jpg") # Зчитування тестового зображення
27 img_hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)            # Перетворення зображення з RGB до HSV
28
29 fig, axes = plt.subplots(nrows=6, ncols=2, figsize=(5, 12)) # Створення макету результату у формі таблиці 6 на 2 графіків та фіксованим розміром кінцевого зображення
30 axes = axes.flatten()                    # "Притулити" зображення до осей графіків
31

```

Рисунок И.1 – Програми тестування моделі на зображеннях різної яскравості, аркуш 1

```

32 gammas = [1] # Масив гамма для графіку, перше значення =1
33 avg_confidences = []# Масив серед. ймов. для графіку
34
35 img_with_boxes, original_confidences = neuralnet_the_im(img.copy()) # Обробка оригінального зображення нейронною мережею, отримання масиву ймовірностей
36 org_av_conf = np.mean(original_confidences) if original_confidences else 0# Отримання середньої ймовірності для першого зображення
37 avg_confidences.append(org_av_conf) # Додати першу середню ймовірність до масиву
38
39 axes[0].imshow(cv2.cvtColor(img_with_boxes, cv2.COLOR_BGR2RGB)) # Вставити оригінальне зображення вибухівки до зображення-результату
40 axes[0].set_title("Оригінал", fontsize=14, fontfamily="Times New Roman", y = -0.2) # Підписати зображення знизу
41 axes[0].axis('off') # Прибрати вісь координат
42
43
44 for i in range(11): # Для інших одинадцяти зображень...
45     buf_img_hsv = img_hsv.copy() # ... створити зображення-буфер,...
46     if i < 4: # ... вибрати значення гамма...
47         gamma = round(0.2 * (i + 1), 2) # 0.2, 0.4, 0.6, 0.8
48     else:
49         gamma = 2 * (i - 3) # 2, 4, 6, 8, 10
50
51     gammas.append(gamma)
52     buf_img_hsv[:, :, 2] = apply_gamma(buf_img_hsv[:, :, 2], gamma) # ... модифікувати канал яскравості (див. функцію вище)...
53     buf_img_corrected = cv2.cvtColor(buf_img_hsv, cv2.COLOR_HSV2RGB)# ... перетворити зображення з HSV на RGB...
54
55     result_img, confidences = neuralnet_the_im(buf_img_corrected) # ...отримання обробленого нейр. мережею зображення та ймовірностей розпізнавання предметів...
56     avg_conf = np.mean(confidences) if confidences else 0 # ... знайти середню ймовірність...
57     avg_confidences.append(avg_conf) # ... доповнити масив середніх ймовірностей.
58
59     axes[i + 1].imshow(result_img) # ...вставити зображення до зображення-результату
60     axes[i + 1].set_title(f"Гамма {gamma}", fontsize=14, fontfamily="Times New Roman", y = -0.2) # Підписати зображення знизу
61     axes[i + 1].axis('off') # Прибрати вісь координат

```

Рисунок И.1, аркуш 2

```
63 plt.tight_layout() # Встановити найменші відступи між зображеннями в таблиці
64 plt.show()        # Вивести результуюче зображення
65
66 plt.figure(figsize=(6, 4)) # Встановлення розміру графіку
67 plt.plot(*args: gammas, avg_confidences, marker='o', color='black') # Побудувати графік функції
68 plt.xlabel(xlabel: "Гамма", fontsize=12, fontfamily="Times New Roman") # Підпис осі x
69 plt.ylabel(ylabel: "Середня ймовірність", fontsize=12, fontfamily="Times New Roman") # Підпис осі y
70 plt.grid(True) # Увімкнути лінії сітки
71 plt.show() # Вивести графік
```

Рисунок И.1, аркуш 3

**ДОДАТОК К**  
**ДЕМОНСТРАЦІЙНИЙ МАТЕРІАЛ**

