ДОДАТОК А

Апробація

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-tehnologiy-avtomatizatsiyi-ta-mehatroniki-kitam

http://itez.zntu.edu.ua/

http://kafea.kdu.edu.ua

Харків 2025

UDC 004.93

ANALYSIS OF COMPUTER VISION SYSTEMS FOR OBJECT RECOGNITION

O.V.Kornienko
Kharkiv National University of Radio Electronics
Ukraine, 61166, Kharkiv, Nauky Ave. 14
E-mail: oleksii.korniienko3@nure.ua

The paper considers the current issue of computer vision systems. An analysis of existing methods of object recognition is conducted. The use of computer vision is due to a number of reasons: increasing the efficiency of object search for mobile robots and quadcopters, flexibility of robotic systems and their automation.

Keywords: computer vision, quadcopter, unmanned systems, robotic systems

АНАЛІЗ СИСТЕМ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ РОЗПІЗНОВАННЯ ОБ'ЄКТІВ

О.В.Корнієнко
Харківський національний університет радіоелектроніки
Україна, 61166, Харків, пр. Науки 14
E-mail: oleksii.korniienko3@nure.ua

В роботі розглянуте актуальне питання систем комп'ютерного зору. Проведено аналіз існуючих методів розпізнования об'єктів. Використання комп'ютерного зору обумовлено рядом причин: підвищення ефективності пошуку об'єктів для мобільних роботів та квадракоптерів, гнучкість роботизованих систем та їх автоматизація.

Ключові слова: комп'ютерний зір, квадракоптер, безпілотні системи, роботозовані системи

RELEVANCE OF THE WORK. Unmanned systems, especially quadcopters, are now gaining popularity, they have a lot of advantages, such as cheapness, speed, easy control. The use of computer vision greatly facilitates the operator's work, performing the work of searching for the necessary targets for him, whether it is a rescue operation or reconnaissance in difficult conditions. Improving computer vision systems will allow robotic systems to find the necessary objects faster and more efficiently, which will also improve the speed of response during autonomous operation.

INTRODUCTION. For more effective use of computer vision, it is necessary to analyze existing systems and the tasks they can perform. The main tasks performed:

- Object recognition
- Tracking of one object
- Recognition of gestures, facial features, facial expressions, etc.
- Orientation in the area
- Search and fixation of photos under given conditions

MATERIAL AND RESEARCH RESULTS. Modern computer vision technologies offer a wide range of solutions - from simple algorithms to complex systems with artificial intelligence. In most cases, the choice of algorithm depends on the specific goal. For example, the simplest computer vision algorithm is sot.py. This simple algorithm is used to capture and track an object. It can be used in enterprises to automate control processes, track quadcopters, video surveillance systems, track the movement of objects in real time, etc. The simplicity of implementing such algorithms allows them to be quickly integrated into practical projects, reducing development and implementation costs.

More complex algorithms can recognize human faces using the method of geometric shapes. In this approach, the face is conventionally divided into a set of key elements that form certain geometric structures. The algorithm analyzes the mutual arrangement of these elements - eyes, nose, mouth, jaw lines, etc. - and creates a unique "map" of the face. This map is then compared with previously stored samples in a database, allowing for accurate identification of the individual.

This technology has wide applications in the field of security and data protection. It can be used, for example, to control access to devices or premises where personal identification is important. The face can also act as an additional or even primary key for authentication, providing ease of use while at the same time providing a high level of protection for personal information.

Sophisticated computer vision systems also use neural networks to increase efficiency and perform more complex tasks, such as recognizing various objects. Unlike simple algorithms that can only track movement or determine basic shapes, neural networks allow for much deeper analysis of visual information. They are able to learn from large amounts of data, recognize subtle differences between objects, adapt to changes in the environment, and take into account the context in which the object is located. However, the development and implementation of neural networks require significant computing resources and training time, but the result often justifies the costs.

These technologies open up new opportunities for automation, analytics, and interaction of systems with the environment, significantly expanding the boundaries of what can be achieved with the help of computer vision.

As an example of using a neural network, let's take the MobileNetV2 model. This is one of the effective and at the same time lightweight deep learning models, which was specially designed to work on devices with limited resources. Due to the optimized structure, MobileNetV2 demonstrates high image processing speed while maintaining a sufficient level of accuracy, which makes it ideal for use in mobile applications, embedded systems, and Internet of Things devices.

The model is capable of recognizing multiple objects simultaneously in real time, which significantly expands the scope of its possible use. It not only allows you to detect and classify objects, but also works effectively in conditions of changing lighting, different viewing angles, and dynamic scenes(Fig.1.1).

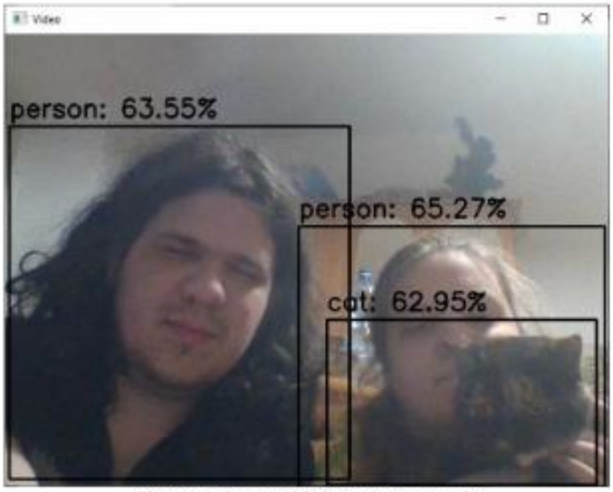

Figure 1.1 – MobileNetV2 example

CONCLUSIONS. The introduction of more powerful computer vision systems opens up many opportunities for future robotics, significantly improving our lives. The use of these systems will help optimize production at the enterprise, improve accuracy and speed. It is also worth noting that computer vision systems can greatly facilitate rescue operations, finding victims even in difficult weather conditions

LITERATURE

1. 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.

2. 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.

3. Невлюдов, І. Ш., Євсєєв, 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.

4. 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.

5. Yevsieiev, V., Maksymova, S., Alkhalaileh, 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.

6. Yevsieiev, V., Maksymova, S., Gurin, D., & Alkhalaileh, A. (2024). HR DATA VISUALIZATION OF THE DISTANCE TO THE OBJECT IN THE COLLABORATIVE ROBOT WORKSPACE BASED ON HC-SR04 SENSOR. *ACUMEN: International journal of multidisciplinary research*, *1*(4), 388-401.

7. Yevsieiev, V., Abu-Jassar, A., & Maksymova, S. (2024). Calculation of the Distance to Objects in Collaborative Robots Workspace Using Computer Vision. *Journal of universal science research*, *2*(11), 240-255.

8. Yevsieiev, V., Maksymova, S., Gurin, D., & Alkhalaileh, A. (2024). Data Fusion Research for Collaborative Robots-Manipulators within Industry 5.0. *ACUMEN: International journal of multidisciplinary research*, *1*(4), 125-137.

9. Yevsieiev, V., Maksymova, S., & Alkhalaileh, A. (2024). Capturing Human Movements in Real Time in Collaborative Robots Workspace within Industry 5.0. *Journal of universal science research*, *2*(10), 232-247.

10. Maksymova, S., Abu-Jassar, A., Gurin, D., & Yevsieiev, V. (2024). Comparative Analysis of methods for Predicting the Trajectory of Object Movement in a Collaborative Robot-Manipulator Working Area. *Multidisciplinary Journal of Science and Technology*, *4*(10), 38-48.

11. Gurin, D., Yevsieiev, V., Maksymova, S., & Abu-Jassar, A. (2024). Effect of Frame Processing Frequency on Object Identification Using MobileNetV2 Neural Network for a Mobile Robot. *Multidisciplinary Journal of Science and Technology*, *4*(8), 36-44.

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

ДОДАТОК Б

Код програми для отримання та обробки зображення

```
import cv2
import numpy as np
import tensorflow as tf
import urllib.request

# Завантаження моделі MobileNetV2 для розпізнавання об'єктів
model = tf.saved_model.load(r"C:\Users\User\Desktop\діплом\модель
нейронкі\ssd-mobilenet-v2\1")
# Список класів об'єктів
class_names = [
    '__background__', 'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus',
        'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'N/A', 'stop sign',
        'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow',
        'elephant', 'bear', 'zebra', 'giraffe', 'N/A', 'backpack', 'umbrella', 'N/A', 'N/A',
        'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball',
        'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard', 'tennis racket',
        'bottle', 'N/A', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl',
        'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza',
        'donut', 'cake', 'chair', 'couch', 'potted plant', 'bed', 'N/A', 'dining table',
        'N/A', 'N/A', 'toilet', 'N/A', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell
phone',
        'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'N/A', 'book',
        'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush'
]
# Функція для розпізнавання об'єктів на зображенні
def detect_objects(image):
    input_tensor = tf.convert_to_tensor(image, dtype=tf.uint8)
    input_tensor = input_tensor[tf.newaxis, ...]
    detections = model.signatures['serving_default'](input_tensor)
    return detections

url = 'http://192.168.43.117/cam-hi.jpg'
detection_start = 0

while True:
    cv2.namedWindow("live transmission", cv2.WINDOW_AUTOSIZE)
    while True:
```

```
        img_resp = urllib.request.urlopen(url)
        imgnp = np.array(bytearray(img_resp.read()), dtype=np.uint8)
        im = cv2.imdecode(imgnp, -1)

        key = cv2.waitKey(5)
        if key == ord('q'):
            break
        if key == 32:
            if detection_start == 0:
                detection_start = 1
            else:
                detection_start = 0

        if detection_start == 1:
            detections = detect_objects(im)
            # Обробка результатів розпізнавання
            num_detections = int(detections['num_detections'][0])
            detection_classes =
detections['detection_classes'][0].numpy().astype(np.int64)
            detection_boxes = detections['detection_boxes'][0].numpy()
            detection_scores = detections['detection_scores'][0].numpy()
            # Виведення результатів на кадр
            for i in range(num_detections):
                if detection_scores[i] > 0.5:
                    box = detection_boxes[i]
                    y1, x1, y2, x2 = box
                    y1, x1, y2, x2 = int(y1 * im.shape[0]), int(x1 * im.shape[1]), int(y2 *
im.shape[0]), int(x2 * im.shape[1])
                    # Отримати назву класу
                    class_id = detection_classes[i]
                    class_name = class_names[class_id] if class_id < len(class_names) else
'Unknown'
                    # Виведення результатів на кадр
                    cv2.rectangle(im, (x1, y1), (x2, y2), (0, 0, 0), 2)
                    label = f"{class_name}: {detection_scores[i]*100:.2f}%"
                    cv2.putText(im, label, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.9, (0, 0, 0), 2)
        cv2.imshow('live transmission', im)
    if cv2.waitKey(1) & 0xFF == 27:
        break
cap.release()
cv2.destroyAllWindows()
```

ДОДАТОК В

Код програми для ESP32-CAM

```
#include <WebServer.h>
#include <WiFi.h>
#include <esp32cam.h>

const char* WIFI_SSID = "www";
const char* WIFI_PASS = "www777www";

WebServer server(80);

static auto loRes = esp32cam::Resolution::find(320, 240);
static auto midRes = esp32cam::Resolution::find(350, 530);
static auto hiRes = esp32cam::Resolution::find(800, 600);
void serveJpg()
{
  auto frame = esp32cam::capture();
  if (frame == nullptr) {
    Serial.println("CAPTURE FAIL");
    server.send(503, "", "");
    return;
  }
  Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame-
>getHeight(),
                static_cast<int>(frame->size()));

  server.setContentLength(frame->size());
  server.send(200, "image/jpeg");
```

```
  WiFiClient client = server.client();
  frame->writeTo(client);
}

void handleJpgLo()
{
  if (!esp32cam::Camera.changeResolution(loRes)) {
    Serial.println("SET-LO-RES FAIL");
  }
  serveJpg();
}

void handleJpgHi()
{
  if (!esp32cam::Camera.changeResolution(hiRes)) {
    Serial.println("SET-HI-RES FAIL");
  }
  serveJpg();
}

void handleJpgMid()
{
  if (!esp32cam::Camera.changeResolution(midRes)) {
    Serial.println("SET-MID-RES FAIL");
  }
  serveJpg();
}



void  setup(){
```

```
Serial.begin(115200);
Serial.println();
{
  using namespace esp32cam;
  Config cfg;
  cfg.setPins(pins::AiThinker);
  cfg.setResolution(hiRes);
  cfg.setBufferCount(2);
  cfg.setJpeg(80);

  bool ok = Camera.begin(cfg);
  Serial.println(ok ? "CAMERA OK" : "CAMERA FAIL");
}
WiFi.persistent(false);
WiFi.mode(WIFI_STA);
WiFi.begin(WIFI_SSID, WIFI_PASS);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
}
Serial.print("http://");
Serial.println(WiFi.localIP());
Serial.println("  /cam-lo.jpg");
Serial.println("  /cam-hi.jpg");
Serial.println("  /cam-mid.jpg");

server.on("/cam-lo.jpg", handleJpgLo);
server.on("/cam-hi.jpg", handleJpgHi);
server.on("/cam-mid.jpg", handleJpgMid);

server.begin();
```

```
}

void loop()
{
  server.handleClient();
}
```

ДОДАТОК Г

Демонстраційний матеріал

Формат	Позиція	Позначення	Найменування	Кількість	Примітка
			<u>Текстові документи</u>		
А4	1	ГЮІК.XXXXXX.025 ПЗ	Пояснювальна записка	А4, 62 с.	
			<u>Інші документи</u>		
	2		Демонстраційні матеріал	А4, 10, с.	

Змін.	Арк.	Номер докум.	Підпис	Дата	ГЮІК.XXXXXX.025 ВД			
Розроб.		Корнієнко О.В.			Розроблення системи комп'ютерного зору для розпізнавання об'єктів на базі нейронної мережи для квадракоптера	Літера	Аркуш	Аркушів
Перевір.		Євсєєв В.В.				Н	1	1
						Кафедра КІТАР ХНУРЕ		
Н.контр.		Демська Н.П.						
Затв.		Невлюдов І.Ш.						