

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

Кафедра комп'ютерних наук

**Теорія розпізнавання образів та класифікації в системах
штучного інтелекту**

Лабораторна робота №6

Виконав:

Студент групи КН-20002Б

Кропивка Анатолій Анатолійович

Київ 2023

Тема: Дослідження методів розпізнавання контурів геометричних об'єктів на графічних зображеннях.

Мета: Отримання практичних навичок використання методів контурного аналізу при розпізнаванні контурів геометричних об'єктів.

Підготовка до роботи: Вивчити й уявити теоретичні відомості щодо способів класифікації образів на основі методів кластерного аналізу.

Хід роботи:

1. Ознайомитись з методичною розробкою до лабораторної роботи.
2. Ознайомитись з рекомендованою літературою.
3. Дослідити основні процеси створення програмних додатків контурного аналізу при розпізнаванні контурів геометричних об'єктів.
4. Виконати завдання до лабораторної роботи.
5. За результатами досліджень скласти звіт з обґрунтованими висновками.

Опис програми

Task 1:

У цьому завданні створюється проста програма, яка відкриває веб-камеру і відображає потік зображення на екрані. Вона використовує OpenCV для отримання доступу до відеопотоку з веб-камери та відображення кадрів у вікні з назвою "WebCam". Програма працює в циклі, зчитуючи кожен кадр з веб-камери, відображаючи його та очікує натискання клавіші Esc для завершення програми.

Task 2:

У цьому завданні створюється програма, яка відкриває веб-камеру і відображає потік зображення на екрані, виконуючи перетворення Canny для виявлення контурів на кадрах. Вона також очікує натискання клавіші Esc для завершення програми.

Task 3:

У цьому завданні створюється програма, яка використовує метод каскадних класифікаторів Нааг для виявлення облич на веб-камері. Програма працює в циклі, зчитуючи кожен кадр з веб-камери, конвертує його у відтінки сірого, використовує каскадний класифікатор Нааг для виявлення облич та наносить прямокутники навколо знайдених облич на кадр.

Код програми

```
#include <opencv2/opencv.hpp>
```

```
#include <iostream>
```

```
#include <vector>
```

// Task 3

```
int main(int, char**) {
```

```
    auto path = "../haarcascade_frontalface_default.xml";
```

```
    cv::VideoCapture capture(0);
```

```
    if (!capture.isOpened()) {
```

```
        std::cout << "Failed to open webcam!" << std::endl;
```

```
        return -1;
```

```
    }
```

```
    cv::CascadeClassifier cascade;
```

```
    if (!cascade.load(path)) {
```

```
        std::cout << "Failed to load cascade!" << std::endl;
```

```
    }
```

```
    while(true) {
```

```
        cv::Mat frame;
```

```
        capture.read(frame);
```

```
        if (frame.empty()) {
```

```
            std::cout << "Failed to capture frame!" << std::endl;
```

```
            break;
```

```
        }
```

```
        cv::Mat gray_frame;
```

```
        cv::cvtColor(frame, gray_frame, cv::COLOR_BGR2GRAY);
```

```
std::vector<cv::Rect> faces;

cascade.detectMultiScale(gray_frame, faces, 1.1, 2, 0 | cv::CASCADE_SCALE_IMAGE, cv::Size(30, 30));

for (const auto& face : faces) {
    cv::rectangle(frame, face, cv::Scalar(0, 255, 0), 2);
}

cv::imshow("WebCam", frame);

if (cv::waitKey(1) == 27) {
    break;
}

};

capture.release();
cv::destroyAllWindows();

return 0;
}
```

// Task 2

```
int main(int, char**) {  
    cv::VideoCapture capture(0);  
  
    if(!capture.isOpened()) {  
        std::cout << "Failed to open webcam!" << std::endl;  
        return -1;  
    }  
  
    while(true) {  
        cv::Mat frame;  
        capture.read(frame);  
  
        if (frame.empty()) {  
            std::cout << "Failed to capture frame!" << std::endl;  
            break;  
        }  
  
        cv::Mat edges;  
        cv::cvtColor(frame, edges, cv::COLOR_BGR2GRAY);  
        cv::Canny(edges, edges, 30, 60);  
  
        if (edges.empty()) {  
            std::cout << "Failed to capture edges!" << std::endl;  
        }  
  
        cv::imshow("WebCam", edges);  
  
        if (cv::waitKey(1) == 27) {  
            break;  
        }  
    }  
}
```

```
};
```

```
capture.release();
```

```
cv::destroyAllWindows();
```

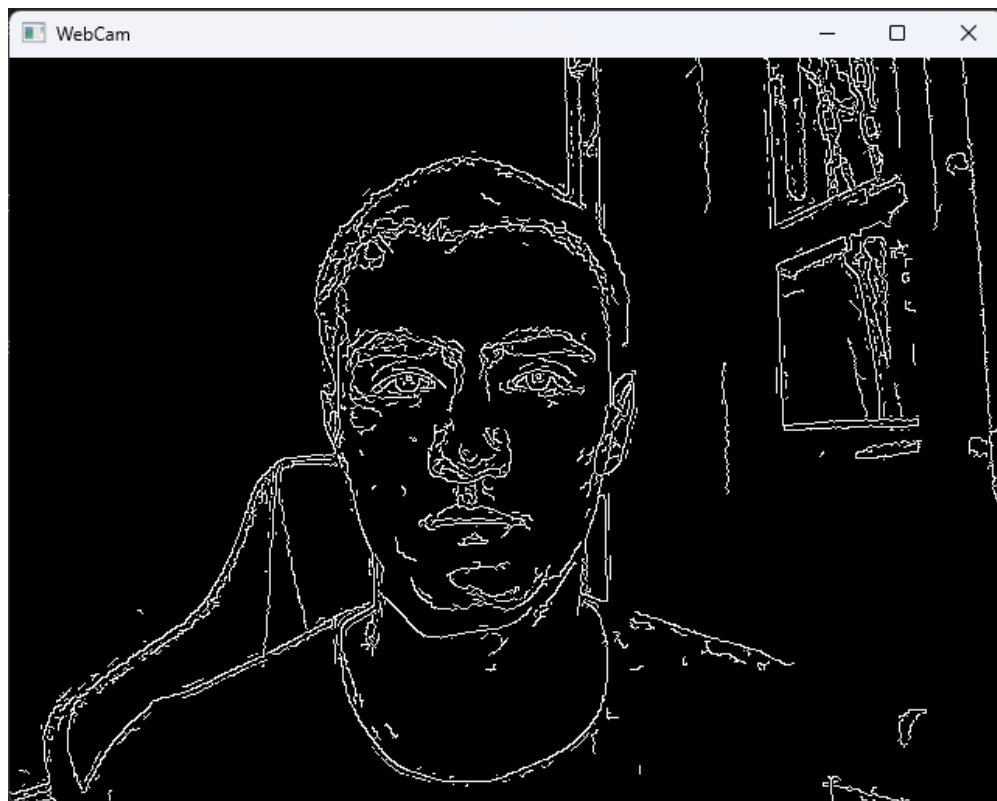
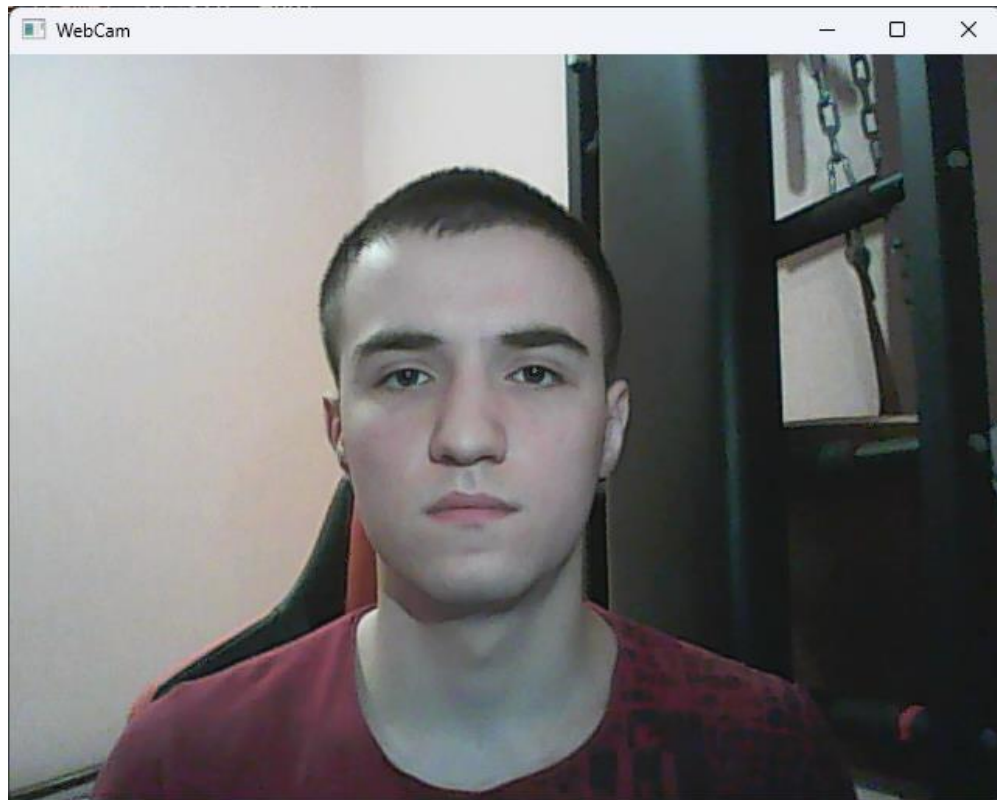
```
return 0;
```

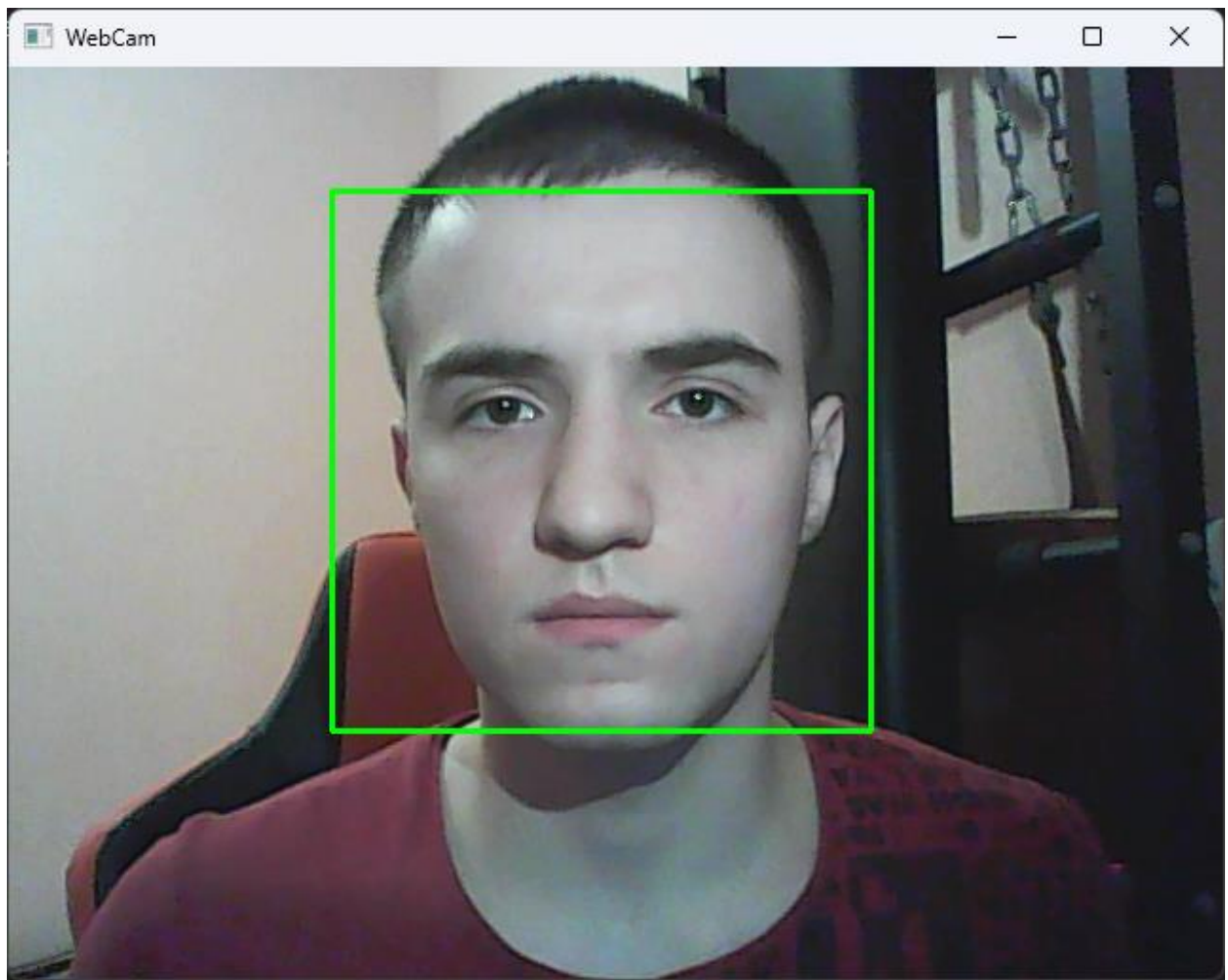
```
}
```

// Taks 1

```
int main(int, char**) {  
    cv::VideoCapture capture(0);  
  
    if(!capture.isOpened()) {  
        std::cout << "Failed to open webcam!" << std::endl;  
        return -1;  
    }  
  
    while(true) {  
        cv::Mat frame;  
        capture.read(frame);  
  
        if (frame.empty()) {  
            std::cout << "Failed to capture frame!" << std::endl;  
            break;  
        }  
  
        cv::imshow("WebCam", frame);  
  
        if (cv::waitKey(1) == 27) {  
            break;  
        }  
    };  
  
    capture.release();  
    cv::destroyAllWindows();  
  
    return 0;  
}
```


Знімки екрану





Висновок: Усі три завдання використовують OpenCV для отримання доступу до веб-камери, отримання кадрів та їх обробки. Кожне завдання виконує різні операції з кадрами, такі як відображення, виявлення контурів або облич. Ці завдання демонструють основні можливості обробки зображень та використання методів розпізнавання образів з використанням OpenCV. Вони можуть слугувати важливою основою для подальших досліджень у галузі розпізнавання образів та класифікації в системах штучного інтелекту.