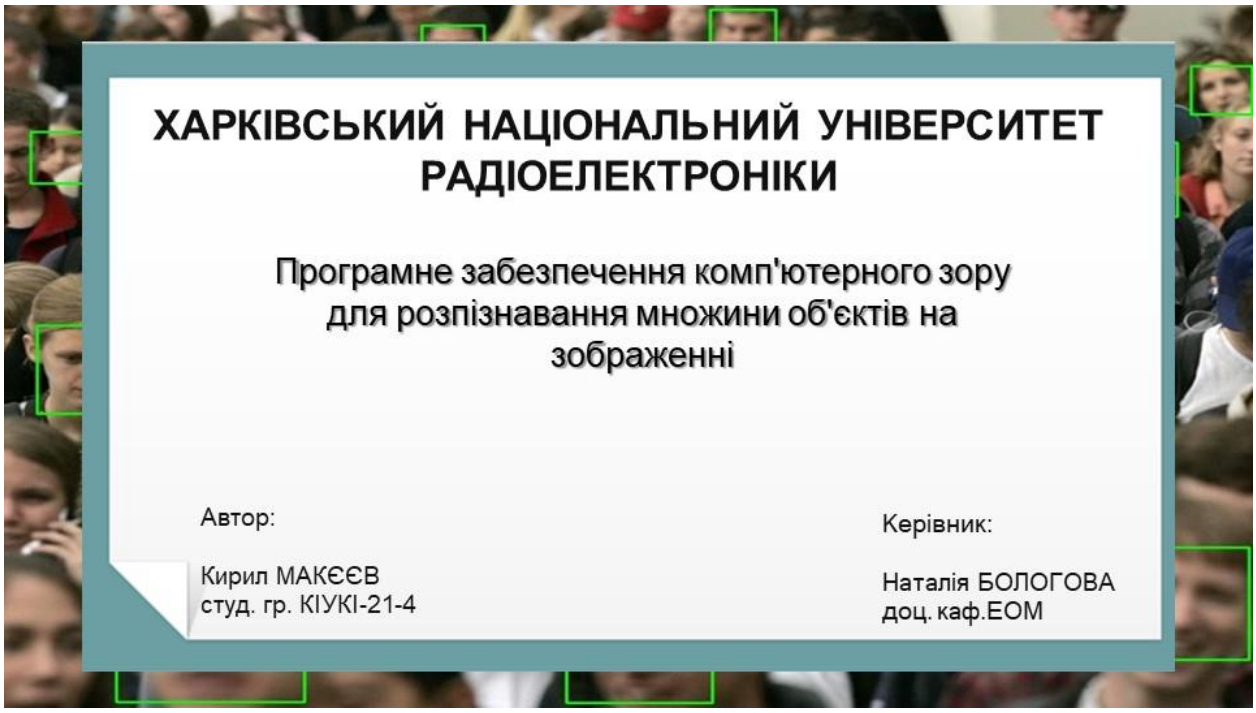
ДОДАТОК А

Графічний матеріал кваліфікаційної роботи



Мета. Завдання

Метою кваліфікаційної роботи є створення програмного продукту для комп'ютерного зору, здатного виявляти різнотипні об'єкти на зображенні з використанням алгоритму Віоли-Джонса.

У ході виконання кваліфікаційної роботи було поставлено такі завдання:

- проаналізувати принципи роботи алгоритму Віоли-Джонса та його особливості в контексті комп'ютерного зору;
- реалізувати програму, що демонструє локалізацію кількох об'єктів із використанням механізмів багатопоточності;
- провести порівняльне тестування реалізації алгоритму в однопотоковому та багатопотоковому виконанні.

2

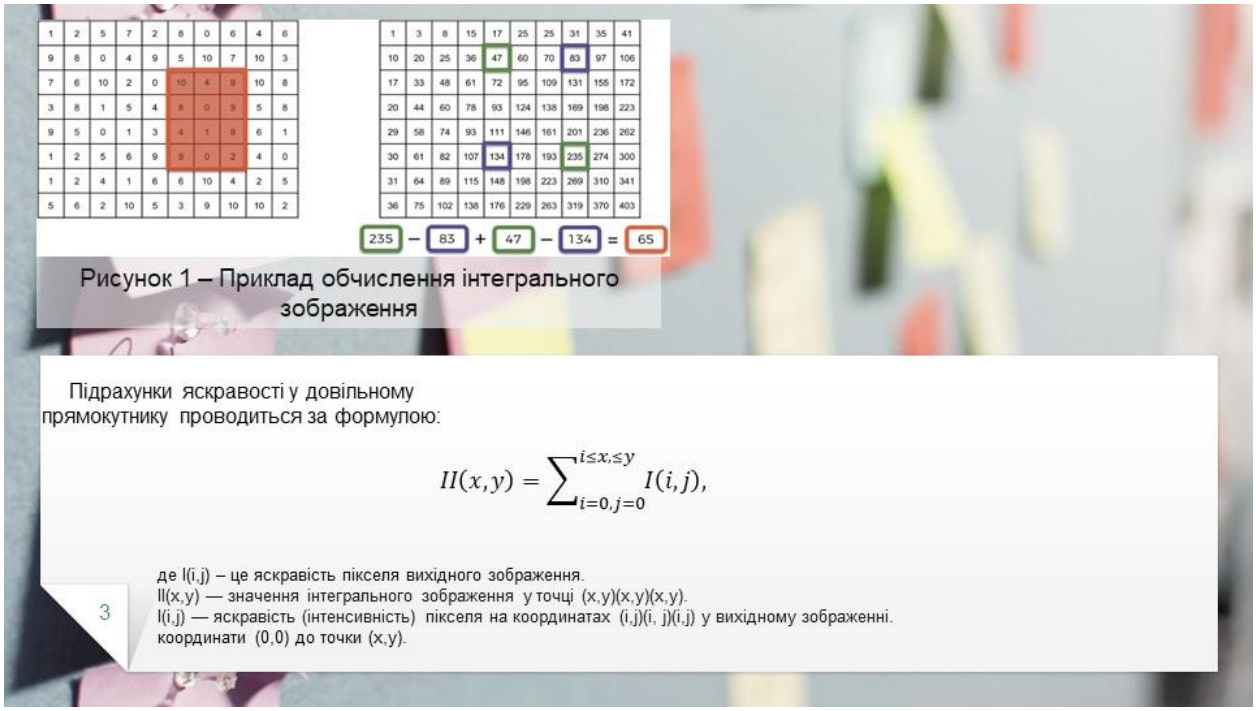


Рисунок 1 – Приклад обчислення інтегрального зображення

Підрахунки яскравості у довільному прямокутнику проводиться за формулою:

$$II(x,y) = \sum_{i=0,j=0}^{i\le x,\le y} I(i,j),$$

де I(i,j) – це яскравість пікселя вихідного зображення.
II(x,y) — значення інтегрального зображення у точці (x,y)(x,y)(x,y).
I(i,j) — яскравість (інтенсивність) пікселя на координатах (i,j)(i, j)(i,j) у вихідному зображенні.
координати (0,0) до точки (x,y).

3

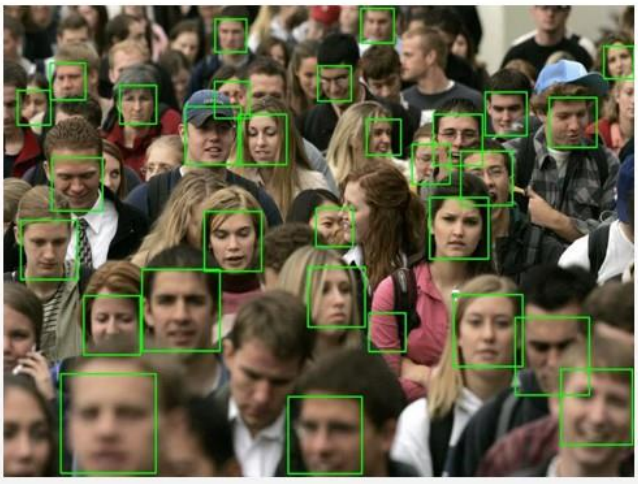


Рисунок 2 – Приклад роботи методу Віоли-Джонса

4

```
Require: scaled input image X, classifier cascade S
Ensure: vector V of detected faces (bounding boxes)
for y ← 0 to X.height − subwindow.height do
    Δ_x ← 1
    for x ← 0 to X.width − subwindow.width do
        Δ_x ← Δ_x − 1
        Δ_y[x] ← Δ_y[x] − 1
        if Δ_x = 0 AND Δ_y[x] = 0 then
            exit-stage ← S(x, y)
            if exit-stage = n then
                R ← (x, y, width, height)
                push R into V
            end if
            if exit-stage < Δ_x,t1 then
                Δ_x = Δ_x,max
            else if Δ_x,t1 ≤ exit-stage < Δ_x,t2 then
                Δ_x = Δ_x,nom
            else
                Δ_x = Δ_x,min
            end if
        else if Δ_x = 0 then
            Δ_x ← Δ_x,min
        else if Δ_y[x] = 0 then
            Δ_y[x] ← Δ_y,min
        end if
    end for
end for
```

Рисунок 3 – Алгоритм методу RASW

5

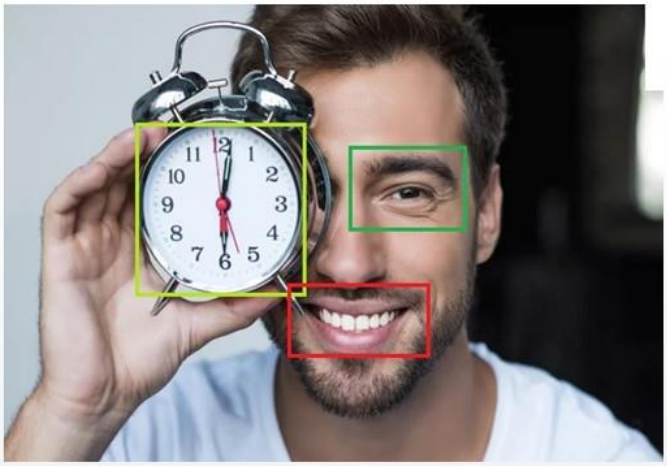


Рисунок 4 – Приклад результату роботи алгоритму

6

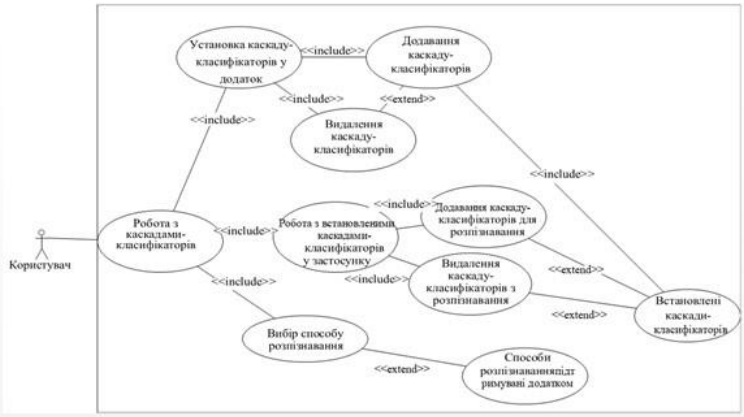
Діаграма бізнес-процесу, розроблена на основі методології IDEF0

Рисунок 5 – IDEF0-діаграма розпізнавання об'єкта

Рисунок 6 – IDEF0-діаграма завантаження навченого каскаду-класифікаторів у застосунок

7

Одним із ключових інструментів у бізнес-аналізі є діаграма варіантів використання, яка дозволяє визначити коло користувачів програмного забезпечення та окреслити перелік доступних їм дій



8

Рисунок 7 – Діаграма варіантів використання програми, що розробляється

Рисунок 9 – Блок-схема узагальненого варіанта однопотокового модуля з розпізнавання об'єкта з відео потоку

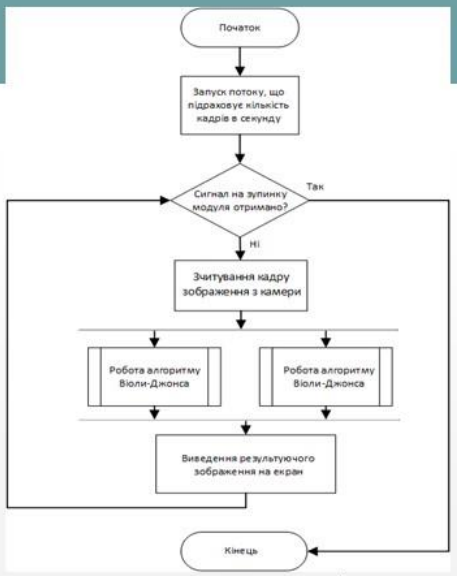


Рисунок 10 – Блок-схема узагальненого варіанта багатопоточного модуля

Проєктна структура застосунку, реалізованого у середовищі IntelliJ IDEA

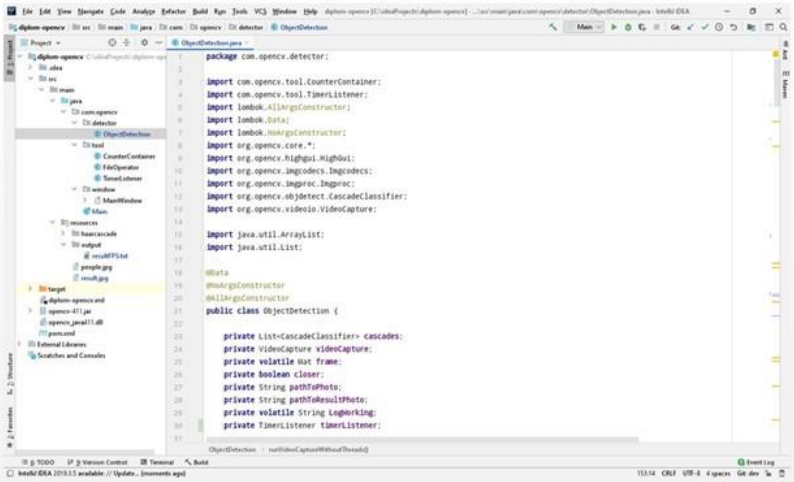


Рисунок 11 – Структура проєкту програми у середовищі IntelliJ IDEA

На основі однопотокового рішення реалізовано багатопоточну версію функції

```
public void runPictureDetection(){
    frame = Imgcodecs.imread(pathToPhoto);
    List<Thread> threads = new ArrayList<>();
    for (int i = 0; i < cascades.size() ; i++) {
        int finalI = i;
        Thread thread = new Thread(() -> {
            int r = (int) (Math.random() * (256));
            int g = (int) (Math.random() * (256));
            int b = (int) (Math.random() * (256));
            System.out.println(Thread.currentThread().getName());
            detectOnFrame(frame, cascades.get(finalI), new Scalar(r, g, b));
        });
        threads.add(thread);
    }
    for (Thread t : threads) {
        t.start();
    }
    boolean flag = true;
    while (flag){
        int counter = 0;
        for (int i = 0; i < threads.size() ; i++) {
            if(threads.get(i).getState() == Thread.State.TERMINATED){
                ++counter;
            }
        }
        if (counter == threads.size()){
            flag = false;
        }
    }
```

11

Рисунок 12 – Фрагмент коду багатопотокового варіанта розпізнавання об’єктів

Функція detectOnFrame, яка реалізує основну логіку алгоритму Віоли-Джонса

```
public void detectOnFrame(Mat frame, CascadeClassifier cascade, Scalar scalar){
    Mat frameGray = new Mat();
    Imgproc.cvtColor(frame, frameGray, Imgproc.COLOR_BGR2GRAY);
    Imgproc.equalizeHist(frameGray, frameGray);
    MatOfRect findedObject = new MatOfRect();
    try {
        cascade.detectMultiScale(frameGray, findedObject, scaleFactor: 1.3, minNeighbors: 1, flags: 0,
                new Size( width: 50, height: 50),new Size( width: 300, height: 300));
    }catch (CvException e){
        e.printStackTrace();
    }
    findedObject.toList().forEach(object->{
        Imgproc.rectangle(frame, new Point(object.x, object.y),
                new Point( x: object.x + object.width, y: object.y + object.height),
                scalar, thickness: 2);
        });
}
```

12

Рисунок 13 – Фрагмент функції detectOnFrame з реалізацією алгоритму Віоли-Джонса

Для перевірки функціональності застосунку тестування проводилося як із використанням завантаженого зображення,

Рисунок 14 – Результат однопотокового розпізнавання на завантаженому зображенні

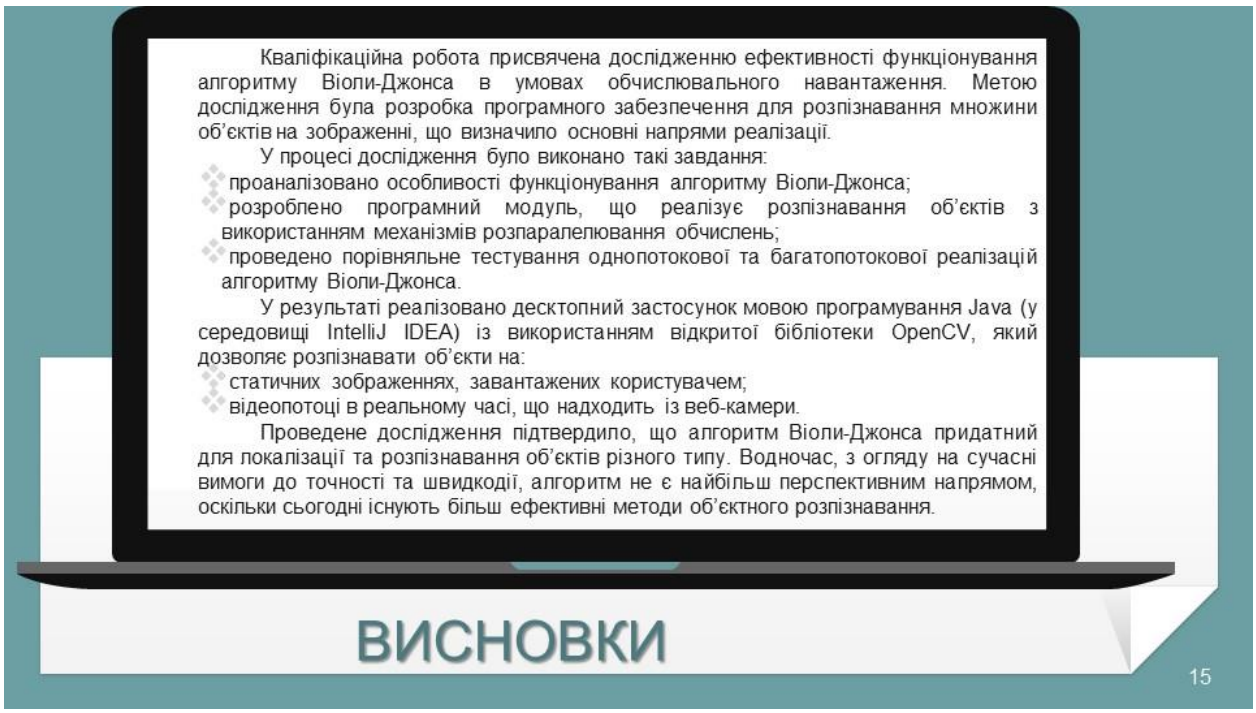
Рисунок 15– Результат багатопотокового розпізнавання на завантаженому зображенні

13

Аналіз продуктивності алгоритму Віоли-Джонса

Кількість каскадів-класифікаторів	Однопоточна реалізація	Багатопотокова реалізація
	Кадри за секунду (640×480)	
1	26	25
3	15	16
6	9	15
9	5	14
12	4	10

15


Кваліфікаційна робота присвячена дослідженню ефективності функціонування алгоритму Віоли-Джонса в умовах обчислювального навантаження. Метою дослідження була розробка програмного забезпечення для розпізнавання множини об'єктів на зображенні, що визначило основні напрями реалізації.
У процесі дослідження було виконано такі завдання:
проаналізовано особливості функціонування алгоритму Віоли-Джонса;
розроблено програмний модуль, що реалізує розпізнавання об'єктів з використанням механізмів розпаралелювання обчислень;
проведено порівняльне тестування однопотокової та багатопотокової реалізацій алгоритму Віоли-Джонса.
У результаті реалізовано десктопний застосунок мовою програмування Java (у середовищі IntelliJ IDEA) із використанням відкритої бібліотеки OpenCV, який дозволяє розпізнавати об'єкти на:
статичних зображеннях, завантажених користувачем;
відеопотоці в реальному часі, що надходить із веб-камери.
Проведене дослідження підтвердило, що алгоритм Віоли-Джонса придатний для локалізації та розпізнавання об'єктів різного типу. Водночас, з огляду на сучасні вимоги до точності та швидкодії, алгоритм не є найбільш перспективним напрямом, оскільки сьогодні існують більш ефективні методи об'єктного розпізнавання.
ВИСНОВКИ
15