

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/G7.00.1/Б /ОК31-2025
	Екземпляр № 1	Арк 33 / 1

ЗАТВЕРДЖЕНО

Науково-методичною радою
Державного університету
«Житомирська політехніка»

протокол від __ ____ 20__ р.
№ __

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ для проведення практичних (лабораторних) занять з навчальної дисципліни «Системи обробки зображень та комп'ютерний зір»

для здобувачів вищої освіти освітнього ступеня «бакалавр»
спеціальностей G7 «Автоматизація, комп'ютерно-інтегровані технології та
робототехніка».

Схвалено на засіданні кафедри
робототехніки, електроенергетики
та автоматизації ім. проф.

Б.Б. Самотокіна,
__ ____ 20__ р.,
протокол № __

Розробники: к.т.н., доцент, декан факультету комп'ютерно-інтегрованих
технологій, мехатроніки і робототехніки Ткачук А.Г.
асистент кафедри робототехніки, електроенергетики та автоматизації ім. проф.
Б.Б. Самотокіна Ткачук Д.Ю
доктор філософії, доцент кафедри робототехніки, електроенергетики та
автоматизації ім. проф. Б.Б. Самотокіна Кравчук А.Р.

Житомир
2025 – 2026 н.р.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 2

Зміст

Лабораторна робота №1

ВІЯВЛЕННЯ КОЛЬОРІВ ЗА ДОПОМОГОЮ OPENCV 3

Лабораторна робота №2

ВІЯВЛЕННЯ ФОРМ ЗА ДОПОМОГОЮ OPENCV 11

Лабораторна робота №3

ВІЯВЛЕННЯ QR-code ЗА ДОПОМОГОЮ OPENCV 21

Лабораторна робота №4

ВІДСЛІДКОВУВАННЯ ЗА КОЛЬОРОМ, ВИКОРИСТОВУЮЧИ OPENCV 24

Лабораторна робота №5

РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ TEMPLATE MATCHING В
OPENCV 29

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 3

Лабораторна робота №1

ВИЯВЛЕННЯ КОЛЬОРІВ ЗА ДОПОМОГОЮ OPENCV

Мета роботи: Ознайомитися з бібліотекою OpenCV та отримати практичні навички у написанні програмного коду розпізнавання кольорів на відеопотоках.

Теоретичні відомості

OpenCV — це бібліотека комп'ютерного зору з відкритим кодом. OpenCV також використовується в багатьох програмах реального часу та має деякі вбудовані функції для виконання операцій визначення кольору та сегментації.

OpenCV може виявляти кольори на зображенні, використовуючи простий метод порівняння кольорів у форматі кольорової моделі BGR (синій, зелений, червоний), кожен з цих каналів може приймати значення від 0 до 255.

RGB і BGR – це два різних способи подання кольорів у цифровому зображенні. Різниця полягає в порядку розташування компонент кольору (червоного, зеленого і синього) у пікселі зображення. Ця різниця у порядку компонент кольору важлива при обробці та відображенні зображень, оскільки неправильне інтерпретування порядку може призвести до некоректного відображення кольорів.

Також дана бібліотека використовується для виявлення кольорів у просторі HSV (відтінок, насиченість, значення). Відтінок вимірюється у градусах від 0 до 360, насиченість і значення вимірюються від 0 до 255.

Порядок виконання роботи

1. Імпортування необхідних бібліотек

Важливим етапом є уникнення неправильної обробки кольорів в програмі. Для цього потрібно конвертувати зображення між колірними просторами BGR і RGB за допомогою імпортування необхідних бібліотек, а саме:

- OpenCV – для читання та обробки зображень.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 4

- Matplotlib – для відображення зображень.

Бібліотека Matplotlib вже інтегрована в програмне середовище, проте OpenCV не є стандартною бібліотекою Python, тому спочатку потрібно її інсталиувати. Для інсталяції бібліотеки на Windows, необхідно мати PIP (зазвичай вже встановлено в системі). PIP — це система керування пакетами/бібліотеками, написаними на Python. Для перевірки наявності/версії PIP, відкриваємо cmd (командний рядок) та виконуємо одну із команд:

```
pip -v
```

```
py -m pip -v
```

Щоб інсталиувати OpenCV, необхідно у командному рядку виконати одну із наступних команд:

```
pip install opencv-python
```

```
py -m pip install opencv-python
```

Далі переходимо у середовище програмування Python та імпортуємо обрані бібліотеки:

```
import cv2
import numpy as np
```

2. Підключення веб-камери

```
cap = cv2.VideoCapture(0)
```

Функція **cv2.VideoCapture()** може мати два аргументу. Аргумент 0 вказує на використання веб-камери, а аргумент 1 вказує на використання шлейфової (зовнішньої) камери, якщо така підключена до системи.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОК31-2022
	Екземпляр № 1	Арк 33 / 5

3. Зчитування кадру з камери

```
while 1:
    ret, frame = cap.read()
```

ret отримає значення True, якщо фрейм існує, інакше False

4. Конвертуємо кольоровий простір BGR в HSV

```
into_hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
```

Коли ми конвертуємо зображення до HSV, ми можемо встановити діапазони до певних компонентів, щоб точніше виділити об'єкти певного кольору. Також це буде використано для створення маски виявлення потрібного кольору.

5. Встановлюємо діапазони кольору HSV (в даному випадку синій)

```
L_limit = np.array([98, 50, 50]) # Нижній ліміт синього
U_limit = np.array([139, 255, 255]) # Верхній ліміт синього
```

Також нижче наведено можливі ліміти HSV для деяких інших кольорів:

```
#ліміти червоного кольору
L_limit = np.array([160, 50, 50])
U_limit = np.array([180, 255, 255])
```

```
#ліміти зеленого кольору
L_limit = np.array([35, 50, 50])
U_limit = np.array([85, 255, 255])
```

```
#ліміти жовтого кольору
L_limit = np.array([20, 50, 50])
U_limit = np.array([85, 255, 255])
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 6

6. Створюємо маску для виділення обраного кольору

```
b_mask = cv2.inRange(into_hsv, L_limit, U_limit)
```

Створення маски за допомогою функції `inRange()` призведе до створення зображення, де колір об'єктів, що потрапляють в діапазон, буде білим, а решта – чорним.

7. Використовуємо маску на оригінальному зображенні

```
# Вилучає пікселі лише в обраному діапазоні кольору, тобто надаємо масці колір
blue = cv2.bitwise_and(frame, frame, mask=b_mask)
```

8. Відображаємо оригінальний кадр та відфільтроване зображення

```
cv2.imshow('Original', frame) # Відображення оригінального кадру
cv2.imshow('Blue Detector', blue) # Відображення виявленого синього кольору
```

9. Прописуємо вихід з програми

```
if cv2.waitKey(1) == 27:
    break # Очікуємо натискання клавіші "ESC" (код 27) для виходу з циклу
cap.release()
# Звільнюємо ресурси веб-камери та закриваємо всі вікна OpenCV
cv2.destroyAllWindows()
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОК31-2022
	Екземпляр № 1	Арк 33 / 7

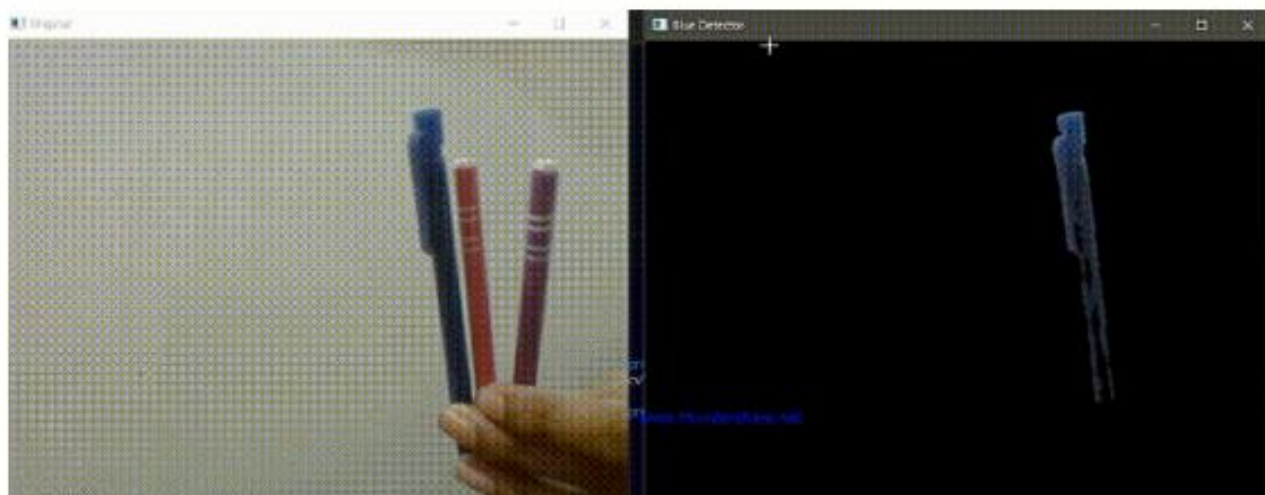


Рис.1.1. Результат виконання програми

МОДЕРНІЗАЦІЯ КОДУ З ВИКОРИСТАННЯМ РАМОК ТА КОНТУРІВ

У реальних задачах виявлення кольорових об'єктів — це лише початковий етап. Зазвичай, поставлені завдання вимагають більш детального аналізу та обробки даних. Важливим є визначення їхнього положення в просторі, розрахунок розмірів і відстеження переміщення в часі. Для досягнення цих цілей часто використовують рамки виділення (**cv2.boundingRect()**), які дозволяють деталізовано визначати області, що містять об'єкти.

Опрацювання контурів дозволяє нам виконати певну фільтрацію для відсіювання малих або непотрібних об'єктів за допомогою обрахунку площі контуру об'єкту (**cv2.contourArea()**). Контур - це замкнений контурний шлях навколо об'єкта. На рис.1.2. наведено приклад роботи коду з використанням рамок та контурів.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 8

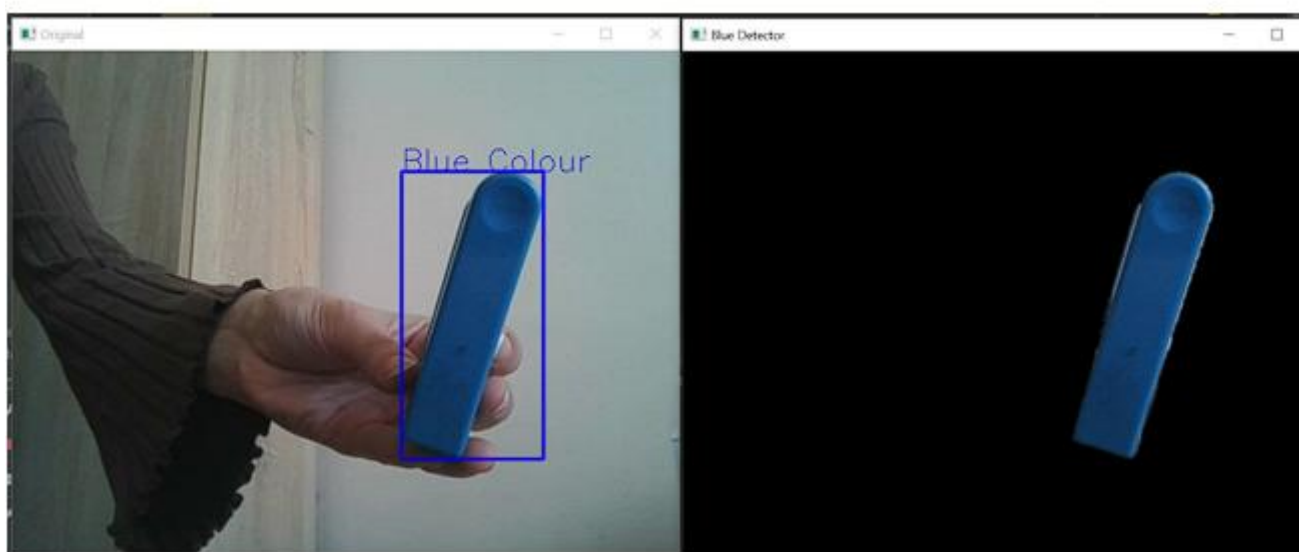


Рис. 1.2. Результат виконання програми

Перед відображенням результатів додаємо частину коду, що буде відповідати за створення контурів та рамок.

1. Знаходимо контури на зображенні в масці *b_mask*

Ми будемо використовувати **cv2.RETR_TREE** для виявлення ієрархії контурів та **cv2.CHAIN_APPROX_SIMPLE** для економії пам'яті, а також спрощення контурів.

```
contours, hierarchy = cv2.findContours(b_mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
```

2. Робимо проходи по всім контурам та виділяємо необхідний колір.

```
for contour in contours:
    area = cv2.contourArea(contour)
    if area > 300:
        x, y, w, h = cv2.boundingRect(contour)
        frame = cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
        cv2.putText(frame, "Blue Colour", (x, y), cv2.FONT_HERSHEY_SIMPLEX, 1.0, (255, 0, 0))
```

Розберемо структуру та команди циклу **for**:

```
# Проходимося по знайденим контурам
for contour in contours:
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 9

```
# Отримуємо площу поточного контуру
area = cv2.contourArea(contour)
```

```
# Перевіряємо, чи площа контуру більше за 300
if area > 300:
```

Дана умова перевіряє, чи площа контуру (area) більше 300 квадратних одиниць пікселів. Це значення може бути налаштоване відповідно до конкретних потреб задачі. Зазвичай воно встановлюється таким чином, щоб відфільтрувати маленькі контури або шум на зображенні. Якщо площа контуру перевищує цей поріг, то він вважається достатньо великим для подальшого аналізу та обробки, і відповідні дії можуть бути застосовані до цього контуру.

```
# Отримуємо координати та розміри оминаючого прямокутника для контуру
x, y, w, h = cv2.boundingRect(contour)
```

```
# Малюємо прямокутник на вихідному зображенні (frame) навколо контуру
frame = cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
```

Задані параметри функції:

- Frame – зображення, на якому потрібно намалювати прямокутник.
- (x, y) – координати верхнього лівого кута прямокутника (x – горизонтальна координата, y – вертикальна координата).
- (x + w, y + h) – координати нижнього правого кута прямокутника
- ((x + w) – горизонтальна координата, (y + h) – вертикальна координата).
- (255, 0, 0) – колір прямокутника у форматі BGR.
- Число 2 – товщина лінії, якою буде намальований прямокутник (значення у пікселях).

```
# Додаємо текст навколо об'єкта
cv2.putText(frame, "Blue Colour", (x, y), cv2.FONT_HERSHEY_SIMPLEX, 1.0, (255, 0, 0))
```

Задані параметри функції:

- Frame – зображення, на якому потрібно надрукувати текст.
- "Blue Colour" – текст, який потрібно надрукувати на зображенні.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 10

- (x, y) – координати, де розташований лівий нижній кут тексту (x – горизонтальна координата, y – вертикальна координата).
- cv2.FONT_HERSHEY_SIMPLEX – шрифт, який використовується для тексту.
- 1.0 – розмір шрифту.
- (255, 0, 0) – колір тексту у форматі BGR

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 11

Лабораторна робота №2

ВИЯВЛЕННЯ ФОРМ ЗА ДОПОМОГОЮ OPENCV

Мета роботи: Продовження ознайомлення з бібліотекою OpenCV та отримати практичні навички у написанні програмного коду розпізнавання форм на відеопотоках.

Теоретичні відомості

Виявлення форми (або аналіз форми) є важливою технікою в обробці зображень та комп'ютерному зорі. Ця техніка дозволяє комп'ютерам ідентифікувати, класифікувати та аналізувати різні форми об'єктів або областей на зображеннях/відеопотоках. Важливим елементом виявлення форми є аналіз геометричних атрибутів об'єктів, таких як розмір, форма, розташування, співвідношення між сторонами та інші характеристики.

Застосування: розпізнавання об'єктів, оцінка якості, медична діагностика, безпека, промислова автоматизація і т.д.

Для чіткого виявлення контурів форми необхідно провести певні процеси, такі як: попередня обробка зображення, виділення країв, опрацювання контурів та апроксимація форм.

Попередня обробка:

Попередня обробка зображення включає в себе перетворення зображення на градації сірого та використання розмиття за Гаусом, щоб зменшити шум у зображенні. Розподіл Гауса використовується для створення маски або фільтра, який згладжує зображення. Цей фільтр пропускає кожний піксель зображення через спеціальну формулу (2.1), яка використовує розподіл Гауса, і обчислює нове значення пікселя на основі навколишніх пікселів.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОК31-2022
	Екземпляр № 1	Арк 33 / 12

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}} \quad (2.1)$$

де σ – стандартне відхилення, яке визначає ширину та розкид розподіл;
 e – число Ейлера (приблизно 2.71828).



Рис. 2.1. Результат виконання фільтрації за допомогою розмиття Гауса

Виділення країв (Edge detection):

Виявлення країв у обробці зображень – це процес визначення місць на зображенні, де відбуваються різкі зміни інтенсивності (яскравості або кольору), що вказує на переходи між об'єктами або структурами. Це допомагає виділити контури та об'єкти на зображенні, роблячи їх більш видимими та готовими до подальшого аналізу. Одним із популярним та ефективним методом виділення країв є Canny. Даний алгоритм складається з наступних етапів:

1. Обробка за розмиттям Гауса;
2. Розрахунок градієнту;
3. Виявлення країв;
4. Підсилення країв;
5. Порогова обробка;
6. З'єднання країв.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 13



Рис. 2.2. Виділення країв методом Canny

Апроксимація форм:

Апроксимація – це процес наближення складних форм до простіших геометричних фігур, таких як лінії, кола, еліпси, або полігони.



Рис. 2.3. Схема логіки роботи апроксимації

Порядок виконання робіт

1. Імпортування необхідних бібліотек

- OpenCV – для читання та обробки зображень.
- Matplotlib – для відображення зображень.

2. Підключення веб-камери (називаємо змінну video)

3. Створення вікна TrackBar

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 14

Задачею даної лабораторної роботи є створення програми для виявлення будь-яких геометричних форм. Щоб програма була універсальною, налаштування мінімальних та максимальних значень для кольорів у просторі HSV буде відбуватися вручну за допомогою TrackBar.

TrackBar – це графічний елемент інтерфейсу користувача (GUI), який дозволяє обирати значення із заданого діапазону, переміщаючи повзунок вздовж певної шкали.



Рис.2.4. TrackBar для налаштування мінімальних та максимальних значень

```
cv2.namedWindow("TrackBar")
cv2.resizeWindow("TrackBar", 600, 300)
```

Перший рядок – це створення вікна з назвою «TrackBar», друга – зміна розміру вікна, в даному випадку (600x300).

4. Створення всіх необхідних шкал (TrackBar)

Створюємо шість трекбарів для налаштування мінімальних та максимальних значень кольорів у просторі кольорів HSV.

«**hue_min**» та «**hue_max**» – ці параметри визначають мінімальне і максимальне значення для кольорового відтінку (Hue) в просторі кольорів HSV. Відтінок (Hue) – це колір, що вказує на тон предмету без врахування яскравості та насиченості, іншими словами це частина кольору. Hue вимірюється в градусах від

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 15

0 до 179 (або 0 до 360, в залежності від конкретної реалізації). Цей параметр використовується для визначення діапазону кольорів, які будуть виділені на зображенні.

«**sat_min**» та «**sat_max**» – ці параметри визначають мінімальну і максимальну насиченість (Saturation) в просторі кольорів HSV. Цей параметр використовується для фільтрації за насиченістю кольору.

«**val_min**» та «**val_max**» – ці параметри визначають мінімальне і максимальне значення яскравості (Value) в просторі кольорів HSV. Цей параметр використовується для фільтрації за яскравістю.

```
cv2.createTrackbar("hue_min", "TrackBar", 0, 179, lambda x: None)
cv2.createTrackbar("hue_max", "TrackBar", 179, 179, lambda x: None)
cv2.createTrackbar("sat_min", "TrackBar", 0, 255, lambda x: None)
cv2.createTrackbar("sat_max", "TrackBar", 255, 255, lambda x: None)
cv2.createTrackbar("val_min", "TrackBar", 0, 255, lambda x: None)
cv2.createTrackbar("val_max", "TrackBar", 255, 255, lambda x: None)
```

Функція **cv2.createTrackbar()** створює трекбар з наступними п'ятьма параметрами функції:

1. **hue_min** – назва **TrackBar**;
2. **TrackBar** – назва вікна, в якому буде розміщений **TrackBar**;
3. **0** – мінімальне значення, яке може приймати **TrackBar**;
4. **179** – максимальне значення, яке може приймати **TrackBar**;
5. **lambda x: None** – функція зворотного виклику, яка виконується, коли користувач переміщає повзунок.

5. Створюємо цикл

В циклі прописуємо зчитування кадру **video** та конвертуємо кольоровий простір з **BGR** в **HSV**.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 16

```
while True:
    ret, img = ...
    hsv = ...
```

ret – змінна для отримання статусу читання відеопотоку.

img – змінна, що містить зчитаний кадр з відеопотоку.

6. Змінні для отримання значення TrackBar

Створюємо шість змінних, які будуть отримувати поточні значення трекбарів для мінімальних і максимальних значень кольорів у просторі HSV. Тобто, при переміщенні повзунків в ці змінні будуть записуватися налаштовані значення.

```
hue_min = cv2.getTrackbarPos("hue_min", "TrackBar")
hue_max = cv2.getTrackbarPos("hue_max", "TrackBar")
sat_min = cv2.getTrackbarPos("sat_min", "TrackBar")
sat_max = cv2.getTrackbarPos("sat_max", "TrackBar")
val_min = cv2.getTrackbarPos("val_min", "TrackBar")
val_max = cv2.getTrackbarPos("val_max", "TrackBar")
```

7. Створення масиву NumPy

Ми створюємо два масиви (lower – для мінімальних значень та upper – для максимальних значень). Дані масиви необхідні для зберігання мінімальних/максимальних значень по трьом параметрам налаштування: Hue, Saturation, Value.

```
lower = np.array([hue_min, sat_min, val_min])
upper = np.array([hue_max, sat_max, val_max])
```

8. Створення маски на HSV зображенні

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 17

Створення бінарної маски відбувається за допомогою команди `inRange()` для виділення області в білий колір, що відповідає налаштованим параметрам, а зайве – в чорний. Параметри, що необхідні для даної функції: змінна, в яку занесено конвертацію BGR в HSV, масиви `lower` та `power`.

```
mask = . . .
```

9. Знаходимо контури на бінарній масці

Створюємо дві змінні в які буде записана інформація про всі виявленні контури (список контурів) – `cnts` та інформацію про ієрархію контурів – `hierarchy`. В параметрах функції вказуємо змінну в яку занесена інформація бінарної маски, метод виявлення тільки зовнішніх контурів – `cv2.RETR_EXTERNAL` та апроксимація, але в даному випадку важливо вказати, що нам не потрібно виконувати спрощення контурів, топко ми маємо зберегти всі точки, що утворюють контур – `cv2.CHAIN_APPROX_NONE`. Структурно даний рядок коду має наступний вигляд:

Змінна 1, Змінна 2 = Функція пошуку контурів (Бінарна маска, Метод пошуку зовнішніх контурів, Метод апроксимації)

10. Обробка знайдених контурів

1. Проходимося по знайденим контурам, отримуємо площу поточного контуру та перевіряємо, чи площа контуру більша за 300.

```
for c in cnts:
    # Підрахунок площі контуру
    area = ...
    # Перевірка, чи площа більше за 300
    ...
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 18

2. Обчислення периметра контуру

```
# Обчислення периметра контуру
peri = cv2.arcLength(c, True)
```

Периметр контуру – це довжина всіх сторін контуру. Ця величина допоможе визначити, наскільки складною є форма об’єкта. Вона використовується як один із критеріїв для визначення типу фігури.

True вказує, що контур є замкнутим, а c – контур, площа якого становить більше 300.

3. Апроксимація контуру

```
# Апроксимація контуру для визначення кількості вершин (углів)
approx = cv2.approxPolyDP(c, 0.02 * peri, True)
```

$(0.02 * \text{peri})$ – коефіцієнт спрощення, що визначає, наскільки точно має бути апроксимований контур. Більший коефіцієнт призводить до меншої кількості вершин (спрощення), а менший коефіцієнт – до більшої кількості вершин (більш точна апроксимація).

Після апроксимації контуру, змінна approx містить список точок, які складають апроксимований контур, в залежності від кількості цих точок буде встановлено тип фігури.

4. Створює області виділення контуру (boundingRect)

```
x, y, w, h = cv2.boundingRect(c)
cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
```

5. Наносимо текст, який буде відображати кількість точок кутів

```
cv2.putText(img, "Points:" + str(len(approx)), (x+w+20, y+h+20), cv2.FONT_HERSHEY_COMPLEX, 0.7, (0, 255, 0), 2)
```

6. Створюємо умови перевірки кількості кутів для визначення типу фігури та підписуємо їх відповідно (всі умови навести за допомогою оператора elif/else)

```
if len(approx) == 4:
    cv2.putText(img, "Rectangle:" + str(len(approx)), (x+w+20, y+h+45), cv2.FONT_HERSHEY_COMPLEX, 0.7, (0, 255, 0), 2)
```

– len – команда для визначення довжини змінної approx;

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 19

- `img` – кадр;
- `str` – тип змінної (текст);
- `cv2.FONT_HERSHEY_COMPLEX` – шрифт, який використовується при додаванні тексту до зображення за допомогою бібліотеки OpenCV. Вона вказує на використання складного (комплексного) шрифту для відображення тексту на зображенні.

11.Виводимо на екран 3 вікна з результатами (`img`, `hsv`, `mask`)

12.Пропишуємо вихід з програми

Результати виконання програми:

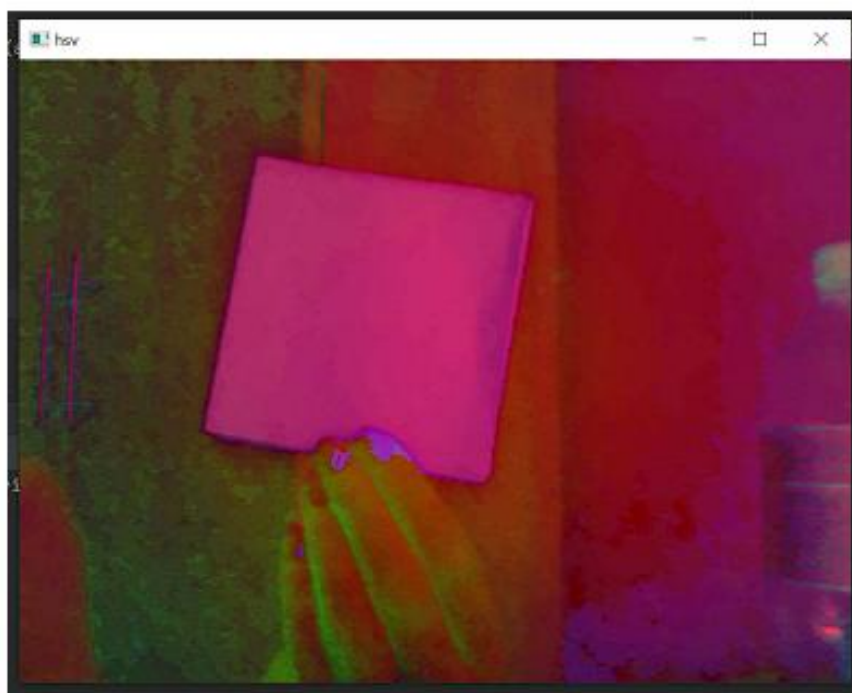


Рис.2.5. Вікно результату виконання конвертації BGR в HSV

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОК31-2022
	Екземпляр № 1	Арк 33 / 20

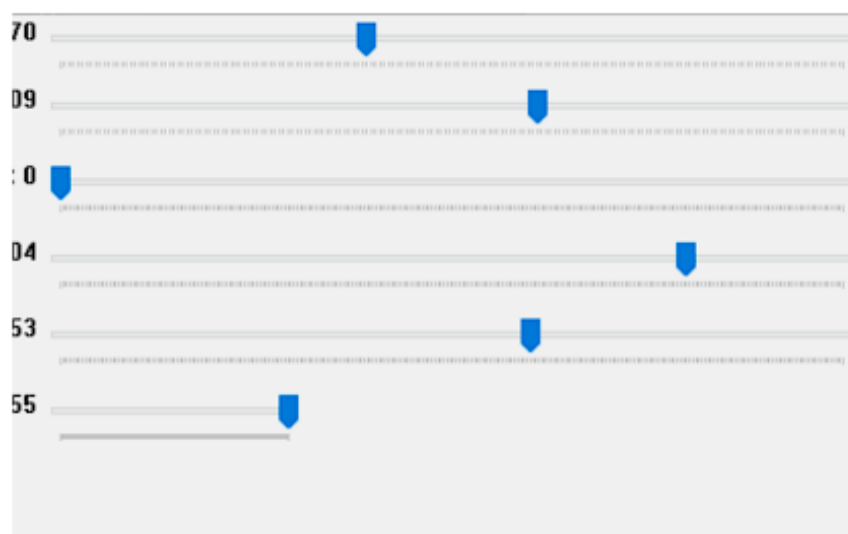


Рис.2.6. Налаштування TrackBar для виявлення об'єкту в руках\

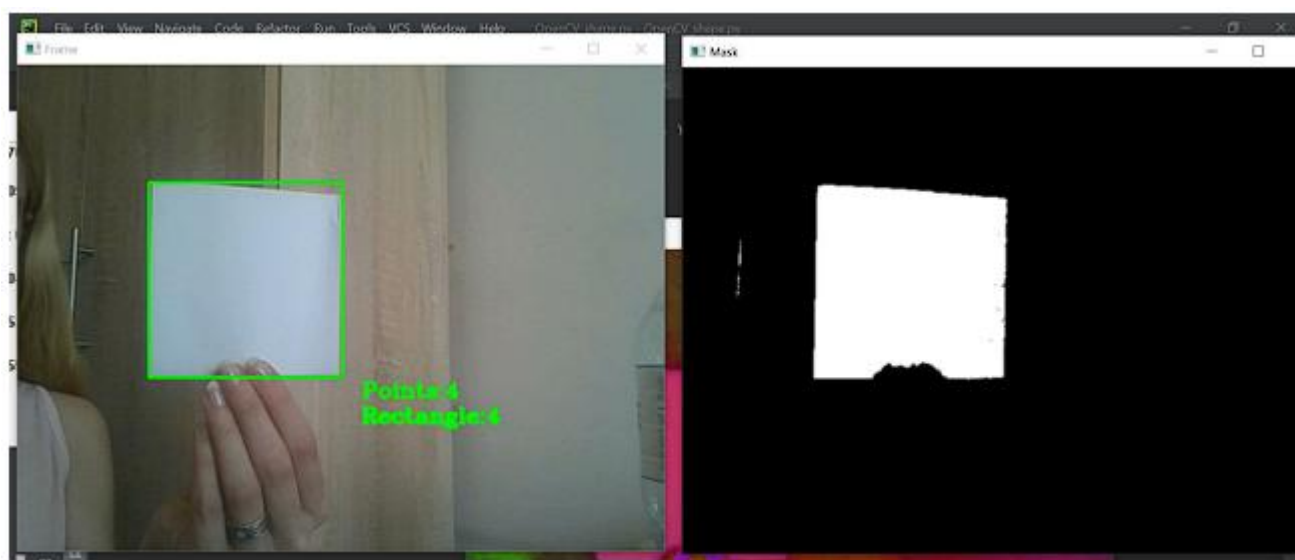


Рис.2.7. Створення бінарної маски та фінальний результат виконання програми

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 21

Лабораторна робота №3

ВИЯВЛЕННЯ QR-code ЗА ДОПОМОГОЮ OPENCV

Мета роботи: Продовження ознайомлення з бібліотекою OpenCV та отримати практичні навички у написанні програмного коду розпізнавання QR-code на відеопотоках.

Теоретичні відомості

Виявлення QR-кодів за допомогою OpenCV - це процес розпізнавання та декодування QR-кодів (Quick Response codes) на візуальних зображеннях або відеопотоках. QR-коди - це двомірні штрих-коди, які зазвичай містять інформацію у вигляді тексту, URL-посилання, номера телефону тощо.

Основні компоненти виявлення QR-кодів за допомогою OpenCV включають наступні етапи:

1. Зчитування зображення або відеопотоку: Ви відкриваєте вхідний потік даних, який може бути фотографією, відеокамерою або відеофайлом.
2. Перетворення в зображення зі спектральними даними: Оскільки QR-коди зазвичай виявляються в чорно-білому кольорі, потрібно перетворити кольоровий кадр або зображення в чорнобілий (градації сірого).
3. Аналіз QR-кодів: OpenCV сприяє виявленню різних об'єктів на зображенні, але для QR-кодів ви можете використовувати спеціальну бібліотеку, таку як pyzbar, для виявлення та декодування QR-кодів на зображенні. Коды декодуються для отримання текстової інформації, яка зазвичай міститься в QR-коді.
4. Відображення результатів: Після виявлення та декодування QR-кодів, ви можете відобразити отриману інформацію або виділити QR-коди на вихідному зображенні або відеопотоці.

Застосування: маркетинг/реклама, збір інформації, автоматизація, комерційна логістика, медицина, робототехніка і т.д.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 22

Порядок виконання роботи

1. Імпортування необхідних бібліотек

- OpenCV – для читання та обробки зображень.
- NumPy – для роботи з масивами.
- pyzbar.pyzbar as pyzbar – для виявлення і декодування QR-кодів.

2. Підключення веб-камери (називаємо змінну cap)

3. Створення змінної (font) яка буде використовуватися для відображення тексту на зображенні

4. Створюємо цикл

В циклі прописуємо зчитування кадру cap з двома змінними: одна для отримання статусу читання відеопотоку, інша записує зчитаний кадр.

```
while True:
```

5. Виявлення QR-code

```
decodedObjects = pyzbar.decode(frame)
for obj in decodedObjects:
```

decodedObjects = pyzbar.decode(frame) – для виявлення і декодування QR-code на певному кадрі. Результат зберігається у змінній decodedObjects, яка містить інформацію про всі виявлені QR на кадрі.

Цикл for obj in decodedObjects: – для перебору всіх виявлених QR у decodedObjects.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 23

6. В циклі наносимо текст, який буде відображати посилання коду

Для оформлення тексту використовуємо `str(obj.data)` – це атрибут, який містить текстові дані, що були декодовані. В загальному для нанесення тексту необхідно використати 7 параметрів ф-ї:

1. Кадр;
2. Текстові данні;
3. Позиція (x, y);
4. Шрифт; 5. Розмір шрифту;
6. Колір формату BGR;
7. Товщина лінії.

7. Виводимо на екран вікно з результатом

8. Прописуємо вихід з програми.

Результати виконання програми:



Рис.3.1. Вікно результату програми зчитування QR-code

*Самостійно організувати автоматичне перенесення на сайт зчитуваного QR-code у web вікні.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 24

Лабораторна робота №4

ВІДСЛІДКОВУВАННЯ ЗА КОЛЬОРОМ, ВИКОРИСТОВУЮЧИ OPENCV

Мета роботи: Продовження ознайомлення з бібліотекою OpenCV та отримати практичні навички у написанні програмного коду відслідковування кольорових об'єктів на відеопотоках.

Теоретичні відомості

Відслідковування об'єктів за допомогою OpenCV – це процес визначення руху та визначення позиції об'єктів на послідовних кадрах відео. Цей процес може бути використаний для різних завдань, таких як відеоспостереження, аналіз руху, віртуальна реальність, і багато інших.

Основні етапи відслідковування об'єктів включають:

1. Вибір об'єкта відслідковування – даний етап включає в себе виділення цільового об'єкта на початковому кадрі відеопотоку. Виділення може бути ручним або автоматичним за алгоритмами сегментації.

2. Ініціалізація алгоритму відслідковування – бібліотека OpenCV має кілька алгоритмів відслідковування, таких як MOSSE, KLT, GOTURN тощо.

MOSSE (Minimum Output Sum of Squared Error)

Це швидкий і легкий алгоритм відслідковування, який базується на мінімізації суми квадратних помилок на вихідних даних. Даний алгоритм підходить для відслідковування швидких та великих об'єктів. Недоліком даного алгоритму є значні похибки відслідковування об'єктів зі значними змінами форми або освітленням.

KLT (Kanade-Lucas-Tomasi)

Tracker Це алгоритм відслідковування, який базується на виборі важливих точок на об'єкті та відстежуванні їх переміщення на кожному кадрі. Використовується даний алгоритм для відслідковування точок на об'єктах, що змінюють форму та рухаються складно, тобто для об'єктів зі складними рухами та

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 25

змiнами форми. Недолiками даного алгоритму може бути обчислювально витратним i менш стiйким до великих змiн освiтлення.

CSRT (Discriminative Correlation Filter with Channel and Spatial Reliability)

Це сучасний алгоритм відслідковування, який використовує фільтри кореляції для визначення руху об'єкта на послідовних кадрах. Він також враховує просторові та кольорові властивості. Даний алгоритм підходить для відслідковування об'єктів зі змінною формою, різними умовами освітлення та частковою прихованістю. Недоліком даного алгоритму є швидкодія (CSRT може бути трохи повільнішим, ніж MOSSE, через більш складний алгоритм). Також популярними алгоритми відслідковування є: MedianFlow, TLD, GOTURN (єдиний алгоритм на основі глибокого навчання), BOOSTING, MIL

3. Зчитування кадрів та обробка даних:

- Визначення позиції об'єкта на кадрі, шляхом порівняння ознак об'єкта на кожному кадрі з ознаками на початковому кадрі.
- Оновлення позиції на кожному кадрі. Ця інформація може бути використана для подальшого аналізу або управління системою.
- Обробка помилок та аналіз витрат відслідковування. Якщо обраний алгоритм не може визначити позицію об'єкта на певному кадрі (наприклад, через затінення або часткове блокування об'єкта), система повинна коректно обробити отримані дані, відповідно поставлених задач.

4. Візуалізація результатів – це може бути відображенням рамки навколо об'єкта або виведенням інформації про позицію об'єкта

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 26

Порядок виконання роботи

1. Імпортування необхідних бібліотек

- OpenCV – для читання та обробки зображень.
- NumPy – для роботи з масивами.

2. Підключення веб-камери (називаємо змінну cap)

3. Ініціалізація змінних для обчислення середнього положення об'єкта

```
x_medium = 0
y_medium = 0
```

4. Створюємо цикл

В циклі прописуємо зчитування кадру cap з двома змінними: одна для отримання статусу читання відеопотоку, інша записує зчитаний кадр. Також прописуємо перетворення кадру BGR формату в кольоровий простір HSV.

```
while True:
```

5. Встановлення та виявлення кольору

Встановлюємо масиви верхніх/нижніх границь необхідного нам кольору, після чого створюємо бінарну маску для обраних лімітів. Створення маски за допомогою функції inRange() призведе до створення зображення, де колір об'єктів, що потрапляють в діапазон, буде білим, а решта – чорним.

6. Знаходимо контури на зображенні в масці

В даному випадку необхідно використовувати метод виявлення ієрархії контурів та апроксимацію для економії пам'яті та спрощення контурів.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 27

7. Сортування контурів

Для мінімізації похибок відслідковування необхідно відсортувати контури в порядку спадання за їх площею. У відеоспостереженні часто важливо визначити найбільший об'єкт, оскільки він, як правило, є головним об'єктом інтересу, тобто об'єктом для подальшого відслідковування.

```
contours = sorted(contours, key=lambda x:cv2.contourArea(x), reverse=True)
```

- sorted() – функція сортування;
- contours – список контурів;
- key=lambda x: cv2.contourArea(x) – метод сортування, в даному випадку використовується лямбда-функція яка приймає контур «x» і повертає його площу, обчислену функцією cv2.contourArea(x);
- reverse – цей параметр визначає порядок сортування:
 - reverse=False – сортування в порядку зростання.
 - reverse=True – сортування в порядку спадання.

8. Цикл перебору контурів для отримання необхідного контуру

В циклі прописуємо отримання координат та розміру оминаючого прямокутника для обраного контуру (boundingRect), після чого обчислюємо середнє положення об'єкта для отримання точки відслідковування. Цикл закінчуємо командою break.

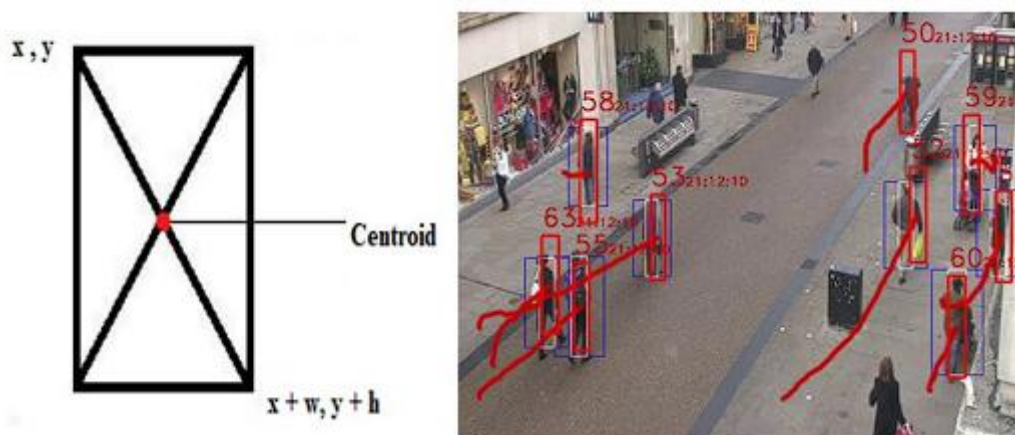


Рис.4.1. Схема відслідковування об'єкта за середньою точкою

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 28

9. Відображення положення об'єкта

В даній роботі будемо використовувати найпростіший спосіб відображення – малювання вертикальної та горизонтальної лінії на кадрі для показу середнього положення об'єкта.

```
cv2.line(frame, (x_medium, 0), (x_medium, 480), (0, 255, 0), 2)
cv2.line(frame, (0, y_medium), (640, y_medium), (0, 255, 0), 2)
```

10. Відображаємо кадр та бінарне зображення (маска)

11. Прописуємо вихід з програми

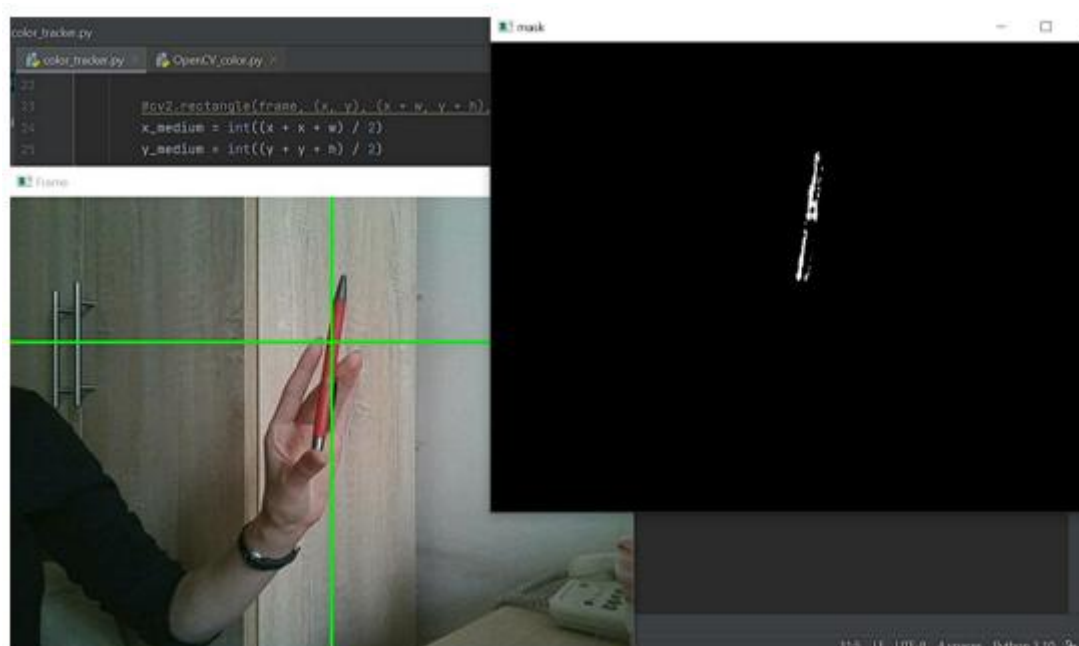


Рис.4.2. Результат виконання програми

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 29

Лабораторна робота №5

РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ TEMPLATE MATCHING В OPENCV

Мета роботи: ознайомлення з одним із базових методів комп'ютерного зору – шаблонним зіставленням (template matching) та отримання практичних навичок пошуку зразків (шаблонів) об'єктів на зображеннях з використанням бібліотеки OpenCV.

Теоретичні відомості

Template Matching (шаблонне зіставлення) — це метод аналізу зображень, який використовується для знаходження області в зображенні, що найбільш подібна до заданого зразка (рис.5.1.). Даний метод базується на ковзанні шаблону по всьому зображенню і обчисленні міри подібності між шаблоном і поточною ділянкою зображення на кожному положенні. Template Matching є детермінованим методом і не використовує ознак або машинного навчання, що робить його ефективним для простих завдань, де масштаб, орієнтація та освітлення об'єкта не змінюються.



Рис. 5.1. Template Matching

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 30

У бібліотеці OpenCV реалізація шаблонного зіставлення здійснюється за допомогою функції `cv2.matchTemplate()`, яка приймає вхідне зображення, шаблон та метод обчислення збігу, і повертає карту відповідності (матрицю подібності). Кожен елемент цієї карти вказує на ступінь схожості між шаблоном та відповідною частиною зображення. Для визначення координат області з найкращим збігом використовується функція `cv2.minMaxLoc()`, яка повертає положення мінімального та максимального значення у матриці збігів.

В OpenCV передбачено кілька методів для обчислення подібності між шаблоном та фрагментом зображення, які можна задати параметром `method` у функції `matchTemplate()`.

Табл. 5.1.

Методи та їх опис

Метод	Повна назва (англ.)	Принцип роботи	Критерій найкращого збігу
<code>cv2.TM_SQDIFF</code>	Squared Difference	Обчислюється сума квадратів різниці пікселів	Мінімальне значення
<code>cv2.TM_SQDIFF_NORMED</code>	Normalized Squared Difference	Те саме, але з нормалізацією	Мінімальне значення
<code>cv2.TM_CCORR</code>	Cross Correlation	Визначає схожість через кореляцію пікселів	Максимальне значення
<code>cv2.TM_CCORR_NORMED</code>	Normalized Cross Correlation	Те саме, з нормалізацією	Максимальне значення
<code>cv2.TM_CCOEFF</code>	Correlation Coefficient	Вираховує статистичну залежність (коефіцієнт кореляції)	Максимальне значення
<code>cv2.TM_CCOEFF_NORMED</code>	Normalized Correlation Coefficient	Те саме, з нормалізацією	Максимальне значення

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 31

Методи, що базуються на обчисленні різниці (TM_SQDIFF, TM_SQDIFF_NORMED), визначають подібність на основі розбіжності між шаблоном та зображенням: чим менша різниця, тим краще збіг. Натомість методи, що базуються на кореляції (TM_CCORR, TM_CCOEFF), відображають ступінь подібності між пікселями і досягають найкращого результату при максимально можливому значенні. Методи без нормалізації чутливі до варіацій яскравості, тоді як нормалізовані методи (_NORMED) дозволяють отримувати стабільніші результати в умовах зміни освітлення.

Вибір методу залежить від характеру задачі. У більшості випадків рекомендується використовувати cv2.TM_CCOEFF_NORMED як один з найстійкіших до фонових відхилень і загального освітлення.

Порядок виконання роботи

1. Імпортування необхідних бібліотек

OpenCV (cv2) використовується для роботи із зображеннями, NumPy (np) — для чисельних обчислень.

2. Завантаження зображення та шаблону

```
img = cv2.imread('scene.jpg')
template = cv2.imread('template.jpg')
h, w = template.shape[:2]
```

- scene.jpg — основне зображення;
- template.jpg — шаблон;
- h, w — висота та ширина шаблону (знадобиться для побудови прямокутника).

3. Застосування функції matchTemplate

```
result = cv2.matchTemplate(img, template, cv2.TM_CCOEFF_NORMED)
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 32

Пошук збігів методом нормалізованого коефіцієнта кореляції. Метод визначає, наскільки добре шаблон співпадає з частиною зображення.

4. Знаходження координат найкращого збігу

```
min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(result)
top_left = max_loc
bottom_right = (top_left[0] + w, top_left[1] + h)
```

Знаходження координат області, де шаблон найбільше збігається із зображенням. Для `TM_CCOEFF_NORMED` використовуємо `max_loc`.

5. Відображення результату

Після виконання коду на зображенні з'являється прямокутник, який виділяє область, де знайдено збіг з шаблоном.

Завдання

1. Вибрати зразок (шаблон) з реального зображення (наприклад, монету або логотип).

2. Виконати шаблонне зіставлення за двома методами:

```
cv2.TM_CCOEFF_NORMED
cv2.TM_SQDIFF_NORMED
```

3. Порівняти результати та координати.

4. Надіслати:

- вихідне зображення;
- шаблон;
- зображення з результатом;
- короткий звіт (3–5 речень).

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.10- 05.01/151.00.1/Б /ОКЗ1-2022
	Екземпляр № 1	Арк 33 / 33

Список інформаційних джерел

1. OpenCV Documentation [Електронний ресурс]. – Режим доступу: <https://docs.opencv.org> – Назва з екрана. – Дата звернення: 27.05.2025.
2. OpenCV – GitHub Repository [Електронний ресурс] // GitHub. – Режим доступу: <https://github.com/opencv/opencv> – Дата звернення: 27.05.2025.
3. Rosebrock A. Practical Python and OpenCV: An Introductory, Example-Driven Guide to Image Processing and Computer Vision / Adrian Rosebrock. – 4th ed. – PyImageSearch, 2019. – 186 p. – ISBN: 978-1-71633-541-2.
4. Howse J., Beyeler M. OpenCV 4 with Python Blueprints: Build Smart Applications with Real-world Use Cases / Joseph Howse, Michael Beyeler. – 2nd ed. – Birmingham: Packt Publishing, 2019. – 234 p. – ISBN: 978-1-78913-107-6.