

ПРОГРАММА

ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ

**“Python-разработка для БПЛА: паттерны проектирования, API-интерфейсы и библиотеки для оптимизации решений”**

**Задание 13.**

**Работа с репозиториями Git на GitLab**

Преподаватель курса

**Максимов Егор Васильевич**

Выполнил

**Зубков Евгений Юрьевич**

**№ потока БПЛА\_256-1, группа 1**

**Москва 2024 г.**

**Задание 13. Работа с репозиториями Git на GitLab**

**Инструкция:**

Шаг 1. Импортировать библиотеку OpenCV и загрузить изображение с камеры БПЛА.

Шаг 2. Применить метод cv2.resize() для уменьшения разрешения изображения в 2 раза. Это позволит ускорить дальнейшую обработку.

Шаг 3. Использовать метод cv2.Canny() для выделения контуров на изображении и поиска нужных объектов. Подобрать оптимальные параметры для конкретной задачи.

Шаг 3. Применить метод cv2.cvtColor() для перевода изображения в оттенки серого. Это уменьшит объем данных.

Шаг 3. Сохранить оптимизированное изображение и измерить время обработки. Сравнить с изначальным вариантом.

import cv2  
import time  
  
# загружаем изображение с камеры БПЛА  
cap = cv2.VideoCapture(0) # используем первую подключенную камеру  
ret, frame = cap.read() # читаем кадр с камеры  
  
if not ret:  
 print("Ошибка: не удалось захватить изображение с камеры.")  
 cap.release()  
 exit()  
  
# Уменьшаем разрешение изображения в 2 раза  
start\_time = time.time()  
height, width = frame.shape[:2]  
resized\_frame = cv2.resize(frame, (width // 2, height // 2))  
  
# Применяем метод Canny   
edges = cv2.Canny(resized\_frame, 100, 200)  
  
# изображение в оттенки серого  
gray\_frame = cv2.cvtColor(resized\_frame, cv2.COLOR\_BGR2GRAY)  
  
# Сохраняем оптимизированное изображение   
cv2.imwrite("optimized\_image.jpg", gray\_frame)

# измеряем время обработки  
end\_time = time.time()  
processing\_time = end\_time - start\_time  
print(f"Время обработки изображения: {processing\_time:.4f} секунд")  
  
# сравнения  
cv2.imshow("Оригинальное изображение", frame)  
cv2.imshow("Уменьшенное изображение", resized\_frame)  
cv2.imshow("Контуры (Canny)", edges)  
cv2.imshow("Оттенки серого", gray\_frame)  
  
# Ждем нажатия любой клавиши, затем закрываем окна  
cv2.waitKey(0)  
cv2.destroyAllWindows()  
  
cap.release()