Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №1**

**Дисциплина: Алгоритмы цифровой обработки мультимедиа**

Работу выполнила: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. А. Сапунова

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. С. Жук

Цель работы: Знакомство и с библиотекой OpenCV.

Задание 1. Установить библиотеку OpenCV

Для установки библиотеки используется команда: pip install opencv-contrib-python. Для работы необходимо включить библиотеку в проект директивой: import cv2.

Задание 2. Вывести на экран изображение. Протестировать три возможных расширения, три различных флага для создания окна и три различных флага для чтения изображения.

Для загрузки картинки в программу воспользуемся функцией cv.imread(filename[, flags]) -> retval. Данная функция возвращает двумерную матрицу, хранящую изображение в формате, считанном из файла. Формат изображения определяется самим изображением, а не расширением файла.

Примеры работы с различными флагами приведены на рис. 1–3.



Рисунок 1 –Чтение изображения без флагов



Рисунок 2 –Чтение изображения с флагом IMREAD\_GRAYSCALE

Изображение выглядит как внутренний, несколько

Автоматически созданное описание

Рисунок 3 –Чтение изображения с флагом IMREAD\_REDUCED\_COLOR\_4

Для создания именованного окна с указанным именем и различными флагами для отображения картинки используется: cv.namedWindow(winname[, flags]) ->None. Для отображения окна используется: cv.imshow(winname, mat) ->None.

При создании именованного окна использовались флаги:

* WINDOW\_NORMAL - позволяет пользователю изменять размер окна, переключиться в полноэкранный режим;
* WINDOW\_AUTOSIZE - пользователь не может менять размер окна, размер ограничен отображаемым изображением;
* WINDOW\_FREERATIO – изображение растягивается настолько, насколько это возможно (без ограничения соотношения).
* WINDOW\_ KEEPRATIO - соотношение изображения соблюдается.

Задание 3. Отобразить видео в окне. Рассмотреть методы класса VideoCapture и попробовать отображать видео в разных форматах, в частности размеры и цветовая гамма.

За работу с видео потоком отвечает класс VideoCapture.

Для отображения видео создается цикл. В первой команде прочитаем кадр из видео потока с помощью функции cap.read(). Эта функция возвращает два значения, первое значение ret – булевское значение, обозначающее, удалось ли выполнить чтение кадра. Сам кадр называем фреймом и сохраняем в формат картинки (двумерная матрица). Если изображение закончилось, ret вернет false и отображение завершится. Далее отображаем полученный фрейм (Рис. 4-5).

Изображение выглядит как текст, растение, цветок, маргаритка

Автоматически созданное описание

Рисунок 4 – Отображение видео

Изображение выглядит как текст, растение

Автоматически созданное описание

Рисунок 5 – Отображение видео в черно-белом формате

Задание 4. Записать видео из файла в другой файл

Для записи видео в файл нужно определить размер исходного изображения с помощью cv2.CAP\_PROP\_FRAME\_WIDTH и cv2.CAP\_PROP\_FRAME\_HEIGHT и fourcc код с помощью cv2.VideoWriter\_fourcc(\*'XVID'). Далее создаем видео поток на запись, класс VideoWriter. Далее считывается каждый кадр видео и записывается в новый файл с помощью функции write(const Mat&image) класса VideoCapture (Рис. 6).

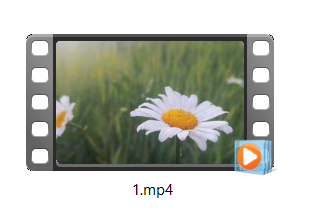


Рисунок 6 – Записанный файл

Задание 5. Отобразить информацию с вебкамеры, записать видео, продемонстрировать видео на следующем занятии.

Для записи видео с вебкамеры используется VideoCapture(1). Для сохранения записанного результата используется Задание 4 (Рис. 7).

Изображение выглядит как дерево, внешний, парк

Автоматически созданное описание

Рисунок 7 – Запись видео с вебкамеры

Задание 6. Подключите телефон, подключитесь к его камере, протранслируйте запись с камеры. Продемонстрировать процесс на ноутбуке преподавателя и своем телефоне.

**Листинг**

import cv2 # task1  
import numpy as np  
import requests  
import imutils  
  
  
# task 2  
#  
  
# IMREAD\_GRAYSCALE  
# IMREAD\_ANYDEPTH  
# cv2.IMREAD\_REDUCED\_COLOR\_4  
def read\_photo():  
 img = cv2.imread(r'.\1604650136\_1.jpg')  
 # WINDOW\_NORMAL  
 # WINDOW\_AUTOSIZE  
 # WINDOW\_FREERATIO  
 cv2.namedWindow('Display window')  
 cv2.imshow('Display window', img)  
 cv2.waitKey(0)  
 cv2.destroyAllWindows()  
  
  
# task 3  
# COLOR\_RGB2HSV COLOR\_BGR2HSV  
def read\_video():  
 cap = cv2.VideoCapture(r'.\1.mp4', cv2.CAP\_ANY)  
 cv2.namedWindow('frame', cv2.WINDOW\_NORMAL)  
 cv2.resizeWindow("frame", 500, 500)  
  
 while True:  
 ret, frame = cap.read()  
 if not ret:  
 break  
  
 # cv2.imshow('frame', frame)  
  
 # Черно-белое  
 gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)  
 cv2.imshow('frame', gray)  
  
 if cv2.waitKey(10) & 0xFF == 27:  
 break  
  
 cap.release()  
 cv2.destroyAllWindows()  
  
  
# task 4  
def read\_video\_write\_to\_file():  
 video = cv2.VideoCapture(r'.\1.mp4', cv2.CAP\_ANY)  
  
 w = int(video.get(cv2.CAP\_PROP\_FRAME\_WIDTH))  
 h = int(video.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))  
 fourcc = cv2.VideoWriter\_fourcc(\*'XVID')  
 video\_writer = cv2.VideoWriter("output.mov", fourcc, 25, (w, h))  
  
 while True:  
 ok, img = video.read()  
 if not ok:  
 break  
  
 video\_writer.write(img)  
 if cv2.waitKey(1) & 0xFF == ord('q'):  
 break  
  
 video.release()  
 cv2.destroyAllWindows()  
  
  
# task 5  
def read\_ip\_write\_to\_file():  
 video = cv2.VideoCapture(0)  
 w = int(video.get(cv2.CAP\_PROP\_FRAME\_WIDTH))  
 h = int(video.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))  
 fourcc = cv2.VideoWriter\_fourcc(\*'XVID')  
 video\_writer = cv2.VideoWriter("output.mov", fourcc, 25, (w, h))  
  
 while True:  
 ok, img = video.read()  
 cv2.imshow('img', img)  
 video\_writer.write(img)  
 if cv2.waitKey(1) & 0xFF == ord('q'):  
 break  
  
 video.release()  
 cv2.destroyAllWindows()  
  
  
def print\_cam():  
 cap = cv2.VideoCapture(0)  
 cap.set(3, 640)  
 cap.set(4, 480)  
 while True:  
 ret, frame = cap.read()  
 gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)  
  
 cv2.imshow('frame', gray)  
 if cv2.waitKey(1) & 0xFF == 27:  
 break  
  
 cap.release()  
 cv2.destroyAllWindows()  
  
  
# task 6  
def phone\_camera():  
 video = cv2.VideoCapture(1)  
  
 w = int(video.get(cv2.CAP\_PROP\_FRAME\_WIDTH))  
 h = int(video.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))  
 fourcc = cv2.VideoWriter\_fourcc(\*'XVID')  
 video\_writer = cv2.VideoWriter("output2.mov", fourcc, 25, (w, h))  
  
 while True:  
 rez, img = video.read()  
 cv2.imshow('Camera', img)  
 video\_writer.write(img)  
  
 if cv2.waitKey(1) & 0xFF == ord('q'):  
 break  
  
 # cv2.destroyAllWindows()  
 # cv2.waitKey(0)  
  
 video.release()  
 cv2.destroyAllWindows()