|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

**Факультет «Информатика и системы управления»**

**Кафедра «Системы обработки информации и управления»**

Отчет по Домашнему заданию

«Распознавание и отслеживание людей в видеопотоке с видеокамеры в реальном масштабе времени. Интерактивное замена фона статическим изображением (3 разных фоновых изображений).»

по дисциплине «Технология мультимедиа»

Выполнил:

студент группы ИУ5-63 Соколов И.В.

подпись, дата

Проверил:

к.т.н., доц., Афанасьев Г.И.

подпись, дата

Москва, 2024

**Задание**

Распознавание и отслеживание людей в видеопотоке с видеокамеры в реальном масштабе времени. Интерактивное замена фона статическим изображением (3 разных фоновых изображений).

**Краткое описание алгоритма**

Нужно обнаружить человека в видеопотоке, после взять его маску (пискели, в которых он находится). Закрашивать фоном те пиксели, которые = 0 (т.е. их нейросеть не распознала)

Для решения задачи работы с камерой воспользуюсь библиотекой openCV.

Для решения задачи распознавания людей воспользуюсь библиотекой Yolo, которая предоставляет готовую нейронную сеть для сегментации.

**Подключаю камеру**

cap = cv2.VideoCapture(0)

**Читаю видеопоток с камеры и отображаю его на экране**.

Процесс продолжается пока не будет выключена камера или пользователь не нажмет клавишу q (quit – выйти).

while True:

cv2.imshow('Custom background', img)

key = cv2.waitKey(10)

if key == ord('q'):

break

**Размер окна и изображения**

Задам фиксированный размер окна - 480 на 640:

cap.set(3, 640)

cap.set(4, 480)

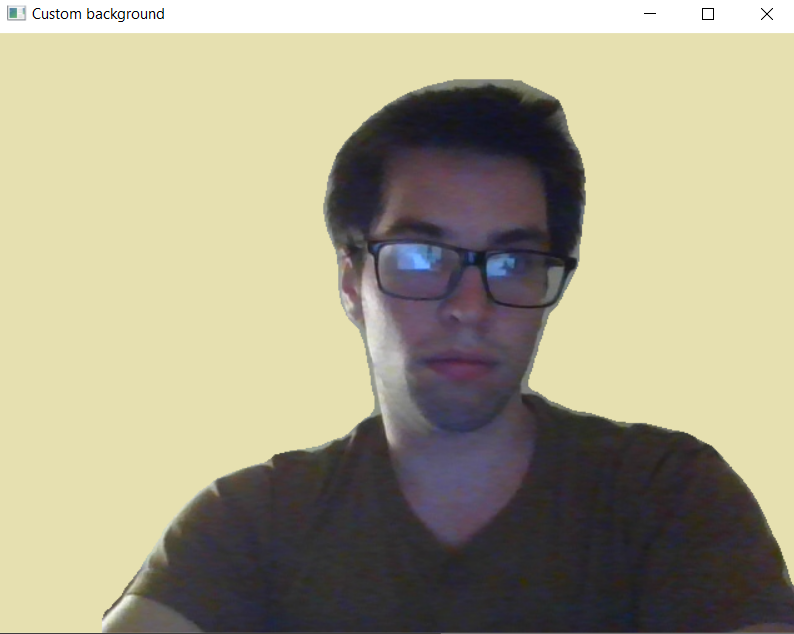
**Отслеживание человека с помощью модели Yolo:**

Инициализирую модель ML:

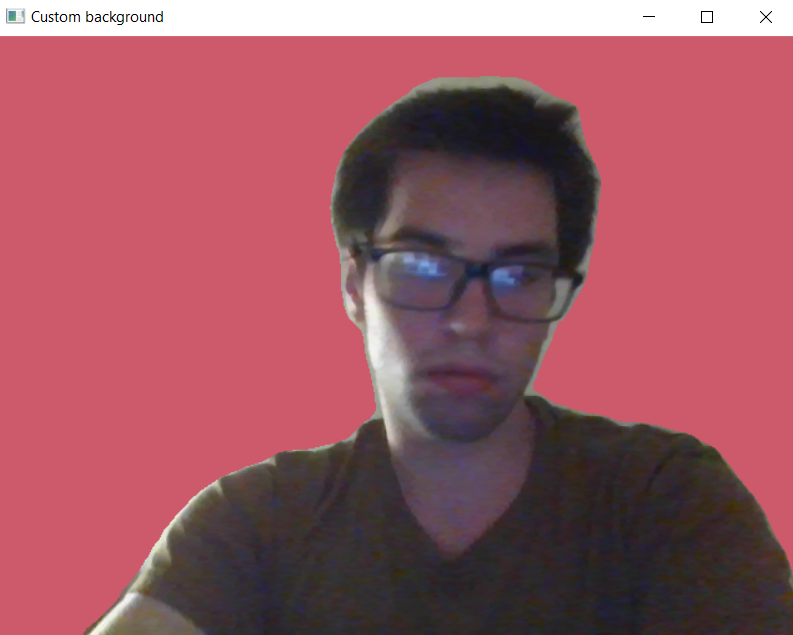
model = YOLO('yolov8n-seg.pt')

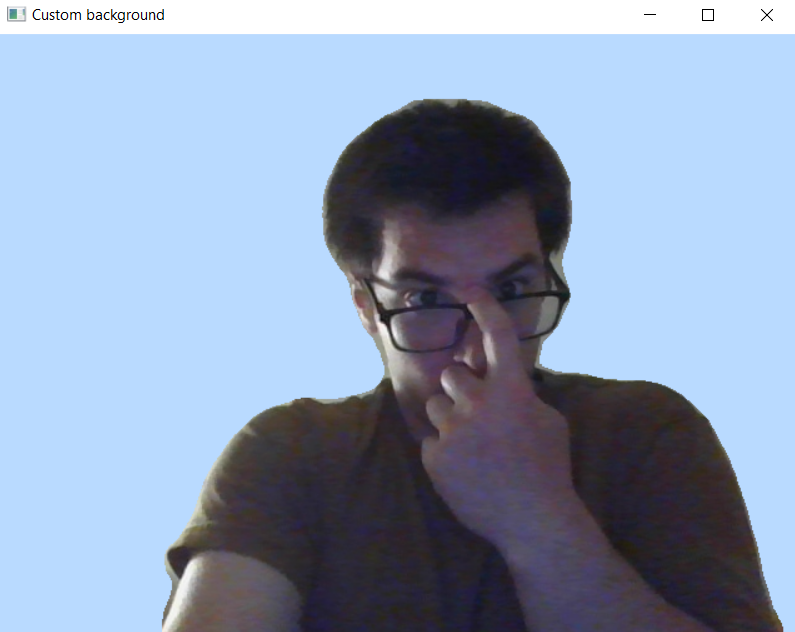
Воспользуюсь ей в цикле, чтобы обработать каждый кадр. Если люди распознаны, возьму массив масок. Если в кадре не распознаны люди, не буду модифицировать изображение.

**Результаты**

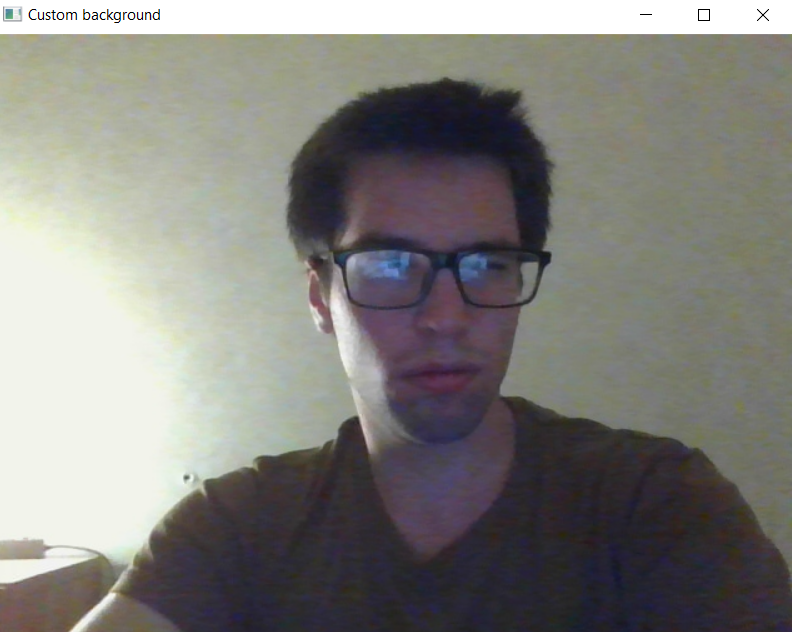


Скриншот 1. Первый фон.

Скриншот 2. Второй фон



Скриншот 3. Третий фон



Скриншот 4. Отключение фона

**Листинг программы**

from ultralytics import YOLO

import cv2

import math

import numpy as np

import torch

model = YOLO('yolov8n-seg.pt')

PowderBlue =np.array([ 176, 224, 230 ])

SlateBlue= np.array([ 106, 90, 205 ])

PeachPuff = np.array([255, 218, 185])

colors = [PowderBlue, SlateBlue,PeachPuff ]

color\_flag = None

cap = cv2.VideoCapture(0)

cap.set(3, 640)

cap.set(4, 480)

while True:

success, img = cap.read()

results = model(img, stream=True, classes = 0)

if color\_flag!=None:

for r in results:

if r.masks == None:

continue

boxes = r.boxes.data

masks = r.masks.data

clss = boxes[:, 5]

people\_indices = torch.where(clss == 0)

people\_masks = masks[people\_indices]

people\_mask = torch.any(people\_masks, dim=0).int()

background\_mask = 1 - people\_mask

channel1 = np.zeros((480,640))

channel2 = np.zeros((480,640))

final\_mask = np.hstack((background\_mask.numpy(), channel1, channel2))

img[ background\_mask.bool() ] = np.ones((1,1,1)) \* colors[color\_flag]

cv2.imshow('Custom background', img)

key = cv2.waitKey(10)

if key == ord('q'):

break

elif key == ord('0'):

color\_flag = None

elif key == ord('1'):

color\_flag = 0

elif key == ord('2'):

color\_flag = 1

elif key == ord('3'):

color\_flag = 2

cap.release()

cv2.destroyAllWindows()

**Выводы**

Мне удалось выполнить задачу распознавания людей и последующей установки фона на потоковом изображении в реальном времени с использованием популярной библиотеки openCV и модели нейронной сети из библиотеки Yolo.