###### МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

###### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

###### НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

###### Факультет информационных технологий

**Кафедра параллельных вычислений**

ОТЧЕТ

О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

«ВЫСОКОУРОВНЕВАЯ РАБОТА С ПЕРИФЕРИЙНЫМИ УСТРОЙСТВАМИ»

студента 2 курса, группы 21209

Усольцева Антона Андреевича

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

М.А.Мичуров

В.А.Перепёлкин

Новосибирск 2022

**СОДЕРЖАНИЕ**

Цель 3

Задачи 3

Описание работы 4

Заключение 6

# ЦЕЛЬ

1. Ознакомиться с программированием периферийных устройств на примере ввода данных с Web-камеры с использованием библиотеки OpenCV.

# ЗАДАЧИ

1. Реализовать программу с использованием OpenCV, которая получает поток видеоданных с камеры и выводит его на экран

2. Выполнить произвольное преобразование изображения.

3. Измерить количество кадров, обрабатываемое программой в секунду. Оценить долю времени, затрачиваемого процессором на обработку (ввод, преобразование, показ) видеоданных, получаемых с камеры.

# ОПИСАНИЕ РАБОТЫ

Листинг 1

#include <iostream>

#include <string>

#include <ctime>

#include <opencv2/core.hpp> // Basic OpenCV structures (cv::Mat, Scalar)

#include <opencv2/imgproc.hpp>

#include <opencv2/videoio.hpp>

#include <opencv2/highgui.hpp>

#include <opencv2/imgcodecs.hpp>

#include <opencv2/objdetect.hpp>

using namespace std;

using namespace cv;

int main(int argc, char\* argv[]) {

VideoCapture cap(0);

Mat img;

CascadeClassifier face\_cascade;

face\_cascade.load("haarcascade\_frontalface\_default.xml");

if (face\_cascade.empty()) { cout << "bad xml"; }

long frame\_count = 0;

clock\_t read\_time, read\_time\_end, face\_time, face\_time\_end;

clock\_t draw\_time, draw\_time\_end, output\_time, output\_time\_end;

clock\_t full\_time, full\_time\_end;

double all = 0, camera = 0, face\_proc = 0, draw = 0,output = 0;

full\_time = clock();

clock\_t start = clock();

while (true) {

read\_time = clock();

cap.read(img);

read\_time\_end = clock();

camera += (read\_time\_end - read\_time);

frame\_count++;

clock\_t end = clock();

double seconds = (double)(end - start) / CLOCKS\_PER\_SEC;

vector<Rect> faces;

face\_time = clock();

face\_cascade.detectMultiScale(img, faces, 1.1, 10);

face\_time\_end = clock();

face\_proc += (face\_time\_end - face\_time);

draw\_time = clock();

putText(img, "FPS:" + to\_string(frame\_count / seconds), Point(10, 40), FONT\_HERSHEY\_DUPLEX, 1, Scalar(0, 69, 255), 1);

for (int i = 0; i < faces.size(); i++)

rectangle(img, faces[i].tl(), faces[i].br(), Scalar(255, 0, 0), 2);

string str = "Detected " + to\_string(faces.size()) + " faсes";

putText(img, str, Point(300, 40), FONT\_HERSHEY\_DUPLEX, 1, Scalar(0, 69, 255), 1);

draw\_time\_end = clock();

draw += (draw\_time\_end - draw\_time);

Данная программа получает кадр с веб-камеры, затем с помощью встроенных функций библиотеки OpenCV распознает лица на кадре и выводит текстовой надписью соответствующее количество распознанных лиц. Также с помощью библиотеки <ctime> программа измеряет следующие данные:

output\_time = clock();

imshow("Image", img);

output\_time\_end = clock();

output += (output\_time\_end - output\_time);

char c = waitKey(5);

if (c == 27) break;

}

full\_time\_end = clock();

all = full\_time\_end - full\_time;

cout << "All time: " << all << endl;

cout << "Camera time: " << camera<<" -> " << camera / all \* 100 << "%" << endl;

cout << "Face\_proc time: " << face\_proc << " -> " << face\_proc / all \* 100 << "%" << endl;

cout << "Draw time: " << draw << " -> " << draw / all \* 100 << "%" << endl;

cout << "Output time: " << output << " -> " << output / all \* 100 << "%" << endl;

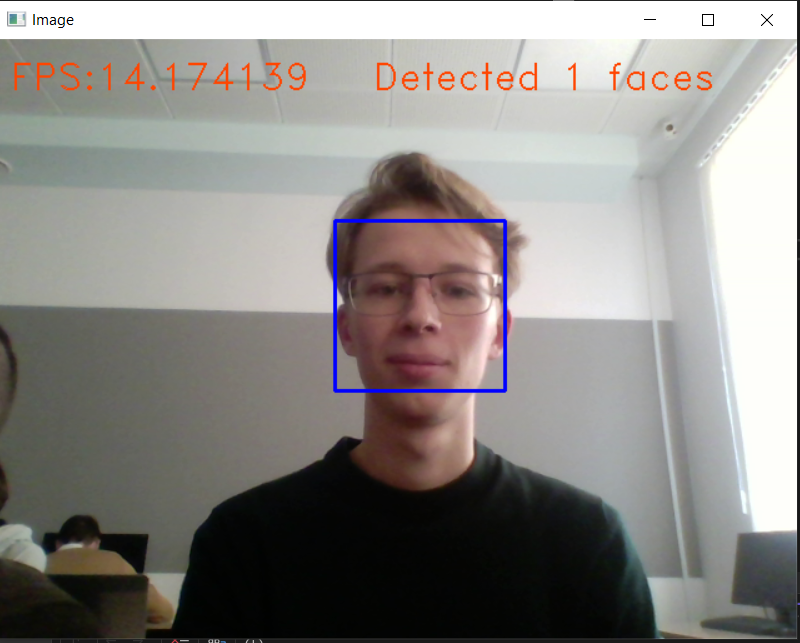
return 0;

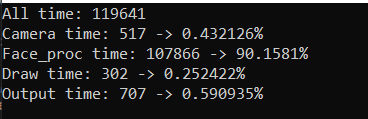
}

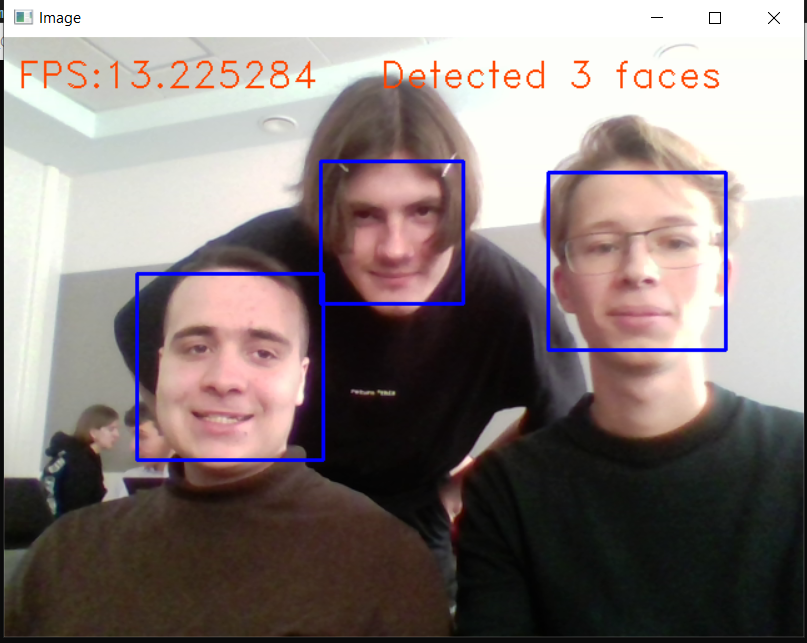
* Количество кадров в секунду (среднее) – количество обработанных кадров поделенное на общее время обработки. Это значение выводится в реальном времени на обработанный кадр.
* Общее время работы части программы (all)
* Время, затраченное на получение кадра с камеры (camera\_time)
* Время, затраченное на поиск лица на кадре (face\_proc)
* Время, затраченное на отрисовку рамки, FPS и количества обнаруженных лиц.
* Время, затраченное на вывод отредактированного кадра на экран.

Также программа выводит процентное соотношение каждого отрезка времени относительно общего времени.

Некоторые результаты:







# ЗАКЛЮЧЕНИЕ

В ходе работы была изучена библиотека OpenCV. С помощью методов этой библиотеки была написана программа для распознавания лиц и вывода на экран количества одновременно распознанных лиц в кадре. Измеренно количество кадров, обрабатываемое программой в секунду и оценены доли времени (в процентах относительного общего времени), затрачиваемого процессором на обработку (ввод, преобразование, показ) видеоданных, получаемых с камеры.