

Разработка системы обнаружения лиц с использованием библиотеки EMGU.CV

Ульянов Егор Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Беликов Андрей Геннадьевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Целью данной статьи является, применение языка программирования C#, среды разработки Visual Studio 2022 и библиотеки «EMGU.CV», в создании программы для обнаружения лиц на изображениях. Практическим результатом является разработанная программа.

Ключевые слова: Обнаружение лиц, библиотека, C#, Visual Studio

Development of a face detection system using the library EMGU.CV

Ulianov Egor Andreevich

Sholom-Aleichem Priamursky State University

Student

Belikov Andrey Gennadievich

Sholom-Aleichem Priamursky State University

Student

Abstract

The purpose of this article is to use the C# programming language, the Visual Studio 2022 development environment and the EMGU library.CV ", in creating a program for detecting faces in images. The practical result is the developed program.

Keywords: Face Detection, Library, C#, Visual Studio

В применении технологии распознавания лиц многие отрасли сейчас видят огромную перспективу. Биометрическая идентификация по лицу – одно из актуальных средств инфобезопасности по всему миру. Распознавание лиц сейчас особенно активно пополняет ряды таких биометрический технологий, как авторизация по голосу или сканирование отпечатков пальцев, которые последние несколько лет применяются в смартфонах и корпоративной безопасности.

Целью данной статьи является создание программы для обнаружения лиц на изображении в среде разработки «Visual Studio» на языке программирования C#.

В своей работе Н. Н. Додобоев, О. И. Кукарцева, Я. А. Тынченко рассмотрели вопросы появления различных языков программирования (в частности C#), определения особенностей этих языков, а также составления основных видов и классификаций языков программирования [1]. З. С. Магомадова рассмотрела языки программирования высокого уровня, особенности, недостатки и сложности в изучении, а также описала несколько легких алгоритмов [2]. В своей работе В.Ж. Жамалова, Т.Т. Каримбаев, Ф.Р. Раймжанова, Э.С. Сатаров рассмотрели применение технологии WPF для создания тестирующей программы с мультимедийными компонентами занятий студентов физкультурников [3]. В статье А.Н. Загребина, И.В. Николаевой рассмотрена тема обнаружения лиц с помощью алгоритмов компьютерного зрения на языке программирования Python, а также основные библиотеки, служащие для реализации алгоритма [4].

Создаем проект «Windows Forms App» и называем его «FaceDetection» см. рисунок 1.

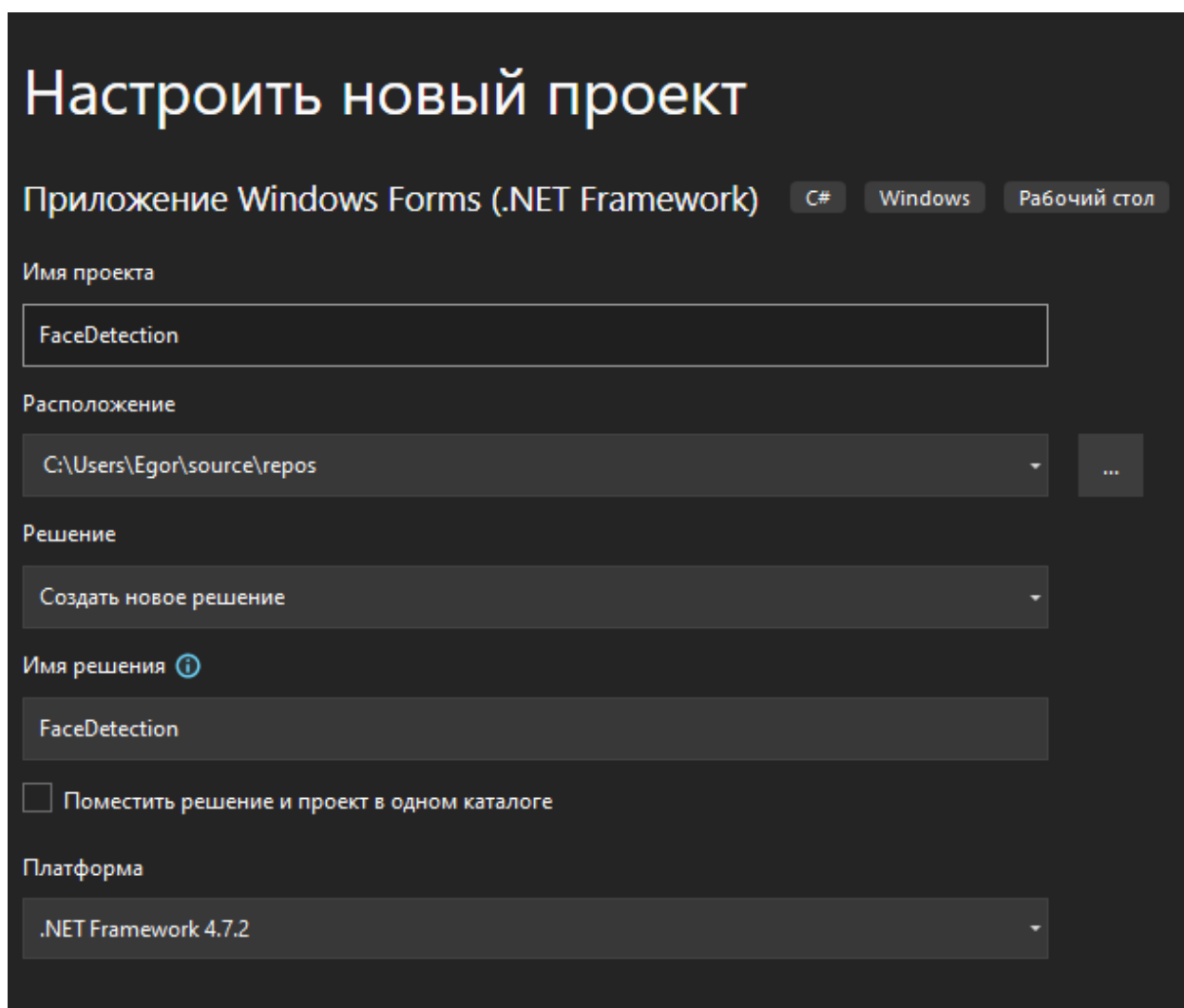


Рис. 1. Создание проекта

Далее в свойствах формы меняем название окна и параметр «StartPosition» как на рисунке 2.

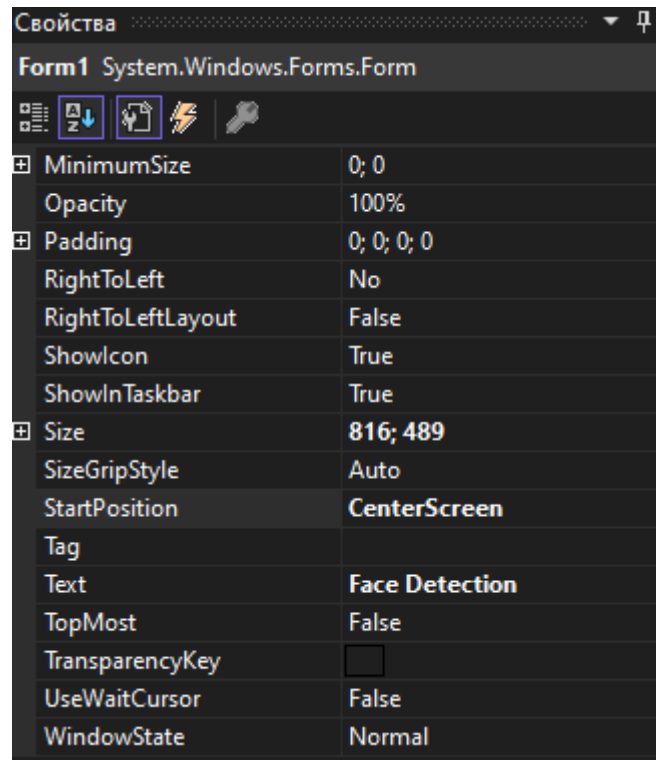


Рис. 2. Свойства формы

Добавляем элементы на форму: «Button» и «PictureBox» и располагаем как на рисунке 3.

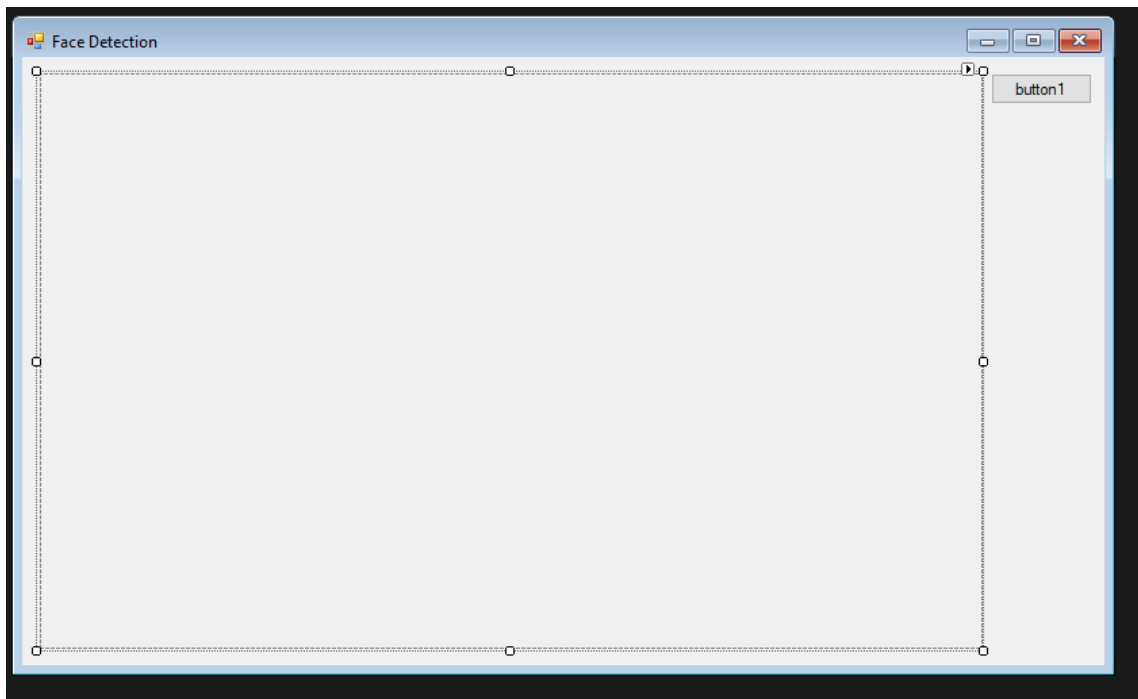


Рис. 3. Расположение элементов

В свойствах кнопки, параметр «Text» меняем на «&Detect» (Знак & подчеркнет первый символ), а в параметре «Name» для простоты обращения к элементу в коде изменим на «btnDetect» см. рисунок 4-5.

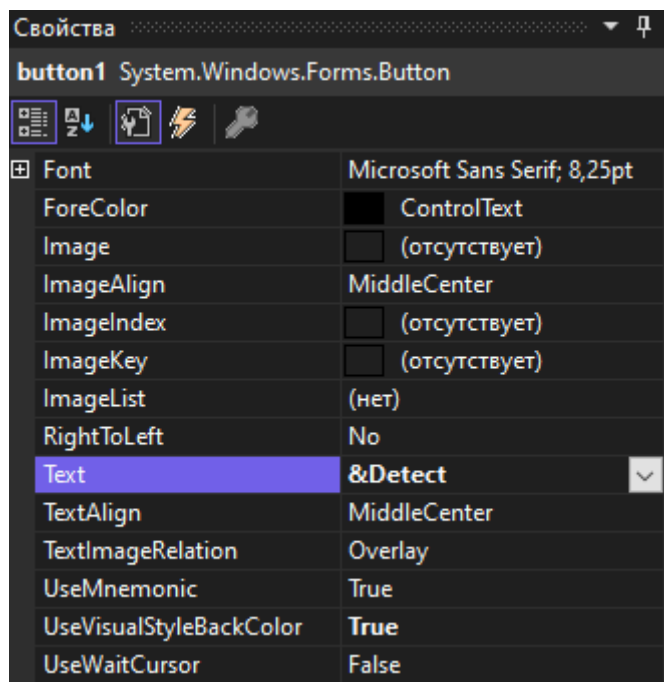


Рис. 4. Свойства элемента «Button»

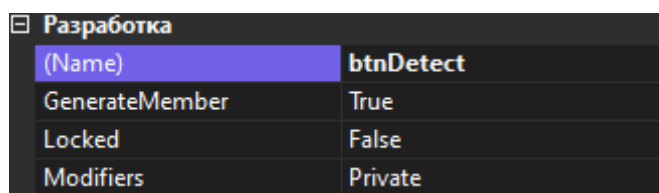


Рис. 5. Свойства элемента «Button»

Далее в свойствах «PictureBox» переименуем элемент и поменяем значение параметра стиля рамки на «Fixed3D» см. рисунок 6.

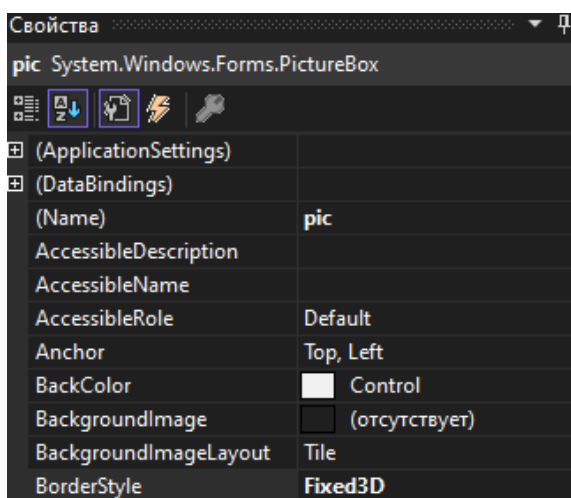


Рис. 6. Свойства элемента «PictureBox»

Теперь необходимо установить библиотеку «EMGU.CV» из «Manage Nuget Packages» в проект. Эта библиотека поможет создавать приложения, поддерживающие обнаружение и распознавание см. рисунок 7-8.

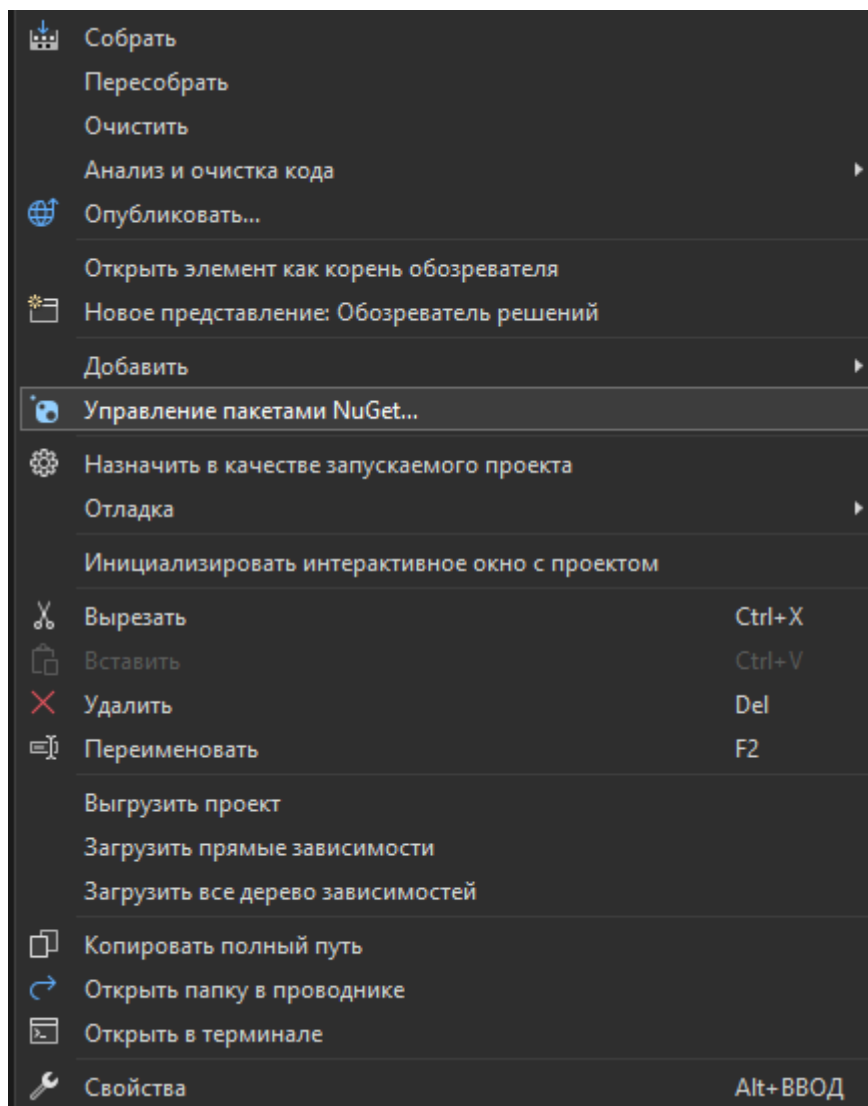


Рис. 7. Запуск «Manage Nuget Packages»

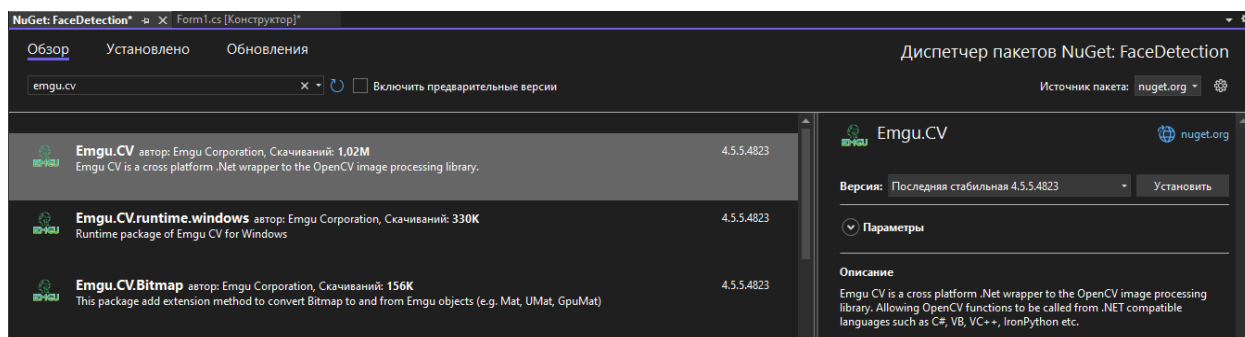


Рис. 8. Установка библиотеки «EMGU.CV»

Далее необходимо скачать файл «haarcascade_frontalface_alt_tree.xml» с репозитория (github) библиотеки, затем скопировать в проект и установить в свойствах всегда копировать см. рисунок 9-10.

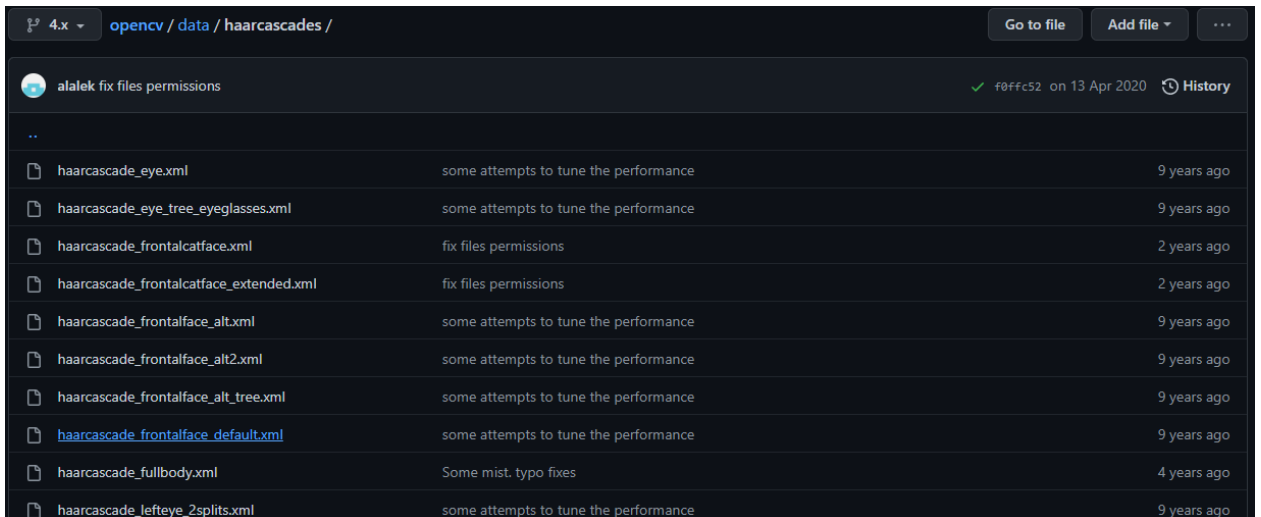


Рис. 9. Репозиторий библиотеки

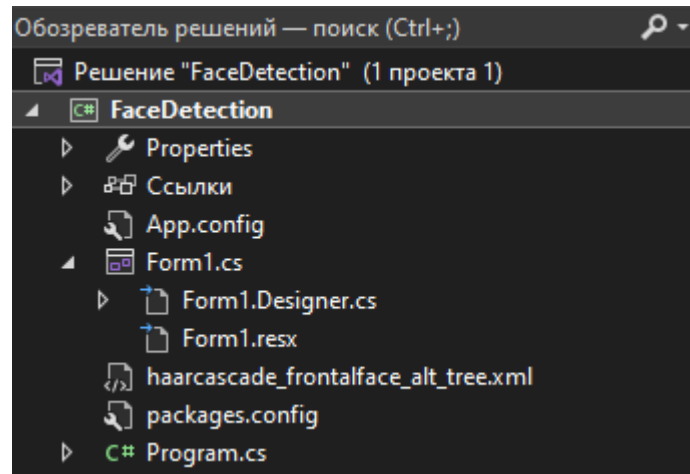


Рис. 10. Файл в проекте

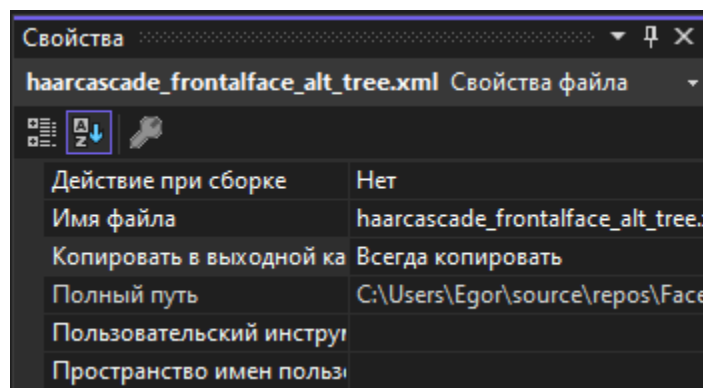


Рис. 11. Свойства файла

Переходим к написанию кода, для начала необходимо добавить события на кнопке, для этого кликаем два раза по элементу и переходим в редактор кода. Сразу подключаем необходимые библиотеки см. рисунок 12.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Emgu.CV;
using Emgu.CV.Structure;
```

Рис. 12. Подключение необходимых библиотек

Создаем класс «CascadeClassifier», который помогает обнаруживать объекты на изображении см. рисунок 13.

```
static readonly CascadeClassifier cascadeClassifier = new CascadeClassifier("haarcascade_frontalface_alt_tree.xml");
```

Рис. 13. Создание класса «CascadeClassifier»

В методе нажатия кнопки, необходимо добавить обработчик события по щелчку, а именно открытие диалогового окна с выбором файла с расширением «.jpg». В этому же методе прописываем использование созданного ранее класса см. рисунок 14.

```
Ссылка: 1
private void btnDetect_Click(object sender, EventArgs e)
{
    using(OpenFileDialog ofd = new OpenFileDialog() { Multiselect = false, Filter = "JPEG[*].jpg" })
    {
        if(ofd.ShowDialog() == DialogResult.OK)
        {
            pic.Image=Image.FromFile(ofd.FileName);
            Bitmap bitmap = new Bitmap(pic.Image);
            Image<Bgr, byte> grayImage = new Image<Bgr, byte>(bitmap);
            Rectangle[] rectangles=cascadeClassifier.DetectMultiScale(grayImage, 1.4, 0);
            foreach(Rectangle rectangle in rectangles)
            {
                using(Graphics graphics = Graphics.FromImage(bitmap))
                {
                    using(Pen pen=new Pen(Color.Red, 1))
                    {
                        graphics.DrawRectangle(pen, rectangle);
                    }
                }
            }
            pic.Image = bitmap;
        }
    }
}
```

Рис. 14. Обработчик события по щелчку

Теперь необходимо протестировать программу, запускаем и выбираем изображение см. рисунок 15-17.

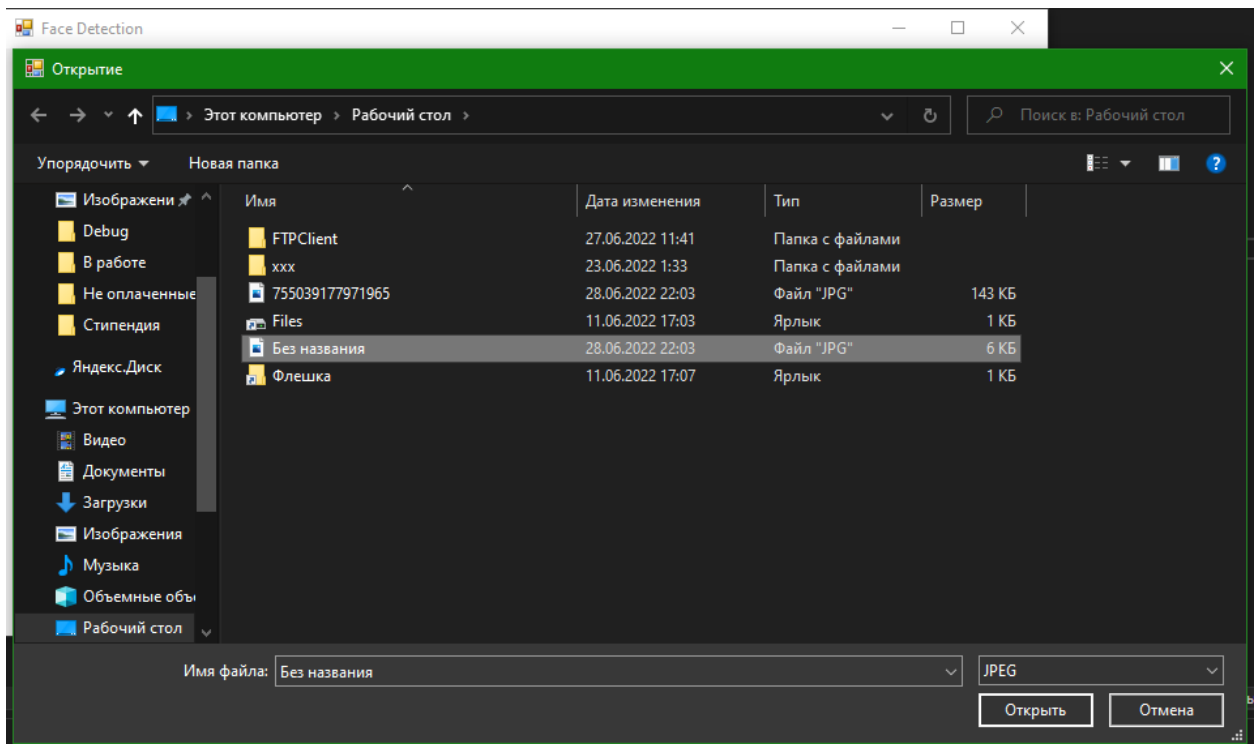


Рис. 15. Выбор файла

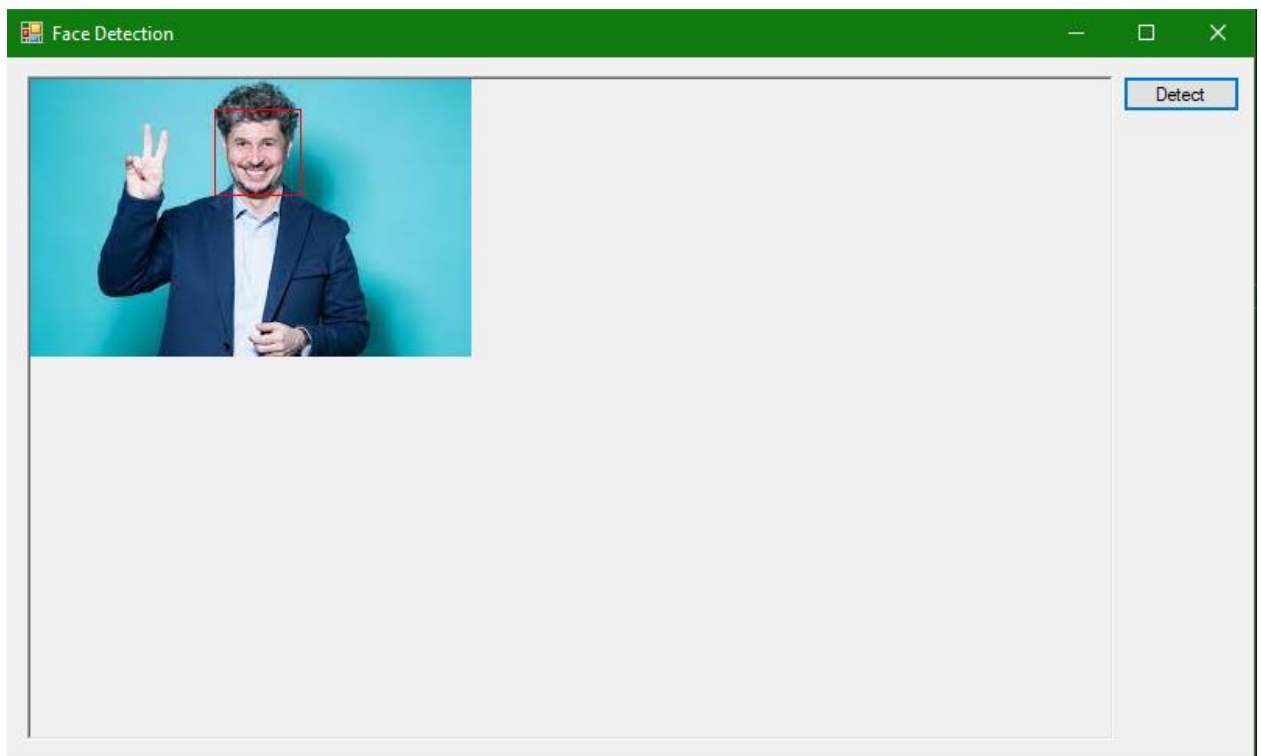


Рис. 16. Работа обнаружения

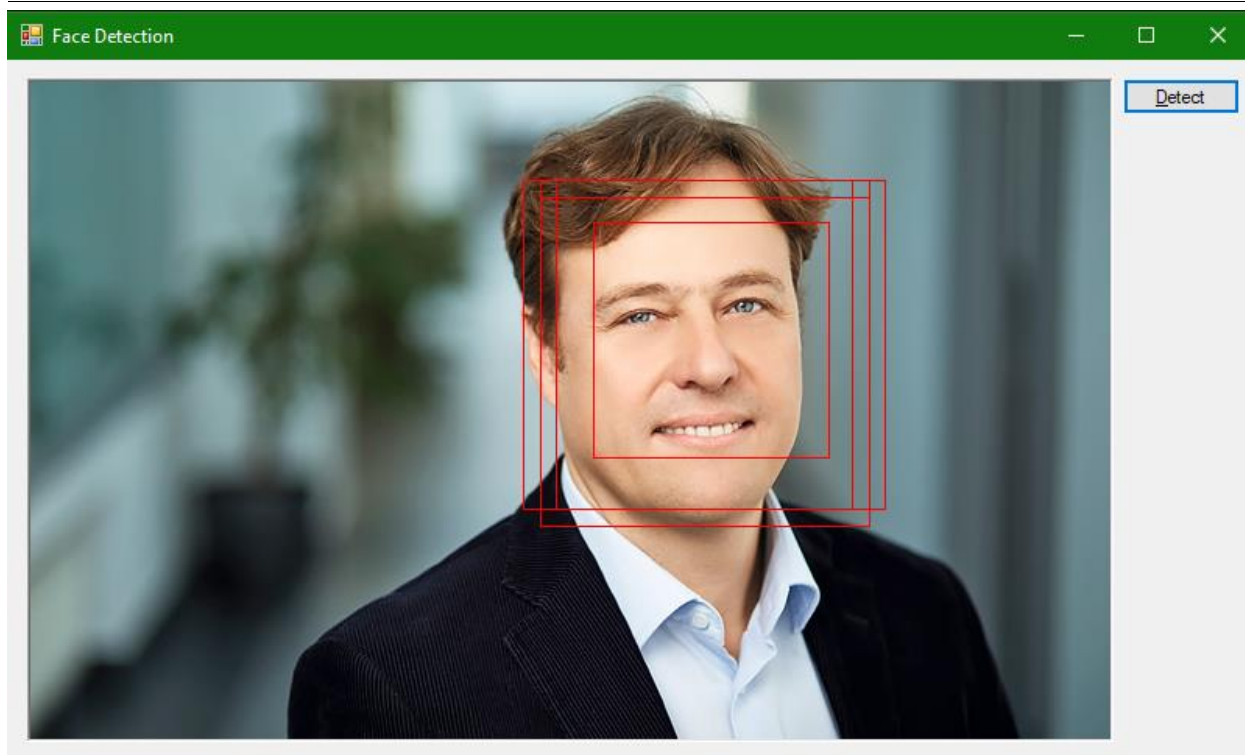


Рис. 17. Работа обнаружения

Таким образом, была написана программа для обнаружения лиц на изображениях в среде разработки «Visual Studio» на языке программирования C#.

Библиографический список

1. Додобоев Н. Н., Кукарцева О. И., Тынченко Я. А. Современные языки программирования // Современные технологии: актуальные вопросы, достижения и инновации. 2014. №5. С. 81-85.
2. Магомадова З. С. Языки программирования высокого уровня // Разработка и применение наукоёмких технологий в эпоху глобальных трансформаций. 2020. №8. С. 94-96.
3. Жамалова В.Ж., Каримбаев Т.Т., Раймжанова Ф.Р., Сатаров Э.С. Программа тестирования с мультимедийными компонентами на основе WPF //Наука и инновационные технологии. 2020. С. 55-60.
4. Загребина А.Н., Николаевой И.В. Алгоритмы компьютерного зрения: обнаружение лиц //Информационное общество: современное состояние и перспективы развития. 2019. №17. С. 151-153.