**МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ**

**Государственное бюджетное профессиональное образовательное учреждение Московской области «Люберецкий техникум имени Героя Советского Союза, летчика-космонавта Ю.А.Гагарина»**

**ОТЧЕТ**

**УЧЕБНОЙ ПРАКТИКИ**

Савицкого Даниила Валентиновича

(Фамилия, имя, отчество студента)

по профессиональному модулю

**\_\_\_\_\_\_\_\_***ПМ.01 «Разработка модулей программного обеспечения***\_\_\_\_\_\_\_\_**

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ для компьютерных систем»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Специальность \_*09.02.07 "Информационные системы и программирование"*\_

Код, название

Курс \_3\_ Группа № \_195\_

Период практики с «12» ноября 2021 г. по «24» ноября 2021г.

Руководитель практики

от техникума Ревнивцев Максим Валерьевич

Жирнова Юлия Витальевна

Люберцы 2021

**Оглавление**

[Введение 4](#_Toc89016056)

[1. Изучение предметной области 6](#_Toc89016057)

[2. Изучение программных средств 7](#_Toc89016058)

[2.1 OpenCV 8](#_Toc89016059)

[2.2 Cosmopy 11](#_Toc89016060)

[2.3 WPF 13](#_Toc89016061)

[2.4 Нейросети 25](#_Toc89016062)

[Выводы 27](#_Toc89016063)

[Список литературы 28](#_Toc89016064)

Студента 3 курса

Савицкий Даниил Валентинович

Специальность: ИС и программирование

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Дата** | **Краткое содержание выполнения работ** | **Отметка руководителя практики**  **(оценка, подпись)** |
| 1 | 2 |  | 3 |
|  |  |  |  |

Руководитель практики

от техникума \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

подпись Фио

# **Введение**

Целью учебной практики является: закрепление теоретических знаний, полученных при изучении базовых дисциплин; изучение организационной структуры предприятия и действующей на нем системы управления; ознакомление с содержанием основных работ и исследований, выполняемых на предприятии или в организации по месту прохождения практики.

Целью учебной практики является реализация компьютерного зрения с помощью библиотеки OpenCV, разработка приложения с помощью WPF и навыки изучения массивов данных с помощью python и miniconda(cosmopy)

Для достижения данной цели требуется решить следующие задачи:

1. Знакомство с литературой
2. Изучить библиотеку OpenCV
3. Реализация приложения на WPF

В ходе учебной практики планируется освоение следующего вида деятельности: «Разработка модулей программного обеспечения для компьютерных систем» (ПК):

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.

ПК 2.1 Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

# **1. Изучение предметной области**

1. **OpenCV** — это open source библиотека компьютерного зрения, которая предназначена для анализа, классификации и обработки изображений. Широко используется в таких языках как C, C++, Python и Java.

* Для утверждения общего стандартного интерфейса компьютерного зрения для приложений в этой области. Для способствования росту числа таких приложений и создания новых моделей использования PC.
* Сделать платформы Intel привлекательными для разработчиков таких приложений за счёт дополнительного ускорения OpenCV с помощью Intel® Performance Libraries (Сейчас включают IPP (низкоуровневые библиотеки для обработки сигналов, изображений, а также медиа-кодеки) и MKL (специальная версия LAPACK и FFTPack)). OpenCV способна автоматически обнаруживать присутствие IPP и MKL и использовать их для ускорения обработки.

2. **Cosmopy** Курс, который помог мне научиться анализировать огромные объемы информации, обрабатывать их, находить и манипулировать необходимыми данными

Для прохождения курса было необходимо использование минимального дистрибутива для Python – Miniconda

**Anaconda** – это полный дистрибутив программного обеспечения вэкосистеме[PyData](https://pydata.org/) , включая сам Python, а также двоичные файлы для нескольких сторонних проектов с открытым исходным кодом. Помимо Anaconda, есть также[Miniconda](https://conda.io/miniconda.html) , минимальный дистрибутив Python, включающий в основном Conda и его зависимости, так что вы можете устанавливать с нуля только те пакеты, которые вам нужны.

[**Conda**](https://conda.io/) – это система управления пакетами, зависимостями и средой, которую можно установить без распространения Anaconda или Miniconda. Он работает в Windows, macOS и Linux и был создан для программ на Python, но может упаковывать и распространять программное обеспечение для любого языка. Основная цель – решить проблемы внешних зависимостей простым способом, загрузив предварительнотого, менеджером среды, поэтому, если вам нужен пакет, который требует другой версии Python, с помощью Conda можно настроить отдельную среду с совершенно другой версией Python, поддерживая вашу обычную версию Python на ваша среда по умолчанию.

3. **Разработка приложения на WPF** (*Windows Presentation Foundation (WPF) — аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML*) помогла мне научиться разрабатывать соединять простейшее приложение с базой данных SQL management Studio (SSMS) — *утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера*

# **2. Изучение программных средств**

1. Для подключения OpenCV на python достаточно:

* Открываем PyCharm и нажимаем New project, выбираем папку, где хотим, чтобы был наш проект, и окружение Virtualenv, слева появятся папки
* File - Settings - Project Interpretater - шестеренка(add) - new enviroment. После этого Apply. Слева появится папка venv. Теперь все установленные библиотеки будут в виртуальном окружении и не будут засорять память
* Там же нажимаем на +, вводим opencv-python, install. Теперь появятся библиотеки нужные (https://youtu.be/iMM4TImNJFM)
* Заходим в Terminal Local python Имя\_файла.py

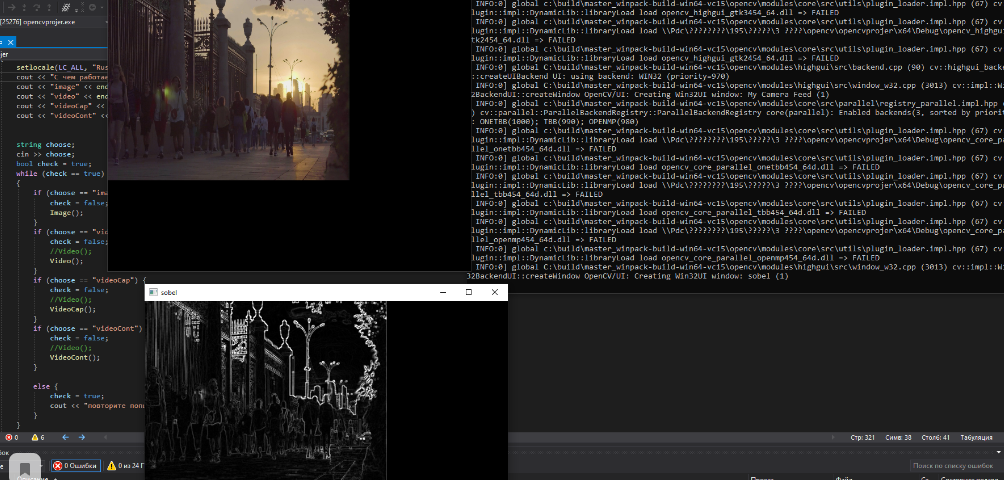
1. Недостатком C# является само подключение, для которого нужны изменения в фалах проекта и добавление дополнительных библиотек:
   * Скачать Open CV с сайта https://opencv.org/releases/ и выбрать версию 4.5.3 для Windows.
   * Установить Open CV (желательно сохранить в диске С или D D:\opencv)
   * Создаем пустой проект C++
   * Кликаем по названию проекта и кликаем на "Добавить" --> “Создать элемент” --> “Файл C++(.cpp)”
   * После этого по названию проекта кликаем правой кнопкой мыши и выбираем "Свойства"
   * Выбираем конфигурацию All Configurations, платформу х64 (в двух местах))
   * Находим C/C++ --> Раскрываем и находим "Общие" --> Находим каталоги включаемых файлов --> указываем путь openCv "...\opencv\build\include"
   * Далее выбираем "Компоновщик" --> "Общие" --> "Дополнительные каталоги библиотек" --> указываем путь "...\opencv\build\x64\vc15\lib"
   * Так же "Компоновщик" находим "Ввод" --> "Дополнительные зависимости и из папки lib выбираем opencv\_world453d.lib
   * Из папки "...opencv\build\x64\vc15\bin" копируете opencv\_world453d.dll в папку ...\Название вашего проекта\x64\Debug
2. WPF это библиотека для создания десктопных приложений для работы с ним мы включаем Visual Studio и создаём проект с WPF.
3. 3Ds Max это среда для создания 3D моделей и образов.

## **2.1 OpenCV**

1. Теория

Оператор Собеля(рис.1) — дискретный дифференциальный оператор, вычисляющий приближённое значение градиента яркости изображения. Результатом применения оператора Собеля в каждой точке изображения является либо вектор градиента яркости в этой точке, либо его норма. Используется в области обработки изображений, в частности, часто применяется в алгоритмах выделения границ.

VideoCapture cap – оператор для использования веб камеры

detectMultiScale — общая функция для распознавания как лиц, так и объектов. Чтобы функция искала именно лица, мы передаём ей соответствующий каскад.  


Оператор Собеля

Функция detectMultiScale принимает 4 параметра:

Обрабатываемое изображение в градации серого.

* Параметр scaleFactor. Некоторые лица могут быть больше других, поскольку находятся ближе, чем остальные. Этот параметр компенсирует перспективу.
* Алгоритм распознавания использует скользящее окно во время распознавания объектов. Параметр minNeighbors определяет количество объектов вокруг лица. То есть чем больше значение этого параметра, тем больше аналогичных объектов необходимо алгоритму, чтобы он определил текущий объект, как лицо. Слишком маленькое значение увеличит количество ложных срабатываний, а слишком большое сделает алгоритм более требовательным.
* minSize — непосредственно размер этих областей.

Сохранение изображения

import cv2

image = cv2.imread("./импорт/путь.расширение")

cv2.imwrite("./экспорт/путь.расширение", image)

Рисование прямоугольников

import cv2

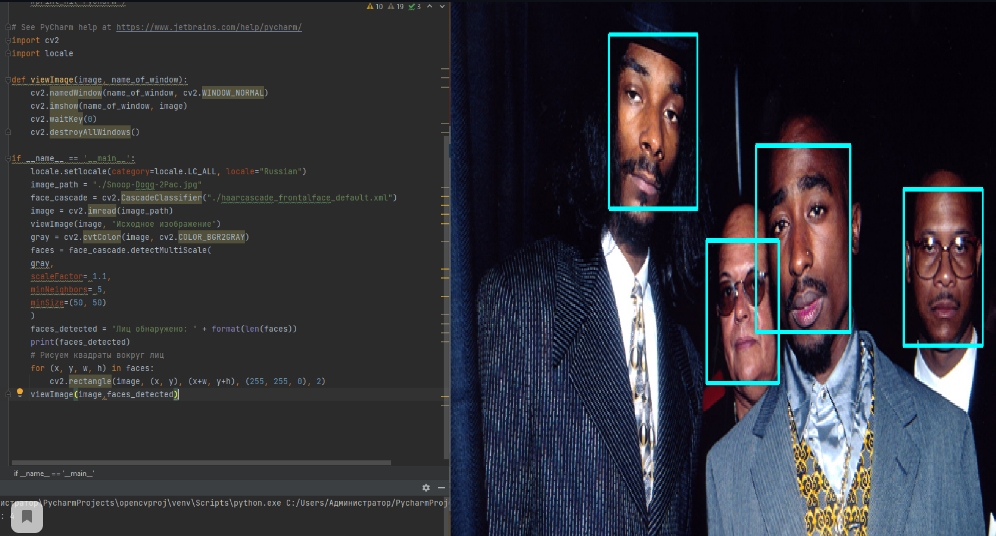
output = image.copy()

cv2.rectangle(output, (2600, 800), (4100, 2400), (0, 255, 255), 10)

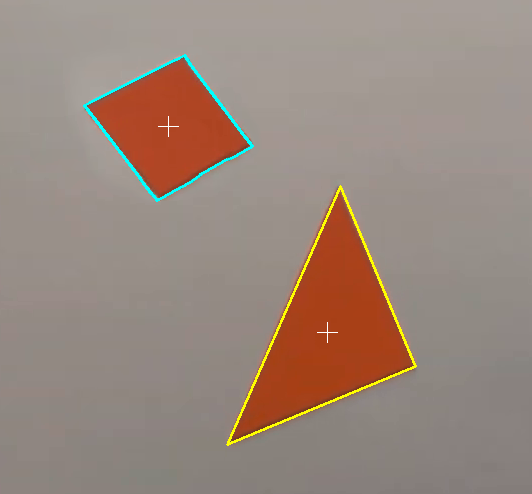
viewImage(output, "Обводим прямоугольником лицо пёсика")

Эта функция принимает 5 параметров:

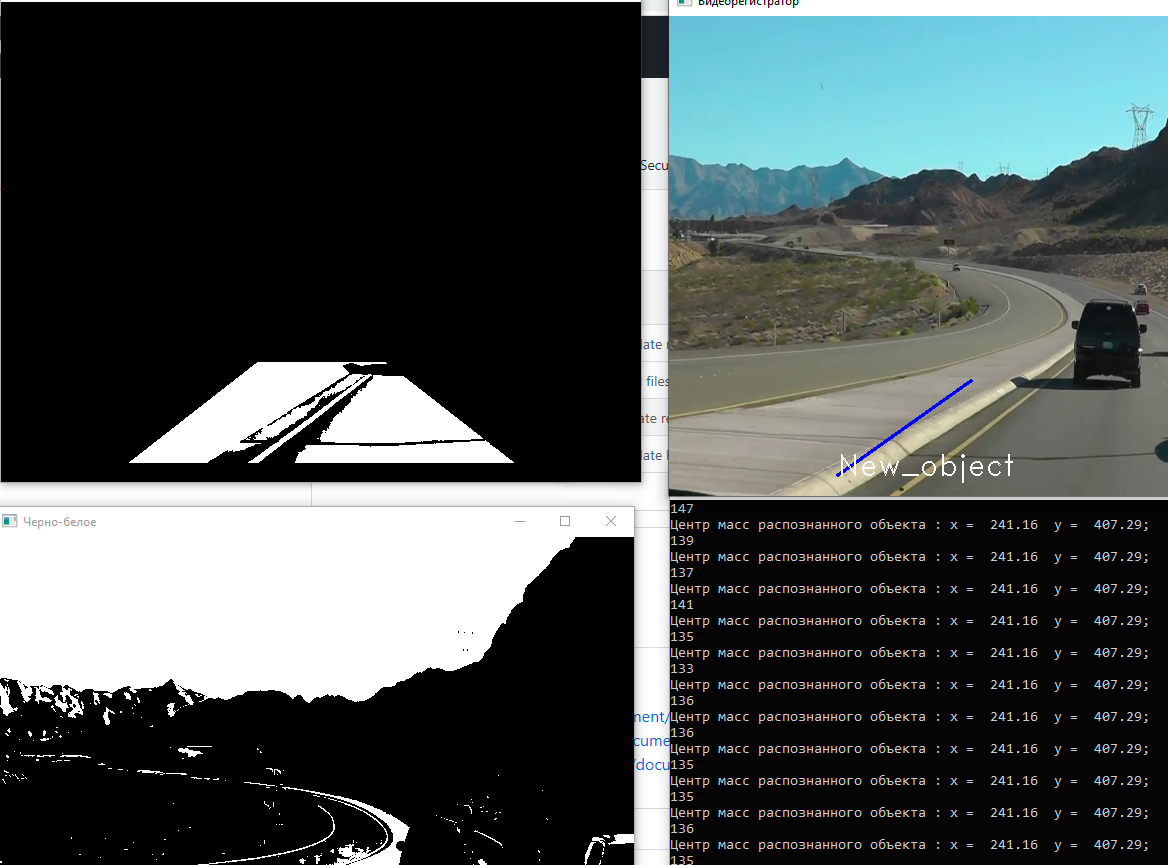
* Само изображение.
* Координата верхнего левого угла (x1, y1).
* Координата нижнего правого угла (x2, y2).
* Цвет прямоугольника (GBR/RGB в зависимости от выбранной цветовой модели).
* Толщина линии прямоугольника.



Распознавание лиц с помощью библиотеки OpenCV



Контуры и центры масс



Распознавание дорожной разметки

## **2.2 Cosmopy**

В данном курсе использовались следующие программы и компоненты:

* Visual Studio Code
* Python
* Miniconda (упрощённый вид дистрибутива Anaconda)

**Anaconda** – это полный дистрибутив программного обеспечения вэкосистеме[PyData](https://pydata.org/) , включая сам Python, а также двоичные файлы для нескольких сторонних проектов с открытым исходным кодом. Помимо Anaconda, есть также[Miniconda](https://conda.io/miniconda.html) , минимальный дистрибутив Python, включающий в основном Conda и его зависимости, так что вы можете устанавливать с нуля только те пакеты, которые вам нужны.

**pandas** (библиотека Python, которую мы используем для анализа данных) имеет структуру, именуемую кадром, и действительно эффективна для представления двумерных данных. Возможно, вы обратили внимание, что при выполнении кода  выводимый результат очень похож на лист Excel. Кадры данных в Pandas действительно удобно представлять как такие листы.

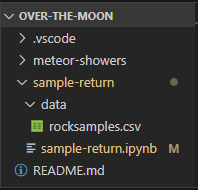
В кадре данных  имеется по одной строке для каждого собранного образца. Однако, как мы уже упоминали, мы хотим проанализировать образцы в совокупности в их связи с конкретными ракетами, на которых они были возвращены.

В рамках курса нам были предоставлены данные, которые вы будете изучать в этом модуле, представляют собой файл со всеми образцами, собранными в шести миссиях "Аполлон" с высадкой на Луну. Файл [rocksamples.csv](https://aka.ms/LearnWithDrG/OverTheMoon/Data2) был создан на основе [каталога лунных образцов и фотографий](https://curator.jsc.nasa.gov/lunar/samplecatalog/index.cfm).

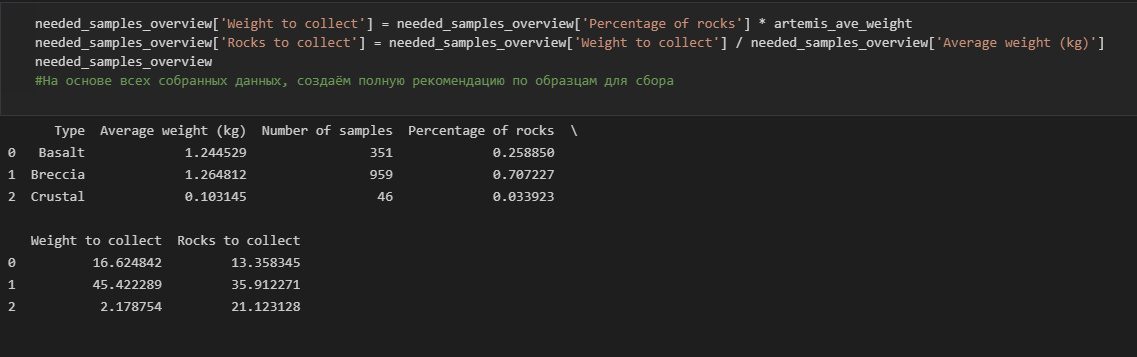
После установки трех перечисленных выше компонентов выполните следующие действия, чтобы подготовить среду.

* Создайте папку с именем *over-the-moon*.
* Откройте папку в Visual Studio Code.
* Создайте папку внутри этой папки с именем *sample-return*.
* Создайте папку внутри этой папки с именем *data*.
* Создайте файл с именем *sample-return.ipynb* в папке *sample-return*.
* Откройте файл *sample-return.ipynb* в Visual Studio Code.

После загрузки CSV-файла и сохранения его в папке данных среда Visual Studio Code должна выглядеть следующим образом:



Далее мы проводим полное изучение и анализ данного файла и в конечном счёте создаём кадр с прогнозом по сбору необходимых лунных пород

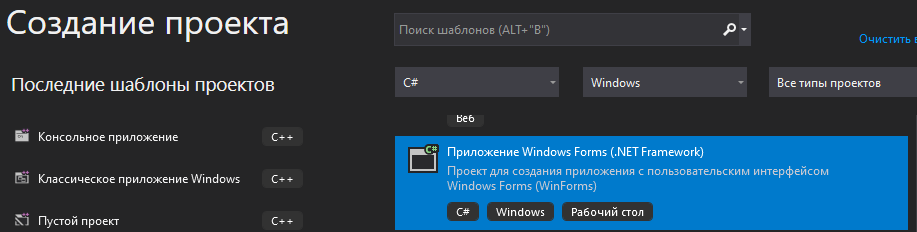


Конечный результат

Полный код имеет очень удобную структуру, каждое действие разбито по блокам и прокомментировано, его можно посмотреть по этой ссылке: <https://clck.ru/ZC3Wb>

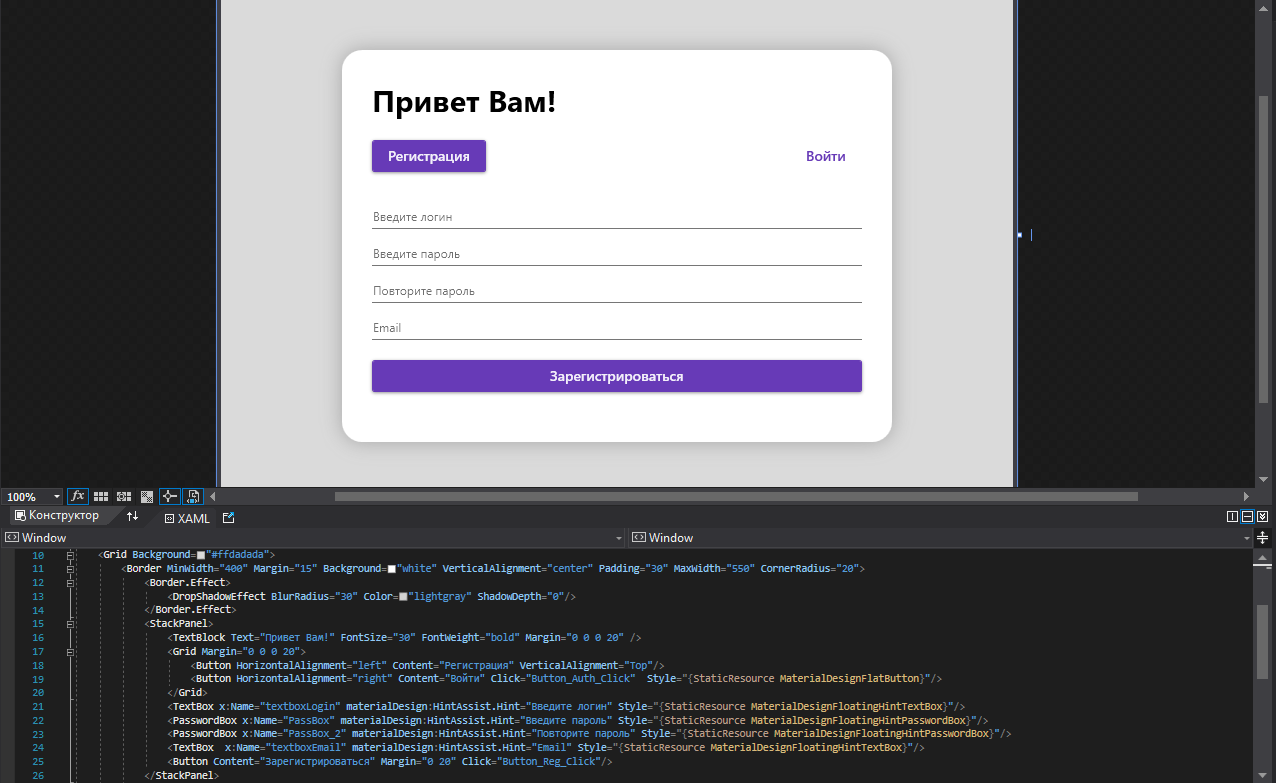
## **2.3 WPF**

* Для прохождения практики установлен **SQL Server Management Studio** и **IDE Visual Studio 2019 Community Edition**
* Создан проект WPF .NET Framework



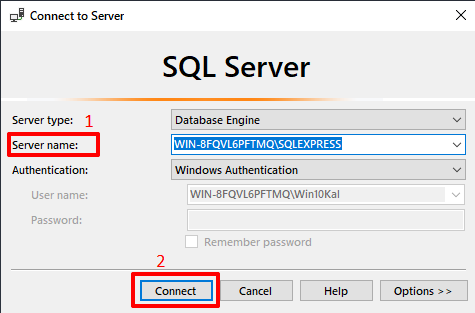
Создание проекта WPF

* Написано приложение с окном Регистрации

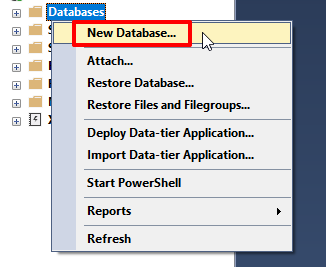


Окно с регистрацией

* Создание базы данных на локальном сервере

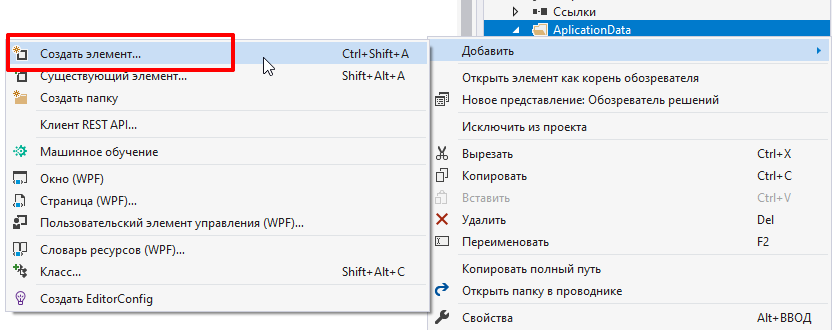


Подключение к серверу

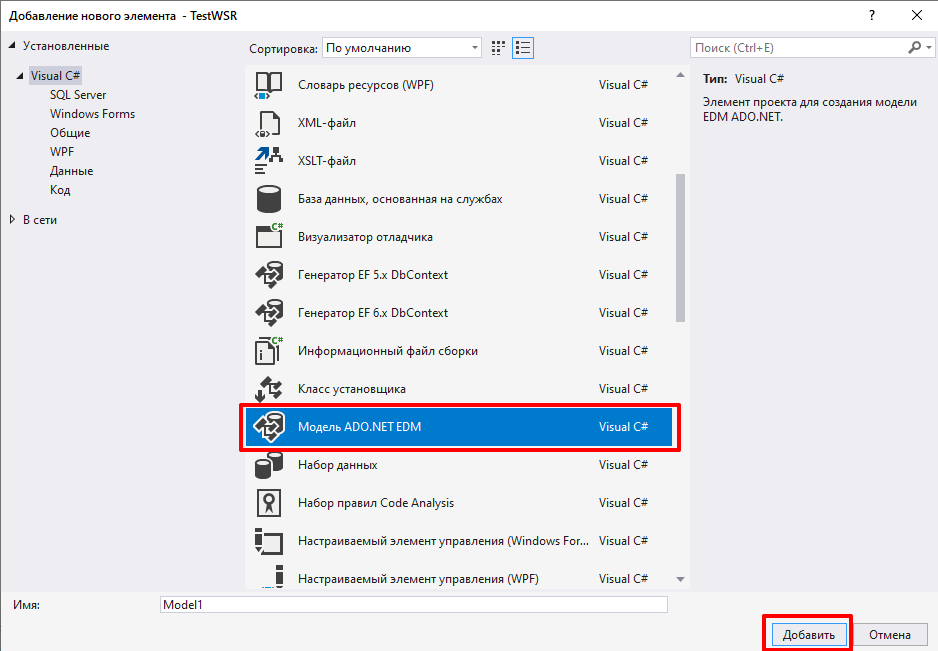


Создание базы данных

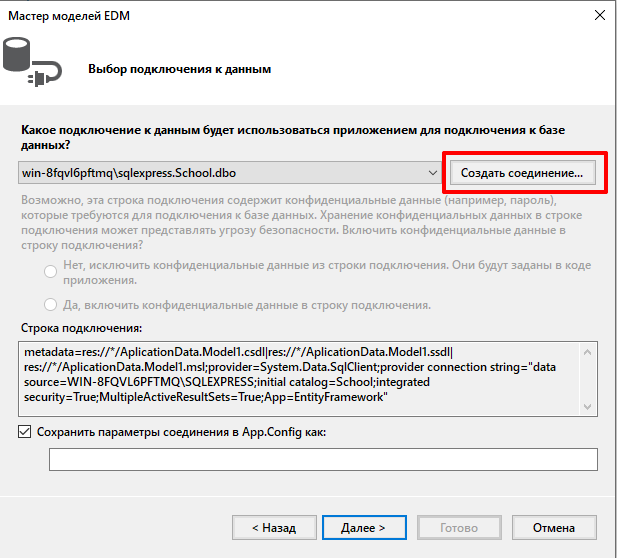
* Создание таблицы Users
* База данных была подключена к приложению с помощью добавления класса **модели данных ADO.NET Entity**



Подключение бд к WPF приложению



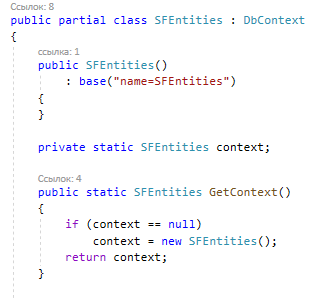
Выбор модели



Создание подключения

Выбраны таблицы и добавлены в проект с названием пространства имен модели SFEntities.

* Написан код в файле Model1.Context.cs, записывающий результаты в таблицу базы данных.

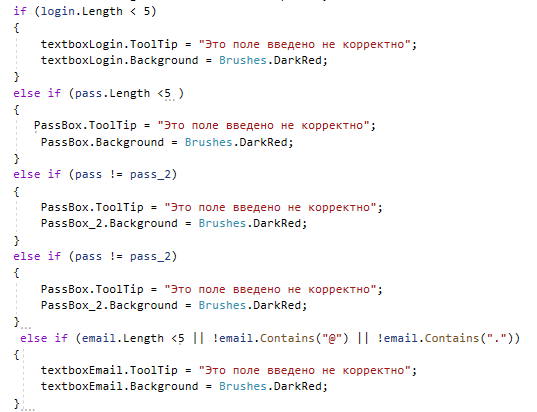


Код GetContext()



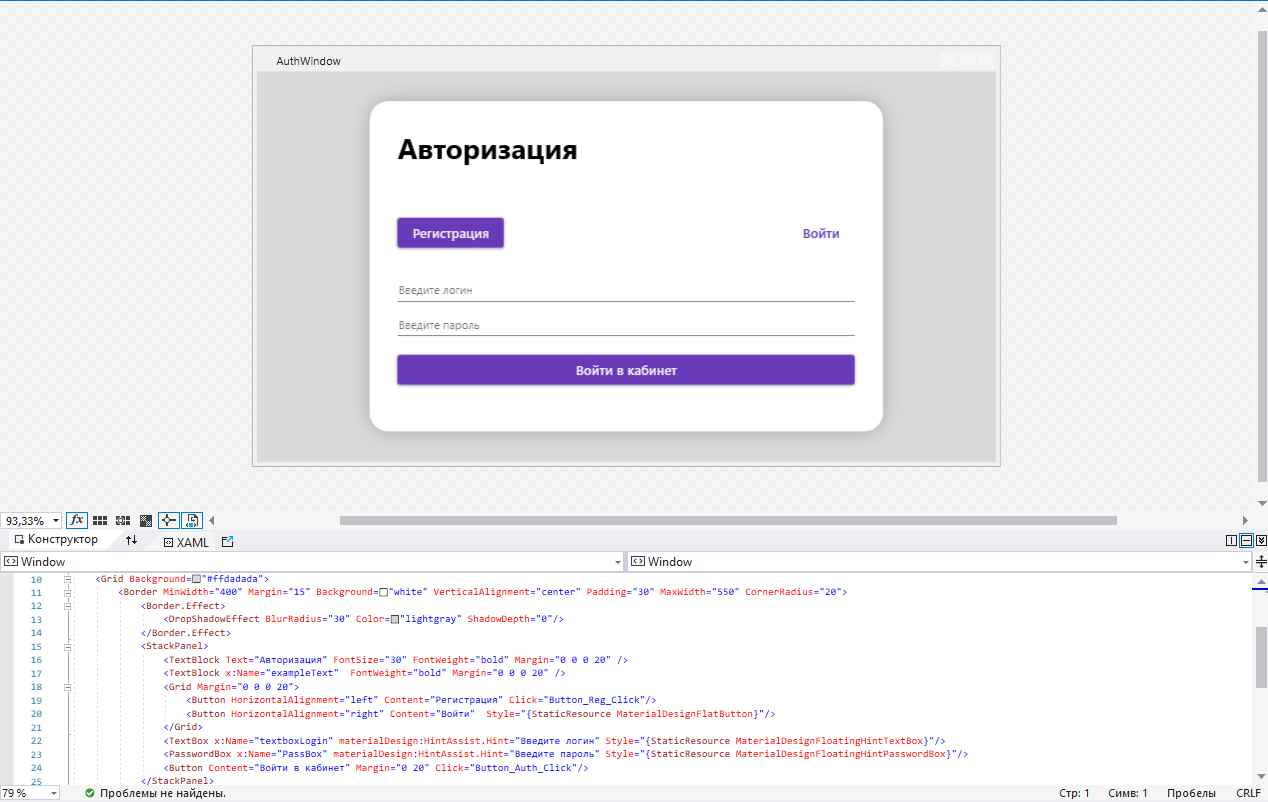
Код для сохранения в таблицу

* Прописаны исключения



Создание условий

* Создано окно с авторизацией



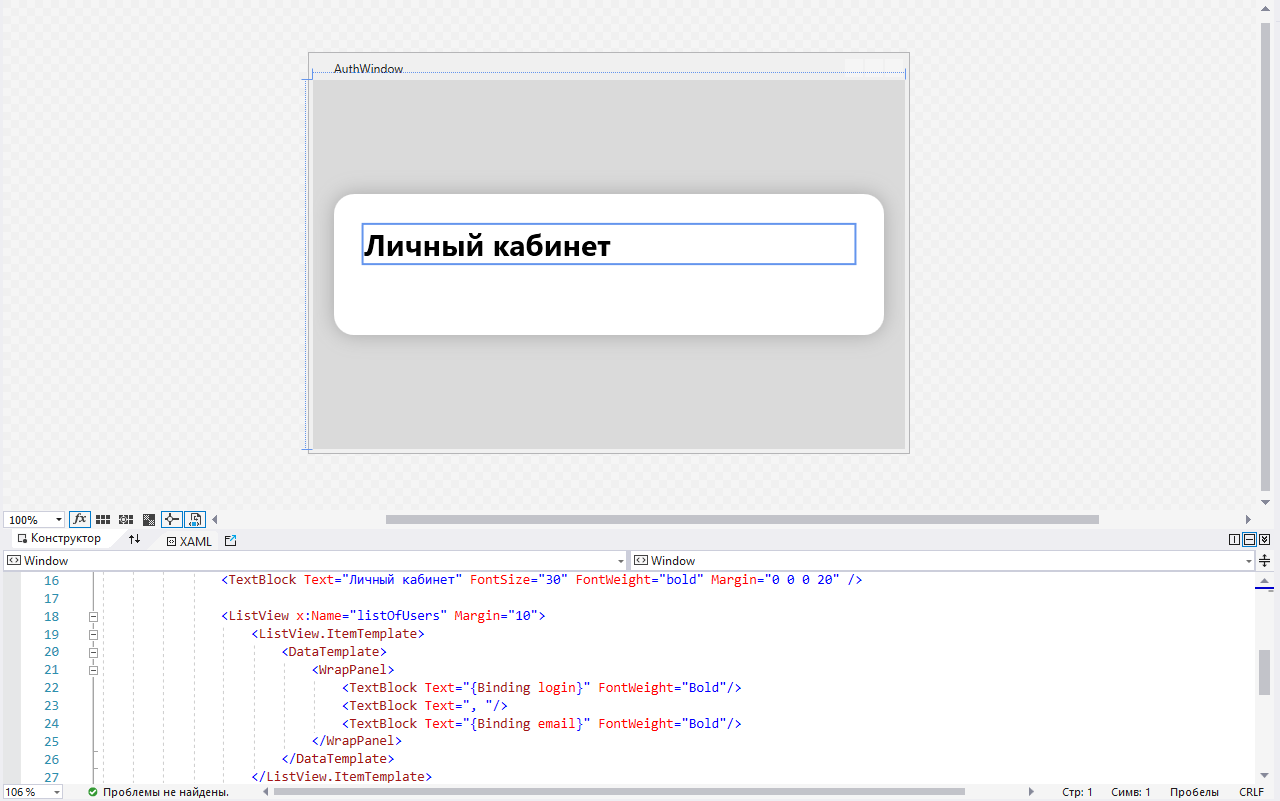
Окно авторизации пользователя

* Написан обработчик событий для проверки данных при авторизации.



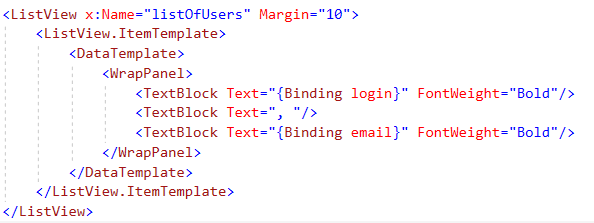
Обработчик событий на кнопку авторизации

* Создано окно с Личным Кабинетом, где написаны все логины и пароли из базы данных

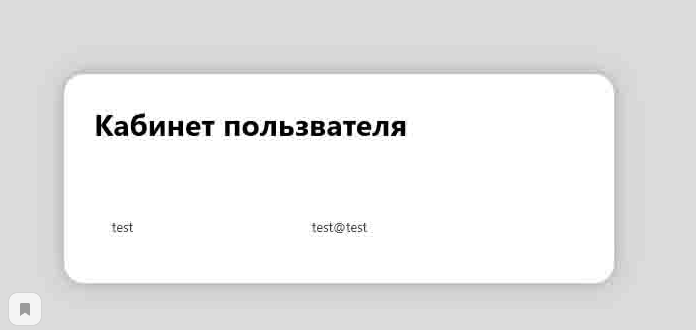


Окно «Личный кабинет»

* Добавлен шаблон для TextBox, чтобы результаты выводились в созданное окно



Код шаблона



Окно с данными из бд

## **2.4 Нейросети**

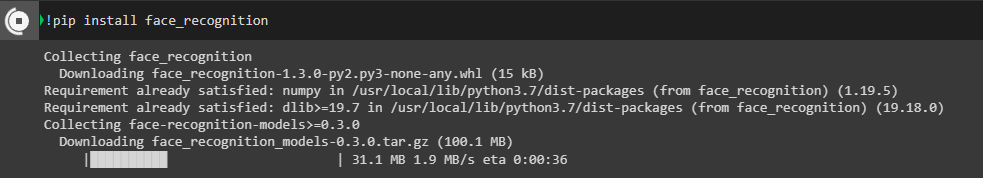
Для написания кода была использована среда Google Collab

Код был написан на основе данного урока: <https://clck.ru/ZC3uY>

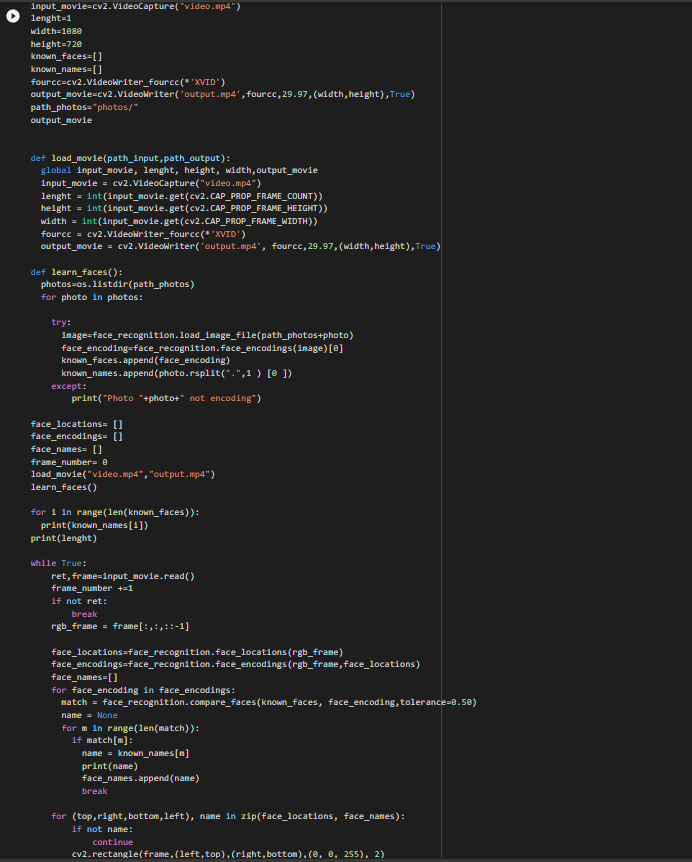
Перед началом работы необходимо подготовить все файлы:

1. Откройте панель слева и нажмите на иконку папки
2. Необходимо создать папку в которой будут храниться фото для распознания
3. Загрузите в данную папку все необходимые фотографии
4. Загрузите исходное видео

Также перед началом работы необходимо установить библиотеку face\_recognition



Код и процесс установки библиотеки



Код программы

В качестве обрабатываемого видео был взят трейлер к фильму “Джентльмены”.

В полученном обработанном видео были обнаруженны следующие персонажи:

* Главный герой Микки Пирсон
* Его близкий друг Рэймонд Смит
* Конкурент гг Генри Голдинг
* Частный детектив Флетчер

Далее представлены результаты обработки:





# **Выводы**

По окончании учебной практики были решены следующие задачи:

* Изучена предметную область.
* Освоен курс “Изучение космоса с помощь Python и Miniconda”.
* Освоено проектирование и создание баз данных.
* Изучено создание запросов для взаимодействия с БД.
* Разработаны приложение на платформе WPF.
* Знакомство с литературой.
* Изучена библиотека OpenCV.

В ходе прохождения учебной практики были сформированы следующие профессиональные компетенции:

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

# **Список литературы**

1. [Орлов С.А.](http://library.sgu.ru/cgi-bin/irbis64r_15/cgiirbis_64.exe?LNG&amp;Z21ID&amp;I21DBN=NIKA&amp;P21DBN=NIKA&amp;S21STN=1&amp;S21REF=1&amp;S21FMT=fullwebr&amp;C21COM=S&amp;S21CNR=20&amp;S21P01=0&amp;S21P02=1&amp;S21P03=A%3D&amp;S21STR=%D0%9E%D1%80%D0%BB%D0%BE%D0%B2%2C%20%D0%A1%D0%B5%D1%80%D0%B3%D0%B5%D0%B9%20%D0%90%D0%BB%D0%B5%D0%BA%D1%81%D0%B0%D0%BD%D0%B4%D1%80%D0%BE%D0%B2%D0%B8%D1%87) Программная инженерия. Технологии разработки программного обеспечения: учебник / С. А. Орлов. – 5-е изд., обновл. и доп. – Москва ; Санкт- Петербург:Питер, 2016. – 640 с.
2. Джепикс Филипп, Троелсен Эндрю. Язык программирования C# 7 и платформы .NET и .NET Core / Джепикс Филипп, Троелсен Эндрю – М.; Вильямс, 2018. – 1328. «Наука», 2017 – 87с.
3. Павловская Т.А. С/С++. Программирование на языке высокого уровня. СПб.: Питер, 2017. – 461 с.
4. Семакин И. Г. Основы алгоритмизации и программирования : учебник для студ. учреждений сред. проф. образования / И. Г. Се- макин, А. П. Шестаков. — М. : Издательский центр «Академия», 2017. — 304 с
5. Хабибулин И.Ш. Программирование на языке высокого уровня. C/C++. – СПб.: БХВ– Петербург, 2017. – 512 с.
6. Прайс Марк. C# 7 и .NET Core. Кросс-платформенная разработка для профессионалов / Прайс Марк – Москва; Вильямс, 2018. – 604.
7. Скит Джон. C# для профессионалов. Тонкости программирования / Скит Джон – Москва; Вильямс, 2019. – 608.
8. Мюллер Д. П. C# для чайников / Мюллер Д. П. – Москва; Москва, 2019, – 608 с.
9. [Главная | GeekBrains - образовательный портал (gb.ru)](https://gb.ru/)
10. https://opencv.org/courses/
11. <https://robocraft.ru/page/opencv/>
12. Skillbox: <https://live.skillbox.ru/playlists/code/python/>