

**КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Кафедра анализа данных и технологий программирования

К.П.Шустова, Е.П.Шустова

**АНАЛИЗ ИЗОБРАЖЕНИЙ
И РАСПОЗНАВАНИЕ ОБРАЗОВ НА PYTHON**

П Р А К Т И К У М

УЧЕБНОЕ ПОСОБИЕ

Казань – 2023

УДК 51-37
ББК 16.333
Ш97

*Принято на заседании учебно-методической комиссии ИВМиИТ
Протокол № 9 от 25 мая 2023 года*

Рецензенты:

кандидат физико-математических наук,
доцент кафедры высшей математики и математического моделирования
института математики и механики им. Н.И. Лобачевского КФУ,

О.А. Широкова

кандидат физико-математических наук,
доцент кафедры прикладной математики и информатики института
компьютерных технологий и защиты информации КАИ **И.А. Кузьмина**

Шустова К.П., Шустова Е.П.

Анализ изображений и распознавание образов на Python
/К.П.Шустова, Е.П.Шустова. – Казань: Казан. ун-т, 2023.–97 с.

Учебное пособие удовлетворяет требованиям новых государственных образовательных стандартов высшего профессионального образования. Оно предназначено для проведения лабораторных и практических занятий со студентами направлений "Прикладная математика и информатика", "Прикладная информатика", "Информатика и вычислительная техника", "Прикладная математика" по дисциплинам "Введение в анализ изображений", "Программные методы обработки изображений и распознавания образов", "Разработка систем распознавания видео-поток".

Работы составлены таким образом, что, выполнив соответствующие задания, студент овладеет навыками обработки изображений в Python; получения информации с изображения, необходимой для его анализа и принятия решения; а также приобретет опыт создания систем поддержки принятия решений в прикладных областях, где требуется анализ изображений и распознавание в видео-поток.

© Шустова К.П., 2023

© Шустова Е.П., 2023

© Казанский федеральный университет, 2023

ВВЕДЕНИЕ

Учебное пособие удовлетворяет требованиям новых государственных образовательных стандартов высшего профессионального образования. Оно предназначено для проведения лабораторных занятий со студентами направления "Прикладная математика и информатика", "Прикладная информатика", "Информатика и вычислительная техника", "Прикладная математика" по дисциплинам "Введение в анализ изображений", "Программные методы обработки изображений и распознавания образов", "Разработка систем распознавания видео-поток".

Работы составлены таким образом, что, выполнив соответствующие задания, студент овладеет навыками обработки изображений на Python; получения информации с изображения, необходимой для его анализа и принятия решения; а также приобретет опыт создания систем поддержки принятия решений в прикладных областях, где требуется анализ изображений и распознавание образов в видео-потоке.

РАБОТА 1

ЗАГРУЗКА ИЗОБРАЖЕНИЯ И ПОЛУЧЕНИЕ ИНФОРМАЦИИ О НЕМ

Цель работы: ознакомление с основными методами загрузки изображений и получение общей информации о нем с использованием внешней библиотеки PIL.

Подготовка к выполнению заданий

1. На компьютере должен быть выход в интернет.
2. Поместите в текущую папку набор цветных изображений.
3. Установите Python и среду разработки, например, PyCharm (Jupyter Notebook или другую удобную для Вас среду разработки).
4. Поместите не в текущую папку набор других цветных изображений.
5. Установите внешнюю библиотеку PIL (Pillow).

Задание 1. Загрузка изображения

1. Подключите модуль Image из библиотеки PIL. Для этого напишите и запустите на выполнение команду:

```
from PIL import Image
```

2. Откройте файл, расположенный в текущем рабочем каталоге. Для этого напишите и запустите на выполнение команду:

```
img = Image.open("img1.jpg")
```

```
# выводим изображение на экран:
```

```
img.show()
```

3. Откройте этот же файл, расположенный в текущем рабочем каталоге, в бинарном режиме. Для этого напишите и запустите на выполнение команду:

```
img = Image.open("Tulips.jpg")
```

```
img = img.convert("L")
```

а в более ранних, чем Release 7.2.0.dev0 Pillow выполняем команду

```
img = Image.open("img1.jpg", "rb")
```

```
# выводим изображение на экран:
```

```
img.show()
```

4. Вместо указания пути к файлу можно передать файловый объект, открытый в бинарном режиме. Для этого напишите и запустите на выполнение команды:

```
# Открываем файл в бинарном режиме
```

```
f = open("img2.gif", "rb")
```

```
# Передаем объект файла
```

```
img = Image.open(f)
```

```
# выводим переданное изображение на экран:
```

```
img.show()
```

```
f.close() # Закрываем файл
```

5. Загрузите изображение из строки, используя модуль BytesIO (в более ранней версии StringIO), предварительно открыв файл в бинарном режиме:

```
Import io
```

```
from PIL import Image
```

```
f = open("img2.gif", "rb").read()
```

```
img = Image.open(io.BytesIO(f))
```

```
img = img.convert("L")
```

```
# выводим изображение на экран:
```

```
img.show()
```

Задание 2. Получение информации об изображении

1. Получите полную информацию об открытом изображении. Для этого выполните следующее:

```
>>> img = Image.open("foto.jpg")
>>> img.size, img.format, img.mode
((800, 600), 'JPEG', 'RGB')
>>> img.info
{'jfif': 258, 'jfif_unit': 0, 'adobe': 100, 'progression': 1,
'jfif_version': (1, 2), 'adobe_transform': 100,
'jfif_density': (100, 100)}
```

2. Получите информацию о цветовой модели изображения.

```
>>> img = Image.open("foto.jpg")
>>> img.mode
'RGB'
```

3. Получите информацию о размере изображения.

img.size

В консоль выведется размер повернутого на 90 градусов изображения:

(600, 800)

4. Получите координаты прямоугольной области, в которую вписывается всё изображение **img** .

img.getbbox()

(0, 0, 800, 600)

ЛИТЕРАТУРА К РАБОТЕ

1. Python Imaging Library (Fork) . – URL: <https://pypi.org/project/Pillow/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobходимое_proxorenok/ (дата обращения: 10.05.2023).

3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).
4. Alex Clark Pillow (PIL Fork) Documentation, Release 7.2.0.dev0, Apr 21, 2020).– URL: <https://readthedocs.org/projects/pillow/downloads/pdf/latest/> (дата обращения: 10.05.2023).

РАБОТА 2

ЦВЕТОВЫЕ МОДЕЛИ

СОЗДАНИЕ И СОХРАНЕНИЕ ИЗОБРАЖЕНИЯ

Цель работы: ознакомление со способами создания и сохранения изображения, а так же с цветовыми моделями в Python с использованием внешней библиотеки PIL.

Подготовка к выполнению заданий

1. На компьютере должен быть выход в интернет.
2. Поместите в текущую папку набор цветных изображений.
3. Поместите не в текущую папку набор других цветных изображений.
4. Установите внешнюю библиотеку PIL.

Задание 1. Получение и изменение цвета пиксела

1. Получите и измените цвет пиксела методом `load()`.

Сначала получите информацию о цветовой модели открытого изображения:

```
>>> img = Image.open("foto.jpg")
>>> img.mode
'RGB'
```

Теперь получите цвет пиксела:

```
obj = img.load()
obj[25, 45] # Получаем цвет пикселя
(122, 86, 62)
# Задаем цвет пикселя (красный)
obj[25, 45] = (255, 0, 0)
img.show() # Просматриваем изображение (цвет пикселя будет
изменен на obj[25, 45], заданный выше.
```

2. Получите и измените цвет пиксела методами `getpixel()` и `putpixel()`.

```
# Получаем цвет пикселя
img.getpixel((25, 45))
(255, 0, 0)
# Изменяем цвет пикселя
img.putpixel((25, 45), (0, 0, 255))
# Получаем цвет пикселя
img.getpixel((25, 45))
(0, 0, 255)
img.show() # Просматриваем изображение
```

Задание 2. Переход от одной цветовой модели изображения к другой

1. Преобразуйте изображение из режима **RGB** в **RGBA**, используя `split()` и `merge`.

`split()` – возвращает каналы изображения в виде кортежа.

`merge(<Режим>, <Каналы>)` – собирает изображение из каналов

(операция, обратная **split**).

```
>>> img = Image.open("foto.jpg")
>>> img.mode
'RGB'
>>> R, G, B = img.split()
>>> mask = Image.new("L", img.size, 128)
>>> img2 = Image.merge("RGBA", (R, G, B, mask) )
>>> img2.mode
'RGBA'
>>> img2.show()
```

2. Преобразуйте изображение из режима **RGB** в **RGBA**, используя **convert()**.

```
img = Image.open("foto.jpg")
```

```
img.mode
```

```
'RGB'
```

```
img2 = img.convert("RGBA")
```

```
img2.mode
```

```
'RGBA'
```

```
img2.show()
```

#прочитать изображение и конвертировать

его в градациях серого:

```
pil_im = Image.open('empire.jpg').convert('L')
```

3. Преобразуйте изображение **RGB** в формат **P**, указав смешивание цветов и адаптивную палитру в 128 цветов.

```
>>> img = Image.open("foto.jpg")
>>> img.mode
'RGB'
>>> img2 = img.convert("P", None,
Image.FLOYDSTEINBERG, Image.ADAPTIVE, 128)
>>> img2.mode
'p'
```

Задание 3. Сохранение изображения

Сохраните изображение в текущую папку.

Сохранение изображения в текущую папку в формате **JPEG**:

```
img.save("tmp.jpg")
# в формате BMP:
img.save("tmp.bmp", "BMP")
f = open("tmp2.bmp", "wb")
# Передаем файловый объект
img.save(f, "BMP")
f.close()
```

Задание 4. Создание нового изображения

1. Создайте черный квадрат.

```
img = Image.new("RGB", (100, 100))
img.show() # Черный квадрат
```

2. Создайте красный квадрат.

```
img = Image.new("RGB", (100, 100), (255, 0, 0))
img.show() # Красный квадрат
```

3. Создайте Зеленый квадрат.

```
img = Image.new("RGB", (100, 100), "green")
img.show() # Зеленый квадрат
```

4. Создайте красный квадрат.

```
img = Image.new("RGB", (100, 100), "#f00")
img.show() # тоже красный квадрат
```

5. Создайте белый квадрат.

```
img = Image.new("RGB", (100, 100), "white")
img.show() # Белый квадрат
```

6. Создайте серый квадрат.

```
img = Image.new("RGB", (320, 240), "silver")
img.show() # Серый квадрат
```

7. Создайте цветной прямоугольник.

```
img = Image.new("RGB", (320, 240), "rgb(205,
```

```
100,200) ")
```

```
img.show() # цветной прямоугольник
```

8. Создайте цветной прямоугольник (Каналы в процентах).

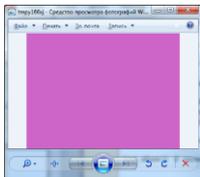
```
img = Image.new("RGB", (320, 240), "rgb(10%,  
100%,40%) ")
```

```
img.show() # цветной прямоугольник. Каналы в процентах
```

9. Создайте цветной прямоугольник заданного цвета. Затем поменяйте этот цвет.

```
img = Image.new("RGB", (640, 480), "rgb(205,  
100,200) ")
```

```
img.show() # сиреневый прямоугольник
```



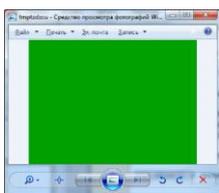
```
for x in xrange(640):
```

```
    for y in xrange(480):
```

```
        img.putpixel((x,y), (0,160,0))
```

```
img.save("okno.png", "PNG")
```

```
img.show() # зеленый прямоугольник
```



10. Создайте прямоугольник заданного цвета (функциональная раскраска).

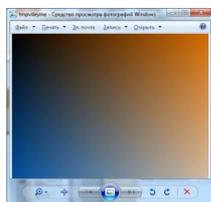
```
img = Image.new("RGB", (640, 480), "rgb(205,  
100,200) ")
```

```
img.show() # сиреневый прямоугольник
```

```
for x in xrange(640):
```

```
    for y in xrange(480):
```

```
img.putpixel((x,y),  
x/3,(x+y)/6,y/3)  
img.save("okno.png", "PNG")  
img.show() # прямоугольник с функциональной раскраской
```



ЛИТЕРАТУРА К РАБОТЕ

1. Python Imaging Library (Fork) . – URL: <https://pypi.org/project/Pillow/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobходимое_proxorenok/ (дата обращения: 10.05.2023).
3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).

РАБОТА 3

КАНАЛЫ ЦИФРОВОГО ИЗОБРАЖЕНИЯ

ГИСТОГРАММА ИЗОБРАЖЕНИЯ

Цель работы: ознакомление со способами извлечения каналов из цифрового изображения и сборки изображения из заданных каналов, а так же научиться строить гистограмму изображения в Python с использованием внешней библиотеки PIL.

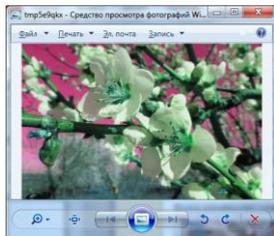
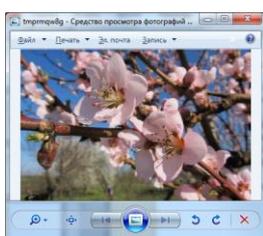
Подготовка к выполнению заданий

1. Поместите в текущую папку набор цветных изображений и изображений в оттенках серого.
2. Поместите не в текущую папку набор других цветных изображений и изображений в оттенках серого.
3. На компьютере должен быть выход в интернет.
4. Установите внешнюю библиотеку PIL.

Задание 1. Замена каналов. Методы `Getpixel`, `putpixel`

Сначала откройте изображение. Затем поменяйте каналы в этом изображении. Для этого выполните следующее:

```
img = Image.open("tsveti.jpg")
img.show() # цвета
for x in xrange(img.size[0]):
    for y in xrange(img.size[1]):
        r,g,b = img.getpixel((x,y))
        img.putpixel((x,y), (b, r,g))
img.show() # замена каналов
```



Задание 2. Сборка изображения из известных каналов изображения

1. Выделите каналы изображения.

```
>>> img = Image.open("foto.jpg")
```

```
>>> R, G, B = img.split()
```

split() – выдает каналы изображения в виде кортежа.

2. Соберите изображение из каналов, полученных выше.

```
>>> img2 = Image.merge("RGB", (R, G, B) )
```

```
>>> img2.mode
```

```
'RGB'
```

```
>>> img2.show()
```

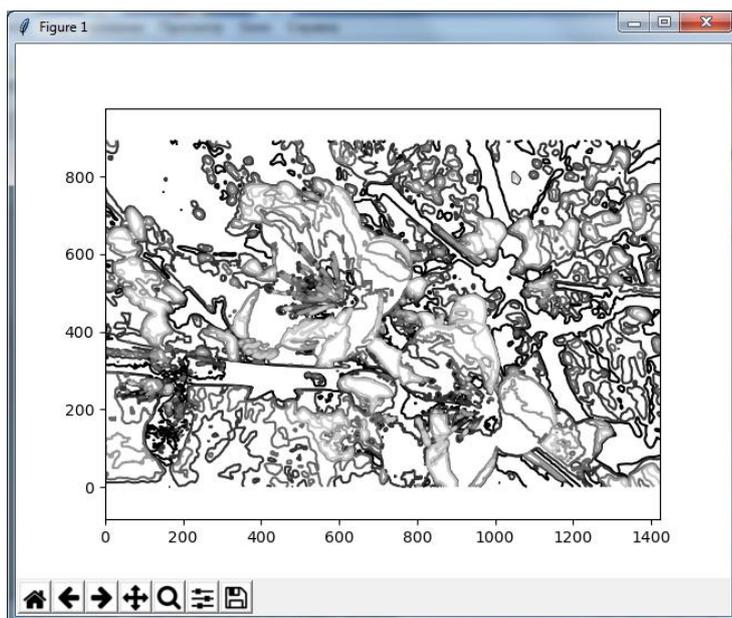
Задание 3. Построение гистограммы

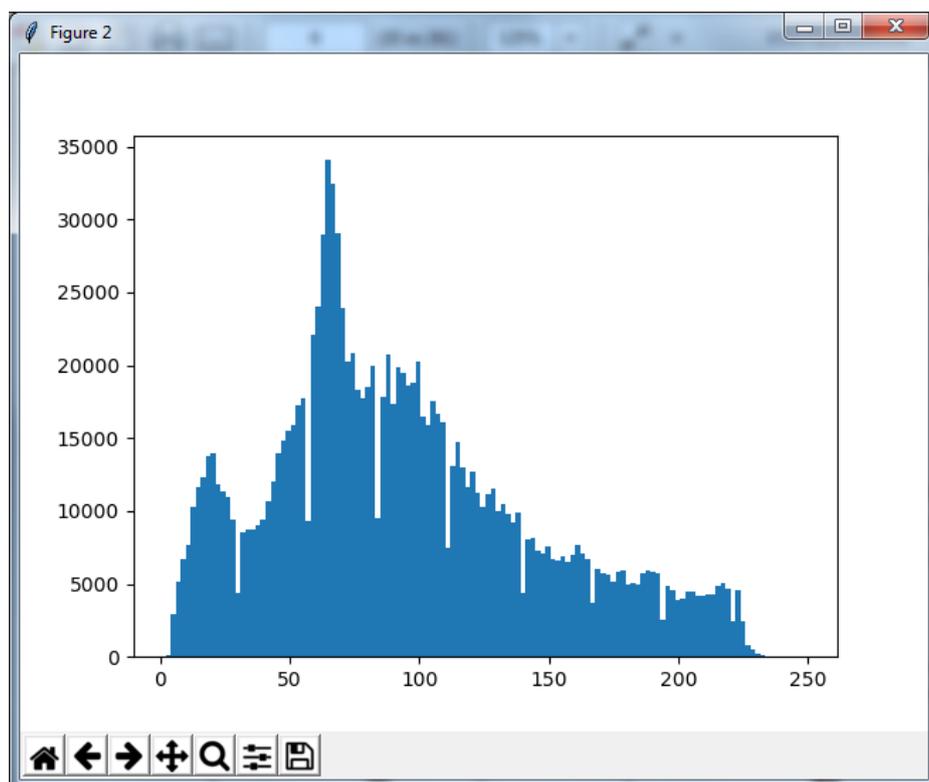
Откройте изображение в переменную im и постройте гистограмму изображения.

Ниже приведен код построения гистораммы.

```
40 """  
41 figure()      #Выдать окно  
42 hist(im.flatten(),128)      #построить гистограмму  
43 show()|
```

Ниже показано открытое исходное изображение и его гистограмма.





ЛИТЕРАТУРА К РАБОТЕ

1. Python Imaging Library (Fork) . – URL: <https://pypi.org/project/Pillow/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobходимое_proxorenok/ (дата обращения: 10.05.2023).
3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).

РАБОТА 4

МАНИПУЛИРОВАНИЕ ИЗОБРАЖЕНИЕМ

Цель работы: научиться на Python с использованием внешней библиотеки PIL копировать изображение, изменять размеры изображения, вырезать область на изображении и изменять размеры вырезанной области, вращать изображение.

Подготовка к выполнению заданий

1. Поместите в текущую папку набор цветных изображений и изображений в оттенках серого.
2. Поместите не в текущую папку набор других цветных изображений и изображений в оттенках серого.
3. На компьютере должен быть выход в интернет.
4. Установите внешнюю библиотеку PIL.

Задание 1. Копирование и изменение размера изображения

1. Скопируйте открытое изображение в и выведите на экран полученную копию

```
# Подключаем модуль:  
from PIL import Image  
  
# Открываем файл:  
img = Image.open("foto.jpg")  
  
# Создаем копию:  
img2 = img.copy()  
  
# Просматриваем копию:  
img2.show()
```

2. Создайте уменьшенную версию изображения указанного размера.

Сначала откройте некоторое изображение и выведите его размеры

```
img=Image.open("foto.jpg")  
img.size # Исходные размеры изображения
```

В итоге в консоль выведется размер этого изображения:

```
(800, 600)
```

Измените размер изображения пропорционально с помощью метода

```
thumbnail (<Размер> [, <Фильтр>])
```

Обратите внимание на то, что изменение размера производится пропорционально, иными словами, за основу берется минимальное значение, а второе значение вычисляется пропорционально первому.

Метод изменяет само изображение и ничего не возвращает.

Фильтры **NEAREST**, **BILINEAR**, **BICUBIC** или **ANTIALIAS** .

```
img.thumbnail((400, 300), Image.ANTIALIAS)
```

```
img.size # Изменяется само изображение
```

В итоге в консоль выведется размер измененного изображения:

```
(400, 300)
```

```
img = Image.open("foto.jpg")
```

```
img.thumbnail((400, 100), Image.ANTIALIAS)
```

```
img.size # Вывод размера изображения
```

Увидим, что размер изменяется пропорционально

В итоге в консоль выведется размер измененного изображения:

```
(133, 100)
```

3. Создайте версию изображения указанного размера

Сначала откройте некоторое изображение и выведите его размеры

```
img = Image.open("foto.jpg")
```

```
img.size # Исходные размеры изображения
```

В итоге в консоль выведется размер измененного изображения:

```
(800, 600)
```

Теперь изменим само изображение

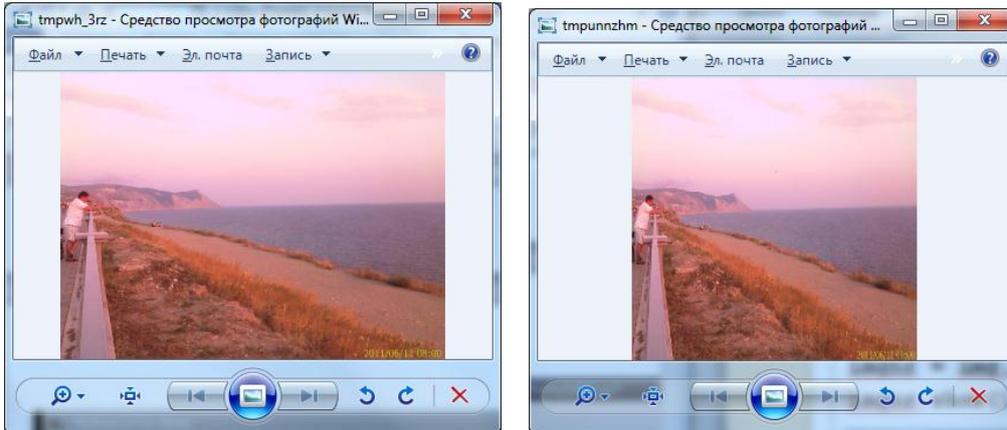
```
img.show()
```

```
newsize = (400, 400)
```

```
imgnr = img.resize(newsize)
```

`imgnr.show()` # Изменяется само изображение

В итоге получим:



Задание 2. Вырезка области изображения и изменение его размеров

1. Вырежьте прямоугольный фрагмент из исходного изображения.

```
>>> img = Image.open("foto.jpg")
>>> img2 = img.crop( [0, 0, 100, 100] ) # Помечаем фрагмент
>>> img2.load() # Считываем фрагмент, создавая новое изображение
>>> img2.size
(100, 100)
```

2. Измените размеры области изображения

Сначала откройте некоторое изображение и выведите его рамеры

```
img = Image.open("tsveti.jpg")
img.size # вывод размеров изображения
img.show()
```

В итоге откроется исходное изображение и в консоль выведется размер исходного изображения:

```
(668, 438)
```

Затем возьмем кусок изображения

```
box = (100, 100, 300, 300) # берем кусок изображения:
```

Затем изменяем само изображение

```
img2 = img.crop(box)
newsize = (400, 400)
```

```
img2nr = img2.resize(newsize)
```

```
img2nr.show() # Изменяется само изображение
```

Получим в итоге:



Задание 3. Вращение изображения

1. Осуществите вращение изображения. Но сначала откройте некоторое изображение и выведите его размеры в консоль.

```
img = Image.open("foto.jpg")
```

```
img.size # Исходные размеры изображения
```

В консоль выведется размер исходного изображения:

```
(800, 600)
```

Затем осуществите поворот на 90 градусов

```
img2 = img.rotate(90) # Поворот на 90 градусов
```

```
img2.size
```

В консоль выведется размер повернутого на 90 градусов изображения:

```
(600, 800)
```

```
img3 = img.rotate(45, Image.NEAREST)
```

```
img3.size # Размеры сохранены, изображение обрезано
```

```
(800, 600)
```

```
img4 = img.rotate(45, expand=True)
```

```
img4.size # Размеры увеличены, изображение полное
```

```
(991, 990)
```

2. Получите горизонтальный и зеркальный образ изображения:

```
>>> img2 = img.transpose(Image.FLIP_LEFT_RIGHT)
>>> img2.show() # Горизонтальный зеркальный образ
>>> img3 = img.transpose(Image.FLIP_TOP_BOTTOM)
>>> img3.show() # Вертикальный зеркальный образ
```

и следующими способами:

```
>>> img4 = img.transpose(Image.ROTATE_90)
>>> img4.show() # Поворот на 90° против часовой стрелки
>>> img5 = img.transpose(Image.ROTATE_180)
>>> img5.show() # Поворот на 180°
```

ЛИТЕРАТУРА К РАБОТЕ

1. Python Imaging Library (Fork) . – URL: <https://pypi.org/project/Pillow/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobходимое_proxoreнок/ (дата обращения: 10.05.2023).
3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).

РАБОТА 5

МАНИПУЛЯЦИИ НА ИЗОБРАЖЕНИИ

Цель работы: ознакомление со способами закраски (заливки) части и всего изображения заданным цветом, вывода координат кликнувших на картинке точек, вставки объектов на изображение, рисования на изображении с использованием внешней библиотеки PIL в Python.

Подготовка к выполнению заданий

Поместите в текущую папку набор цветных изображений и изображений в оттенках серого.

1. Поместите не в текущую папку набор других цветных изображений и изображений в оттенках серого.
2. На компьютере должен быть выход в интернет.
3. Установите внешнюю библиотеку PIL.

Задание 1. Закраска (заливка) части и всего изображения заданным цветом

1. Закрасьте область изображения красным цветом. Для этого сначала откройте исходное изображение:

```
img = Image.open("foto.jpg")
```

Затем закрасьте на нем прямоугольную область:

```
img.paste( (255, 0, 0), (0, 0, 100, 100) )
```

```
img.show()
```

2. Залейте все изображение зеленым цветом.

```
img = Image.open("foto.jpg")
```

```
img.paste( (0, 128, 0), img.getbbox() )
```

```
img.show()
```

Задание 2. Комбинированные манипуляции на изображении

1. Загрузите изображение, создайте его уменьшенную, а затем вставьте ее в исходное изображение, причем вокруг вставленного изображения отобразите рамку красного цвета.

```
>>> img = Image.open("foto.jpg")
>>> img2 = img.resize( (200, 150) ) # Создаем миниатюру
>>> img2.size
(200, 150)
>>> img.paste( (255, 0, 0), (9, 9, 211, 161) ) # Рамка
>>> img.paste(img2, (10, 10) ) # Вставляем миниатюру
>>> img.show()
```

2. Выведите белую полупрозрачную горизонтальную полосу высотой 100 пикселей. Для этого выполните:

```
>>> img = Image.open("foto.jpg")
>>> white = Image.new("RGB", (img.size[0],100), (255,255,255))
>>> mask = Image.new("L", (img.size[0], 100), 64) # Маска
>>> img.paste(white, (0, 0), mask)
>>> img.show()
```

3. Копируйте и вставьте повернутую часть изображения:

```
box = (100,100,400,400)
region = pil_im.crop(box)
region = region.transpose(Image.ROTATE_180)
pil_im.paste(region,box)
```

Задание 3. Вывод координат кликнутых на картинке точек

Откройте в картинку, выведите ее на экран и напишите программу, выводящую координаты трех кликнутых на картинке точек:

```
56 from PIL import Image
57 from pylab import *
58 im = array(Image.open("tsveti.jpg"))
59 imshow(im)
60 print("Please click 3 points")
61 x = ginput(3)           потом на изображении нажмите 3 точки
62 print('you clicked:',x)
```

Python

~\Documents\2018 fleshki\Evg4\internet1\Python

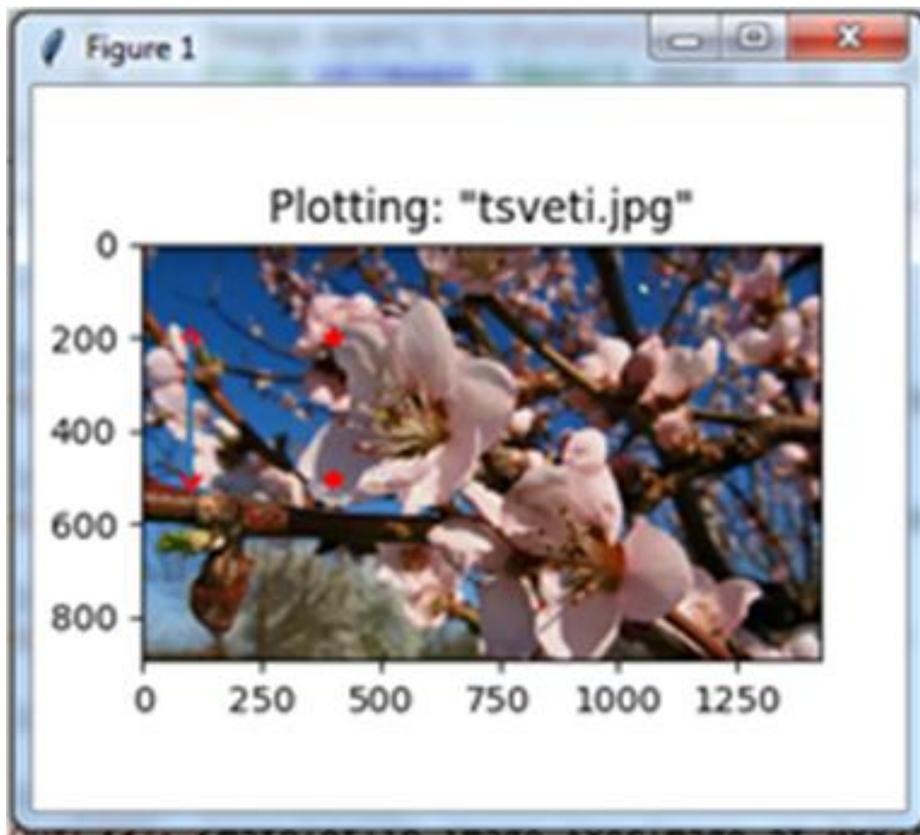
```
In [72]: print('you clicked:',x)
you clicked: [(1145.0498472116119, 735.42929585627348), (818.06425673717763,
368.45266194251997), (425.21106530386453, 439.02509154131872)]
```

```
In [73]: |
```

Задание 4. Рисование на изображении

1. Поставьте точки и нарисуйте линии на изображении.

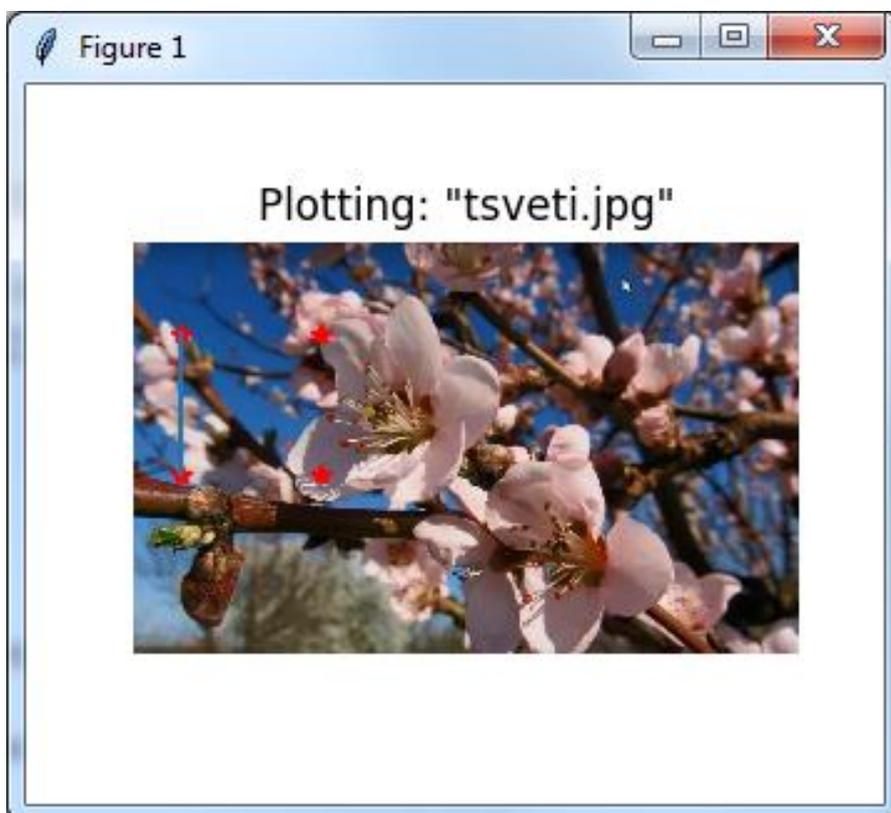
```
1 from PIL import Image
2 from pylab import *
3 # запишем файл-изображение в массив (read image to array)
4 im = array(Image.open("tsveti.jpg"))
5 # plot the image
6 imshow(im)
7 # some points
8 x = [100,100,400,400]
9 y = [200,500,200,500]
10 # plot the points with red star-markers
11 plot(x,y, 'r*')
12 # line plot connecting the first two points
13 plot(x[:2],y[:2])
14 # add title and show the plot
15 title('Plotting: "tsveti.jpg"')
16 show()
```

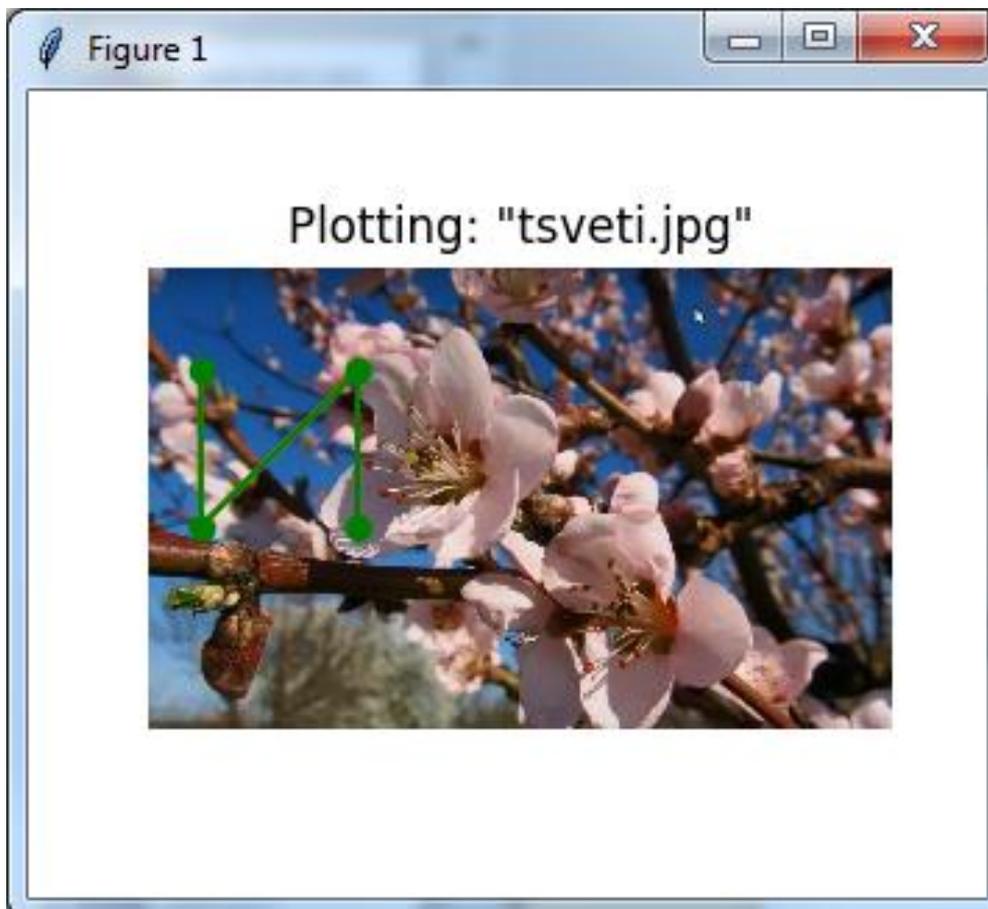
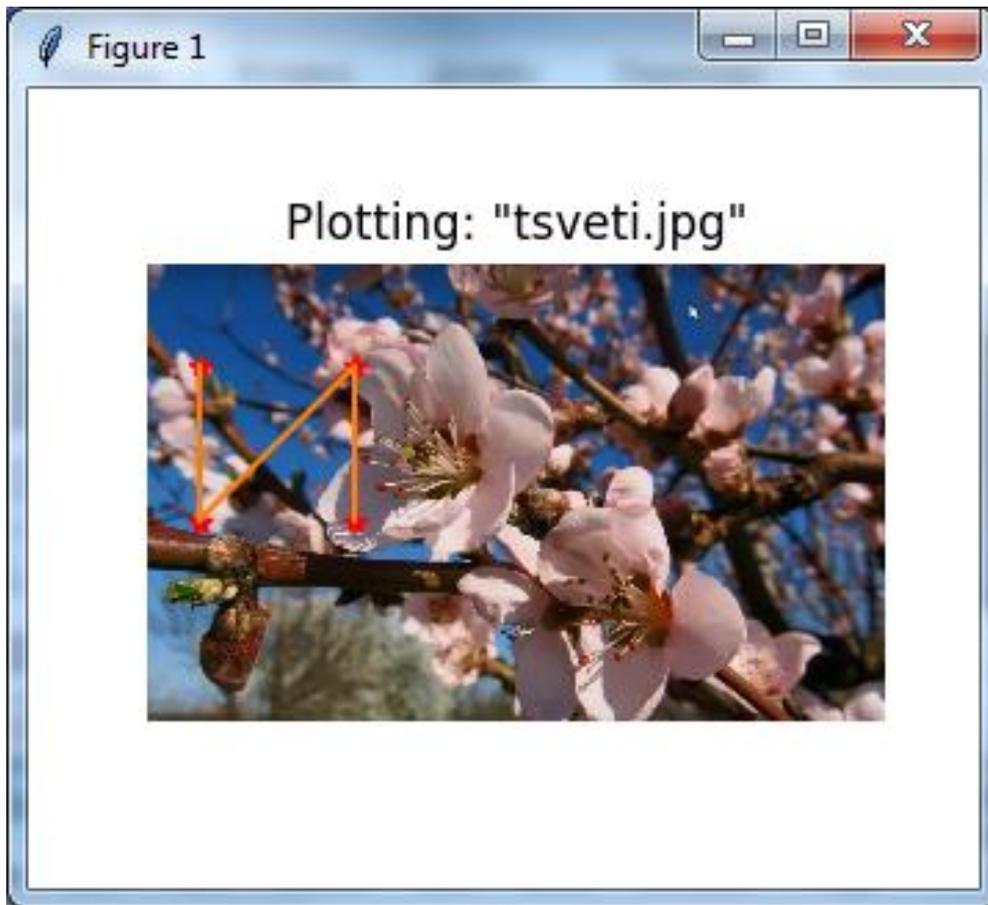


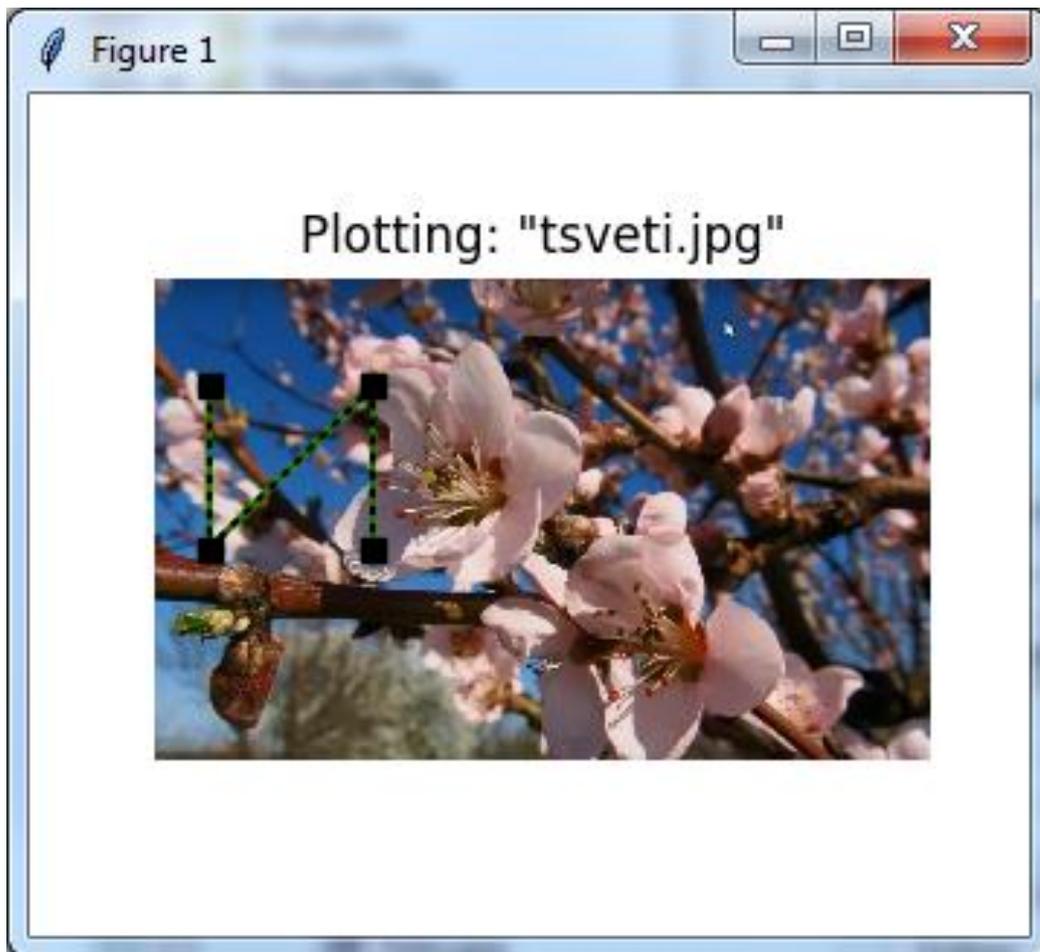
```

17 axis('off') #убрать оси
18 plot(x,y) #синяя сплошная линия по умолчанию
19 plot(x,y,'go-') # зеленая линия с круглыми маркерами
20 plot(x,y,'ks:') # бчерная пунктирная линия с квадратными маркерами

```







ЛИТЕРАТУРА К РАБОТЕ

1. Python Imaging Library (Fork) . – URL: <https://pypi.org/project/Pillow/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobходимое_proxorenok/ (дата обращения: 10.05.2023).
3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).

РАБОТА 6

ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЙ. КОНТУРЫ НА ИЗОБРАЖЕНИИ

Цель работы: ознакомление со способами фильтрация изображений и получения контуров объектов на изображении с использованием внешней библиотеки PIL в Python.

Подготовка к выполнению заданий

1. Поместите в текущую папку набор цветных изображений. Среди них должны быть изображения:
 - резкое, т.е. в котором все детали видны четко,
 - размытое,
 - с шумами разных видов,
 - резкое, но в тенях плохо видны детали.
2. Поместите не в текущую папку набор других цветных изображений. Среди них тоже должны быть изображения:
 - резкое, т.е. в котором все детали видны четко,
 - размытое,
 - с шумами разных видов,
 - резкое, но в тенях плохо видны детали.
3. На компьютере должен быть выход в интернет.
4. Установите внешнюю библиотеку PIL.

Замечание 1. Полезная функция

В дальнейшем нам понадобятся списки изображений для обработки. Вот как можно создать список имен файлов всех изображений в папке:

```
import os

def get_imlist(path):
    """ Returns a list of filenames for
        all jpg images in a directory. """

    return [os.path.join(path,f) for f in os.listdir(path) if f.endswith('.jpg')]
```

Примите все файлы изображений в список имен файлов (список файлов) и преобразуйте изображения к файлам **JPEG**.

Сначала получите **filelist** указано в замечании Затем выполните:

```
from PIL import Image
import os

for infile in filelist:
    outfile = os.path.splitext(infile)[0] + ".jpg"
    if infile != outfile:
        try:
            Image.open(infile).save(outfile)
        except IOError:
            print "cannot convert", infile
```

Задание 1. Фильтрация изображений

1. Примените к изображению каждый из следующих фильтров:
BLUR, CONTOUR, DETALL, EDGE_ENHANCE, EDGE_ENHANCE_MORE,
EMBOSS, FIND_EDGES, SHARPEN, SMOOTH и SMOOTH_MORE
из модуля **ImageFilter** и выведите отфильтрованное изображение на экран. Например:

```
>>> from PIL import ImageFilter
>>> img = Image.open("foto.jpg")
>>> img2 = img.filter(ImageFilter.EMBOSS)
>>> img2.show()
```

Задание 2. Контур на изображении (один из способов)

1. Выделите контуры на изображении.

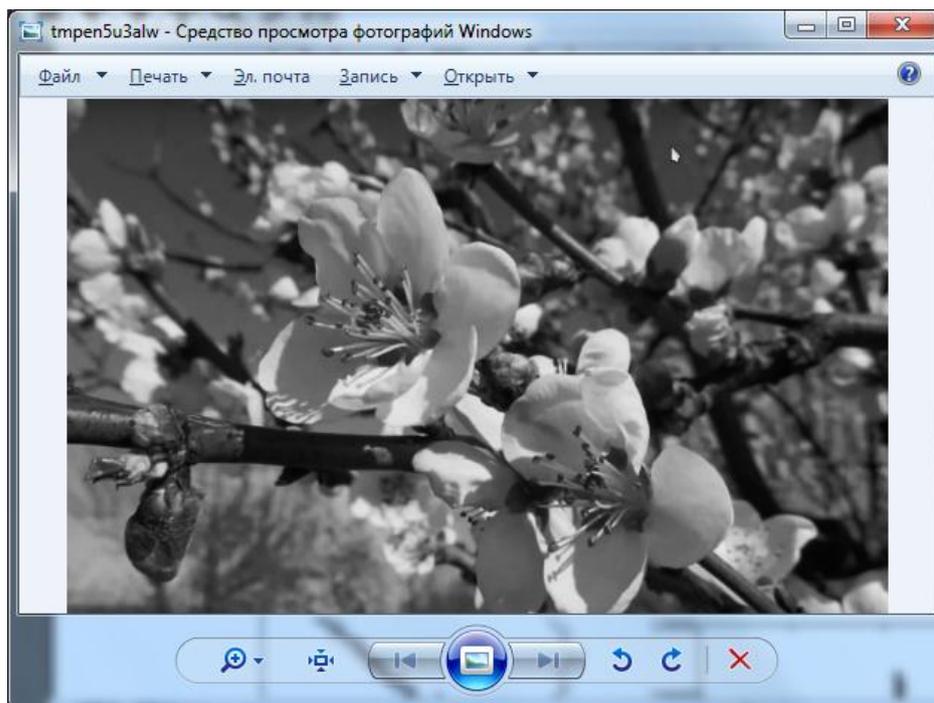
```
from PIL import Image
from pylab import *

# read image to array
im = array(Image.open('empire.jpg').convert('L'))

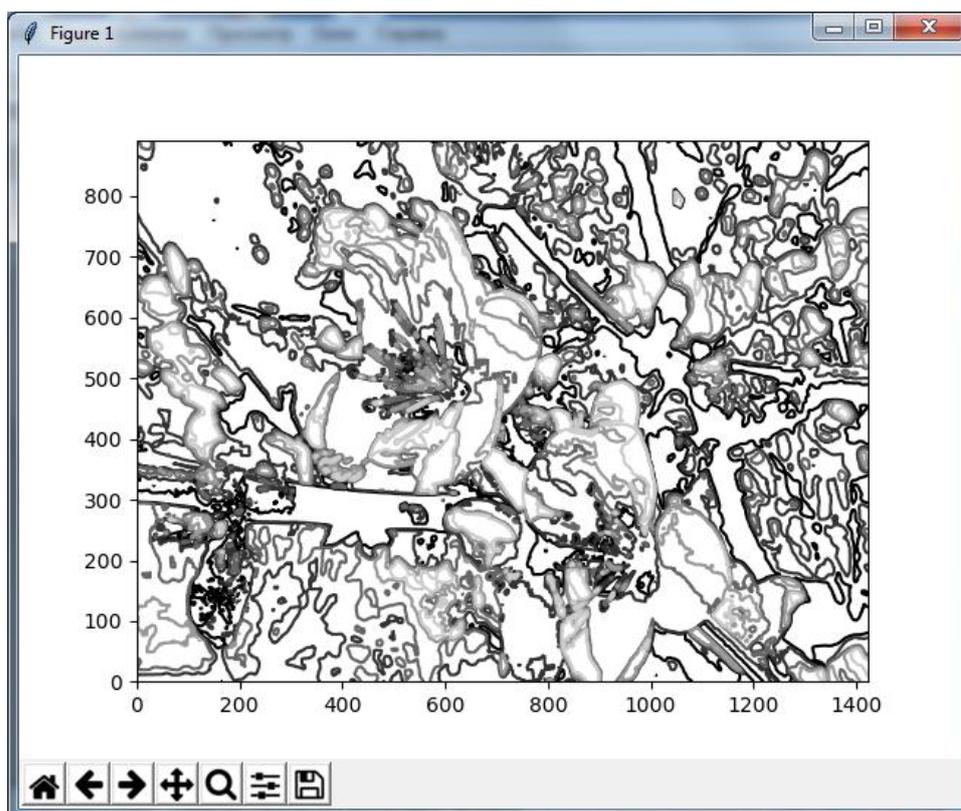
# create a new figure
figure()
# don't use colors
gray()
# show contours with origin upper left corner
contour(im, origin='image')
axis('equal')
axis('off')
```

Откроем исходное изображение, запустив команду:

```
im.show()
```



После выполнения приведенной выше программы на экране получим:



ЛИТЕРАТУРА К РАБОТЕ

1. Python Imaging Library (Fork) . – URL: <https://pypi.org/project/Pillow/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobходимое_proxorenok/ (дата обращения: 10.05.2023).
3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).

РАБОТА 7

СППР «ОПРЕДЕЛЕНИЕ КАЧЕСТВА ПОВЕРХНОСТИ ПОРИСТОГО МАТЕРИАЛА». Часть 1

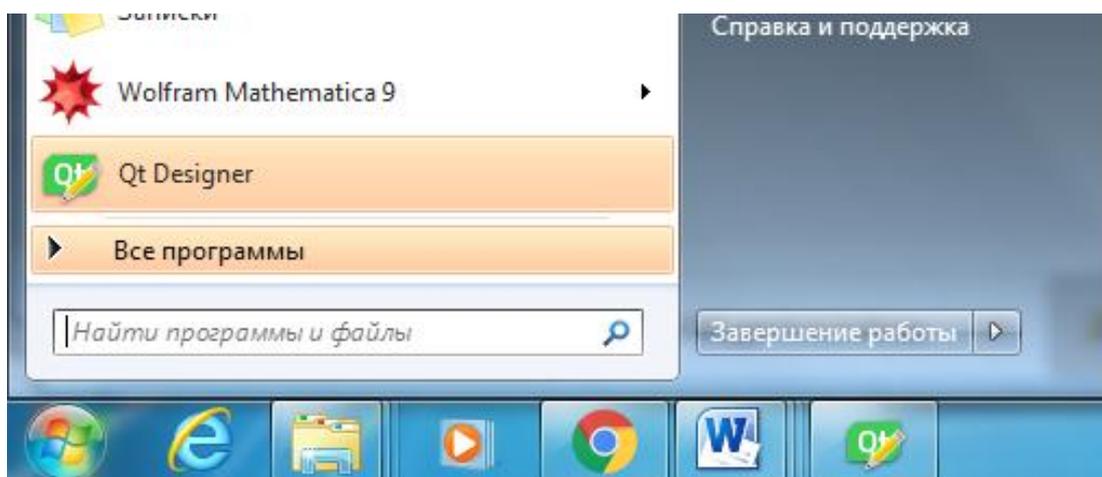
Цель работы: создание интерфейса СППР «Определение качества поверхности пористого материала».

Подготовка к выполнению задания.

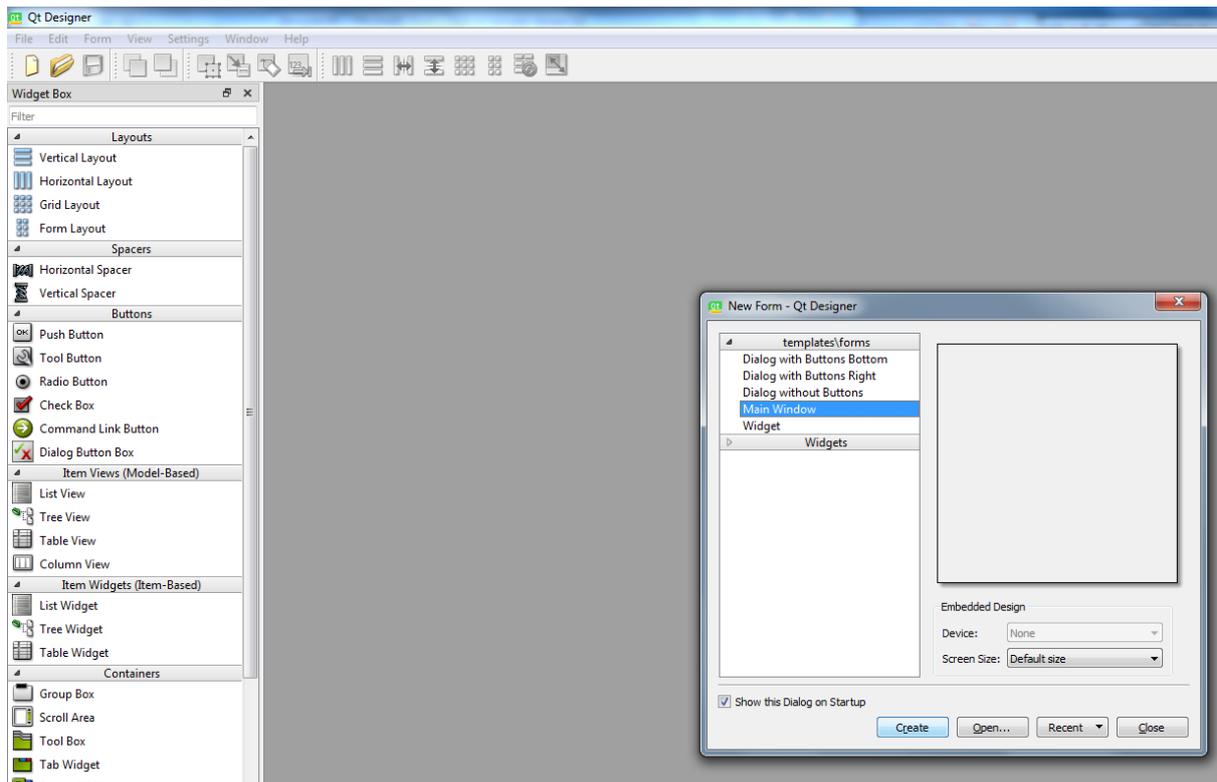
1. На компьютере должен быть выход в интернет.
2. Скачайте и установите внешнюю библиотеку **PyQt5**. Ссылки для скачивания: <https://sourceforge.net/projects/pyqt/files/>, <https://www.riverbankcomputing.com> или <https://www.lfd.uci.edu/~gohlke/pythonlibs/#vlfid>.
3. Скачайте и установите программу **QtDesigner**. Ссылка для скачивания <https://build-system.fman.io/qt-designer-download>.

Задание. Создание интерфейса СППР.

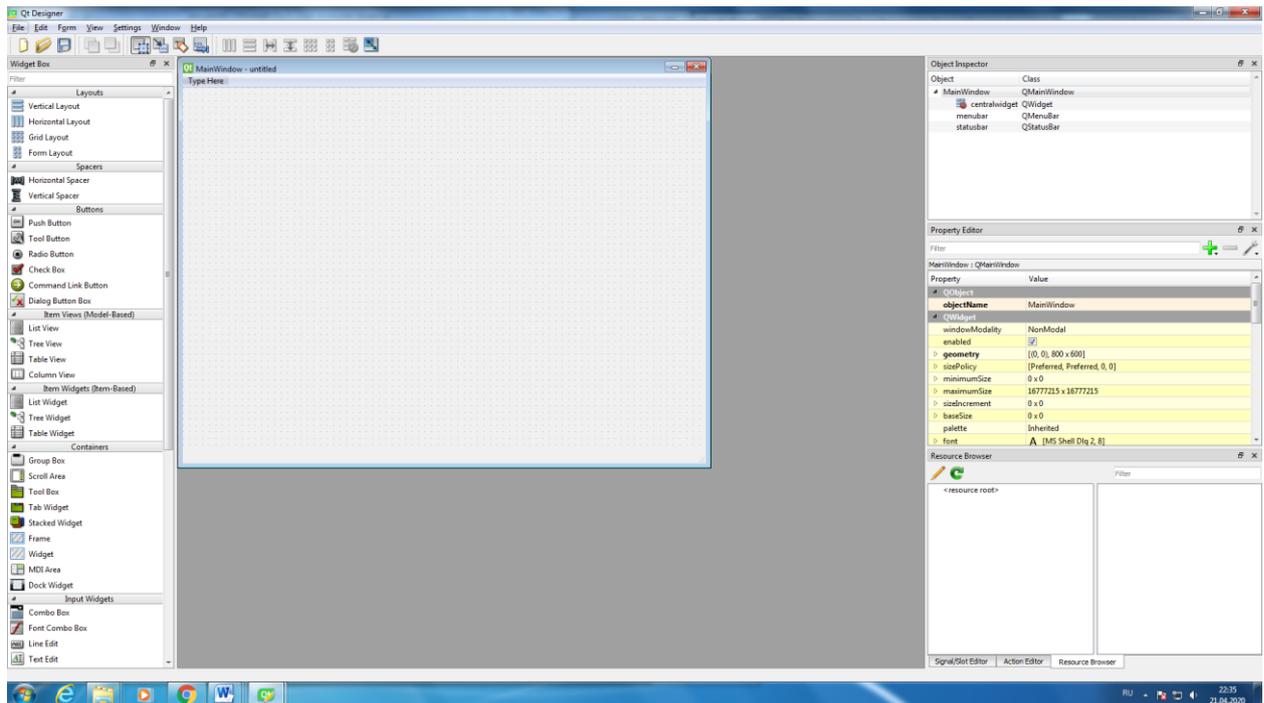
Войдите в программу **Designer**, например, через проводник, кликнув на **Designer**:



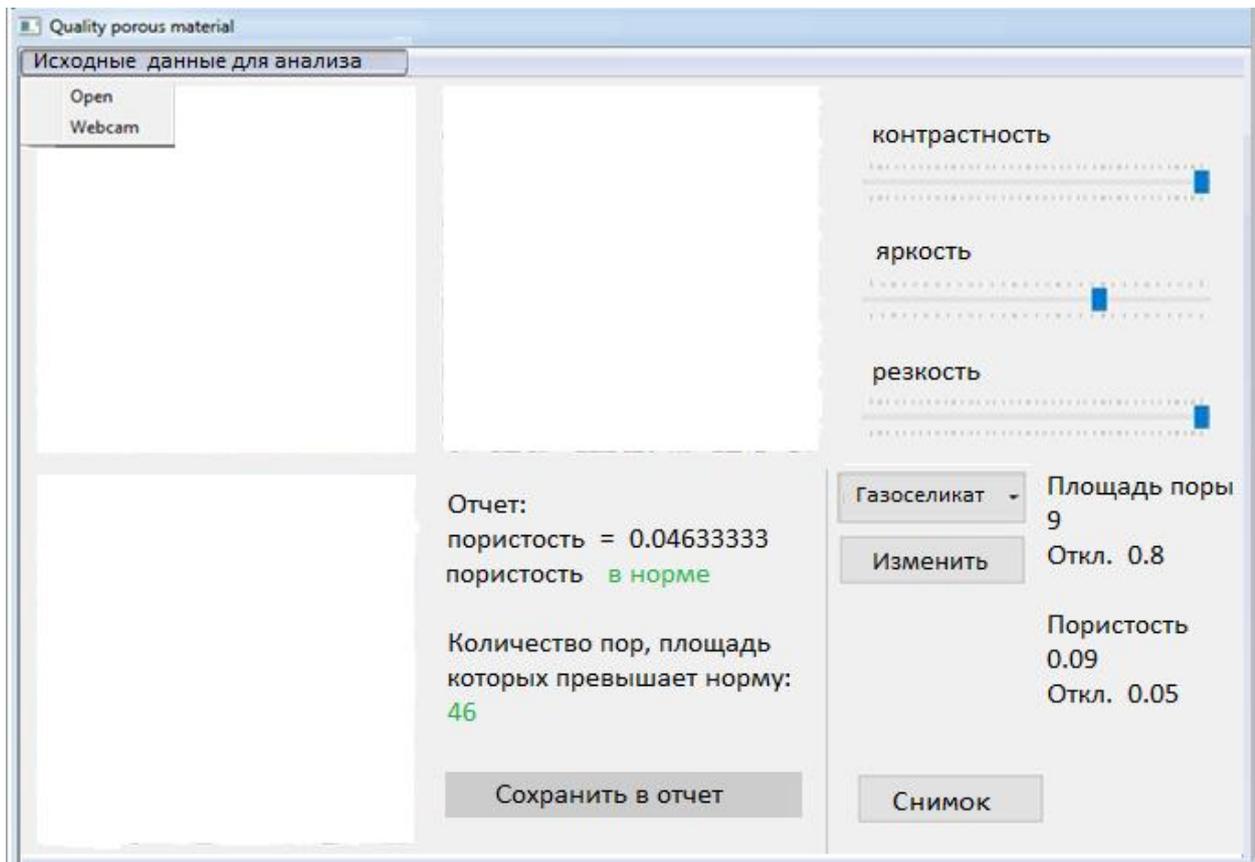
Откроется главное окно программы **Designer**. Выберите в нем **MainWindows** и кликнете **Creator**:



Откроется форма:



Перетаскивая в рабочую область окна нужные виджеты, создайте интерфейс СППР «Определение качества поверхности пористого материала». Примерный интерфейс этой СППР:



В левом верхнем окне размещается текущее изображение материала, справа от него – изображение для анализа системой, в нижнем окне – оставляются только те поры, площадь которых больше допустимой.

С помощью движков, расположенных в правой области формы, ЛПР может улучшить качество изображения для анализа системой. Тем самым ЛПР может настроить СППР для работы.

После клика на кнопку «Изменить» в готовой СППР должно открываться диалоговое окно для работы с БД материалов. Создайте интерфейс и этого окна:

Dialog

ID	Наименование	Площадь поры	Откл. от площади	Пористость	Откл. от пористости
1 0	Материал2	12.0	5.0	0.1	0.01
2 1	Материал3	9.0	8.0	0.15	0.01
3 4	материал1	6.0	4.0	0.2	0.02

Добавить запись

Удалить запись

Исходная база данных в одном из следующих заданий будет прописываться в коде программы. Поэтому при открытии этого диалогового окна будет отображаться в верхней его области. В ней можно увидеть количество материалов, их названия, порядковые номера и параметры. В нижней области формы мы можем изменить (вставить или удалить) строки в уже существующей базе данных. Таким образом, у пользователя есть возможность добавления нового материала в базу данных, изменения текущих значений параметров, а также удаления материалов из базы. Но пока на этой форме верхнее окно пустое и кнопки не действуют.

Кликнув на кнопке с разворачивающимся меню, расположенной на главной форме СППР выше только что указанной, выбираем из разворачивающегося списка наименование пористого материала для контроля качества. После этого справа от этой кнопки появляются считанные из БД характеристики нормы для выбранного материала.

В кнопке «Исходные данные для анализа» выбирается файл с текущим изображением или указывается, что текущее изображение поступает с камеры в режиме реального времени или берется из файла.

Рядом с нижним окном – область отчета по результатам анализа.

Пока на форме нашей СППР есть виджеты, но они не действуют. Т.о., Вы должны создать файлы с интерфейсом форм нашей системы.

1. Сохраните свои файлы, подготовленные в **Qt Designer** как **uisppr.ui** и **uibd.ui**.
2. Далее можем поступить одним из приведенных ниже двух способов:

Способ 1. Использовать .ui-файлы напрямую из Python-кода.

Создайте другой файл, скажем **uisppr.py**, со следующим содержимым:

```
from PyQt5 import uic
from PyQt5.QtWidgets import QApplication

Form, Window = uic.loadUiType("uisppr.ui")

app = QApplication([])
window = Window()
form = Form()
form.setupUi(window)
window.show()
app.exec_()
```

Аналогично создайте другой файл, скажем **uibd.py**, со следующим содержимым:

```
from PyQt5 import uic
from PyQt5.QtWidgets import QApplication
```

```
Form, Window = uic.loadUiType("uibd.ui")
```

```
app = QApplication([])
```

```
window = Window()
```

```
form = Form()
```

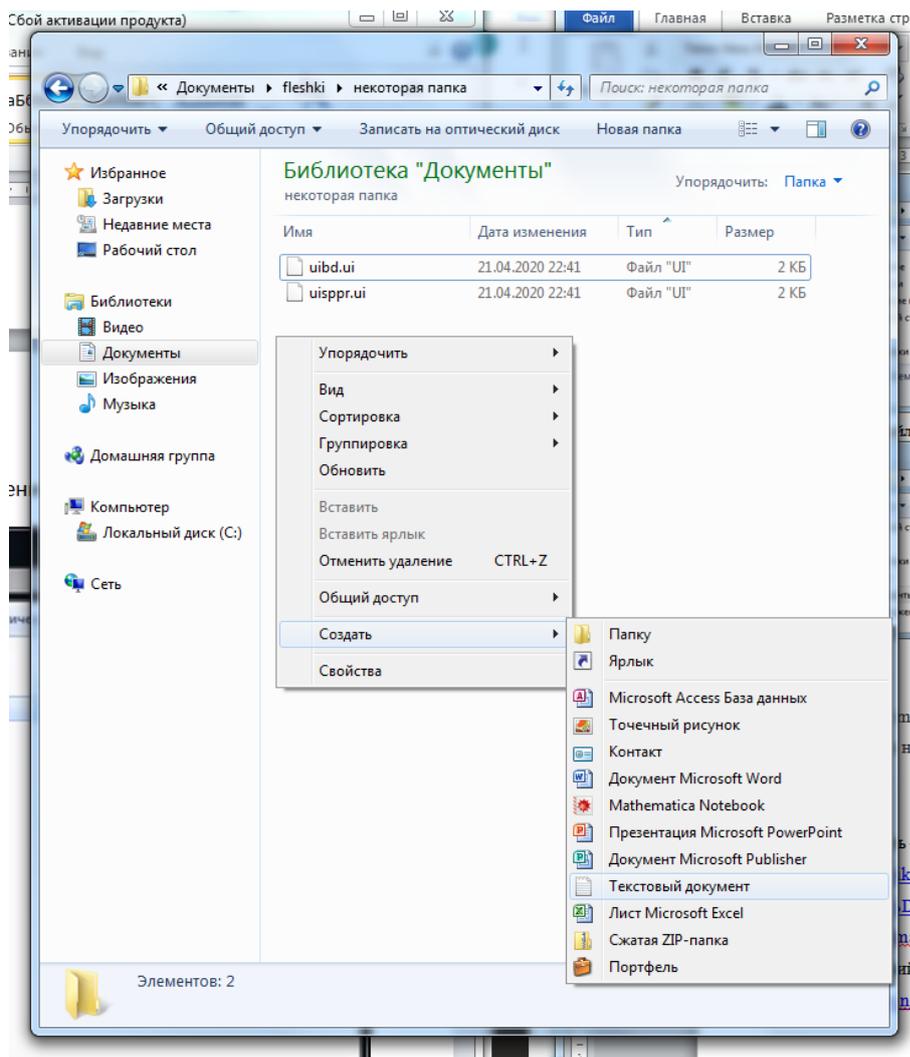
```
form.setupUi(window)
```

```
window.show()
```

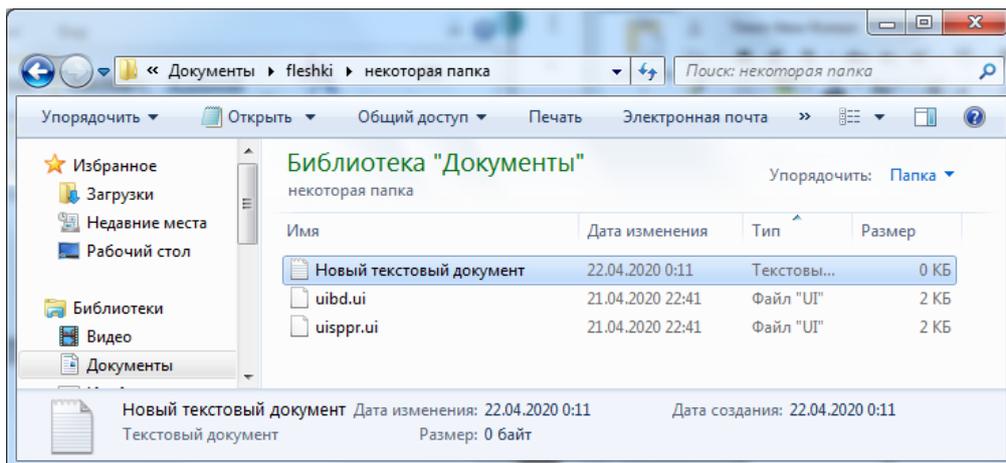
```
app.exec_()
```

Способ 2. Конвертировать .ui-файл в Python-файл.

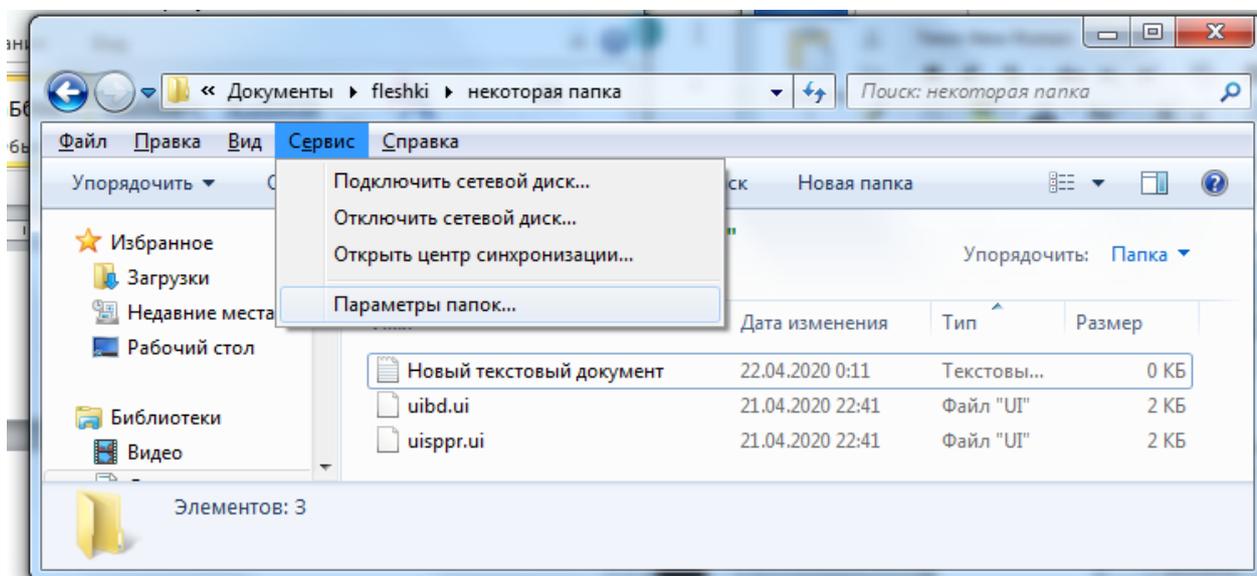
Это можно делать по-разному. Сделайте это так. Сначала создайте **bat**-файл. Для этого сначала создаем текстовый документ, через контекстное меню:



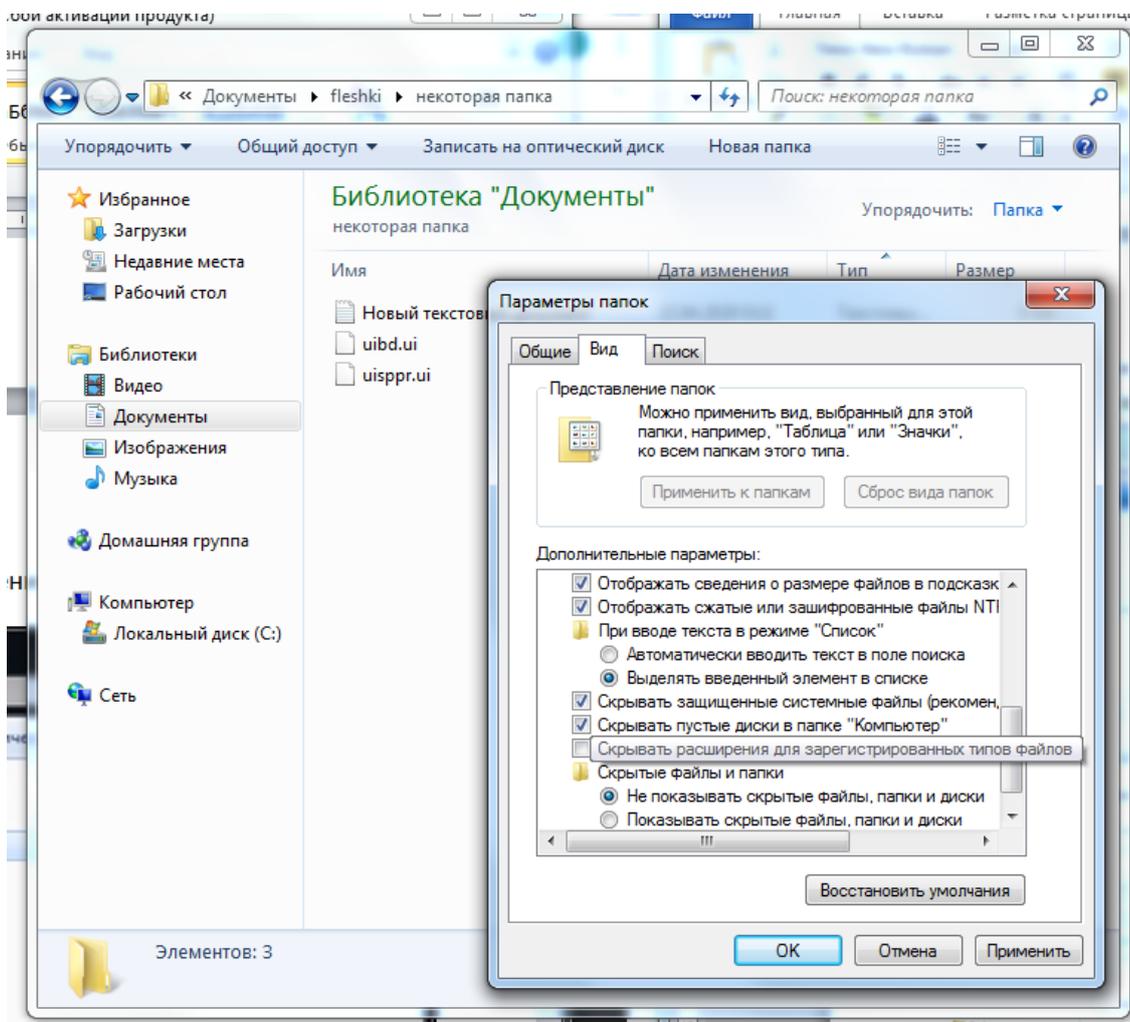
Может так быть, что расширение не отображается:



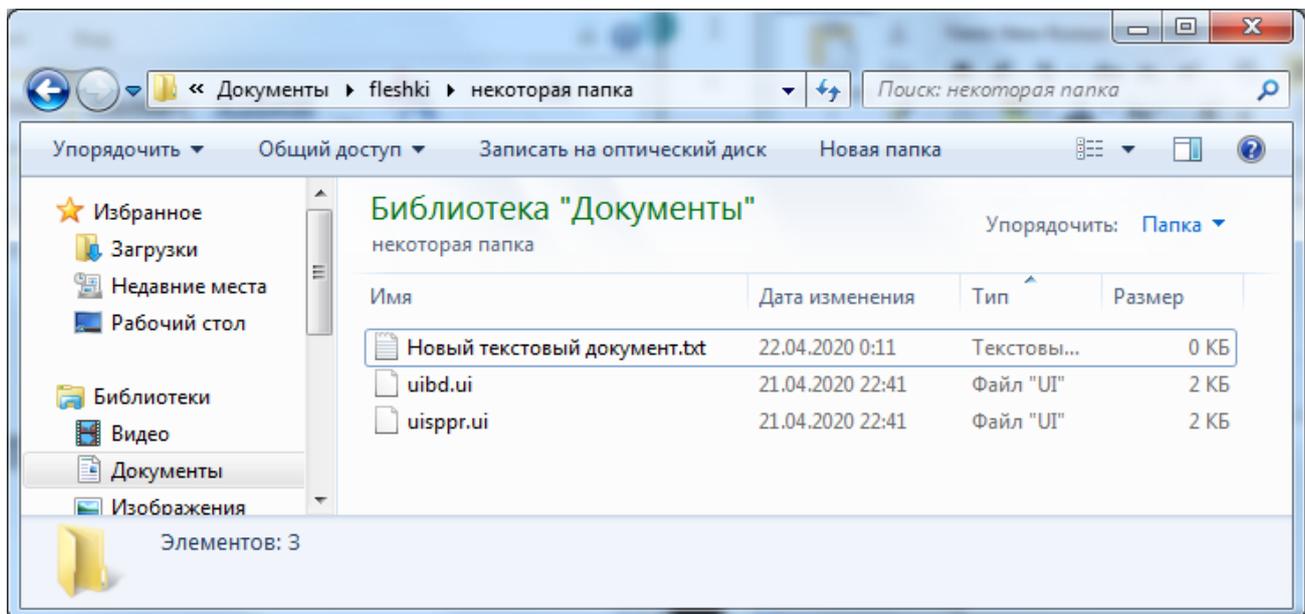
Для того чтобы оно стало видимым надо нажать клавишу **Alt**. Появится наверху меню. Надо выбрать: «Сервис» - «Параметры папок»:



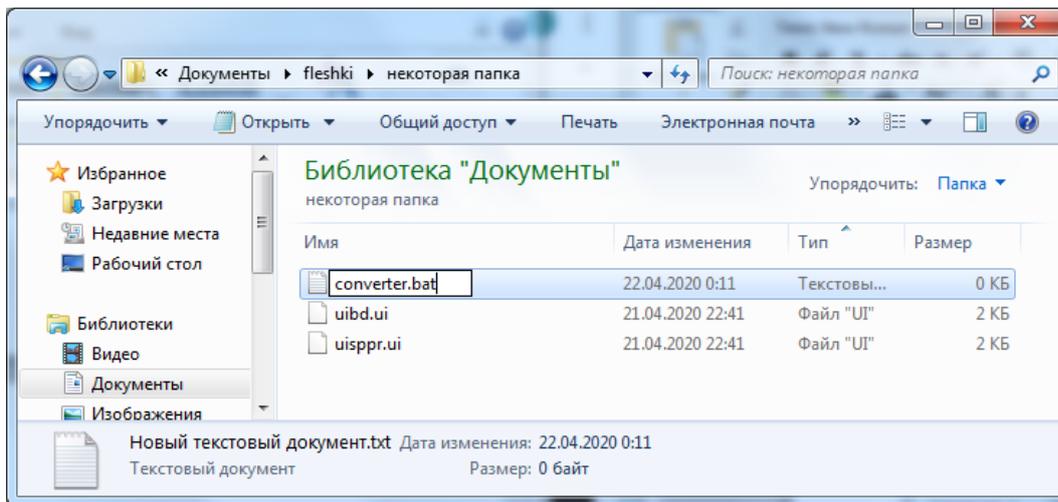
В раскрывшемся окне пойти на вкладку «Вид». Галочки не должно быть:



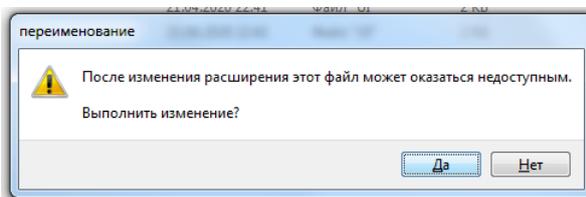
И у вас появится расширение .txt:



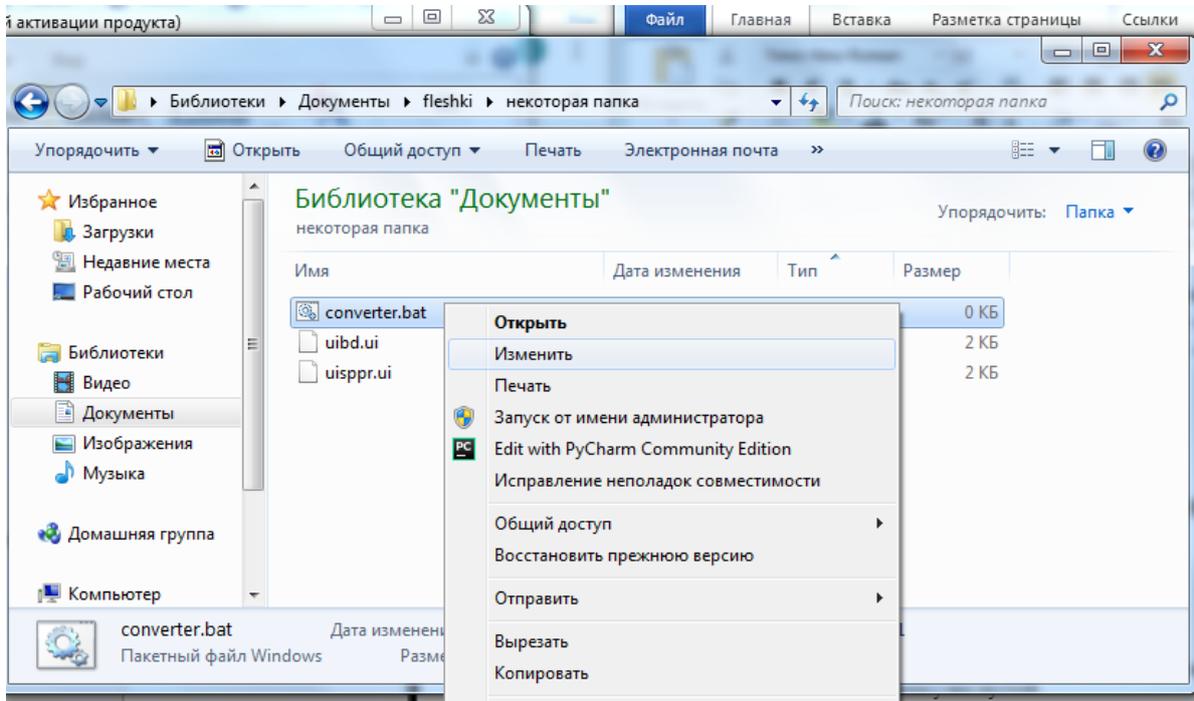
Теперь этот файл переименуем и сами напишем ему расширение .bat:



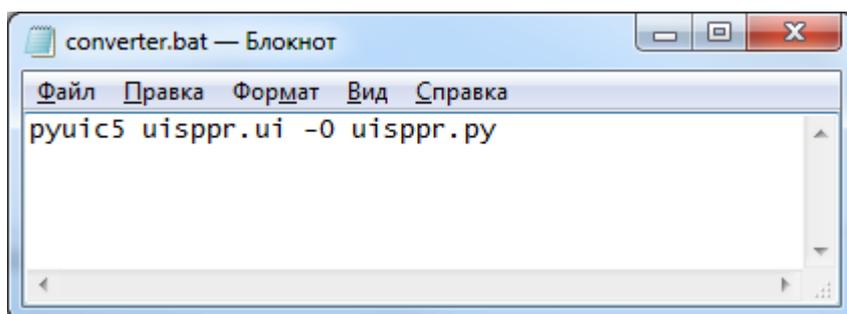
После этого появится сообщение:



Кликаем «Ok». Файл **converter.bat** и будет преобразовывать наш .ui-файл в .py-файл. Файл converter.bat пока у нас пустой. Изменяем его:



В открывшемся окне пишем команду:



Сохраняем измененный файл и закрываем его.

Запускаем .bat-файл, два раза щелкнув на левую клавишу мыши. Ждем некоторое время и видим, что конвертация ui-файла в .py-файл **uisppr.py** произошла. Открыв и запустив его на выполнение, получим форму с нашим интерфейсом. Аналогичную процедуру сделайте и для файла **uibd.ui**.

ЛИТЕРАТУРА К РАБОТЕ

1. Руководство по Qt Designer. – URL: <http://doc.crossplatform.ru/qt/4.5.0/designer-manual.html> (дата обращения: 10.05.2023).
2. Qt Documentation. – URL: <https://doc.qt.io/qt-5/qt designer-manual.html> (дата обращения: 10.05.2023).
3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренко, В. А. Дронов. - СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).

РАБОТА 8

СППР «ОПРЕДЕЛЕНИЕ КАЧЕСТВА ПОВЕРХНОСТИ ПОРИСТОГО МАТЕРИАЛА». Часть 2

Цель работы: создание базы данных для СППР «Определение качества поверхности пористого материала».

Подготовка к выполнению задания

1. На компьютере должен быть выход в интернет.
2. На компьютере должна быть возможность работы в SQL.
3. Скачайте и установите а проект внешнюю библиотеку `sqlite3`.

Задание. Создание БД СППР

1. Создайте проект `Project_porosity`.
2. Создайте в этом проекте Python-файл с именем `DSS_porosity`.
3. Поместите в текущую папку, т.е. в папку проекта `Project_porosity`, набор изображений (цветных и в оттенках серого) различных пористых материалов, приготовленный ранее.
4. Установите в проект `Project_porosity` следующие внешние библиотеки: NumPy, SQLite, **PIL (Pillow-PIL)**, OpenCV (cv2)..
5. В Python-файле `DSS_porosity` создайте базу данных СППР, в которой будут записаны нормативные значения характеристик (пористость, допустимая площадь пор, а также допустимые для них средние квадратические отклонения), четырех пористых материалов. Эта база данных должна иметь следующую таблицу:

	ID	Наименование	Площадь поры	Откл. от площади	Пористость	Откл. от пористости
1	0	Материал2	12.0	5.0	0.1	0.01
2	1	Материал3	9.0	8.0	0.15	0.01
3	4	материал1	6.0	4.0	0.2	0.02

Для этого наберите и запустите на выполнение скрипт создания базы данных, например, из четырех материалов:

```
import numpy as np
import cv2
import sqlite3
import PIL.Image as Img
import PIL.ImageEnhance as Enhance

rows = [
    (0, 'Материал2', 12.0, 5.0, 0.1, 0.01),
    (1, 'Материал3', 9.00, 8.0, 0.15, 0.01),
    (2, 'Материал4', 15.0, 8.0, 0.2, 0.5),
    (3, 'Материал5', 14.0, 7.0, 0.3, 0.7),
]

conn = sqlite3.connect(self.db_name)
cur = conn.cursor()
cur.execute("""CREATE TABLE Materials
              (ID      INTEGER NOT NULL PRIMARY KEY
              AUTOINCREMENT, NAME TEXT,
              PORE_AREA_MEAN REAL NOT NULL,
              PORE_AREA_STD REAL NOT NULL,
              POROUS_MEAN REAL NOT NULL,
              POROUS_STD REAL NOT NULL
              )""")

cur.executemany("""INSERT INTO Materials values
(?,?,?, ?, ?, ?)""", rows)
conn.commit()
conn.close()
```

ЛИТЕРАТУРА К РАБОТЕ

1. Python Imaging Library (Fork) . – URL: <https://pypi.org/project/Pillow/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobxodimoe_proxorenok/ (дата обращения: 10.05.2023).
3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov->

РАБОТА 9

СППР «ОПРЕДЕЛЕНИЕ КАЧЕСТВА ПОВЕРХНОСТИ ПОРИСТОГО МАТЕРИАЛА». Часть 3

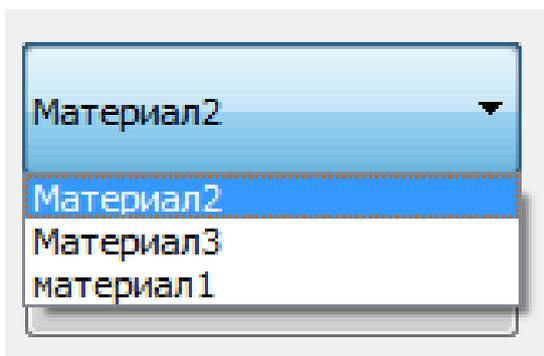
Цель работы: создание системы управления базой данных для СППР «Определение качества поверхности пористого материала».

Подготовка к выполнению задания.

1. На компьютере должен быть выход в интернет.
2. Откройте созданный на практическом занятии 9 проект **Project_porosity**.
3. Ознакомьтесь с приложением 1.

Задание 1. Создание системы управления БД СППР

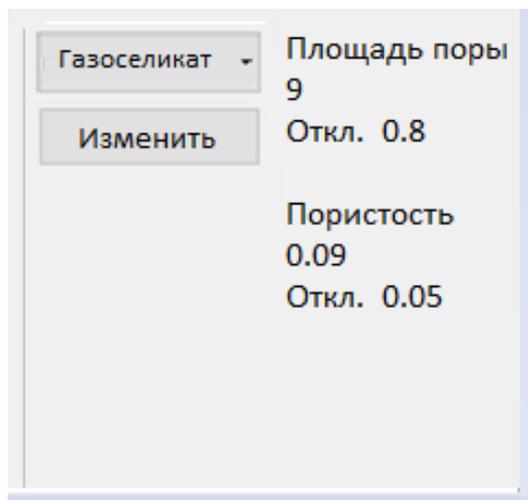
1. Откройте в проекте **Project_porosity** файл **DSS_porosity**, созданный на практическом занятии 9.
2. Напишите код для формирования списка и выбора из него нужного для анализа материала:



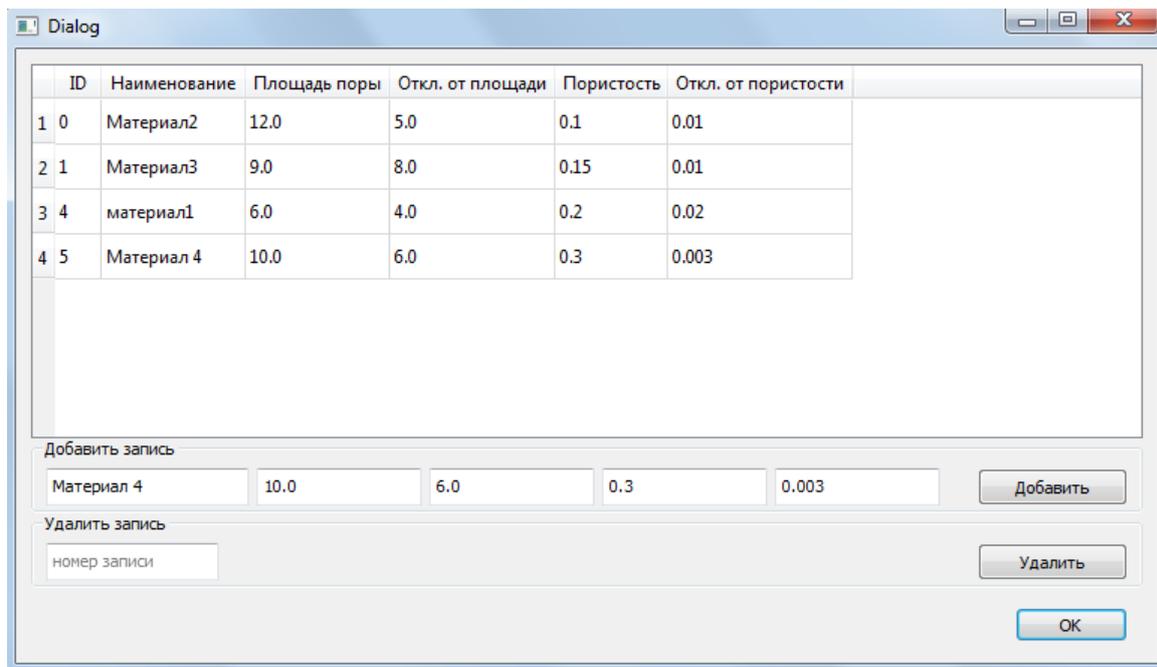
В данном виджете выбирается нужный нам материал из базы данных, уже заранее внесенный в систему со своим названием, порядковым номером и

параметрами, для которого мы будем ставить эксперименты из окна СППР (см. практическое занятие 8). Она создается с помощью компонента QComboBox.

Напишите код такой, чтобы после выбора материала из этого списка считывались соответствующие нормативные значения и отображались справа от этой кнопки (см. практическое занятие 8):



3. Напишите код для открытия и функционирования всех виджетов формы для изменения БД, которая должна открываться при нажатии на кнопку «Изменить» из окна СППР (см. практическое занятие 8). Лицу, принимающему решение, через поля ввода на ней должна быть предоставлена возможность изменения указанных выше допустимых значений средних квадратических отклонений. Так же должна быть возможность добавления образца материала, если его нет в базе данных СППР, а так же удаления записи из БД пористых материалов. Вид этой формы должен быть следующий:



Ниже представлен код обработки нажатия на кнопку добавления .

```

def push_button_add_click(self):
    material_name =
self.text_edit_material_name.toPlainText()
    material_area =
self.text_edit_material_area.toPlainText()
    material_area_std =
self.text_edit_material_area_std.toPlainText()
    material_porous =
self.text_edit_material_porous.toPlainText()
    material_porous_std =
self.text_edit_material_porous_std.toPlainText()

    data = [material_name, material_area,
material_area_std, material_porous, material_porous_std]

    flag = [True if (m is not None and m != '') else
False for m in data ]

    if flag:
        try:
            material_area = float(material_area)
            material_area_std = float(material_area_std)
            material_porous = float(material_porous)
            material_porous_std =
float(material_porous_std)
            connect = sqlite3.connect(self.db_name)
            crsr = connect.cursor()
            crsr.execute("""INSERT INTO Materials(NAME,
PORE_AREA_MEAN, PORE_AREA_STD, POROUS_MEAN, POROUS_STD)
VALUES (?, ?, ?, ?, ?)""", (material_name,

```

```

material_area, material_area_std,
material_porous_std))
        connect.commit()
        connect.close()
        self.load_materials()
        self.fill_table()

    except Exception as e:
        print(e)

```

В данной форме представлена так же команда удаления записи из базы данных. Для удаления записи из БД ЛПР необходимо ввести в ячейку номер материала и нажать на кнопку удаления. После чего произойдет удаление строки из таблицы базы данных. Ниже представлен код обработки нажатия на кнопку «Удалить».

```

def push_button_delete_click(self):
    index = self.text_edit_material_id.toPlainText()
    try:
        index = int(index) - 1
        if 0 <= index <= len(self.materials)-1:
            connect = sqlite3.connect(self.db_name)
            crsr = connect.cursor()
            row = self.materials.pop(index)
            id = row[0]
            crsr.execute('DELETE FROM Materials WHERE ID=?',
(id,))

            connect.commit()
            connect.close()
            self.load_materials()
            self.fill_table()
    except Exception as e:
        print(e)

    self.text_edit_material_id.setPlainText('')
    self.text_edit_material_id.clear()

```

ЛИТЕРАТУРА К РАБОТЕ

1. Python Imaging Library (Fork) . – URL: <https://pypi.org/project/Pillow/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-

3631-8.-Текст:электронный).–URL:

https://codernet.ru/books/python/python_3_samoe_neobходимое_proxorenok/

(дата обращения: 10.05.2023).

3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).

РАБОТА 10

СППР «ОПРЕДЕЛЕНИЕ КАЧЕСТВА ПОВЕРХНОСТИ ПОРИСТОГО МАТЕРИАЛА». Часть 4

Цель работы: создание формы для улучшения качества изображения, предназначенного для анализа в СППР «Определение качества поверхности пористого материала».

Подготовка к выполнению заданий.

1. Откройте созданный на практическом занятии 9 проект **Project_porosity**.
4. Поместите в текущую папку набор изображений (цветных и в оттенках серого) различных пористых материалов.
5. Ознакомьтесь с приложением 1

Задание 1. Создание формы для улучшения качества изображения

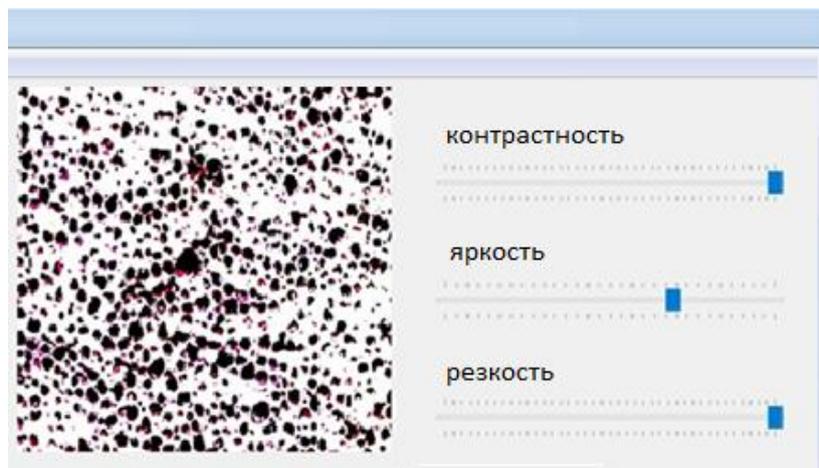
1. Откройте в проекте **Project_porosity** файл **DSS_porosity**.
2. Опишите метод, обрабатывающий изменение значений контрастности, яркости, резкости:

```

def set_transformed_frame(self, image):
    image = Img.fromarray(image)
    image =
Enhance.Contrast(image).enhance(self.contrast_slider.value()/10)
    image =
Enhance.Brightness(image).enhance(self.brightness_slider.value()/10)
    image =
Enhance.Sharpness(image).enhance(self.sharpness_slider.value()/10)
    image = np.array(image)
    self.transform_img = image
    image = gui.QImage(image.data, image.shape[1], image.shape[0],
gui.QImage.Format_RGB888)
    self.transformed_frame.setPixmap(gui.QPixmap.fromImage(image))

```

2. Реализуйте улучшение изображения для анализа, которое должно размещаться в окне слева от движков управления контрастностью, яркостью, резкостью изображения (см. практическое занятие 8):



Для управления контрастностью, яркостью, резкостью исходного изображения используются такие компоненты библиотеки PyQt5, как QSlider. Задайте диапазоны возможных значений контрастности, яркости и резкости по примеру кода, приведенного ниже:

```

self.contrast_slider = wdgts.QSlider(core.Qt.Horizontal)
self.contrast_slider.setRange(-200, 200)
self.contrast_slider.setTickPosition(wdgts.QSlider.TicksBothSides)
self.contrast_slider.setValue(10)

```

ЛИТЕРАТУРА К РАБОТЕ

1. Python Imaging Library (Fork) . – URL: <https://pypi.org/project/Pillow/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobходимое_proxorenok/ (дата обращения: 10.05.2023).
3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).

РАБОТА 11

СППР «ОПРЕДЕЛЕНИЕ КАЧЕСТВА ПОВЕРХНОСТИ ПОРИСТОГО МАТЕРИАЛА». Часть 5

Цель работы: Реализация алгоритма определения пор и вывода результата анализа для СППР «Определение качества поверхности пористого материала».

Подготовка к выполнению заданий.

1. Откройте созданный на практическом занятии 9 проект **Project_porosity**.
2. Ознакомьтесь с приложением 1.

Задание 1. Реализация алгоритма определения пор и вывода результата

1. Откройте в проекте `Project_porosity` файл `DSS_porosity`.
2. Теперь рассмотрим основную часть анализа изображения. Ниже представлен программный код конечной обработки изображения, обнаружения контуров пор, вычисления пористости и определения аномальных пор.

```
def explore(self, image):
    """
    Входной параметр:
    image - исследуемое изображение
    Выход:
    image - изображение с контурами пор
    area_c - отношение площади всех пор ко всей площади
    изображения (пористость)
    len(bad_conrours) - количество 'плохих' пор
    """
    image = np.copy(image)
    # дополнительная обработка шумов
    blurred = cv2.GaussianBlur(image, (5, 5), 0)
    # конвертация BGR формата в формат HSV
    hsv = cv2.cvtColor(blurred, cv2.COLOR_BGR2HSV)

    lower_black = np.array([0, 0, 0])
    upper_black = np.array([120, 120, 120])
    # определяем маску для обнаружения контуров пор.
    # будут выделены поры в заданном диапазоне
    mask = cv2.inRange(hsv, lower_black, upper_black)
    # получаем массив конутров
    _, contours, _ = cv2.findContours(mask,
cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)

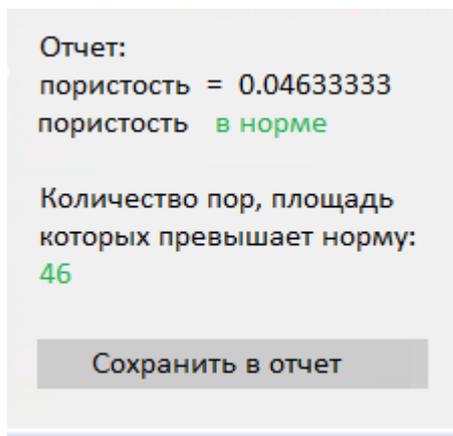
    good_contours = []
    bad_contours = []
    area_c = 0
    # находим поры, не превышающие нормативную площадь
    for contour in contours:
        # также подсчитываем общую площадь пор
        area_c += cv2.contourArea(contour)
        if self.mat_area - self.mat_area_std <=
cv2.contourArea(contour) <= self.mat_area +
self.mat_area_std:
            good_contours.append(contour)
        else:
            bad_contours.append(contour)
```

```

area_c = area_c / (image.shape[0] * image.shape[1])
# выделяем 'хорошие' поры зеленым цветом
cv2.drawContours(image, good_contours, -1, (0, 255,
0), 3)
# выделяем 'плохие' поры красным цветом
cv2.drawContours(image, bad_contours, -1, (255, 0,
0), 3)
return image, area_c, len(bad_contours)

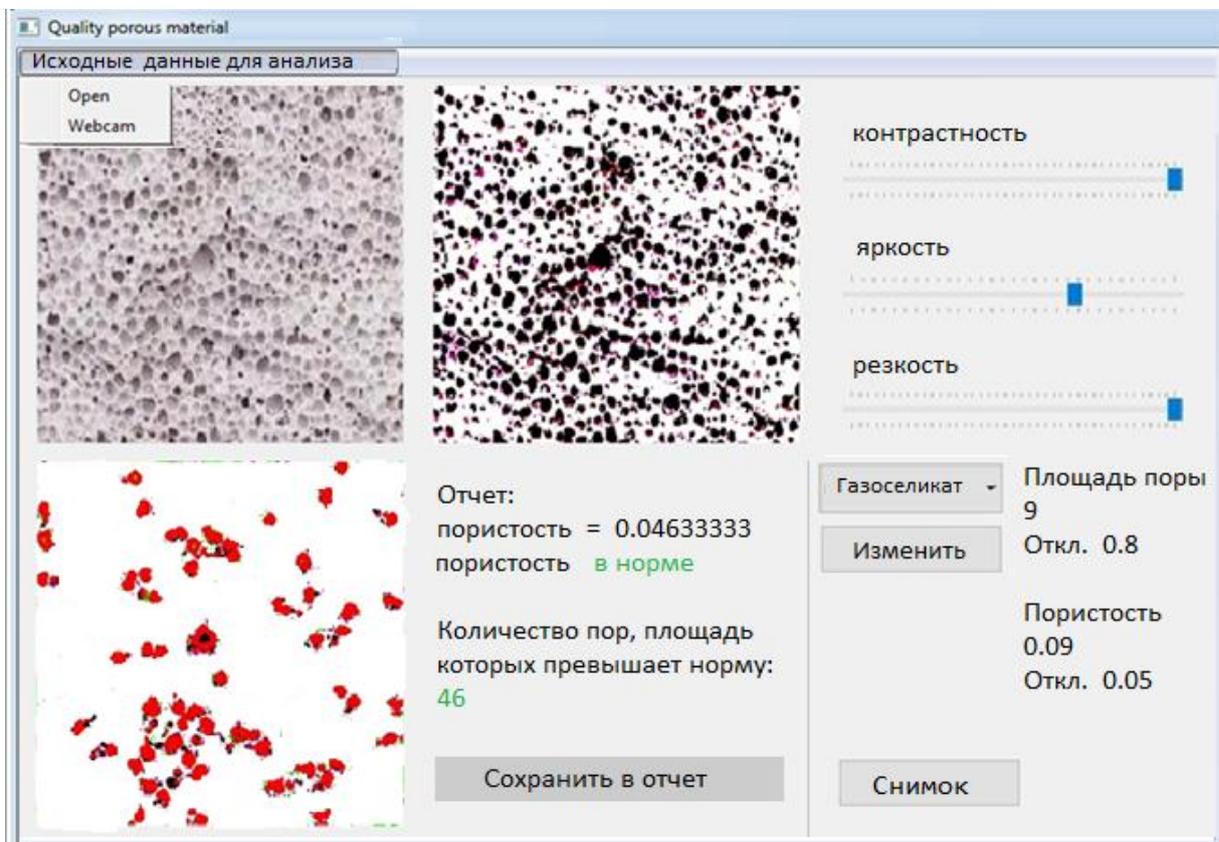
```

Отчет по результатам анализа выводится в форму, представленную на рисунке:



Задание 2. Тестирование работы созданной СППР

Откройте проект «Определение качества поверхности пористого материала»:



Продемонстрируйте работу созданной СППР на разных пористых материалах. Следите за правильностью выполнения системой каждого из задаваемых пользователем запросов, а также следите за динамичностью и интерактивностью работы созданной Вами СППР. Сделайте вывод о качестве работы алгоритма, заложенного в базу знаний созданной СППР.

ЛИТЕРАТУРА К РАБОТЕ

1. Пористость. Определение. – URL:
<https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%80%D0%B8%D1%81%D1%82%D0%BE%D1%81%D1%82%D1%8C> (дата обращения: 10.05.2023).
2. NEXSYS ImageExpert™ Sample 2. Программа для качественного анализа изображений структур методом сравнения с эталонными шкалами. – URL:
http://www.nexsys.ru/nexsys_iesam2x.htm (дата обращения: 10.05.2023).
3. Комплекс видеоизмерительный для анализа микроструктур и макроструктур материалов «VESTRA Imaging System». – URL:
<http://latemi.ru/catalog/vestra-imaging-system/> (дата обращения: 10.05.2023).
4. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. - СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).

РАБОТА 12

РАСПОЗНАВАНИЕ ОБЪЕКТОВ НА КАРТИНКЕ

Цель работы: написать программу, распознающую объекты на картинке.

Подготовка к выполнению задания

1. На компьютере должен быть выход в интернет.
2. Установите ImageAI и необходимые для ImageAI внешние библиотеки (см. приложение 2).
3. Зайдите на официальный сайт GitHub, скачайте модель RetinaNet и поместите эту модель в ту же папку, где находится `main.py`.

Ссылки для скачивания:

- **ImageAI** документация: <https://imageai.readthedocs.io/en/latest/>.
- **ImageAI GitHub**
<https://github.com/OlafenwaMoses/ImageAI/tree/master/imageai/Detection>,

Задание. Распознавание объектов с картинки

1. Подключите библиотеку `imageai.Detection`, т.е. наберите:

```
1 from imageai.Detection import ObjectDetection
```

Класс `ObjectDetection` позволит обнаруживать различные объекты на картинках.

2. Подключите библиотеку `OS` -это встроенная библиотека в `Python`. Она позволяет работать с операционной системой.

```
2 import os
```

3. Создайте новую переменную `exec_path`, в нее поместим `getcwd()` -эта функция позволяет определить путь к этому проекту- нам этот путь понадобится для того чтобы мы могли обнаружить различные объекты(

например тот же самый объект **RetinaNet** , который нам и потребуется для того чтобы мы могли обнаруживать различные объекты)

```
3
4 exec_path = os.getcwd()
```

4. Далее создайте на основе **ObjectDetection** несколько свойств(объектов).

Первым делом создаем переменную **detector** и в нее поместим класс **ObjectDetection()** , т.е. создаем объект на основе класса **ObjectDetection()**

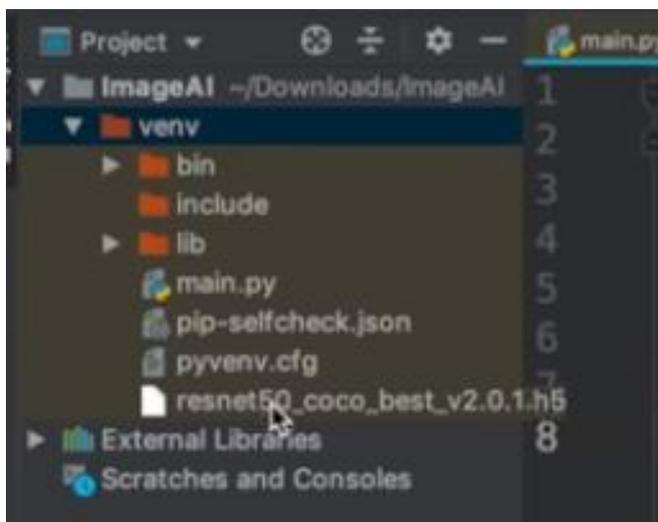
```
5
6 detector = ObjectDetection()
```

5. Далее берем **detector** и через точку обращаемся к различным функциям, которые находятся в классе **ObjectDetection** .

Первая функция это функция **setModelTypeAsRetiaNet** . Данная функция позволяет установить то, что мы используем RetiaNet модель для обнаружения объектов.

```
7 detector.setModelTypeAsRetinaNet()
```

Далее устанавливаем путь к нашей модели с именем **resnet50_coco_best_v2.0.1.h5** (см на рисунок ниже на этом рисунке стрелочкой указано это имя):



Для этого используем полный путь, т.е. будем писать в коде `exec_path` (см. ниже), т.е. говорим, где вообще находится данный файл (в этой переменной находится полный путь к данному проекту), а дальше говорим, какой файл мне необходимо открыть, т.е. как бы открываем. Это будет наша модель:

```
8 detector.setModelPath(os.path.join(
9     exec_path, "resnet50_coco_best_v2.0.1.h5")
10 )
```

6. Затем обращаемся опять к `detector` подгружаем данную модель (поэтому используем функцию `loadModel()`)

```
11 detector.loadModel()
```

7. Дальше находим различные совпадения на какой либо картинке. Для этого создадим `list`. Это будет наш список, в который будут помещаться различные обнаружения. В него мы запишем следующее: мы берем `detector`, берем функцию `detectObjectsFormmage` (т.е. обнаружить объекты из изображения).

Внутри указываем `input_image`, т.е. исходное изображение, с которого мы будем обнаруживать все объекты. Чтобы указать расположение этого изображения используем модуль `OS`, дальше используем переменную `path`. Дальше используем функцию `join`, т.е. объединяем несколько строк между собой: `elec_path` - это полный путь к данному проекту. Указываем название картинки, которую хотим открыть `objects.jpg`.

Путь `output_image_path` - это то место, куда мы сохраним полученную фотографию. Пишем знак равно: `=`, и копируем из предыдущей строчки, меняя только название файла `nev_Objects.jpg`, в который сохраняем.

```

12
13 list = detector.detectObjectsFromImage(
14     input_image=os.path.join(exec_path, "objects.jpg" ),
15     output_image_path=os.path.join(exec_path, "new_objects.jpg")
16 )

```

Запустим на решение. Немного ждем:

```

1 from imageai.Detection import ObjectDetection
2 import os
3
4 exec_path = os.getcwd()
5
6 detector = ObjectDetection()
7 detector.setModelTypeAsRetinaNet()
8 detector.setModelPath(os.path.join(
9     exec_path, "resnet50_coco_best_v2.0.1.h5")
10 )
11 detector.loadModel()
12
13 list = detector.detectObjectsFromImage(
14     input_image=os.path.join(exec_path, "objects.jpg"),
15     output_image_path=os.path.join(exec_path, "new_objects.jpg")
16 )

```

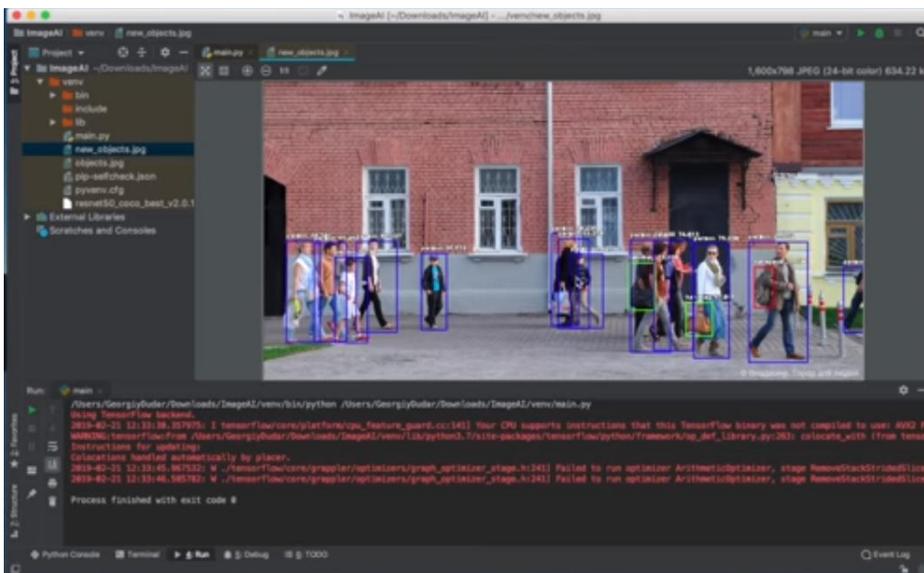
Run: main

```

/Users/GeorgiyDudler/Downloads/ImageAI/venv/bin/python /Users/GeorgiyDudler/Downloads/ImageAI/venv/main.py
Using TensorFlow backend.
2019-02-21 12:33:38.357975: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 PM
WARNING:tensorflow:From /Users/GeorgiyDudler/Downloads/ImageAI/venv/lib/python3.7/site-packages/tensorflow/python/framework/op_def_library.py:263: colocate_with [from tensor
instructions handled automatically by placer.
2019-02-21 12:33:45.967532: W tensorflow/core/grappler/optimizers/graph_optimizer_stage.h:241] Failed to run optimizer ArithmeticOptimizer, stage RemoveStackStridedSlice5
2019-02-21 12:33:46.585782: W tensorflow/core/grappler/optimizers/graph_optimizer_stage.h:241] Failed to run optimizer ArithmeticOptimizer, stage RemoveStackStridedSlice5
Process finished with exit code 0

```

Процесс закончился. Видим, что у нас создался новый файл. Если откроем его, то увидим на изображении выделенные программой объекты:



Все те объекты, которые были найдены, обведены в рамку и прописано (в процентах) на сколько они подходят под определенный объект.

Можем еще дописать дополнительные характеристики:

```
13 list = detector.detectObjectsFromImage(  
14     input_image=os.path.join(exec_path, "objects.jpg" ),  
15     output_image_path=os.path.join(exec_path, "nev_objects.jpg"),  
16     minimum_percentage_probability=30,  
17     display_percentage_probablity=False,  
18     display_object_name=False  
19 )
```

16-я строка - это минимальный процент точности. Т.е. то, что ниже числа, указанного после знака равенства, не будет выделяться в рамку (true-истина т.е. будет указываться, false-не будет указываться).

Запустим дополненную программу на решение. Немного ждем.

Когда процесс обработки закончится увидим, что у нас создан новый файл.

Если откроем его, то увидим:



ЛИТЕРАТУРА К РАБОТЕ

1. Official English Documentation for ImageAI! — ImageAI 2.1.5. – URL: <https://imageai.readthedocs.io/en/latest/> (дата обращения: 10.05.2023).
2. ImageAI GitHub. – URL: <https://github.com/OlafenwaMoses/ImageAI/tree/master/imageai/Detection> (дата обращения: 10.05.2023).

РАБОТА 13

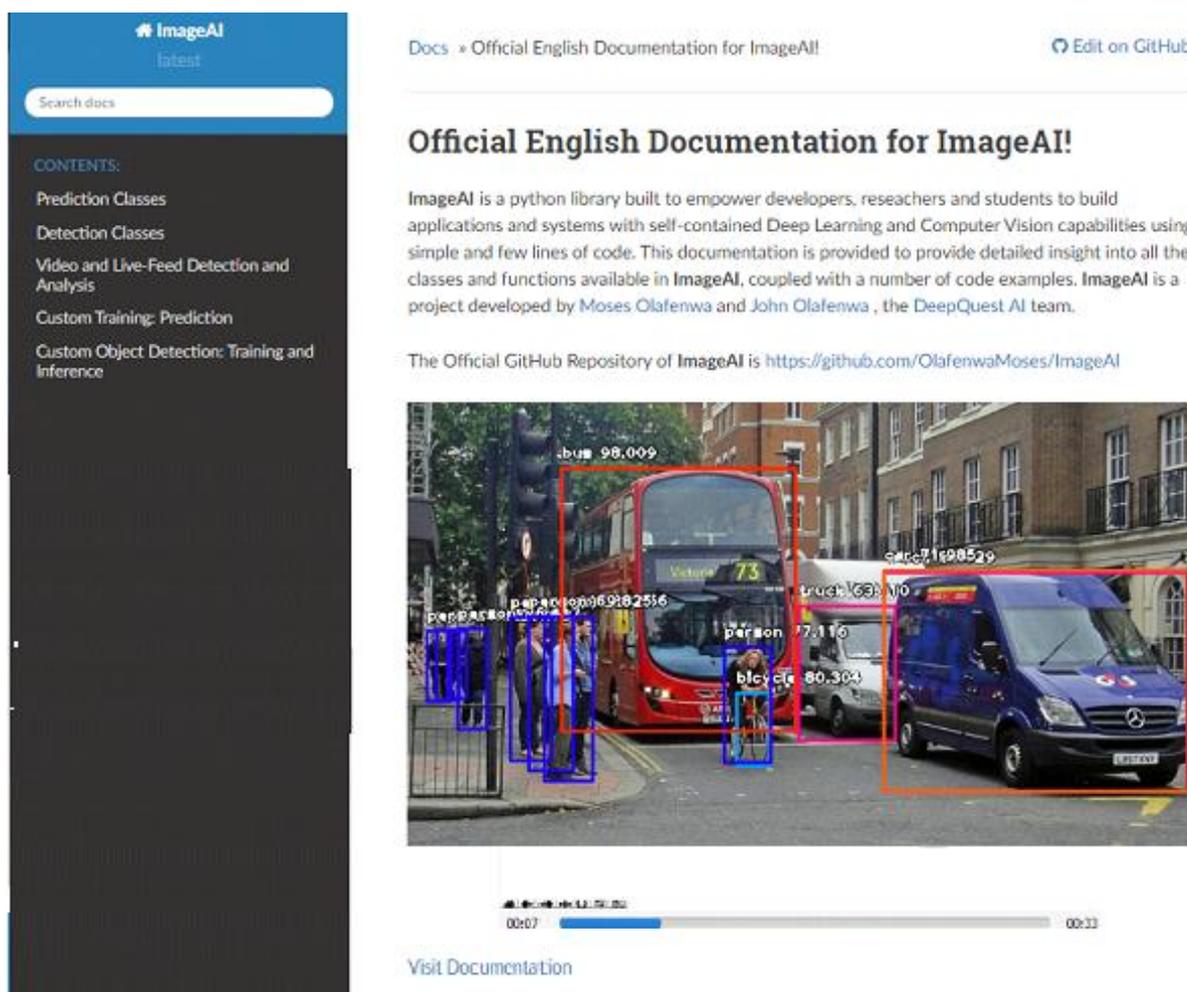
ОТСЛЕЖИВАНИЕ ОБЪЕКТОВ НА ВИДЕО

Суть работы: написать программу для отслеживания объектов на видео.

Подготовка к выполнению задания

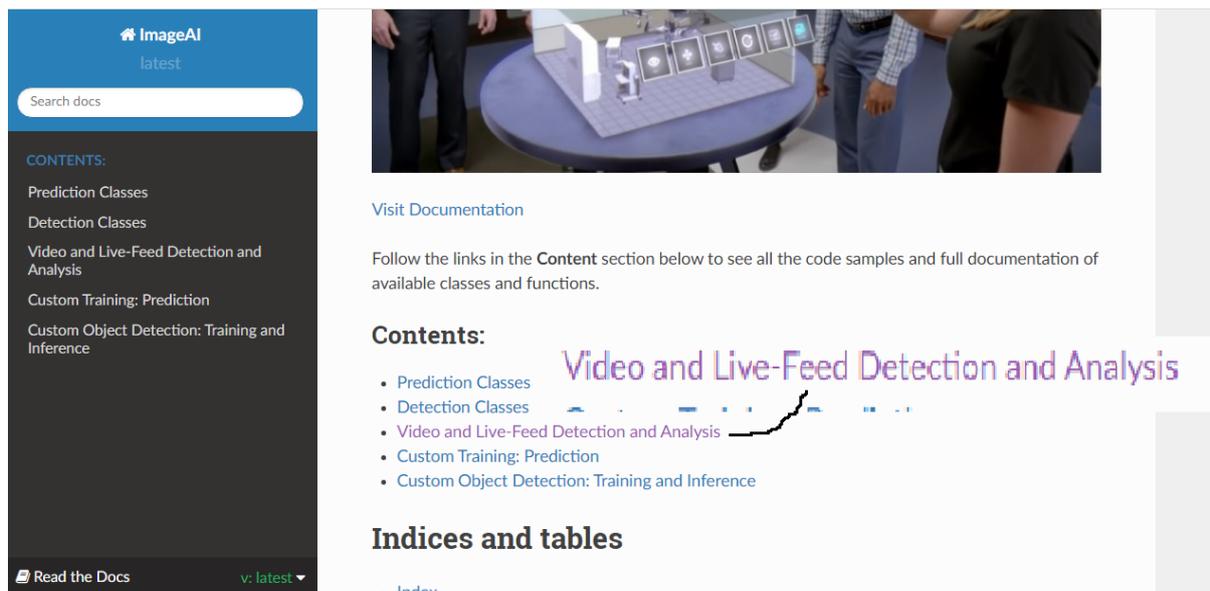
Скачайте на рабочий стол файл **yolo.h5** (см. п.1 ниже) и любой файл с видео небольшой продолжительности (12-15 секунд), дайте ему имя **traffic.mp4** (см. п2 ниже).

п.1 Перейдем на официальный сайт документации по **ImageAI**. Ссылка на официальный сайт: <https://imageai.readthedocs.io/en/latest/>. Выйдем на страницу:

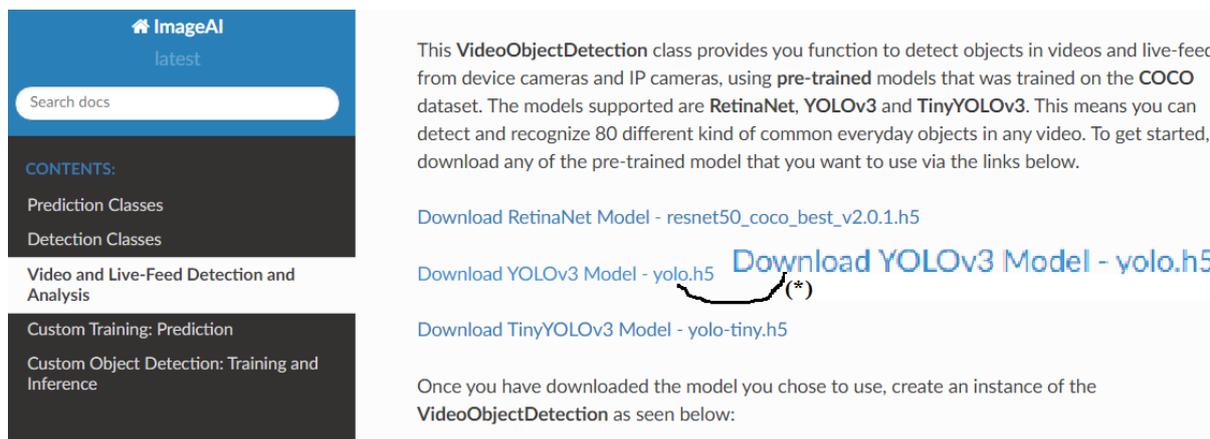


The screenshot shows the official English documentation for ImageAI. The left sidebar contains a search bar and a table of contents with the following items: Prediction Classes, Detection Classes, Video and Live-Feed Detection and Analysis, Custom Training: Prediction, and Custom Object Detection: Training and Inference. The main content area features the title "Official English Documentation for ImageAI!" and a brief description of the library. Below the text, there is a video player showing a street scene with several objects detected and labeled with bounding boxes and confidence scores: a bus (98.009), a person (77.116), a bicycle (80.304), a truck (63.10), and a car (71.98529). A "Visit Documentation" link is located at the bottom of the video player.

Прокрутите эту страницу сайта еще ниже, пока не увидите:



Кликните по выделенной строке (см. предыдущий скрин) получим:



Можно кликнуть все три строчки. Для начала скачайте, **YOLOV3 Model**.

Кликнув на строку, помеченную на скришоте символом (*)), получим:

1.0
Sebcb87
Verified
Compare

Models for Image Recognition and Object Detection

OlafenwaMoses released this on 25 May 2018 · 189 commits to master since this release

This release contains pre-trained models for Image Recognition and Object Recognition tasks in ImageAI

Assets 9

DenseNet-BC-121-32.h5	31.7 MB
inception_v3_weights_tf_dim_ordering_tf_kernels.h5	91.7 MB
resnet50_coco_best_v2.0.1.h5	146 MB
resnet50_weights_tf_dim_ordering_tf_kernels.h5	98.1 MB
squeezenet_weights_tf_dim_ordering_tf_kernels.h5	4.83 MB
yolo-tiny.h5	33.9 MB
yolo.h5 (**)	237 MB
Source code (zip)	
Source code (tar.gz)	

Найдите в этом списке **yolo.h5** (см. (**)), скачайте его на рабочий стол.

п.2 Скачайте любой файл с видео небольшой продолжительности (12-15 секунд) и дайте ему имя **traffic.mp4**.

Закомментируйте код предыдущего практического занятия. Это строки 1-19. Ниже продолжите написание кода для видео. Теперь можно приступить к набору кода для настоящей работы.

Задание. Отслеживание объектов на видео

1. Подключите библиотеку `imageai.Detection`. Класс `VideoObjectDetection` позволит обнаруживать различные объекты на видео.

```
20
21 from imageai.Detection import VideoObjectDetection
```

2. Подключите библиотеку `OS` - это встроенная библиотека в `Python`. Она позволяет работать с операционной системой.

```
22 import os
```

3. Создайте новую переменную `execution_path`, в которой указывается путь к данному проекту:

```
23  
24 execution_path = os.getcwd()
```

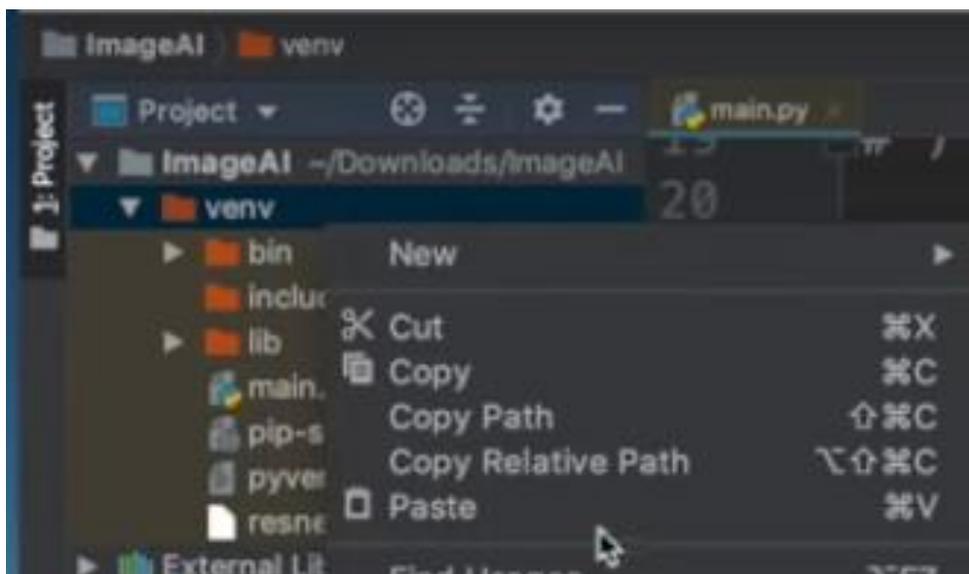
4. Создайте на основе класса `VideoObjectDetection` определенный объект `detector`

```
25  
26 detector = VideoObjectDetection()
```

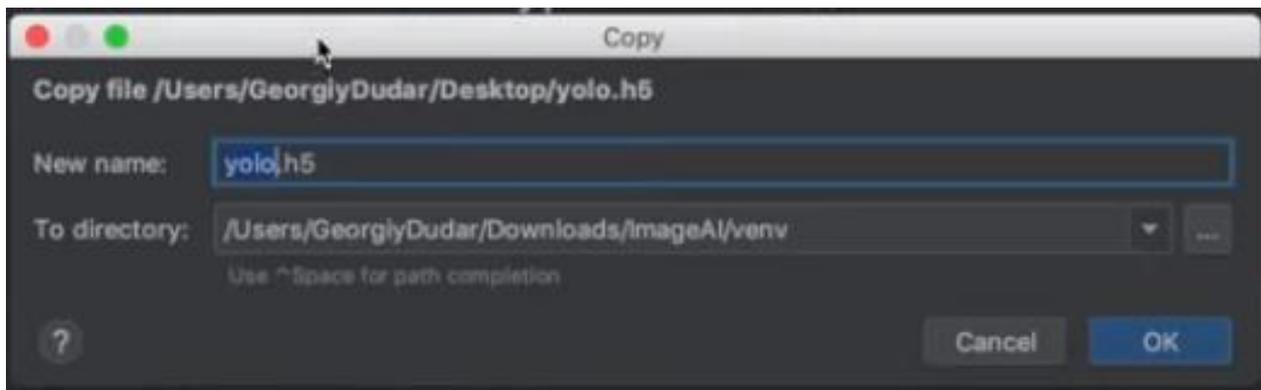
5. Наберите команду

```
27 detector.setModelTypeAsYOLOv3()
```

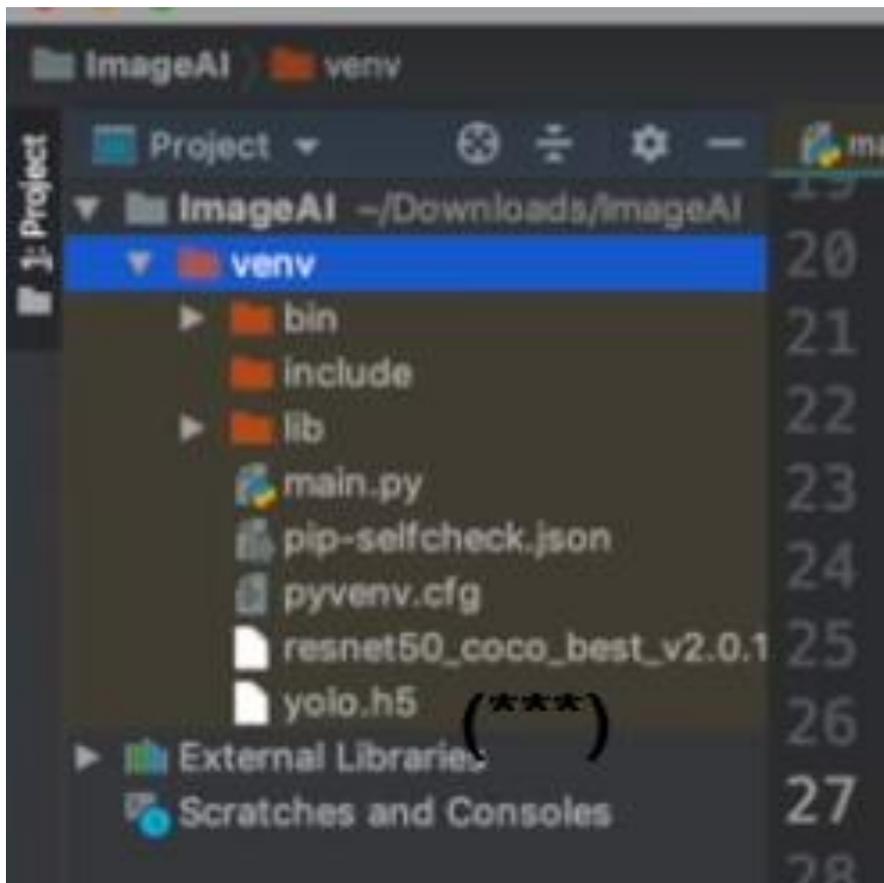
Здесь используем модель `YOLOv3()`, причем это та модель, которая была скачена нами ранее на рабочий стол. Давайте ее сразу скопируйте с рабочего стола и вставьте в сам проект (т.е. идем по цепочке `venv-Paste`):



Вставьте теперь имя этого файла в поле ввода:



Затем нажимаем кнопку **ок**. Увидим, что в текущей папке появился этот файл **yolo.h5** (см. ниже (***))



Как видите, мы эту модель находим в данном проекте, т.е. по пути к данному проекту. Поэтому в коде пишем:

```
28 detector.setModelPath(os.path.join(execution_path, "yolo.h5"))
```

или так

```
28 detector.setModelPath(os.path.join(execution_path, "venv/yolo.h5"))
```

На этом скриншоте в строке 28 после запятой видим " **yolo.h5**" - это сам

этот объект `yolo.h5`, его мы добавили в текущую папку (см. (***) на скриншоте перед последним скриншотом).

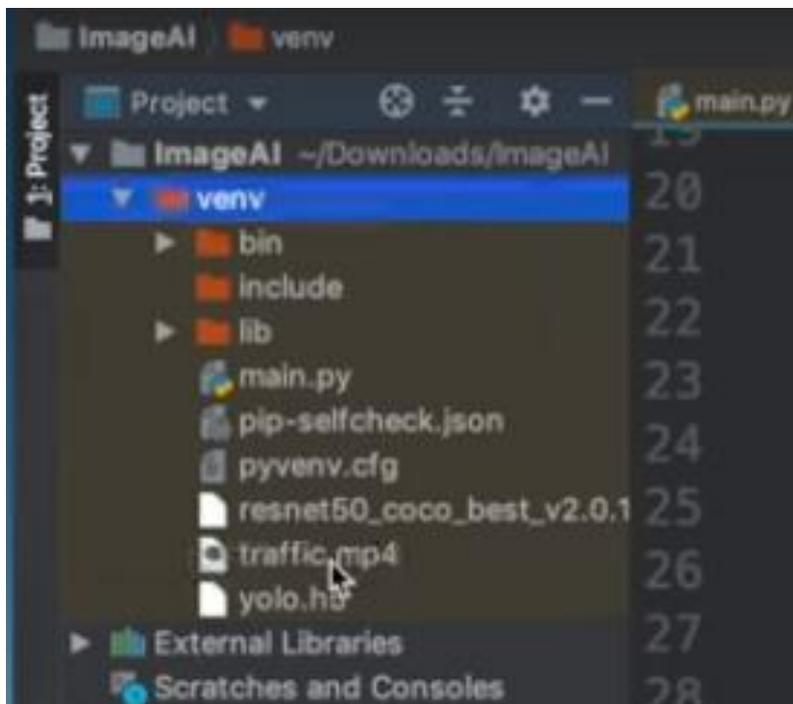
6. Далее подгружаем саму модель

```
29 detector.loadModel()
```

7. Затем создаем список кадры (frames).

```
30
31 video_path = detector.detectObjectsFromVideo(
32     input_file_path=os.path.join(execution_path, "traffic.mp4"),
33     output_file_path=os.path.join(execution_path, "traffic_detected"),
34     frames_per_second=20,
35     log_progress=True
36 )
```

Здесь через функцию `detectObjectsFormVideo` говорим с какого файла мы все берем (см. в коде `input_file_path=`). В данном случае мы берем с файла `traffic.mp4`. Это видео, скаченное нами ранее на рабочий стол. Скопируйте его в текущую папку нашего проекта. Видим, что файл `traffic.mp4` добавился:



Далее говорим, в какой файл будем это все конвертировать

(см. в коде `output_file_path=`). Здесь мы все это конвертируем в файл

traffic_detected.mp4. Названия можем прописывать любые, например назвали здесь **"traffic_detected"**.

Дальше говорим сколько будет кадров (frames) в секунде. Мы устанавливаем 20.

Дальше мы говорим, что `все различные логи, т.е то как программа обрабатывает наше видео, будут выписываться в консоль.

Эта же команда крупнее:

```
video_path = detector.detectObjectsFromVideo(  
    input_file_path=os.path.join(execution_path, "traffic.mp4"),  
    output_file_path=os.path.join(execution_path, "traffic_detected"),  
    frames_per_second=20,  
    log_progress=True  
)
```

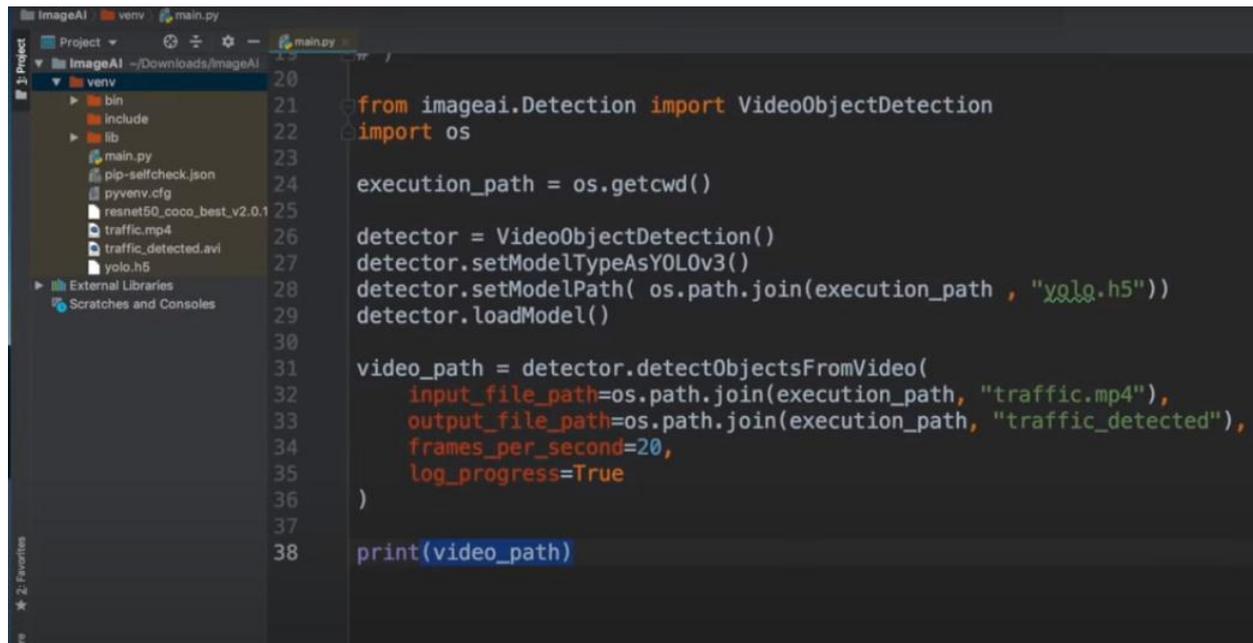
8. Далее выписываем **video_path** в саму консоль

```
37  
38     print(video_path)
```

9. Теперь запустите всю программу на выполнение.

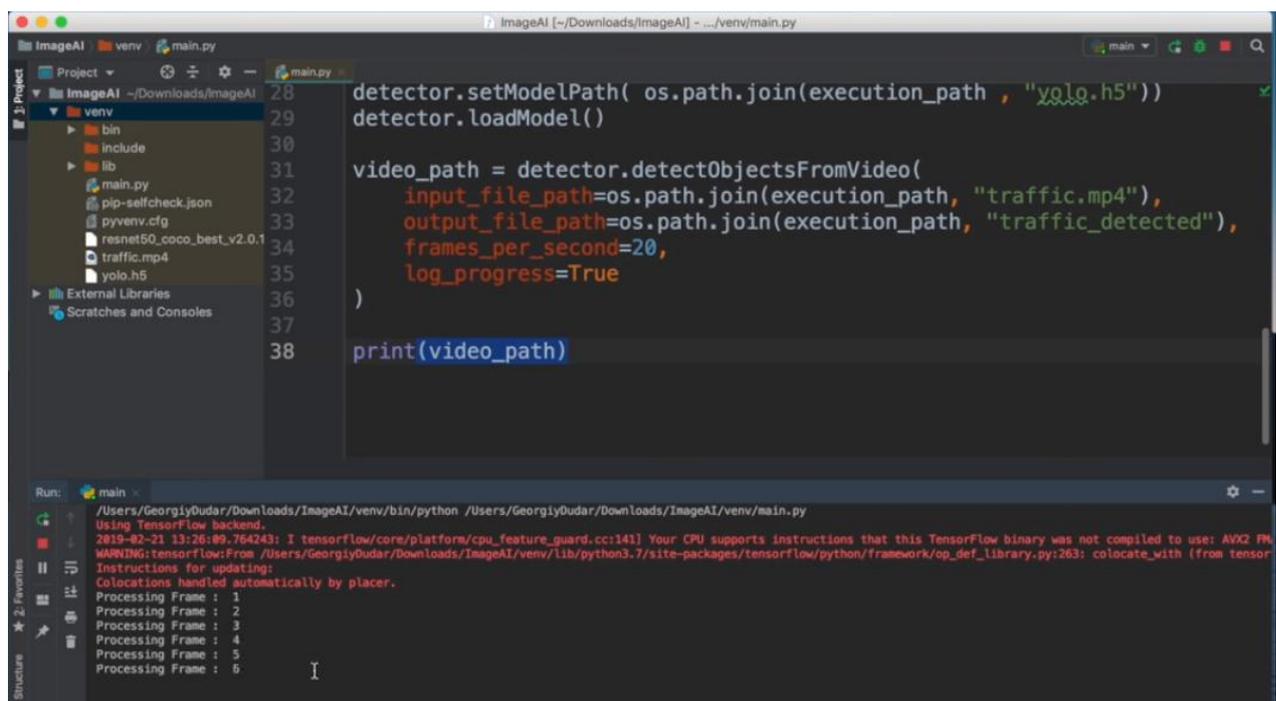
```
21     from imageai.Detection import VideoObjectDetection  
22     import os  
23  
24     execution_path = os.getcwd()  
25  
26     detector = VideoObjectDetection()  
27     detector.setModelTypeAsYOLOv3()  
28     detector.setModelPath(os.path.join(execution_path, "yolo.h5"))  
29     detector.loadModel()  
30  
31     video_path = detector.detectObjectsFromVideo(  
32         input_file_path=os.path.join(execution_path, "traffic.mp4"),  
33         output_file_path=os.path.join(execution_path, "traffic_detected"),  
34         frames_per_second=20,  
35         log_progress=True  
36     )  
37  
38     print(video_path)
```

В итоге проект должен выглядеть следующим образом:



```
20
21 from imageai.Detection import VideoObjectDetection
22 import os
23
24 execution_path = os.getcwd()
25
26 detector = VideoObjectDetection()
27 detector.setModelTypeAsYOLOv3()
28 detector.setModelPath( os.path.join(execution_path , "yolo.h5"))
29 detector.loadModel()
30
31 video_path = detector.detectObjectsFromVideo(
32     input_file_path=os.path.join(execution_path, "traffic.mp4"),
33     output_file_path=os.path.join(execution_path, "traffic_detected"),
34     frames_per_second=20,
35     log_progress=True
36 )
37
38 print(video_path)
```

10. После запуска этой программы видим, что видео обрабатывается. И обрабатывается покадрово:



```
28 detector.setModelPath( os.path.join(execution_path , "yolo.h5"))
29 detector.loadModel()
30
31 video_path = detector.detectObjectsFromVideo(
32     input_file_path=os.path.join(execution_path, "traffic.mp4"),
33     output_file_path=os.path.join(execution_path, "traffic_detected"),
34     frames_per_second=20,
35     log_progress=True
36 )
37
38 print(video_path)
```

```
Run: main
/Users/GeorgiyDudar/Downloads/ImageAI/venv/bin/python /Users/GeorgiyDudar/Downloads/ImageAI/venv/main.py
Using TensorFlow backend.
2019-02-21 13:26:09.764243: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
WARNING:tensorflow:From /Users/GeorgiyDudar/Downloads/ImageAI/venv/lib/python3.7/site-packages/tensorflow/python/framework/op_def_library.py:263: colocate_with (from tensor
Instructions for updating:
Colocations handled automatically by placer.
Processing Frame : 1
Processing Frame : 2
Processing Frame : 3
Processing Frame : 4
Processing Frame : 5
Processing Frame : 6
```

Когда процесс закончится увидим, что у нас создан новый файл. Откроем его. Увидим, что объекты на видео распознаются и берутся в рамку.

Таким образом, при помощи данной библиотеки мы можем распознавать объекты, как на видео, так и просто на фотографиях.

ЛИТЕРАТУРА К РАБОТЕ

1. Official English Documentation for ImageAI! — ImageAI 2.1.5. – URL: <https://imageai.readthedocs.io/en/latest/> (дата обращения: 10.05.2023).

РАБОТА 14

СППР «МНОГОЗАДАЧНАЯ СИСТЕМА РАСПОЗНАВАНИЯ». Часть 1

Цель работы: Написать программу для создания СППР, осуществляющей распознавания объектов на картинке и отслеживания объектов на видео с заданной пользователем вероятностью.

Подготовка к выполнению задания.

1. На компьютере должен быть выход в интернет.
2. Выполните работы 12 и 13, приведенные выше.
3. Поместите в проект набор картинок и видео для тестирования работы СППР, а так же все необходимые внешние библиотеки согласно [4, 5].

Задание. Распознавание объектов с заданной вероятностью.

1. Откройте в проекте `Project_porosity` файл `DSS_porosity`, созданный на практическом занятии 9.
2. Запишите в word-отчет пути, в которые у Вас на компьютере(ноутбуке) установлены: Python, QtDesigner, PyCharm и внешние библиотеки проекта.
3. Создайте интерфейс СППР, используя **Qt Designer**.

В интерфейсе СППР на главной форме сверху в меню реализуйте для пользователя возможность установки выполнения задач: распознавания объектов на картинке и отслеживания объектов на видео (из файла, с камеры). После выбора пользователем соответствующей строки меню (задачи) в окне, расположенном ниже этого меню, должна появиться соответствующая картинка или соответствующее видео с еще

нераспознанными объектами. Причем если загружается видео, то рядом расположите кнопки для его запуска (останова).

Справа от указанного выше окна на форме СППР реализуйте для пользователя возможность установки вероятности распознавания объектов на картинке и на видео.

3. Напишите код для вывода в указанное выше окно на форме результата срабатывания алгоритмов распознавания (отслеживания) в зависимости с указанными пользователем задачами. Алгоритмы должны учитывать указанную пользователем вероятность, то есть выделяться в рамки должны только те объекты, вероятность распознавания которых не ниже указанной пользователем. Сами алгоритмы см. из лабораторных работ 13, 14 из [1].
4. Разместите на форме кнопку «Запустить алгоритм распознавания» с указанной выше вероятностью. Если вероятность пользователем не указана, то пусть алгоритм находит и выделяет в рамки все найденные на нем объекты. Если вероятность пользователем указана, то пусть срабатывает алгоритм из задания 3 настоящей работы.

ЛИТЕРАТУРА К РАБОТЕ

1. Python Imaging Library (Fork) . – URL: <https://pypi.org/project/Pillow/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobходимое_proxorenok/ (дата обращения: 10.05.2023).
3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата

обращения: 10.05.2023).

4. Official English Documentation for ImageAI! — ImageAI 2.1.5. – URL: <https://imageai.readthedocs.io/en/latest/> (дата обращения: 10.05.2023).
5. ImageAI GitHub. – URL: <https://github.com/OlafenwaMoses/ImageAI/tree/master/imageai/Detection> (дата обращения: 10.05.2023).

РАБОТА 15

СППР «МНОГОЗАДАЧНАЯ СИСТЕМА РАСПОЗНАВАНИЯ». Часть 2

Цель работы: Формирование для СППР «Распознавание и отслеживание объектов» списка всех найденных на данном видео (из файла/с камеры) или картинке объектов с указанной пользователем вероятностью.

Подготовка к выполнению задания

1. На компьютере должен быть выход в интернет.
2. Выполните работу 4, приведенную выше.

Задание. Формирование списка всех найденных объектов

1. Приведите скриншоты форм проводника на которых видны полные пути, в которые у Вас на компьютере(ноутбуке) установлены: Python, QtDesigner, PyCharm и внешние библиотеки проекта.
2. В ui-файл главной формы Вашей СППР под строкой для задания нижней границы вероятности распознавания объектов разместите окно с лентой прокрутки. Сюда будет выводиться список, сформированный в задании 3 (см. ниже). Приведите скриншоты полученной ui-формы так, чтобы были видны наименования (имена) всех объектов на форме.
3. Напишите код для вывода на форму СППР «Распознавание и отслеживание объектов», созданную в задании 1, списка с полями:
 - наименование распознанного объекта с вероятностью не ниже указанной

пользователем (см. в задании 1),

- количество таких объектов,

- номер кадра для видео из файла и время съемки кадра для видео с камеры.

На видео (картинке) в окне с результатом эти объекты и только они должны выделяться в квадратик после запуска алгоритма на выполнение. То есть это будет список всех найденных на данном видео (из файла/с камеры) или картинке объектов с указанной выше вероятностью.

4. Протестируйте работу написанного в задании 3 кода, сделайте скриншоты с демонстрацией срабатывания его для всех опций (для видео (из файла, с камеры) и картинки) и разместите их в word-отчет. Скриншоты формируйте следующим образом: на заднем плане в окне PyCharm виден полностью код для соответствующего этапа тестирования, форма СППР с установлением опции. Ниже разместите форму с результатом срабатывания алгоритма, соответствующего демонстрации срабатывания опции. Скриншоты сопровождайте поясняющим текстом.

ЛИТЕРАТУРА К РАБОТЕ

1. Python Imaging Library (Fork) . – URL: <https://pypi.org/project/Pillow/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobходимое_proxoreнок/ (дата обращения: 10.05.2023).
3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).

4. Official English Documentation for ImageAI! — ImageAI 2.1.5. – URL: <https://imageai.readthedocs.io/en/latest/> (дата обращения: 10.05.2023).
5. ImageAI GitHub. – URL: <https://github.com/OlafenwaMoses/ImageAI/tree/master/imageai/Detection> (дата обращения: 10.05.2023).

РАБОТА 16

СППР «МНОГОЗАДАЧНАЯ СИСТЕМА РАСПОЗНАВАНИЯ». Часть 3

Цель работы: Разработка меню подзадач для СППР. Подзадачами являются две следующие задачи: распознавать все объекты, которые можно найти с помощью ImageAI; распознавать некоторые объекты, которые интересуют пользователя.

Подготовка к выполнению задания

1. Выполните работу 15, приведенную выше.
2. Посмотрите видео «Чтение данных из Excel файла в Python. Библиотека openpyxl в Python», <https://www.youtube.com/watch?v=d5jHpP5uI>
Полезно так же посмотреть: Работа с документами Excel в Python. Использование библиотек. Записки программиста, https://www.youtube.com/watch?v=n9MwRe_qJI
3. Посмотрите приведенный ниже пример создания списка с чекбоксами:

```
import sys
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *

class Window(QWidget):
    def __init__(self, rows, columns):
        super().__init__()

        self.table = QTableWidgetItem(rows, columns, self)

        for row in range(rows):
            widget = QWidget()
            checkbox = QCheckBox()
```

```

checkbox.setChecked(Qt.Unchecked)
layoutH = QHBoxLayout(widget)
layoutH.addWidget(checkbox)
layoutH.setAlignment(Qt.AlignCenter)
layoutH.setContentsMargins(0, 0, 0, 0)

self.table.setCellWidget(row, 0, widget)
self.table.setItem(row, 1, QTableWidgetItem(str(row)))

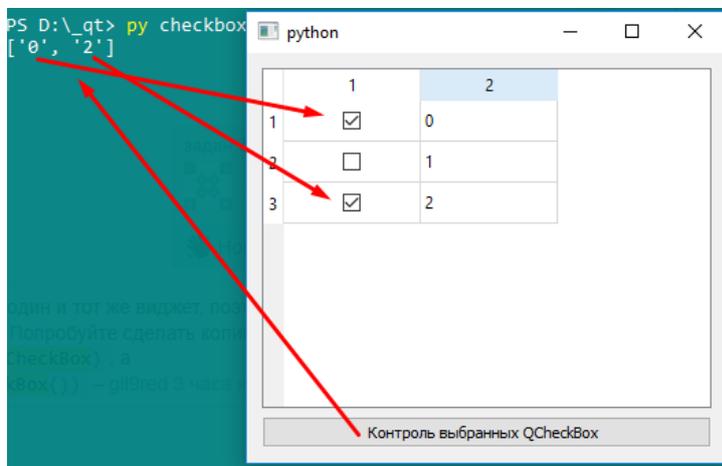
self.button = QPushButton("Контроль выбранных QCheckBox ")
self.button.clicked.connect(self.ButtonClicked)

layoutV = QVBoxLayout(self)
layoutV.addWidget(self.table)
layoutV.addWidget(self.button)

def ButtonClicked(self):
    checked_list = []
    for i in range(self.table.rowCount()):
        if self.table.cellWidget(i,
0).findChild(type(QCheckBox())).isChecked():
            checked_list.append(self.table.item(i, 1).text())
    print(checked_list)

if __name__ == '__main__':
    app = QApplication(sys.argv)
    window = Window(3, 2)
    window.resize(350, 300)
    window.show()
    sys.exit(app.exec_())

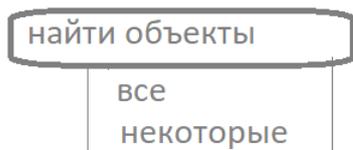
```



Задание. Задание подзадач распознавания для СППР

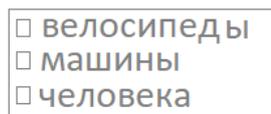
1. Приведите скриншоты форм проводника на которых видны полные пути, в которые у Вас на компьютере(ноутбуке) установлены: Python, QtDesigner, PyCharm и внешние библиотеки проекта.
2. В ui-файл главной формы Вашей СППР в верхнем меню разместите

кнопку с выбором задачи распознавания:



Здесь «все» означает что надо находить с заданной пользователем точностью и выводить в список все объекты, которые можно найти с помощью ImageAI. Эта задача уже решена в лабораторной работе 16. Поэтому если пользователь выберет эту опцию, то должен заработать уже написанный Вами алгоритм этой работы.

Если пользователю надо найти только конкретный (конкретные) нужный (нужные) ему объект (объекты), то он должен указать это, кликнув на «некоторые». Тогда должен возникнуть список с чекбоксами. Например:



В нем пользователь должен кликнуть соответствующий (соответствующие) чекбокс (чекбоксы) и программа должна запомнить, что искать надо именно эти кликнутые объекты. То есть в этом случае надо находить с заданной пользователем точностью и выводить в список все найденные кликнутые в чекбоксах пользователем объекты.

Организуите этот список объектов с чекбоксами, которые может отыскивать Ваша система, считав наименования объектов из БД в Excel-файле. То есть в Excel-файле запишите в колонку наименования тех объектов, которые можно будет отыскивать в Вашей системе. Затем считайте его в Python и разместите в разворачивающемся меню рядом с соответствующими чекбоксами. Рекомендую перед помещением этого списка в разворачивающемся меню сначала создать список чекбоксов соответствующей длины.

В этом случае пользователь интересуется есть ли нужный ему

объект на картинке/видео. Поэтому здесь алгоритм должен находить каждый из указанных пользователем объектов, если заранее неизвестно есть ли он там. Если какого-то из указанных объектов нет на картинке/видео, то система должна сообщить об этом.

Приведите код для указанного выше меню и формирования указанного выше списка с контролем правильности принятия программой указанных опций. Команды кода сопроводите комментариями.

3. Напишите алгоритм реализации поиска объектов, указанных пользователем в меню (то есть с учетом опций из задания 2).

Приведите соответствующие фрагменты кода для решения этих двух задач. Команды кода сопроводите комментариями.

4. Протестируйте работу написанного в задании 2 и 3 кода, сделайте скриншоты с демонстрацией срабатывания его для всех опций (для видео (из файла, с камеры) и картинки) и разместите их в word-отчет. Скриншоты формируйте следующим образом: на заднем плане в окне PyCharm виден полностью код (выделите его синим цветом) для соответствующего этапа тестирования, форма СППР с установлением опции. Ниже разместите форму с результатом срабатывания алгоритма, соответствующего демонстрации срабатывания опции. Скриншоты сопровождайте поясняющим текстом.

ЛИТЕРАТУРА К РАБОТЕ

1. Python Imaging Library (Fork) . – URL: <https://pypi.org/project/Pillow/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobxodimoe_proxorenok/

(дата обращения: 10.05.2023).

3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).
4. Official English Documentation for ImageAI! — ImageAI 2.1.5. – URL: <https://imageai.readthedocs.io/en/latest/> (дата обращения: 10.05.2023).
5. ImageAI GitHub. – URL: <https://github.com/OlafenwaMoses/ImageAI/tree/master/imageai/Detection> (дата обращения: 10.05.2023).

РАБОТА 17

СППР «МНОГОЗАДАЧНАЯ СИСТЕМА РАСПОЗНАВАНИЯ». Часть 4

Цель работы: создание логических правил наблюдений и сигнализации для СППР.

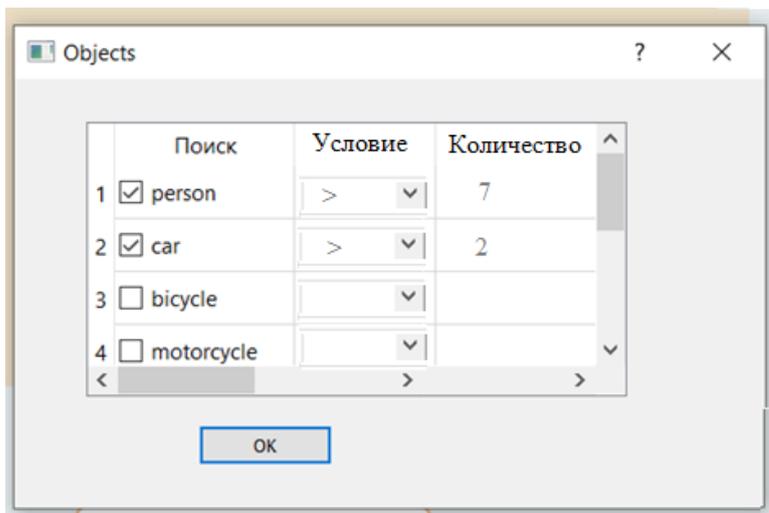
Подготовка к выполнению задания.

1. Выполните работу 16, приведенную выше.

Задание. Создание логических правил наблюдений и сигнализации для СППР.

1. Приведите скриншоты форм проводника на которых видны полные пути, в которые у Вас на компьютере(ноутбуке) установлены: Python, QtDesigner, PyCharm и внешние библиотеки проекта.
2. Дайте пользователю в настройках через интерфейс Вашей СППР указывать правила наблюдения и реализуйте сигнализацию звуком о

наступлении событий, указанных в этих правилах. Например, если найденных системой объектов оказывается в кадре (на картинке) больше, чем указал пользователь через интерфейс в настройках наблюдения, то система должна просигнализировать звуком о наступлении такого события. Причем пользователь через чекбокс на форме настройки наблюдения указывает на какие объекты он налагает правила. Например:



3. Приведите ui-файл и скриншот соответствующего фрагмента кода для решения этой задачи. Команды кода сопроводите комментариями.
4. Протестируйте работу написанного, сделайте скриншоты с демонстрацией срабатывания его для всех опций (для видео (из файла, с камеры) и картинки) и разместите их в word-отчет. Скриншоты формируйте следующим образом: на заднем плане в окне PyCharm виден полностью код (выделите его синим цветом) для соответствующего этапа тестирования, форма СППР с установлением опции. Ниже разместите форму с результатом срабатывания алгоритма, соответствующего демонстрации срабатывания опции. Скриншоты сопровождайте поясняющим текстом.
5. Разместите полный код Вашего приложения с комментариями.

ЛИТЕРАТУРА К РАБОТЕ

1. Python Imaging Library (Fork) . – URL: <https://pypi.org/project/Pillow/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobходимое_proxorenok/ (дата обращения: 10.05.2023).
3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).
4. Official English Documentation for ImageAI! — ImageAI 2.1.5. – URL: <https://imageai.readthedocs.io/en/latest/> (дата обращения: 10.05.2023).
5. ImageAI GitHub. – URL: <https://github.com/OlafenwaMoses/ImageAI/tree/master/imageai/Detection> (дата обращения: 10.05.2023).

РАБОТА 18

СППР «МНОГОЗАДАЧНАЯ СИСТЕМА РАСПОЗНАВАНИЯ». Часть 5

Цель работы: вырезка области изображения и его увеличение в СППР.

Подготовка к выполнению задания

1. Выполните работу 17, приведенную выше.

Задание. Вырезка области изображения и его увеличение в СППР

1. Приведите скриншоты форм проводника на которых видны полные пути, в которые у Вас на компьютере(ноутбуке) установлены: Python, QtDesigner, PyCharm и внешние библиотеки проекта.
2. Дайте пользователю в настройках через интерфейс Вашей СППР перед запуском алгоритма распознавания вырезать мышкой кусок изображения, на котором алгоритм должен будет распознавать объекты, и увеличивать этот кусок до нужного пользователю размера пропорционально.

То есть пусть в окно формы настройки изначально выводится поданное в систему изображение (картинка, видео из файла или камеры). Затем пользователь с помощью мышки берет в прямоугольник область интереса. Первый клик пользователя соответствует левому верхнему углу этого прямоугольника. Затем на этой же форме дайте пользователю возможность увеличить вырезанное изображение до нужного пользователю размера. Затем этот кусок выведите на главную форму вместо того, который выводился у вас в окно главной формы в предыдущих лабораторных работах. Далее все алгоритмы должны работать как и раньше, но только уже с этим вырезанным куском и увеличенным (если пользователь его увеличивал). На форме настройки предусмотрите кнопку «undo» - возврата к предыдущему изображению (пусть она позволяет отменить до 5 последних действий) и возврата исходного изображения.

Приведите ui-файл и скриншот соответствующего фрагмента кода для решения этой задачи. Команды кода сопроводите комментариями.

3. Протестируйте работу написанного кода, сделайте скриншоты с демонстрацией срабатывания его для всех опций (для видео (из файла, с камеры) и картинки) и разместите их в word-отчет. Скриншоты формируйте следующим образом: на заднем плане в окне PyCharm виден полностью код (выделите его синим цветом) для соответствующего этапа

тестирования, форма СППР с установлением опции. Ниже разместите форму с результатом срабатывания алгоритма, соответствующего демонстрации срабатывания опции. Скриншоты сопровождайте поясняющим текстом.

4. Разместите полный код Вашего приложения с комментариями.

ЛИТЕРАТУРА К РАБОТЕ

1. Python Imaging Library (Fork) . – URL: <https://pypi.org/project/Pillow/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobходимое_proxoreнок/ (дата обращения: 10.05.2023).
3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).
4. Official English Documentation for ImageAI! — ImageAI 2.1.5. – URL: <https://imageai.readthedocs.io/en/latest/> (дата обращения: 10.05.2023).
5. ImageAI GitHub. – URL: <https://github.com/OlafenwaMoses/ImageAI/tree/master/imageai/Detection> (дата обращения: 10.05.2023).

РАБОТА 19

СППР «МНОГОЗАДАЧНАЯ СИСТЕМА РАСПОЗНАВАНИЯ». Часть 6

Цель работы: перевод в оттенки серого и увеличение контрастности и четкости изображения в СППР.

Подготовка к выполнению задания.

1. Выполните работу 18, приведенную выше.
2. Перевод изображения в изображение в оттенках серого (режим L) и обратно в цветное (режим P):

```
from PIL import Image
im = Image.open("im.jpg")
im_l = im.convert('L')
im_p = im.convert('P')
im.show()
im_l.show()
im_p.show()
```

3. Посмотрите следующие учебные материалы:

Линейное контрастирование изображений [1, стр. 19]

Соляризация изображения [1, стр. 20]

Препарирование изображений [1, стр. 21].

Задание. Перевод в оттенки серого и увеличение контрастности в СППР

1. Приведите скриншоты форм проводника на которых видны полные пути, в которые у Вас на компьютере(ноутбуке) установлены: Python, QtDesigner, PyCharm и внешние библиотеки проекта.
2. Дайте пользователю в настройках через интерфейс Вашей СППР перед запуском алгоритма распознавания включать и выключать режим перевода изображения в изображение в оттенках серого.

Сделайте это на той же форме настройки, которая сделана в лабораторной работе 19.

Если пользователь кликнул на опцию «перевод изображения в изображение в оттенках серого», то текущее на данный момент изображение должно перевестись в оттенки серого и стать новым текущим.

Если пользователь выключит этот режим, то должно возвратиться то цветное изображение, которое до перевода в оттенки серого было текущим.

Приведите `ui`-файл и скриншот соответствующего фрагмента кода для решения этой задачи. Команды кода сопроводите комментариями.

3. На той же самой панели настройки для включенной опции перевода изображения в изображение в оттенках серого должны появиться строки:

3.1. линейное контрастирование

3.2. задание режима более четкого изображения мелких объектов,

3.3. задание режима более четкого изображения крупных объектов.

Слева в каждой строке должен разместиться символ развернуть/свернуть, при клике на которую должны соответственно появляться/исчезать соответствующие формы. Справа в каждой строке разместите кнопку «undo» - возврата к предыдущему изображению (пусть она позволяет отменить до 5 последних действий) и возврата исходного изображения (то есть до двигания движков).

Здесь если развернут режим «линейное контрастирование», то появляются движки для установки пользователем желаемых значений минимальной `gmin` и максимальной `gmax` выходной яркости (см. [1], стр. 19), двигая которые пользователь видит вместо ранее выведенного изображения в окне формы новое изображение, соответствующее установленным значениям параметров.

Если включен режим более четкого изображения мелких объектов, то открываются под этими словами формочки с движками для каждого из

параметров методов, предназначенных именно для этой цели. Это методы: соляризация, препарирование (случай «ж») [1, стр. 20, 21].

Если включен режим более четкого изображения крупных объектов, то открывается под этими словами формочка с движками для параметров метода, предназначенного именно для этой цели. Это метод: бинаризация, то есть препарирование (случай «а») [1, стр. 20, 21].

С помощью инструментов, сделанных в задании 3, пользователь может менять как ему надо изображение в оттенках серого, полученное после клика на активацию случая перевода в оттенки серого, реализованного в задании 2.

Приведите `ui`-файл и скриншот соответствующего фрагмента кода для решения каждой из задач задания 3. Команды кода сопроводите комментариями.

Далее при распознавании объектов система уже должна работать согласно этим установкам как для картинки, так и для видео из файла и с камеры.

4. Протестируйте работу написанного кода, сделайте скриншоты с демонстрацией срабатывания его для всех опций (для всего видео (из файла, с камеры) и картинки и области на нем/ней) и разместите их в `word`-отчет. Скриншоты формируйте следующим образом: на заднем плане в окне `PyCharm` виден полностью код (выделите его синим цветом) для соответствующего этапа тестирования, форма СППР с установлением опции. Ниже разместите форму с результатом срабатывания алгоритма, соответствующего демонстрации срабатывания опции. Скриншоты сопровождайте поясняющим текстом.
5. Разместите полный код Вашего приложения с комментариями.

ЛИТЕРАТУРА К РАБОТЕ

1. Обработка изображений [Электронный ресурс]: учеб. пособие. В 2 ч. Ч. 2. Цифровая обработка изображений / [сост.: Б.Н. Грудин, В.С. Плотников, С.В. Полищук]; Дальневосточный федеральный университет. – Электрон. дан. – Владивосток: Издательский дом Дальневост. федерал. ун-та, 2013. – Режим доступа: Computer university network. – Загл. с экрана. – URL: <https://studfile.net/preview/3368585/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobходимое_proxorenok/ (дата обращения: 10.05.2023).
3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).
4. Official English Documentation for ImageAI! — ImageAI 2.1.5. – URL: <https://imageai.readthedocs.io/en/latest/> (дата обращения: 10.05.2023).

РАБОТА 20

СППР «МНОГОЗАДАЧНАЯ СИСТЕМА РАСПОЗНАВАНИЯ». Часть 7

Цель работы: удаление шумов на изображении в СППР.

Подготовка к выполнению задания.

1. Посмотрите какие фильтры можно применять для удаления шумов (см. задание 1 в работе 6, приведенной выше).

Задание. Удаление шумов на изображении в СППР

1. Приведите скриншоты форм проводника на которых видны полные пути, в которые у Вас на компьютере(ноутбуке) установлены: Python, QtDesigner, PyCharm и внешние библиотеки проекта.
2. Дайте пользователю на дочерней форме, полученной в предыдущей лабораторной работе, через интерфейс перед запуском алгоритма распознавания удалять шумы на текущем изображении. Пусть пользователь это делает, двигая движки для параметров методов удаления шумов. То изображение, которое получилось для текущей установки движков, должно считаться в СППР текущим и оно должно появиться на главной форме СППР после закрытия этой дочерней формы.

Разместите кнопку «undo» - возврата к предыдущему изображению (пусть она позволяет отменить до 5 последних действий) и возврата исходного изображения (то есть до двигания движков). Рядом разместите кнопку отмены возврата.

Приведите иі-файл и скриншот соответствующего фрагмента кода для решения этой задачи. Команды кода сопроводите комментариями.

3. Протестируйте работу написанного кода для фильтрации, сделайте скриншот с демонстрацией срабатывания его. Следом за демонстрацией срабатывания фильтра разместите соответствующие скриншоты срабатывания всех алгоритмов распознавания на соответствующем

отфильтрованном изображении. Сделайте это для всех опций (для всего видео и его области (из файла, с камеры) и картинки) и разместите их в word-отчет. Скриншоты формируйте следующим образом: на заднем плане в окне PyCharm виден полностью код (выделите его синим цветом) для соответствующего этапа тестирования, форма СППР с установлением опции. Ниже разместите форму с результатом срабатывания алгоритма, соответствующего демонстрации срабатывания опции. Скриншоты сопровождайте поясняющим текстом.

4. Разместите полный код Вашего приложения с комментариями.

ЛИТЕРАТУРА К РАБОТЕ

1. Обработка изображений [Электронный ресурс]: учеб. пособие. В 2 ч. Ч. 2. Цифровая обработка изображений / [сост.: Б.Н. Грудин, В.С. Плотников, С.В. Полищук]; Дальневосточный федеральный университет. – Электрон. дан. – Владивосток: Издательский дом Дальневост. федерал. ун-та, 2013. – Режим доступа: Computer university network. – Загл. с экрана. – URL: <https://studfile.net/preview/3368585/> (дата обращения: 10.05.2023).
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8.-Текст:электронный).–URL: https://codernet.ru/books/python/python_3_samoe_neobходимое_proxorenok/ (дата обращения: 10.05.2023).
3. Python 3 и PyQt 6. Разработка приложений/ Н. А. Прохоренок, В. А. Дронов. -СПб.: Б:ХВ-Петербург, 2023.- 832 с.: ил.-(Профессиональное программирование). – URL: <https://coollib.net/b/617121-vladimir-dronov-python-3-i-pyqt-6-razrabotka-prilozheniy/readp?p=186&cnt=9000> (дата обращения: 10.05.2023).
4. Official English Documentation for ImageAI! — ImageAI 2.1.5. – URL: <https://imageai.readthedocs.io/en/latest/> (дата обращения: 10.05.2023).

*Методы и классы Python, используемые для создания
СППР «Определение качества поверхности пористого материала»*

В программе использованы следующие методы для работы с изображениями:

cv2.VideoCapture - метод для захвата видео;

cv2.cvtColor – метод конвертации цветовых каналов изображения;

cv2.cvtGaussianBlur - метод удаления шумов с изображения;

cv2.findContours - метод поиска контуров. Поиск контуров использует для работы монохромное изображение и работает так, что все пиксели картинки с ненулевым цветом будут интерпретироваться как 1, а все нулевые останутся 0;

cv2.contourArea - метод задающий площадь контура;

cv2.drawContours - метод изображающий контуры в кадре.

В программе использовать следующие классы:

PIL.Image.open - класс, представляющий объект изображения;

PIL.ImageEnhance.Contrast - класс, изменяющий контрастность изображения;

PIL.ImageEnhance.Brightness - класс, изменяющий яркость изображения;

PIL.ImageEnhance.Sharpness - класс, изменяющий резкость изображения;

В программе для работы с БД использовать следующие методы:

sqlite3.connection;

sqlite.connection().cursor;

sqlite.connection().cursor().execute.

Последовательность установки
библиотек для работы с *ImageAI*.

- Загрузите и установите **Python 3.7** , **Python 3.8** , **Python 3.9** или **Python 3.10**.
- Установить зависимости (ЦП)

```
pip install cython pillow>=7.0.0 numpy>=1.18.1 opencv-  
python>=4.1.2 torch>=1.9.0 --extra-index-url  
https://download.pytorch.org/whl/cpu torchvision>=0.10.0 --  
extra-index-url https://download.pytorch.org/whl/cpu  
pytest==7.1.3 tqdm==4.64.1 scipy>=1.7.3 matplotlib>=3.4.3  
mock==4.0.3
```

- Установить зависимости (GPU/CUDA)

```
pip install cython pillow>=7.0.0 numpy>=1.18.1 opencv-  
python>=4.1.2 torch>=1.9.0 --extra-index-url  
https://download.pytorch.org/whl/cu102 torchvision>=0.10.0 -  
-extra-index-url https://download.pytorch.org/whl/cu102  
pytest==7.1.3 tqdm==4.64.1 scipy>=1.7.3 matplotlib>=3.4.3  
mock==4.0.3
```

- Если вы планируете обучать пользовательские модели ИИ, выполните приведенную ниже команду, чтобы установить дополнительную зависимость.

```
pip install  
pycocotools@git+https://github.com/gautamchitnis/cocoapi.git  
@cocodataset-master#subdirectory=PythonAPI
```

- **ImageAI**

```
pip install imageai --upgrade
```

1. **Изменить тему.** Для того чтобы изменить тему в PyCharm надо выполнить последовательность действий: в меню выбрать **файл-Setting-** слева от **Editor** стоит треугольничек, щелкаем по нему и выбираем **Plugins-** появится таблица, в левом столбце выбираем плагин (желаемую тему), а в правом столбце вверху печатается выбранная тема и в этом же столбце ниже покажется как она выглядит, выберем, например, **Material Theme UI** после того как она установится переходите в **Editor-Color Scheme** (нажать на значёк слева от **Color Scheme**)-**General-**в правом столбце видим во второй строке **Scheme** нажимаем на треугольничек (расположенный около знака настроек) высветится подменю тем в нем выбираем например **Material Oceanic** или другую понравившуюся например **Darcula** и выставляете данную тему т.е. нажимаете **ОК**.

Далее Вы можете настроить саму тему у себя каким цветом у Вас будет отображаться тот или иной код. Для этого в **Color Scheme** выбираете **Python** – и если хотите поменять цвет ключевых слов в коде, то в правом столбце слева выбираете **keyword** (-англ. ключевое слово)(в коде, изображенном в низу выделяются ключевые слова) и справа в foreground ((-англ. передний план) щелкните по прямоугольнику расположенному рядом, появится окошко с диаграммой цветов, выберите желаемый цвет и нажмите кнопку **Choose - ОК** . Или можно было, после того как в **Color Scheme** выбрали **Python**, в появившемся во втором столбце внизу примере кода щелкнуть на нужное Вам слово, и аналогично в правом столбце справа выбрать для него цвет.

2. **Изменить шрифты и их размеры.**

В меню выбрать **файл-Setting- Editor- Font** (-англ. шрифт).

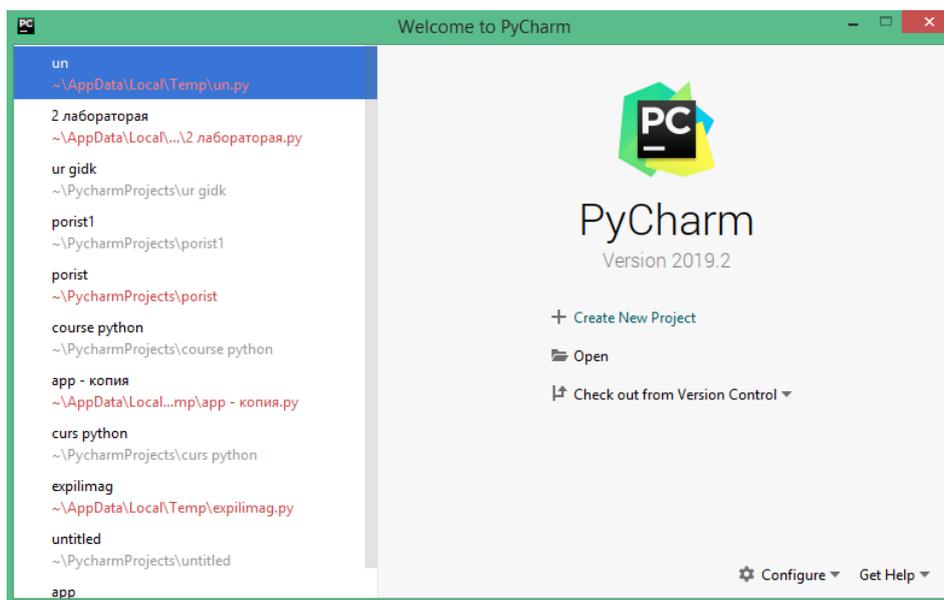
3. **Вызов функции автозаполнение: Ctrl и пробел**

4. **Комментирование строки (находясь на англ): Ctrl и /**

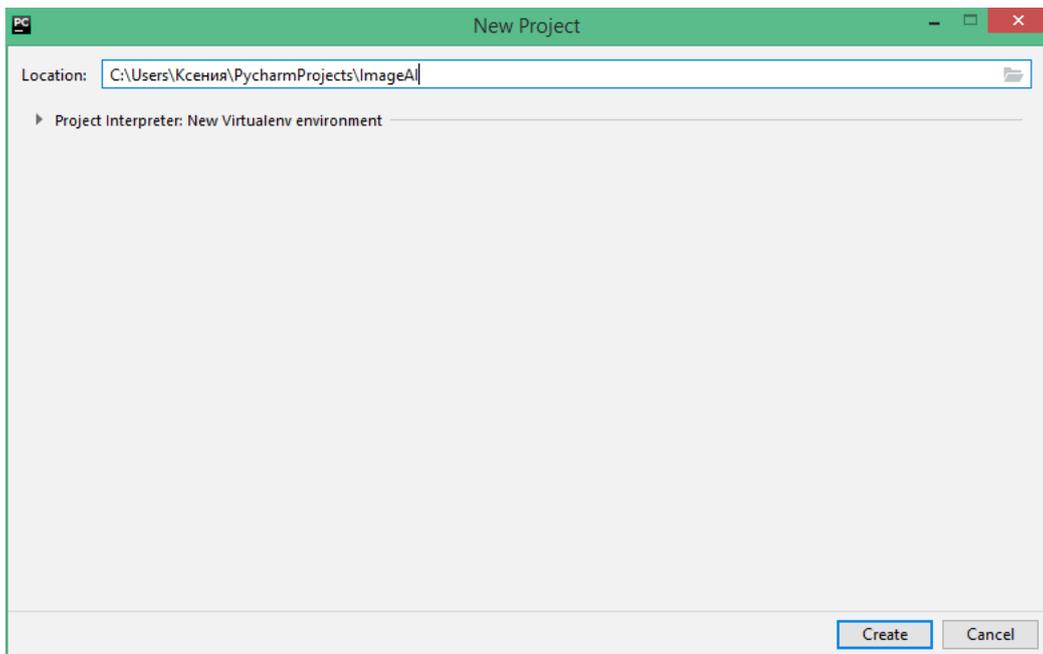
5. **Комментирование группы строк:** выделить эти строки и **Ctrl и /**
6. **Сворачивает выделенную часть кода (находясь на русск):** **Ctrl и /**
7. **Сворачивать блок кода:** **Ctrl и +**
8. **Разворачивать блок кода:** **Ctrl и -**
9. **PyCharm отформатирует Ваш код согласно стандарту PEP 8:**
Ctrl и Alt и L
10. **Открыть консоль Python:** в меню выбрать Tools - Python Console. Тогда в меню внизу слева появится консоль Python и список заданных переменных справа см. скрин в низу

Создание нового проекта

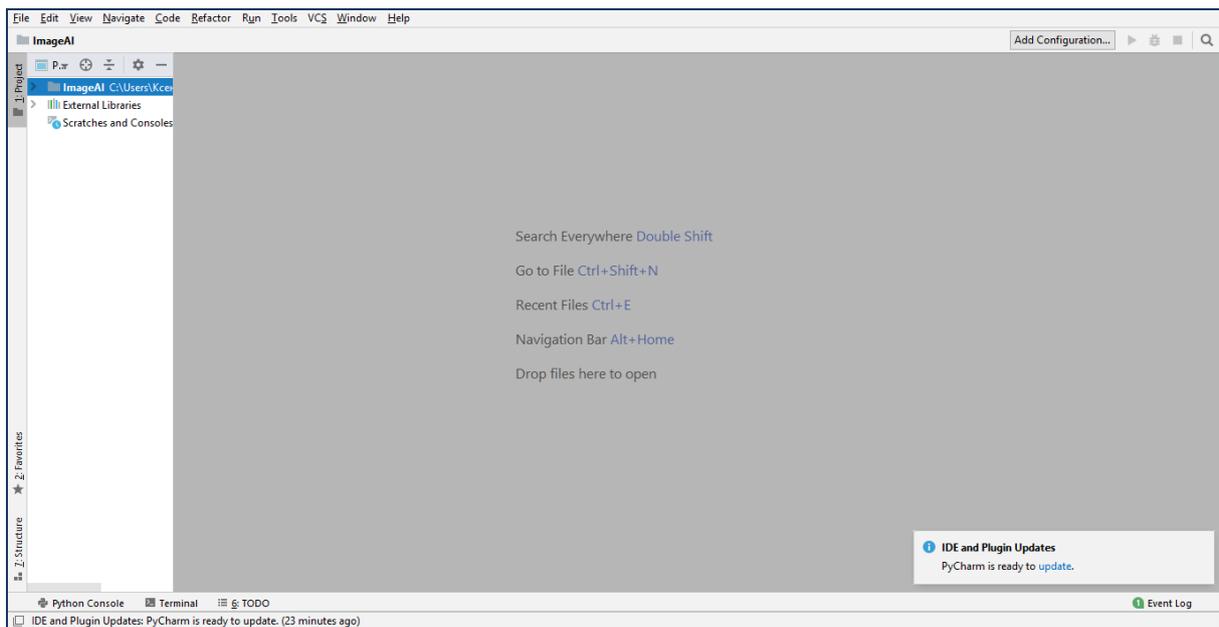
1. Запустите **PyCharm**.
2. В правой области появившегося окна выберите **Create New Project** (**Создать новый проект**).



3. Во вновь появившемся окне укажите путь к проекту, в нашем случае проект будет размещаться в папке **ImageAI**, которая собственно и будет названием проекта. Затем нажмите кнопку **Create**.

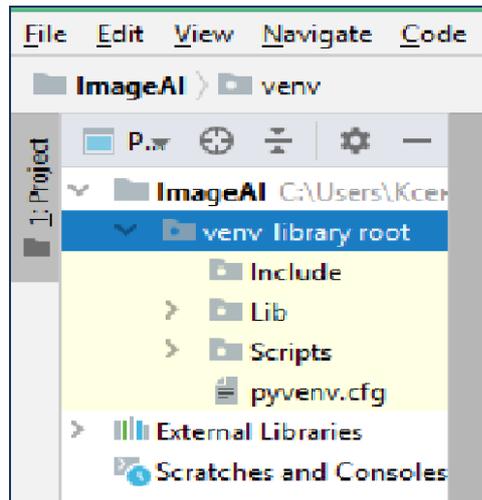


В результате будет создан пустой проект.

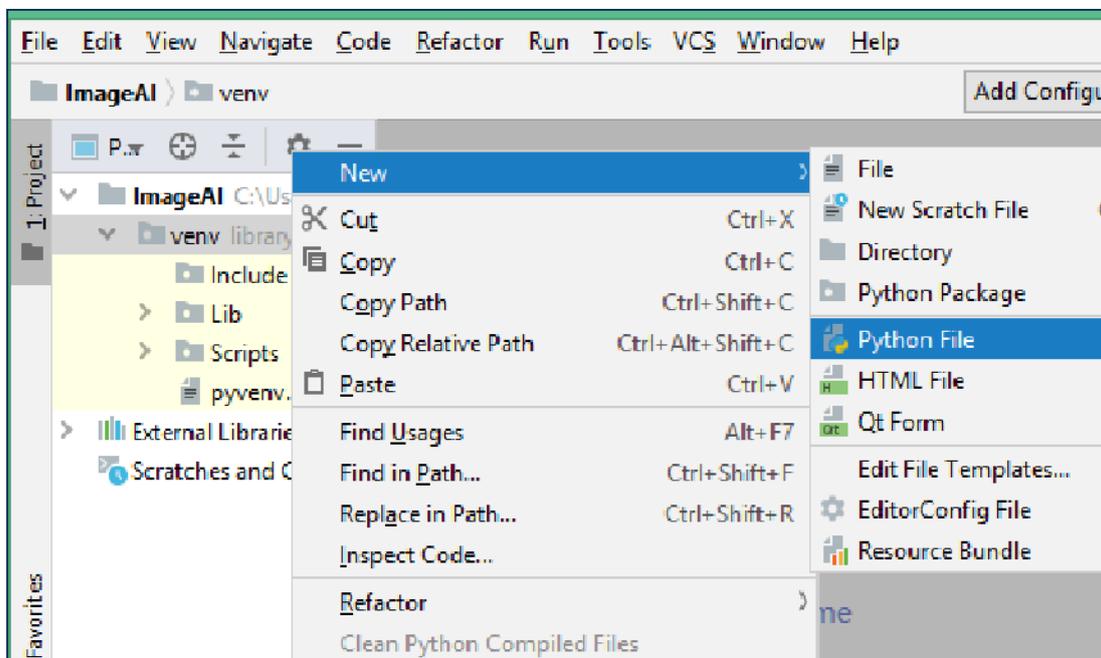


Создание нового документа в проекте

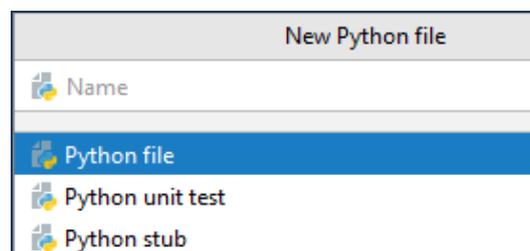
Для того чтобы создать новый документ необходимо щелкнуть на значок « > », расположенный слева от названия проекта. Из выпадающего списка выберем пункт **venv**



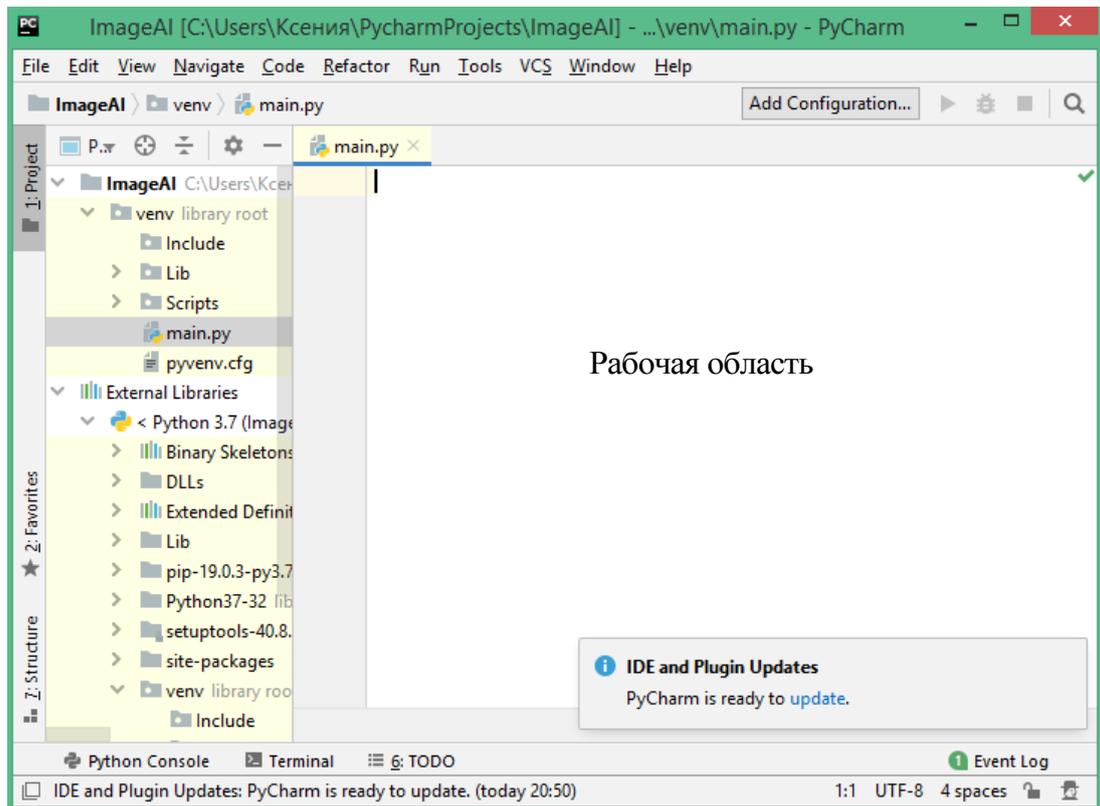
и путем нажатия правой кнопки мыши в появившемся контекстном меню осуществим переход **New** → **Python File**.



Откроется окно, в котором необходимо указать название файла, например, **main**. По завершении нажмите клавишу **Enter** (кнопку **OK**).



В рабочей области вновь созданного документа, наберите код программы:

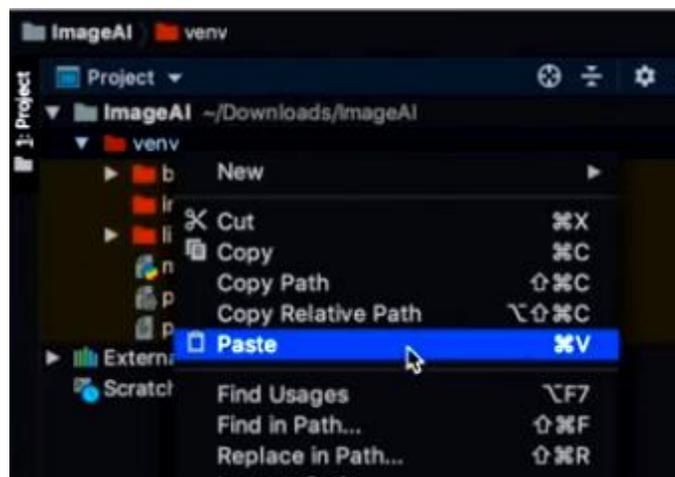


Если для решения поставленной задачи нужны библиотеки, то сначала надо выполнить установку всех необходимых библиотек в проект.

Вставка документа (файла) в проект

Вставить файл в проект можно следующим образом:

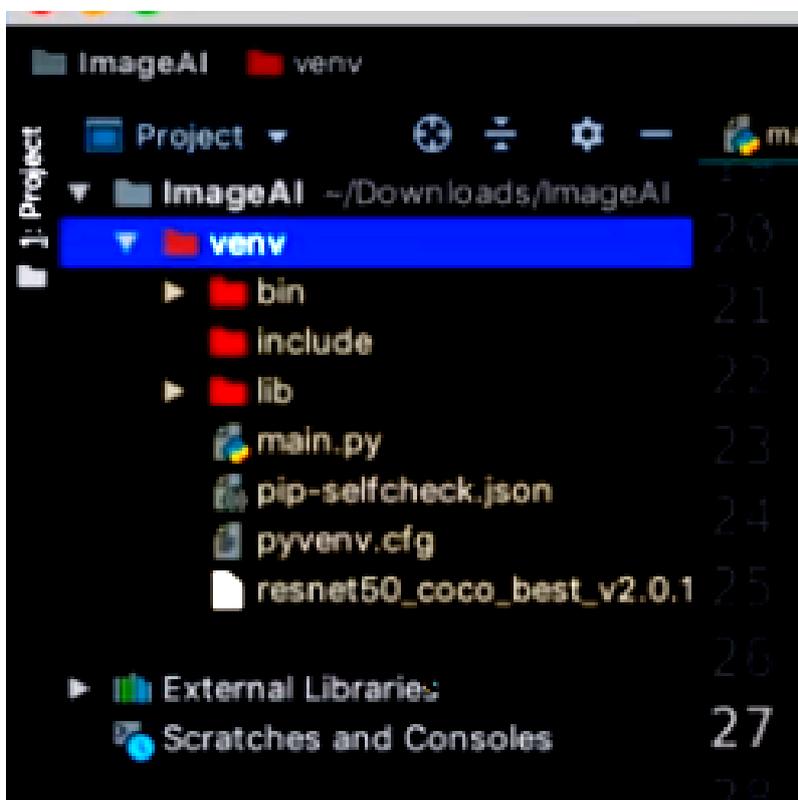
- 1) выберите файл и скопируйте его;
- 2) используя правую кнопку мыши, выберите из контекстного меню папки **venv** команду **Paste**;



3) в появившемся окне в строке **New name** отобразится имя файла, который надо вставить;



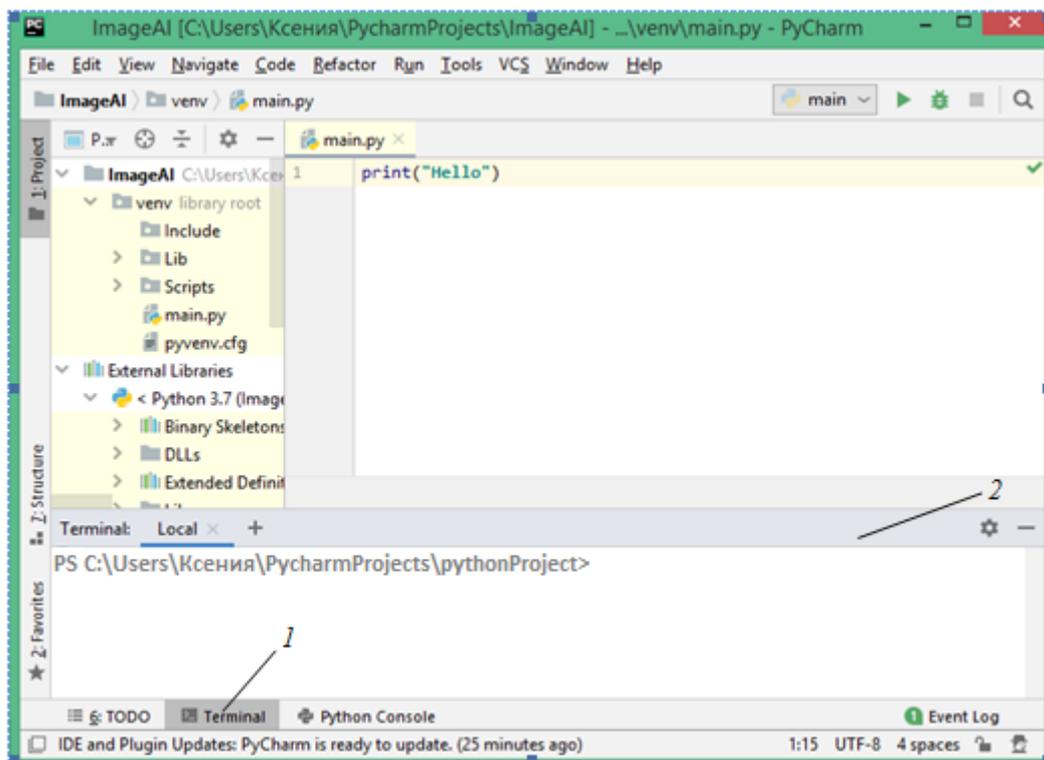
4) нажмите кнопку **ОК** и Вы увидите этот файл в текущей папке.



Установка внешних библиотек в проект

Способ 1 В PyCharm, в меню нажмите **File -> Settings -> Project: [название проекта] -> Project Interpreter**. В открывшемся окне нажмите на знак плюса в правой части окна->В поле сверху(в поисковой строке) открывшегося окна напишите название библиотеки, которую хотите установить, выберите её в списке снизу и нажмите **Install Package** и подождите пока библиотека установится.

Способ 2 (через Terminal, используя **pip**) . Войдите в **Terminal** (см.рис. ниже вкладку 1):



После приглашения для ввода команды, вводим команду в окне 2, например, для установки внешней библиотеки Pillow: `pip install pillow`

Нажмите клавишу **Enter**, начнется загрузка этой библиотеки.

Если библиотека была установлена ранее, то она просто добавляется в данный проект. В противном случае она скачается, установится и добавится в данный проект. Теперь Вы ее можете посмотреть в папочке **External Libraries**, отвечающей за дополнительные библиотеки проекта.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
РАБОТА 1. ЗАГРУЗКА ИЗОБРАЖЕНИЯ И ПОЛУЧЕНИЕ ИНФОРМАЦИИ О НЕМ.	4
РАБОТА 2. ЦВЕТОВЫЕ МОДЕЛИ. СОЗДАНИЕ И СОХРАНЕНИЕ ИЗОБРАЖЕНИЯ.....	7
РАБОТА 3. КАНАЛЫ ЦИФРОВОГО ИЗОБРАЖЕНИЯ. ГИСТОГРАММА ИЗОБРАЖЕНИЯ.....	13
РАБОТА 4. МАНИПУЛИРОВАНИЕ ИЗОБРАЖЕНИЕМ	16
РАБОТА 5. МАНИПУЛЯЦИИ НА ИЗОБРАЖЕНИИ.....	21
РАБОТА 6. ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЙ. КОНТУРЫ НА ИЗОБРАЖЕНИИ.....	27
РАБОТА 7. СППР «ОПРЕДЕЛЕНИЕ КАЧЕСТВА ПОВЕРХНОСТИ ПОРИСТОГО МАТЕРИАЛА» Часть 1.....	31
РАБОТА 8. СППР «ОПРЕДЕЛЕНИЕ КАЧЕСТВА ПОВЕРХНОСТИ ПОРИСТОГО МАТЕРИАЛА». Часть 2.	41
РАБОТА 9. СППР «ОПРЕДЕЛЕНИЕ КАЧЕСТВА ПОВЕРХНОСТИ ПОРИСТОГО МАТЕРИАЛА». Часть 3.	43
РАБОТА 10. СППР «ОПРЕДЕЛЕНИЕ КАЧЕСТВА ПОВЕРХНОСТИ ПОРИСТОГО МАТЕРИАЛА». Часть 4.	47
РАБОТА 11. СППР «ОПРЕДЕЛЕНИЕ КАЧЕСТВА ПОВЕРХНОСТИ ПОРИСТОГО МАТЕРИАЛА». Часть 5.	49
РАБОТА 12. РАСПОЗНАВАНИЕ ОБЪЕКТОВ НА КАРТИНКЕ.....	53
РАБОТА 13. ОТСЛЕЖИВАНИЕ ОБЪЕКТОВ НА ВИДЕО.....	58

РАБОТА 14. СППР «МНОГОЗАДАЧНАЯ СИСТЕМА РАСПОЗНАВАНИЯ». Часть 1.....	65
РАБОТА 15. СППР «МНОГОЗАДАЧНАЯ СИСТЕМА РАСПОЗНАВАНИЯ». Часть 2.....	68
РАБОТА 16. СППР «МНОГОЗАДАЧНАЯ СИСТЕМА РАСПОЗНАВАНИЯ». Часть 3.....	70
РАБОТА 17. СППР «МНОГОЗАДАЧНАЯ СИСТЕМА РАСПОЗНАВАНИЯ». Часть 4.....	74
РАБОТА 18. СППР «МНОГОЗАДАЧНАЯ СИСТЕМА РАСПОЗНАВАНИЯ». Часть 5.....	76
РАБОТА 19. СППР «МНОГОЗАДАЧНАЯ СИСТЕМА РАСПОЗНАВАНИЯ». Часть 6.....	79
РАБОТА 20. СППР «МНОГОЗАДАЧНАЯ СИСТЕМА РАСПОЗНАВАНИЯ». Часть 7.....	83
Приложение 1. Методы и классы Python, используемые для создания СППР «Определение качества поверхности пористого материала»	85
Приложение 2. Последовательность установки библиотек для работы с ImageAI.....	86
Приложение 3. Изменение основных настроек в PyCharm.....	87
Приложение 4. Работа над проектом в PyCharm.....	89

**Шустова Ксения Петровна
Шустова Евгения Петровна**

АНАЛИЗ ИЗОБРАЖЕНИЙ И РАСПОЗНАВАНИЕ ОБРАЗОВ НА PYTHON

ПРАКТИКУМ

Учебное пособие

Корректурa авторов

Казанский (Приволжский) федеральный университет
Казань, ул. Кремлевская, д.18