ЛАБОРАТОРНАЯ РАБОТА №3

ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЙ

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 зображения 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, DETAIL, EDGE_ENHANCE, EDGE_ENHANCE_MORE, EMBOSS, FIND_EDGES, SHARPEN, SMOOTH и SMOOTH_MORE

из модуля **ImageFilter** и выведите отфильтрованное изображение на экран.

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

Задание 2. Контуры на изображении (один из способов).

1. Выделите контуры на изображении.

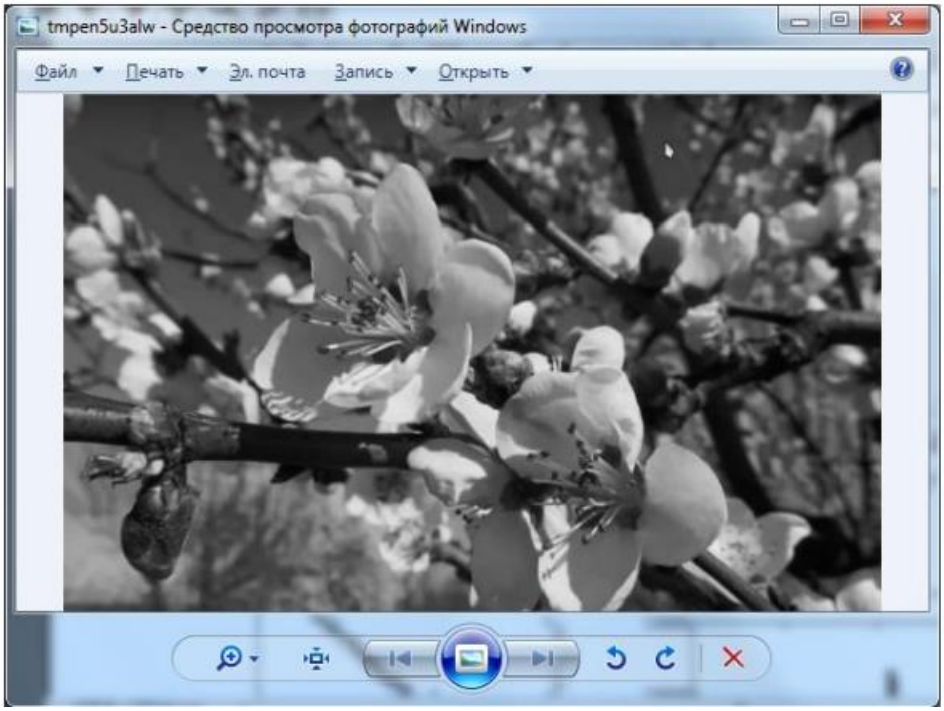
```
from PIL import Image
from pylab import *

# 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:

