

Размытие изображения в рабочей среде Python посредством библиотеки SciPy с помощью фильтра Гаусса

Звайгзне Алексей Юрьевич

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

В данной статье описывается обработка изображений в PyCharm на языке программирования Python с помощью фильтра Гаусса прописанной в библиотеке SciPy. Рассматриваются необходимые дополнительно библиотеки и команды для ввода-вывода, а также для сохранения конечного файла на диск.

Ключевые слова: Pycharm, Python, Scipy, фильтр Гаусса.

Blurring an image in a Python production environment using the SciPy library using a Gauss filter

Zvaigzne Alexey Yurievich

Sholem Aleichem Priamursky State University

Student

Abstract

This article describes image processing in the Python programming language using the Gauss filter prescribed in the SciPy library. The necessary additional libraries and commands for I/O, as well as for saving the final file to disk, are considered.

Keywords: Pycharm, Python, Scipy, Gaussian filter.

1 Введение

1.1 Актуальность

На текущий момент всё чаще появляется необходимость в автономной обработке фотографий посредством языка программирования без использования ресурсоемких программ, запуск которых также занимает огромное время. Запуск того же Adobe Photoshop занимает от 20 секунд до нескольких минут на разных машинах, нам не всегда требуется тонкая обработка фотографии, а их количество может исчисляться десятками или сотнями, требующими простых действий. В этой статье будет описано как можно быстро наложить эффект Гаусса, без использования стороннего софта, только лишь язык программирования Python и библиотека Scipy. Результатом данной статьи будет код, который будет без особых усилий в Pycharm обрабатывать фото с помощью эффекта Гаусса в формате JPG и сохраняя готовый результат на диск.

1.2 Обзор исследований

В статье «Python Tutorial» описывается работа с языком программирования Python[1]. В своей работе P.Virtanen и др. описывали историю развития библиотеки, её структуру и функционал, выделяя SciPy как отдельный язык программирования [2]. Андрою Фоули в своей статье описывает процесс создания рабочей области для выведения графической информации, создания графиков [3]. Общие аспекты в обработке изображений на языке программирования Python описывают в своей публикации Сараджишвили С.Э, Леонтьев В.В., Воинов Н.В. [4]. Juan Nunez-Iglesias, Stéfan van der Walt, Harriet Dashnow в своей книги показали принцип написания кода для выполнения различных задач включая эффект Гаусса, с использование библиотек SciPy, Pandas, skit-image [5].

1.3 Цель исследования

Целью данной статьи является создание эффекта размытия в PyCharm на языке программирования Python с помощью фильтра Гаусса.

1.4 Постановка задачи

Конечным итогом работы является конечное изображение с примененным эффектом Гаусса.

2 Материалы и методы

Для написания кода используются интегрированная среда разработки Pycharm и библиотека Scipy.

3 Результаты и обсуждения

Фильтр Гаусса (Gaussian filter) обычно используется в цифровом виде для обработки двумерных сигналов (изображений) с целью снижения уровня шума. Однако при ресемплинге он даёт сильное размытие изображения.

Для обработки изображения необходимо создать новый проект в PyCharm версии 2021.2.2 и Python версии 3.9.5, для этого запускаем саму программу и в окне управления проектами жмем «New project» для перехода в окно создания нового проекта (рис.1).

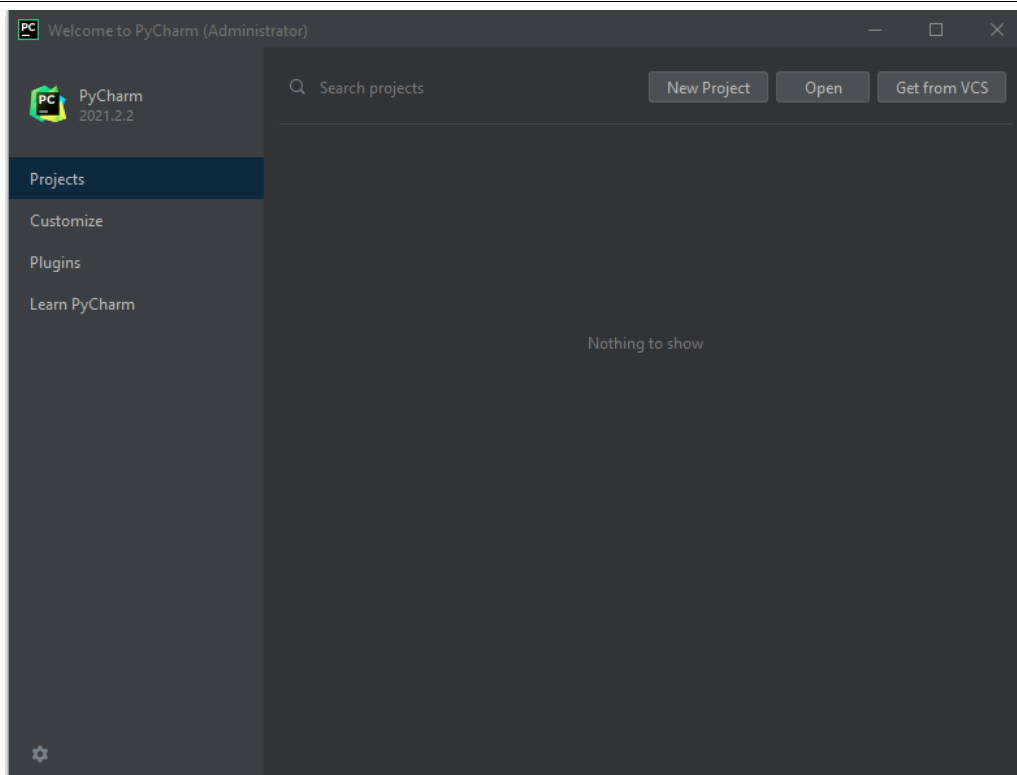


Рис.1. Главное меню Pycharm

Далее просто нажимаем кнопку «Create» для создания нового проекта (см. рис. 2).

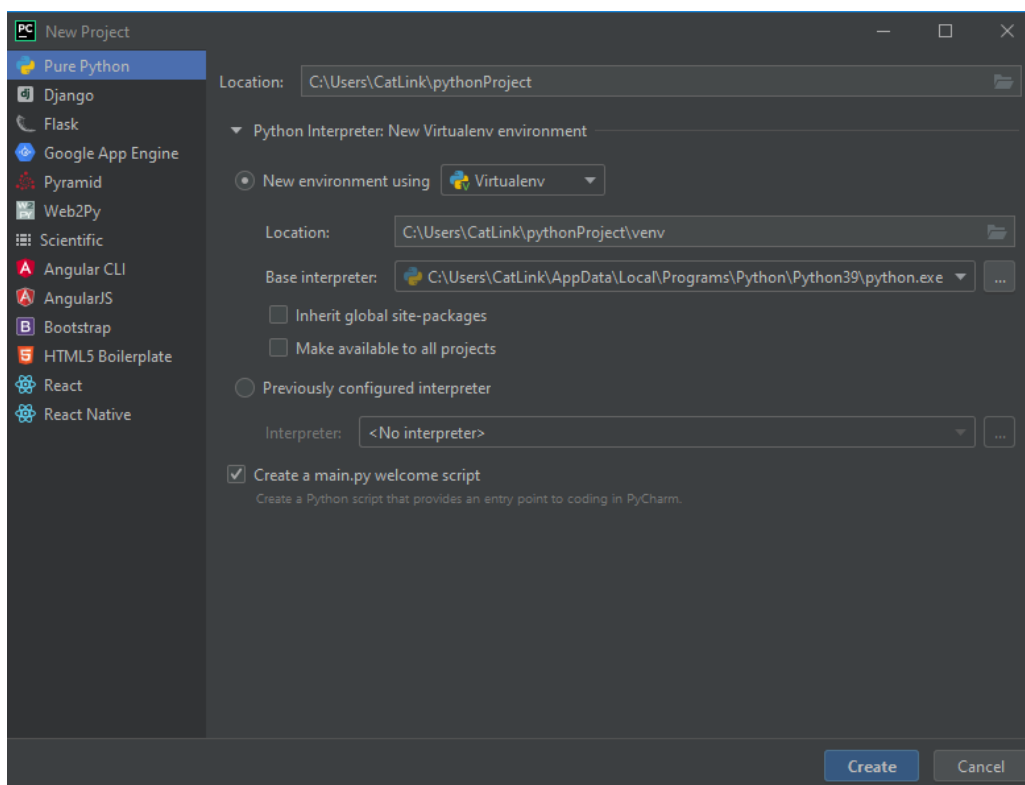


Рис.2. Создание нового проекта в PyCharm

После создания нового проекта и загрузки всех необходимых модулей, программа покажет рабочую среду с примером кода на языке Python, который нужно удалить (см. рис. 3).

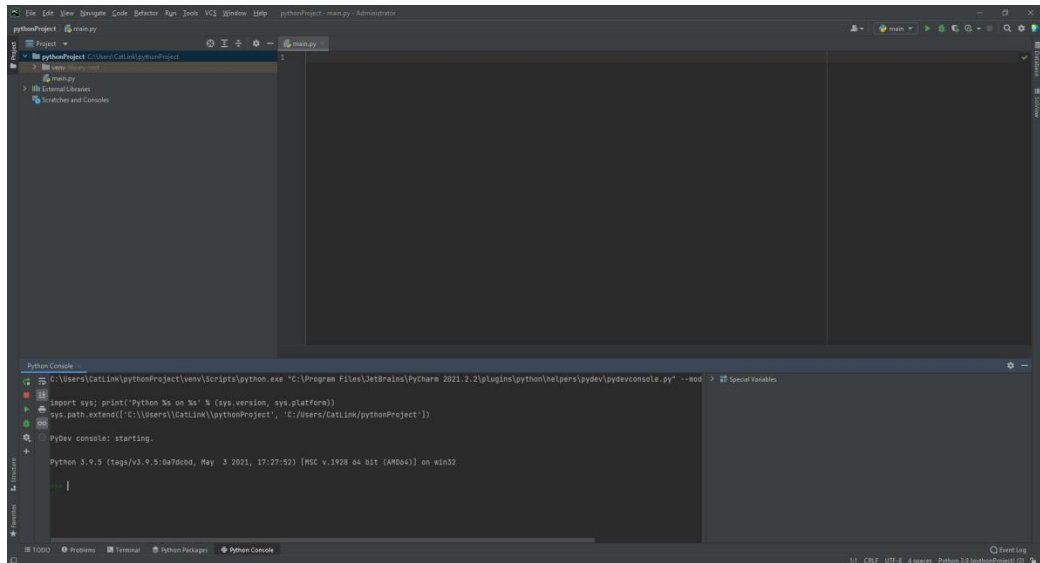


Рис.3. Вид рабочей среды в PyCharm

Для начала работы с библиотеками необходимо добавить их в наш проект, для этого открываем Terminal и прописываем туда следующие команды:

```
pip install mpimg  
pip install scipy  
pip install matplotlib
```

Попробуем обработать пример взятый с официальных источников и разберемся в работе имеющихся значений, а также заставим код обрабатывать изображение:

```
from scipy import misc  
import matplotlib.pyplot as plt  
from scipy.ndimage import gaussian_filter  
  
fig = plt.figure()  
plt.gray() # окрашивает полученное изображение в серые тона  
ax1 = fig.add_subplot(121) # левая сторона  
ax2 = fig.add_subplot(122) # правая сторона  
ascent = misc.ascent() # импортируемый пример изображения  
result = gaussian_filter(ascent, sigma=5) # размытие данного выше  
изображение по Гауссу, где sigma - сила эффекта размытия  
ax1.imshow(ascent) # вывод левого изображения  
ax2.imshow(result) # вывод правого изображения  
plt.show()
```

Данный код должен выводить в верхней правой части экрана графическое изображение в виде графика с подписью осей, где слева - данное изображение, а справа конечный результат обработки программы (рис. 4).

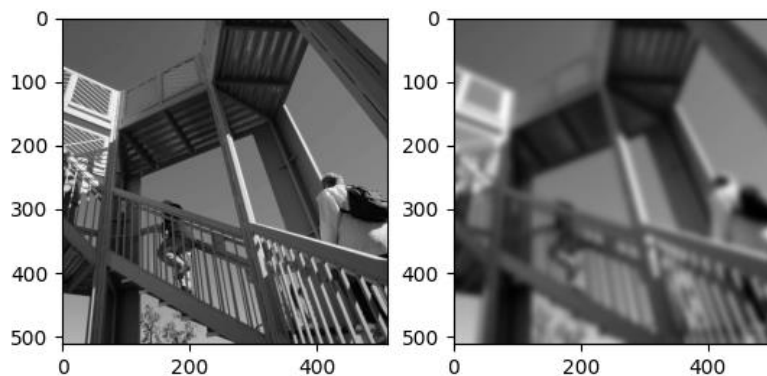


Рис. 4 Результат работы программы

Изменяя значение σ в 10 строчке можно добиться другого результата размытия (рис. 5-6).

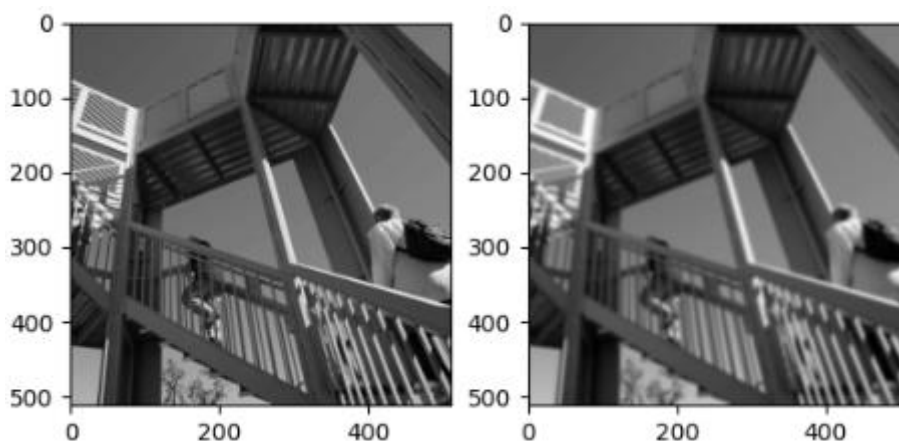


Рис. 5. Результат работы программы при значении $\sigma=2$

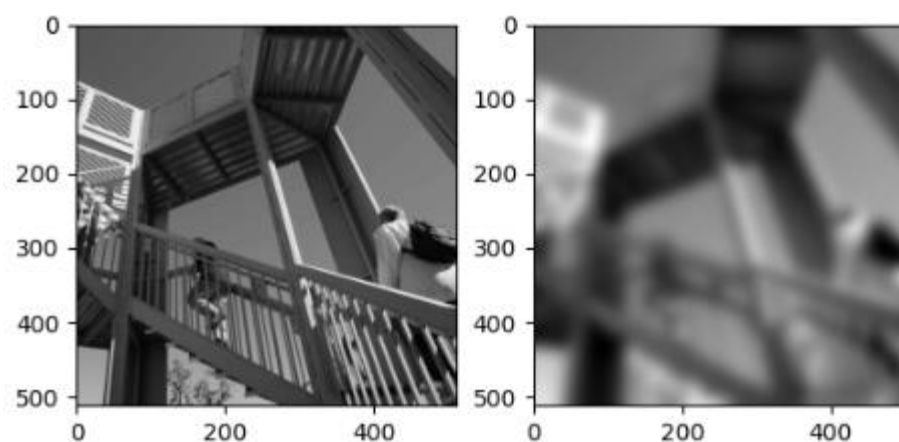


Рис. 6. Результат работы программы при значении $\sigma=10$

Как видно из полученных результатов, чем больше значение σ , тем сильнее эффект размытия, чем оно ближе к нулю, тем меньше полученное

изображение отличается от исходного. При значении σ ниже 0 конечное изображение так же не будет отличаться от исходного.

Так как хотим использовать свое изображение, а не из примера, необходимо удалить первые 2 строчки кода и после них добавить команды импорта функций, позволяющих напрямую из проводника брать файлы и работать с медиафайлами. Добавим в исходный код возможность добавить свое изображение, для этого необходимо его изменить.

В 8 строчке необходимо указать файл, который нужно обработать посредством Гаусса, где `cat.jpg` конечное имя файла, лежащее в той же папке где и сам файл проекта `main.py`. Для того чтобы избавиться от черно-белого фильтра необходимо удалить 9 строчку целиком (рис.7).

Конечный вид кода:

```
from matplotlib import pyplot as plt
from scipy.ndimage import gaussian_filter
import matplotlib.image as mpimg
```

```
fig = plt.figure()
plt.gray()
ax1 = fig.add_subplot(121)
ax2 = fig.add_subplot(122)
ascent = mpimg.imread("cat.jpg")
result = gaussian_filter(ascent, sigma=5)
ax1.imshow(ascent)
ax2.imshow(result)
plt.show()
mpimg.imsave("cat2.jpg", arr=result)
```

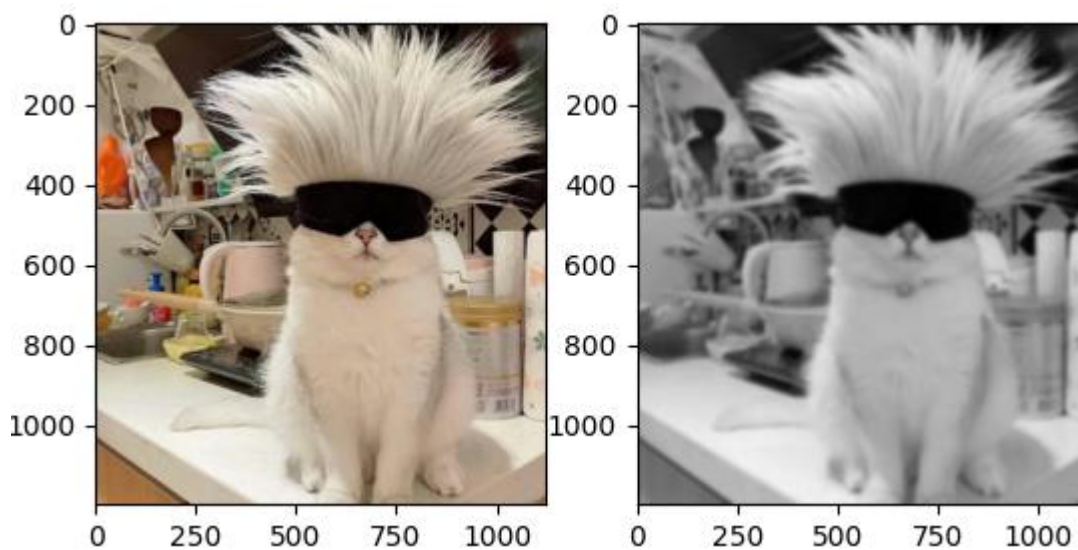


Рис. 7. Полученное изображение

Команда «`mpimg.imsave("cat2.jpg", arr=result)`» сохраняет полученное изображение, где `cat2.jpg` имя конечного файла. В результате

многочисленных попыток изменения кода было установлено, что итоговый цветовой акцент всегда будет в черно-белом формате по причине того, что размытие влияет на двумерные сигналы без учета цвета.

Библиографический список

1. Van Rossum G., Drake Jr F. L. Python tutorial. Amsterdam : Centrum voor Wiskunde en Informatica, 1995. С. 141.
2. Virtanen P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python //Nature methods. 2020. Т. 17. №. 3. С. 261-272.
3. Fowlie A., Bardsley M. H. Superplot: a graphical interface for plotting and analysing MultiNest output //The European Physical Journal Plus. 2016. Т. 131. №. 11. С. 1-10.
4. Сараджишвили С. Э., Леонтьев В. В., Воинов Н. В. Введение в обработку изображений на языке Python. СПб.: Федеральное государственное автономное образовательное учреждение высшего образования "Санкт-Петербургский политехнический университет Петра Великого", 2020. 36 с.
5. Nunez-Iglesias J., Van Der Walt S., Dashnow H. Elegant SciPy: The art of scientific python. – " O'Reilly Media, Inc.", 2017.