



Бесплатная электронная книга

УЧУСЬ

computer-vision

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#computer-  
vision

.....	1
<b>1:</b> .....	<b>2</b>
.....	2
Examples.....	4
.....	4
.....	5
.....	<b>8</b>

---

# Около

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [computer-vision](#)

It is an unofficial and free computer-vision ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official computer-vision.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

# глава 1: Начало работы с компьютерным видением

## замечания

Цифровая обработка изображений и компьютерное зрение - интересное поле, поскольку оно прекрасно расположено между математикой и информатикой. Поэтому очень полезно понять основы и применить их с помощью программирования, чтобы понять эту тему.

Цифровые изображения - это дискретизация 2 или 3 мерных сигналов. Другими словами, цифровые изображения являются выборками наборов пикселей или вокселей непрерывных доменов.

$$f : \mathbb{R}^2 \supset \Omega \rightarrow \mathbb{R}$$

Где  $f$  - цифровое изображение поверх  $\Omega$ : область прямоугольного изображения

Для простоты мы обсудим только 2-мерные цифровые изображения, такие как те, что есть в наших авариях StackOverflow.

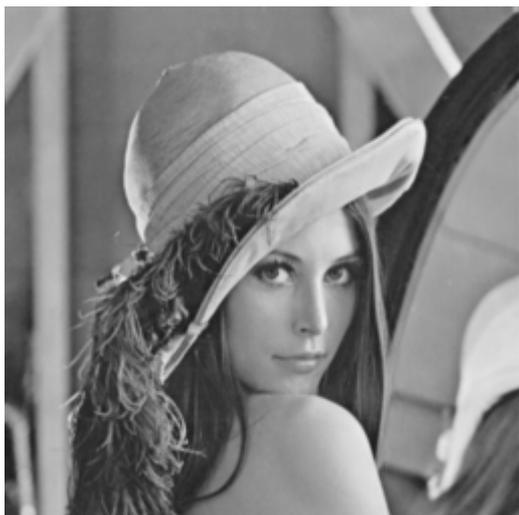
О пикселях: Быстрая заметка о значениях пикселей, прежде чем мы начнем обсуждать типы изображений. Как правило, пиксели начинаются со значения 0, что означает отсутствие света (черный), достигает 1, максимальной интенсивности (например, белого), и они представлены целыми числами.

**Двоичные изображения:** черно-белые изображения. Каждый пиксель равен 0 или 1, каждый пиксель может быть представлен бит. Они не очень популярны, поскольку они обычно используются в научных приложениях или для других операций обработки изображений в качестве маски, например.

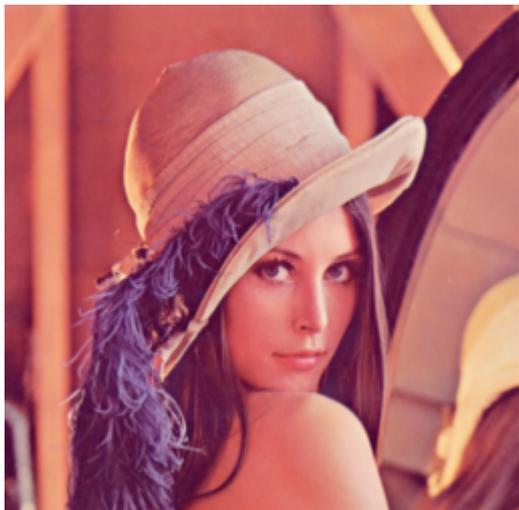


Пример двоичного изображения. (Предупреждение значения пикселя изображения этого файла не обязательно являются бинарными, это для демонстрации, также это Лена, звезда мира обработки изображений)

**Изображения в оттенках серого:** благодаря онлайн-фильтрам все хорошо знакомы с этими изображениями. Эти изображения обычно имеют один байт на пиксель, а 0 - черный, а 255 - белый, все между ними - это другой оттенок серого цвета, учитывая, что люди могут различать только 40 оттенков серого, этого диапазона достаточно для многих приложений (обратите внимание, что значения пикселей здесь отображаются от 0 до 1 до байтовых значений 0 - 255)



**Цветные изображения:** Наконец, наиболее распространенный тип цифрового изображения, цветные изображения. Здесь мы должны упомянуть о концепции каналов. Цифровые изображения, также имеют каналы, на самом деле, бинарные и полутонные изображения, описанные выше, также имеют каналы. Наиболее распространенным описанием будет модель RGB (Red-Green-Blue), в этом случае изображение имеет 3 канала (не путайте их с размерами, это все еще 2D-изображения), чтобы описать покраснение, синеватость и зеленость изображения. В этом случае каждый пиксель представляет собой триплет, значение от 0 до 255 (от красного до самого красного), 0 - 255 (от зеленого до зеленого), 0 - 255 (синий не самый синий). Для этой модели пиксель {0,0,0} черный, {255,255,255} - белый, {255,0,0} - красный, {255, 255, 0} - желтый. Тем не менее, цвет является очень широкой темой, и вы можете проверить ссылки для получения дополнительной информации.



## Гипер-спектральные изображения:

После того, как мы обсудили каналы, говорить о гиперспектральных изображениях проще. Эти изображения могут иметь сотни каналов и обычно используются в микроскопии, спутниковой визуализации и т. Д.

### Показания

1. Выборка сигнала: [https://en.wikipedia.org/wiki/Sampling\\_\(signal\\_processing\)](https://en.wikipedia.org/wiki/Sampling_(signal_processing))
2. Библийская обработка цифровых изображений: RC Gonzalez, RE Woods: Цифровая обработка изображений. Третье издание, Pearson Prentice Hall, Upper Saddle River, 2008.
3. Обзор компьютерного зрения (до глубокого обучения): Р. Селиски: Компьютерное зрение: алгоритмы и приложения. Springer, Нью-Йорк, 2010.
4. Чтобы получить представление о бинарных, полутоновых, цветных изображениях: <https://en.wikipedia.org/wiki/Grayscale>

## Examples

### Установка или настройка

Подробные инструкции по настройке или установке компьютерного зрения.

Для этой и последующих серий компьютерного зрения я буду использовать Python 2 в качестве языка программирования. Python - это общий выбор для научного сообщества, он бесплатный, у него много библиотек бесплатно и с открытым исходным кодом, и если вы новичок в программировании своего одного из самых простых для изучения и начала программирования.

Теперь настроим, если вы используете Linux, у вас, вероятно, уже есть python, просто

откройте терминал и введите «python», чтобы проверить, все ли уже работает. Другие, можете проверить [эту ссылку](#) и загрузить python 2.7.

Во-вторых, нам нужно будет установить библиотеки, которые мы будем использовать в исходном коде. Теперь обратите внимание, что эта настройка предназначена для этого примера, на более поздних этапах мы добавим разные библиотеки, и, разумеется, для разных приложений компьютерного зрения могут потребоваться определенные библиотеки, такие как OpenCV. Нам понадобится только одна библиотека, которая будет установлена в нашей системе для запуска кода. При использовании python я обычно устанавливаю зависимости, используя 'pip'. Это простой инструмент для установки модулей python, вы также можете проверить его по [этой ссылке](#)

Теперь мы готовы установить нужную нам библиотеку PyPNG. Если вы используете пипс, все, что вам нужно сделать, это

пипетка PyPNG

в терминале, если вы используете Linux / Mac, в командной строке, если вы используете Windows.

Кроме того, для этих упражнений вам нужно получить изображения, которые можно найти в ссылке github вместе с исходным кодом и ноутбуками ipython.

<https://github.com/Skorkmaz88/compvis101>

Теперь нам должно быть хорошо идти на тренировки

## Примеры

Это очень простая серия упражнений для обработки изображений и компьютерного зрения Python, предназначенная для введения этих тем с небольшими практиками. Я заранее извиняюсь за любые ошибки новобранец, он все еще находится в разработке. В этой серии я буду ограничивать цифровые изображения, которые мы можем использовать для файлов PNG для простоты, и я также обсужу тему сжатия изображения.

Пожалуйста, клонируйте репозиторий, если вы еще этого не сделали, или можете просто загрузить его [здесь](#) через Github:

```
git clone https://github.com/Skorkmaz88/compvis101
```

Есть два файла, которые вы можете использовать, один из них - tutorial0.py, а другой - readImages.ipynb, второй - это ipython-ноутбук, если вы предпочитаете использовать это. Но два файла делают то же самое.

У кода есть пояснения в комментариях, я

```

# libs
import png

# We create a greyscale image as described in our text.
# To do that simply, we create a 2D array in python.
# x and y, x being horizontal and y being vertical directions.

x = []
y = []
# Play around with these pixels values to get different grayscale images, they should be
# in range of 0 - 255.
white = 255
gray = 128
black = 0
width = 100
height = 300

# Add 100 x 100 rectangle as just white(255) valued pixels
for i in range(0, 100):
    for j in range(0,100):
        y.append(white); # Pixel (i,j) is being set to a value, rest is coding trick to nest
two lists
        x.append(y)
        y = []

# Add 100 x 100 rectangle as just mid-gray(128) valued pixels
for i in range(0, 100):
    for j in range(0,100):
        y.append(gray);
        x.append(y)
        y = []

# Add 100 x 100 rectangle as just black(0) valued pixels
for i in range(0, 100):
    for j in range(0,100):
        y.append(black);
        x.append(y)
        y = []

# output image file
f = open('out.png', 'wb')
w = png.Writer(width, height , greyscale=True, bitdepth=8)
w.write(f, x)
f.close()
# If everything went well, you should have 3 vertically aligned rectangles white, gray and
black
# Check your working folder

# PART 2
# Read a grayscale image and convert it to binary

# This time we will binarize a grayscale image, to do that we will read pixels and according
to threshold we set
# we will decide if that pixel should be white or black

# This file is originally 8 bit png image, can be found in github repository, you should use
only this type of
# images if you want to change the image.
f = open('./img/lenaG.png', 'r')

```

```

r=png.Reader(file=f)
# You will the details about the image, for now pay attention to size and bitdepth only.
img = r.read()

width = img[0]
height = img[1]
# Threshold value for binarizing images,
threshold = 128
print "Input image size is: "+ str(width)+ " pixels as width, " + str(height) + " pixels as height"

f_out = open('lenaBinary.png', 'wb')
w = png.Writer(width, height , greyscale=True, bitdepth=1)

pixels = img[2]

x = []
y = []

# Let's traverse the Lena image
for row in pixels:
    for pixel in row:
        p_value = pixel
        # Now here we binarize image in pixel level
        if p_value > threshold:
            p_value = 1
        else:
            p_value = 0

        y.append(p_value);
    x.append(y)
    y = []

w.write(f_out, x)
f_out.close()

```

Если все будет хорошо, поздравляю! Вы создали изображение с нуля и выполнили преобразование первого пикселя на существующем изображении. Проверьте свою рабочую папку, чтобы увидеть новые изображения

Прочитайте Начало работы с компьютерным зрением онлайн: <https://riptutorial.com/ru/computer-vision/topic/5710/начало-работы-с-компьютерным-зрением>

## кредиты

S. No	Главы	Contributors
1	Начало работы с компьютерным видением	<a href="#">Community</a> , <a href="#">Semih Korkmaz</a>