

 免费电子书

学习

computer-vision

Free unaffiliated eBook created from
Stack Overflow contributors.

#computer-
vision

.....	1
1:	2
.....	2
Examples.....	3
.....	3
.....	3
.....	6

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

1:

◦ ◦

23◦ ◦

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

$f\Omega$

StackOverflow◦

◦ 01◦

◦ 01◦ ◦



◦ Lena

◦ 025540010 - 255



◦ ◦ ◦ RGB - - 32D◦ 0 - 2550 - 2550 - 255◦ {0,0,0}{255,255,255}{255,0,0}{255,255,0}◦ ◦



◦ ◦

1. [https //en.wikipedia.org/wiki/Sampling_signal_processing](https://en.wikipedia.org/wiki/Sampling_signal_processing)
2. RC Gonzalez RE Woods◦ Pearson Prentice Hall Upper Saddle River 2008◦
3. R◦ Szeliski◦ 2010◦
4. [https //en.wikipedia.org/wiki/Grayscale](https://en.wikipedia.org/wiki/Grayscale)

Examples

◦

Python 2◦ Python◦

Linux python'python'◦ python 2.7◦

◦ OpenCV◦ ◦ python'pip'◦ python

PyPNG◦ pip

pip PyPNG

Linux / Mac Windows◦

ipython github◦

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

Python◦ ◦ PNG◦

Github

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

tutorial0.pyreadingImages.ipynbpython. ◦ ◦

```
# 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
```




S. No	□	Contributors
1	□□□□□□□□	Community , Semih Korkmaz