배우기

# computer-vision

#computer-vision

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.

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:

. .

2 3 . , .

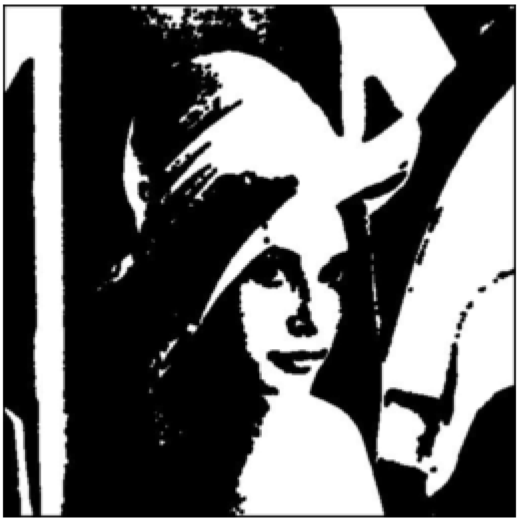
```
f : R² ⊃ Ω → R
```

f Ω : Rectangular image domain

StackOverflow 2 .

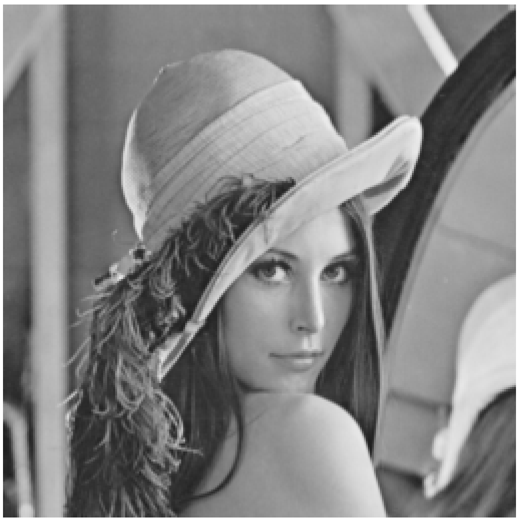
: . ( ) 0 1, (:) .

: . 0 1, . .



. (. . Lena, Image Processing world)

: . 1 1 , 0 255 . , 40 . 0 1 0 255)



: , , . . 2 . RGB (Red-Green-Blue) . , 3 (. 2D). , 0 - 255 ( ), 0 - 255 ( ), 0 - 255 ( ) . {0,0,0} {255,255,255} , {255,0,0} , {255, 255, 0} . .

- :

. , .

1. : https://en.wikipedia.org/wiki/Sampling_(signal_processing)

2. : RC Gonzalez, RE Woods : . 3 , Pearson Prentice Hall, Upper Saddle River, 2008.

3. ( ) : R. Szeliski : : . , , 2010.

4. , , : https://en.wikipedia.org/wiki/Grayscale

Examples

.

Python 2 . . , . .

. " . , python 2.7 .

, . OpenCV . . 'pip' . ,

PyPNG .

PyPNG pip

Linux / Mac Windows

ipython github .

https://github.com/Skorkmaz88/compvis101

.

Python , . . PNG .

Github .

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

. tutorial0.py readingImages.ipynb ipython . .

.

```
 # 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 shoud 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()
```

□□ □□ □ □□□□□, □□□! □□□□ □□□□ □□□ □□ □□□□□ □ □□ □□ □□ □□□ □□□□□□. □ □□□□ □□□ □□ □□□ □□□□□□.

□□□ □□□ □□ □□□□ □□ : https://riptutorial.com/ko/computer-vision/topic/5710/□□□-□□-□□□□

□□□

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