

 無料電子ブック

学習

computer-vision

Free unaffiliated eBook created from
Stack Overflow contributors.

#computer-
vision

.....	1
1:	2
.....	2
Examples.....	4
.....	4
.....	4
.....	7

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: コンピュータビジョンのまり

デジタルとコンピュータビジョンはとコンピュータサイエンスのにしくしているのでいす。したがって、をし、トピックをするためのプログラミングをすることはにちます。

デジタルは、2または3のである。いえれば、デジタルは、ドメインのピクセルまたはボクセルのサンプリングされたセットである。

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

f が Ω のデジタルである

わかりやすくするために、StackOverflowアバターのような2デジタルについてのみします。

ピクセルについてについてするに、ピクセルについてのクイックノート。のとして、は、をさ
ない0でまり、1にし、えば、にし、それらはでされる。

バイナリモノクロのみ。ピクセルは0または1のいずれかであり、ピクセルはビットですことができます。それらは、ににされるか、またはマスクなどののににされるので、あまりにられていな
い。



バイナリイメージの。このファイルのピクセルはずしもバイナリではありませんが、これはデモ
ンストレーションのためのものです、これはImage Processing worldのでもあるLenaです

グレースケールオンラインフィルタのおかげで、もがまだこれらのをよくっています。これらの
はに1ピクセルあたり1バイトで、は0、は255です。はの40しかできないので、そののすべてがグ
レーであり、このはくのアプリケーションでです。01から0255のバイトにマップされます



カラーにも、モノクロデジタルタイプのカラー。ここではチャンネルのコンセプトについて述べてみます。デジタルはまた、チャンネルを、には、のバイナリおよびグレースケールもチャンネルをします。RGB - - モデルがモノクロです。この、は、のみ、を3つのチャンネルとしないでください。この、ピクセルは0から255まで、0から255まで、0から255までの3つのです。このモデルでは、ピクセル{0,0,0}は、{255,255,255}は、{255,0,0}は、{255、255、0}はです。ただし、は、について述べてすることができます。



ハイパースペクトル

チャンネルをした、ハイパースペクトルについて述べてみます。これらにはチャンネルがあり、など述べてされます。

み

1. シグナルサンプリング [https://en.wikipedia.org/wiki/Sampling_\(signal_processing\)](https://en.wikipedia.org/wiki/Sampling_(signal_processing))
2. デジタルのRC Gonzalez、REウツズデジタル。3、ピアソンプレッティスホール、アッパーサドル、2008。
3. コンピュータビジョンレビュー—いままでR. Szeliskiコンピュータビジョンアルゴリズムとアプリケーション。Springer、New York、2010

4. バイナリ、グレースケール、カラーのをるには <https://en.wikipedia.org/wiki/Grayscale>

Examples

インストールまたはセットアップ

コンピュータビジョンのセットアップまたはインストールにするな。

このコンピュータビジョンのシリーズでは、Python 2をプログラミングとしてします。Pythonはコミュニティにとっての、オープンソースのライブラリがたくさんあります。プログラミングをぶのがめての、プログラミングをめるのがです。

しておけば、LinuxをしているのであればPythonをとっているでしょう。をいて、'python'とタイプして、すべてがすでにしているかどうかをしてください。のは、[このリンク](#)をチェックしてPython 2.7をダウンロードできます。

に、するライブラリをソースコードにインストールするがあります。ここでは、このはこののためにされています。ので、のライブラリをします。もちろん、なるコンピュータビジョンアプリケーションにはOpenCVなどのライブラリがです。コードをするには、システムにインストールするライブラリが1つだけです。Pythonをする、はに'pip'をとてをインストールします。これは、Pythonモジュールをインストールするためのなツールです、あなたはまた、[このリンク](#)をしてチェックすることができます

これで、なライブラリPyPNGをインストールするができました。あなたがうがあるすべてのパイプをしている

PyPNGをインストールする

Linux / Macをしているは、Windowsをしているはコマンドラインでします。

また、このでは、ソースコードとipythonノートブックのにあるgithubのリンクにあるイメージをするがあります。

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

、たちはのためにくべきである

これはになとコンピュータビジョンのPythonのエクササイズシリーズです。ルーキーのいがあればにしくってありますが、まだです。このシリーズでは、にするためにできるデジタルをPNGファイルにします。ここでは、のトピックについてもします。

あなたはまだっていないは、リポジトリのクローンをする、またはあなたはにそれをダウンロードすることができますしてください[このGithub](#)

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

2つのファイルがあり、そのうちの1つはtutorial0.pyで、もう1つはreadingImages.ipynbです。もう1つはipythonノートブックです。しかし、2つのファイルがじことをしています。

コードにはコメントのがあります

```
# 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()

```

すべてがうまくいくなら、おめでとう🎉からイメージを📄し、📄のイメージで📄のピクセルレベル📄を📄しました。📄フォルダを📄して📄しい📄を📄してください

オンラインでコンピュータビジョンの📄まりを📄む📄 <https://riptutorial.com/ja/computer-vision/topic/5710/コンピュータビジョンの📄まり>

クレジット

S. No		Contributors
1	コンピュータビジョンのまとめ	Community , Semih Korkmaz