



**EBook Gratuito**

# APPENDIMENTO computer-vision

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

#computer-  
vision

# Sommario

Di.....	1
<b>Capitolo 1: Iniziare con la visione del computer</b> .....	<b>2</b>
Osservazioni.....	2
Examples.....	4
Installazione o configurazione.....	4
Esempi.....	5
<b>Titoli di coda</b> .....	<b>8</b>

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)

# Capitolo 1: Iniziare con la visione del computer

## Osservazioni

L'elaborazione digitale delle immagini e la visione artificiale sono un campo interessante poiché si trova in una posizione magnifica tra matematica e informatica. Pertanto, è molto utile comprendere i fondamenti e applicarli utilizzando la programmazione per comprendere l'argomento.

Le immagini digitali sono discretizzazione di segnali bidimensionali o tridimensionali. In altre parole, le immagini digitali sono campionate in serie di pixel o voxel di domini continui.

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

Dove  $f$  è immagine digitale su  $\Omega$ : dominio immagine rettangolare

Per semplicità, discuteremo solo di immagini digitali bidimensionali, come quelle dei nostri avatar StackOverflow.

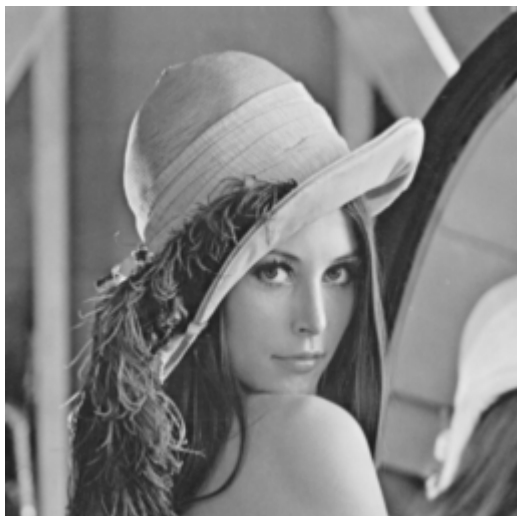
Informazioni sui pixel: una breve nota sui valori dei pixel prima di iniziare a discutere sui tipi di immagini. Come regola del pollice, i pixel iniziano con il valore 0 che non indica luce (nero), raggiunge 1, l'intensità massima (ad es. Bianco) e sono rappresentati in numeri interi.

**Immagini binarie:** solo immagini in bianco e nero. Ogni pixel è 0 o 1, ogni pixel può essere rappresentato da un bit. Non sono molto conosciuti, in quanto sono generalmente utilizzati in applicazioni scientifiche o per altre operazioni di elaborazione delle immagini come maschera per esempio.

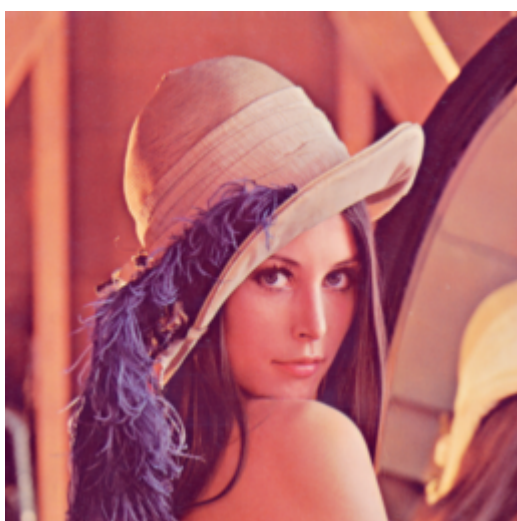


Un esempio di immagine binaria. (Attenzione i valori dei pixel dell'immagine di questo file non sono necessariamente binari, questo è per la dimostrazione, anche questa è Lena, la stella del mondo di Image Processing)

**Immagini in scala di grigi:** grazie ai filtri online, tutti sanno ancora molto bene queste immagini. Queste immagini sono generalmente un byte per pixel, con 0 che è nero e 255 che è bianco, tutto tra un tono diverso è un grigio, dato che gli umani possono distinguere solo 40 sfumature di grigio, questo intervallo è sufficiente per molte applicazioni (Si noti che i valori di pixel qui sono mappati da 0 a 1 a valori di byte 0 - 255)



**Immagini a colori:** infine, tipo di immagine digitale più comune, immagini a colori. Dobbiamo menzionare il concetto di canali qui. Le immagini digitali hanno anche canali, in realtà le immagini binarie e in scala di grigi descritte sopra hanno anche canali. La descrizione più comune sarebbe il modello RGB (rosso-verde-blu), in tal caso l'immagine ha 3 canali (non confonderlo con le dimensioni, queste sono ancora immagini 2D) per descrivere il rosso, il blu e il verde dell'immagine. In questo caso, ciascun pixel è una terzina, un valore compreso tra 0 e 255 (nessun rosso per la maggior parte rosso), 0 - 255 (nessun verde per la maggior parte verde), 0 - 255 (nessun blu per la maggior parte blu). Per questo modello, pixel  $\{0,0,0\}$  è nero,  $\{255,255,255\}$  è bianco,  $\{255,0,0\}$  è rosso,  $\{255, 255, 0\}$  è giallo. Tuttavia, il colore è un argomento molto ampio e puoi controllare i riferimenti per ulteriori informazioni.



**Immagini iper-spettrali:**

Dopo aver discusso i canali, parlare di immagini iper-spettrali è più facile. Queste immagini possono avere centinaia di canali e comunemente usati in microscopia, immagini satellitari ecc.

letture

1. Campionamento del segnale: [https://en.wikipedia.org/wiki/Sampling\\_\(signal\\_processing\)](https://en.wikipedia.org/wiki/Sampling_(signal_processing))
2. Bibbia di elaborazione digitale delle immagini: RC Gonzalez, RE Woods: Digital Image Processing. Terza edizione, Pearson Prentice Hall, Upper Saddle River, 2008.
3. Revisione computer vision (fino al Deep Learning): R. Szeliski: Computer Vision: Algorithms and Applications. Springer, New York, 2010.
4. Per avere una comprensione delle immagini a colori binari, in scala di grigi, a colori: <https://en.wikipedia.org/wiki/Grayscale>

## Examples

### Installazione o configurazione

Istruzioni dettagliate su come installare o installare la visione del computer.

Per questo e per le seguenti serie di computer vision, userò Python 2 come linguaggio di programmazione. Python è una scelta comune per la comunità scientifica, è gratuito, ha un sacco di librerie gratuite e open-source, e se sei nuovo alla programmazione è uno dei più semplici da imparare e iniziare a programmare.

Ora configurazione, se usi Linux probabilmente hai già python, apri il terminale e digita "python" per controllare se tutto funziona già. Altri, possono controllare [questo link](#) e scaricare python 2.7.

Secondo, avremo bisogno di installare le librerie che useremo nel codice sorgente. Ora, una nota qui, questa configurazione è progettata per questo esempio, in fasi successive, aggiungeremo diverse librerie e, ovviamente, diverse applicazioni di visione artificiale potrebbero richiedere librerie specifiche come OpenCV. Avremo bisogno di una sola libreria da installare nel nostro sistema per eseguire il codice. Quando utilizzo python, generalmente installo le dipendenze usando 'pip'. È un semplice strumento per installare i moduli python, puoi anche controllarlo tramite [questo link](#)

Ora siamo pronti per installare la libreria di cui abbiamo bisogno, PyPNG. Se stai usando pip tutto quello che devi fare è

```
pip installa PyPNG
```

in terminale se si utilizza Linux / Mac, in riga di comando se si utilizza Windows.

Inoltre, per questo esercizio, è necessario ottenere immagini che possono essere trovate nel link github accanto al codice sorgente e ai taccuini ipython.

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

Ora, dovremmo essere bravi a fare esercizio fisico

## Esempi

Questa è una serie di esercizi di Python per l'elaborazione di immagini e la visione artificiale, progettata per introdurre questi argomenti con poche pratiche. Mi dispiace in anticipo per eventuali errori del debuttante, è ancora in fase di sviluppo. In questa serie, limiterò le immagini digitali che possiamo usare ai file PNG per semplicità, che discuterò anche del tema della compressione delle immagini.

Per favore clonare il repository se non lo hai già fatto, oppure puoi semplicemente scaricarlo [qui](#) tramite Github:

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

Ci sono due file che puoi usare, uno di loro è tutorial0.py e altro è readingImages.ipynb, il secondo è un notebook ipython se preferisci usare quello che puoi fare. Ma due file stanno facendo la stessa cosa.

Il codice ha spiegazioni nei commenti, io sono

```
# 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()

```

Se tutto ha funzionato bene, congratulazioni! Hai creato un'immagine da zero ed eseguito la



prima trasformazione a livello di pixel su un'immagine esistente. Controlla la tua cartella di lavoro per vedere le nuove immagini

Leggi Iniziare con la visione del computer online: <https://riptutorial.com/it/computer-vision/topic/5710/iniziare-con-la-visione-del-computer>

## Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con la visione del computer	<a href="#">Community</a> , <a href="#">Semih Korkmaz</a>