

# Multimedia

## Homework 1

Alessandro Trigo

30 Aprile 2024

## 1 Obiettivo

Descrivi decentemente l'obiettivo

## 2 Codice sorgente

Fai introduzione

### Task 1

La prima richiesta dell'homework era divisa in due macro parti, la prima era quella di selezionare e mostrare un'immagine mentre la seconda richiesta chiedeva di calcolare l'entropia dell'immagine.

L'immagine scelta è un'immagine di *Spider-Man* a colori, di conseguenza è necessario estrarne la luminanza per farla diventare in binario e nero. Dopo aver caricato l'immagine a colori con l'opportuna funzione `imread` del pacchetto `matplotlib.image` è necessario usare la funzione `cvtColor` del pacchetto `cv2` di `opencv`. Il seguente script, dopo aver eseguito le suddette operazioni si occupa di mostrare le due immagini a schermo attraverso una `subplot`.

```
1 # Prepare to load the image
2 img_file_name = "spiderman.jpg"
3 current_dir = os.getcwd()
4
5 # construct the full path to the image file
6 img_path = os.path.join(current_dir, "multimedia", "hw-1",
7                           "script", "imgs", img_file_name)
8
9 # loads the colored image
10 img = mpimg.imread(img_path)
11
12 # extracts the luminance
13 gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
14
15 # creates a figure with two subplots
16 fig, axs = plt.subplots(1, 2, figsize=(12, 6))
17
18 # displays the colored image in the first subplot
19 axs[0].imshow(img, cmap='gray')
20 axs[0].set_title('Colored image')
21 axs[0].axis('off')
22
23 # displays the grayscale image in the second subplot
24 axs[1].imshow(gray_img, cmap='gray')
25 axs[1].set_title('Grayscale image')
26 axs[1].axis('off')
27 plt.show()
```

Dopo aver eseguito lo script soprastante si può notare che la luminanza dell'immagine è stata estratta con successo (figura 1).



Figura 1: Estrazione della luminanza da una immagine a colori.

In secondo luogo è necessario calcolare l'entropia dell'immagine in bianco e nero. L'entropia di di un'immagine, ma in genere di qualsiasi tipo di sorgente (o meglio, variabile aleatoria  $X$ ), è un numero che indica l'**informazione media** ed è definita come segue:

$$H(X) = E[I(X)] = \sum_{i=1}^M p_i \log_2 \left( \frac{1}{p_i} \right)$$

Dove con  $M$  si indica il numero di elementi nell'insieme  $X$  e con  $p_i$  si indica la probabilità di accadere dell'elemento  $i$ -esimo della variabile aleatoria  $X$ . Questa formula si riassume nel seguente script, dove l'immagine viene trasportata e convertita in un vettore di pixel monodimensionale. In secondo luogo attraverso la funzione `numpy.histogram` si contano il numero di occorrenze per ogni valore di pixel. Infine, per calcolare la probabilità, si divide il numero di occorrenze per il numero totale di occorrenze, escludendo eventuali valori diversi da zero.

```

1  # flatten the transposed matrix to read pixels row by row
2  rasterScan = np.transpose(gray_img).flatten()
3
4  # count the occurrences of each pixel value
5  occurrences = np.histogram(rasterScan, bins=range(256))[0]
6
7  # calculate the relative frequencies

```

```

8 | p = occurrences / np.sum(occurrences)
9 |
10 | # remove zero-values of probability
11 | p = p[p > 0]
12 |
13 | # compute and display the entropy
14 | HX = - np.sum(p * np.log2(p))
15 | print(f"\nThe entropy of {img_file_name} is {HX:.3f} bpp\n")

```

Dopo aver eseguito lo script si ottiene che l'entropia dell'immagine scelta è di circa **7.530 bpp**.

## Task 2

## 3 Conclusioni