



COMPUTER VISION

Tesina

Realizzazione delle esercitazioni di OpenCV del corso di
Computer Vision con Jupyter Notebook.



SOMMARIO

Introduzione.....	3
Jupyter Notebook.....	3
Anaconda	3
Repository	4
I notebook.....	4
i pacchetti utilizzati.....	6
Gestione flusso video webcam	6
Le Esercitazioni	7
Riferimenti	9



INTRODUZIONE

Lo scopo di questa tesina è quello di realizzare i laboratori di OpenCV in Python del corso di Computer Vision dell'anno accademico 2018/2019 con Jupyter Notebook. Ogni esercitazione è stata svolta in un unico notebook, per un totale di cinque notebook.

JUPYTER NOTEBOOK

Il notebook Jupyter, nato dalla generalizzazione del precedente IPython, ambiente di programmazione interattivo, è un'applicazione web gratuita e open-source che permette di creare e condividere documenti che contengano codice, equazioni, testi ed immagini incorporate. Con Jupyter è possibile unire le fasi di raccolta dei dati, scrittura del codice, visualizzazione di grafici e tabelle, nonché la possibilità di rendere il codice condivisibile sulla piattaforma GitHub.

In particolare, il codice è di volta in volta modificabile ed eseguibile in tempo reale. Quanto realizzato può essere successivamente esportato in modo automatico in formato HTML, PDF e LaTeX. [1]

ANACONDA

Per sviluppare i notebook ho utilizzato Anaconda, una distribuzione open-source di Python e R. Questa distribuzione fornisce diverse applicazioni, tra cui Jupyter Notebook. [2]



REPOSITORY

È possibile trovare i notebook delle esercitazioni di OpenCV del corso di Computer Vision 2018/2019 attraverso questo link:

https://github.com/Gioxor/Esercitazioni_OpenCV

Durante il corso di Computer Vision si è iniziato a trattare di OpenCV dalla terza esercitazione, per questo la numerazione dei notebook inizia dal tre.

Ho collegato il repository su Binder per poter interagire con il codice, ma gli script per mostrare il flusso webcam non funzionano.

I NOTEBOOK

Una particolarità dei notebook è quella di poter mettere insieme del testo formattato (celle di markdown) e del codice insieme in modo da rendere più intuitivo il codice e ben ordinato. Ho cercato di dividere le varie sezioni del codice e scrivere una piccola descrizione per rendere più leggibile tutto il flusso. All'inizio di ogni notebook ho diviso la sezione dove c'è l'importazione dei pacchetti da importare:

Pacchetti da importare

```
In [1]: %matplotlib inline
        %pylab inline
        import cv2 as cv
        import matplotlib.pyplot as plt
        import numpy as np
        import IPython.display
        import time
        import PIL.Image
        from io import BytesIO
        from IPython.display import clear_output
```

Così da non dover re-importare ogni volta per ogni esercizio dell'esercitazione diverso i pacchetti necessari. Lo stesso per quanto riguarda le parti codice in comune con i vari esercizi di ogni laboratorio. Ho cercato di dividere quando potevo anche i vari esercizi con i vari output in modo da apprezzare meglio le differenze rispetto ai precedenti punti:

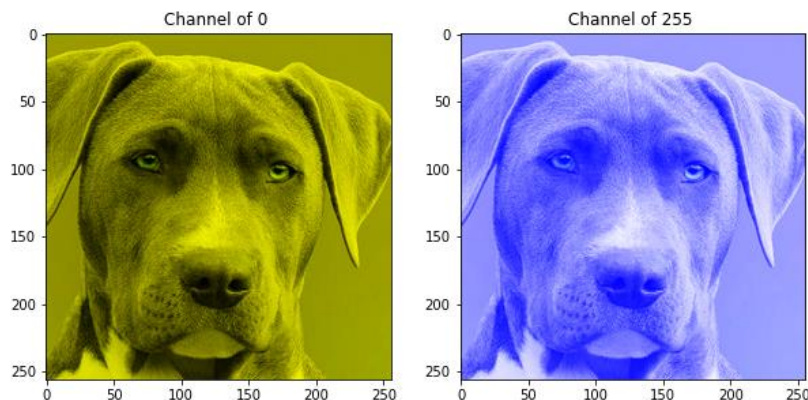
Ultimi due punti:

4. Rimpiazzare ora un canale con una matrice di zeri. Qual è l'effetto macroscopico?
5. Rimpiazzare, infine, lo stesso canale di prima con una matrice i cui valori sono a 255. Qual è l'effetto macroscopico?

```
In [21]: imgy = np.copy(imgrgb)
imgy[:, :, 2] = 0
plt.subplot(1,2,1)
plt.imshow(imgy)
plt.title("Channel of 0 ")

imgz = np.copy(imgrgb)
imgz[:, :, 2] = 255
plt.subplot(1,2,2)
plt.imshow(imgz)
plt.title("Channel of 255 ")

plt.show()
```





I PACCHETTI UTILIZZATI

Matplotlib – Per la visualizzazione dei grafici e delle immagini.

OpenCV – Per tutte le funzioni di Computer Vision.

Numpy – Usata per le funzioni matematiche.

Ipython.display – API che fornisce delle funzioni per la visualizzazione di immagini e video. Usata per gestire il flusso video della webcam.

Time – per calcolare i tempi e frame rate

PIL.Image – per convertire i frame

GESTIONE FLUSSO VIDEO WEBCAM

Lo script visto a lezione su Pycharm non funzionava su Jupyter Notebook, così ho cercato per il web qualche altro metodo che potesse andare bene, trovando uno script simile a quello visto a lezione ma dove usava il modulo IPython.display: [3]

Tuttavia, con questa soluzione il frame rate del flusso video risultava basso, circa 4-5 frame al secondo. Così continuando la ricerca ho trovato uno script di un ragazzo che, combinando altri esempi, ha trovato un modo per aumentare di più di cinque volte il frame rate. [4]

Il problema che con i metodi visti in precedenza ogni frame veniva convertito in PNG e risulta abbastanza lenta come conversione, mentre se si converte in JPEG, il frame rate aumenta considerevolmente fino a più di 35 FPS.

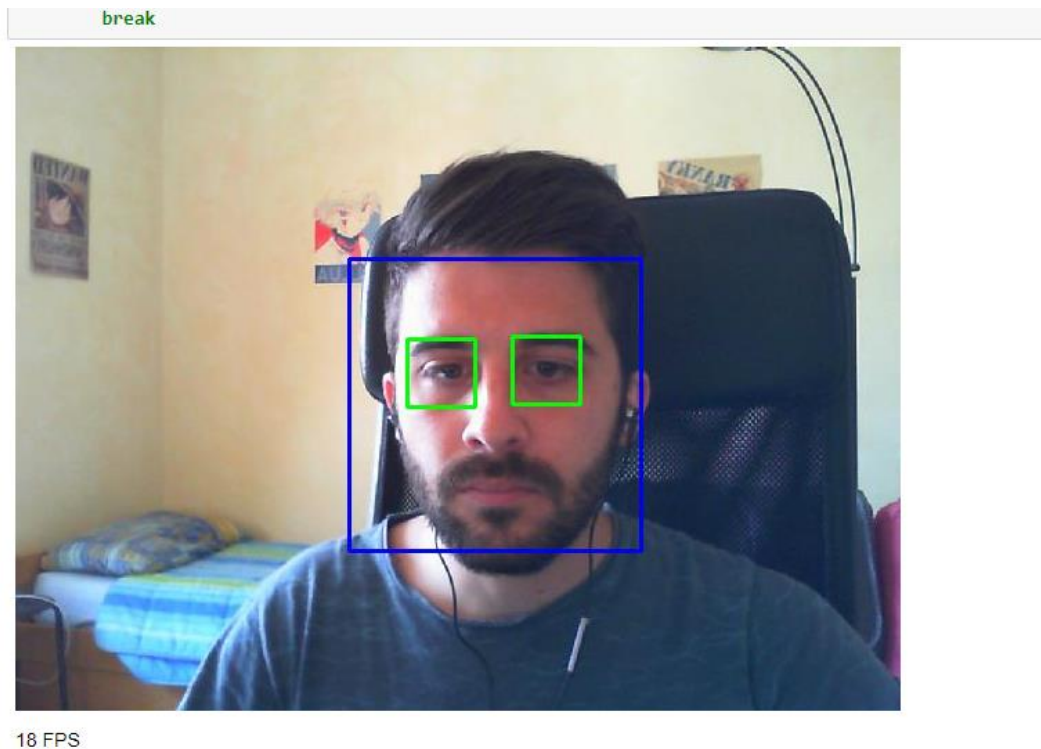
LE ESERCITAZIONI

Ho adattato gli script per il flusso video su Jupyter nelle esercitazioni che ho svolto durante l'anno, avendo tutto sommato un buon risultato.



34 FPS

In questo screen del flusso video dell'esercitazione 3 possiamo vedere che arriviamo oltre i 30 FPS.



In questo screen del flusso video dell'esercitazione 6 sulla Face Detection possiamo vedere che il frame rate scende rispetto a prima, causato sicuramente dal tracciamento delle linee per il volto e occhi.

Nelle varie esercitazioni dove viene richiesto di mostrare il video da webcam, si può visualizzare un flusso video alla volta. Come indicato nei notebook quando si vuole passare da un flusso ad un altro bisogna stoppare il kernel o semplicemente schiacciare due volte la `i` da tastiera per chiudere il flusso video corrente.

Per il resto degli esercizi non ho trovato qualche particolare problematica rispetto a come quando sono eseguiti su Pycharm.



RIFERIMENTI

[1] Jupyter Project:

https://en.wikipedia.org/wiki/Project_Jupyter

[2] Anaconda:

[https://en.wikipedia.org/wiki/Anaconda_\(Python_distribution\)](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution))

[3] Displaying Video in IPython Notebook:

<https://github.com/bikz05/ipython-notebooks/blob/master/computer-vision/displaying-video-in-ipython-notebook.ipynb>

[4] Displaying webcam video in IPython notebook at (relatively) high framerate:

https://github.com/ktaletsk/NCCV/blob/master/1_OpenCV_Jupyter_webcam.ipynb