Visione artificiale

Alessandro Pioggia, Luca Rengo, Federico Brunelli, Leon Baiocchi

20 febbraio 2022

Indice

1 Intro **2**
- 1.1 Che cosa vede un computer? . . . 2
- 1.2 Perchè è difficile? . . . 2
- 1.3 Perchè è importante? . . . 2
- 1.4 Possibili applicazioni - Panoramica generale . . . 3
- 1.5 Strumenti di lavoro . . . 3
  - 1.5.1 Python . . . 3
  - 1.5.2 OpenCV . . . 3
  - 1.5.3 NumPy . . . 3
  - 1.5.4 OpenCV-Python . . . 3
  - 1.5.5 Jupyter Notebook . . . 4

2 Python **5**
- 2.1 Descrizione . . . 5
- 2.2 Fun fact . . . 5
- 2.3 Zucchero sintattico . . . 5
  - 2.3.1 Indentazione . . . 5
  - 2.3.2 Variabili . . . 6
  - 2.3.3 Tipi di dati . . . 6
  - 2.3.4 Oggetti, valori, tipi . . . 7
  - 2.3.5 Alcuni oggetti predefiniti . . . 8

3 NumPy **9**

4 Immagini **10**

5 Calibrazione **11**

6 Filtri **12**

7 Analisi **13**

8 Movimento **14**

9 Riconoscimento **15**

Capitolo 1

Intro

Idea = cercare di dotare le macchine della vista, si cerca di insegnare ai computer come interpretare le informazioni presenti in immagini e video.

1.1 Che cosa vede un computer?

Un pc vede una matrice di pixel ed ogni pixel è rappresentato da un numero. Questo è il punto di partenza che ci permetterà di capire che cosa c'è nell'immagine.

1.2 Perchè è difficile?

Punti di vista diversi, occlusione (un oggetto ne copre un altro), distorsione, movimento (a volte può tornare utile), variazioni-intra-classe, sfondo complesso (sfondo non omogeneo, pieno di informazioni che non ci interessano).

1.3 Perchè è importante?

Ha molteplici applicazioni, quali

- Sicurezza stradale;
- Salute;
- Prevenzione del crimine;
- Protezione civile;
- Divertimento;
- Domotica.

1.4 Possibili applicazioni - Panoramica generale

- Misurazione, conteggio, qualità : Hanno in comune il fatto di volere misurare la qualità di un qualche oggetto;
  - Misurazione precisa non a contatto : In campo industriale pensiamo ad una catena di montaggio in cui passano prodotti che hanno bisogno di un controllo qualità, che viene svolto da una macchina che sfrutta tecniche di visione artificiale (Controllo delle quantità di liquido in una bottiglietta).
  - Conteggio di veicoli e persone in una piazza, in generale stime dimensionali;
- Elaborazione immagini avanzata
  - Miglioramento immagini;
  - Immagini mediche;
  - Segmentazione di aree agricoli, utili per individuare le immagini aeree e satellitari.
- Riconoscimento
  - Classificazione/individuazione di oggetti : Tag inseriti all'interno dell'immagine, Ricerca per somiglianza di immagini, riconoscere segnali stradali;
  - Riconoscimento persone : punta non solo a classificare, bensì riconoscere univocamente una persona.
- Movimento
  - Video sorveglianza : individuare un ladro, ritrovare uno zaino perso;
  - Navigazione e guida autonoma.

1.5 Strumenti di lavoro

1.5.1 Python

Standard di fatto per lo sviluppo di software scientifico, tra le altre cose anche di visione artificiale. Linguaggio di alto livello, che ha un certo livello di astrazione, in generale con poche linee, se utilizzate in modo corretto possiamo esprimere concetti piuttosto complessi.

1.5.2 OpenCV

OpenCV è la libreria standard per quanto riguarda la visione artificiale, è sviluppata in C++, però è utilizzata anche da altri linguaggi (quali python).

1.5.3 NumPy

La Lazzaro ce lo ha già spiegato.

1.5.4 OpenCV-Python

Una volta richiamato OpenCV da python, tutte le strutture dati (vettori, punti nelle immagini, ecc.) sono array numPy.

1.5.5 Jupyter Notebook

Ambiente web, in cui vengono scritti piccoli pezzi di codice, immagini e altro al fine di creare dei documenti interattivi.

```
//Hello.java
import javax.swing.JApplet;
import java.awt.Graphics;

public class Hello extends JApplet {
   public void paintComponent(Graphics g) {
      g.drawString("Hello, world!", 65, 95);
   }
}
```

Capitolo 2

Python

Python è il Leonardo Di Caprio dei linguaggi di programmazione. Ma perchè?

2.1 Descrizione

- Python è un linguaggio ad alto livello, general purpose, può essere utilizzato per qualunque tipo di applicazione;
- il codice deve essere facile da leggere, è una prerogativa;
- occorrono poche linee di codice per sviluppare anche concetti complessi;
- supporta programmazione : oop, imperativa e funzionale;

2.2 Fun fact

L'autore, Guido Van Rossum, sceglie il nome python perchè amante del gruppo comico Monty Python, attivo negli anni 70-80.

2.3 Zucchero sintattico

2.3.1 Indentazione

```
//Non si puo' cambiare indentazione in un blocco
print("Salve Cesena")
   print("Salve di nuovo!")

//L'indentazione definisce i blocchi di codice
if 42 < 0:
   print("ciao")
print("ciao ma in corsivo")

//andata a capo
istruzione_molto_lunga = 2 + 5 \
                  + 6 + 7

//Piu' istruzioni in una singola riga
print("a"); print("b")
```

2.3.2 Variabili

```
//Le variabili in python non si dichiarano, vengono create automaticamente al
    momento dell'inizializzazione sono case sensitive, iniziano con una lettera
    o underscore e i caratteri ammessi sono le Lettere, i numeri e l'underscore
    (codifica UNICODE)

x = 4
y = "Visione artificiale"
print(x, y)

//Assegnamento dello stesso valore a piu' variabili

x = y = z = 90

//Variabili globali e locali

a = "cool" //Variabile globale

def function():
   a = "bad" //Variabile locale alla funzione

//Se aggiungo global la variabile globale a, definita inizialmente verra'
    modificata
def function():
   global a = "bad"
```

2.3.3 Tipi di dati

Python è tipizzato, solo che, una volta inizializzata una variabile non abbiamo un modo per specializzare un tipo, una volta assegnato un valore viene definito automaticamente (se scrivo tra "", l'oggetto diventerà una stringa). La variabile non ha tipo, è l'oggetto creato che ha un tipo. Il tipo non è legato alla variabile ma all'oggetto!!! I tipi predefiniti di python sono:

- Testo : str
- Numeri : int, float, complex
- Sequenze : list, tuple, range
- Dizionari : dict
- Insiemi : set, frozenset
- booleani : bool
- binari : bytes, bytearray, memoryview

```
x = "ciao" //stringa
x = 20 //int
x = 20.5 //float
x = 1j //numero complesso
x = ["ciao", "come", "stai"] //list
x = ("ciao", "come", "stai") //tuple
x = range(5) //range
x = {"nome" : "Va", "codice" : 17633} //dict
x = {"ciao", "amico"} //set
x = frozenset(x) //frozenset, ovvero un set non modificabile
x = True //bool
x = b"ABCD" //bytes
x = bytearray(5) //bytearray
x = memoryview(bytes(5)) //memoryview
//con type viene stampato il tipo della variabile
type(x)
```

2.3.4 Oggetti, valori, tipi

Qualunque dato in python è un oggetto, ogni oggetto è caratterizzato da : un tipo, un'identità ed un valore. Inoltre:

- il tipo determina le operazioni che l'oggetto supporta;
- l'identità non cambia mai;
- Il tipo determina se un oggetto è mutabile o meno;
- le variabili sono riferimenti ad oggetti.

Python in sè per sè è lento, perchè però ha questo successo? Perchè in un programma sono poche le componenti che devono essere efficienti e sono spesso algoritmi. Gli algoritmi verranno presi da NumPy o altre librerie, che hanno la loro implementazione in C o C++, garantendo massima efficienza.

```
//L'assegnamento crea un oggetto in memoria di tipo Float, a cui la variabile fa
   riferimento
answer = 3.4
print('Type:', type(answer))
print('Identity', id(answer))
print('Value', answer)

//L'assegnamento copia il riferimento dell'oggetto nella nuova variabile
spam = answer
print(spam.equals(answer)) //ritornera' True

 //L'istruzione seguente NON modifica l'oggetto ma crea un nuovo oggetto
    contenente il risultato e ne assegna il riferimento alla variabile: si puo'
    osservare infatti che l'identita' dell'oggetto associato a 'answer' e'
    cambiata, mentre l'identita' di spam e' la stessa
 answer *= 2
 print('Identita nuovo oggetto:', id(answer))
 print('Identita vecchio oggetto :', id(spam))
```

2.3.5 Alcuni oggetti predefiniti

Il loro tipo non fa parte dei tipi di base visti prima, hanno un valore particolare.

- None : None è l'unico oggetto della classe NoneType, è simile a null ma c'è una differenza sostanziale: quando una variabile è a null significa che non ha un riferimento ad un oggetto mentre il None è un oggetto che esiste in memoria (ne esiste solo uno). Ponendo una variabile = None, la variabile punta a quell'oggetto.
- Ellipsis : Sono i puntini di sospensione, è utile in NumPy
- NotImplemented : Metodi numerici e di confronto possono restituire questo valore se non implementano l'operazione per determinati operandi.

```
//Sintassi per oggetto Ellipsis
x = ...
y = Ellipsis
print(x, y)
//Confronto fra stringa e numero, restituisce notImplemented
ret = 'testo'._eq_(42)
```

2.4 Numeri

I numeri possono essere rappresentati attraverso int, float e Complex, possono essere convertiti utilizzando i relativi costruttori (es: float()).

2.5 Stringhe

Tutti i caratteri messi dentro ad una stringa con triplici apici vengono considerati, non succede la stessa cosa con apici doppi o singoli. Non esiste il tipo carattere! E' possibile accedere alle stringhe come se fossero array(liste python).

Capitolo 3

NumPy

```
//Hello.java
import javax.swing.JApplet;
import java.awt.Graphics;

public class Hello extends JApplet {
   public void paintComponent(Graphics g) {
      g.drawString("Hello, world!", 65, 95);
   }
}
```

Capitolo 4

Immagini

```
//Hello.java
import javax.swing.JApplet;
import java.awt.Graphics;

public class Hello extends JApplet {
   public void paintComponent(Graphics g) {
      g.drawString("Hello, world!", 65, 95);
   }
}
```

Capitolo 5

Calibrazione

```
//Hello.java
import javax.swing.JApplet;
import java.awt.Graphics;

public class Hello extends JApplet {
   public void paintComponent(Graphics g) {
      g.drawString("Hello, world!", 65, 95);
   }
}
```

Capitolo 6

Filtri

```
//Hello.java
import javax.swing.JApplet;
import java.awt.Graphics;

public class Hello extends JApplet {
   public void paintComponent(Graphics g) {
      g.drawString("Hello, world!", 65, 95);
   }
}
```

Capitolo 7

Analisi

```
//Hello.java
import javax.swing.JApplet;
import java.awt.Graphics;

public class Hello extends JApplet {
   public void paintComponent(Graphics g) {
      g.drawString("Hello, world!", 65, 95);
   }
}
```

Capitolo 8

Movimento

```
//Hello.java
import javax.swing.JApplet;
import java.awt.Graphics;

public class Hello extends JApplet {
   public void paintComponent(Graphics g) {
      g.drawString("Hello, world!", 65, 95);
   }
}
```

Capitolo 9

Riconoscimento

```
//Hello.java
import javax.swing.JApplet;
import java.awt.Graphics;

public class Hello extends JApplet {
   public void paintComponent(Graphics g) {
      g.drawString("Hello, world!", 65, 95);
   }
}
```