



Dipartimento Politecnico di Ingegneria e Architettura (DPIA)
Corso di Visione Artificiale

Progetto: 3D Printer Defects Classification **Descrizione del software**

Studente: Lorenzo Zaccomer

Data: 26 aprile 2022

Introduzione

Si andranno ad esaminare i software installati, questo per fornire un riferimento a chi volesse testare sulla propria macchina questo progetto.

Indice

1	Software	1
1.1	Introduzione	1
1.2	Python	1
1.3	Framework utilizzati	1
1.4	Installazioni	1
1.5	Online vs Offline	2

1 Software

1.1 Introduzione

Si vanno a specificare i software utilizzati e le relative versioni installate, più una riflessione insieme ad un suggerimento.

1.2 Python

Ho utilizzato due versioni di Python differenti, avendo due macchine differenti, una è la versione *3.9.0* a 64 bit, mentre l'altra è la *3.8.10* a 64 bit.

1.3 Framework utilizzati

I framework da installare sono i seguenti:

- pytorch *1.10.0*
- torchvision *0.11.1*
- matplotlib *3.4.3*
- numpy *1.19.4*
- OpenCV *4.5.3.56* (opzionale)

Si specifica che torchvision è una libreria del progetto pytorch.

Ho utilizzato PyTorch perché un framework open-source pensato per la *computer vision*, cadendo a pennello con applicazioni come questa, inoltre è presente un'ampia documentazione ben dettagliata con una moltitudine di esempi applicativi.

OpenCV è un framework anch'esso applicabile per la *computer vision*, più per il rilevamento di immagini da una fotocamera piuttosto che elaborazioni, è opzionale perché avevo valutato di implementare la cattura vera e propria di un frame dalla videocamera del computer e classificarla, ma questo andava ben oltre gli scopi del progetto.

1.4 Installazioni

Tramite Visual Studio Code, una volta che avete installato Python in locale sul vostro computer, aprite il relativo terminale ed eseguite il comando¹:

```
pip install torchvision
```

Listing 1.1: Esempio di installazione

¹per esempio per installare opencv si deve utilizzare cv2, vedi documentazione specifica

1.5 Online vs Offline

Nonostante ci si stia dirigendo verso una compilazione di codici *online*, ho preferito per questo progetto lavorare seguendo l'approccio classico, questo perché nonostante servizi come *Google Colab* (non ne conosco altri) siano molto utili per testare software molto velocemente, e senza installare nulla sul proprio computer locale, tale praticità decade quando le righe di codice diventano molte, e non vi è possibilità di seguire uno schema *divide et impera* del codice stesso, ovvero spostare blocchi di codice in un file separato, inoltre essendo basato sul cloud, tutti i file devono essere caricati su Google Drive, però se non si ha spazio sufficiente? Paghi ovviamente.

Inoltre non rimane una *cache* di ciò che si è eseguito, per cui una volta chiuso il browser, alla sua riapertura è necessario rieseguire tutti i pezzi di codice e riscaricare nuovamente tutti i pacchetti, con uno spreco di banda elevato, ma nonostante questo, per fare degli esperimenti è uno strumento che può tornare utile in molti contesti.

Per quanto riguarda la compilazione offline, suggerisco di installare Visual Studio Code, il quale è un IDE molto leggero, versatile e scaricabile gratuitamente, inoltre in accoppiata a Git (da installare localmente), insieme all'estensione per VS Code Git Graph, permette di velocizzare di molto il lavoro di scrittura.