

# Docker Setup per Progetto di Computer Vision - CV@TR2

---

## 1. Introduzione

Questo documento descrive il setup di un ambiente Docker per un progetto di Computer Vision basato su Python, Jupyter Notebook e librerie di Deep Learning. L'obiettivo è creare un contenitore portatile e ripetibile per sviluppare ed eseguire codice di anomaly detection su immagini e video.

## 2. Struttura del Progetto

La struttura della cartella del progetto è la seguente:

```
CV@TR2/  
├── code/  
│   ├── ObjectDetectionWithML_Python/  
│   └── [notebook e script Python]  
├── docker/  
│   ├── Dockerfile  
│   └── requirements.txt  
├── .gitignore  
└── README.md
```

---

## 3. Dockerfile - Spiegazione Dettagliata

Il file Dockerfile si trova in docker/Dockerfile ed è strutturato come segue:

```
FROM python:3.10-slim
```

```
WORKDIR /app
```

```
COPY requirements.txt.
```

```
RUN pip install --upgrade pip && pip install -r requirements.txt
```

```
COPY ../code/ObjectDetectionWithML_Python /app/code
```

## ***EXPOSE 8888***

***CMD ["jupyter", "notebook", "--ip=0.0.0.0", "--port=8888", "--notebook-dir=/app/code", "--no-browser", "--allow-root"]***

---

Spiegazione delle righe:

- FROM python:3.10-slim: utilizza un'immagine leggera di Python 3.10 come base.
- WORKDIR /app: imposta la directory di lavoro all'interno del container.
- COPY requirements.txt :. copia il file requirements.txt nel container.
- RUN pip install ...: aggiorna pip e installa le librerie elencate.
- COPY ../code/ObjectDetectionWithML\_Python ...: copia i file Python e notebook nel container.
- EXPOSE 8888: espone la porta per accedere a Jupyter Notebook.
- CMD [...]: avvia Jupyter Notebook all'interno del container.

## **4. Librerie nel requirements.txt - Motivazione**

Il file requirements.txt contiene:

- numpy - operazioni matematiche e gestione array
- pandas - analisi e manipolazione dati tabellari
- matplotlib - visualizzazione grafici
- opencv-python - elaborazione immagini e video
- torch, torchvision - deep learning e reti neurali
- notebook - supporto Jupyter
- scikit-learn - strumenti di machine learning
- tqdm - barre di progresso
- albumentations - libreria per data augmentation

## **5. Comandi per Costruire e Avviare il Container**

Aprire il terminale nella cartella docker/ ed eseguire:

1. Costruzione dell'immagine Docker:

```
docker build -t cvtr2-project.
```

---

2. Avvio del container:

```
docker run -p 8888:8888 cvtr2-project
```

---

3. (Facoltativo) Avvio con volume per salvare file localmente:

```
docker run -p 8888:8888 -v  
../code/ObjectDetectionWithML_Python:/app/code cvtr2-project
```

---

## **6. Conclusione**

Questo setup consente di isolare l'ambiente di sviluppo, garantire portabilità e replicabilità del progetto. L'utilizzo di Docker è particolarmente utile per progetti basati su AI e Computer Vision che richiedono librerie complesse e consistenza tra ambienti diversi.