



**UNIVERSITÀ
DEGLI STUDI
DI GENOVA**

Università degli Studi di Genova
Dipartimento di Scienze Matematiche, Fisiche e Naturali del DIBRIS

Implementazione di una Pipeline di Addestramento per Modelli di Object Detection basata su Faster R-CNN

Corso di Laurea in Informatica

Candidato:
Loris Prataiolo

Relatore:
Prof.ssa Giovanna Guerrini

Tutor Aziendale:
Dott. Luca Bianconi

Anno Accademico 2024/2025

Abstract

La presente tesi si propone di mostrare la progettazione di una pipeline per l'addestramento di modelli di intelligenza artificiale su KubeFlow in alternativa ad Apache Airflow, utilizzando come modello di base: Faster R-CNN. l'obiettivo di questo lavoro, è la realizzazione di un sistema, robusto e adattabile a diversi contesti, in grado di poter gestire in autonomia l'intero ciclo di addestramento di un modello di AI. Il progetto è stato sviluppato su Google Cloud Platform (GCP), utilizzando come orchestratore il sistema Google Kubernetes Engine (GKE), e KubeFlow per la realizzazione effettiva della pipeline. Il risultato è un sistema versatile, in grado di supportare diversi modelli di AI, e diverse tipologie di dataset; mostrando l'efficacia di KubeFlow nel semplificare il ciclo di training di un modello, rispetto ad Airflow.

Contents

| | | |
|----------|--|----------|
| 1 | Introduzione | 4 |
| 1.1 | Contesto e obiettivi del progetto | 4 |
| 2 | Tecnologie e Modelli Object Detection | 5 |
| 2.1 | Funzionamento ad alto livello dei modelli di AI per Object Detection | 5 |
| 2.1.1 | Le Bounding Box | 5 |
| 2.1.2 | L'importanza del dataset | 6 |
| 2.1.3 | Modello Faster R-CNN | 7 |

Chapter 1

Introduzione

La presente tesi è frutto del lavoro svolto durante il tirocinio curricolare presso STAM Tech S.r.l., azienda Genovese operante nel settore IT, e attiva nello sviluppo di tecnologie innovative a livello nazionale e internazionale.

1.1 Contesto e obiettivi del progetto

Il progetto è stato pensato per esplorare e implementare soluzioni innovative per l'integrazione di strumenti di pipelining AI alternativi ad Apache Airflow, con l'obiettivo di migliorare l'efficienza e la scalabilità dei flussi di lavoro di machine learning. Il cuore del progetto risiede nell'utilizzo di **KubeFlow**, e del suo modulo **Pipelines** per l'escuzione e monitoraggio del processo di training dei modelli.

Gli obiettivi principali del progetto includono:

- Studio e comprensione delle funzionalità di KubeFlow e della sua integrazione con Kubernetes.
- Sviluppo di pipeline di machine learning utilizzando KubeFlow Pipelines, prendendo come base il modello Faster R-CNN.
- Studio dello strumento **MLFlow** come strumento di raccolta e monitoraggio delle metriche di training dei modelli.
- Valutazione dell'efficacia di KubeFlow rispetto ad Apache Airflow in termini di scalabilità, facilità d'uso e di come l'integrazione di MLFlow possa migliorare il monitoraggio dei modelli in fase di training.

Il lavoro svolto ha richiesto uno studio approfondito delle tecnologie coinvolte, nonché una fase di sperimentazione pratica su come allenare un modello di object detection, e uno studio di come monitorare le metriche di training in modo efficace.

Chapter 2

Tecnologie e Modelli Object Detection

2.1 Funzionamento ad alto livello dei modelli di AI per Object Detection

I modelli di intelligenza artificiale di Object Detection sono progettati per identificare e localizzare oggetti specifici all'interno di immagini o video. Questa classe di modelli combina tecniche di computer vision e deep learning per analizzare il contenuto visivo e riconoscere pattern associati a diverse categorie di oggetti.

Il processo di *Object Detection* generalmente consiste in due fasi:

- **Rilevamento delle regioni di interesse (Region Proposal):** In questa fase, il modello identifica le aree dell'immagine che potrebbero contenere oggetti. Tecniche come Selective Search o Region Proposal Networks (RPN) sono comunemente utilizzate per generare queste proposte.
- **Classificazione e localizzazione:** Una volta identificate le aree di interesse, il modello classifica ciascuna regione in una delle categorie predefinite e determina la posizione esatta dell'oggetto all'interno della regione tramite le **bounding box**.

2.1.1 Le Bounding Box

Come accennato in precedenza, le bounding box sono dei rettangoli che vengono utilizzati per localizzare gli oggetti all'interno di un'immagine, o di un frame nel caso di un video. Esse sono definite da quattro coordinate: (x_{\min}, y_{\min}) e (x_{\max}, y_{\max}) , che rappresentano gli angoli superiore sinistro e inferiore destro del rettangolo, rispettivamente. Ad ogni bounding box viene associata anche una **classe** che identifica il tipo di oggetto rappresentato (ad esempio, persona, auto, cane, ecc.) e una **confidenza** che indica la probabilità che l'oggetto rilevato appartenga a quella classe.

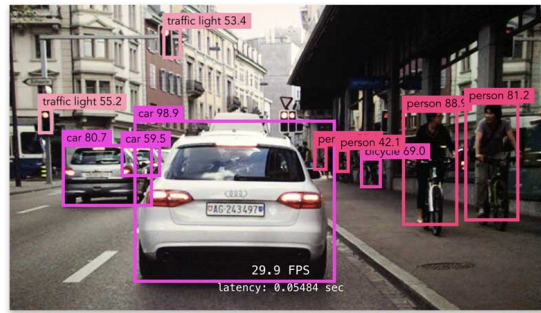


Figure 2.1: Un frame di una video camera, che mostra le vetture e semafori identificati da bounding box, alle quali sono associati la classe e la confidenza.

Fonte: <https://machinethink.net/blog/object-detection/>

Tuttavia, per motivi di semplicità e standardizzazione, le bounding box sono solitamente definite all'interno di un file di annotazione associato all'immagine. La struttura tipica di un dataset per l'addestramento e il test di modelli di Object Detection viene rappresentato nel seguente modo:

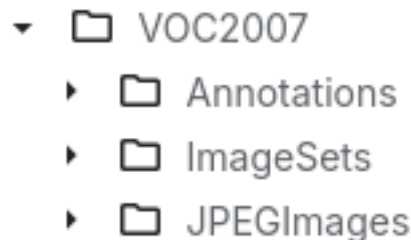


Figure 2.2: Esempio struttura tipica di un dataset per l'addestramento e il test di modelli di Object Detection, attraverso PASCAL VOC 2007.

Fonte: <https://www.kaggle.com/datasets/zaraks/pascal-voc-2007>

Una nota che vale la pena sottolineare ora ma che verrà approfondita più in avanti, è che il dataset non viene dato in pasto al modello così com'è, perchè ogni modello, a seconda della sua architettura, richiede che i dati siano formattati in un certo modo. Per questo motivo, prima di essere utilizzato per l'addestramento, il dataset deve essere **preprocessato** per adattarsi ai requisiti specifici del modello scelto.

2.1.2 L'importanza del dataset

Il dataset in un contesto di Object Detection è un aspetto cruciale per il successo dell'addestramento di un modello focalizzato su questo specifico compito. Il dataset non deve avere solo una quantità sufficiente di immagini per ogni classe, ma deve essere anche il più vario e ampio possibile, permettendo al modello di imparare a riconoscere gli oggetti in diverse condizioni. Infatti il riconoscimento di oggetti viene influenzato da molteplici fattori, tra cui:

- **Angolazione e prospettiva:** Una delle difficoltà principali nell'addestramento di modelli di Object Detection è la variazione della prospettiva da cui un oggetto può essere visto. Variando l'Angolazione cilindro può apparire come un cerchio, un ovale o un rettangolo, a seconda dell'angolazione da cui viene osservato.
- **Illuminazione e condizioni ambientali:** L'illuminazione ha una grande influenza sulla definizione e visibilità degli oggetti. Anche in questo caso, un oggetto può apparire molto diverso in condizioni di luce intensa rispetto a condizioni di scarsa illuminazione. Inoltre, condizioni atmosferiche come pioggia, nebbia o neve possono ulteriormente complicare il riconoscimento degli oggetti.

- **Occlusione:** Gli oggetti possono risultare parzialmente nascosti da altri oggetti o elementi nell'ambiente. Questo fenomeno, rappresenta una sfida significativa per i modelli di Object Detection, poichè devono essere in grado di identificare gli oggetti anche quando non sono completamente visibili.

Il dataset è quindi un tassello fondamentale per l'addestramento di questi modelli, siccome quello che si vuole fare, è insegnare al modello a ottenere informazioni dalle immagini, in modo che possa generalizzare e riconoscere gli oggetti in nuove immagini mai viste prima.

La fase di addestramento la si analizzerà più avanti, durante la fase implementativa del progetto, ma è importante sottolineare che il dataset viene solitamente pensato per essere suddiviso in tre sottoinsiemi distinti:

- **Training set:** Utilizzato per addestrare il modello
- **Validation set:** Utilizzato per ottimizzare i parametri del modello
- **Test set:** Utilizzato per valutare le prestazioni finali del modello

2.1.3 Modello Faster R-CNN

Bibliography

- Raghav Bali. Object detection basics - a comprehensive beginner's guide (part 1). <https://towardsdatascience.com/object-detection-basics-a-comprehensive-beginners-guide-part-1-f57380c89b78/>, 2024.
- Matthijs Hollwmans. One-stage object detection. <https://machinethink.net/blog/object-detection/>, 2018.
- Serdarildercaglar. Object detection model training: A comprehensive guide for beginners. <https://medium.com/@serdarildercaglar/object-detection-model-training-a-comprehensive-guide-for-beginners-d5a1274e7ac7>, 2024.