UNIVERSITÀ DEGLI STUDI ROMA TRE
DIPARTIMENTO DI INGEGNERIA CIVILE, INFORMATICA E DELLE TECNOLOGIE AERONAUTICHE

Progetto Machine Learning AA 2022/2023

Riconoscimento ammaloramenti stradali

Daniele Fippi

Abstract:

Lo scopo del progetto riguarda lo studio e l'applicazione di librerie Python per il Machine Learning. Il caso in esame riguarda il caso della Computer Vision per la rilevazione e il riconoscimento delle tipologie di ammaloramenti stradali contenuti in un dataset di immagini e un esame delle prestazioni delle tecnologie utilizzate.

1 Introduzione

La computer vision è una disciplina di grande interesse sia per il mondo accademico che per l'industria, dato che lo sviluppo di un sistema altamente affidabile richiede continui miglioramenti dal punto di vista prestazionale e che il suo impiego richiede particolare attenzione per determinate mansioni. Il caso d'uso in esame impiega la computer vision per il rilevamento e la classificazione di ammaloramenti stradali. Tale processo richiede l'addestramento di una rete neurale: nel caso in esame si farà uso della libreria Python YOLOv8 di ultralytics (https://github.com/ultralytics/ultralytics).

2 Computer Vision

La visione artificiale (nota anche come computer vision) è l'insieme dei processi che mirano a creare un modello approssimato del mondo reale (3D) partendo da immagini bidimensionali (2D). Lo scopo principale della visione artificiale è quello di riprodurre la vista umana. Vedere è inteso non solo come l'acquisizione di una fotografia bidimensionale di un'area ma soprattutto come l'interpretazione del contenuto di quell'area (fonte: Wikipedia). Il deep learning, sottoinsieme del machine learning, viene in forte aiuto quando si tratta di svolgere task che riguardano la computer vision, grazie all'impiego di architetture complesse di reti neurali. Dando per scontate le unità elementari e il funzionamento generale di una rete neurale, si può passare alla spiegazione del funzionamento di una rete neurale convoluzionale (CNN), una soluzione standard per quanto riguarda la computer vision. YOLO, seppur fa uso delle CNN, adotta una soluzione più complessa che comprende ulteriori moduli che verranno spiegati in seguito.

2.1 Convolutional Neural Networks (CNN)

Le reti neurali convoluzionali (CNN) rappresentano una classe di reti neurali deep specializzate nell'elaborazione di dati strutturati a griglia, come le immagini, nel nostro caso. Una CNN è composta principalmente da strati convoluzionali, strati di pooling e strati completamente connessi. Il primo strato, il convoluzionale, utilizza filtri o kernel per eseguire operazioni di convoluzione sull'input. Questi filtri scorrono sull'immagine di input, estraggono caratteristiche locali e generano mappe di caratteristiche. L'operazione di pooling riduce la dimensione spaziale delle mappe

di caratteristiche, riducendo così il numero di parametri e fornendo invarianza alle traslazioni.

2.2 YOLO

Yolo è una libreria Python per la computer vision, mantenuta da ultralytics. La sua ultima release è la versione 8 (YOLOv8), rilasciata nel 2023.

Molte tecnologie, come le R-CNN (Regional Convolutional Neural Networks), selezionano all'interno delle immagini un insieme di regioni con l'obiettivo di sottoporle a un classificatore ed ottenere delle bounding box per ciascun oggetto classificato. YOLO, d'altro canto, lavora su tutta l'immagine con il fine di percepire il maggior numero di informazioni possibili sul contesto dell'immagine e si è osservato che ciò porta a delle prestazioni migliori.

2.3 YOLO: Architettura e funzionamento

[1] L'immagine viene suddivisa in SxS celle, dove ciascuna cella può contenere B bounding boxes, con un proprio confidence score, che esprime quanto è probabile che all'interno del box vi sia un oggetto. Per ciascuna di esse si calcola la probabilità condizionata che il bounding box appartenga a una determinata classe.
Per quanto concerne l'architettura la parte iniziale è composta da un insieme di livelli convoluzionali (24) talvolta seguiti da maxpool layers per estrarre le features, mentre i livelli fully connected finali (2) servono per la predizione delle probabilità di appartenere a una classe e le coordinate.
Il livello finale usa la funzione di attivazione lineare, mentre gli altri livelli utlizzano la rectifier linear activation:

$$\phi(x) = \begin{cases} x & \text{if } x > 0, \\ 0.1x & \text{altrimenti} \end{cases}$$

YOLO, inoltre, è pre-addestrato su una certa attività, la quale viene utilizzata come punto di partenza per addestrare un modello su un'altra attività simile o correlata. In questo caso si parla di transfer learning.

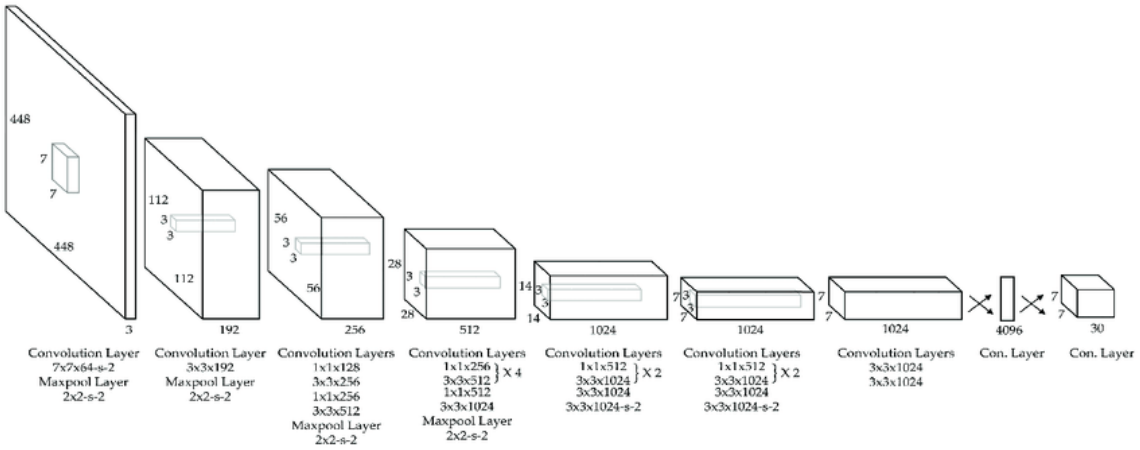

Figure 1: YOLO Architecture

3 Sperimentazione

La sperimentazione è stata condotta su una macchina messa a disposizione da Google Colab, con le seguenti caratteristiche: XYZ. Il dataset è relativo alla challenge RDDC2022 riguardante la detection di ammaloramenti stradali, in particolare, nel caso in esame, si farà riferimento ad ammaloramenti stradali presenti sulle strade di tre paesi: Norvegia, Repubblica Ceca e Stati Uniti. Al momento dell'esecuzione, le immagini sono ridimensionate a una risoluzione pari a 640x640. Il dataset viene suddiviso come segue:

- il 70% verrà impiegato per il training set
- il 20% verrà impiegato per il validation set
- il 10% verrà impiegato per il test set

Il sistema deve essere in grado di distinguere 4 classi di immagini:

- **D00**: Longitudinal cracks
- **D10**: Lateral cracks
- **D20**: Alligators
- **D40**: Potholes

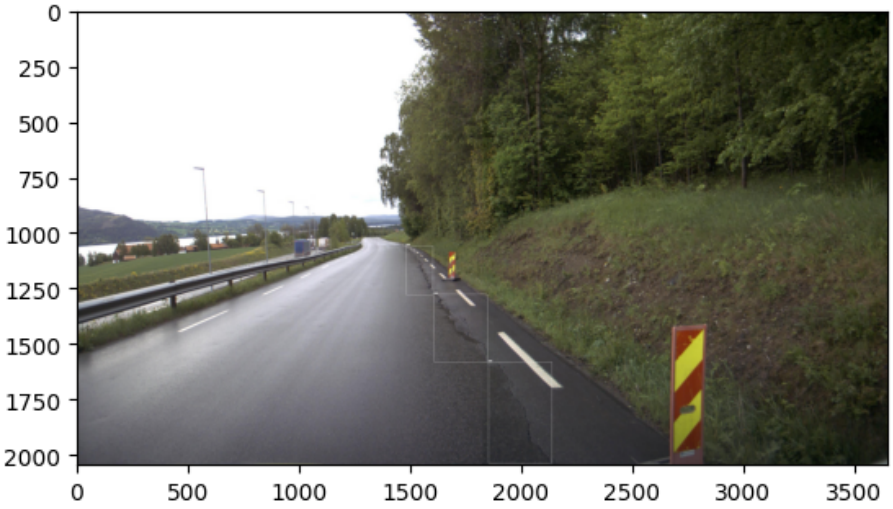

Figure 2: Esempio di etichettatura

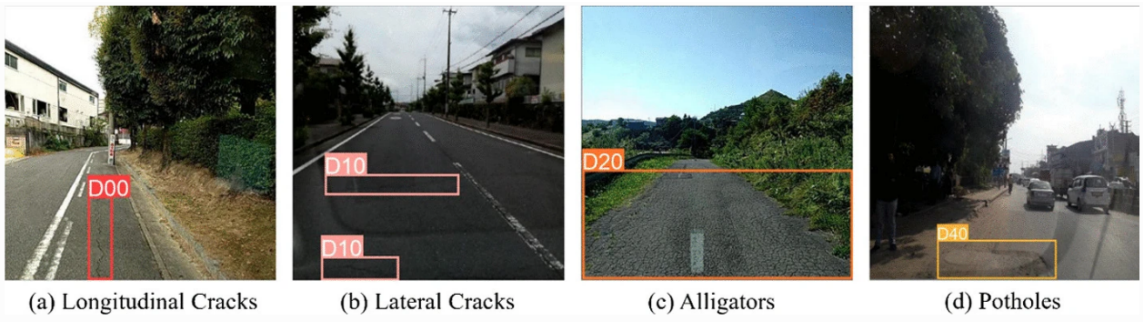

Figure 3: Esempio di classi di immagini[2]

3.1 Risultati ottenuti

In questa sezione verranno descritti i risultati ottenuti in seguito all'addestramento della rete. Per la misura delle prestazioni si utilizzano le seguenti metriche, molto comuni in ambito di computer vision:

- **Precision-Recall curve**
- **Average Precision (AP)**: per ogni classe viene calcolata l'area sotto la curva precision-recall. Questa area rappresenta la precisione media per quella classe. L'Average Precision (AP) è calcolata facendo la media delle precisioni medie di tutte le classi.
- **Mean Average Precision (mAP)**: la mAP è ottenuta facendo la media delle Average Precision di tutte le classi

Dataset	Images	Instances	Epochs	P	R	F1	mAP50
United States	480	1061	50	73.5%	45%	55.8%	52.3%
Norway	816	1151	20	24.1%	15.4%	18.8%	15.4%
Czech	356	587	50	30%	31.7%	30.8%	20.6%

Table 1: Performances

Si può evincere dai risultati che il punteggio migliore è stato ottenuto dal dataset United_States. La motivazione principale va ritrovata nel fatto che il dataset è composto da ammaloramenti stradali più presenti e marcati rispetto a Norway e Czech. Si riportano, inoltre, le prestazioni per le singole classi di United_States:

Class	Instances	P	R	F1	mAP50
all	1061	0.735	0.451	0.558	0.523
D00	636	0.661	0.745	0.70	0.76
D10	333	0.61	0.497	0.54	0.568
D20	83	0.669	0.561	0.61	0.609
D40	9	1	0	0	0.154

Table 2: USA: Perfoarmces on single classes

Le prestazioni migliori si hanno in corrispondenza di crepature longitudinali presenti nell'immagine, mentre il caso meno performante riguarda la tipologia di ammaloramento potholes (in italiano comunemente chiamate buche).

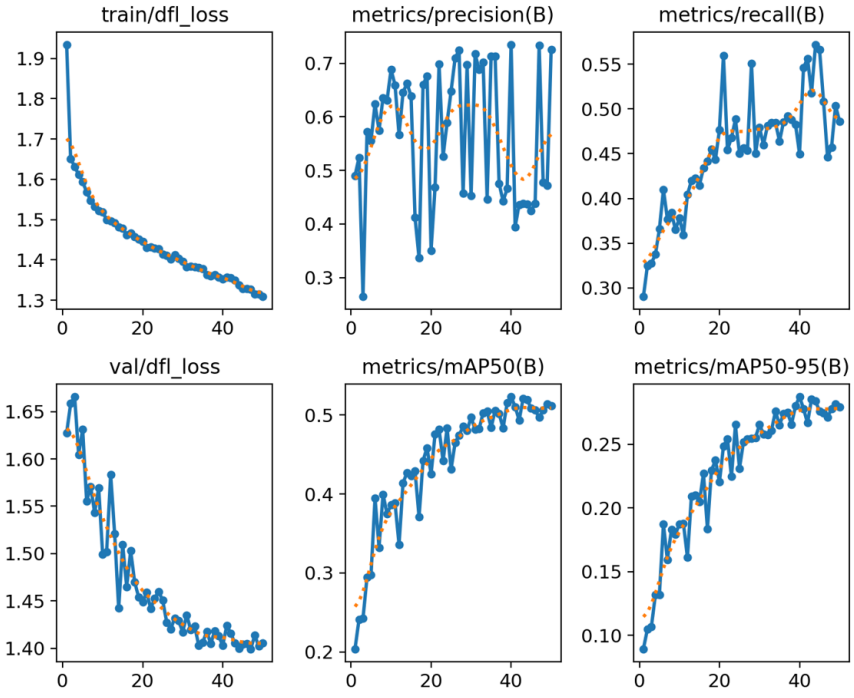

Figure 4: Performances on dataset through epochs

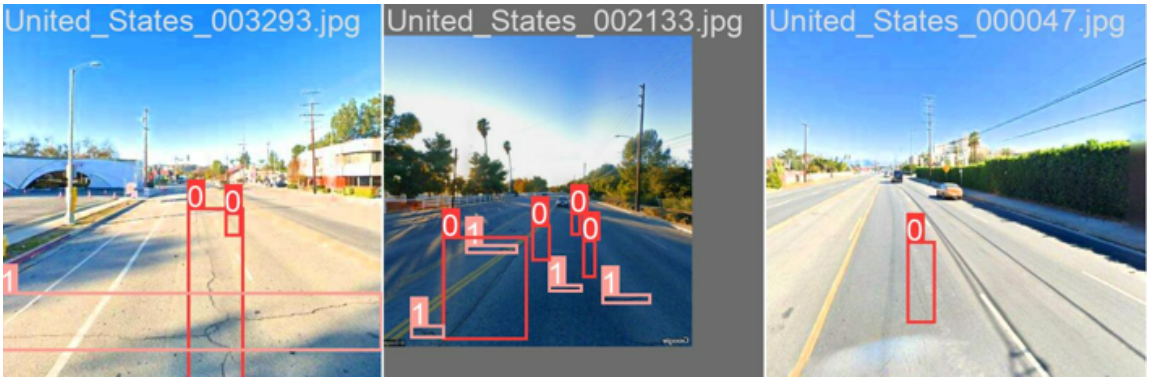

Figure 5: Some examples

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. 2015.

[2] F. Wan, C. Sun, H. He, G. Lei, L. Xu, and T. Xiao. Yolo-lrdd: a lightweight method for road damage detection based on improved yolov5s. *EURASIP Journal on Advances in Signal Processing*, 2022(1), Oct. 2022.