

Report progetto Intelligenza artificiale a.a. 2022/2023

Akira Petrolini, Matricola 0001039335

July 2023

Abstract

Confronto di diverse Convolutional Neural Network per la classificazione delle malattie di una pianta di pomodoro a partire dalle foto delle sue foglie.

1 Introduzione

1.1 Perché è un problema?

Nella mia città di provenienza, ovvero Parma, si ha la terza provincia per coltivazione di pomodoro da industria in Italia¹.

Un problema molto comune in questo tipo di coltivazioni è quello delle malattie che possono colpire le piante, e riconoscere il prima possibile se una pianta è sana o infetta è molto importante per garantire un intervento tempestivo nel caso in cui la pianta fosse malata.

Ciò è molto importante perché negli ultimi anni, a causa degli atipici fenomeni climatici registrati, come ad esempio la siccità nell'estate del 2022, le coltivazioni hanno già subito ingenti danni, ed evitare che esse siano danneggiate ulteriormente da malattie può essere centrale.

1.2 Per chi è un problema?

Come detto in precedenza, questo problema è molto importante per le aziende che coltivano pomodori, ma in generale è anche un problema per chiunque coltivi pomodori anche per uso privato nei propri orti.

Dunque, la sua utilità potrebbe trovare spazio non solo per un uso industriale, ma anche per uno domestico.

1.3 Che beneficio porterebbe una soluzione ipotetica al problema?

Una soluzione al problema dell'identificazione delle malattie delle piante di pomodoro garantirebbe un trattamento più veloce della pianta per quanto riguarda l'uso indus-

¹<https://www.parmatoday.it/economia/campagna-pomodoro-da-industria-2022-parma-la-terza.html>

triale, mentre per l'uso domestico andrebbe a fornire conoscenza a privati cittadini che potrebbero non conoscere necessariamente quali malattie possano affliggere una pianta di pomodoro.

1.4 Come si intende affrontare la realizzazione di una soluzione?

Attraverso l'uso di Convolutional Neural Networks (CNNs) è possibile analizzare immagini per estrarre da esse conoscenza. L'uso di CNNs permette di creare un classificatore che a partire dall'immagine di una foglia della pianta di pomodoro possa riconoscere se la pianta è sana oppure quale malattia abbia con una certa precisione.

1.5 Breve presentazione dei risultati ottenuti

E' possibile visualizzare graficamente in modo completo i risultati ottenuti all'interno della seguente sessione TensorBoard <https://tensorboard.dev/experiment/FGarrMz4QaSrlDs7xhlJfA/>

Table 1: Confronto dei risultati delle reti neurali

Modello	Epoca	Lr	Loss (Train)	Loss (Val)
AlexNet from Scratch	10	0.0001	1.7163	1.5209
AlexNet Fine Tuned	10	0.0001	0.5648	0.2756
ResNet-18 Fine Tuned	9	0.0001	0.8035	0.6280
MobileNet v3 Fine Tuned	10	0.0001	0.6908	0.4567

Modello	Accuratezza (Train)	Accuratezza (Val)
AlexNet from Scratch	39.09%	46.45%
AlexNet Fine Tuned	80.74%	90.99%
ResNet-18 Fine Tuned	74.51%	80.82%
MobileNet v3 Fine Tuned	77.36%	85.13%

2 Metodo Proposto

2.1 Ricerca della soluzione: quali alternative sono state considerate nella scelta della soluzione proposta?

Il metodo delle CNN è stato scelto in quanto si presta bene al riconoscimento di caratteristiche all'interno di immagini.

Quale CNN sia più performante è l'oggetto di questo studio, e per questo motivo sono state provate più architetture diverse.

Il Dataset scelto è invece stato scaricato dal sito Kaggle² e contiene oltre 20.000 immagini con 10 classi di malattie e una classe Healthy.

Il Dataset ha la seguente distribuzione ¹.

Inoltre, ecco alcune immagini presenti nel dataset come esempio da mostrare ².

²<https://www.kaggle.com/datasets/cookiefinder/tomato-disease-multiple-sources>

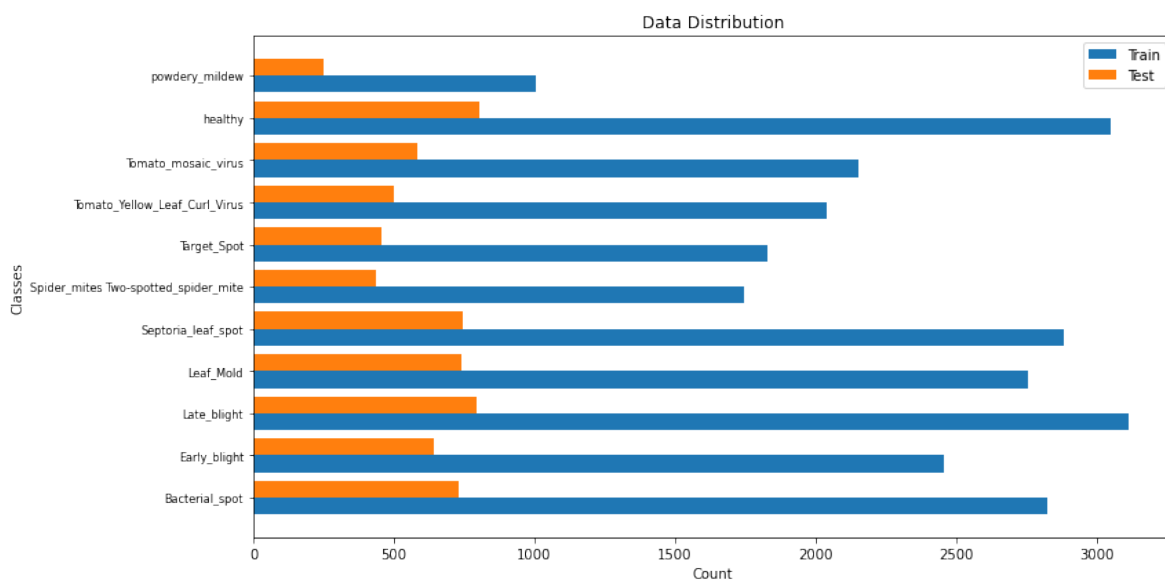


Figure 1: Distribuzione delle classi delle immagini contenute in train e validation

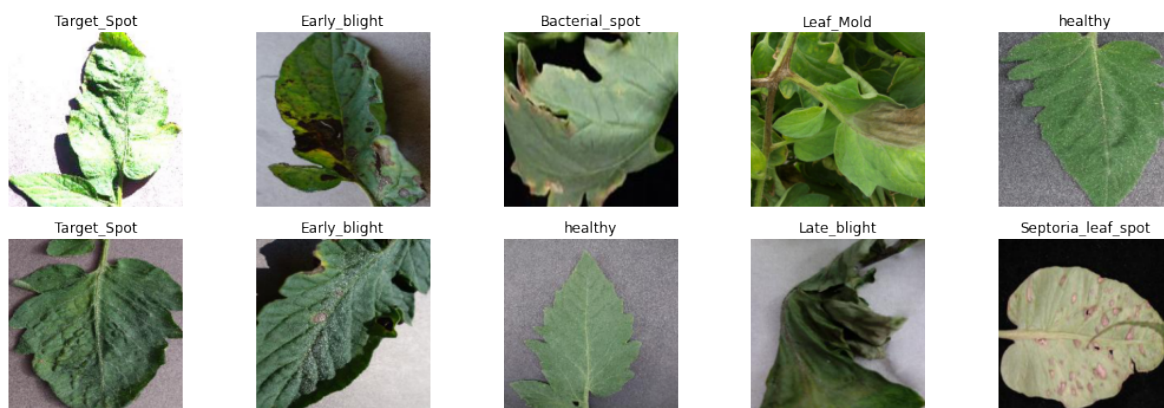


Figure 2: 10 immagini estratte a caso dal dataset e la loro classe di appartenenza

2.2 Giustificazione della scelta: perché sono state adottate alcune soluzioni piuttosto che...?

Le reti scelte sono quelle più facilmente implementabili potendo addestrare in ambiente locale senza ausilio di GPU compatibile con CUDA.

Il Dataset è invece stato scelto in quanto il più grande e con lo score di usability maggiore all'interno del sito Kaggle che riguardasse questo tema.

2.3 Descrizione del metodo per la misurazione delle performance

Le misurazioni di performance sono state effettuate su Training e Validation Set.

In particolare, la funzione di loss da minimizzare scelta è stata la cross-entropia

$$\text{CrossEntropy}(y, \hat{y}) = - \sum_i y_i \log(\hat{y}_i)$$

Mentre l'accuracy è calcolata nel seguente modo:

$$\text{Accuracy} = \frac{\text{Numero di previsioni corrette}}{\text{Numero totale di campioni}}$$

3 Risultati sperimentali

3.1 Istruzioni per la dimostrazione

3.2 Elenco delle tecnologie usate per gli esperimenti (i.e., versione del software) per la riproducibilità.

Processore: 1,4 GHz Intel Core i5 quad-core

Visual Studio Code Versione: 1.79.2 (Universal)

Ambiente virtuale Conda (Python 3.9.12)

Inoltre la riproducibilità è garantita da una funzione `fix_random()` che prende come argomento un intero e fissa il seed per la generazione di numeri casuali.

3.3 Risultati della configurazione migliore.

La configurazione migliore tra quelle provate è AlexNet con fine tuning dei pesi per quanto riguarda il valore di accuratezza sul Validation set.

3.4 Studio di ablazione: comparazione tra le diverse configurazioni.

Le diverse configurazioni e i loro parametri di valutazione sono visibili alla Tabella [1](#).

4 Discussione e conclusioni

4.1 Discussione dei risultati ottenuti

I risultati ottenuti sono accettabili considerando il basso numero di epoche in cui le tecnologie sono state allenate. Aumentandole, e facendo ulteriori tuning di iperparametri fissati all'interno del codice ritengo che sia possibile aumentare ulteriormente l'accuracy. Considerando che con un modello Fine-tuned di AlexNet si possa comunque raggiungere un 90% di accuracy sul validation set in sole 10 epoche, sono ottimista che si possa raggiungere un'accuracy con un addestramento con più epoche tale per cui un'azienda possa usare AlexNet per riconoscere le malattie delle piante di pomodoro.

4.2 Il metodo proposto rispetta le attese?

I modelli hanno un'alta precisione sul test set, di cui il migliore raggiunge circa il 90 %. Per un'applicazione industriale non è tuttavia un ottimo risultato, in quanto sarebbe una precisione elevata ma che sbaglia 1 volta su 10.

Una diagnosi sbagliata potrebbe portare a una cura errata della pianta, che farebbe perdere i soldi dei farmaci utilizzati, oltre che risultare inutile.

Tengo a precisare che comunque il progetto non è finalizzato all'utilizzo immediato della rete addestrata, ma nel cercare di capire quale tra le tecnologie testate fosse la migliore CNN per un progetto simile.

4.3 Limitazioni

Molte limitazioni nel progetto sono dovute all'architettura su cui sono state addestrate le reti.

Non avendo una GPU compatibile con CUDA, è stato possibile utilizzare modelli sofisticati o aumentare ad esempio alcuni iperparametri come il numero di epoche.

4.4 Lavori futuri

Una volta scelto il modello che riesca ad avere le migliori prestazioni, come nel mio caso AlexNet con fine tuning dei pesi. Sarebbe interessante sviluppare un'applicazione che possa sfruttare il modello allenato per, ad esempio, riconoscere le malattie della pianta di pomodoro direttamente dalla fotocamera dello smartphone di chi scarica l'applicazione, o da una foto scattata precedentemente e salvata in galleria.