

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Corso di Laurea Magistrale in Informatica

**Predizione dell'origine geografica di
Oli di Oliva tramite tecniche di
Deep Learning per l'elaborazione di immagini**

Relatore:

Prof. Maurizio Gabbrielli

Presentata da:

Michele Celozzi

Correlatore:

Dott. Stefano Pio Zingaro

III Sessione

Anno Accademico 2021/2022

Introduzione

L'olio extravergine d'oliva è uno dei prodotti alimentari più pregiati e apprezzati al mondo, grazie alle sue proprietà organolettiche e benefiche per la salute. Tuttavia, negli ultimi anni si è assistito ad un aumento delle frodi riguardanti questo prodotto, che hanno causato danni economici e reputazionali a molte aziende produttrici e hanno compromesso la fiducia dei consumatori.

Le frodi riguardanti l'olio extravergine d'oliva possono assumere diverse forme, come l'aggiunta di oli di bassa qualità, la miscelazione con oli di altre origini o la vendita di prodotti scadenti come se fossero di alta qualità. Queste pratiche fraudolente non solo danneggiano i produttori e i consumatori, ma hanno anche ripercussioni negative sull'ambiente e sulla salute pubblica.

In particolare, le frodi possono essere commesse in diverse fasi della filiera produttiva, dalla produzione alla distribuzione e alla vendita. Di seguito sono riportate alcune delle principali tipologie di frodi riguardanti l'olio extravergine d'oliva:

- Miscelazione con oli di bassa qualità: questa pratica consiste nell'aggiunta di oli di oliva di qualità inferiore (ad esempio oli di oliva raffinati o di provenienza estera) all'olio extravergine d'oliva, al fine di aumentare la quantità prodotta e ridurre i costi. In alcuni casi, viene utilizzato anche olio di semi.
- Vendita di oli scadenti come extravergine: in alcuni casi, gli oli di bassa qualità vengono venduti come olio extravergine d'oliva, anche se non

rispondono ai requisiti stabiliti dalla legge. Ciò può avvenire attraverso l'etichettatura ingannevole o la falsificazione dei certificati di qualità.

- Falsificazione dell'origine: in alcuni casi, l'olio extravergine d'oliva viene prodotto in un Paese diverso da quello indicato sull'etichetta o nel certificato di qualità. Ciò può avvenire per evitare le restrizioni commerciali o per trarre vantaggio dall'immagine di un Paese noto per la produzione di olio extravergine d'oliva di alta qualità.
- Produzione di olio extravergine d'oliva non conforme: ciò può avvenire a causa di errori o negligenze nella produzione, come l'uso di olive di bassa qualità, la lavorazione con attrezzature non adeguate o l'utilizzo di tecniche di estrazione non corrette.
- Vendita di oli extravergine d'oliva adulterati: in alcuni casi, gli oli extravergine d'oliva vengono adulterati con sostanze chimiche o additivi per migliorarne il sapore o l'aspetto, o per prolungarne la durata di conservazione. Questa pratica può essere dannosa per la salute e costituisce un'infrazione alla legge.

Queste sono solo alcune delle tipologie di frodi riguardanti l'olio extravergine d'oliva, che possono avere ripercussioni negative sulla salute dei consumatori, sull'ambiente e sull'economia.

È quindi importante garantire la qualità e l'autenticità di questo prodotto di eccellenza, attraverso controlli adeguati e tramite l'utilizzo di metodologie e strumenti che permettano di riconoscere l'origine geografica di un olio.

Soluzione proposta

In questa tesi si propone l'utilizzo di tecniche di Deep Learning con l'obiettivo di prevedere l'origine geografica di un olio d'oliva a partire dalle immagini dei grafici relativi alla sua composizione chimica.

In particolar modo, rispetto a quanto descritto in letteratura si è scelto di utilizzare un approccio del tutto nuovo basato sull'elaborazione delle immagini, del quale ad oggi non sono conosciute eventuali applicazioni nel campo delle frodi olearie, basando il lavoro di tesi sulle ricerche e gli studi già effettuati.

Il contrasto delle frodi olearie è un ambito in cui le aziende produttrici di olio hanno da sempre espresso interesse in quanto alla base dell'individuazione di truffe, in modo particolare verso le frodi che riguardano l'adulterazione di un olio. Inoltre, fornisce un supporto per il contrasto di pratiche fraudolente, tutelando gli interessi economici e reputazionali dei produttori, oltre a garantire la vendita di prodotti sani e di qualità ai consumatori.

Nel capitolo 1 verrà riportata la revisione della letteratura in cui si descrivono i principali articoli alla base di questa tesi.

Nel capitolo 2 verrà fornita una descrizione dettagliata della metodologia e delle relative scelte prese. Verranno esposti gli step seguiti per ottenere il miglior modello che riesce a predire nel modo più corretto l'origine geografica degli oli. Inoltre, verrà descritto il modello utilizzato e le varie fasi di recupero dei dati, definizione del dataset, fine-tuning del modello e la fase di test.

Nel capitolo 3 verranno mostrati e descritti i risultati ottenuti dalle analisi e a seguire le conclusioni nelle quali vengono delineati i possibili lavori futuri.

Indice

Introduzione	i
1 Stato dell'arte	1
2 Metodo e Materiali	3
2.1 Gas Cromatografia - Spettrometria di Massa	3
2.2 Dataset utilizzato	6
2.3 Setup dell'esperimento	14
2.3.1 Scelta del modello	19
2.4 Inferenza	27
3 Risultati e Discussione	29
Conclusioni	39
3.1 Conclusioni	39
3.2 Lavori Futuri	41
Bibliografia	43

Elenco delle figure

2.1	Struttura GC-MS [4]	4
2.2	Grafico analisi chimica di un olio greco	5
2.3	Dataset Completo Idrocarburi	7
2.4	Dataset Completo VOCs	8
2.5	Dataset Completo Cultivar	9
2.6	Split in Train, Val e Test Set (n. samples/percentuali)	10
2.7	Funzionamento Leave-One-Out Cross-Validation [11]	12
2.8	Fine tuning [15]	17
2.9	Architettura MLP-Mixer [19]	21
2.10	Architettura ResNet [21]	22
2.11	Architettura ViT [22]	24
2.12	Confronto performance MLP-Mixer, ResNet e ViT	25
3.1	Matrice di confusione delle predizioni del modello ViT Small .	32
3.2	Matrice di confusione delle predizioni del modello ViT Base .	35

Elenco delle tabelle

3.1	Metriche di valutazione del ViT Small per ogni classe	33
3.2	Metriche di valutazione del ViT Base per ogni classe	36

Capitolo 1

Stato dell'arte

L'utilizzo dei modelli di machine learning per la predizione dell'origine geografica degli oli extravergine di oliva è una delle aree di ricerca su cui differenti studi hanno mostrato un particolare interesse. Diversi modelli di Machine Learning sono in grado di apprendere dalle caratteristiche chimiche degli oli extravergine di oliva e di prevedere la loro origine geografica.

Uno degli studi che sono stati già condotti in questo senso è [1], in cui si affrontano le stesse problematiche delineate in questo lavoro. In particolare modo, nella tesi citata si è deciso di seguire un approccio orientato alle feature estratte dall'analisi chimica degli oli. Tramite una Rete Neurale Convolutionale si effettua la predizione dell'origine geografica dando in input al modello le feature estratte dai grafici, i quali rappresentano la composizione chimica dell'olio. L'estrazione delle feature viene realizzata tramite dei metodi di riconoscimento e individuazione dei picchi in una heat map, i quali altro non sono che le feature che caratterizzano il dataset.

Viene proposto tra i vari lavori futuri, quello di utilizzare direttamente le immagini al fine di ottenere un'analisi diretta basata su un approccio orientato all'elaborazione delle immagini.

È stato svolto un secondo studio [2], in cui il dataset è composto da dati scalari relativi alle analisi chimiche effettuate. In questo caso, si propone di predire la varietà dell'olio extravergine di oliva della Spagna basandosi sull'utilizzo del classificatore LDA. In particolare, gli oli in questione hanno come provenienza geografica diverse aree della Spagna, dunque si ridimensiona il problema a zone limitate nella stessa nazione.

Infine, un ultimo paper [3], riguardo l'analisi oggetto di questa tesi, tratta della creazione di un modello, denominato dagli autori POLIVAR, basato su reti neurale allo scopo di predire e identificare l'origine geografica di oli di oliva monovarietali. Nello specifico, viene utilizzato un Multilayer Perceptron per eseguire gli esperimenti.

Questi sono solo alcuni degli studi condotti nell'ambito della predizione della provenienza geografica di oli, molti altri sono stati effettuati in questa direzione aventi tutti una caratteristica in comune: l'utilizzo di un dataset composto da dati scalari relativi alla composizione chimica ottenuta dalla Gas Cromatografia.

Inoltre, per il tipo di problema che si propone di risolvere in questa tesi, non sono stati condotti, allo stato attuale, particolari studi riguardo all'utilizzo di grafici reattivi alla composizione chimica di alimenti (o altri tipi di prodotti) con modelli basati su reti neurali.

Capitolo 2

Metodo e Materiali

In questo capitolo, verranno descritti gli approcci e i metodi utilizzati nel lavoro di tesi. In particolare, verranno spiegati la tecnica analitica utilizzata per l'analisi degli oli, il dataset su cui è stato condotto lo studio, il modello scelto per l'analisi, il linguaggio e le librerie o framework utilizzati ed infine gli step seguiti per raggiungere gli obiettivi fissati.

2.1 Gas Cromatografia - Spettrometria di Massa

Nella presente sezione, verrà fornita una descrizione riguardo la tecnica di analisi dei vari oli che compongono il dataset utilizzato in questa tesi, seguendo le informazioni ottenute dagli articoli [4], [5], [6], [7] e [8].

La Gas Cromatografia Spettrometria di Massa (GC-MS) è una tecnica analitica utilizzata per l'identificazione e la quantificazione di composti chimici in una miscela. In questa tecnica, un campione viene vaporizzato e separato in componenti individuali attraverso una colonna di separazione (cromatografia).

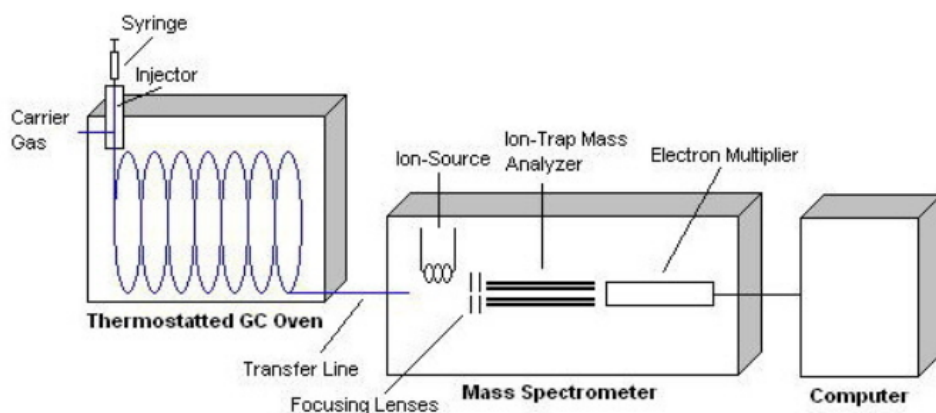


Figura 2.1: Struttura GC-MS [4]

Il processo di separazione avviene tramite il movimento del campione attraverso la colonna di separazione con l'aiuto di un gas di trasporto (solitamente elio o azoto). I componenti della miscela si muovono attraverso la colonna a diverse velocità, a seconda della loro affinità per la fase stazionaria. In questo modo, i componenti della miscela si separano l'uno dall'altro e vengono eluiti dalla colonna in tempi diversi.

Dopo essere stati separati dalla colonna di separazione, i componenti della miscela vengono rilevati da un rivelatore, che genera un segnale proporzionale alla concentrazione del componente eluito. La GC è una tecnica analitica altamente sensibile e selettiva che viene comunemente utilizzata nell'industria chimica, farmaceutica e alimentare per analizzare composti organici volatili e semi-volatili.

Successivamente, i componenti separati vengono analizzati uno alla volta mediante uno spettrometro di massa per determinare la loro massa molecolare e la loro struttura chimica.

Lo spettrogramma di massa è il risultato dell'analisi spettrometrica di massa, che registra la distribuzione delle masse dei composti presenti nel campione. Questa tecnica scompone una sostanza in frammenti più piccoli e determina la massa elettrica di ogni frammento per determinare la composizione della sostanza originale.

Il processo di spettrometria di massa inizia con la ionizzazione della sostanza da analizzare. Una volta ionizzata, la sostanza viene introdotta in un campo magnetico dove viene separata in base alla sua massa elettrica. Questo avviene perché ogni ione ha una carica elettrica e una massa specifica. L'interazione tra la carica elettrica e il campo magnetico fa sì che gli ioni si curvino su una traiettoria circolare, con un raggio di curvatura inversamente proporzionale alla loro massa elettrica.

Il raggio di curvatura degli ioni viene quindi misurato da un rivelatore, che è in grado di individuarli in base alla loro massa. Questa informazione viene quindi utilizzata per generare uno spettro di massa che rappresenta la distribuzione di massa degli ioni presenti nella sostanza.

Lo spettro di massa può essere utilizzato per identificare la composizione chimica della sostanza. Ciò avviene confrontando quello della sostanza analizzata con quello di composti noti presenti in una libreria di spettri di massa. Di seguito un esempio dell'analisi svolta tramite questa tecnica.

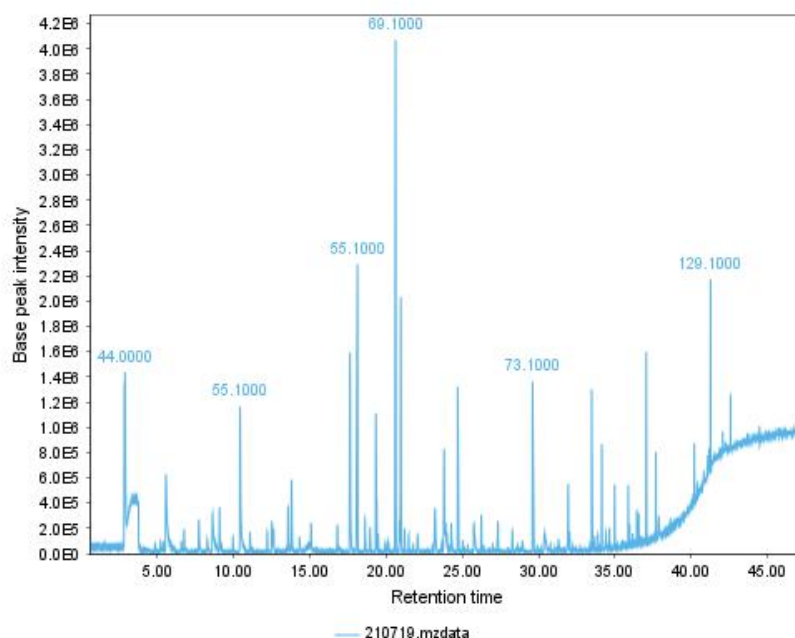


Figura 2.2: Grafico analisi chimica di un olio greco

In questo grafico, l'asse delle x rappresenta il tempo di ritenzione delle particelle, ovvero il tempo che impiega la molecola ad essere separata dalle altre, mentre l'asse delle y rappresenta l'intensità del segnale ionico. In particolare, ogni particella ha una polarità e a seconda della polarità ci saranno delle caratteristiche diverse per ogni particella. Particelle con la stessa polarità risulteranno unite, mentre quelle con diversa polarità separate; questo, comporta una differenza nel tempo di separazione delle molecole durante il processo di eluzione dei componenti.

Il picco più alto corrisponde al composto più abbondante e la posizione dei picchi corrisponde alla massa molecolare del composto.

L'interpretazione dello spettrogramma di massa può fornire informazioni sulla struttura chimica dei composti presenti nel campione. Inoltre, l'utilizzo di librerie di spettri di massa può aiutare nella determinazione dell'identità dei composti sconosciuti presenti nel campione.

2.2 Dataset utilizzato

I dati sono stati forniti dall'Università di Firenze che ha già svolto in precedenza uno studio [9] simile sfruttando dati scalari ottenuti con la medesima tecnica analitica. Tali dati consistono nella rappresentazione in grafici dell'analisi chimica di idrocarburi e VOCs su 218 campioni di oli provenienti da "Grecia", "Grecia-Peloponneso", "Grecia-Creta", "Italia", "Spagna", "Tunisia" e "Portogallo", relativi alla campagna olearia 2021-2022. I campioni in questione sono stati raccolti ed analizzati dai laboratori Carapelli¹ e poi recuperati dall'Università di Firenze.

Dalla raccolta di questi dati si è ottenuto un dataset iniziale composto da sei classi tra loro molto sbilanciate. Alcune di queste sono composte da un numero non sufficientemente alto di esempi così da renderle poco rappresentative. È stato scelto, dunque, di raggruppare le classi "Grecia", "Grecia-

¹Carapelli Firenze S.p.A., Via Leonardo da Vinci 31, 50028, Tavarnelle Val di Pesa (Firenze), Italy

Peloponneso” e “Grecia-Creta” in un’unica classe “Grecia” ottenendo cinque possibili valori per la classe target.

I dati forniti dall’Università di Firenze riguardano due tipologie di analisi differenti: quella sugli idrocarburi e quella sui VOCs. Gli idrocarburi e i VOCs (Volatile Organic Compounds) degli oli d’oliva sono composti organici non volatili e volatili rispettivamente che si trovano naturalmente nell’olio d’oliva e che sono costituiti da atomi di carbonio e idrogeno.

La presenza di elevati livelli di idrocarburi o VOCs può indicare una possibile contaminazione dell’olio d’oliva con sostanze estranee, come l’olio di semi o altri oli vegetali.

Come è possibile osservare dai grafici seguenti la distribuzione dei samples nelle varie classi non risulta avere grandi differenze, tuttavia ogni olio differisce da ogni altro sulla base degli idrocarburi e dei VOCs, i quali differiscono a loro volta in quanto è possibile ottenere, per uno stesso olio, grafici completamente diversi relativi a queste due tipologie di componenti a causa della loro differente natura.

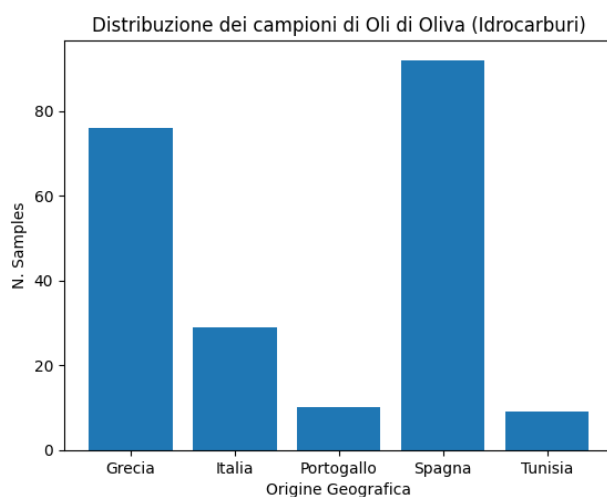


Figura 2.3: Dataset Completo Idrocarburi

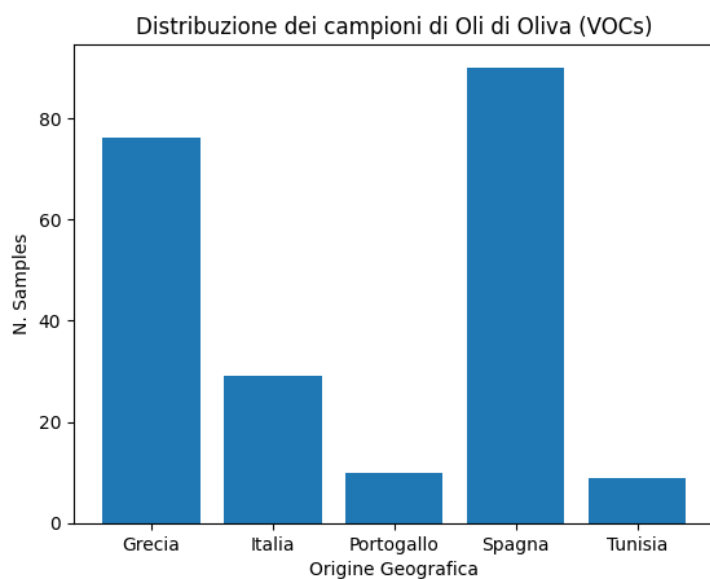


Figura 2.4: Dataset Completo VOCs

Dunque, per il lavoro e lo studio di questa tesi è stato scelto di condurre le analisi utilizzando solamente i dati sugli idrocarburi al fine di osservare il comportamento dei modelli e come questi si adattano al problema che ci si propone di risolvere.

Successivamente, è stata osservata la distribuzione degli esempi oltre che sulla provenienza geografica dell'olio, come appena visto, anche sulla cultivar di ogni olio.

La cultivar di un olio si riferisce alla varietà di pianta da cui viene estratto l'olio, ovvero la sua origine botanica. Ci sono molte varietà di piante da olio diverse, ognuna delle quali può produrre oli con caratteristiche organolettiche e chimiche diverse.

L'origine botanica è importante perché influisce sulle caratteristiche dell'olio stesso, come il sapore, l'aroma e il profilo nutrizionale.

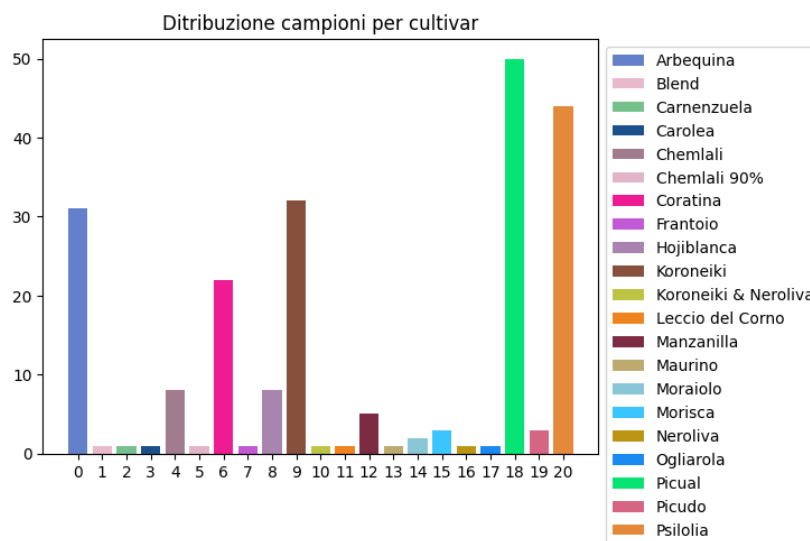


Figura 2.5: Dataset Completo Cultivar

Come mostrato in questo grafico, per il dataset fornito sono presenti un numero molto alto di cultivar di cui la maggior parte di esse sono rappresentate da un numero davvero molto esiguo di esempi che non permette di ottenere informazioni sufficientemente rilevanti. Dunque, si è scelto di non condurre lo studio in questo senso in quanto vorrebbe dire predire la cultivar di un olio a partire da un dataset che contiene un numero di classi troppo elevato rispetto a quelle che contengono un numero di esempi sufficientemente rappresentativo. Inoltre, in fase di preprocessing al fine di ottenere un dataset valido per l'addestramento e l'inferenza del modello si potrebbe provvedere all'eliminazione delle classi con un numero di campioni non sufficiente, ma in questo modo si otterrebbe un dataset troppo piccolo tanto da non poter eseguire lo studio che questa tesi propone.

A seguito, dunque, di tali osservazioni si è scelto di risolvere il problema di predire l'origine geografica dell'olio e non la cultivar.

I dati ottenuti dalle analisi sono rappresentati da file di tipo `mzData` che contengono tutte le informazioni relative alle componenti chimiche che contraddistinguono un olio dall'altro. Per ottenere le immagini che compongono

il dataset su cui effettuare lo studio è stato utilizzato il software “MZmine” [10] per l’esportazione delle immagini dei grafici in formato JPEG.

Dopo aver ottenuto il dataset, questo è stato suddiviso in tre subset: training set, validation set e test set. Tale suddivisione è stata effettuata al fine di analizzare il comportamento di tre modelli differenti con lo scopo di ottenere il miglior modello per il problema di predizione oggetto del lavoro di questa tesi.

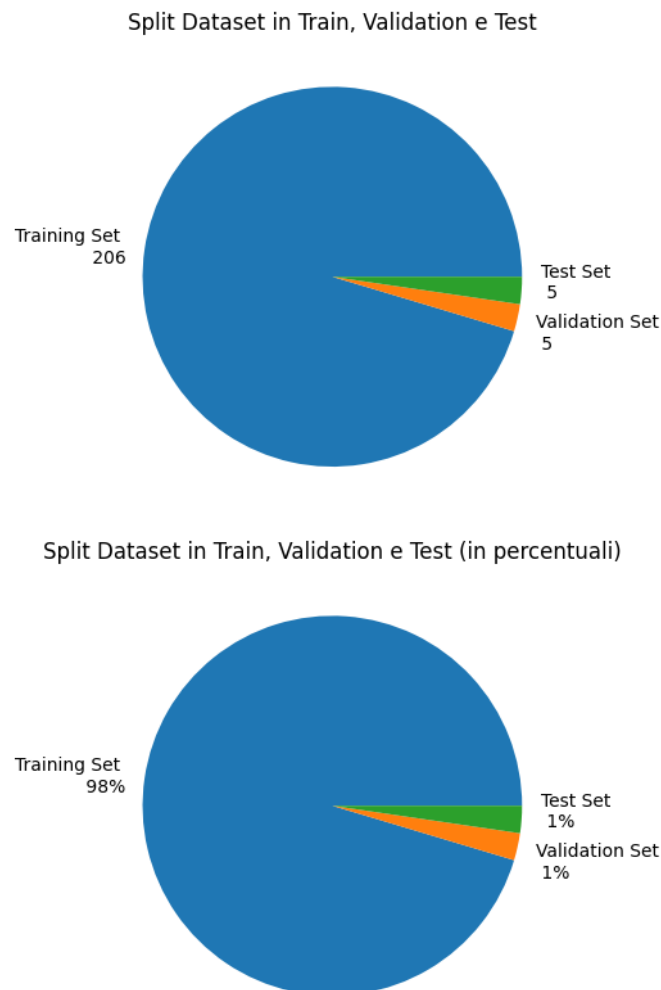


Figura 2.6: Split in Train, Val e Test Set (n. samples/percentuali)

Nello specifico, come è possibile notare dai grafici, dato il numero ridotto di campioni nel dataset è stato scelto di effettuare la suddivisione nel seguente modo:

- training set: costituisce il 98% del dataset con 208 samples distribuiti su 5 classi in modo sbilanciato.
- validation set: formato da un campione per ogni classe (1% del dataset originale).
- test set: costituito da un numero di osservazioni pari a 1 per ogni classe (1% del dataset originale).

Sono stati quindi effettuati degli esperimenti per l'individuazione del modello che meglio caratterizza i dati in input e, successivamente, si è deciso di utilizzare la Leave-One-Out Cross-Validation [11] per la validazione del modello scelto. Questa tecnica è utilizzata nell'apprendimento automatico per valutare le prestazioni di un modello. Si tratta di un tipo di metodo di cross-validazione che prevede l'addestramento del modello su tutti i dati disponibili tranne uno, utilizzando poi il modello stesso per fare una previsione sul dato lasciato fuori.

In pratica, nella LOOCV, il modello viene addestrato su tutti i dati tranne uno, mentre il dato lasciato fuori viene utilizzato per testare la performance del modello. Questo processo viene ripetuto per ogni dato presente nel dataset, in modo che ogni volta un solo dato viene utilizzato come set di test mentre tutti gli altri vengono utilizzati come set di addestramento.

In questo modo, si ottiene una valutazione accurata delle prestazioni del modello, perché questo viene testato su tutti i dati disponibili, infatti, ad oggi, risulta essere uno dei metodi più forti di validazione. Tuttavia, il processo di addestramento richiede un grande sforzo computazionale e può incidere sul tempo di addestramento.

In particolare, si ottengono un numero di predizioni pari al numero di samples contenuti nel dataset che verranno utilizzate per il calcolo dell'accuratezza del modello.



Figura 2.7: Funzionamento Leave-One-Out Cross-Validation [11]

L'immagine sopra mostra un esempio di come funziona LOOCV:

1. si consideri un dataset di 6 osservazioni. Si seleziona l'osservazione 6 e si rimuove dal dataset;
2. si addestra il modello sul rimanente dataset di 5 osservazioni;
3. si utilizza il modello addestrato per prevedere il valore dell'osservazione 6 rimossa;
4. si calcola l'errore tra la previsione del modello e il valore reale dell'osservazione 6;
5. si ripete il procedimento per ogni osservazione del dataset (ovvero si seleziona l'osservazione precedente come quella da rimuovere e si ripete l'addestramento del modello e il calcolo dell'errore);
6. alla fine di questo processo, si ottiene una stima dell'errore medio del modello, che può essere utilizzata per valutare la sua performance.

In sintesi, LOOCV è una tecnica di validazione incrociata molto utile quando si dispone di un dataset con un numero ridotto di osservazioni, poiché utilizza tutte le osservazioni disponibili per l'addestramento del modello e la valutazione delle sue prestazioni.

Sia nella prima suddivisione realizzata del dataset che in quella ottenuta da LOOCV, il dataset di training presenta uno sbilanciamento tra le classi, ovvero alcune classi sono rappresentate da un numero significativamente inferiore di esempi rispetto ad altre. In questi casi, il modello potrebbe presentare difficoltà nel distinguere tra le classi sotto-rappresentate e sovrappesare quelle sovra-rappresentate.

Il Weighted Random Sampler [12] aiuta a risolvere questo problema, fornendo un modo per dare un maggior peso alle classi sotto-rappresentate durante la fase di addestramento. Ciò significa che questi esempi verranno presentati al modello più frequentemente, migliorando la sua capacità di distinguere tra le classi con un minor numero di osservazioni.

Inoltre, può essere utilizzato per effettuare il campionamento stratificato, ovvero campionare i dati in modo che la distribuzione delle classi nel campione estratto rispecchi la distribuzione delle classi nel dataset completo.

Il Weighted Random Sampler è una classe utilizzata in PyTorch per creare un campionario casuale ponderato, utilizzato per il campionamento di dataset di dati non bilanciati.

Questa tecnica di campionamento risolve il problema campionando casualmente le osservazioni del dataset in modo che ogni classe sia rappresentata in modo proporzionale alla sua frequenza.

Il campionario ponderato seleziona casualmente un elemento del dataset in base alla sua probabilità di peso, la quale è calcolata come il rapporto tra la frequenza di quella classe e la frequenza totale di tutte le classi nel dataset.

Il Weighted Random Sampler è stato utilizzato con PyTorch DataLoader per creare batch di dati bilanciati che rispecchiano la distribuzione delle classi

nel dataset.

Riassumendo, tale tecnica di campionamento aiuta a gestire lo sbilanciamento di classe nei dataset di addestramento, garantendo che il modello sia esposto in modo più uniforme a tutte le classi, migliorando così la sua capacità di generalizzazione.

In generale, il Weighted Random Sampler è un modo utile per affrontare questo tipo di problema in modo efficiente e semplice, senza dover raccogliere più dati per le classi sotto rappresentate o utilizzare tecniche di data augmentation per bilanciare il dataset.

Relativamente alle tecniche di data augmentation, è stato scelto di non utilizzarle in quanto in letteratura non ci sono ad oggi studi riguardo l'applicazione di questa tecnica a immagini che rappresentano grafici. Infatti, risultano essere tecniche molto complesse nel caso menzionato in quanto non è possibile avere conoscenze sufficientemente utili per aumentare i dati del dataset. Inoltre, con un numero di campioni ridotto diventa più complicato cercare le giuste caratteristiche a supporto di tecniche di data augmentation.

2.3 Setup dell'esperimento

Il lavoro di tesi è stato realizzato tramite l'utilizzo del linguaggio di programmazione Python versione 3.9.6 in quanto particolarmente adatto nell'ambito dell'Intelligenza Artificiale grazie alle molteplici librerie disponibili a supporto della costruzione e dell'addestramento dei modelli, in modo particolare permettono di effettuare suddivisione del dataset, preprocessing dei dati e costruire modelli.

La principale libreria utilizzata per l'addestramento ed inferenza del modello è PyTorch [12], di cui è stata utilizzata la versione 1.13.1 la quale utilizza la piattaforma di calcolo parallelo CUDA versione 11.6.

Questa libreria è stata sviluppata principalmente da Facebook AI Research e viene utilizzata per creare modelli di Deep Learning come reti neurali, in particolare quelle con architetture di tipo convoluzionale (CNN), ri-

corrente (RNN) o Transformer. PyTorch è stato creato per offrire una maggiore flessibilità e una migliore scalabilità rispetto ad altri framework di Deep Learning.

Una delle sue caratteristiche distintive è la capacità di utilizzare il concetto di “computazione a flusso di dati” (data flow computation), che consente di definire e manipolare reti neurali in modo più intuitivo e naturale rispetto ad altri framework. Inoltre, offre una vasta gamma di funzionalità utili per la ricerca e lo sviluppo di modelli di machine learning, tra cui un sistema di autograd per il calcolo dei gradienti e la gestione della backpropagation, una vasta gamma di funzionalità di ottimizzazione, un'ampia selezione di dataset pre-elaborati e pre-addestrati, e molto altro ancora.

È stata utilizzata la libreria torchvision [12] fornita direttamente da PyTorch per la trasformazione delle immagini nel dataset. In particolare, si applicano le seguenti trasformazioni alle immagini del dataset originale:

- **Resize:** ridimensiona un'immagine ad una nuova dimensione specificata. Questa funzione viene utilizzata in quanto i modelli utilizzati in questo lavoro accettano in input immagini di dimensioni 224×224 , a differenza delle immagini contenute nel dataset che hanno una grandezza pari a 499×390 ;
- **ToTensor:** converte un'immagine PIL (Python Imaging Library) o NumPy array in un tensore PyTorch;
- **Normalize:** normalizza un tensore di immagine in base a una media e una deviazione standard predefinite; nel caso degli esperimenti di questa tesi, la media e deviazione standard sono state definite rispettivamente $(0.485, 0.456, 0.406)$ e $(0.229, 0.224, 0.225)$.

La libreria è stata anche utilizzata per ottenere un oggetto Dataset a partire dalla suddivisione in cartelle della collezione di dati. Nel dettaglio, il dataset è formato da 218 immagini di grafici della composizione chimica dell'olio. Queste immagini all'interno della cartella “dataset” sono a loro volta

suddivise in cartelle il cui nome corrisponde alla loro classe target di appartenenza. Al fine di ottenere un oggetto Dataset, input del modello utilizzato, si è usufruito del metodo ImageFolder che restituisce in output una serie di tuple immagine-classe, dove l'immagine è rappresentata come un tensore. Inoltre, tramite questo metodo si applicano le suddette trasformazioni.

È stata utilizzata una prima suddivisione del dataset in training set, validation set e test set utilizzata per validare gli iperparametri del modello. Successivamente, per ottenere una validazione più forte, come descritto nella sezione precedente, è stata utilizzata la Leave-One-Out Cross-Validation.

La libreria utilizzata per la selezione dei dati di train e di test è Scikit-Learn [13] che mette a disposizione il metodo LeaveOneOut il quale permette di ottenere gli indici dei sample del dataset originale che dovranno comporre il training e test set per ogni split possibile, dove il numero totale di split corrisponde al numero di immagini contenute nel dataset originale.

Per questo lavoro di tesi è stato scelto di utilizzare modelli preaddestrati, ovvero modelli che sono stati addestrati su grandi quantità di dati prima di essere utilizzati per una specifica attività o compito. Questi sono generalmente addestrati su dataset di grandi dimensioni utilizzando tecniche di apprendimento supervisionato, non supervisionato o di rinforzo. Una volta addestrati, i modelli preaddestrati possono essere utilizzati per svolgere diverse attività di apprendimento automatico.

Un vantaggio di questi modelli è che richiedono meno tempo e risorse per l'addestramento rispetto a quello di un nuovo modello da zero. Inoltre, poiché sono stati addestrati su grandi quantità di dati, possono produrre risultati più accurati rispetto ai modelli addestrati su dataset più piccoli.

Grazie all'utilizzo di "timm" [14], la libreria creata da Ross Wightman, è stato possibile recuperare i modelli preaddestrati ed implementare il fine-tuning dei modelli.

Il fine-tuning [15] è una tecnica che consiste nell'addestrare un modello preaddestrato su un nuovo insieme di dati specifici per migliorare le prestazioni del modello su un particolare compito o dominio di applicazione.

In pratica, il fine-tuning prevede l'utilizzo di un modello preaddestrato che ha già imparato alcune caratteristiche generali del dominio dell'apprendimento automatico, successivamente viene ulteriormente addestrato su un insieme di dati più piccolo e specifico per migliorare la sua capacità di risolvere un particolare problema. Può essere eseguito in diversi modi, come la sostituzione dell'ultimo strato del modello con un nuovo strato di classificazione, l'aggiunta di nuovi strati al modello esistente o l'addestramento del modello intero con un learning rate ridotto.

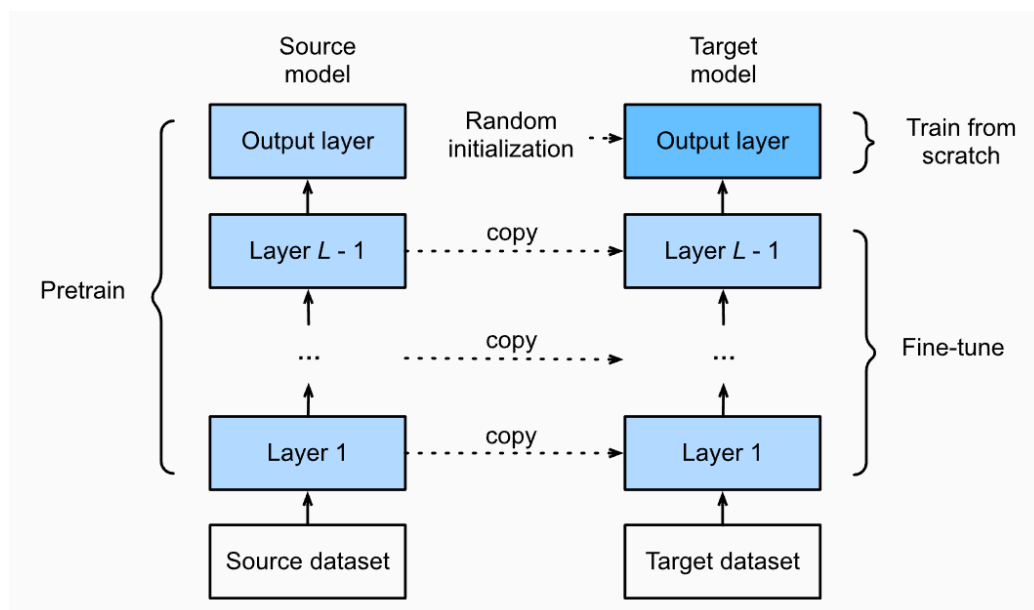


Figura 2.8: Fine tuning [15]

Tale tecnica è utile perché consente di utilizzare modelli preaddestrati per risolvere compiti specifici con minori risorse di elaborazione e di tempo computazionale e può portare a prestazioni migliori rispetto alla costruzione di un nuovo modello da zero.

Nel caso del lavoro di questa tesi, si è scelto di sostituire l'ultimo strato del modello con un nuovo strato che mantiene i pesi dell'ultimo layer del modello preaddestrato, ma che modifica il numero di classi in output, corrispondente al numero di classi nel dataset, ovvero cinque classi.

Per l'addestramento delle reti neurali è stato scelto di utilizzare come funzione di Loss la Cross-Entropy Loss e come ottimizzatore lo Stochastic Gradient Descent (SGD).

La Cross-Entropy Loss [16] è una funzione di costo utilizzata spesso come obiettivo di addestramento nei modelli di Deep Learning. Essa viene utilizzata soprattutto in problemi di classificazione multiclasse, dove l'obiettivo del modello è quello di assegnare una delle possibili classi a un determinato input. È una misura della discrepanza tra la distribuzione di probabilità predetta dal modello e la distribuzione di probabilità reale dei dati di training.

In generale, il modello cerca di minimizzare la Cross-Entropy Loss durante la fase di addestramento, in modo da avvicinare la distribuzione di probabilità predetta dal modello alla distribuzione reale dei dati di training. Questo si traduce in una migliore capacità del modello di classificare correttamente i dati di test.

Lo Stochastic Gradient Descent (SGD) [17] è un algoritmo di ottimizzazione ampiamente utilizzato in Deep Learning per aggiornare i pesi del modello durante la fase di addestramento.

L'obiettivo dell'algoritmo SGD [18] è quello di minimizzare la funzione di costo del modello. L'algoritmo è definito "stocastico" perché utilizza un sottoinsieme casuale dei dati di training, detto "mini-batch", per calcolare la derivata della funzione di costo rispetto ai pesi del modello e aggiornarli. Questo rende l'algoritmo più efficiente rispetto al Gradient Descent classico, che utilizza tutti i dati di training per calcolare la derivata della funzione di costo e aggiornare i pesi del modello.

L'algoritmo SGD ha dimostrato di essere un'efficace soluzione per l'addestramento di modelli di Deep Learning su grandi dataset, grazie alla sua capacità di gestire grandi quantità di dati utilizzando un sottoinsieme casuale di essi, senza compromettere la qualità delle prestazioni del modello. Tuttavia, è importante selezionare un buon learning rate e scegliere la giusta strategia di aggiornamento dei pesi del modello, in modo da evitare problemi come l'overfitting o l'impossibilità di raggiungere la convergenza.

Infine, la fase di training di tutti i modelli viene ripetuta n volte, dove n è il numero di split effettuati dal Leave-One-Out Cross-Validation che corrisponde al numero di osservazioni presenti nel dataset, come spiegato nella sezione precedente.

2.3.1 Scelta del modello

La fase di scelta del modello dipende:

- dal tipo di problema che si sta cercando di risolvere;
- dalle caratteristiche specifiche, dal volume e dalla complessità dei dati disponibili;
- dalla quantità di risorse computazionali disponibili;
- dalle prestazioni richieste dal modello.

Al fine di ottenere il miglior modello che meglio generalizza i dati, sono stati effettuati esperimenti utilizzando tre differenti architetture con configurazioni differenti.

I tre modelli confrontati sono: MLP-Mixer base, il Vision Transformer base e small ed infine ResNet50D. Una caratteristica comune a tutti questi modelli è che accettano input di dimensioni 224×224 , inoltre sono tutte reti preaddestrate a cui è stato applicato uno step di fine-tuning.

MLP-Mixer è un'architettura di rete neurale proposta nel paper [19].

L'idea alla base di MLP-Mixer è quella di utilizzare solo Multilayer Perceptron (MLP) per costruire una rete neurale convoluzionale (CNN) senza utilizzare operazioni di convoluzione. Ciò è in contrasto con le tradizionali reti neurali convoluzionali che utilizzano layer convoluzionali.

In MLP-Mixer, un'immagine di input viene divisa in patch di dimensione fissa e ogni patch viene rappresentata da un vettore. La rete neurale è costituita da una serie di blocchi di elaborazione, in cui ogni blocco prende in input un insieme di patch rappresentate da vettori e restituisce un nuovo insieme di patch rappresentate da nuovi vettori.

Ogni blocco di elaborazione è costituito da due parti: un'operazione MLP per estrarre le caratteristiche delle patch e un'operazione di mescolamento (mixing) per combinare le informazioni tra le patch.

L'operazione MLP in MLP-Mixer è simile all'operazione di trasformazione di punto fisso utilizzata nelle reti neurali di tipo Transformer, ma in questo caso i vettori di input vengono trasformati utilizzando una serie di layer MLP.

L'operazione di mescolamento (mixing) utilizza un'operazione di permutazione delle dimensioni dei vettori seguita da un'operazione MLP per combinare le informazioni tra le diverse patch.

MLP-Mixer ha dimostrato di ottenere risultati competitivi su diversi benchmark di classificazione di immagini rispetto alle tradizionali reti neurali convoluzionali. Tuttavia, MLP-Mixer richiede un numero elevato di parametri rispetto alle reti neurali convoluzionali tradizionali, il che può limitare la sua utilità in alcune applicazioni con risorse limitate.

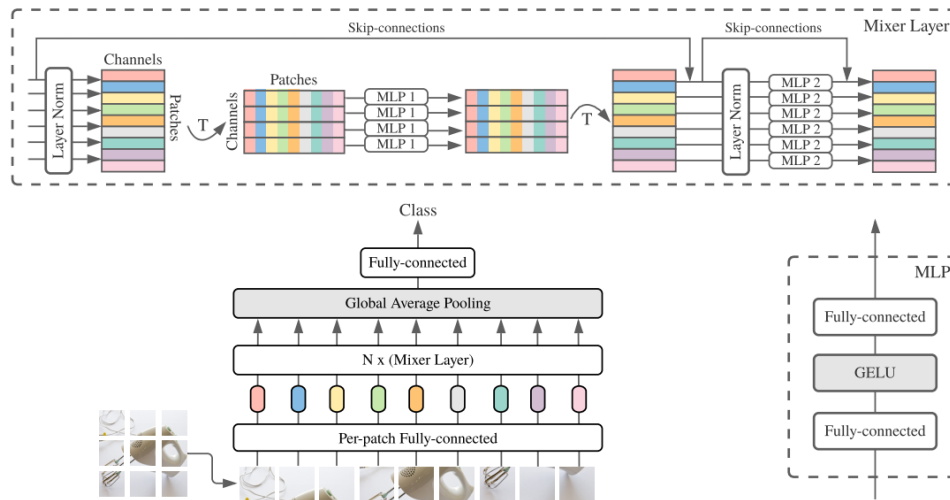


Figura 2.9: Architettura MLP-Mixer [19]

ResNet (Residual Network) è un'architettura di rete neurale convoluzionale (CNN) proposta nell'articolo [20].

Il problema che ResNet cerca di risolvere è il degrado del gradiente durante la backpropagation nelle reti neurali profonde. In particolare, quando si addestrano reti di questo tipo, diventa difficile aggiungere nuovi strati senza che ciò causi un degrado delle prestazioni del modello.

Per risolvere questo problema, ResNet utilizza blocchi di connessioni residue (residual connections) per aiutare a propagare il gradiente attraverso la rete. In un blocco di connessioni residue, l'output di un blocco viene sommato direttamente all'input del blocco successivo. Questo permette di bypassare i singoli strati, consentendo al gradiente di fluire più facilmente attraverso la rete, anche in presenza di molti strati.

Ognuno dei blocchi residui di ResNet contiene più strati convoluzionali e di normalizzazione del batch. I blocchi possono essere combinati in diverse configurazioni, con una profondità della rete variabile a seconda del compito da risolvere.

ResNet ha dimostrato di essere molto efficace in diverse applicazioni di visione artificiale, come il riconoscimento di immagini e la segmentazione del-

Quest'ultimo è composto da più blocchi di encoding, ognuno dei quali contiene più multi-head attention e uno strato completamente fully connected. La multi-head attention permette al modello di considerare le relazioni tra i vari token all'interno della sequenza, mentre lo strato completamente connesso consente di combinare le informazioni provenienti dai vari token in un'unica rappresentazione complessiva.

Una volta che l'immagine è stata trasformata in una sequenza di token, l'Encoder Transformer di ViT può essere utilizzato per eseguire una varietà di task di visione artificiale, come la classificazione di immagini o la segmentazione semantica.

Il ViT è stato dimostrato essere altamente competitivo rispetto ad altre architetture di rete neurale per compiti di classificazione su grandi set di dati di immagini, come ad esempio l'ImageNet. Inoltre, ha il vantaggio di essere molto flessibile e adattabile, in quanto può essere utilizzato anche per altri compiti di visione artificiale, a condizione che l'input sia rappresentabile come una sequenza di vettori.

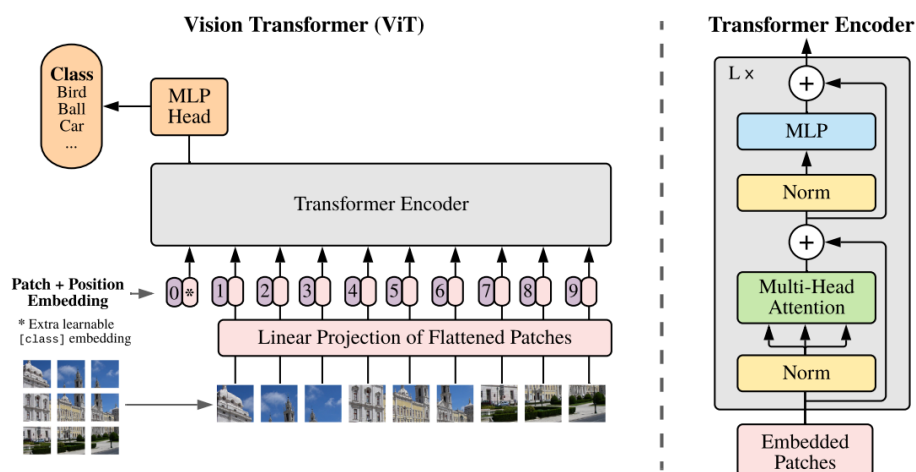


Figura 2.11: Architettura ViT [22]

Tutti questi modelli sono stati preaddestrati sul dataset ImageNet ed accettano input di dimensioni 224×224 . Le differenze tra i modelli confrontati riguardano in particolare la loro configurazione.

Nel caso dell'MLP-Mixer è stata utilizzata la versione base del modello, più grande e complessa della versione small, ha una profondità di 12 layer e utilizza patch di dimensioni 16×16 con buone capacità di elaborazione, in grado di catturare rappresentazioni più complesse e dettagliate delle immagini.

In merito alla rete ResNet, è stata utilizzata la ResNet50D, ovvero una variante della rete neurale ResNet50 (Residual Network con 50 layer) che utilizza il pooling di profondità (depthwise pooling).

Il pooling di profondità è una tecnica utilizzata per ridurre la dimensione delle feature map in profondità consentendo di ridurre il numero di parametri e di aumentare l'efficienza computazionale della rete.

ResNet50D utilizza il pooling di profondità per ridurre le dimensioni delle feature map nei primi strati della rete, prima dell'applicazione dei blocchi

residuali. In questo modo, la rete può gestire un maggior numero di feature map in modo più efficiente e ridurre il costo computazionale complessivo della rete.

Riguardo al ViT, invece, sono state testate due versioni: la base e la small entrambe con patch di 16×16 . La principale differenza tra ViT Small e ViT Base è la loro complessità e dimensione.

ViT Small ha una profondità di 8 layer e un totale di 24 milioni di parametri, mentre ViT Base ha una profondità di 12 layer e un totale di 86 milioni di parametri.

Sono state confrontate le performance dei modelli, tutti addestrati sullo stesso dataset con il medesimo preprocessing e la medesima impostazione delle immagini come descritto precedentemente.

A seguito di tale confronto sono stati ottenuti i seguenti risultati.

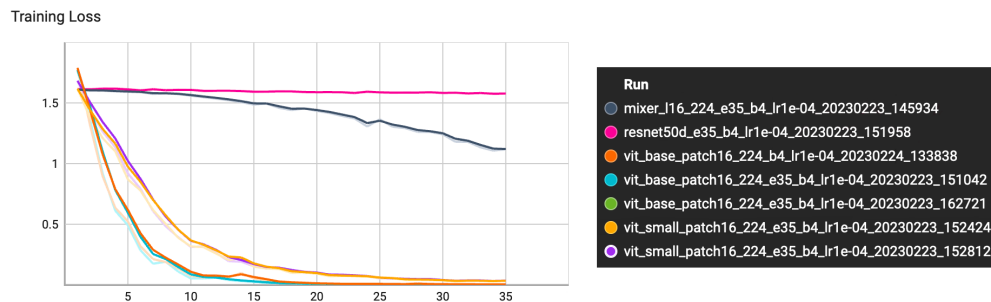


Figura 2.12: Confronto performance MLP-Mixer, ResNet e ViT

La scelta del modello è stata dettata dalle seguenti proprietà:

- buone capacità nell'operare con input di immagini di grafici
- convergenza più veloce verso l'errore minimo
- generalizzazione senza rischio di overfitting

Dal grafico nell'immagine sopra è possibile notare che il miglior modello che rispetta le suddette proprietà è il Vision Transformer, il quale permette

di raggiungere un buon valore di loss a partire dalla quinta epoca rispetto a tutti gli altri modelli.

Infatti, come è possibile osservare, il peggior modello che meno riesce a risolvere il tipo di problema analizzato in questa tesi risulta essere la ResNet; questa rete ha un comportamento costante in tutte le 35 epoche e non riesce ad apportare miglorie man mano che ad ogni epoca apprende nuove informazioni, dunque questo modello non è adatto a problemi di classificazione basati su grafici in quanto la sua loss iniziale e quella finale risulta pressoché identica.

Nel caso dell'MLP Mixer, invece, si può notare come i primi miglioramenti, e dunque una riduzione della loss, si verificano intorno alla ventesima epoca. Tuttavia, al concludersi dell'addestramento con la trentacinquesima epoca non si riesce ad ottenere una loss al di sotto del valore 1. Tale comportamento rende anche questo modello poco adatto alla manipolazione del tipo di problema trattato, in particolar modo non è in grado di gestire come input immagini di grafici.

Dunque, la scelta conseguente l'analisi e le considerazioni appena descritte è stata quella di utilizzare un Vision Transformer (ViT) come modello di predizione, con lo scopo di predire l'origine geografica degli oli tramite l'elaborazione di immagini di grafici della loro composizione chimica.

L'addestramento del modello finale, ottenuto dall'esperimento appena descritto, è stato eseguito utilizzando le due varianti small e base scegliendo un numero di epoche pari a 35, learning rate pari a $1 - e4$ e la dimensione dei minibatch pari a 4.

2.4 Inferenza

La fase di inferenza del Deep Learning si riferisce alla fase in cui il modello di apprendimento automatico addestrato viene utilizzato per effettuare previsioni su nuovi dati. Durante questa fase, il modello utilizza i pesi e i bias appresi durante la fase di addestramento per produrre una previsione basata sui dati di test forniti in input.

In generale, la fase di inferenza comporta le seguenti operazioni:

1. raccolta dei dati di input: i dati di input vengono raccolti e preparati per l'elaborazione da parte del modello;
2. normalizzazione dei dati: i dati vengono normalizzati in modo da ridurre la varianza e portarli alla stessa scala utilizzata durante la fase di addestramento;
3. passaggio dei dati attraverso il modello: i dati normalizzati vengono passati attraverso il modello per ottenere una previsione;
4. interpretazione delle previsioni: il risultato prodotto dal modello viene interpretato per ottenere una previsione significativa.

Nel caso di questa tesi i primi due punti elencati non sono stati eseguiti in quanto tali operazioni sono state effettuate preventivamente sul dataset completo prima della suddivisione, in training set e test set, tramite la Leave-One-Out Cross-Validation.

Questa fase è stata ripetuta n volte pari al numero di split ottenuti dalla LOOCV, il quale corrisponde al numero di osservazioni all'interno del dataset. Le predizioni ottenute sono state tutte raccolte e utilizzate per la visualizzazione dei risultati tramite una matrice di confusione.

L'implementazione della matrice di confusione è stata resa possibile tramite due metodi messi a disposizione da Scikit-Learn [13], ovvero “`confusion_matrix`” e “`ConfusionMatrixDisplay`”, rispettivamente per la costruzione e visualizzazione della matrice di confusione.

Capitolo 3

Risultati e Discussione

In questo capitolo, verranno presentati i risultati ottenuti in fase di inferenza dei modelli. In particolare, la fase di test è stata eseguita per ogni possibile split del dataset iniziale che si ricorda dipendere dal numero di campioni nel dataset. Infatti, il numero di predizione corrisponde al numero di osservazioni contenute nel dataset originale.

È stata utilizzata una matrice di confusione ([23], [24]) al fine di osservare il comportamento del modello e di calcolare diverse metriche di valutazione delle performance del modello a partire dai risultati ottenuti.

La matrice di confusione risulta essere un ottimo strumento di valutazione delle prestazioni in quanto mostra il numero di volte che l'algoritmo ha classificato correttamente e erroneamente le istanze di ciascuna classe.

In particolare, essa è composta da quattro elementi principali:

- True Positive (TP): il numero di istanze positive correttamente classificate come positive;
- False Positive (FP): il numero di istanze negative erroneamente classificate come positive;

- True Negative (TN): il numero di istanze negative correttamente classificate come negative;
- False Negative (FN): il numero di istanze positive erroneamente classificate come negative.

Inoltre, la matrice di confusione può essere utilizzata per identificare eventuali errori di classificazione dell'algoritmo e per individuare eventuali pattern o tendenze nelle prestazioni dell'algoritmo su diverse classi di dati.

Le metriche di cui si è fatto uso in questa tesi aiutano a valutare le performance di un algoritmo di classificazione in modo più dettagliato rispetto alla semplice accurata classificazione complessiva. Quelle scelte sono: l'accuratezza, la precisione, la recall e F1 score.

L'accuratezza (accuracy) [23] è una metrica che indica la percentuale di istanze totali che sono state classificate correttamente dall'algoritmo di classificazione. Per calcolare l'accuratezza a partire dalla matrice di confusione, si devono sommare i valori corretti (TP e TN) e dividere per il numero totale di istanze:

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

La precisione (precision) [23] misura la proporzione di istanze classificate come positive che sono effettivamente positive.

In altre parole, la precisione indica la percentuale di risultati positivi classificati correttamente dal modello rispetto al numero totale di istanze classificate come positive.

Si calcola come:

$$precision = \frac{TP}{TP+FP}$$

La recall (o sensitivity) [23] indica la percentuale di istanze positive che sono state correttamente classificate dal modello rispetto al numero totale di istanze effettivamente positive e si ottiene nel seguente modo:

$$recall = \frac{TP}{TP+FN}$$

L’F1-score [23] è la media armonica della precisione e della recall. Questa metrica tiene conto sia della precisione che della recall e fornisce una valutazione complessiva delle prestazioni dell’algoritmo di classificazione.

In generale, un alto valore di F1-score indica che l’algoritmo di classificazione ha prestazioni migliori, ovvero ha una buona combinazione tra precisione e recall.

La formula è la seguente:

$$F1 - score = \frac{2 \times (precision \times recall)}{precision + recall}$$

Nella pagina seguente viene mostrata la matrice di confusione relativa ai risultati ottenuti a seguito dell’addestramento e dell’inferenza del modello ViT Small.

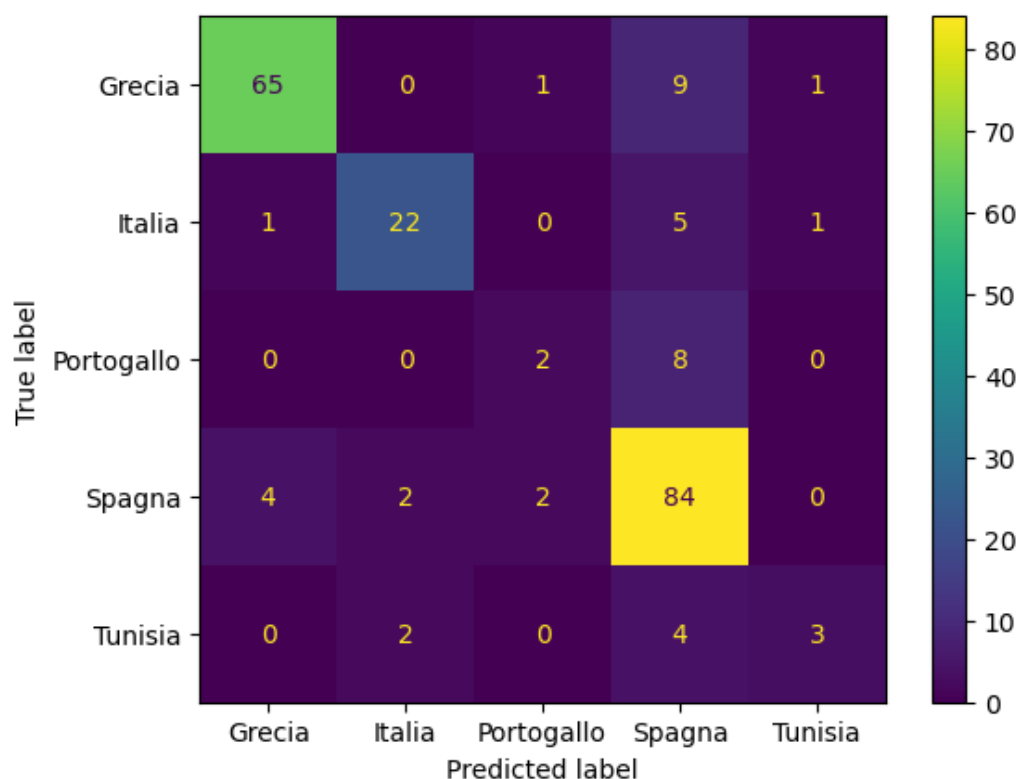


Figura 3.1: Matrice di confusione delle predizioni del modello ViT Small

Da questa immagine emerge subito come il modello ViT Small, relativamente al problema che ci si è proposti di risolvere in questa tesi, ha un numero basso di falsi negativi. Nello specifico, riesce a differenziare in buon modo tra le varie classi evitando overfitting e, essendo una rete neurale preaddestrata su ImageNet, è in grado di distinguere molto bene tra i grafici passati in input al modello e le immagini comuni di ImageNet, le quali tipicamente rappresentano paesaggi, oggetti o animali.

Applicando le metriche di valutazione esposte precedentemente si ottengono le seguenti informazioni che forniscono una specifica descrizione delle performance del modello utilizzato:

Metrica	Grecia	Italia	Portogallo	Spagna	Tunisia
<i>accuracy</i>	92.59%	94.91%	94.91%	84.26%	96.3%
<i>precision</i>	93%	85%	40%	76%	60%
<i>recall</i>	86%	76%	20%	91%	33%
<i>F1 – score</i>	0.89	0.80	0.27	0.83	0.43

Tabella 3.1: Metriche di valutazione del modello per ogni classe

È possibile notare come la classe su cui il modello ha riscontrato una minor accuratezza è la classe “Spagna” con una percentuale del 84.26% che tuttavia è un buon risultato rispetto al dataset a disposizione.

Per quanto riguarda il rapporto tra le istanze correttamente classificate per una classe e il numero totale di istanze correttamente classificate per ogni classe, ovvero la precisione, il target “Portogallo” si rivela quello con una percentuale. Questo permette di osservare come il modello classifica erroneamente un gran numero di campioni per tale classe e dimostra la bassa capacità del ViT di classificare bene i dati ad essa associati.

Tale risultato è dovuto, in particolar modo, al numero ridotto di esempi per questo target, con un numero maggiore infatti si riuscirebbe ad avviare a tale percentuale e a migliorare la precisione del modello.

Analogamente, anche il calcolo della recall rileva una situazione simile. Infatti, in questo caso per “Portogallo” e “Tunisia” si è ottenuto un risultato pari al 20% e al 33% rispettivamente, mentre è possibile constatare come il target con una recall più alta risulta essere “Spagna”, a differenza della precisione in cui la classe con un valore più alto è “Grecia”.

Da queste percentuali è facile affermare come tali classi influiscano negati-

vamente sulle prestazioni complessive del modello, mostrando come il ViT Small risulti poco capace di distinguere i dati relativi alle classi “Portogallo” e “Tunisia”.

Relativamente all’F1-Score è possibile notare come i valori ottenuti sono proporzionati e in linea con l’andamento generale, in particolare le classi maggiormente rappresentate nel dataset sono quelle in cui il modello ha prestazioni migliori con un F1-score relativamente alto vicino a 0.90, mentre quelle in cui il predittore ha peggiori performance corrispondono ai target per cui sono presenti un numero ridotto di osservazioni nel dataset.

Nel complesso, il modello presenta delle performance generalmente buone ma che vengono limitate dalle classi per cui sono presente un numero ridotto di campioni all’interno del dataset.

Infine, l’accuratezza generale del modello ottenuto corrisponde all’81,48%, un valore non troppo alto ma che relativamente alla quantità di dati a disposizione per il lavoro di tesi risulta essere una percentuale buona che dimostra come il modello implementato riesce a ben classificare le immagini e a distinguere tra le immagini del dataset, su cui è stato preaddestrato, e i grafici in input alla rete.

A seguire la matrice di confusione per il modello Vit Base che si ricorda differire dal Vit Small per complessità e dimensione: il primo è composto da 12 layer e circa 86 milioni di parametri, il secondo da 8 layer e con circa 28 milioni di parametri.

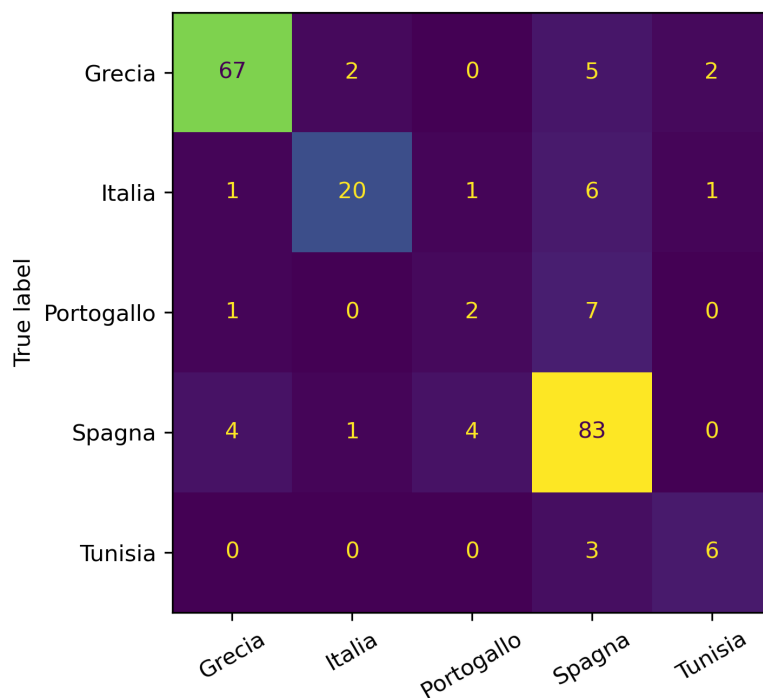


Figura 3.2: Matrice di confusione delle predizioni del modello ViT Base

Anche in questo caso per il modello ViT Small è presente un numero basso di falsi negativi e dunque è in grado di separare in buon modo i target evitando overfitting e risulta anch'esso capace di distinguere tra i grafici passati in input al modello e le immagini comuni di ImageNet.

I risultati ottenuti dalle metriche di valutazione delle prestazioni per questa variante del modello sono riassunti nella tabella seguente.

Metrica	Grecia	Italia	Portogallo	Spagna	Tunisia
<i>accuracy</i>	93.06%	94.44%	93.98%	86.11%	97.22%
<i>precision</i>	92%	87%	29%	80%	67%
<i>recall</i>	88%	69%	20%	90%	67%
<i>F1 – score</i>	0.90	0.77	0.24	0.85	0.67

Tabella 3.2: Metriche di valutazione del modello per ogni classe

In questo specifico caso le accuratèzze per ogni classe seguono lo stesso andamento dell’esperimento condotto su ViT Small ma con delle percentuali leggermente più alte. Questo comportamento è dato dalla diversa complessità e configurazione della rete, infatti la presenza di più layer di profondità si traduce nella possibilità di ottenere prestazioni più alte. Anche in questo caso la classe “Spagna” risulta quella con una percentuale più bassa pari all’86.11%, che tuttavia è comunque un buon valore a dimostrazione della bontà del modello.

Relativamente alla precisione, con il modello ViT Base la classe “Portogallo” continua ad essere quella in cui la rete ha una minore precisione. La particolarità è che, come è possibile osservare, si è ottenuta una percentuale sensibilmente più bassa rispetto alla variante Small come anche nel caso della classe “Grecia” seppur con una piccolissima variazione, mentre per gli altri target si è registrato un aumento della precisione. Dunque, si conferma la tendenza per cui un numero ridotto di esempi per alcuni target porta a una precisione più bassa rispetto alle classi ben rappresentate.

Per la recall il comportamento del modello ViT Base rispetto ad ogni classe rimane analogo con una variazione significativa per la classe “Tunisia”. Rispetto alla variante Small, questo modello incrementa la percentuale

di sensitivity di questo target fino ad arrivare al 67%, incremento che, seppur in piccola parte, riguarda anche le classi “Grecia” e “Spagna”. Dunque, il modello riesce ad effettuare una buona classificazione delle istanze positive in modo corretto mantenendo, anche in questo caso, un’influenza non differente nelle prestazioni complessive del modello.

Infine, l’F1-Score ottenuto per ogni target ha continuato a mantenere un andamento proporzionato a quello della precisione e della recall. È degno di nota evidenziare come anche in questo caso alcune classi presentano un valore di F1-score maggiore rispetto al modello ViT Small, nel dettaglio il target “Tunisia” è quello che risulta possedere un punteggio più alto rispetto all’esperimento precedente.

Il modello si dimostra, dunque, superiore al ViT Small anche se con poche migliorie, infatti dall’analisi dei risultati appena descritta è possibile notare come le prestazioni complessive del modello sono leggermente più alte, mantenendo ancora una volta un’importante contributo negativo da parte dei target sotto-rappresentati. L’accuratezza generale per questa variante del modello corrisponde all’82.41%, un valore leggermente incrementato rispetto al precedente esperimento.

In generale, alcuni risultati associati a determinate classi si sono dimostrati molto bassi in virtù del fatto che alcune di esse, come descritto nel capitolo precedente, presentano un numero di sample molto ridotto che dunque porta ad una conseguente riduzione delle prestazioni riguardo quel determinato target.

Considerando l’andamento generale e specifico per le classi della rete neurale utilizzata con entrambe le varianti, si può affermare che integrando ai dati nel dataset un numero maggiore di campioni si potrebbero dunque ottenere prestazioni più alte, nonché un’ottima accuratezza del modello.

Concludendo, la variante del modello ViT che meglio si adatta ai tipi di dati in input e che meglio li generalizza è quella Base che grazie a un numero più alto di layer e di parametri, rispetto alla variante Small, risulta ottenere prestazioni migliori ma che data la sua complessità potrebbe avere, tuttavia, un impatto in termini di costo computazionale della rete.

Conclusioni e Lavori Futuri

3.1 Conclusioni

Grazie al lavoro di questa tesi è stato possibile analizzare un nuovo approccio e un nuovo modello nell'ambito della predizione dell'origine geografica degli oli d'oliva. In particolar modo, considerando gli articoli revisionati su cui si è basato questo lavoro, è possibile affermare che sono stati utilizzati un approccio e tecniche di Deep Learning di cui, ad oggi, non disponiamo di conoscenze riguardo il loro impiego nel caso di studio di questa tesi.

Sono state riscontrate alcune difficoltà iniziali in quanto affrontare il problema della predizione dell'origine geografica degli oli d'oliva a partire dai grafici della loro composizione chimica, rappresenta un problema non poco complesso, il quale ha richiesto alcune conoscenze base di ciò che i grafici rappresentano e di come interpretare i grafici stessi.

Inoltre, è stata necessaria una particolare attenzione riguardo l'utilizzo di immagini rappresentanti grafici in reti neurali. Questi dati non sono comuni, ma sono più complessi dal punto di vista della determinazione delle feature più rilevanti per la modellazione del problema che si è proposto di risolvere.

Sono stati confrontati modelli di Deep Learning emergenti quali reti neurali MLP-Mixer e Vision Transformer insieme alla rete neurale moderna ResNet. I modelli elencati non sono stati ancora applicati al problema che questa tesi propone di risolvere come descritto nel capitolo dello stato dell'arte.

Nei casi riportati, infatti, la ricerca condotta si è sempre basata sull'utilizzo di dati scalari ottenuti dall'analisi chimica delle componenti degli oli oppure tramite l'estrazione delle feature dalle immagini relative ai grafici della composizione chimica dell'olio stesso.

Un altro contributo che questa tesi introduce è quello di effettuare uno studio basandosi sui grafici della composizione chimica di un prodotto. Infatti, allo stato attuale, un numero davvero molto esiguo di ricerche sono state condotte utilizzando come input di una rete neurale immagini utilizzate e descritte in questo lavoro di tesi.

Tramite questi approcci e queste tecniche di Deep Learning è stato possibile raggiungere delle prestazioni e dunque accuratezze relativamente buone rispetto al dataset a disposizione, le quali però sono influenzate negativamente da classi target sotto-rappresentate.

Come mostrato, i risultati si sono rivelati non troppo alti per le metriche di valutazione considerate. In alcuni casi relativi ad alcuni specifici target, sono state riscontrate performance basse ma, come descritto nel capitolo precedente, dovute a un numero ridotto di campioni associati alle classi per cui si sono ottenuti tali informazioni.

Per quest'ultima ragione, non è stato possibile dunque ottenere risultati migliori, ma considerando la problematica riguardante i dati a disposizione è possibile considerare i risultati ottenuti come un buon esito, infatti nel momento in cui si dispone di un alto numero di dati per l'addestramento e l'inferenza, si potrebbe arrivare ad avere un modello maggiormente performante.

Concludendo, il modello Vision Transformer (ViT), addestrato e testato sui dati forniti dall'Università di Firenze, risulta essere il modello più adatto per la predizione dell'origine geografica degli oli d'oliva alla base dell'individuazione e del contrasto delle frodi relative a questo prodotto d'eccellenza; nello specifico, risulta essere il modello che maggiormente generalizza senza rischiare overfitting e che meglio riesce ad operare con grafici.

Infine, anche se le prestazioni sono risultate non molto alte, tuttavia è ne-

cessario contestualizzare i risultati sulla base dei dati a disposizione, infatti aumentando il numero di osservazioni si può ottenere un conseguente aumento delle performance, che si traduce nella possibilità di ottenere un buon modello per il problema trattato.

3.2 Lavori Futuri

Diversi lavori futuri possono essere delineati a partire da questa tesi. In particolar modo, si potrebbe effettuare un'ulteriore analisi utilizzando i VOCs anziché gli idrocarburi. Si potrebbe testare il comportamento del modello utilizzato in questo lavoro con la stessa configurazione e confrontarlo con l'utilizzo di altri modelli di Deep Learning emergenti, allo scopo di osservare il loro comportamento e le loro capacità di generalizzazione.

Un altro lavoro futuro riguarda la predizione delle cultivar degli oli d'oliva. Si potrebbe considerare dunque una diversa classe target, a patto che si disponga di più dati che siano sufficientemente rappresentativi per tutte le classi. Dunque, si potrebbe raccogliere un maggior numero di campioni così da ottenere un buon numero di esempi per ogni classe, ovvero per ogni cultivar, e un minor divario tra i vari target.

In ultimo, si potrebbero reperire un maggior numero di dati relativi a più origini geografiche differenti e costruire un modello di Fraud Detection che, utilizzando il modello sviluppato e testato in questa tesi, possa determinare e avvertire della non conformità dell'olio in quanto adulterato al fine di contrastare la vendita al consumatore di un prodotto falsificato, a danno dei produttori e degli stessi consumatori.

Bibliografía

- [1] Carminati A. (2019), *Machine Learning Techniques to Improve Olive Oil Classification*, <https://www.politesi.polimi.it/bitstream/10589/149400/3/Carminati%20-%20MSc%20Thesis.pdf>
- [2] M.J. Lerma-García, J.M. Herrero-Martínez, G. Ramis-Ramos, E.F. Simó-Alfonso (2007), *Prediction of the genetic variety of Spanish extra virgin olive oils using fatty acid and phenolic compound profiles established by direct infusion mass spectrometry*, <https://www.sciencedirect.com/science/article/abs/pii/S0308814607012277>
- [3] Jorge Cervera-Gascó, Adrián Rabadán, Eulogio López-Mata, Manuel Álvarez-Ortí, José E. Pardo (2022), *Development of the POLIVAR model using neural networks as a tool to predict and identify monovarietal olive oils*, <https://www.sciencedirect.com/science/article/abs/pii/S0956713522004716>
- [4] Thet K., Woo N. (nd), *Gas Chromatography*, [https://chem.libretexts.org/Bookshelves/Analytical_Chemistry/Supplemental_Modules_\(Analytical_Chemistry\)/Instrumental_Analysis/Chromatography/Gas_Chromatography](https://chem.libretexts.org/Bookshelves/Analytical_Chemistry/Supplemental_Modules_(Analytical_Chemistry)/Instrumental_Analysis/Chromatography/Gas_Chromatography)
- [5] Woodford C. (nd), *Chromatography*, <https://www.explainthatstuff.com/chromatography.html>

-
- [6] (2022), *Gas Chromatography Explained: What It Is and How It Works*, <https://blog.perkinelmer.com/posts/gas-chromatography-explained-what-it-is-and-how-it-works/>
- [7] Ben (2022), *What is a Mass Spectrometer and How Does it Work?*, <https://www.excedr.com/blog/mass-spectrometer-function/>
- [8] Woodford C. (nd), *Mass Spectrometer*, <https://www.explainthatstuff.com/how-mass-spectrometers-work.html>
- [9] Cecchi L., Migliorini M., Giambanelli E., Rossetti A., Cane A., Mulinacci N., Melania F. (2020), *Authentication of the geographical origin of virgin olive oils from the main worldwide producing countries: A new combination of HS-SPME-GC-MS analysis of volatile compounds and chemometrics applied to 1217 samples*, <https://www.sciencedirect.com/science/article/abs/pii/S0956713520300724>
- [10] Pluskal T., Castillo S., Villar-Briones A., Oresic M. (2010), *MZmine 2: modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data*, <https://pubmed.ncbi.nlm.nih.gov/20650010/>
- [11] Panagiotis A. (nd), *Cross-Validation: K-Fold vs. Leave-One-Out*, <https://www.baeldung.com/cs/cross-validation-k-fold-looleave-one-out-cross-validation>
- [12] *PyTorch Documentation*, <https://pytorch.org/docs/stable/index.html>
- [13] *Scikit-Learn User Guide*, https://scikit-learn.org/stable/user_guide.html
- [14] *timm*, <https://huggingface.co/docs/timm/index>
- [15] Zhang A., Lipton C. Z., Li M., Smola J. A. (nd) *Dive into Deep Learning* <https://d2l.ai/index.html>

- [16] *CrossEntropyLoss*, <https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html?highlight=cross+entropy>
`torch.nn.CrossEntropyLoss`
- [17] Srinivasan V. A., (2019), *Stochastic Gradient Descent — Clearly Explained !!* <https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31>
- [18] *Stochastic Gradient Descent* <https://scikit-learn.org/stable/modules/sgd.html>
- [19] Tolstikhin I., Houlsby N., Kolesnikov A., Beyer L., Zhai X., Unterthiner T., Yung J., Steiner A., Keysers D., Uszkoreit J., Lucic M., Dosovitskiy A., (2021) *MLP-Mixer: An all-MLP Architecture for Vision* <https://arxiv.org/pdf/2105.01601.pdf>
- [20] He K., Zhang X., Ren S., Sun J., (2015) *Deep Residual Learning for Image Recognition* <https://arxiv.org/pdf/1512.03385.pdf>
- [21] Provino A., (2020) *ResNet CNN Networks — Deep Learning Engineer Italia* <https://andreaprovino.it/resnet/>
- [22] Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer ., Heigold G., Gelly S., Uszkoreit J., Houlsby N., (2020) *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* <https://arxiv.org/pdf/2010.11929.pdf>
- [23] Narkhede S., (2018) *Understanding Confusion Matrix* <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>
- [24] Brownlee J., (2016) *What is a Confusion Matrix in Machine Learning* <https://machinelearningmastery.com/confusion-matrix-machine-learning/>

Ringraziamenti

Mi è doveroso dedicare questo spazio del mio elaborato alle persone che hanno contribuito, con il loro instancabile supporto, alla realizzazione dello stesso.

In primis, un ringraziamento al mio relatore prof. Maurizio Gabbrielli, per il grande supporto e incoraggiamento, per le conoscenze trasmesse durante tutto il percorso di studi e di realizzazione dell'elaborato.

Desidero rivolgere un ringraziamento speciale al mio correlatore dott. Stefano Pio Zingaro per l'immensa pazienza, il suo infinito supporto, per i suoi indispensabili consigli, per avermi dato suggerimenti e per avermi guidato in tutto il percorso di svolgimento della tesi.

Ringrazio infinitamente i miei genitori e la mia famiglia che mi hanno sempre sostenuto, appoggiando ogni mia decisione, fin dalla scelta del mio percorso di studi e che da sempre mi sostengono nella realizzazione dei miei progetti. Non finirò mai di ringraziarvi per avermi permesso di arrivare fin qui.

Un grazie speciale ad Alessandro Schiavoni, nonostante l'abbia conosciuto solo recentemente, è stata una persona speciale che mi ha supportato, incoraggiato e mi è stato sempre accanto nelle sfide più ardue dell'ultimo periodo del percorso universitario.

Un grazie di cuore alla mia migliore amica e collega Alessandra Boccuto, con cui ho condiviso l'intero percorso universitario. Con la pazienza che la contraddistingue mi ha supportato e mi ha motivato nei momenti più difficili. Senza i suoi consigli, non ce l'avrei mai fatta.

Ringrazio i miei cari amici Beatrice Cipolletta per il suo supporto, per la motivazione e per il sostegno, Alessandro Giacchè e Andrea Di Ubaldo con i quali ho avuto il piacere di condividere momenti di studio e svolto progetti per gli esami, oltre che per la loro vicinanza e il loro supporto, come anche il mio caro amico Davide Monnati.

Infine, dedico questa tesi a me stesso, ai miei tanti sacrifici, al mio impegno e alla mia tenacia che mi hanno permesso di arrivare fin qui.