



CORSO DI LAUREA IN INFORMATICA

Nexus AI

[Link alla repository](#)

Autori:

Michele Antonio Annunziata
Vito Vernellati

Docenti:

Prof. Polese Giuseppe
Prof.ssa Caruccio Loredana

Indice

1	Introduzione al problema	3
1.1	Il Machine Learning e la medicina di precisione	3
1.2	La diagnosi precoce dei melanomi	3
1.3	Il problema degli approcci IA in ambito medico	3
1.4	Il problema affrontato	4
1.4.1	Specifiche PEAS	4
1.5	Il problema dell'explainability	4
1.5.1	In ambito clinico	5
1.6	Il problema dei dati mancanti e sbilanciati	5
2	Data understanding and data preparation	6
2.1	Panoramica delle opzioni disponibili	6
2.1.1	Approccio con sole immagini	6
2.1.2	Approccio con immagini e feature testuali	6
2.1.3	Approccio scelto e motivazioni	6
2.2	Data gathering	7
2.2.1	Dataset construction	7
2.2.2	Analisi dei dataset costruiti	8
2.3	Struttura della repository	9
2.4	Data preparation	10
2.4.1	Pulizia dei dati	10
2.4.2	Normalizzare i dati	10
2.4.3	Data augmentation	11
2.4.4	Bilanciamento delle classi	12
3	Sviluppo del modello	13
3.1	Quale modello scegliere	13
3.1.1	Due differenti approcci	13
3.2	Come valutare	14
3.3	Implementazione	14

4	Training e evaluation	15
4.1	Specifiche training	15
4.2	Metriche di valutazione	16
5	Usabilità	18
5.1	Leggi di Lehman	18
5.2	Interfaccia grafica	18
5.2.1	Piccolo ma importantissimo disclaimer	19
5.2.2	Perché Streamlit?	19
6	Conclusioni	20
6.1	Possibili miglioramenti	20
6.1.1	Espansione dei dati	20
6.1.2	Ottimizzazione di configurazione degli iperparametri	20
6.1.3	Altri aspetti limitanti	21
6.2	Considerazioni finali	21

Elenco delle figure

1	Bilanciamento delle classi iniziale dei dataset	8
2	Bilanciamento delle classi iniziale dei dataset	11
3	Studio dell'errore e dell'accuratezza all'aumentare delle epoche	16
4	Confusion Matrices	17
5	Interfaccia grafica	19

Elenco delle tabelle

1	Metadati utilizzati per lo splitting	8
2	Valori metriche di validazione prima epoca	16
3	Valori metriche di validazione ultima epoca	17

1 Introduzione al problema

1.1 Il Machine Learning e la medicina di precisione

Il tema della medicina di precisione con l'ausilio dell'intelligenza artificiale è una tematica sempre più trattata dalla comunità scientifica. L'obiettivo principale di molti studi è l'applicazione di metodi di Machine Learning per definire modelli computazionali in grado di supportare diagnosi cliniche e migliorare la prevenzione. In particolare, l'analisi di immagini mediche attraverso modelli "*intelligenti*" rappresenta una svolta significativa nel settore, consentendo di automatizzare processi diagnostici.

1.2 La diagnosi precoce dei melanomi

Nel contesto della dermatologia, il melanoma cutaneo rappresenta una delle sfide più critiche poiché la sua aggressività può risultare fatale se non diagnosticato precocemente. Seppur una diagnosi precoce aumenterebbe significativamente le possibilità di sopravvivenza, questa richiede competenze specialistiche e strumenti adeguati che non sempre sono disponibili in tutti i contesti clinici.

L'utilizzo di applicazioni che supportano tecnologie di Machine Learning rappresenta un supporto immediato, concreto e prezioso per migliorare l'identificazione delle lesioni cutanee sospette. Questi modelli permettono anche agli esperti di dominio che operano in contesti con risorse limitate, di accedere ad una diagnosi precoce con cui si distingue lesioni benigne e lesioni maligne.

Tutto ciò contribuisce alla riduzione di mortalità associata al melanoma e, più in generale, al miglioramento della qualità complessiva delle cure mediche.

1.3 Il problema degli approcci IA in ambito medico

Come per qualsiasi approccio di Machine Learning e Intelligenza Artificiale, l'applicazione in ambito medico presenta sfide specifiche.

Una delle principali difficoltà è la reperibilità di dati di alta **qualità**: le immagini di lesioni cutanee devono essere ben annotate e rappresentative della popolazione target per consentire un elevato livello di generalizzazione del modello. La dicotomia tra i dati utilizzati dal mondo della "*ricerca*" e i dati utilizzati dal mondo della "*produzione*" rende molti modelli inutilizzabili nell'effettivo, soprattutto a causa dell'eccessivo apporto di dati di sintesi.

Un altro aspetto cruciale è senza dubbio l'**explainability**: bisogna cercare di fornire agli utilizzatori del modello una qualche motivazione per cui l'algoritmo di Machine Learning ha fornito una certa

predizione. Ciò è fondamentale per conquistare la fiducia, e un certo grado di confidenza, degli utenti (medici e pazienti nella fattispecie).

1.4 Il problema affrontato

Il progetto **Nevus AI** mira a sviluppare un modello di Machine Learning in grado di classificare immagini di nei in due categorie principali: benigni e maligni. Partendo da un dataset composto da immagini dermatologiche ben annotate, l'obiettivo è creare un modello accurato, robusto e interpretabile, ponendo particolare enfasi sull'analisi dei dati e sulla trasparenza/comprensibilità.

Il progetto è corredato dalla repository disponibile qui.

1.4.1 Specifiche PEAS

Per consentire di inquadrare il sistema **Nevus AI** come un agente intelligente per l'analisi automatizzata delle immagini dermatologiche, si definiscono di seguito le specifiche del modello PEAS.

Performance. Le prestazioni del sistema vengono valutate utilizzando metriche standard per la classificazione. L'obiettivo è auspicabilmente ridurre i falsi positivi ed i falsi negativi.

Environment. L'ambiente in cui opera l'agente è costituito da un insieme di immagini di nei. Questo ambiente è **statico**, poiché le immagini fornite non cambiano nel tempo, ed è **osservabile**, **deterministico**, **sequenziale** e **continuo**.

Actuators. L'agente fornisce come output la classificazione dell'immagine caricata, indicando la probabilità che il neo sia benigno o maligno.

Sensors. L'agente percepisce l'ambiente elaborando le immagini caricate dagli utenti.

1.5 Il problema dell'explainability

Come già accennato precedentemente, uno dei problemi principali che rende inutilizzabili nella realtà dei modelli ottimi in letteratura, è l'explainability, la quale misura quanto il processo decisionale su cui si basa il modello sia comprensibile a chi lo utilizza. Maggior explainability comporta maggior fiducia, maggior comprensione, maggior benefici in ambito legale e maggior capacità di rilevare eventuali bias.

1.5.1 In ambito clinico

Un dermatologo che non riesce a comprendere perché un modello ha classificato una lesione come maligna difficilmente farà affidamento su di esso. Analogamente, un paziente potrebbe non accettare una diagnosi basata esclusivamente su una "scatola nera" senza una spiegazione convincente. Da ciò si evince quanto sia importante soffermarsi ed evidenziare il grado di explainability.

1.6 Il problema dei dati mancanti e sbilanciati

Un altro problema rilevante è sicuramente la disponibilità di dati di qualità. In ambito dermatologico, le immagini possono essere fortemente sbilanciate tra classi (ad esempio, potrebbero esserci più campioni di nei benigni ai maligni) o contenere informazioni incomplete. Questo porta inevitabilmente il modello a ridurre le sue capacità predittive. Naturalmente, la progettazione di modelli di Machine Learning capaci di generalizzare richiede una gestione rigorosa ed accurata dei dati iniziali.

2 Data understanding and data preparation

2.1 Panoramica delle opzioni disponibili

Durante la fase di analisi iniziale, sono stati identificati e considerati due approcci principali per sfruttare i dati raccolti nel contesto del problema.

2.1.1 Approccio con sole immagini

Questo approccio si basa esclusivamente sull'utilizzo delle immagini delle lesioni cutanee come input per il modello di classificazione. I dati visivi sono particolarmente indicativi per il riconoscimento delle caratteristiche morfologiche e dei pattern associati alle lesioni cutanee.

L'impiego di strumenti sofisticati di Machine Learning come le reti neurali convoluzionali rende questo approccio particolarmente efficace, poiché le immagini forniscono una base ricca di informazioni che possono essere sfruttate allo scopo di estrarre *automaticamente* le caratteristiche maggiormente rilevanti. L'esclusione di eventuali dati testuali tuttavia, potrebbe limitare la capacità di integrazione di aspetti contestuali e complementari che potrebbero arricchire l'analisi.

2.1.2 Approccio con immagini e feature testuali

In questo approccio, oltre alle immagini, vengono utilizzate feature testuali come input aggiuntivi per il modello. Tali informazioni possono includere generiche annotazioni descrittive che servono come supporto al miglioramento delle capacità predittive del modello.

Tuttavia, l'uso di dati testuali può presentare problematiche significative, quali incompletezza, incoerenza o mancanza di standardizzazione, le quali possono inevitabilmente distorcere il contenuto informativo introducendo rumore e compromettendo l'efficacia dei risultati ottenuti.

2.1.3 Approccio scelto e motivazioni

Dopo un'attenta valutazione delle opzioni disponibili ed un'analisi della proporzione costo-benefici, è stato deciso di adottare l'**approccio con sole immagini**.

La scelta è stata motivata principalmente dai seguenti fattori:

- **qualità delle immagini:** le immagini presenti nelle varie collezioni sono di alta qualità e rappresentano un punto di partenza solido per l'analisi visiva approfondita delle lesioni cutanee.

Concentrare gli sforzi computazionali sull'elaborazione di immagini consente una focalizzazione sui pattern visivi, i quali rappresentano il principale elemento diagnostico.

- **ottimizzazione delle risorse e dei tempi:** la pipeline basata su sole immagini risulta essere maggiormente semplice e scalabile rispetto ad una pipeline multimodale. Ciò consente di rispettare i vincoli di tempo e risorse stabiliti in fase di pianificazione.
- **limitazioni dei dati testuali:** Le informazioni testuali disponibili sono incomplete e non sempre standardizzate, il che potrebbe introdurre ambiguità o rumore nei dati. Questa mancanza di coerenza ha influenzato la scelta di evitare l'integrazione di dati testuali.

2.2 Data gathering

Il dataset utilizzato e le risorse necessarie per proseguire con il completamento del progetto sono stati raccolti dalla piattaforma **ISIC Archive** disponibile qui.

In particolare sono stati inizialmente esaminati ed esplorati i dataset presenti nelle collezioni "*BCN20000*" (disponibile qui) e "*HAM10000*" (disponibile qui).

Entrambi i dataset includono due tipi di dati:

- immagini delle lesioni cutanee. Tutta la raccolta *HAM10000* presenta immagini di dimensioni 600×450 , mentre quelle nel dataset *BCN20000* sono di dimensioni 1024×1024 ;
- metadati testuali che forniscono informazioni aggiuntive sulle caratteristiche cliniche e contestuali delle lesioni.

Nelle sezioni successive saranno riportate tutte le scelte adottate con le relative motivazioni.

2.2.1 Dataset construction

Per rispettare l'approccio scelto, descritto nelle sezioni precedenti, è stato deciso di costruire internamente un dataset contenente esclusivamente immagini di lesioni cutanee, selezionate dai dataset *HAM10000* e *BCN20000*.

La costruzione dei dataset è stata effettuata tramite uno script python sviluppato internamente. Lo script ha eseguito le seguenti operazioni:

- **lettura dei metadati** per ciascuno dei due dataset. In Tabella 1 sono riportate le feature esaminate per lo splitting.

<i>isic_id</i>	<i>diagnosis₁</i>
----------------	------------------------------

Tabella 1: Metadati utilizzati per lo splitting

- **creazione delle cartelle annotate:** per ciascun dataset, sono stati creati tre gruppi principali di immagini basati sulle etichette fornite.

1. **nei_benigni:** contenente le immagini classificate come benigni.
2. **nei_maligni:** contenente le immagini classificate come maligni.
3. **nei_indeterminati:** contenente le immagini classificate come indeterminate.

Il risultato finale ottenuto è la creazione di due cartelle distinte contenente le immagini presenti nei dataset "*HAM10000*" e "*BNC20000*".

2.2.2 Analisi dei dataset costruiti

Dopo la costruzione dei dataset mediante lo script che ha suddiviso le immagini in base alle etichette presenti nei metadati, è stata effettuata un'analisi dettagliata della distribuzione delle classi in ciascun dataset. I risultati sono riassunti in Figura 1.

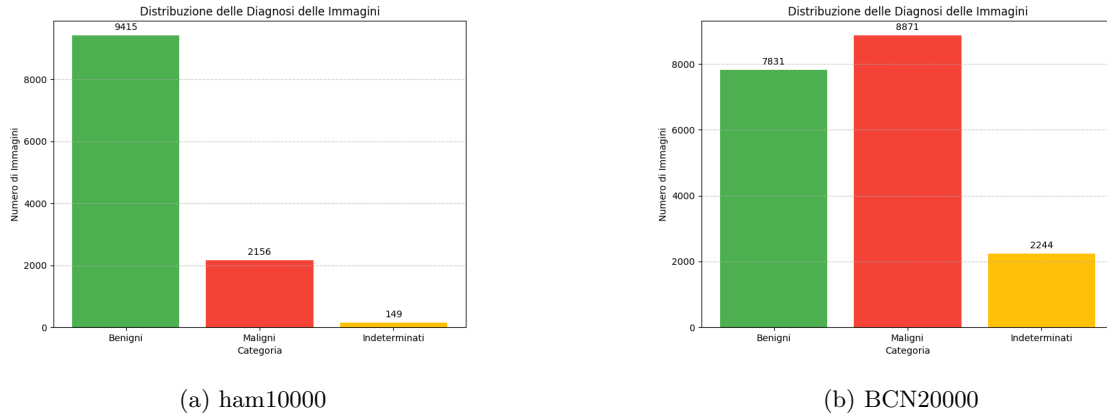


Figura 1: Bilanciamento delle classi iniziale dei dataset

L'analisi della distribuzione ha evidenziato una significativa differenza nel bilanciamento delle classi tra i due dataset. Nel dataset *HAM10000*, la classe dei nei benigni rappresenta la maggioranza, con una proporzione di circa il 80.5% rispetto al 18.5% dei nei maligni. Inoltre circa l'1% è rappresentato da immagini con etichetta "*indeterminato*". Questa forte discrepanza potrebbe influire negativamente

sull'addestramento del modello, portandolo a favorire previsioni sulla classe dominante, con conseguenti problemi di generalizzazione.

Al contrario, nel dataset *BCN20000*, la distribuzione delle classi è più bilanciata, con una proporzione circa del 41.5% di nei benigni e del 47% di nei maligni. Inoltre circa l'11.5% è rappresentato da immagini con etichetta "*indeterminato*". Questo bilanciamento offre un punto di partenza più solido per l'addestramento del modello, riducendo la necessità di applicare tecniche di riequilibrio e riducendo il divario tra classe maggioritario e minoritaria.

Il problema delle classi non bilanciate. In generale, quando non si hanno abbastanza campioni per discriminare delle classi differenti, si è in una situazione sgradevole poiché, così facendo, le capacità di generalizzare del modello vanno a decrescere per le classi minoritarie. Una situazione ideale sarebbe quella in cui tra le classi ci sia un totale bilanciamento in termini di numero di campioni.

Scelta effettuata. Sulla base delle analisi eseguite, è stato scelto di utilizzare il dataset costituito dalle immagini presenti in *BCN20000*. Sono riportate di seguito le motivazioni principali.

- **Bilanciamento delle classi.** La distribuzione delle immagini nei benigni e maligni nel dataset *BCN20000* è maggiormente equilibrata rispetto al dataset *HAM10000*, è ciò consente di ridurre notevolmente il rischio di bias verso la classe dominante.
- **Rappresentatività dei dati maligni.** La maggiore presenza di immagini di nei maligni nel dataset costruito a partire da *BCN20000* consente al modello di apprendere più efficacemente le caratteristiche distintive di queste lesioni, migliorando la sensibilità diagnostica.

Questa decisione mira a garantire che il modello possa esser robusto ed il più capace possibile a discernere le classi.

2.3 Struttura della repository

La repository è organizzata in modo da garantire una chiara separazione logica tra dati, codice e documentazione. Di seguito viene descritta la struttura delle principali cartelle e file presenti:

- *dataset*: contiene le sottocartelle relative ai dataset presi in esame. Ognuna di queste sottocartelle presenta una cartella *diagrams* contenente i grafici e rappresentazioni statistiche relative ai dati, ed una cartella *metadata* utilizzata per lo splitting;

- *docs*: raccoglie la documentazione e la presentazione relativa al progetto;
- *scripts*: contiene tutti gli script sviluppati per il progetto. In particolare lo script per lo splitting del dataset (*data_splitting.py*), lo script per il training (*training.py*) e lo script utilizzato per creare l'interfaccia grafica (*ui.py*);
- *README.md*: fornisce informazioni su come installare, configurare ed utilizzare il progetto;
- *requirements.txt*: contiene la lista delle dipendenze necessarie per eseguire gli script della repository, facilitando la configurazione dell'ambiente di sviluppo.

Questa organizzazione consente una gestione efficiente del progetto, facilitando la manutenibilità e l'eventuale estensione delle funzionalità. La repository è disponibile qui.

2.4 Data preparation

2.4.1 Pulizia dei dati

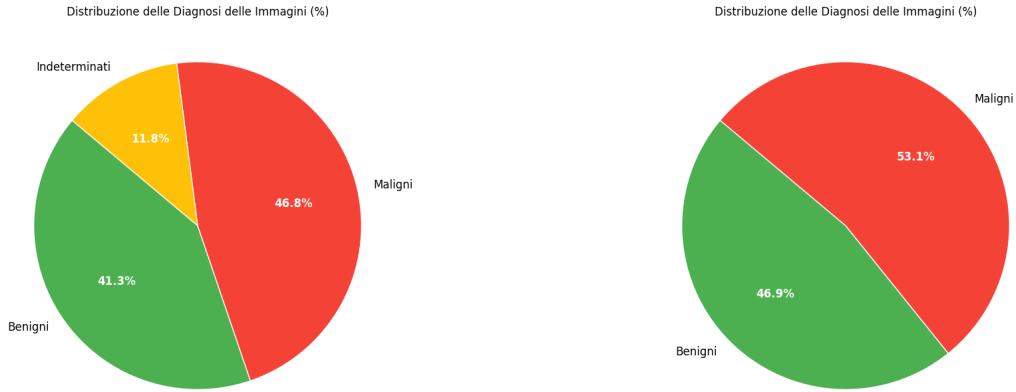
Pulire i dati per renderli utilizzabili da un modello di Machine Learning è un passo fondamentale per garantire qualità e affidabilità. Pulizia dei dati significa capire in che modo sopravvivere alla mancanza o al rumore dei dati di partenza.

Nel contesto di questo progetto, sono state adottate le seguenti misure:

- **Esclusione delle immagini indeterminate.** Tutte le immagini presenti nella cartella denominata "nei_indeterminati" sono state scartate. Ciò perché le etichette "indeterminate" non forniscono alcune informazioni utili per l'addestramento del modello e potrebbero introdurre ambiguità durante la fase di apprendimento.
- **Verifica delle immagini corrotte.** Le immagini non leggibili o corrotte sono state eliminate per evitare errori durante il caricamento e la fase di pre-elaborazione (tale operazione è stata effettuata tramite lo script python per lo splitting delle cartelle citato nelle sezioni precedenti)

2.4.2 Normalizzare i dati

Generalmente, siccome i domini delle caratteristiche sono differenti, un passo fondamentale è quello di modificare i valori in scala per evitare che le scale più ampie possano monopolizzare la fase di apprendimento. Attributi con grosse differenze di scala provocano problemi per il modello, il quale potrebbe sottostimare/sovrastimare la rilevanza di una caratteristica.



(a) BCN20000 considerando "indeterminati"

(b) BCN20000 escludendo "indeterminati"

Figura 2: Bilanciamento delle classi iniziale dei dataset

Nel caso di Nevus AI. Non essendo presenti feature numeriche tradizionali, la fase di feature scaling è stata applicata esclusivamente alle immagini, normalizzando direttamente i valori dei pixel. I valori sono stati scalati nell'intervallo $[0, 1]$ dividendo ciascun pixel per 255.

Questo processo migliora la stabilità numerica durante l'addestramento del modello e rende il dataset coerente per le tecniche di deep learning.

2.4.3 Data augmentation

Generalmente nei problemi di visione artificiale, un numero ridotto di immagini può portare a fenomeni di overfitting, in cui il modello si adatta eccessivamente ai dati di training, perdendo capacità di generalizzazione su nuovi esempi.

Per affrontare questo problema, viene adottata una tecnica fondamentale chiamata Data Augmentation. Essa consiste nell'applicare trasformazioni casuali alle immagini del dataset, generando nuove varianti dei dati esistenti senza modificarne l'etichetta. Questo processo migliora la capacità del modello di generalizzare e rende la rete neurale più robusta alle variazioni reali delle immagini, come illuminazione, angolazione e scala.

In particolare nel contesto di Nevus AI sono state apportate operazioni di

1. *rotazione casuale* per ruotare le immagini fino a 30° e simulare diverse angolazioni.

2. *traslazione orizzontale e verticale* per spostare le immagini lungo gli assi delle ascisse e delle ordinate fino al 20% della loro dimensione.
3. *shear transformation* per applicare una distorsione affine e simulare variazioni prospettiche.
4. *zoom* per ingrandire o ridurre l'immagine fino al 20% e simulare distanze differenti.
5. *flip orizzontale* per invertire casualmente le immagini orizzontalmente.

2.4.4 Bilanciamento delle classi

Come già detto nelle sezioni precedenti, lo squilibrio tra le classi è un problema comune nei dataset. In Figura 2 è mostrata la distribuzione delle classi del dataset ricavato dalle immagini in "BCN20000".

Per affrontare questo problema e migliorare le capacità di Nei problemi di classificazione standard, una tecnica comune per affrontare lo squilibrio è l'uso di **SMOTE** (*Synthetic Minority Oversampling Technique*), che genera nuovi campioni sintetici interpolando i dati esistenti della classe minoritaria. Tuttavia, SMOTE è progettato principalmente per dati tabellari e numerici e non si adatta bene alle immagini.

SMOTE in Nevus? SMOTE crea nuovi campioni interpolando dati numerici tra punti vicini nello spazio delle feature. Nel caso delle immagini, questa interpolazione non conserva le caratteristiche strutturali, dunque la generazione sintetica potrebbe creare artefatti che confondono il modello invece di aiutarlo a generalizzare meglio.

In sintesi, SMOTE non è in grado di generare nuove immagini coerenti con la distribuzione originale, al contrario delle tecniche di Data Augmentation.

3 Sviluppo del modello

3.1 Quale modello scegliere

Il task in esame, la classificazione delle lesioni cutanee, richiede un approccio robusto e accurato per distinguere tra lesioni benigne e maligne. Per tale task è stato deciso di adottare una tipologia di architettura allo stato dell'arte per lavorare con le immagini, ossia le CNN (*Convolutional Neural Network*).

3.1.1 Due differenti approcci

Considerando l'efficacia delle reti neurali convoluzionali (CNN) nell'elaborazione delle immagini, sono state esplorate due opzioni principali:

- **starting from scratch**, un approccio che comporta la costruzione da zero di una CNN, progettata specificamente per il dataset di lesioni cutanee. Richiede un numero significativo di dati e tempo di addestramento per ottenere risultati competitivi;
- **sequential transfer learning**, una tecnica in cui un modello pre-addestrato in un dominio D_s su un task T_s viene riutilizzato, adattato o perfezionato per risolvere un nuovo task T_t in un dominio D_t . Questo metodo permette di risparmiare tempo e risorse, e di trasferire le conoscenze acquisite da task generici di classificazione delle immagini, con risultati spesso più rapidi e accurati.

Approccio scelto. Dopo un'analisi comparativa, è stato deciso di adottare l'approccio di sequential transfer learning. Questa scelta è motivata dall'alta qualità delle reti preaddestrate disponibili e dalla loro comprovata efficacia in contesti simili.

Fasi principali. Il sequential transfer learning si basa su un adattamento progressivo da un task sorgente a un task target, seguendo diverse fasi fondamentali:

- 1) Il modello viene addestrato su un dataset sorgente *grande e generico*, così da formare la base di ciò che si deve apprendere.
- 2) Successivamente il modello pre-addestrato viene ulteriormente raffinato utilizzando un dataset target più specifico.

2.1) **Congelamento** dei layer. I primi strati della rete che costituiscono le caratteristiche generali vengono congelati per preservare le conoscenze già acquisite.

2.2) **Fine-tuning**. Gli strati finali vengono ulteriormente addestrati con i dati target per apprendere caratteristiche più specifiche. Durante questa fase i pesi di questi strati vengono aggiornati per affinare il modello e migliorare la precisione per il compito specifico.

3.2 Come valutare

La valutazione dei modelli di classificazione è cruciale per garantire la loro affidabilità e precisione. È stato scelto di utilizzare un *training-test split* secondo il **principio di Pareto** 80 – 20. In particolare

- l'80% delle immagini è stato utilizzato per l'addestramento del modello (training set), consentendo alla rete neurale di apprendere le caratteristiche delle lesioni cutanee.
- Il 20% delle immagini è stato riservato per il test (*testset*), fornendo una valutazione imparziale delle prestazioni su dati mai visti durante l'addestramento.

Le metriche di valutazione includeranno l'**accuracy**, la **precision**, la **recall**, la metrica **F1-Score** e la **confusion matrix**.

3.3 Implementazione

Per l'implementazione del modello e delle metriche di valutazione, sono state utilizzate le seguenti librerie:

- **TensorFlow/Keras**, per la creazione, addestramento e valutazione delle CNN.
- **scikit-learn**, per il pre-processing dei dati e la gestione della cross-validation.
- **pandas** e **numpy**, per il caricamento e la manipolazione dei dataset.
- **matplotlib** e **seaborn**, per la creazione di grafici e visualizzazioni delle metriche.
- **streamlit**, per l'implementazione dell'interfaccia grafica.

4 Training e evaluation

4.1 Specifiche training

L'addestramento di una rete neurale profonda richiede una serie di scelte strategiche per ottimizzare le prestazioni del modello e garantire una buona generalizzazione sui dati di test. Nel nostro caso, utilizziamo un approccio basato su transfer learning, sfruttando una rete pre-addestrata (**NASNetMobile**) e adattandola al nostro specifico compito di classificazione.

Poiché NASNetMobile è stato addestrato su un vasto dataset di immagini generiche (ImageNet), la rete ha già appreso una serie di caratteristiche di base, come bordi, texture e forme. Tuttavia, per il nostro problema specifico, abbiamo bisogno di adattare queste conoscenze al riconoscimento delle lesioni cutanee.

L'addestramento del modello si è basato su 20 epoche, un numero scelto per garantire un equilibrio tra tempo di addestramento e performance. Un numero inferiore di epoche potrebbe non consentire al modello di apprendere sufficientemente, mentre un numero troppo elevato rischierebbe di causare **overfitting**, ovvero un adattamento eccessivo ai dati di training che ridurrebbe la capacità di generalizzazione del modello su dati mai visti. Inoltre, per mitigare il rischio di overfitting, è stato implementato un meccanismo di **early stopping**, che interrompe l'addestramento se il miglioramento sulla validazione si arresta per diverse epoche consecutive.

La configurazione del modello è costituita come segue:

- **congelamento** dei pesi dei primi strati della rete, mantenendo le caratteristiche generali apprese dal modello di base.
- **aggiunta** di nuovi layer personalizzati, tra cui un livello di Flattening per trasformare i dati in un formato compatibile con i livelli completamente connessi, un livello Dense con attivazione ReLU, che introduce una non linearità nel modello, consentendo di apprendere rappresentazioni più complesse ed un livello di output con attivazione Softmax, che restituisce le probabilità di appartenenza alle due classi (neo benigno o maligno).
- **compilazione e ottimizzazione** del modello.
 - Metodo di ottimizzazione avanzato basato sulla discesa del gradiente adattiva, chiamato **ottimizzatore Adam**, il quale accelera la convergenza e stabilizza l'aggiornamento dei pesi.

- Si utilizza la funzione di **perdita binary crossentropy**, la quale è particolarmente adatta ai problemi di classificazione binaria e permette di valutare quanto la probabilità predetta per ogni classe si avvicina al valore reale.
- Si utilizza il parametro dell'**accuratezza** per monitorare le prestazioni del modello.

In Figura 3 è mostrato l'andamento della training loss, della validation loss, della training accuracy e della validation accuracy epoca dopo epoca. L'obiettivo ideale è osservare una decrescita costante della loss e un aumento dell'accuratezza, sia per il training set che per il validation set.



Figura 3: Studio dell'errore e dell'accuratezza all'aumentare delle epoche

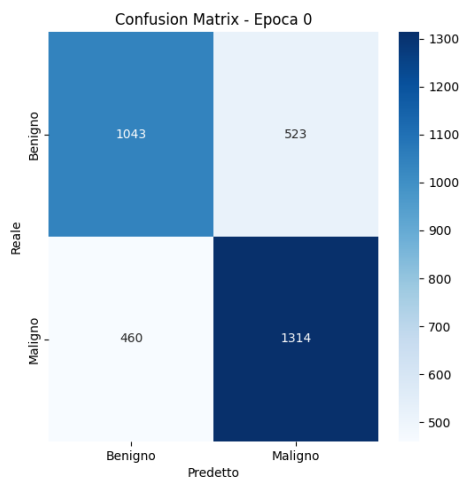
4.2 Metriche di valutazione

In Tabella 2 sono riportate le prestazioni del modello rispetto la prima epoca:

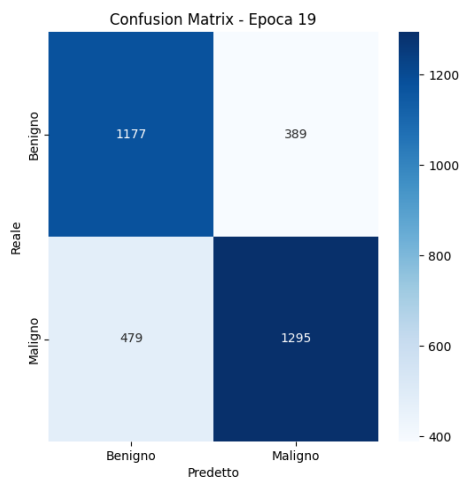
Metrica	Formula	Risultato
<i>precision</i>	$\frac{TP}{TP+FP}$	0.66
<i>recall</i>	$\frac{TP}{TP+FN}$	0.69
<i>f1 - score</i>	$\frac{2 \cdot precision \cdot recall}{precision + recall}$	0.67
<i>accuracy</i>	$\frac{TP+TN}{TP+TN+FP+FN}$	0.70

Tabella 2: Valori metriche di validazione prima epoca

In Tabella 3 sono riportate le prestazioni del modello rispetto l'ultima epoca:



(a) Confusion Matrix prima epoca



(b) Confusion Matrix ultima epoca

Figura 4: Confusion Matrices

Metrica	Formula	Risultato
<i>precision</i>	$\frac{TP}{TP+FP}$	0.75
<i>recall</i>	$\frac{TP}{TP+FN}$	0.71
<i>f1 - score</i>	$\frac{2 \cdot precision \cdot recall}{precision + recall}$	0.73
<i>accuracy</i>	$\frac{TP+TN}{TP+TN+FP+FN}$	0.74

Tabella 3: Valori metriche di validazione ultima epoca

5 Usabilità

L'usabilità di un sistema software, e in particolare di un modello di intelligenza artificiale, è un aspetto cruciale che ne determina il valore pratico. Un modello accurato, ma difficile da integrare, rischia di perdere gran parte del suo potenziale impatto.

5.1 Leggi di Lehman

Nel campo dell'ingegneria del software, le leggi di Lehman forniscono un quadro teorico utile per comprendere come i sistemi software si evolvono nel tempo. In linea di principio, occorre capire come mantenere nel tempo lo stato del modello. Un sistema software che non cambia nel tempo è inutile.

Due di queste leggi sono particolarmente rilevanti per il rilascio e l'usabilità di un modello di machine learning:

- **cambiamento continuo:** un sistema software non può rimanere statico, e per rimanere utile deve adattarsi ai cambiamenti dei requisiti, delle tecnologie e dell'ambiente operativo. Nel caso di un modello di deep learning, questo significa che deve essere aggiornato con nuovi dati e tecniche per mantenere la sua accuratezza e rilevanza.
- **complessità crescente:** man mano che un sistema viene modificato ed esteso, tende a diventare più complesso, a meno che non vengano adottate strategie specifiche per gestire tale crescita. Questo principio evidenzia l'importanza di un'architettura modulare e di un'interfaccia ben progettata, in modo da mantenere il modello facilmente utilizzabile e aggiornabile.

5.2 Interfaccia grafica

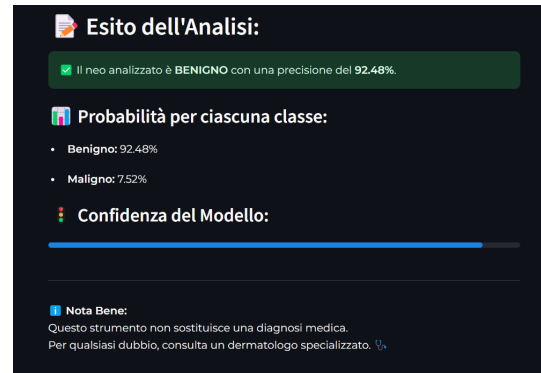
Per facilitare l'interazione con il modello addestrato ed, in generale, rendere, per quanto possibile, migliore l'interazione tra l'utente finale e la "macchina" è stata sviluppata un'interfaccia grafica utilizzando Streamlit, un framework open-source che permette di creare applicazioni web interattive in modo rapido e semplice.

L'interfaccia realizzata consente agli utenti di:

1. caricare un'immagine da analizzare.
2. eseguire la classificazione utilizzando il modello addestrato.
3. visualizzare il risultato in modo chiaro e comprensibile.



(a) Caricamento dell'immagine



(b) Visualizzazione esito dell'analisi

Figura 5: Interfaccia grafica

5.2.1 Piccolo ma importantissimo disclaimer

È importante sottolineare che i risultati forniti dal modello non sono da considerarsi una diagnosi medica. Il modello è stato addestrato su un dataset specifico e, per quanto accurato, può commettere errori. L'output deve essere interpretato come un supporto e non come un verdetto definitivo.

5.2.2 Perché Streamlit?

Streamlit è stato scelto per diversi motivi:

- **Semplicità d'uso e velocità** di sviluppo. Permette di costruire interfacce grafiche con poche righe di codice, senza la necessità di sviluppare una web app complessa, ed essendo pensato per applicazioni di data science, si integra perfettamente con modelli di machine learning.
- **Interattività** immediata. Offre componenti predefiniti per il caricamento di immagini e l'integrazione con i risultati del modello.

6 Conclusioni

Lo sviluppo di questo progetto ha permesso di esplorare e applicare tecniche di deep learning per la classificazione delle lesioni cutanee, dimostrando l'efficacia del transfer learning e l'importanza di una pipeline ben strutturata per un task di machine learning. Tuttavia, come in ogni progetto, vi sono margini di miglioramento e alcune limitazioni da considerare per future evoluzioni.

6.1 Possibili miglioramenti

L'attuale implementazione rappresenta una solida base per l'analisi delle immagini dermatologiche, ma diverse strategie potrebbero essere adottate per migliorarne le prestazioni e l'usabilità.

6.1.1 Espansione dei dati

Uno dei passi più significativi per migliorare le capacità di generalizzazione del modello è l'ampliamento del dataset. Attualmente, il modello è stato allenato su immagini provenienti da dataset specifici, ma sarebbe utile integrare dati aggiuntivi da altre fonti, garantendo una maggiore diversità nelle immagini.

In particolare, includere dataset più eterogenei potrebbe aiutare a migliorare la robustezza del modello rispetto a variazioni di illuminazione delle immagini, tipologie di pelle e dispositivi di acquisizione delle immagini.

Caratteristiche testuali. Attualmente, il modello si basa esclusivamente sulle informazioni contenute nelle immagini, ma un'estensione naturale potrebbe consistere nell'integrazione di dati testuali. Tale integrazione, attraverso modelli multimodali che combinano visione artificiale e **NLP** (Natural Language Processing), potrebbe migliorare significativamente le prestazioni del modello.

6.1.2 Ottimizzazione di configurazione degli iperparametri

Attualmente, la configurazione del modello è stata scelta sulla base di buone pratiche e di test empirici. Tuttavia, per ottenere migliori prestazioni, si potrebbe esplorare l'ottimizzazione degli iperparametri attraverso tecniche di **ricerca automatica**.

L'ottimizzazione potrebbe riguardare aspetti come numero di neuroni da utilizzare nei nuovi layer, numero di epoche ottimale e tecniche di regolarizzazione per ridurre l'overfitting.

Limiti computazionali. Uno dei principali vincoli è stato rappresentato dalle risorse hardware disponibili. Nonostante l'uso di Google Colab, l'addestramento del modello ha richiesto diverse ore, limitando la possibilità di eseguire esperimenti approfonditi con diverse configurazioni di rete e iperparametri. L'uso di GPU più potenti o cluster di calcolo distribuito potrebbe ridurre significativamente i tempi di training, consentendo di testare più configurazioni in meno tempo e migliorare la performance del modello.

6.1.3 Altri aspetti limitanti

Come accennato in precedenza, il modello non è infallibile. Ciò è dovuto alla black-box nature delle reti neurali che rende difficile interpretare le decisioni del modello. Ciò rappresenta un grande equivoco per quanto riguarda l'**explainability**.

Inoltre, nonostante le tecniche adottate (data augmentation, early stopping), il modello potrebbe aver appreso caratteristiche specifiche del dataset di addestramento, riducendo la sua efficacia su immagini reali non viste (**overfitting**).

Infine, per garantire una maggiore accessibilità, il modello potrebbe essere distribuito su piattaforme cloud in modo da permettere l'elaborazione di immagini su larga scala e da più utenti contemporaneamente (**scalabilità**).

6.2 Considerazioni finali

Questo progetto ha dimostrato la fattibilità dell'utilizzo del deep learning per task di classificazione in ambito medico, evidenziando sia i suoi punti di forza che le sue principali limitazioni.

L'utilizzo pratico del modello richiede ulteriori miglioramenti, sia in termini di progettazione, che di accuratezza, usabilità e accessibilità.

Il progetto pone le basi per possibili applicazioni in ambito medico, sebbene sia fondamentale ricordare che modelli di questo tipo devono sempre essere utilizzati come **supporto alla diagnosi** e **NON** come strumento decisionale autonomo.