UNIVERSITÀ
POLITECNICA
DELLE MARCHE

Facoltà di Ingegneria
Corso di Laurea in Ingegneria Biomedica

Sviluppo di algoritmi di ML per la classificazione di immagini ottenute da test per la valutazione del danno genotossico

Development of machine learning algorithms for the classification of images obtained from tests to assess genotoxic damage

Candidato:
Maria Cecilia Morigi

Relatore:
Prof. Alessandro Nardi

Correlatore:
Prof. Giuseppe d'Errico

Anno Accademico 2025-2026

UNIVERSITÀ
POLITECNICA
DELLE MARCHE

Facoltà di Ingegneria
Corso di Laurea in Ingegneria Biomedica

Sviluppo di algoritmi di ML per la classificazione di immagini ottenute da test per la valutazione del danno genotossico

Development of machine learning algorithms for the classification of images obtained from tests to assess genotoxic damage

Candidato:
Maria Cecilia Morigi

Relatore:
Prof. Alessandro Nardi

Correlatore:
Prof. Giuseppe d'Errico

Anno Accademico 2025-2026

Università Politecnica delle Marche
Facoltà di Ingegneria
Corso di Laurea in Ingegneria Biomedica
Via Brecce Bianche – 60131 Ancona (AN), Italy

Indice

Elenco delle figure

Listings

Introduzione

La valutazione dell'integrità del materiale genetico rappresenta uno dei pilastri fondamentali della tossicologia moderna, della diagnostica biomedica e della ricerca oncologica. Difatti, agenti chimici e/o fisici, possono indurre il danneggiamento del materiale genetico per cause endogene o esogene. Tra le metodologie biochimiche impiegate per stimare quantitativamente le rotture a singolo o doppio filamento del DNA, il saggio della cometa (o Comet assay), occupa una posizione di preminenza grazie alla sua elevata sensibilità, versatilità e relativo basso costo operativo. Questa tecnica si basa sulla carica elettrica negativa del DNA, che, sottoposto ad un campo elettrico in un gel di agarosio, migra verso l'anodo; nuclei integri migreranno in maniera compatta, mentre nuclei con DNA soggetto a rotture del filamento migreranno determinando la caratteristica struttura morfologica che ricorda la forma di una cometa, da cui il nome della tecnica. Mediante fluorocromi è poi possibile marcare il DNA e rilevare quantitativi di materiale nella "testa" e nella "coda", più o meno estesa.

Nonostante l'efficacia ampiamente documentata in letteratura, la metodologia di analisi convenzionale risente di significativi limiti operativi. Tradizionalmente, la classificazione del livello di danno genotossico viene delegata all'ispezione visiva al microscopio a fluorescenza da parte di un operatore umano, il quale assegna ogni singola cellula a una delle cinque classi di danno previste dai protocolli internazionali. Questo approccio manuale introduce una forte componente di soggettività, specialmente nella rilevazione dei confini morfologici tra classi adiacenti lievi. Inoltre, il processo risulta molto lungo e penalizzato dall'affaticamento visivo dell'operatore, limitando la riproducibilità statistica dei test e configurandosi come un evidente collo di bottiglia per le attività di laboratorio ad alta produttività. Queste tecniche sono oggi affiancate da metodi quantiativi di analisi di immagine, che comunque risentono di un elevato livello di soggettività interoperatore.

In questo scenario tecnologico e clinico si inserisce il presente lavoro di tesi, sviluppato nell'ambito delle attività di tirocinio computazionale e sperimentale. L'obiettivo primario dell'elaborato consiste nella progettazione, ottimizzazione e validazione di una pipeline software automatizzata in linguaggio Python, basata sull'impiego di algoritmi di Intelligenza Artificiale e, nello specifico, di Reti Neurali Convoluzionali, per la classificazione oggettiva e automatica delle immagini di nuclei sottoposti al test della cometa.

Il flusso di lavoro ha previsto una combinazione tra la fase sperimentale di laboratorio e la modellizzazione informatica. Una parte del progetto è stata dedicata alla

conduzione di sessioni di acquisizione ottica diretta al microscopio per l'espansione del dataset, permettendo di addestrare l'architettura su una solida base statistica di oltre mille campioni. I risultati ottenuti sono stati infine analizzati non solo mediante metriche quantitative standard, ma anche attraverso un approccio qualitativo, volto a investigare le cause biologiche e ottiche alla base delle misclassificazioni del modello.

Capitolo 1

Fondamenti Biologici e Strumenti Computazionali

1.1 Il danno genotossico

La molecola di DNA è un polimero a doppia elica composto da nucleotidi, ciascuno dei quali presenta una base di zucchero e fosfato, legato a una base azotata. Essa è la sede dell'informazione genetica che coordina la vita e i meccanismi cellulari.

L'integrità strutturale del materiale genetico può essere alterata da cause endogene, che derivano dal normale logoramento fisico e dal metabolismo della cellula stessa, o da cause esogene, come agenti chimici, fisici o biologici (es. raggi X, agenti alchilanti). Tale capacità è definita in ambito biologico e tossicologico, come genotossicità e agenti in grado di causare questo danno sono definiti agenti genotossici [14].

Questi, possono aggredire la struttura del DNA in punti diversi, generando lesioni con caratteristiche fisiche diverse, tra cui le più rilevanti sono le rotture a singolo e doppio filamento, che spezzano nettamente la continuità fisica del cromosoma. La severità di tali lesioni è variabile, ed è indfluenzata in larga parte dalla localizzazione di queste e dalla attivazione di meccanismi di riparo del DNA.

1.2 Il Saggio della Cometa (Comet Assay)

Il Comet Assay, noto scientificamente come elettroforesi su gel di singole cellule, è una tecnica microelettroforetica utilizzata per rilevare e quantificare le rotture del filamento di DNA a livello della singola cellula.

Questo metodo fu introdotto per la prima volta da Östling e Johanson, 1984 come metodica per l'individuazione del danno genetico.

Nel loro esperimento cellule di pazienti sottoposti a radioterapia vennero lisate, sottoposte a elettroforesi in un sottile strato di agarosio, e colorate con un marcatore fluorescente. Essendo le molecole di DNA cariche negativamente, l'applicazione di un campo elettrico le spinge a migrare verso l'anodo.

Le immagini ricavate, che vennero successivamente denominate comete a causa della loro morfologia, furono utilizzate per quantificare l'entità del danno genotossico.

Östling e johanson osservarono che la quantità di DNA che si era liberato dalla testa della cometa era proporzionale all' irradiazione ricevuta.

Tuttavia questo metodo presentava delle limitazioni strutturali, infatti, lavorando con pH neutro le condizioni di lisi erano isufficienti per rimuovere tuttte le proteine, di conseguenza la coda della cometa risultava un alone di DNA dovuto alla perdita del superavvolgimento.

Si ebbe poi un punto di svolta nel 1988 grazie al lavoro di Singh, che modificando il protocollo e operando in condizioni fortemente alcaline (quindi con un pH>13) strutturò un protocollo in grado di srotolare completamente il DNA, permettendo di rilevare sia rotture a doppio filamento che quelle a filamento singolo.

Oggi il Comet Assay è uno degli stumenti fondamentali in numerosi campi di ricerca, come in quello farmacologico per verificare che nuove molecole o farmaci non causino mutazioni, ma anche nell'oncologia clinica permettendo di monitorare i danni al DNA indotti dalla radioterapia o dalla chemioterapia; inoltre, trova larghissimo impiego nel biomonitoraggio ambientale, dove viene utilizzato per misurare effetti genotossici di inquinanti ambientali in diversi modelli [5].

Per esempio, è stato utilizzato per studiare gli effetti genotossici di contaminanti ambientali in bivalvi di acqua dolce e marina (Frenzilli et al. 2001; Jha et al. 2005; Rank et al. 2005; Steinert et al. 1998).

Oltre all'utilizzo in contesti di monitoraggio ambientale, trova largo impiego anche in ambito biomedico, utilizzato per stimare il rischio di insorgenze di malattie, come carcinoma a cellule renali, tumori della vescica, dell'esofago e del polmone (Djuzenova et al. 1999; Lin et al. 2007; Schabath et al. 2003; Shao et al. 2005); inoltre, applicato in contesti *in-vitro* (cioè su cellule in coltura), il test della cometa viene proprosto anche come alternativa ai test citogenetici nello sceening della genotossicità dei farmaci (Witte et al. 2007) [1].

Tra le applicazioni biomediche, il test della cometa è stato utilizzato anche per la valutazione del danneggiamento del DNA nei gameti maschili maturi (Morris et al., 2002). Il materiale genetico degli spermatozoi è vulnerabile all'azione di agenti come stress ossidativo, difetti di maturazione o esposizione ad agenti esogeni, che possono portare alla frammentazione del DNA. La presenza di un'elevata percentuale di spermatozoi con DNA gravemente danneggiato è strettamente correlato all'infertilità maschile e può ridurre le probabilità di procreazione [3].

In questo contesto, il test della cometa può essere utilizzato anche per valutare la sicurezza di protocolli di crioconservazione del liquido seminale, ma anche per testare l'efficacia di sostanze protettive antiossidanti, mirate a preservare l'integrità del genoma durante lo stoccaggio a basse temperature.

1.2.1 Principi Biologici e Preparazione del Campione

Il funzionamento del test si basa su un principio fisico ed elettrochimico semplice; la macromolecola del DNA è naturalmente carica negativamente, e in una cellula sana si presenta molto compatta e aggrovigliata su se stessa "superavvolto", tuttavia, quando un agente genotossico colpisce la cellula provoca la rottura dello scheletro chimico, il filamento si rilassa, perde tensione e si frammenta.

Applicando a queste cellule un campo elettrico, i frammenti di DNA danneggiati iniziano a migrare verso il polo positivo, e maggiore è la quantità di materiale danneggiato maggiore sarà il materiale genetico presente nella coda della cometa.

La procedura per la preparazione del campione si articola in diversi passaggi; le cellule da analizzare una volta prelevate, vengono inglobate in una sospensione di gel di agarosio e depositate su un vetrino, che verrà poi immerso in una soluzione altamente salina e ricca di detergenti, con lo scopo di rimuovere le proteine che tengono insieme il DNA.

Avviene poi il passaggio di denaturazione, dove i vetrini vengono trasferiti in una soluzione elettroforetica, che ha lo scopo di separare la doppia elica del DNA, viene applicata una corrente elettrica che permette alla cellula di effettuare la sua migrazione dal catodo all'anodo.

Una volta finita la corsa i vetrini vengono lavati e colorati con un marcatore fluorescente, che una volta legato al DNA permette la visione della cometa attraverso un microscopio a fluorescenza [9].

1.2.2 Le Classi Morfologiche di Danno

Al termine della preparazione del campione e dell'elettroforesi, le cellule osservate al microscopio hanno assunto la loro tipica forma a cometa .

Per quantificare il danno genetico è necessario identificare visivamente la quantità di materiale genetico rimasto nella testa (nucleo) e il DNA migrato nella coda.

Tradizionalmete le comete vengono classificate in cinque categorie in base al danno che presentano.

Nella Classe 1 la cellula appare come una sfera densa, compatta e luminosa, il DNA è intatto e confinato all'interno del nucleo, mentre la coda è assente o comunque impercettibile visivamente.

Nella Classe 2 inizia a rendersi visibile una corta e debole scia luminosa, la testa della cometa mantiene ancora la maggior parte del materiale genetico e la sua integrità strutturale, ma una piccola percentuale di frammenti hanno incominciato a migrare.

Nella Classe 3 la coda diventa chiaramente definita e più allungata rispetto alla Classe 2 con una testa che incomincia a perdere la sua luminosità, quindi rappresenta un nucleo che è danneggiato, e con una porzione di DNA che è migrato, però sebbene la testa della cometa sia visibilmente danneggiata mantiene comunque la sua morfologia tradizionale.

Per quanto riguarda la Classe 4 si nota che la maggior parte del materiale genetico è migrato nella coda, che risulta molto allungata, espansa e luminosa, di contro, la testa della cometa è drasticamente rimpicciolita e presenta una fluorescenza residua molto debole.

Infine, la Classe 5 spesso definita come Ghost, presenta una testa quasi totalmente assente o ridotta ad un punto flebile, mentre la quasi totalità del materiale genetico si trova nella coda, creando una nube di DNA.

Questa morfologia è il segno inequivocabile di una frammentazione genetica pressoché totale.

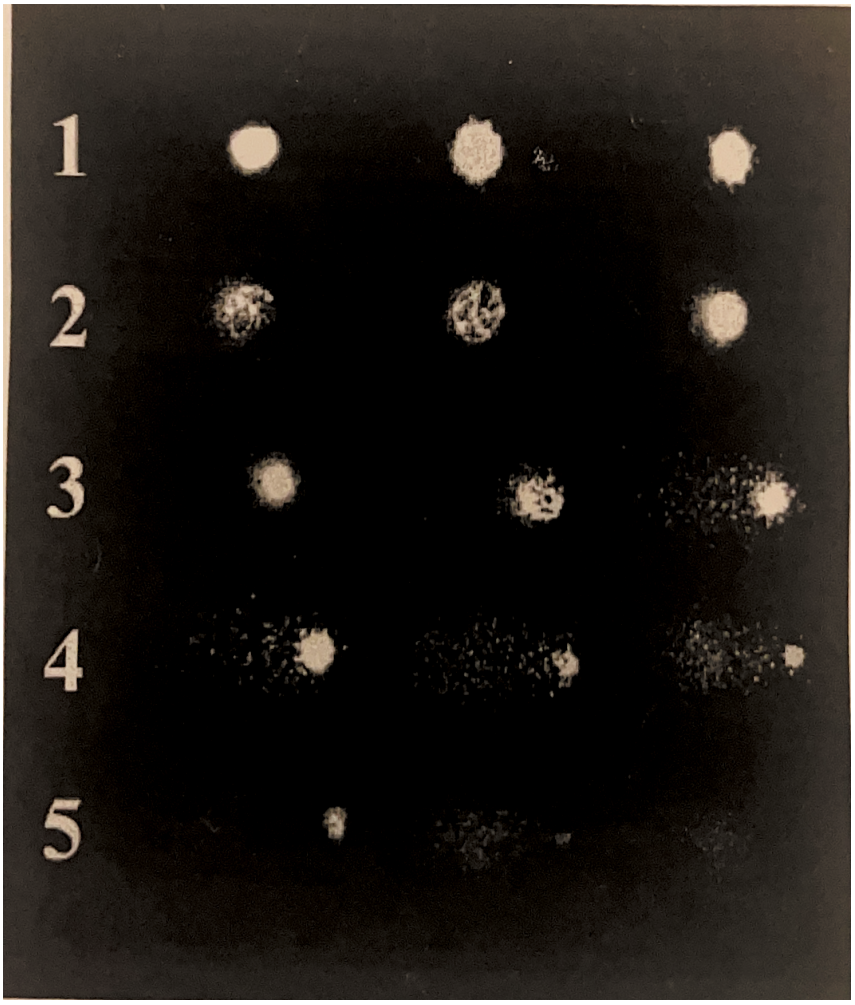

Figura 1.1: Rappresentazione visiva delle classi morfologiche di danno.

1.2.3 Analisi di immagine: tecniche quantitative e qualitative

L'analisi delle immagini ottenute tramite il test della cometa prevede due fasi principali che sono la localizzazione e l'isolamento di ogni singola cometa e la successiva fase di misurazione; originariamente, l'analisi delle comete veniva eseguita in modo puramente qualitativo, ma lo sviluppo tecnologico ha permesso di ottenere dati quantitativi attraverso l'uso di software dedicati.

L'analisi qualitativa, comunemente detta visual scoring, si basa sull'osservazione diretta del campione al microscopio, consiste quindi in una stima visiva soggettiva del livello di danno al DNA, le comete vengono classificate dall'operatore in diverse categorie in base alla lunghezza e alla forma della coda.

L'ispezione manuale delle immagni è fortemente dipendente dall'interpretazione visiva del singolo ricercatore, provocando una variabilità dei risultati tra laboratori differenti, ostacolando la standardizzazione del saggio come diagnostica universale.

Questa metodologia è molto inefficiente ed estremamente dispendiosa in termini di tempo.

Sebbene la valutazione visiva sia altamente soggettiva, risulta un metodo semplice e conveniente per chi non ha la possibilità di acquistare costosi programmi informatici.

Per superare la soggettività dell'osservazione umana, si ricorre all'analisi di immagine computerizzata, questo metodo è preferibile poiché fornisce informazioni quantificate e oggettive sulle comete attraverso un calcolo basato sui pixel, qui vengono presi in considerazioni parametri come il Tail Moment, calcolato come il prodotto tra la lunghezza della coda e la percentuale di fluorescenza, tra gli strumenti disponibili per questa analisi vi è per esempio il Comet Score, un software che permette l'elaborazione delle immagini.

Bisogna però tenere conto che, le acquisizioni a microscopio presentano spesso difetti di gel o coloranti non legati, rendendo quindi sensibile il software ai rumori di fondo, può accadere che due comete si sovrappongano e il modello computazionale non riconosca le due comete, portando quindi a fonderle insieme e considerarle come una sola.

Queste limitazioni mettono in evidenza come le tecniche attuali sia qualitative che quantitative tradizionali, non siano in grado di garantire l'oggettività e l'affidabilità richiesta, è in questo contesto che l'applicazione delle reti neutrali convoluzionali e del Deep Learning si propongono come soluzione per superare i limiti sia dell'analisi visiva che del Machine Learning [4, 6, 7, 8, 10, 11, 12].

1.3 Intelligenza Artificiale in Ambito Medico e Biologico

L'introduzione dell'intelligenza Artificiale nel panorama medico e biologico ha segnato un punto di svolta metodologico senza precedenti [13] .

Storicamente, l'analisi dei campioni biologici e la diagnostica delle immagini è sempre stata effettuata da un operatore umano, che sebbene l'occhio umano sia abile nell'individuare pattern complessi, la valutazione manuale porta con se molte limitazioni.

Oggi con l'introduzione di algoritmi intelligenti si ha la possibilità di automatizzare questi processi garantendo una standardizzazione oggettiva, con la possibilità di analizzari moli di dati in tempi incompatibili alle capacità umane, bisogna però precisare che l'IA non va a sostiture il medico o il biologo ma svolge una funzione di supporto ad esso, permettemdo di rendere più leggero il lavoro.

1.3.1 Dal Machine Learning Classico al Deep Learning

L'elaborazione dei dati grezzi, come le matrici di pixel ha rappresentato per decenni una sfida complessa per i sistemi computazionali tradizionali, prima dell'avvento delle moderne architetture, le tecniche tradizionali di Machine Learning erano fortemente limitate nella loro capacità di processare i dati.

Costruire un'architettura di Machine Learning classico richiedeva un'intensa fase di progettazione umana, ed affinché un classificatore standard potesse operare con successo era necessario l'intervento di un ingegnere o di un esperto.

Questo modulo software aveva il compito di trasformare i dati grezzi in un'adeguata rappresentazione vettoriale interna, estraendo metriche calcolate a priori da cui il sottosistema di apprendimento potesse rilevare i pattern desiderati.

Questo approccio presentava un evidente limite, infatti, l'efficacia del classificatore era totalmente delegata all'abilità umana, rendendo il sistema rigido e vulnerabile a variazioni impreviste nell'illuminazione, nel rumore o nell'orientamento del campione.

Il Deep Learning va a risolvere queste problematiche introducendo il paradigma del Representation Learning, che a differenza dell'approccio tradizionale una architettura Deep Learning è progettata per essere alimentata direttamente con dati grezzi, imparando in totale autonomia le rappresentazioni e le caratteristiche necessarie per la classificazione.

Dal punto di vista strutturale il Deep Learning è costituito da molteplici livelli di elaborazione, implementati tramite moduli non lineari disposti in cascata.

Ogni modulo trasforma la rappresentazione del livello precedente in una rappresentazione a un livello di astrazione superiore.

Nel contesto delle immagini il processo avviene in modo gerarchico, il primo strato impara a rilevare primitive visive a bassa frequenza, come bordi e linee, il secondo strato combina i bordi per formare motivi strutturali, in questo modo tutti gli strati successivi assemblano tali motivi in parti di oggetti, fino ad arrivare agli strati più

profondi che sono in grado di prelevare la forma più complessa di una cellula sana o danneggiata.

La peculiarità di questo sistema è che gli strati di estrazione non sono progettati da un essere umano, ma essi vengono appresi in modo completamente automatico.

La rete calcola una funzione obbiettivo che misura l'errore tra la classificazione predetta e la classe reale di appartenenza, successivamente, attraverso l'algoritmo di retropropagazione dell'errore, quindi la rete valuta l'errore commesso e individua in che misura ogni suo parametro interno abbia contribuito a tale sbaglio, infine, attraverso cicli successivi di addestramento il sistema corregge questo errore e aggiorna i paramentri.

In sisntesi, nel Deep Learning non è più l'ingegnere a dover definire le caratteristiche dell'immagine, il lavoro viene spostato sull'addestramento della rete neutrale che impara da sola i dettagli necessari per riconoscere le comete ignorando i difetti visivi [2].

1.3.2 Reti Neurali Convoluzionali (CNN)

Sebbene le classiche reti neurali (MLP) siano in grado di approssimare qualsiasi funzione continua, la loro applicazione diretta sulle immagini biomediche ad alta risoluzione risulta inefficace, infatti, un'immagine digitale è una matrice bidimensionale o tridimensionale di pixel.

In un MLP (Multi Layer Perceptron) l'immagine deve diventare un vettore monodimensionale, andando quindi a eliminare la topologia spaziale originale, inoltre, essendo tutti i nodi collegati tra di loro, questo crea una combinazione dei parametri addestrabili rendendo così la rete sensibile all'overfitting [2].

Le reti neurali convoluzionali (CNN) vanno a risolvere queste criticità, sono inspirate ai campi recettivi della corteccia cerebrale animale, sono progettate per l'elaborazione di dati con una struttura a griglia, riducendo i parametri da analizzare, in sintesi guardano l'immagine come farebbe l'occhio umano.

Le CNN funzionano seguendo alcuni passaggi fondamentali, il primo tra essi sono gli strati convoluzionali, responsabili dell'estrazione delle feature map, dove invece di connettere ogni neurone all'intero input, la convoluzione opera attraverso dei filtri che scorrono in modo sequenziale sull'immagine, questo permette al filtro di imparare a riconoscere i bordi dell'immagine, in questo modo la cometa verrà identificata indipendentemente dalle sue coordinate all'interno dell'immagine.

Poichè l'operatore convoluzionale è una trasformazione lineare è matematicamente necessario introdurre una non linearità, qui entra in gioco un altro passaggio definito come ReLU (Rectified Linear Unit), [2].

$$f(x) = max(0, x)$$

Il compito di questa formula è quello di azzerare tutte le attivazioni negative, lasciando però inalterate quelle positive.

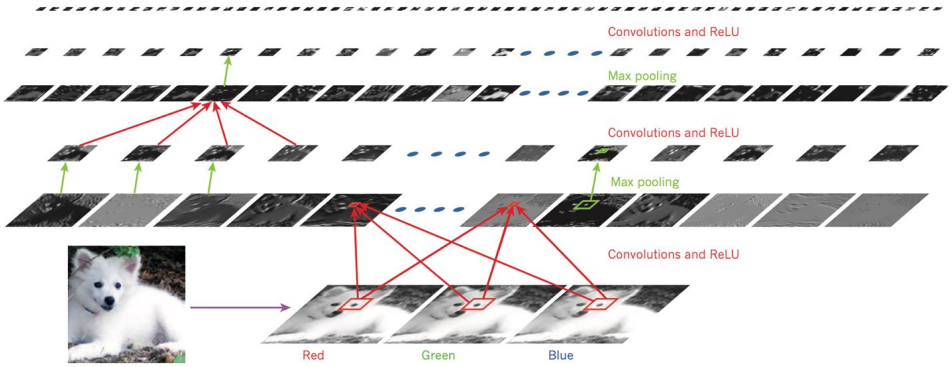

Figura 1.2: esempio di funzionamento della Rete Neutrale Convoluzionale, [13]

Se gli strati convoluzionali servono a individuare i piccoli dettagli all'interno dell'immagine, lo strato di Pooling ha il compito di riassumere, raggruppando insieme tutte le informazioni simili.

Poiché in una foto le caratteristiche di un oggetto possono trovarsi in posizioni diverse, il programma riesce a riconoscerlo comunque facendo un raggruppamento dei dettagli, in pratica il sistema guarda un'area dell'immagine e conserva solo il segnale più forte, queste aree scorrono poi su tutta l'immagine passando da una zona all'altra.

In questo modo si rimpicciolisce la quantità di dati da elaborare e si crea un sistema che non si fa ingannare dal rumore di fondo; il procedimento viene poi ripetuto altre volte [13].

l'ultimo strato possiede un numero di nodi pari al numero di classi che ci servono, nel nostro caso sono cinque, l'attivazione di questo strato è determinata dalla funzione Softmax la quale normalizza i valori di output in una distribuzione di probabilità, la classe con la probabilità più alta costituisce la classificazione finale assegnata dal modello alla cellula esaminata.

Capitolo 2

Scopo della Tesi

Alla luce di quanto introdotto nel Capitolo 1, in questo lavoro di tesi è stato progettato, sviluppato e validato un sistema di classificazione e analisi automatizzata delle immagini del Comet Assay basato su tecniche di Intelligenza Artificiale, con un focus specifico sulle architetture di Deep Learning.

L'obiettivo è quello di esplorare la possibilità di superare le limitazioni intrinseche di questa procedura di classificazione, riducendo la soggettività valutativa dell'approccio qualitativo.

Per raggiungere l'obiettivo, il progetto è stato articolato in diversi passaggi specifici.

Inizialmente, è stato costruito un dataset di immagini previamente acquisite e classificate da più operatori.

Il dataset è stato utilizzato per addestrare un modello appositamente sviluppato per interpretare la cellula e distinguere correttemante testa e coda della cometa, attraverso l'implementazione di una Rete Neurale Convoluzionale, dove in seguito tutte le fasi di Training e di Validation sono state poi controllate attraverso metriche statistiche standard.

Infine, è stato condotto il test della cometa su cellule estratte ex-novo, e immagini dei nuclei sono state acquisite al microscopio a fluorescenza per poi essere sottoposte a valutazione nel modello sviluppato, con lo scopo di comparare la classificazione proposta dal modello con quella dell'operatore.

Questo lavoro si interpone come ponte tra l'ingegneria dei sistemi di elaborazione dell'informazione e la biologia, fornendo un contributo che possa migliorare il Comet Assay per le applicazioni diagnostiche, cliniche e di screening tossicologico su larga scala.

Capitolo 3

Materiali e Metodi

3.1 Acquisizione Sperimentale e Costruzione del Dataset di riferimento

In questo capitolo viene presentato il dataset di riferimento e come questo è stato gestito.

Nello sviluppo di architetture basate sul Deep Learning, la qualità, la consistenza e la rappresentatività del dataset costituiscono il fattore primario per determinare le performance di generalizzazione del modello.

Diversamente dagli algoritmi classici, in cui le regole di classificazione vengono programmate esplicitamente in modo analitico, le Reti Neurali Convoluzionali (CNN) deducono le feature discriminanti direttamente dalla distribuzione spaziale e dalle intensità dei pixel forniti in ingresso, di conseguenza, la costruzione del database di riferimento non rappresenta una mera fase di raccolta, ma il primo e più critico step dell'intera pipeline ingegneristica e sperimentale.

Il materiale di partenza per la costruzione del database è stato fornito in precedenza, e non consisteva in immagini di singole cellule ma in mosaici di comete (n=25) di cellule di mitili della specie *Mytilus galloprovincialis* mantenuti in condizioni di controllo o esposti a cadmio, metallo con attività genotossica. Ciascun mosaico è stato suddiviso isolando foto delle singole cellule (Fig. 3.1 e 3.2).

I mosaici vengono abitualmente fatti per l'analisi di immagine quantitativa via software, in quanto in un singolo vetrino sono presenti migliaia di cellule e relative comete. Il ritaglio e la selezione delle cellule idonee all'addestramento del modello è stato necessario per fornire a questo immagini adeguate a singola cellula, eliminando elementi che potrebbero confondere il processo di apprendimento (Fig. 3.3).

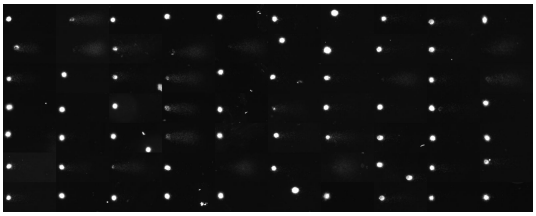

Figura 3.1: Mosaico di comete

Figura 3.2: Cometa successivamente separata

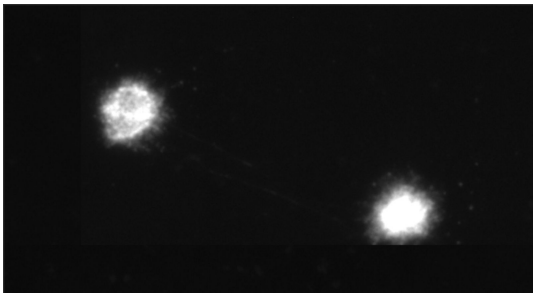

Figura 3.3: Immagine di cometa non adatta alla nostra classificazione

Una volta isolate tutti i nuclei e il materiale migrato, queste sono state classificate nelle cinque classi in modo qualitativo senza l'ausilio di programmi per la classificazione, seguendo la metodica di classificazione presentata nel capitolo 1.2.2. Questo ha prodotto un database inizale di 1319 cellule, come riportato nella figura 3.4.

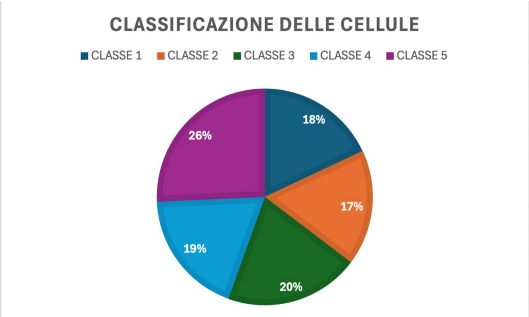

Figura 3.4: Suddivisione delle cellule nelle rispettive classi da parte dell'operatore

Questo passaggio ha costituito la base di conoscenza sufficientemente vasta alla Rete Neurale per l'apprendimento delle feature tipiche di ciascuna classe, e rendere il database di riferimento affidabile in vista della classificazione automatica del Deep Learning.

3.2 Implementazione Software e Architettura del Modello Computazionale

Conclusa la fase di acquisizione ottica e costruzione del dataset, il lavoro è stato spostato alla progettazione e implementazione software dell'algoritmo di classificazione; l'obiettivo di questa fase è lo sviluppo di un sistema computazionale che restituisca in output una predizione accurata della classe di danno genotossico, cercando di minimizzare l'intervento soggettivo dell'operatore.

3.2.1 Ambiente di Sviluppo e Preparazione del Dataset

Per la stesura, il collaudo e l'esecuzione del codice è stato scelto Google Colaboratory; l'intero algoritmo è stato sviluppato in linguaggio Python grazie al suo vasto ecosistema di librerie scientifiche di calcolo tensoriale.

Il primo passo ha richiesto l'importazione delle librerie fondamentali e l'inizializzazione dell'ambiente di calcolo.

Listing 3.1: Importazione delle librerie necessarie

```
1
2  import tensorflow as tf
3  from tensorflow import keras
4  from tensorflow.keras import layers
5  import matplotlib.pyplot as plt
6  import os
7
8  from google.colab import drive
9  drive.mount('/content/drive')
10
11 dataset_path = '/content/drive/MyDrive/DATASET_CM'
12
13 print("Librerie caricate e Drive montato con successo!")
```

L'architettura software poggia su un ecosistema di librerie specifiche, ciascuna con un compito ben preciso.

TensorFlow e Keras rappresentano il core computazionale dell'intero progetto: questi moduli forniscono gli strumenti matematici necessari per assemblare strato dopo strato la rete neurale e per gestirne l'addestramento.

Troviamo poi le librerie NumPy, impiegata per manipolare le immagini una volta convertite in matrici numeriche e tensori, mentre per la generazione dei grafici è stato utilizzato Matplotlib.

Questo primo blocco di codice, rappresenta la base della implementazione software.

3.2.2 Strategia di Addestramento del Modello

Per istruire un'architettura Deep Learning a riconoscere e classificare le morfologie del danno genotossico, non è sufficiente fornire l'intero database di riferimento; infatti, se il modello elaborasse tutti i dati a sua disposizione non avrebbe nessun parametro oggettivo per valutare la capacità predittiva su nuove immagini mai analizzate; per questo motivo il dataset è stato suddiviso in due sottoinsiemi, *Training Set* e *Validation Set*.

Il *Training Set* costituisce la porzione predominante dei dati, costituendo l'80% delle immagini a disposizione. Esso rappresenta il materiale di studio della Rete Neurale: durante la fase di addestramento, l'algoritmo analizza queste immagini e tenta via via una predizione della classe di danno, confrontando poi il risultato con l'etichettatura reale fornita dall'operatore e calcola un margine di errore, definito Loss. Utilizzando la retropropagazione, il modello si aggiorna cercando di minimizzare questi errori, imparando le feature delle comete.

Il *Validation Set* è composto da un 20% del nostro database e funge da test: in questo passaggio, i dati vengono nascosti alla rete; alla fine di ogni ciclo di addestramento, definito epoca, il modello viene interrogato sulle immagini appartenenti al Validation Set: non essendo mai state processate, elaborate o gestite dall'algoritmo, la valutazione fornita da questo è indice della sua prestazione su questo sottoinsieme, e rappresenta quindi la capacità di generalizzazione dell'algoritmo.

Il Validation Set è fondamentale per la riuscita del Deep Learning: infatti, permette di evitare l'overfitting, ossia quando una rete neurale invece che dedurre le regole generali che distinguono le varie classi, inizia ad imparare a memoria le peculiarità delle immagini di training.

Nel blocco di script successivo, viene definita l'architettura delle Rete Neurale, vengono definiti i livelli convoluzionali, e viene svolta l'operazione di ReLU, seguita poi dalla softmax.

Una volta suddiviso il database e definita l'architettura della CNN, viene avviato il processo di apprendimento: in questa fase l'algoritmo ricerca il minimo della funzione di costo procedendo per approssimazioni successive nel tempo. L'unità di misura temporale e computazionale viene definita epoca: un'epoca rappresenta un ciclo di elaborazione completo, durante il quale l'intero training set attraversa la rete neurale.

L'addestramento è stato inizalizzato con un limite massimo di 30 epoche; tuttavia, l'algoritmo non è stato obbligato a compiere tutti i cicli, poiché questo potrebbe esporlo ad una elevata probabilità di overfitting. Per questo è stato aggiunto l'early stopping, che agisce come un supervisore che al termine di ogni epoca valuta l'andamento della funzione di costo sul set di validazione (val_loss).

La regola applicata in questa fase è stata quella di interrompere forzatamente l'addestramento qualora dopo cinque epoche successive il valore della val_loss non registri alcun miglioramento, scartando i dati ricavati dall'ultima epoca e registrando i dati appartenenti alla sua prestazione migliore.

Listing 3.2: Suddivisione tra Training e Validation Set

```
import math
import tensorflow as tf

TRAIN_PCT = 80  # Percentuale per l'addestramento
VAL_PCT = 20    # Percentuale per la validazione

batch_size = 32
img_height = 128
img_width = 128

val_split_keras = VAL_PCT / 100.0

print(f"Caricamento Training Set ({TRAIN_PCT}%)...")
train_ds = tf.keras.utils.image_dataset_from_directory(
    dataset_path,
    validation_split=val_split_keras,
    subset="training",
    seed=123,
    color_mode="rgb",
    image_size=(img_height, img_width),
    batch_size=batch_size
)

print(f"Caricamento Validation Set ({VAL_PCT}%)...")
val_ds = tf.keras.utils.image_dataset_from_directory(
    dataset_path,
    validation_split=val_split_keras,
    subset="validation",
    seed=123,
    color_mode="rgb",
    image_size=(img_height, img_width),
    batch_size=batch_size
)

print("\n--- Riepilogo dei Set (in numero di batch) ---")
print(f"Training batches  : {tf.data.experimental.cardinality(
    train_ds).numpy()}")
print(f"Validation batches: {tf.data.experimental.cardinality(
    val_ds).numpy()}\n")

AUTOTUNE = tf.data.AUTOTUNE
train_ds = train_ds.cache().shuffle(1000).prefetch(buffer_size=
    AUTOTUNE)
val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)
```

Listing 3.3: Costruzione dei livelli della Rete Neurale Convoluzionale

```
num_classes = 5

data_augmentation = keras.Sequential(
  [
    layers.RandomFlip("horizontal_and_vertical", input_shape=(
        img_height, img_width, 3)),
    layers.RandomRotation(0.2),
    layers.RandomZoom(0.1),
  ]
)

model = keras.Sequential([
  data_augmentation,
  layers.Rescaling(1./255),

  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),

  layers.Flatten(),
  layers.Dense(128, activation='relu'),
  layers.Dropout(0.5),
  layers.Dense(num_classes, activation='softmax')
])

model.compile(optimizer='adam',
              loss=tf.keras.losses.
                  SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

model.summary()
```

Listing 3.4: Addestramento del modello attraverso le epoche

```
epochs = 30 # Inizia con 30 epoche

early_stopping = tf.keras.callbacks.EarlyStopping(
    monitor='val_loss',
    patience=5,
    restore_best_weights=True
)

print("Inizio addestramento...")
history = model.fit(
  train_ds,
  validation_data=val_ds,
  epochs=epochs,
  callbacks=[early_stopping]
)
print("Addestramento completato!")
```

3.2.3 Progettazione degli Strumenti di Testing e Interazione

Al termine della fase di addestramento, l'attenzione è stata spostata sulla quantificazione delle prestazioni del modello e sulla sua integrazione in un software. Quindi da un lato dobbiamo certificare l'affidabilità della rete neurale su dati mai osservati prima e dall'altro dobbiamo creare un'interfaccia grafica utilizzabile dal personale di laboratorio che non richieda alte competenze di programmazione.

Quindi è stato introdotto uno script dedicato esclusivamente al testing automatizzato del Test Set, ossia un insieme di fotografie di nuclei e comete acquisiti al microscopio ex-novo, mai processate dalla rete neurale in nessuna delle epoche.

Per ognuna di queste immagini il modello viene interrogato, registrando la classe predetta, fornendo in percentuale la sicurezza dell'algoritmo nella sua classificazione. Le prestazioni del modello sono poi state confrontate con la classificazione fornita da operatori diversi, al fine di verificare che il modello abbia effettivamente imparato le regole di classificazione.

Dal momento che la classificazione offerta dal modello è soggetta ad incertezza, lo script prodotto fornisce oltre alla classificazione una percentuale di sicurezza. Grazie a questa, l'operatore ha la possibilità di verificare la classe di appartenenza.

A fine di rendere l'architettura neurale un vero e proprio strumento diagnostico integrabile nella routine di un laboratorio, l'ultima fase del progetto ha riguardato lo sviluppo di un'interfaccia grafica direttamente all'interno del codice di Google Colab, questo permette all'operatore di caricare una serie di micrografie a fluorescenza direttamente dal proprio archivio cloud. Queste immagini vengono aperte dalla libreria PIL e il codice applica le medesime trasformazioni effettuate durante l'addestramento.

L'ambiente interattivo stampa a schermo l'immagine caricata affiancata dai risultati diagnostici, in particolare viene estratta la classe morfologica di appartenenza e il grado di confidenza espresso sempre in valore percentuale. Come già detto, fornire questo valore statistico è fondamentale per l'interazione uomo macchina poiché permette al biologo o clinico di validare con sicurezza le predizioni ad alta confidenza, o di procedere con una ispezione manuale nei casi in cui l'algoritmo mostra incertezza.

Lo sviluppo dell'interfaccia grafica permette anche all'operatore di andare a ricercare una specifica immagine di interesse nel vasto database di comete che sono state caricate, infatti, ha la possibilità di selezionarne solo una e l'algorimo gli restituirà la classe di appartenenza e la sicurezza nella classificazione.

Listing 3.5: Caricamento delle immagini mai viste dall'algoritmo

```
from google.colab import drive
drive.mount('/content/drive', force_remount=True)

import os
percorso = '/content/drive/MyDrive/CM_COMET'
print("File visti da Colab in questo momento:")
print(os.listdir(percorso))

import numpy as np
import os

class_names = ['CLASSE1', 'CLASSE2', 'CLASSE3', 'CLASSE4', '
    CLASSE5']

dataset_path = '/content/drive/MyDrive/CM_COMET'
print("Inizio classificazione automatica...\n")
print("-" * 50)

for nome_file in sorted(os.listdir(dataset_path)):
    percorso_immagine = os.path.join(dataset_path, nome_file)

    if os.path.isfile(percorso_immagine):


        img = tf.keras.utils.load_img(
            percorso_immagine, target_size=(128, 128),
                color_mode="rgb"
        )


        img_array = tf.keras.utils.img_to_array(img)
        img_array = tf.expand_dims(img_array, 0)


        predictions = model.predict(img_array, verbose=0)


        indice_classe_vincente = np.argmax(predictions[0])
        classe_predetta = class_names[indice_classe_vincente]
        confidenza = 100 * np.max(predictions[0])


        print(f"File: {nome_file}")
        print(f"Verdetto: {classe_predetta} (Sicurezza dell'
            algoritmo: {confidenza:.2f}%)")
        print("-" * 50)
```

Listing 3.6: Sviluppo dell'interfaccia grafica (GUI)

```
import os
import numpy as np
import tensorflow as tf
from google.colab import drive

print("Aggiornamento di Google Drive in corso...")
drive.mount('/content/drive', force_remount=True)

cartella_immagini = '/content/drive/MyDrive/Single'

print("\nEsplorazione della cartella in corso...")
tutti_i_file = os.listdir(cartella_immagini)
print(f"Colab ha trovato {len(tutti_i_file)} file totali nella
    cartella (compresi file nascosti).")

lista_file = []
for file in sorted(tutti_i_file):
    if file.lower().endswith(('.png', '.jpg', '.jpeg')):
        lista_file.append(file)

print("\nImmagini PNG/JPG pronte per l'analisi:")
for indice, nome_file in enumerate(lista_file):
    print(f"[{indice}] {nome_file}")

print("-" * 50)

scelta = input("Digita il numero dell'immagine che vuoi
    analizzare e premi Invio: ")

try:
    indice_scelto = int(scelta)
    nome_immagine_scelta = lista_file[indice_scelto]
    percorso_completo = os.path.join(cartella_immagini,
        nome_immagine_scelta)

    print("-" * 50)
    print(f"Hai selezionato il file: {nome_immagine_scelta}")
    print("Avvio classificazione...")

    img = tf.keras.utils.load_img(percorso_completo,
        target_size=(128, 128), color_mode="rgb")
    img_array = tf.keras.utils.img_to_array(img)
    img_array = tf.expand_dims(img_array, 0)

    predictions = model.predict(img_array, verbose=0)

```

```
43      indice_classe_vincente = np.argmax(predictions[0])
44      classe_predetta = class_names[indice_classe_vincente]
45      confidenza = 100 * np.max(predictions[0])
46
47      print(f"Verdetto Finale: {classe_predetta} (Sicurezza dell'
            algoritmo: {confidenza:.2f}%)")
48
49  except ValueError:
50      print("Errore: Devi digitare un numero, non lettere o
            simboli.")
51  except IndexError:
52      print("Errore: Il numero digitato non corrisponde a nessuna
            immagine nella lista.")
53  except Exception as e:
54      print(f"Si e verificato un errore: {e}")
```

3.2.4 Validazione del modello con campioni *ex-novo*

Al fine di ottenere delle nuove immagini di nuclei sottoposti al test della Cometa da utilizzare nella fase di test e popolare la sezione del dataset "Test Set", cellule di mitili della specie *Mytilus galloprovincialis* sono state processate *ex-novo* in laboratorio con microelettroforesi e successiva acquisizione di immagini.

Il primo step ha previsto il prelievo dell'emolinfa dal muscolo adduttore degli organismi, ossia il fluido cicrolante nei tessuti dei mitili contenente le cellule di interesse (emociti). L'operazione consiste nel prelievo di un volume di emolinfa compreso tra i 700 e gli 800 µL, e richiede estrema cautela per evitare il danneggiamento delle fibre muscolari e la contaminazione del fluido con cellule di popolazioni differenti (Fig. 3.5).

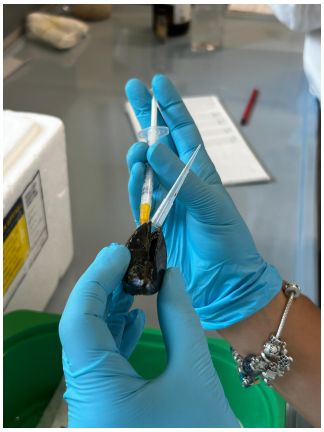

Figura 3.5: Prelievo dell'emolinfa dai mitili

I campioni sono stati trasferiti all'interno di microtubi di tipo Eppendorf per essere poi sottoposti alla fase di centrifuga con un'accelerazione di 6000 rpm per 4 minuti; questa permette la sedimentazione del materiale cellulare, formando un pellet sul fondo della provetta; il surnatante viene rimosso, conservandone una piccola quantità essenziale per la risospensione del campione.

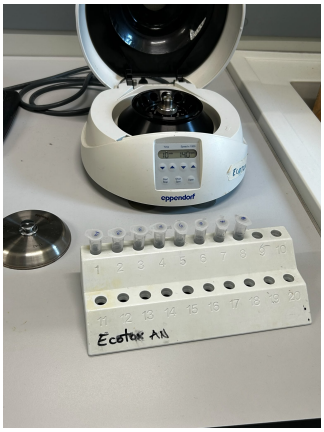

Figura 3.6: Centrifuga per Eppendorf

I vetrini portaoggetto sono stati preparati con un primo strato base di agarosio.

Successivamente, 20 microlitri di sospensione cellulare sono stati miscelati con 180 microlitri di gel. Di questa miscela, 100 microlitri sono stati posizionati sopra lo strato base di agarosio del vetrino (posto precedentemente) creando un reticolo tridimensionale in cui le cellule restano intrappolate; i vetrini sono stati preparati in doppia replica analitica, e sono stati lasciati incubare in frigorifero al buio per 2 ore.

Al fine di consentire la corretta migrazione del DNA, è necessaria la rottura delle membrane cellulari: per questo, i vetrini sono stati immersi in una soluzione di lisi composta da dimetilsulfossido e Triton X (0.01% e 0.001%, rispettivamente), seguita poi da un'incubazione di circa 10 minuti con Proteinasi K, un passaggio cruciale quando si lavora con invertebrati in modo da prevenire la frammentazione eccessiva del DNA.

Figura 3.7: Materiali utilizzati per la preparazione del gel di agarosio e della soluzione di lisi

I campioni sono stati poi posti all'interno della camera elettroforetica, per una prima fase di denaturazione della doppia elica di DNA, seguita dall'applicazione del campo elettrico che permette la migrazione delle cellule; terminata questa fase, i vetrini sono stati immersi in un buffer di rinaturazione, ed infine disidratati tramite un bagno in etanolo. Al termine, questi verranno conservati a -20° fino al momento dell'analisi.

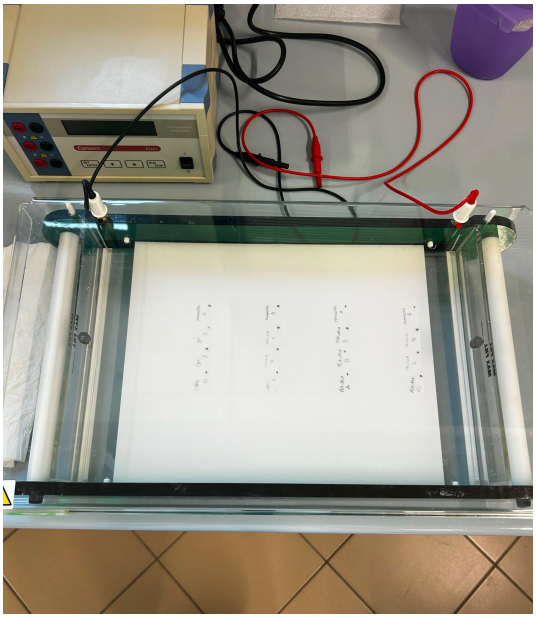

Figura 3.8: Camera elettroforetica per la migrazione dei campioni

Per l'analisi, i campioni vengono reidratati con delle gocce di marcatore fluorescente capace di legarsi al DNA, e i vetrini vengono esaminati attraverso l'utilizzo del microscopio a fluorescenza, eseguendo una mappatura visiva del vetrino, ritagliando le singole comete. Nel caso di questo lavoro di tesi, queste sono state utilizzate nella fase di testing del modello.

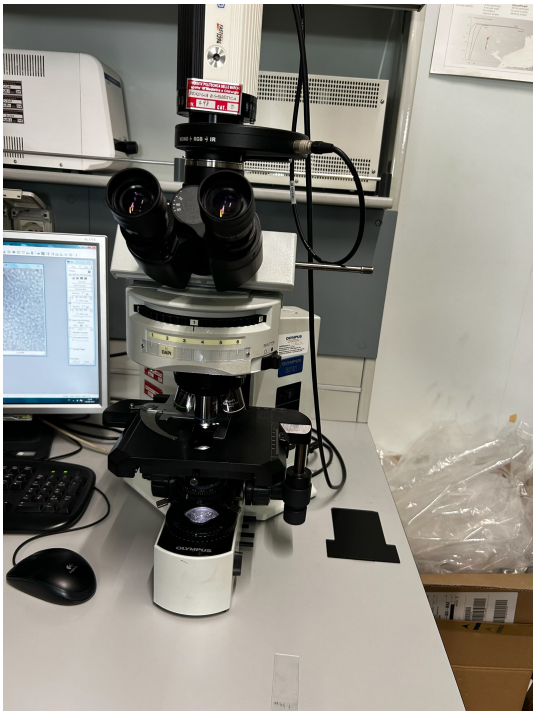

Figura 3.9: Microscopio a fluorescenza utlizzato per la visione e prelievo delle immagini delle comete

Capitolo 4

Risultati Sperimentali e Conclusioni

Vengono presentati di seguito i risultati ottenuti durante lo sviluppo, con l'obiettivo di quantificare le reali capacità di apprendimento e la validità diagnostica delle rete neurale sviluppata per la classificazione automatizzata del comet assay.

4.1 Analisi delle Prestazioni di Classificazione

La prima fase di valutazione delle prestazioni prevede lo studio delle curve di appredimento e delle matrici di confusione ottenute durante lo sviluppo del codice.

4.1.1 Andamento delle Curve di Apprendimento

La valutazione delle reali capacità predittive dell'architettura neurale si basa sull'analisi visiva e analitica delle curve di apprendimento, grafici generati estraendo i dati al termine del processo di addestramento che tracciano l'evoluzione dell'Accuracy e della funzione Loss lungo lo scorrere delle epoche. Questi sono presentati sia per il Training Set che per il Validation Set.

Durante la fase di sviluppo, il rapporto percentuale per la suddivisione del dataset non è stato imposto a priori, ma è stato deciso da una serie di test: vista la dimensione finita del set, una percentuale di immagini destinate al Training Set eccessiva sarebbe stata a discapito delle dimensioni del Validation Set, rendendo così il collaudo dell'algoritmo poco informativo; al contario, una dimensione del Validation Set a sfavore del Training Set avrebbe impedito ai livelli convoluzionali di imparare in modo adeguato le *feature* delle cellule.

Per individuare il bilanciamento ideale, il modello è stato compilato e addestrato ripetutamente variando i parametri TRAIN_PCT e VAL_PCT, monitorando per ciascuna configuarazione il comportamento delle curve di apprendimento.

Abbiamo effettuato come prima prova un partizionamento del database del 70% per il Training e del 30% per la Validation. Il grafico riportato in figura 4.1 mostra l'andamento dell'algoritmo con questa suddivisione: il grafico a sinistra traccia la percentuale di predizioni corrette, con una rapida salita iniziale nei primi cicli di addestramento che indica che la rete sta estraendo le regole corrette. Da circa la quindicesima epoca, l'accuratezza di validazione (linea arancione) e di addestramento (linea blu) smettono di crescere in modo simultaneo, a causa della scarsità di dati.

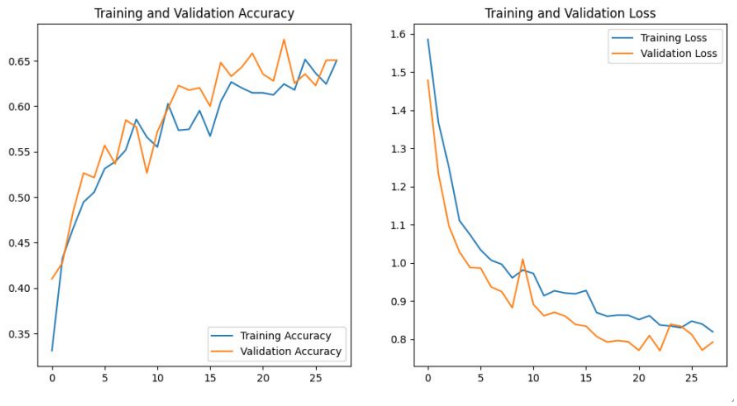

Figura 4.1: Curve di apprendimento nel caso 70% per il training e 30% per la validation

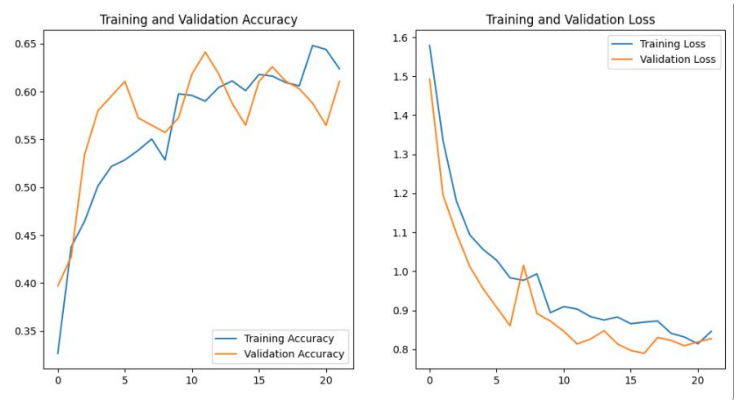

Figura 4.2: Curve di apprendimento nel caso 90% per il testing e 10% per la validation

Il grafico di destra, relativo alla misurazione dell'errore matematico, mostra che la curva della Training Loss (linea blu), incomincia a scendere bloccandosi poi non riuscendo a superare lo 0.8, mentre l'andamento della Validation Loss (linea arancione) si presenta molto frastagliata, mostrando che la rete fa fatica a stabilizzarsi.

In seguito è stata presa in considerazione una configurazione più accentuata rispetto a quella precedente, con una suddivisione del 90% per il Training Set e del 10% per il Validation Set, come si vede dalla figura 4.2 i grafici presentano delle criticità molto evidenti.

Osservando il grafico a sinistra (Training and Validation Accuracy), la curva di addestramento (linea blu) mostra la capacità della rete neurale di estrarre un numero maggiore di feature e l'accuratezza sale in modo più convinto; tuttavia emerge un difetto andando a guardare la curva della Validation (linea arancione), caratterizzata da importanti oscillazioni, registrando crolli improvvisi seguiti da picchi, che rappresentano la conseguenza di un Validation Set scarso di dati.

Il grafico a destra (Loss), conferma l'instabilità del sistema, mentre la curva di training (linea blu) scende in maniera fluida, la curva di validation (linea arancione) è molto frastagliata.

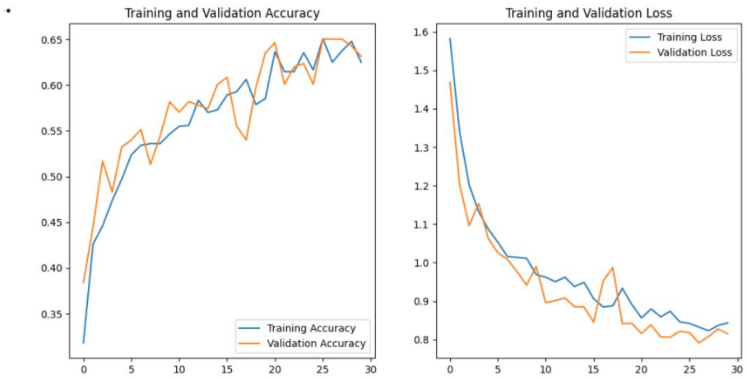

Figura 4.3: Curve di apprendimento nel caso 80% per il training e 20% per la validation

In seguito a queste considerazioni, è stata esplorata una partizione del dataset in 80% per il Training Set e 20% per il Validation, ipotizzandola come la suddivision ottimale per il corretto funzionamento dell'algoritmo. I risultati di Training e Validation di questa partizione sono riportati nella figura 4.3.

L'andamento dell'Accuracy (grafico a sinistra), dimostra un apprendimento coerente: la curva di addestramento (linea blu) cresce progressivamente, stabilizzandosi attorno al 65%, mentre la curva di validation (linea arancione) segue quella di training per l'intero processo, dimostrando che il modello è riuscito ad imparare in modo adeguato le regole di classificazione delle cellule.

Nel grafico di destra (Loss), entrambe le curve presentando un andamento asintotico passando da un errore iniziale superiore a 1.5 fino ad arrivare ad uno 0.85; la Validation Loss (linea arancione) non effettua salti troppo elevati, seguendo in modo abbastanza coerente la curva del Training Loss.

In ambito di Deep Learning, è fondamentale sottolineare che le prestazioni e le capacità di generalizzazione di un'architettura neurale sono dipendenti dalla natura e dalla qualità del database fornito durante la fase di addestramento: le reti neurali non possiedono una logica pre-programmata su un problema biologico ma deducono i criteri di classifciazione esclusivamente dagli esempi forniti, evidenziando l'importanza dei criteri di acquisizione e della qualità dei dataset nel generare reti con comportamenti diversi.

4.2 Validazione su campioni *ex-novo*

4.2.1 Matrici di Confusione e Metriche di Errore

Per validare le scelte progettuali discusse nel paragrafo precedente, il collaudo del modello è stato trasferito su un Test Set indipendente, composto da cellule esterne al dataset di Training e Validation, previamente classificate (Ground Truth). Tale set è stato valutato e l'esito, rappresentato mediante una Matrice di Confusione, che permette di ispezionare il comportamento della rete su ogni singola classe di danno genotossico.

Lo sviluppo della matrice di confusione è andato di pari passo con quello delle curve di addestramento, e vengono quindi riportate Matrici di Confusione per ciascuna delle tre partizioni testate (70-30; 90-10; 80-20).

L'indagine è partita dal collaudo della rete neurale addestrata con il partizionamento 70% per il training e 30% per la validation, configurazione che già durante l'analisi delle curve di addestramento aveva presentato delle problematiche di underfitting.

Nella matrice di confusione presentata in figura 4.4 viene confermata questa probemlematica.

Idealmente nella matrice la totalità delle predizioni dovrebbe concentrarsi sulla diagonale principale, che rappresenta i nuclei correttamente classificati. Dalla figura si può notare come il modello riesca ad identificare con una elevata precisione gli estremi morfologici (classe 1 e la classe 5), mentre una maggiore imprecisione emerge per le classi intermedie.

Questo dimostra che la rete neurale non avendo ottenuto abbastanza dati in fase di training, non ha sviluppato i filtri convoluzionali capaci di identificare le microvariazioni di densità, lunghezza e luminosità tra le classi due e tre, rendendo il modello inaffidabile.

Rispetto allo scenario precedente, la configurazione 90% per il training e il 10% per la validation, ha permesso alla rete di consolidare il riconoscimento delle morfologie della classi 1 e 5; migliorano anche i risultati ottenuti nella classe 2, ma rimangono delle problematiche per quanto riguarda la classi 3 e 4, ad indicare l'incertezza dell'algoritmo nella classificazione intermedia in un ambito di danno continuo con limiti di classe spesso non definiti.

Avendo a disposizione un Validation Set esiguo, la rete assegna in maniera incorretta alla classe 4 una larga porzione di cellule invece appartenenti alla classe 4.

Nella figura 4.6 viene illustrata la matrice di confusione generata dal modello addestrato con uno split dell'80% per il training e del 20% per la validation, precedentemente eletta come configurazione ottimale. Si può notare che in questa matrice il comportamento del modello è più predittivo e meglio bilanciato rispetto ai precedenti.

Il primo dato di rilievo è la consolidazione delle prestazioni sulle classi estreme (classe 1 e 5). Si può notare che gli angoli della matrice presentano una prevalenza di zeri: ciò significa che la rete neurale ha azzerato gli errori gravi, e a differenza delle

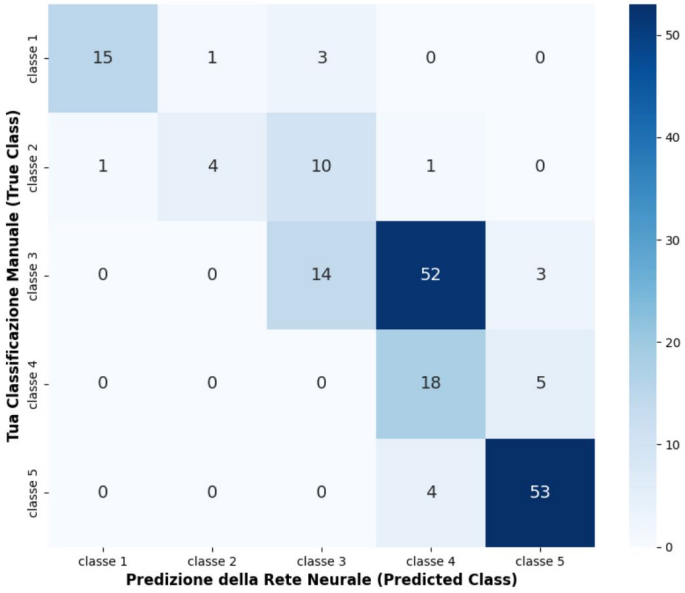

Figura 4.4: Matrice di confusione della casistica 70% traininhg e 30% validation

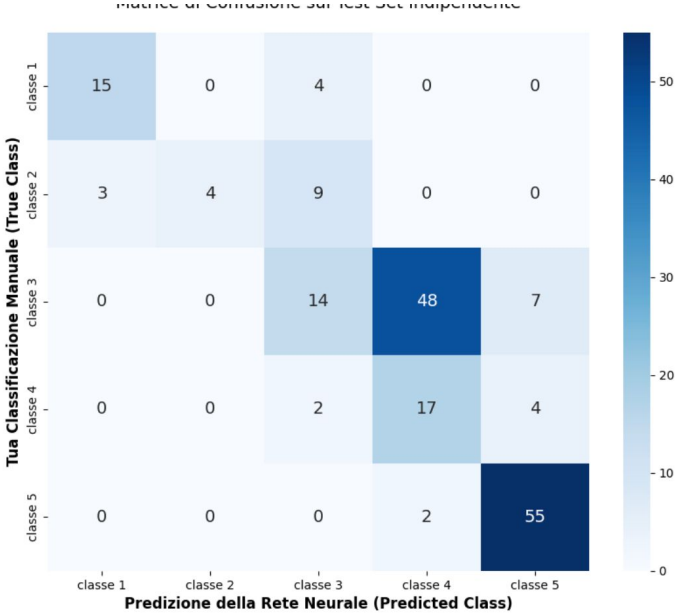

Figura 4.5: Matrice di confusione nella casistica 90% training e 10% validation

configurazioni precedenti gli errori si distribuiscono lungo le celle immediatamente adiacenti alla digonale principale. Seppure le prestazoini risultano migliori, la matrice conferma però la classe 3 come la morfologia più complessa da mappare: in tutti e tre i casi questa classe rimane quella che mostra il maggior numero di errori, sebbene con l'utilizzo di questo split gli errori risultano diminuiti rispetto ai tentativi precedenti.

In definitiva, la terza configurazione di split, rappresenta il miglior compromesso ingegneristico ottenibile su questo specifico set di dati.

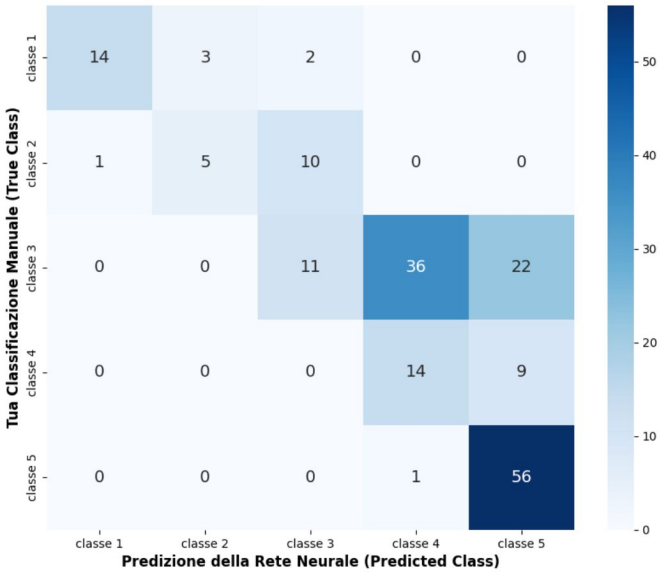

Figura 4.6: Matrice di confusione nella csistica 80% training e 20% validation

4.2.2 Analisi campioni *ex-novo*: risultati operatore vs. risultati ML

Per verificare l'affidabilità dell'algoritmo su dati non utilizzati in fase di addestramento, immagini di nuclei di cellule ottenuti *ex-novo* sono stati dapprima classificati da un operatore e poi sottoposti a classificazione dall'algoritmo sviluppato. I risulati sono stati poi sottoposti a confronto diretto per verifcare l'affidabilità delle predizioni algoritmiche.

Gli istogrammi riportati nelle figure 4.7, 4.8 e 4.9 illustrano la sovrapposizione tra la Classificazione Manuale (CM) e quella del Machine Learning (ML) per i tre partizionamenti testati (70-30, 90-10 e 80-20).

L'osservazione dei tre grafici permette di valutare come la rete abbia distribuito le proprie incertezze al variare della suddivisione del database.

- Casisitica 70-30 (figura 4.7): Il modello manifesta un evidente sbilanciamento verso la classe 4 (75 predizioni contro le 23 reali).

  L'algoritmo tende a sovrastimare il danno, classificando cellule di classe 3 in classe 4.

- Casistica 90-10 (figura 4.8): Anche in questo scenario persiste un massiccio accumulo di falsi positivi nella classe 4: è plausibile che avendo effettuato la validazione su un numero di campioni troppo ristretto, la rete non sia riuscita a generalizzare correttamente le transizioni morfologiche.

- Casistica 80-20 (figura 4.9): Nonostante permangano difficoltà per le classi centrali, a differenza delle classificazioni precedenti sposta il carico predittivo sulla classe 5, migliorando le predizioni della classe 4. Permane tuttavia una sovrastima della severità del danno.

Si può notare come in tutte e tre le casistiche la classe 3 sia quella con maggiore difficoltà di predizione corretta: questa discrepanza non deve essere vista come come un errore computazionale della macchina, ma bensì come il rifelsso della natura intrinsecamente sfumata del fenomeno osservato.

Il passaggio da una Classe 2 a una Classe 3 rappresenta un gradiente continuo di frammentazione del DNA.

L'operatore umano valuta queste transizioni basandosi sulla propria sensibilità visiva e sull'esperienza pregressa.

Al contrario, la rete neurale si affida puramente alla densità dei pixel e ai gradienti di intensità luminosa; di fronte all'ambiguità di una Classe 3 borderline, l'algoritmo tende matematicamente a ricollocare il dato in classi contigue in modo sistematico, distribuendo l'incertezza sui livelli di danno adiacenti.

In generale quando si progettano sistemi di diagnostica, bisogna tenere in conto il costo dell'errore, se il software classifica una classe 3 (danno medio) come una classe 1 (cellula sana), l'errore ha un costo maggiore, poichè si verificherebbe il rischio di ignorare una componente di danno rilevante.

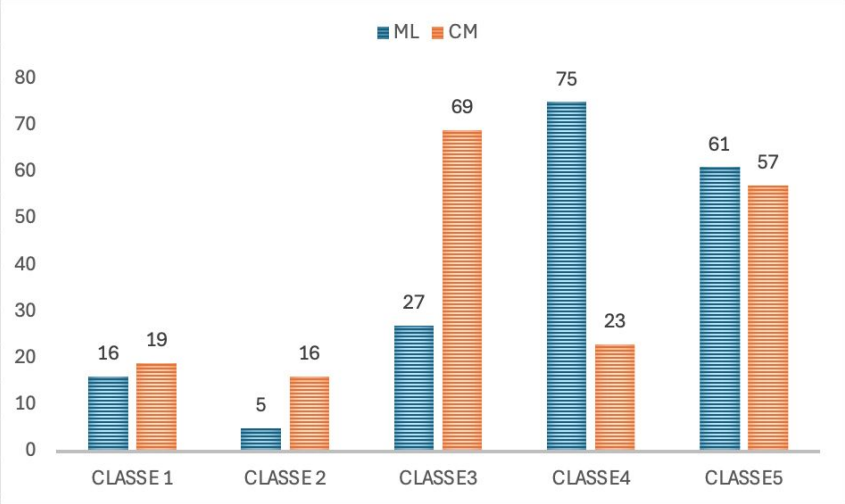

Figura 4.7: Risultati Machine Learning vs. risultati operatore umano nella casistica 70-30

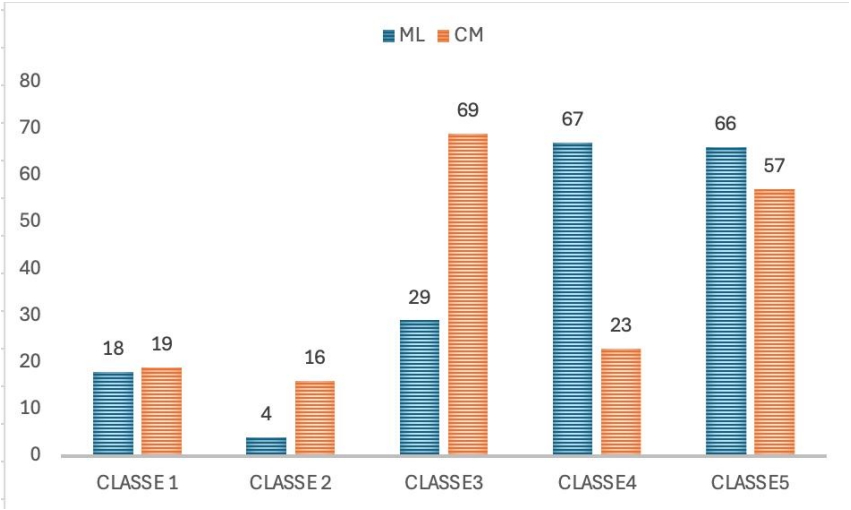

Figura 4.8: Risultati Machine Learning vs. risultati operatore umano nella casistica 90-10

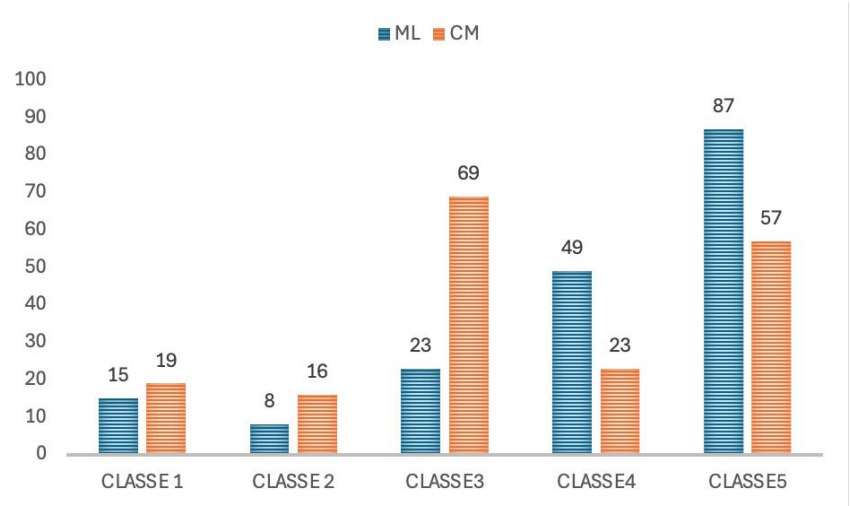

Figura 4.9: Risultati Machine Learning vs. risultati operatore umano nella casistica 80-20

4.3 Conclusioni e Prospettive

Il presente lavoro di tesi ha dimostrato la possibilità dell'impegno di architetture di Deep Learning, nello specifico di reti neurali per la classificazione dei risultati del Comet Assay. Tuttavia è risultata anche evidente la necessità di implementare l'efficienza di classificazione, verosimilmente ampliando la dimensione del dataset di addestramento.

Lo sviluppo del codice, il partizionamento del database e l'implementazione in ambiente cloud hanno generato un modello predittivo capace di processare le cellule con rapidità.

Tuttavia, l'analisi delle metriche di validazione ha fatto emergere riflessioni fondamentali sulla natura stessa del dato biologico, delinando le prospettive future per l'evoluzione di questo strumento.

Il risultato ottenuto sulle classi intermedie, in particolare per le classi 3 e 4 non rappresenta un limite dell'architettura informatica, bensì riflette la complessità biologica dell'informazione estrapolata dal campione e la natura continua di questa.

La valutazione del danno genotossico (che per natura è continuo) attraverso metriche di classificazione discrete porta con sé un'inevitabile difficoltà intrinseca, specialmente in cellule e nuclei bordeline: risultano in questo caso ancora evidenti difficoltà dell'algoritmo predittivo a identificare in maniera lineare questo continuum di danno e a contingentarlo in classi discrete.

Da questa consapevolezza derivano le principali direttrici per gli sviluppi futuri:

1. Necessità di espandere il set di dati: sebbene i risultati dimostrano che la topologia dei filtri convoluzionali sia corretta, l'attuale identificazione delle classi morfologiche di danno non risulta perfetta; per colmare i gap emersi per le transizioni intermedie, la soluzione non risiede in una riscrittura del codice ma bensì in un potenziamento massivo del database. Fornire all'algoritmo un archivio fotografico ben più vasto, che comprenda migliaia di varianti per ogni singola classe di frammentazione, permetterebbe alla rete di ridurre in modo significativo le ambiguità visive.

2. L'applicazione di algoritmi di Machine Learning basati su approcci quantitativi, piuttosto che qualitativi, rappresenta la soluzione ottimale in contesti operativi in cui la severità del danno segue una distribuzione continua. In questi scenari, infatti, le soglie di classificazione risultano spesso incerte, soggettive e prive di demarcazioni strutturali evidenti. Lo sviluppo di algoritmi in grado di applicare metriche quantitative dedicate alla misura della quantità di DNA nella coda od all'intensità di pixel nella testa potrebbe aumentare il potere applicativo e risolutivo sensibilmente.

In conclusione, l'algoritmo sviluppato fornisce una base solida e pronta all'uso, che se alimentata da un database molto più vasto di quello utilizzato, potrebbe portare a risultati più soldi e affidabili, trasformando questo applicativo prototipale in uno

strumento utile. Tale applicativo non si sostituirebbe agli standard analitici routinari ma risulterebbe integrabile, utile e di supporto nelle fasi di screening del potenziale genotossico.

Bibliografia

[1] Dhawan, A., Bajpayee, M., Parmar, D., Comet assay: a reliable tool for the assessment of DNA damage in different models, Cell Biology and Toxicology, 2009.

[2] Goodfellow, I., Bengio, Y., Courville, A., Deep Learning, MIT Press, 2016.

[3] Gajski, G., Ravlić, S., Godschalk, R., et al., Application of the comet assay for the evaluation of DNA damage in mature sperm, Mutation Research, 2021.

[4] Beleon, A., Pignatta, S., Arienti, C., et al., CometAnalyser: A user-friendly, open-source deep-learning microscopy tool for quantitative comet assay analysis, Computational and Structural Biotechnology Journal, 2022.

[5] Fairbairn, D.W., Olive, P.L., O'Neill, K.L., The comet assay: a comprehensive review, Mutation Research, 1995.

[6] Afiahayati, Anarossi, E., Yanuaryska, R.D., Mulyana, S., GamaComet: A Deep Learning-Based Tool for the Detection and Classification of DNA Damage from Buccal Mucosa Comet Assay Images, Diagnostics, 2022.

[7] Mehta, P., Namuduri, S., Barbe, L., et al., AI Enabled Ensemble Deep Learning Method for Automated Sensing and Quantification of DNA Damage in Comet Assay, ECS Sensors Plus, 2023.

[8] Atila, Ü., Baydilli, Y.Y., Sehirli, E., Turan, M.K., Classification of DNA damages on segmented comet assay images using convolutional neural network, Computer Methods and Programs in Biomedicine, 2020.

[9] Azqueta, A., Collins, A.R., The essential comet assay: a comprehensive guide to measuring DNA damage and repair, Archives of Toxicology, 2013.

[10] Hong, Y., Han, H.-J., Lee, H., et al., Deep learning method for comet segmentation and comet assay image analysis, Scientific Reports, 2020.

[11] Lecca, P., Lecca, M., Ihekwaba-Ndibe, A.E., AI-based methods for the assessment of DNA damage and repair mechanisms, Frontiers in Systems Biology, 2026.

[12] Katikaneni, R., Comet Former: A Novel Deep Learning Architecture using the U-MixFormer Model to optimize Comet Assay Segmentation, Research Square, 2025.

[13] LeCun, Y., Bengio, Y., Hinton, G., Deep learning, Nature, 2015.

[14] Choudhuri, S., Kaur, T., Jain, S., et al., A review on genotoxicity in connection to infertility and cancer, Chemico-Biological Interactions, 2021.