



UNIVERSITÀ DEGLI STUDI DI CAGLIARI

FACOLTÀ DI SCIENZE

Corso di Laurea Triennale in Informatica

***Classificazione di immagini istopatologiche con
tecniche di apprendimento miste***

Relatore

Dott. Andrea Loddo

Prof.ssa Cecilia Di Ruberto

Studente

Marco Usai

60/61/65762

Anno accademico: 2021-2022

CAPITOLO 1	3
INTRODUZIONE	3
CAPITOLO 2	4
MACHINE LEARNING E DEEP LEARNING	4
2.1 <i>Machine Learning</i>	4
2.2 <i>Deep Learning</i>	4
2.3 <i>Reti neurali</i>	4
2.3.1 <i>Reti neurali convoluzionali</i>	5
2.4 <i>Estrazione delle feature</i>	5
CAPITOLO 3	6
MATERIALI E METODI	6
3.1 <i>Dataset utilizzato</i>	6
3.2 <i>Feature handcrafted</i>	6
3.2.1 <i>Local Binary Pattern</i>	6
3.2.2 <i>Gray-Level Co-occurrence Matrix</i>	7
3.2.3 <i>Momenti di Zernike</i>	7
3.2.4 <i>Momenti di Chebyshev</i>	8
3.2.5 <i>Momenti di Legendre</i>	9
3.2.6 <i>Istogramma del colore</i>	9
3.2.7 <i>Auto-correlogram</i>	12
3.2.8 <i>Feature di Haar</i>	12
3.3 <i>Feature deep</i>	13
3.3.1 <i>AlexNet</i>	13
3.3.2 <i>GoogleNet</i>	13
3.3.3 <i>VGG</i>	13
3.3.4 <i>EfficientNet-b0</i>	13
3.3.5 <i>DenseNet</i>	14
3.3.6 <i>ResNet</i>	14
3.3.7 <i>DarkNet</i>	14
3.3.8 <i>Inception-v3</i>	14
3.3.9 <i>Inception-ResNet-v2</i>	14
3.3.10 <i>Xception</i>	15
3.4 <i>Classificazione</i>	15
3.5 <i>Classificatori utilizzati</i>	16
3.5.1 <i>k-Nearest Neighbors</i>	16
3.5.2 <i>Fine Tree</i>	16
3.5.3 <i>Random Forest</i>	16
3.5.4 <i>Ensemble</i>	16
3.5.5 <i>Support Vector Machine</i>	17
3.6 <i>Valutazione</i>	17
3.6.1 <i>Confusion matrix</i>	17
3.6.2 <i>Metriche utilizzate</i>	18
CAPITOLO 4	20
RISULTATI SPERIMENTALI	20
4.1 <i>Risultati ottenuti</i>	20
4.2 <i>Combinazione tra feature</i>	28
4.3 <i>Sperimentazione cross-dataset</i>	30
4.4 <i>Discussione</i>	32
CAPITOLO 5	34
CONCLUSIONE	34
RIFERIMENTI BIBLIOGRAFICI	35

Capitolo 1

INTRODUZIONE

Il tumore gastrico è il quinto tumore più comune che ci sia al mondo, nonché il quarto più letale. Questa patologia è addirittura accompagnata da una previsione piuttosto negativa, dato che il tasso di sopravvivenza globale a 5 anni è pari circa al 20%.

Nonostante vengano svolti numerosi studi sui parametri patobiologici della malattia, sussistono ancora varie perplessità circa la capacità di capire come si evolveranno tali dati tra i pazienti affetti, ragion per cui il tasso di sopravvivenza ha raggiunto quel valore. A questo problema si aggiunge il carico di lavoro dei medici, che è sempre molto pesante e lungo. Essi, pur osservando tali immagini al microscopio attentamente, è facile che possano incorrere in alcuni problemi; potrebbe capitare che ignorino dei particolari fondamentali, portando così a una diagnosi errata della malattia.

Sarebbe dunque utile un supporto informatico che possa svolgere automaticamente e con precisione delle diagnosi istopatologiche, riducendo al minimo la possibilità di imbattersi nelle problematiche appena citate.

In questi anni, la tecnologia informatica si è evoluta a tal punto da poter aiutare i medici a diagnosticare le immagini della patologia, grazie alle innovative tecniche di Machine Learning e Deep Learning. In questa tesi, viene spiegato il modo in cui le suddette immagini vengono classificate in due classi rappresentanti le immagini di cellule sane e tumorali. Per fare ciò, sono stati adoperati alcuni classificatori ed è stata effettuata l'estrazione di particolari feature handcrafted, e, tramite l'utilizzo di Reti Neurali Convoluzionali, l'estrazione di feature deep.

L'obiettivo del lavoro è quello di allenare tali classificatori al fine di creare dei modelli accurati che classifichino nel modo più corretto possibile le immagini istopatologiche nelle rispettive classi. Tale procedura è svolta tramite l'estrazione di feature specifiche per entrambe le tecniche di apprendimento.

I contributi realizzati sono molteplici:

- È stata svolta un'analisi comparativa tra una moltitudine di feature handcrafted e deep, in modo da orientare la ricerca su quelle più affini ai task affrontati;
- Sono stati addestrati ed esaminati cinque differenti algoritmi di Machine Learning, addestrati con tutte le feature così da poter offrire un metodo stabile e performante per la classificazione delle predette immagini istopatologiche, con relativa distinzione tra sane e malate;
- È stata effettuata un'analisi approfondita su ulteriori categorie di feature, generate dalla combinazione delle feature più performanti di entrambe le tipologie;
- Infine, è stata eseguita una classificazione cross-dataset in cui, attraverso i modelli creati per una particolare risoluzione di immagine, sono state classificate immagini di risoluzione e caratteristiche differenti.

Di seguito un elenco che definisce cosa si tratterà nei prossimi capitoli:

- Nel capitolo 2 è stata fatta una panoramica generale sulle tecniche di apprendimento utilizzate ed è stato descritto in che cosa consiste l'estrazione delle feature.
- Nel capitolo 3 è stato descritto il dataset di immagini studiato; sono state elencate le varie feature handcrafted e deep e sono stati descritti nel dettaglio le tecniche di classificazione e i vari processi che le costituiscono.
- Nel capitolo 4 sono stati analizzati i risultati ottenuti.
- Nel capitolo 5 sono state tratte le conclusioni.

Infine, si è tentato di adeguare i modelli delle immagini 160x160 per classificare dataset di immagini di dimensioni differenti.

Capitolo 2

MACHINE LEARNING E DEEP LEARNING

2.1 Machine Learning

Si definisce “Machine Learning” l’insieme di tecniche che si occupano di addestrare e migliorare dei sistemi in base ai dati che vengono forniti in input. Non è altro che una branca dell’intelligenza artificiale che utilizza vari metodi per migliorare l’accuratezza e le performance degli algoritmi che identificano determinati pattern di dati. Viene chiamato anche “Apprendimento automatico”, poiché l’identificazione dei vari pattern svolto dagli algoritmi viene effettuato in maniera autonoma, senza istruzioni esplicite. Utilizzano algoritmi di tipo supervisionato o non supervisionato. I primi svolgono l’analisi di pattern a seconda degli input passati, facendo automaticamente delle predizioni accurate sui valori in output sulla base di esempi ideali costituiti da coppie di input; i secondi classificano i dati in input in base a ciò che hanno già imparato in precedenza, secondo le caratteristiche comuni, cercando di effettuare ragionamenti e predizioni sui successivi input. Perché ciò avvenga, è necessario stabilire delle metriche o particolari feature utilizzate per distinguere ogni immagine. Si procede con l’estrazione di tali feature, tramite le quali si addestrano dei classificatori per fare predizioni il più accurate possibile [22].

2.2 Deep Learning

Il Deep Learning è una particolare tecnica di Machine Learning che utilizza reti neurali, particolari reti che simulano il comportamento della mente umana, formate da neuroni interconnessi che gestiscono i dati. Esse verranno spiegate nel dettaglio nel prossimo paragrafo. Viene chiamato anche “Apprendimento profondo” e si definisce come quella classe di algoritmi che utilizzano vari livelli a cascata per estrarre feature. La tecnica è in grado di apprendere in modo non supervisionato livelli gerarchici multipli di feature di dati. Ogni livello ha come input l’output del livello precedente.

2.3 Reti neurali

Una rete neurale è un’operazione matematica che prende degli input e restituisce output. Gli input sono identificati come una serie di neuroni, che rappresentano i dati che forniranno al neurone in output un certo valore di previsione. Sono necessari tre elementi fondamentali per creare una rete: Training data, architettura di rete e opzioni algoritmiche. Il primo è un insieme di esempi in input, spesso vettori di valori di attributi continui o discreti, ad ognuno dei quali è associata una risposta in output, solitamente un valore di una classe o numerico. Tali esempi formano un modello supervisionato, che sarà in grado di determinare il valore-obiettivo per nuovi dati. La seconda dà una struttura generale sulla matematica dietro alla mappatura tra input e output; tuttavia la rete possiede molti parametri conformi: pesi e bias dei layer [28]. I pesi definiscono la quantità di input trasferito tra neuroni, il bias è un valore costante che viene sommato al calcolo che coinvolge un determinato peso. Pesi e bias possono essere definiti come una serie di manopole che regolano il modello d’interesse per ottimizzarlo. I pesi vengono moltiplicati ai valori dei dati in input e sommati al bias, costituendo così una funzione di attivazione, funzione che definisce l’output finale di ogni neurone della rete. Dunque, in base alla scelta dei parametri, la stessa architettura può mappare gli stessi input in diversi output. Durante l’addestramento di una rete, i parametri vengono determinati dalla scelta dei valori in input. Questa determinazione dipende dall’errore, definito da una particolare funzione detta loss function, che vi è tra le predizioni della rete e gli output conosciuti. I valori dei parametri con la minima quantità di errore saranno quelli presi in considerazione. L’addestramento di una rete consiste nel minimizzare questa quantità di errore rifinando i parametri, mediante la tecnica della retropropagazione

dell'errore, che consiste, invece di calcolare il valore di loss mediante i parametri, nel calcolare i valori dei parametri tramite la discesa del gradiente, una tecnica che aggiorna i valori dei parametri di una certa quantità lungo la direzione in cui la funzione di perdita decresce il più velocemente possibile, con una matematica complessa e calcolandone la derivata parziale rispetto al parametro considerato [25].

2.3.1 Reti neurali convoluzionali

Le reti neurali convoluzionali, o CNN, sono un tipo di reti neurali usate dove gli input hanno una struttura bidimensionale, come le immagini. Sono composte da layer e le informazioni vengono trasferite tra layer. L'output rappresenta il livello di qualità delle predizioni effettuate dalla rete. Le immagini in input vengono trasformate in un set di caratteristiche che gli ultimi layer utilizzano per classificare. Questo avviene mediante l'architettura della rete, che stabilisce l'ordine dei layer. I layer contengono tanti parametri, detti pesi, che determinano come debbano comportarsi quando analizzano dei dati. I loro valori sono determinati dal training della rete su certe label, etichette rappresentanti le classi a cui appartengono i dati. Due reti con la stessa architettura potrebbero comportarsi diversamente a seconda dei dati training che vengono passati a entrambi, perciò dal training dipende la qualità delle predizioni e la precisione di una rete [26].

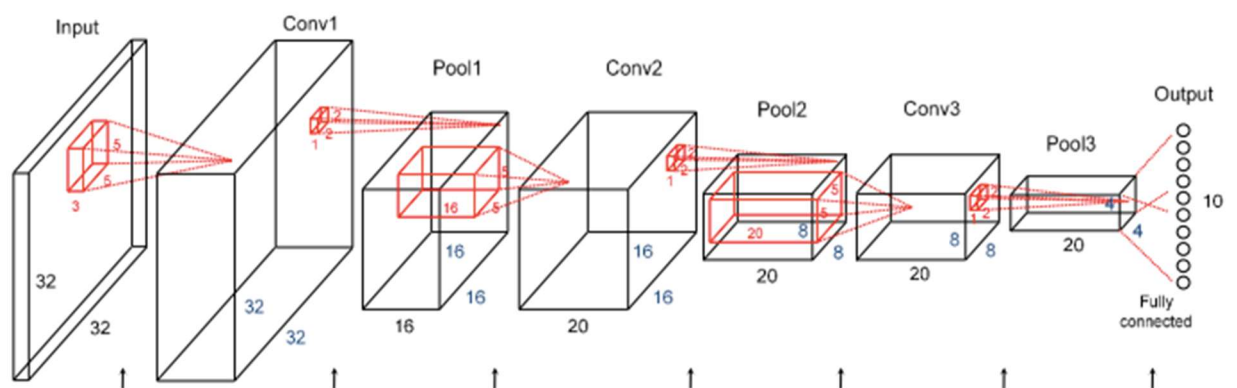


Figura 2.1: Esempio di rete neurale convoluzionale.

2.4 Estrazione delle feature

Una feature è una proprietà individuale e misurabile di un dato e favorisce l'apprendimento delle tecniche precedentemente descritte. Le feature possono essere di vari tipi: alcune corrispondono alla tessitura, altre al colore e altre a vari aspetti matematici, che nel complesso formano il dato nel suo insieme.

L'estrazione delle feature è un prerequisito fondamentale del Machine Learning e del Deep Learning. Essa consiste nell'estrarre le feature da un insieme di dati, che avranno un ruolo puramente informativo, esse infatti permettono l'apprendimento da parte delle due tecniche.

Molti esperti ritengono che questa tecnica sia, difatti, la chiave per creare dei modelli efficaci.

Per quanto concerne il Machine Learning, vi è la necessità di estrarre le cosiddette feature handcrafted. Per il Deep Learning, vengono estratte le feature deep. Nel capitolo seguente, verranno mostrate quali feature nello specifico sono state estratte per questo esperimento.

Capitolo 3

MATERIALI E METODI

3.1 Dataset utilizzato

Per lo studio delle immagini, è stato impiegato il dataset GasHisSDB [3], suddiviso in tre sottocartelle secondo una dimensione diversa. Il lavoro è stato svolto inizialmente sulla sottocartella 160x160, che contiene ben 33284 immagini di tessuti di quella dimensione in pixel, distribuite in due sottocartelle rappresentanti la classe “normale” e la classe “anormale”. Un’immagine appartiene alla classe normale se non contiene regioni cancerogene, sostanzialmente se le cellule osservate al microscopio sono regolari. Viceversa, appartiene alla classe anormale quell’immagine che contiene più del 50% di aree cancerogene. Per eseguire tali estrazioni e classificazioni, si è diviso l’intero dataset di immagini in due parti: una utilizzata per l’addestramento, contenente l’80% di immagini e una utilizzata per il testing, contenente il 20%, allo stesso modo di come si osserva nell’articolo dedicato [1]. Le altre due sottocartelle contengono immagini istopatologiche di altre dimensioni: una con immagini 120x120 e una con immagini 80x80, entrambe in quantità differenti e distribuite allo stesso modo nelle due sottocartelle in base alla presenza o assenza delle aree infette.

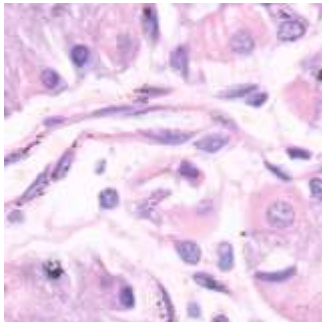


Figura 3.1: Esempio di immagine “normale”, che non contiene alcuna regione cancerogena, perciò è regolare.

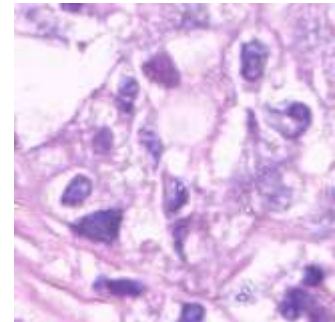


Figura 3.2: Esempio di immagine “anormale”, contenente aree cancerogene.

3.2 Feature handcrafted

3.2.1 Local Binary Pattern

Local Binary Pattern (LBP) [4] è un particolare tipo di descrittore. Il suo funzionamento consiste in:

- Dividere l’immagine in celle
- Per ogni pixel in ogni cella, comparare quello centrale a quelli più vicini in tutte le direzioni
- Quando il pixel centrale ha un valore maggiore di quello dei suoi vicini, scrivere “0”, altrimenti “1”. Si otterrà così un numero binario

- Computare l'istogramma risultante per ogni cella, in modo da osservare quali pixel siano minori o maggiori di quello centrale
- Normalizzare l'istogramma
- Concatenare tutti gli istogrammi, operazione che fornirà il vettore di feature di tutta l'immagine.

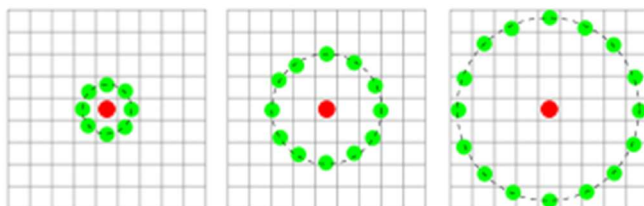


Figura 3.3: Esempio di LBP.

Nell'esperimento in questione è stato utilizzato l'LBP18, che prevede un pixel centrale e 8 vicini in tutte le direzioni.

La dimensione dell'LBP nel dataset di immagini dello studio in questione è risultata $N \times 36$.

3.2.2 Gray-Level Co-occurrence Matrix

La Gray-level Co-occurrence Matrix, o GLCM, è una matrice definita per un'immagine digitale e si calcola matematicamente con la seguente formula:

$$C_{\Delta x, \Delta y}(i, j) = \sum_{p=1}^n \sum_{q=1}^m \begin{cases} 1, & I(p, q) = i \text{ e } I(p + \Delta x, q + \Delta y) = j \\ 0, & \text{altrimenti} \end{cases}$$

L'immagine I considerata ha dimensioni $n \times m$ pixel e il suo valore si riferisce al valore di grigio di un determinato pixel, che può essere qualsiasi, da un numero binario a un valore di 32 bit per immagini a colori. Inoltre è parametrizzata da un offset $\Delta x, \Delta y$ che rende la matrice sensibile alle rotazioni [7].

In particolare, sono state estratte le cosiddette feature di Haralick. Queste feature sono utilizzate per classificare un'immagine secondo importanti informazioni che derivano dai pattern della texture, ovvero regole ripetute che stabiliscono uno schema di una feature. Questo tipo di caratteristiche sono calcolate utilizzando la GLCM. Alla fine della computazione si ottiene una matrice nella quale sono contenute le Haralick features, di seguito elencate: Energy, Correlation, Inertia, Entropy, Inverse Difference Moment, Sum Average, Sum Variance, Sum Entropy, Difference Average, Difference Variance, Difference Entropy, Information measure of correlation 1, Information measure of correlation 2. Per un approfondimento delle singole feature si consiglia l'articolo originale [21].

Nel caso dello studio in questione, è stata ottenuta una matrice dalla dimensione $N \times 52$.

3.2.3 Momenti di Zernike

I momenti, o polinomi, di Zernike sono una sequenza polinomiale di polinomi ortogonali sul disco unitario [8].

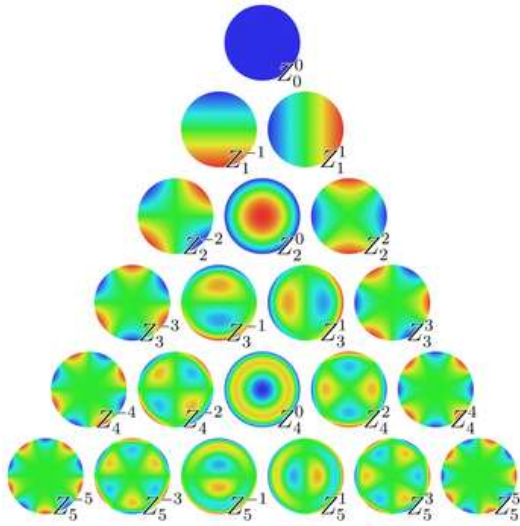


Figura 3.4: I primi 21 polinomi di Zernike, ordinati verticalmente secondo il grado radiale e orizzontalmente secondo il grado azimutale [30].

Nell'esperimento sono stati utilizzati diversi polinomi di Zernike contraddistinti dal range. Nello specifico sono stati impiegati i seguenti range: 4-2, 4-4, 5-3, 5-5, 6-2, 6-4, 6-6.

3.2.4 Momenti di Chebyshev

I momenti, o polinomi, di Chebyshev [9] sono le componenti di particolari successioni polinomiali, contraddistinti in polinomi di prima e seconda specie. La prima è costituita dai seguenti polinomi:

$$T_0(x) = 1;$$

$$T_1(x) = x;$$

$$T_2(x) = 2x^2 - 1;$$

$$T_3(x) = 4x^3 - 3x;$$

$$T_4(x) = 8x^4 - 8x^2 + 1;$$

$$T_5(x) = 16x^5 - 20x^3 + 5x;$$

$$T_6(x) = 32x^6 - 48x^4 + 18x^2 - 1;$$

$$T_7(x) = 64x^7 - 112x^5 + 56x^3 - 7x;$$

$$T_8(x) = 128x^8 - 256x^6 + 160x^4 - 32x^2 + 1;$$

$$T_9(x) = 256x^9 - 576x^7 + 432x^5 - 120x^3 + 9x.$$

La seconda specie, invece, è costituita dai seguenti polinomi:

$$U_0 = 1;$$

$$U_1 = 2x;$$

$$\begin{aligned}
U_2 &= 4x^2 - 1; \\
U_3 &= 8x^3 - 4x; \\
U_4 &= 16x^4 - 12x^2 + 1; \\
U_5 &= 32x^5 - 32x^3 + 6x; \\
U_6 &= 64x^6 - 80x^4 + 24x^2 - 1; \\
U_7 &= 128x^7 - 192x^5 + 80x^3 - 8x; \\
U_8 &= 256x^8 - 448x^6 + 240x^4 - 40x^2 + 1; \\
U_9 &= 512x^9 - 1024x^7 + 672x^5 - 160x^3 + 10x.
\end{aligned}$$

I polinomi con grado pari rappresentano delle funzioni pari, mentre quelli con grado dispari rappresentano funzioni dispari.

3.2.5 Momenti di Legendre

I momenti di Legendre sono le soluzioni dell'equazione di Legendre

$$(1 - x^2)y'' - 2xy' + ky = 0 \quad x \in (-1,1).$$

L'equazione di Legendre è un'equazione differenziale che si incontra in fisica e altri settori tecnologici, ad esempio nella soluzione in coordinate sferiche dell'equazione di Laplace o equazioni differenziali alle derivate parziali.

Tali soluzioni formano una successione di polinomi specifici, detti polinomi di Legendre:

$$\begin{aligned}
P_0(x) &= 1; \\
P_1(x) &= x; \\
P_2(x) &= \frac{1}{2}(3x^2 - 1); \\
P_3(x) &= \frac{1}{2}(5x^3 - 3x); \\
P_4(x) &= \frac{1}{8}(35x^4 - 30x^2 + 3); \\
P_5(x) &= \frac{1}{8}(63x^5 - 70x^3 + 15x); \\
P_6(x) &= \frac{1}{16}(231x^6 - 315x^4 + 105x^2 - 5).
\end{aligned}$$

3.2.6 Istogramma del colore

Quando si osserva un'immagine digitale, solitamente si pensa di stare osservando tantissimi colori diversi. In realtà, il numero di colori presenti dipende dallo spazio di colore che viene considerato. Esistono più spazi di colore che vengono impiegati in diversi campi, come il cinema, la stampa o la televisione. Lo spazio di colore maggiormente utilizzato è il modello RGB, le cui specifiche sono state descritte nel 1936 dalla Commission internationale de l'éclairage [31]. che ha effettivamente solo 3 colori. Ogni pixel che compone un'immagine può assumere un valore differente tra rosso, blu e verde, e a seconda della loro intensità, determinata quantitativamente da un numero che va da 0 a

255, può assumere anche altri colori e tonalità. Ad esempio, perché si noti il colore rosso, i pixel attivi devono essere soltanto quelli di colore rosso. Perché si veda un rosso più scuro, quel pixel deve abbassare la quantità di rosso. Perché si veda il colore bianco, i pixel devono avere al contempo attivi il rosso, il blu e il verde [19].

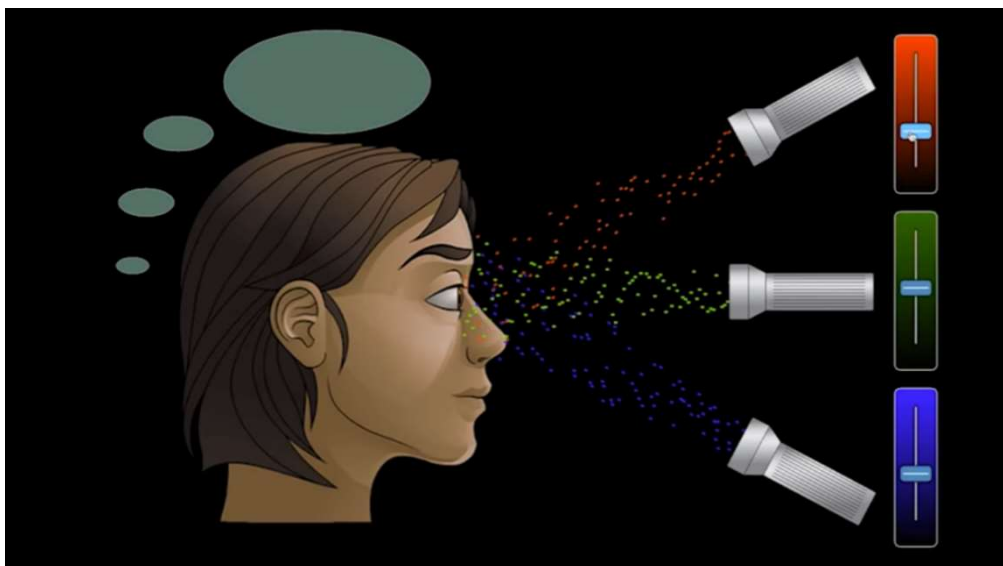


Figura 3.5: *Raffigurazione della percezione dei colori di un'immagine digitale agli occhi umani. I pixel attivi con le differenti intensità RGB permettono di osservare un altro colore.*

L'istogramma del colore è la rappresentazione della distribuzione di colori in un'immagine. Esso riporta il numero di pixel contenenti dei colori entro una lista fissata di un certo range che va da 0 a 255 per il rosso, il blu e il verde.



Figura 3.6: Esempio di istogramma del colore. Ogni colore osservabile nell'immagine è descritto dall'andamento degli istogrammi a destra. L'arancione acceso della luce del Sole è visibile dove i grafici presentano un picco di rosso e verde e un livello basso di blu, sull'estrema destra.

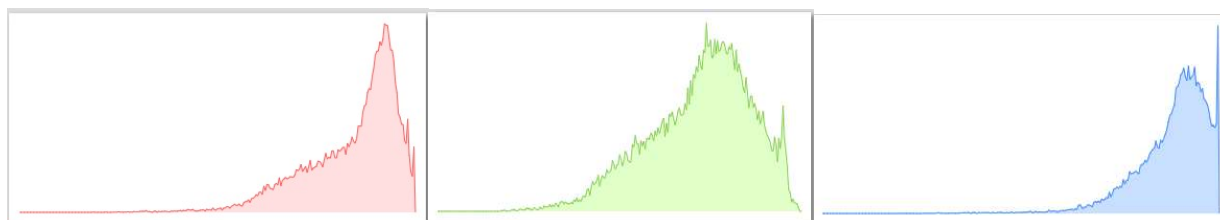
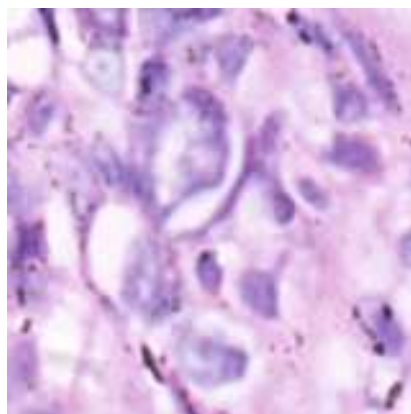


Figura 3.7: Esempio di istogramma del colore applicato a un'immagine istopatologica anormale.

3.2.7 Auto-correlogram

L'auto-correlogram è un grafico che raffigura l'autocorrelazione di una serie storica in funzione del ritardo con cui viene calcolata. Una serie storica è l'insieme di valori che un certo fenomeno assume in successivi istanti di tempo.

Ponendo il caso che Y sia un fenomeno e t gli istanti di tempo in cui esso avviene, una serie si dice che presenta un ritardo in una nuova variabile, ad esempio X, se:

$$X_t = Y_{t-K}$$

dove K è la variabile del ritardo.

Di seguito è rappresentata un'idea della successione della serie storica:

Y_t	Y_{t-1}	Y_{t-2}	Y_{t-3}	\dots	Y_{t-K}
Y_1					
Y_2	Y_1				
Y_3	Y_2	Y_1			
Y_4	Y_3	Y_2	Y_1		
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
Y_{T-2}	Y_{T-3}	Y_{T-4}	Y_{T-5}	\vdots	Y_{T-K-2}
Y_{T-1}	Y_{T-2}	Y_{T-3}	Y_{T-4}	\vdots	Y_{T-K-1}
Y_T	Y_{T-1}	Y_{T-2}	Y_{T-3}	\vdots	Y_{T-K}

dove T è l'ultimo istante di tempo.

Nel caso del nostro esperimento, la serie storica di variabili riguarda le numerose feature che compongono l'intero dataset di immagini.

3.2.8 Feature di Haar

Con feature di Haar [5] s'intende una serie di filtri per immagini sotto forma di sommatorie e sottrazioni di parti rettangolari delle immagini stesse.



Figura 3.8: Esempi di feature di Haar. Nelle aree in bianco l'area sottesa viene sommata e in quelle in nero viene sottratta.

Il valore di tali filtri deriva dalla somma delle tonalità di grigio dei pixel nelle aree in bianco meno il valore dei pixel nelle aree in nero.

3.3 Feature deep

Sono state estratte le feature del dataset calcolando le attivazioni delle seguenti CNN:

3.3.1 AlexNet

AlexNet è una delle architetture di rete neurale più popolari al giorno d'oggi. Fu ideata da Alex Krizhevsky per l'ImageNet Large Scale Visual Recognition Challenge, e ha raggiunto un tasso di errore del 15.3%.

AlexNet è una rete neurale convoluzionale che prende in input immagini 227x227x3. Contiene 25 layer ed è profonda 8, di cui i primi 5 sono convoluzionali e gli ultimi 3 totalmente connessi.

I primi due strati sono collegati a strati di max-pooling sovrapposti, in modo così da estrarre un numero massimo di caratteristiche. Il terzo, quarto e quinto strato convoluzionale sono direttamente collegati agli strati completamente connessi. Gli output dei layer convoluzionali e di quelli completamente connessi, sono collegati alla funzione di attivazione non lineare ReLU. Lo strato finale di output è collegato ad uno strato di attivazione softmax. Per diminuire l'overfitting, che verrà spiegato meglio in un paragrafo successivo, durante il processo di training, la rete utilizza strati di dropout [10].

3.3.2 GoogleNet

GoogleNet è una rete neurale convoluzionale che prende in input immagini 224x224x3. Contiene 144 layer ed è profonda 22.

Se una rete è costituita da molti strati profondi, può incorrere nel problema dell'overfitting. Per risolvere questo problema è stata proposta quest'architettura, mettendo a disposizione filtri con più dimensioni che possono operare sullo stesso livello. Con ciò, la rete diventa in realtà più ampia piuttosto che più profonda. L'operazione di convoluzione viene eseguita su input con tre dimensioni di filtro: (1x1), (3x3) e (5x5) [16].

3.3.3 VGG

VGG è una comune architettura di rete neurale, proposta da Karen Simonyan ed Andrew Zisserman dell'Università di Oxford. Fu applicata all'ImageNet Challenge nel 2014, ottenendo un'accuracy del 92.7%.

VGG19 è una rete neurale convoluzionale che prende in input immagini 224x224x3 ed è stata utilizzata per quest'esperimento. Contiene 47 layer ed è profonda 19 [6].

3.3.4 EfficientNet-b0

EfficientNet-b0 è una rete neurale convoluzionale che prende in input immagini 224x224x3. Contiene 290 layer. Risulta molto simile a GoogleNet.

EfficientNet ha la facoltà di scalare ogni dimensione di profondità, larghezza o risoluzione utilizzando un coefficiente di compressione. Invece di scalare arbitrariamente come molte altre reti, EfficientNet utilizza un set di coefficienti corretto apposta per i dati d'interesse [17].

3.3.5 DenseNet

DenseNet è una rete neurale convoluzionale che prende in input immagini $224 \times 224 \times 3$ e per questo esperimento è stata utilizzata la versione 201, la quale contiene 201 layer.

Per ogni layer, le feature dei precedenti layer vengono utilizzate come input dei successivi. DenseNet possiede dei vantaggi: rafforza la propagazione delle feature, utilizza il riuso di feature e riduce notevolmente il numero di parametri [18].

3.3.6 ResNet

Poiché le reti neurali profonde sono molto difficili da addestrare e possono risultare overfitting, un team di Microsoft ha introdotto un framework per eseguire un training della rete in modo più efficiente e ottenere ottimi risultati pur avendo una notevole profondità. Quando si addestrano reti profonde l'accuratezza potrebbe incomberne in un degrado. Questo è il "problema del degrado" e viene trattato da questa architettura. ResNet utilizza la tecnica del "residual mapping" per combattere questo problema. ResNet, attraverso questa, invece di provare ad apprendere alcune caratteristiche, prova da apprendere dei residui, che non sono altro che la differenza tra l'input di un livello e la caratteristica estratta. ResNet, a differenza di altre reti, riesce a migliorare i risultati di classificazione all'aumentare della profondità.

ResNet prende in input immagini $224 \times 224 \times 3$. Contiene un diverso numero di layer in base alla variante utilizzata. Nell'esperimento svolto in questa tesi, si sono utilizzate le versioni con 18, 50 e 101 layer di profondità.

3.3.7 DarkNet

DarkNet è una rete neurale convoluzionale che prende in input immagini $256 \times 256 \times 3$. Contiene un diverso numero di layer in base alla variante utilizzata. Nell'esperimento svolto in questa tesi, si sono utilizzate le versioni con 19 e 53 layer di profondità.

3.3.8 Inception-v3

Inception-v3 è una rete neurale convoluzionale che prende in input immagini $299 \times 299 \times 3$. Contiene 316 layer in tutto ed è profonda 48. Inception-v3 si concentra sulla riduzione della potenza di calcolo a differenza delle precedenti versioni di Inception. Questa idea è stata proposta nel documento "Rethinking the Inception Architecture for Computer Vision" [29], pubblicato nel 2015. È stato scritto da Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe e Jonathon Shlens. Rispetto alle reti VGG, le Inception hanno dimostrato una computazione più efficiente, sia per quanto riguarda il numero di parametri generati dalla rete che per il costo delle risorse. Nel modello Inception-v3, sono state introdotte delle tecniche per ottimizzare la rete a ridurre i vincoli e facilitare l'adattamento del modello, ad esempio convoluzioni fattorizzate, regolarizzazione, riduzione delle dimensioni e calcoli parallelizzati [12].

3.3.9 Inception-ResNet-v2

Inception-ResNet-v2 è una rete neurale convoluzionale che prende in input immagini $299 \times 299 \times 3$. Contiene 825 layer in tutto ed è profonda 164.

La rete è progettata come una combinazione fra la struttura dell'Inception e di ResNet. Molti filtri convoluzionali di varia dimensione sono combinati con connessioni residue, tipiche di ResNet. L'uso di queste ultime permette di evitare il problema del degrado e riduce il tempo di training [11].

3.3.10 Xception

Xception è una rete neurale convoluzionale che prende in input immagini 299x299x3. Contiene 171 layer in tutto ed è profonda 71.

Xception è un'evoluzione dell'Inception.

Nell'Inception, sono state utilizzate convoluzioni 1x1 per comprimere l'input originale, e per ognuno degli input sono stati usati diversi filtri. Xception agisce in modo contrario: prima applica i filtri su ogni profondità e poi comprime gli input con convoluzioni 1x1. Inoltre, a differenza dell'Inception, Xception non utilizza operazioni non lineari [13].

3.4 Classificazione

La classificazione è un procedimento col quale una serie di dati viene catalogata in diverse categorie in base ai risultati dell'addestramento di un modello, mediante l'utilizzo di un classificatore. In questo esempio, si vuole classificare una serie di immagini, nelle categorie "normale" e "anormale". Tuttavia, è possibile che il modello statistico creato dalla classificazione si basi su troppi parametri, ciò non permette una performance accurata della classificazione quando si lavora su nuovi dati perché il modello diventa troppo sensibile a loro. In questo caso, si ha il cosiddetto "overfitting".

Il procedimento utilizzato per effettuare e verificare tale classificazione è suddiviso in quattro step fondamentali:

- Allenare un classificatore e creare il modello contenente tutti i dati di "training", cioè utilizzati per l'addestramento della classificazione, con le rispettive "label" o etichette, che rappresentano le classi a cui apparterranno, in quantità pari al numero di dati coinvolti. Tali dati non sono altro che le feature estratte, di cui abbiamo parlato in precedenza.
- Effettuare delle predizioni su dei particolari dati di test, dello stesso dataset e solitamente in quantità minore di quelli di training.
- Generare una "matrice di confusione", riportante le statistiche che descrivono quanto siano accurate le predizioni appena eseguite, mediante l'utilizzo di label di test.
- Estrarre tali statistiche per verificare i risultati quali l'accuratezza, la precisione, l'f1-score e altri parametri che verranno spiegati meglio nel capitolo successivo.

Ogni step verrà descritto nei paragrafi che seguono.

```
% kNN - LBP:
modellBPkNN = fitcknn( featuresLBPTv, YTrain);
predLBPKNN = predict(modelLBPKNN, featuresLBPTest);
cmLBPKNN = confusionmat(YTest, predLBPKNN);
```

Figura 3.9: Esempio di codice eseguito in Matlab per allenare il classificatore kNN per creare il modello contenente le feature LBP nelle rispettive label; effettuazione delle predizioni e generazione della confusion matrix.

3.5 Classificatori utilizzati

3.5.1 k-Nearest Neighbors

Il k-Nearest Neighbors, o kNN, è un tipo di classificatore di apprendimento non parametrico, che effettua classificazioni e previsioni basandosi sulle caratteristiche degli oggetti vicini a quello considerato. L'input è costituito da k dati di training. L'output è il valore della proprietà dell'oggetto, cioè la media dei valori di k più vicini.

Si compone di tre fasi principali:

- una fase di apprendimento, dove lo spazio viene spartito in regioni in base alle posizioni e caratteristiche degli oggetti di training
- il calcolo della distanza, di solito euclidea o Manhattan
- una fase di classificazione, dove a ogni oggetto viene assegnata una classe se questa è la più frequente fra i k oggetti più vicini a quello considerato.

3.5.2 Fine Tree

Il Fine Tree, o Decision Tree, è un particolare tipo di albero che utilizza una struttura dati ad albero per effettuare predizioni. Esso è un albero di classificatori dove ogni nodo interno è associato ad una particolare feature. Da questo nodo dipartono tanti archi quanti sono i possibili valori che la feature può assumere, mentre le foglie rappresentano le categorie. L'algoritmo di Fine Tree seleziona i nodi che hanno contribuito in modo minore alla suddivisione in categorie, e li elimina e unisce le istanze al livello superiore. Al termine, è possibile ripercorrere l'albero dalla radice verso il basso per ottenere la classificazione dell'istanza [24].

3.5.3 Random Forest

Il Random Forest è un classificatore ottenuto dall'aggregazione di più alberi decisionali mediante il bagging, che consiste nell'assemblare più alberi riducendone la varianza, ovvero la caratteristica che porta i dati di training a variare fino ad ottenere un modello completamente diverso. Inoltre, questo classificatore viene utilizzato per minimizzare l'overfitting dei dati di training [15].

3.5.4 Ensemble

L'Ensemble è un classificatore che mette insieme le predizioni di più modelli per aumentarne le prestazioni. Si compone di tre principali tecniche: il Bagging, il Boosting e lo Stacking [23].

- Il Bagging porta a creare un insieme di classificatori, ognuno dei quali effettuerà delle predizioni sul modello e verrà scelta la classe coi migliori risultati. I risultati di classificazione, sia su training che su test, verranno utilizzati per avere un'unica risposta conclusiva.
- Il Boosting, invece, è simile al precedente ma ogni classificatore influenza la risposta conclusiva, tramite un peso che verrà calcolato mediante l'errore di accuratezza di ciascuno dei classificatori. Si rischia l'overfitting.
- Lo Stacking introduce un nuovo classificatore, detto meta-classificatore, che utilizza le predizioni di altri sotto-modelli per un ulteriore addestramento.

3.5.5 Support Vector Machine

La Support Vector Machine, o SVM, è un particolare classificatore che ha come obiettivo trovare la retta di separazione che massimizza il margine tra le classi, ovvero la distanza minima dalla retta ai punti delle classi [14].

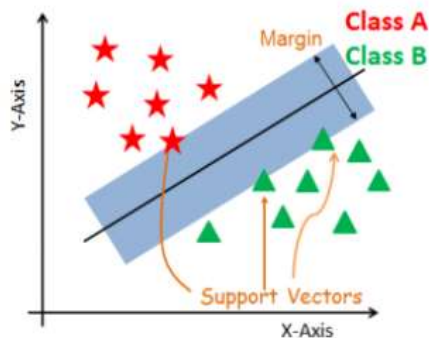


Figura 3.10: Nell'immagine vi è la retta che massimizza il margine tra le classi. Le stelle e i triangoli sono i vettori di supporto che risiedono sul margine, nonché i valori di una classe più vicini alla retta, complessivamente in minor quantità.

La particolarità dell'SVM, a differenza degli altri classificatori, è quella di considerare nel training i valori meno rilevanti del modello, cioè quelli che tendono a far parte dell'altra classe. Se la classificazione è adatta con gli esempi più complessi, lo sarà sicuramente anche per tutti i restanti.

3.6 Valutazione

3.6.1 Confusion matrix

La matrice di confusione, detta anche tabella di errata classificazione, fornisce la rappresentazione dell'accuratezza di una classificazione [20].

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figura 3.11: Struttura di una confusion matrix.

Si distinguono quattro casistiche differenti:

- True positive, TP, quando la classe prevista 1 corrisponde alla classe reale 1. Il modello ha previsto correttamente le classi.
- True negative, TN, quando la classe prevista 0 corrisponde alla classe reale 0. Il modello ha previsto correttamente le classi.
- False positive, FP, quando la classe prevista 1 è diversa dalla classe reale, che è 0. Il modello ha sbagliato a rispondere affermativamente.
- False negative, FN, quando la classe prevista 0 è diversa dalla classe reale, che è 1. Il modello ha sbagliato a rispondere negativamente.

3.6.2 Metriche utilizzate

Al fine di ottenere i risultati, sono state impiegate le seguenti metriche:

- Accuracy: rapporto tra numero di predizioni corrette e la quantità totale di predizioni.

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

- Precision: rapporto tra True Positive e la somma dei positivi, True Positive e False Positive. Misura quanto sia efficiente il modello quando la predizione è positiva.

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

- Recall: misura la sensibilità del modello. Rapporto tra True Positive e la somma tra True Positive e False Negative. Garantisce che siano presenti molti True Positive e pochi False Negative, per questo è la metrica maggiormente richiesta in ambito economico e medico, inoltre si usa parecchio nelle classificazioni binarie.

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

- Specificity: Rapporto tra True Negative e la somma tra True Negative e False Positive

$$specificity = \frac{TN}{TN + FP}$$

- F1-Score: media pesata tra Precision e Recall. Si ritiene sia una metrica maggiormente indicata dell'accuracy, poiché considera anche i False Positive e i False Negative.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

- MCC: Matthew Correlation Coefficient, coefficiente che considera tutte le possibilità delle classificazioni binarie, ovvero entrambi i Positive ed entrambi i Negative, e risulta parecchio efficiente per dataset sbilanciati, infatti molto utile per questo esperimento.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

- BACC: Balanced Accuracy, adatta per trattare dati sbilanciati. Prende in considerazione la Recall e la Specificity [27].

$$BACC = \frac{recall + specificity}{2}$$

Capitolo 4

RISULTATI SPERIMENTALI

4.1 Risultati ottenuti

Per ogni coppia costituita da classificatore allenato e feature estratta, sono state calcolate le rispettive confusion matrix, da cui sono state ricavate le statistiche, che hanno fornito il valore esatto di ogni metrica citata nel paragrafo precedente, per entrambe le classi del dataset.

Ogni metrica possiede un valore compreso tra 0% e 100%, tranne MCC che può assumere anche dei valori negativi per via della sua definizione. Maggiormente si avvicina a 100%, migliore è il risultato ottenuto. Per questo lavoro è stata realizzata un'analisi sulla base della metrica F-Score, la quale è molto utile per valutare la bontà di un classificatore in quanto prende in considerazione sia la precision che la recall, offrendo una statistica non dipendente da una classe in particolare.

Di seguito sono riportati i valori tabellari per ogni coppia, in verde sono riportati i valori di F-Score maggiori di 80%:

Features	Classifiers	Accuracy	Precision	Recall	Specificity	F-Score	MCC	BACC
	kNN	64.64%	71.97%	68.15%	59.24%	70.01%	27.07%	63.70%
	SVM	44.67%	96.29%	9.00%	99.47%	16.47%	17.91%	54.23%
Color histogram	Random Forest	73.26%	77.24%	79.19%	64.15%	78.20%	43.66%	71.67%
	Fine Tree	68.78%	74.49%	73.71%	61.22%	74.10%	34.84%	67.46%
	Ensemble	73.38%	76.15%	81.62%	60.72%	78.79%	43.37%	71.17%
	kNN	57.34%	70.04%	51.66%	66.06%	59.46%	17.42%	58.86%
	SVM	67.30%	69.18%	82.99%	43.20%	75.45%	28.71%	63.09%
Auto-correlogram	Random Forest	71.76%	72.89%	84.97%	51.47%	78.47%	39.09%	68.22%
	Fine Tree	62.78%	69.39%	68.97%	53.26%	69.18%	22.20%	61.12%
	Ensemble	73.86%	77.57%	79.96%	64.50%	78.75%	44.86%	72.23%
	kNN	42.17%	56.99%	18.40%	78.67%	27.82%	-3.61%	48.53%
	SVM	62.18%	62.38%	94.59%	12.38%	75.18%	12.45%	53.49%
Haar	Random Forest	62.48%	62.65%	94.22%	13.71%	75.26%	13.61%	53.97%
	Fine Tree	59.70%	62.56%	83.31%	23.43%	71.46%	8.33%	53.37%
	Ensemble	62.15%	62.75%	92.26%	15.89%	74.70%	12.75%	54.07%

Tabella 4.1: La tabella mostra le statistiche relative alle performance di ogni classificatore utilizzato, quando addestrato con le feature di colore considerate.

Come si evince dalla tabella 4.1, i risultati non sono stati molto soddisfacenti in confronto alle altre feature. Per l'istogramma del colore e l'auto-correlogram i migliori classificatori sono stati il Random Forest e l'Ensemble; per il primo il peggior risultato è stato ottenuto dalla SVM, per il secondo è stato ottenuto dal kNN. Per le feature di Haar la miglior classificazione è stata fatta da Random Forest e SVM; e la peggiore dal kNN, con un valore pari al 27%. Il Random Forest e l'Ensemble pare che beneficino delle feature del colore, anche se non in modo rilevante, come si vedrà in seguito.

Features	Classifiers	Accuracy	Precision	Recall	Specificity	F-Score	MCC	BACC
	kNN	69.51%	75.32%	73.86%	62.82%	74.58%	36.50%	68.34%
	SVM	62.64%	70.00%	67.06%	55.85%	68.50%	22.69%	61.46%
LBP18	Random Forest	79.57%	80.74%	87.03%	68.11%	83.77%	56.61%	77.57%
	Fine Tree	71.22%	76.23%	76.26%	63.47%	76.25%	39.74%	69.87%
	Ensemble	79.10%	81.37%	84.95%	70.13%	83.12%	55.82%	77.54%
	kNN	61.06%	68.53%	66.05%	53.41%	67.26%	19.29%	59.73%
	SVM	62.07%	73.64%	58.21%	68.00%	65.02%	25.64%	63.10%
GLCM	Random Forest	76.78%	78.69%	84.55%	64.84%	81.52%	50.63%	74.69%
	Fine Tree	68.78%	74.85%	72.99%	62.32%	73.91%	35.10%	67.66%
	Ensemble	75.91%	78.32%	83.26%	64.61%	80.72%	48.85%	73.93%

Tabella 4.2: La tabella mostra le statistiche relative alle performance di ogni classificatore utilizzato, quando addestrato con le feature di texture considerate.

Come si evince dalla tabella 4.2, per entrambe le feature i migliori classificatori sono stati il Random Forest e l'Ensemble, i quali hanno ottenuto un valore di F-Score maggiore dell'80%; e il peggior risultato è stato ottenuto dalla SVM, con valori inferiori al 70%. Il Random Forest e l'Ensemble, anche qui, giocano un ruolo importante, arrivando a buoni risultati.

Features	Classifiers	Accuracy	Precision	Recall	Specificity	F-Score	MCC	BACC
	kNN	57.14%	64.73%	64.24%	46.25%	64.48%	10.46%	55.24%
	SVM	62.78%	70.05%	67.34%	55.77%	68.66%	22.90%	61.55%
ZM66	Random Forest	65.93%	68.89%	79.76%	44.69%	73.93%	26.10%	62.22%
	Fine Tree	59.17%	66.50%	65.67%	49.18%	66.08%	14.81%	57.43%
	Ensemble	64.50%	67.93%	78.42%	43.12%	72.80%	22.96%	60.77%
	kNN	57.07%	64.74%	63.94%	46.51%	64.34%	10.42%	55.23%
	SVM	62.63%	62.73%	94.37%	13.87%	75.36%	14.15%	54.12%
ZM64	Random Forest	65.42%	68.54%	79.32%	44.08%	73.53%	24.97%	61.70%
	Fine Tree	58.49%	66.06%	64.73%	48.91%	65.39%	13.58%	56.82%
	Ensemble	64.25%	67.53%	78.92%	41.71%	72.78%	22.17%	60.32%
	kNN	57.28%	65.29%	62.92%	48.61%	64.08%	11.43%	55.77%
	SVM	64.47%	69.42%	73.91%	49.98%	71.59%	24.40%	61.94%

ZM62	Random Forest	65.90%	68.70%	80.28%	43.81%	74.04%	25.89%	62.05%
	Fine Tree	58.95%	66.05%	66.29%	47.66%	66.17%	13.97%	56.98%
	Ensemble	63.77%	67.20%	78.47%	41.18%	72.40%	21.11%	59.83%
	kNN	57.40%	65.03%	64.19%	46.97%	64.60%	11.12%	55.58%
	SVM	63.84%	69.78%	71.08%	52.72%	70.43%	23.93%	61.90%
ZM55	Random Forest	65.19%	68.16%	79.84%	42.70%	73.54%	24.26%	61.27%
	Fine Tree	58.06%	65.74%	64.24%	48.57%	64.98%	12.73%	56.40%
	Ensemble	64.22%	67.52%	78.84%	41.75%	72.75%	22.12%	60.30%
	kNN	58.00%	65.75%	63.99%	48.80%	64.86%	12.70%	56.39%
	SVM	64.43%	69.63%	73.19%	50.97%	71.37%	24.55%	62.08%
ZM53	Random Forest	64.92%	68.13%	79.07%	43.20%	73.19%	23.81%	61.13%
	Fine Tree	58.03%	65.62%	64.48%	48.11%	65.05%	12.54%	56.30%
	Ensemble	64.55%	67.73%	79.22%	42.02%	73.02%	22.83%	60.62%
	kNN	57.82%	65.36%	64.58%	47.43%	64.97%	11.98%	56.01%
	SVM	61.89%	61.74%	97.47%	7.24%	75.60%	11.24%	52.35%
ZM44	Random Forest	65.19%	68.53%	78.65%	44.53%	73.24%	24.61%	61.59%
	Fine Tree	58.00%	65.81%	63.79%	49.10%	64.79%	12.80%	56.45%
	Ensemble	64.52%	67.53%	79.76%	41.10%	73.14%	22.60%	60.43%
	kNN	56.62%	64.81%	62.08%	48.23%	63.42%	10.21%	55.15%
	SVM	61.17%	61.11%	98.71%	3.50%	75.49%	7.44%	51.11%
ZM42	Random Forest	64.14%	67.74%	77.90%	43.01%	72.47%	22.23%	60.46%
	Fine Tree	58.13%	65.46%	65.38%	47.01%	65.42%	12.38%	56.19%
	Ensemble	63.72%	67.05%	78.84%	40.50%	72.47%	20.88%	59.67%

Tabella 4.3: La tabella mostra le statistiche relative alle performance di ogni classificatore utilizzato, quando addestrato con i momenti di Zernike.

Come si evince dalla tabella 4.3, i classificatori Random Forest, Ensemble e SVM hanno ottenuto per tutte le feature in questione i risultati migliori, mentre il kNN quelli peggiori. Il rendimento non è stato così notevole, poiché tutti i valori di F-Score sono sotto il 76%.

Features	Classifiers	Accuracy	Precision	Recall	Specificity	F-Score	MCC	BACC
	kNN	65.92%	72.69%	70.04%	59.58%	71.34%	29.36%	64.81%
	SVM	73.01%	74.52%	84.23%	55.77%	79.08%	42.06%	70.00%
CH3	Random Forest	76.58%	79.25%	83.09%	66.59%	81.12%	50.43%	74.84%
	Fine Tree	67.87%	73.91%	72.57%	60.65%	73.23%	33.07%	66.61%
	Ensemble	76.33%	79.15%	82.69%	66.55%	80.88%	49.92%	74.62%

	kNN	65.22%	71.96%	69.77%	58.25%	70.85%	27.81%	64.01%
	SVM	73.43%	75.35%	83.41%	58.10%	79.18%	43.15%	70.75%
CH4	Random Forest	76.27%	78.60%	83.56%	65.07%	81.01%	49.62%	74.31%
	Fine Tree	68.42%	73.96%	73.88%	60.04%	73.92%	33.91%	66.96%
	Ensemble	76.96%	79.53%	83.43%	67.01%	81.43%	51.22%	75.22%
	kNN	66.16%	72.41%	71.28%	58.29%	71.84%	29.45%	64.78%
	SVM	75.92%	77.37%	85.14%	61.75%	81.07%	48.61%	73.45%
CH5	Random Forest	78.11%	79.99%	85.17%	67.28%	82.50%	53.56%	76.22%
	Fine Tree	71.11%	76.15%	76.17%	63.35%	76.16%	39.52%	69.76%
	Ensemble	79.07%	81.90%	84.03%	71.47%	82.95%	55.92%	77.75%
	kNN	64.98%	71.52%	70.11%	57.10%	70.81%	27.08%	63.61%
	SVM	76.42%	77.37%	86.31%	61.22%	81.59%	49.62%	73.76%
CH6	Random Forest	78.02%	79.71%	85.47%	66.59%	82.49%	53.32%	76.03%
	Fine Tree	71.37%	76.58%	75.97%	64.30%	76.27%	40.19%	70.14%
	Ensemble	79.18%	81.79%	84.42%	71.12%	83.09%	56.08%	77.77%

Tabella 4.4: La tabella mostra le statistiche relative alle performance di ogni classificatore utilizzato, quando addestrato con i momenti di Chebishev di prima specie.

Come si evince dalla tabella 4.4, i classificatori Random Forest, Ensemble e SVM hanno ottenuto per tutte le feature in questione i risultati migliori, mentre il kNN quelli peggiori. Queste feature hanno ottenuto dei valori di F-Score più soddisfacenti delle precedenti, dato che alcuni hanno superato l'80%. La coppia più proficua è stata CH6-Ensemble, con un valore di F-Score pari all'83%. Inoltre, in confronto alle altre feature handcrafted, l'SVM ha avuto il miglior rendimento.

Features	Classifiers	Accuracy	Precision	Recall	Specificity	F-Score	MCC	BACC
	kNN	65.57%	72.09%	70.41%	58.13%	71.24%	28.38%	64.27%
	SVM	70.98%	70.64%	89.14%	43.09%	78.82%	37.10%	66.11%
CHd3	Random Forest	76.40%	78.73%	83.63%	65.30%	81.11%	49.92%	74.46%
	Fine Tree	69.70%	75.27%	74.43%	62.44%	74.85%	36.76%	68.43%
	Ensemble	77.21%	79.76%	83.58%	67.43%	81.63%	51.77%	75.50%
	kNN	64.97%	71.82%	69.39%	58.17%	70.59%	27.34%	63.78%
	SVM	71.10%	70.79%	89.01%	43.58%	78.86%	37.38%	66.30%
CHd4	Random Forest	76.64%	78.37%	84.85%	64.04%	81.48%	50.28%	74.44%
	Fine Tree	69.18%	74.63%	74.40%	61.14%	74.52%	35.52%	67.77%
	Ensemble	78.16%	80.46%	84.45%	68.50%	82.41%	53.77%	76.47%

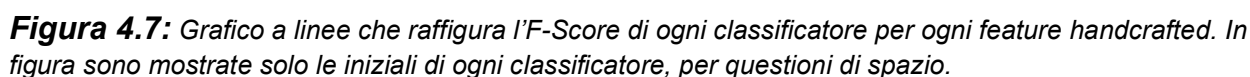
	kNN	65.46%	72.09%	70.14%	58.29%	71.10%	28.24%	64.21%
	SVM	72.50%	71.57%	90.55%	44.76%	79.95%	40.78%	67.66%
CHd5	Random Forest	78.07%	79.84%	85.34%	66.90%	82.50%	53.43%	76.12%
	Fine Tree	71.05%	76.12%	76.07%	63.35%	76.09%	39.41%	69.71%
	Ensemble	79.48%	81.57%	85.42%	70.36%	83.45%	56.60%	77.89%
	kNN	65.24%	71.84%	70.09%	57.79%	70.95%	27.71%	63.94%
	SVM	72.51%	71.57%	90.60%	44.72%	79.97%	40.82%	67.66%
CHd6	Random Forest	78.11%	79.66%	85.76%	66.36%	82.60%	53.48%	76.06%
	Fine Tree	70.47%	75.94%	75.00%	63.50%	75.47%	38.38%	69.25%
	Ensemble	80.14%	81.88%	86.31%	70.67%	84.04%	57.95%	78.49%

Tabella 4.5: La tabella mostra le statistiche relative alle performance di ogni classificatore utilizzato, quando addestrato con i momenti di Chebishev di seconda specie.

Come si evince dalla tabella 4.5, i classificatori Random Forest ed Ensemble hanno ottenuto per tutte le feature in questione i risultati migliori, mentre il kNN quelli peggiori. Queste feature hanno raggiunto i valori più elevati in assoluto tra tutti i momenti invarianti; quasi tutte le coppie superano l'80%. I polinomi di sesto grado, allenanti il classificatore Ensemble, hanno raggiunto il miglior risultato delle feature handcrafted, con un valore pari all'84%.

Features	Classificators	Accuracy	Precision	Recall	Specificity	F-Score	MCC	BACC
	kNN	65.96%	72.88%	69.77%	60.11%	71.29%	29.59%	64.94%
	SVM	71.76%	71.81%	87.87%	47.01%	79.03%	38.92%	67.44%
LM3	Random Forest	76.16%	78.70%	83.13%	65.45%	80.86%	49.46%	74.29%
	Fine Tree	68.39%	74.37%	72.97%	61.37%	73.66%	34.17%	67.17%
	Ensemble	76.84%	79.42%	83.36%	66.82%	81.34%	50.96%	75.09%
	kNN	65.54%	72.11%	70.29%	58.25%	71.19%	28.36%	64.27%
	SVM	71.73%	71.74%	87.97%	46.78%	79.03%	38.85%	67.38%
LM4	Random Forest	76.54%	78.76%	83.88%	65.26%	81.24%	50.18%	74.57%
	Fine Tree	69.07%	74.82%	73.76%	61.87%	74.29%	35.50%	67.81%
	Ensemble	77.12%	79.43%	83.98%	66.59%	81.64%	51.50%	75.28%
	kNN	66.32%	72.57%	71.38%	58.55%	71.97%	29.81%	64.97%
	SVM	73.32%	72.65%	89.73%	48.11%	80.29%	42.61%	68.92%
LM5	Random Forest	78.25%	80.20%	85.09%	67.73%	82.58%	53.87%	76.41%
	Fine Tree	71.32%	76.16%	76.64%	63.16%	76.40%	39.87%	69.90%
	Ensemble	78.61%	80.92%	84.65%	69.33%	82.74%	54.75%	76.99%
	kNN	64.85%	71.30%	70.24%	56.57%	70.76%	26.71%	63.40%
	SVM	73.29%	72.67%	89.61%	48.23%	80.25%	42.53%	68.92%
LM6	Random Forest	77.41%	79.37%	84.72%	66.17%	81.96%	52.03%	75.45%
	Fine Tree	70.81%	75.95%	75.82%	63.12%	75.88%	38.92%	69.47%
	Ensemble	79.38%	81.74%	84.92%	70.86%	83.30%	56.44%	77.89%

Come si evince dalla tabella 4.6, i classificatori Random Forest, Ensemble e SVM hanno ottenuto per tutte le feature in questione i risultati migliori, mentre il kNN quelli peggiori. I momenti di Legendre hanno ottenuto dei risultati notevoli, visto che i valori di F-Score partono dal 70% per arrivare a superare l'80%.



I valori di F-Score si possono ritenere elevati se superano il 90%, molto elevati se si avvicinano al 100%. Come possiamo notare dalle tabelle raffigurate, i valori di F-Score non raggiungono tale valore. Ad ogni modo, la coppia che ha prevalso è quella formata dai polinomi di Chebyshev di seconda specie di grado 6 e dal classificatore Ensemble, con un valore di F-Score lievemente superiore all'84%.

Features	Classificators	Accuracy	Precision	Recall	Specificity	F-Score	MCC	BACC
	Ensemble	40.56%	63.44%	4.39%	96.11%	8.21%	1.23%	50.25%
	Fine Tree	75.00%	79.47%	79.19%	68.57%	79.33%	47.72%	73.88%
AlexNet	kNN	80.85%	83.80%	84.77%	74.82%	84.28%	59.78%	79.80%
	Random Forest	84.02%	85.55%	88.57%	77.03%	87.03%	66.29%	82.80%

	SVM	51.43%	61.71%	52.21%	50.25%	56.56%	2.40%	51.23%
	Ensemble	45.13%	82.28%	11.98%	96.04%	20.91%	13.82%	54.01%
	Fine Tree	78.25%	82.04%	82.04%	72.42%	82.04%	54.46%	77.23%
DarkNet 19	kNN	83.99%	86.47%	87.20%	79.05%	86.84%	66.40%	83.13%
	Random Forest	88.30%	88.68%	92.49%	81.87%	90.54%	75.33%	87.18%
	SVM	67.58%	75.01%	69.69%	64.34%	72.26%	33.53%	67.02%
	Ensemble	39.73%	65.63%	1.04%	99.16%	2.05%	1.02%	50.10%
	Fine Tree	81.64%	84.78%	84.95%	76.57%	84.86%	61.55%	80.76%
DarkNet 53	kNN	88.25%	88.89%	92.11%	82.32%	90.48%	75.25%	87.22%
	Random Forest	90.30%	90.72%	93.55%	85.30%	92.11%	79.58%	89.42%
	SVM	90.93%	92.96%	91.99%	89.30%	92.47%	81.07%	90.64%
	Ensemble	40.14%	61.14%	3.20%	96.88%	6.08%	0.21%	50.04%
	Fine Tree	84.92%	87.51%	87.60%	80.80%	87.56%	68.42%	84.20%
DenseNet 201	kNN	88.21%	89.04%	91.84%	82.63%	90.42%	75.16%	87.23%
	Random Forest	91.93%	92.61%	94.20%	88.46%	93.40%	83.05%	91.33%
	SVM	94.43%	95.41%	95.39%	92.95%	95.40%	88.33%	94.17%
	Ensemble	41.87%	84.62%	4.91%	98.63%	9.28%	9.39%	51.77%
	Fine Tree	78.79%	82.62%	82.29%	73.41%	82.46%	55.64%	77.85%
EfficientNet-b0	kNN	87.53%	89.01%	90.60%	82.82%	89.80%	73.79%	86.71%
	Random Forest	89.89%	89.96%	93.77%	83.92%	91.83%	78.71%	88.85%
	SVM	93.84%	94.69%	95.16%	91.81%	94.93%	87.09%	93.49%
	Ensemble	43.44%	78.96%	9.03%	96.30%	16.20%	10.26%	52.67%
	Fine Tree	73.52%	78.02%	78.35%	66.10%	78.18%	44.49%	72.22%
GoogleNet	kNN	76.13%	79.89%	80.98%	68.69%	80.43%	49.85%	74.83%
	Random Forest	83.25%	84.10%	89.21%	74.10%	86.58%	64.55%	81.65%
	SVM	52.29%	63.04%	51.31%	53.79%	56.58%	4.99%	52.55%
	Ensemble	41.93%	84.87%	5.01%	98.63%	9.46%	9.58%	51.82%
	Fine Tree	80.29%	84.15%	83.11%	75.96%	83.63%	58.88%	79.54%
Inception-ResNet-v2	kNN	85.40%	86.62%	89.76%	78.70%	88.16%	69.21%	84.23%
	Random Forest	89.08%	89.19%	93.28%	82.63%	91.19%	76.99%	87.95%
	SVM	93.59%	94.28%	95.19%	91.12%	94.73%	86.54%	93.16%
	Ensemble	45.95%	81.27%	13.99%	95.05%	23.87%	14.45%	54.52%
	Fine Tree	74.54%	79.21%	78.60%	68.30%	78.90%	46.81%	73.45%
Inception-v3	kNN	80.88%	83.65%	85.04%	74.48%	84.34%	59.80%	79.76%
	Random Forest	85.52%	85.64%	91.42%	76.46%	88.44%	69.39%	83.94%
	SVM	82.71%	86.61%	84.52%	79.92%	85.55%	64.06%	82.22%
	Ensemble	40.98%	82.80%	3.22%	98.97%	6.21%	7.07%	51.10%

	Fine Tree	80.13%	83.67%	83.48%	74.97%	83.58%	58.42%	79.23%
ResNet 101	kNN	87.89%	88.57%	91.87%	81.79%	90.19%	74.48%	86.83%
	Random Forest	89.59%	89.76%	93.48%	83.62%	91.58%	78.07%	88.55%
	SVM	92.10%	93.74%	93.18%	90.44%	93.46%	83.49%	91.81%
	Ensemble	40.15%	80.00%	1.59%	99.39%	3.11%	4.39%	50.49%
	Fine Tree	76.73%	81.20%	80.13%	71.50%	80.66%	51.46%	75.82%
ResNet 18	kNN	84.21%	85.89%	88.47%	77.68%	87.16%	66.73%	83.07%
	Random Forest	86.99%	87.32%	91.87%	79.50%	89.53%	72.54%	85.68%
	SVM	87.32%	89.22%	89.93%	83.31%	89.58%	73.40%	86.62%
	Ensemble	40.05%	75.31%	1.51%	99.24%	2.97%	3.35%	50.38%
	Fine Tree	81.30%	84.62%	84.47%	76.42%	84.55%	60.87%	80.45%
ResNet 50	kNN	87.34%	88.12%	91.42%	81.07%	89.74%	73.30%	86.24%
	Random Forest	89.92%	90.12%	93.63%	84.23%	91.84%	78.77%	88.93%
	SVM	93.01%	93.96%	94.54%	90.67%	94.25%	85.35%	92.61%
	Ensemble	39.90%	73.13%	1.22%	99.31%	2.39%	2.59%	50.26%
	Fine Tree	76.40%	80.97%	79.79%	71.20%	80.37%	50.80%	75.49%
VGG	kNN	81.01%	83.54%	85.49%	74.13%	84.51%	60.03%	79.81%
	Random Forest	85.98%	86.61%	90.92%	78.40%	88.71%	70.41%	84.66%
	SVM	52.67%	63.00%	52.95%	52.23%	57.54%	5.06%	52.59%
	Ensemble	39.70%	58.65%	1.51%	98.36%	2.95%	-0.49%	49.94%
	Fine Tree	79.38%	83.30%	82.49%	74.59%	82.89%	56.94%	78.54%
Xception	kNN	85.80%	88.05%	88.59%	81.52%	88.32%	70.23%	85.06%
	Random Forest	88.58%	89.08%	92.49%	82.59%	90.75%	75.94%	87.54%
	SVM	91.96%	93.14%	93.63%	89.41%	93.38%	83.15%	91.52%

Tabella 4.8: La tabella mostra le statistiche relative alle performance di ogni classificatore utilizzato, quando addestrato con le feature deep considerate. In verde sono evidenziati i valori di F-Score maggiori del 90%.

Come si evince dalla tabella 4.8, le reti neurali AlexNet, GoogleNet, Inception-v3, ResNet 18 e VGG hanno ottenuto risultati poco considerevoli, avendo ottenuto valori inferiori al 90%. Le altre, invece, hanno ottenuto risultati più soddisfacenti; una particolare menzione va a DenseNet 201 ed EfficientNet-b0, le quali hanno avuto il rendimento migliore.

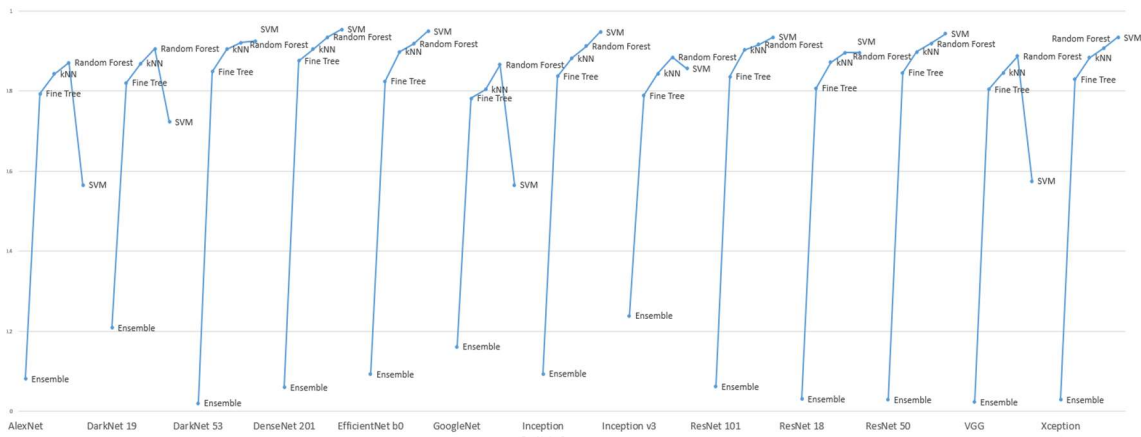


Figura 4.9: Grafico a linee che raffigura l’F-Score di ogni classificatore per ogni feature deep.

Notiamo che l’Ensemble non ha replicato gli eccellenti risultati ottenuti con le feature handcrafted ma gli altri classificatori hanno compensato in modo notevole, superando notevolmente i valori di F-Score delle feature handcrafted precedentemente analizzate.

I valori di F-Score che hanno prevalso, in questo caso, sono quelli delle reti neurali DenseNet 201 ed EfficientNet-b0. La prima, insieme al classificatore SVM, ha superato il 95%; la seconda, ugualmente insieme al classificatore SVM, è lievemente inferiore, con un valore pari a 94.9%.

4.2 Combinazione tra feature

Nonostante i considerevoli risultati ottenuti dalla classificazione descritta precedentemente, si è tentato comunque di migliorare le prestazioni. Si è svolto, quindi, un ulteriore esperimento al fine di verificare la possibilità di combinare il potere rappresentativo evidenziato dalle feature handcrafted e deep.

In particolare, sono state prese in considerazione la feature handcrafted e le due feature deep col rendimento più elevato. A questo proposito, abbiamo osservato che le feature in questione sono: i momenti di Chebishev di seconda specie di sesto grado, DenseNet 201 ed EfficientNet-b0.

Perciò, sono stati eseguiti i seguenti passaggi:

- Sono state effettuate delle combinazioni fra feature, mediante delle concatenazioni tra matrici.
- Sono state combinate tutte le 4 possibili coppie fra le tre feature sopracitate.
- Sono stati allenati nuovamente tre classificatori: kNN, Random Forest e SVM.
- Sono state generate le confusion matrix e sono state estratte le corrispettive statistiche.
- Sono stati raccolti i risultati.

I risultati ottenuti per il dataset studiato sono i seguenti:

Features	Classifiers	Accuracy	Precision	Recall	Specificity	F-Score	MCC	BACC
	kNN	88.21%	89.04%	91.84%	82.63%	90.42%	75.16%	87.23%
CHd6 + DenseNet 201	Random Forest	92.16%	92.83%	94.35%	88.80%	93.58%	83.53%	91.57%
	SVM	94.41%	95.14%	95.66%	92.50%	95.40%	88.29%	94.08%

	kNN	87.55%	89.01%	90.63%	82.82%	89.81%	73.82%	86.72%
CHd6 + EfficientNet-b0	Random Forest	89.65%	90.19%	93.03%	84.46%	91.59%	78.21%	88.74%
	SVM	94.05%	94.78%	95.44%	91.92%	95.11%	87.53%	93.68%
	kNN	90.30%	91.05%	93.13%	85.94%	92.08%	79.59%	89.54%
DenseNet 201 + EfficientNet-b0	Random Forest	91.83%	92.31%	94.37%	87.92%	93.33%	82.82%	91.15%
	SVM	94.89%	95.76%	95.81%	93.49%	95.78%	89.31%	94.65%
	kNN	90.31%	91.07%	93.13%	85.98%	92.09%	79.63%	89.56%
CHd6 + DenseNet 201 + EfficientNet-b0	Random Forest	92.26%	92.67%	94.72%	88.50%	93.68%	83.74%	91.61%
	SVM	95.03%	95.86%	95.93%	93.64%	95.90%	89.59%	94.79%

Tabella 4.10: La tabella mostra le statistiche relative alle performance di ogni classificatore utilizzato, quando addestrato con le feature combinate considerate. In verde sono evidenziati i valori di F-Score maggiori del 90%.

Si può evincere dalla tabella 4.10 come i valori di F-Score siano notevolmente più alti di quelli ottenuti in precedenza, così come quelli delle altre statistiche. I valori di F-Score superano quasi tutti il 90%.

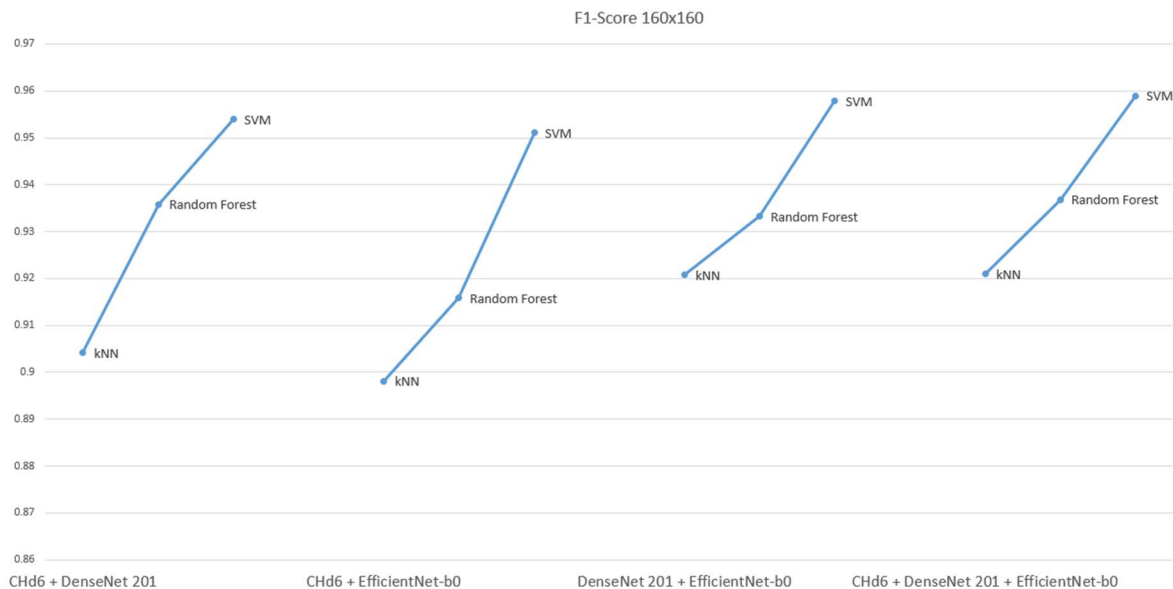


Figura 4.11: Grafico a linee che raffigura l'F-Score di ogni classificatore per ogni feature combinata.

Il classificatore che ha avuto la meglio, per ogni coppia feature-classificatore, è la SVM, con valori di F-Score che sfiorano il 96%. Il rendimento è stato ulteriormente migliorato, come da obiettivo prefissato.

4.3 Sperimentazione cross-dataset

Una volta raggiunto l'obiettivo di ottimizzare il rendimento della classificazione di immagini 160x160, si è tentato di utilizzare le stesse metodologie per classificare immagini istopatologiche di due dataset di risoluzione differente: uno contenente immagini 120x120 e uno contenente immagini 80x80.

Per la precisione, si sono utilizzati gli stessi modelli addestrati sulle immagini 160x160, costituiti dalle feature combinate e dagli stessi tre classificatori, per effettuare delle predizioni sugli altri due dataset.

Nel caso del dataset formato da immagini 120x120, sono stati ottenuti i seguenti valori:

Features	Classificators	Accuracy	Precision	Recall	Specificity	F-Score	MCC	BACC
	kNN	86.69%	90.56%	87.68%	85.08%	89.09%	72.10%	86.38%
CHd6 + DenseNet 201	Random Forest	89.04%	95.12%	86.78%	92.74%	90.76%	77.87%	89.76%
	SVM	86.41%	97.42%	80.20%	96.53%	87.98%	74.51%	88.37%
	kNN	85.17%	88.66%	87.25%	81.79%	87.95%	68.71%	84.52%
CHd6 + EfficientNet-b0	Random Forest	87.38%	90.20%	89.36%	84.15%	89.78%	73.30%	86.76%
	SVM	85.02%	96.81%	78.42%	95.79%	86.65%	72.04%	87.10%
	kNN	88.43%	91.65%	89.50%	86.69%	90.56%	75.66%	88.09%
DenseNet 201 + EfficientNet-b0	Random Forest	89.55%	94.36%	88.43%	91.37%	91.30%	78.51%	89.90%
	SVM	85.88%	98.40%	78.50%	97.92%	87.33%	74.19%	88.21%
	kNN	88.42%	91.64%	89.50%	86.67%	90.55%	75.65%	88.08%
CHd6 + DenseNet 201 + EfficientNet-b0	Random Forest	89.56%	94.79%	87.99%	92.12%	91.26%	78.67%	90.05%
	SVM	85.82%	98.45%	78.36%	97.98%	87.26%	74.12%	88.17%

Tabella 4.12: La tabella mostra le statistiche relative alle performance di ogni classificatore utilizzato, quando addestrato con le feature combinate considerate per immagini 120x120. In verde sono evidenziati i valori di F-Score maggiori del 90%.

Dalla tabella 4.12 si può evincere che anche per questo dataset, i valori siano molto soddisfacenti. Essi, difatti, sfiorano quasi tutti il 90%.

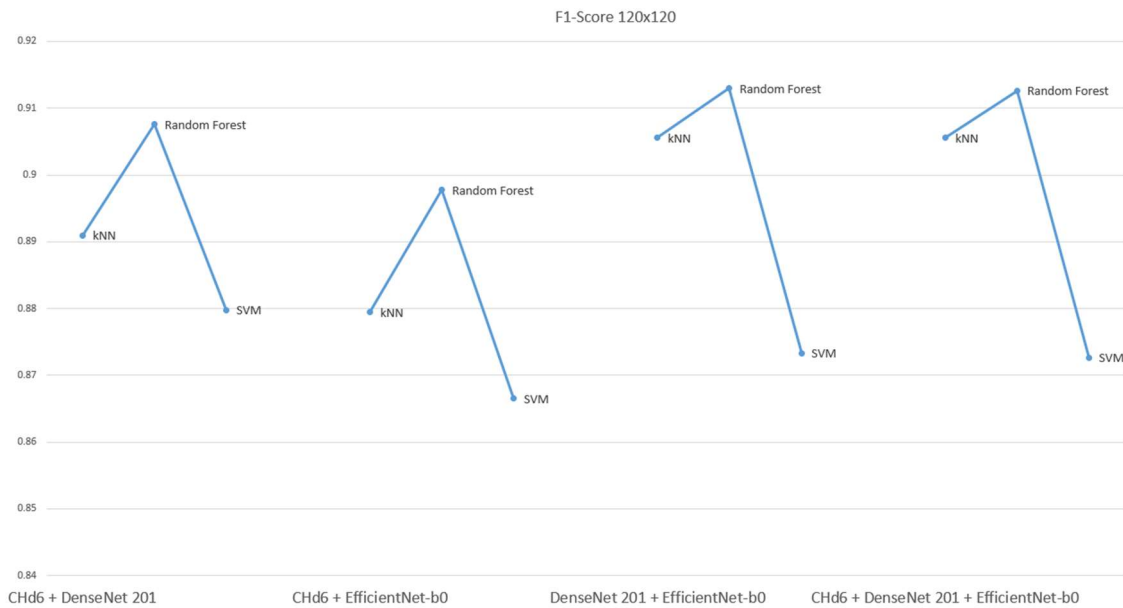


Figura 4.13: Grafico a linee che raffigura l'F-Score di ogni classificatore per ogni feature combinata di immagini 120x120.

Features	Classifiers	Accuracy	Precision	Recall	Specificity	F-Score	MCC	BACC
CHd6 + DenseNet 201	kNN	77.05%	86.23%	73.23%	82.70%	79.20%	54.89%	77.97%
	Random Forest	78.89%	93.29%	69.62%	92.60%	79.73%	61.41%	81.11%
	SVM	68.92%	96.18%	49.89%	97.07%	65.70%	49.83%	73.48%
CHd6 + EfficientNet-b0	kNN	63.58%	89.19%	44.34%	92.05%	59.23%	39.09%	68.20%
	Random Forest	71.38%	92.27%	56.79%	92.96%	70.31%	50.63%	74.88%
	SVM	54.36%	96.04%	24.53%	98.50%	39.07%	31.44%	61.51%
DenseNet 201 + EfficientNet-b0	kNN	74.70%	88.69%	66.02%	87.55%	75.69%	52.89%	76.78%
	Random Forest	79.73%	92.84%	71.54%	91.84%	80.81%	62.39%	81.69%
	SVM	59.96%	96.81%	34.02%	98.34%	50.34%	39.00%	66.18%
CHd6 + DenseNet 201 + EfficientNet-b0	kNN	74.70%	88.69%	66.02%	87.55%	75.69%	52.89%	76.78%
	Random Forest	78.44%	93.88%	68.33%	93.41%	79.09%	61.10%	80.87%
	SVM	60.31%	96.72%	34.66%	98.26%	51.03%	39.39%	66.46%

Tabella 4.14: La tabella mostra le statistiche relative alle performance di ogni classificatore utilizzato, quando addestrato con le feature combinate considerate per immagini 80x80. In verde sono evidenziati i valori di F-Score maggiori dell'80%.

I valori di F-Score del dataset 80x80 sono lievemente più bassi degli altri. La quantità di immagini era ingente; inoltre, la difficoltà delle immagini rappresentate è aumentata, rendendo la classificazione con i modelli addestrati più complessa.

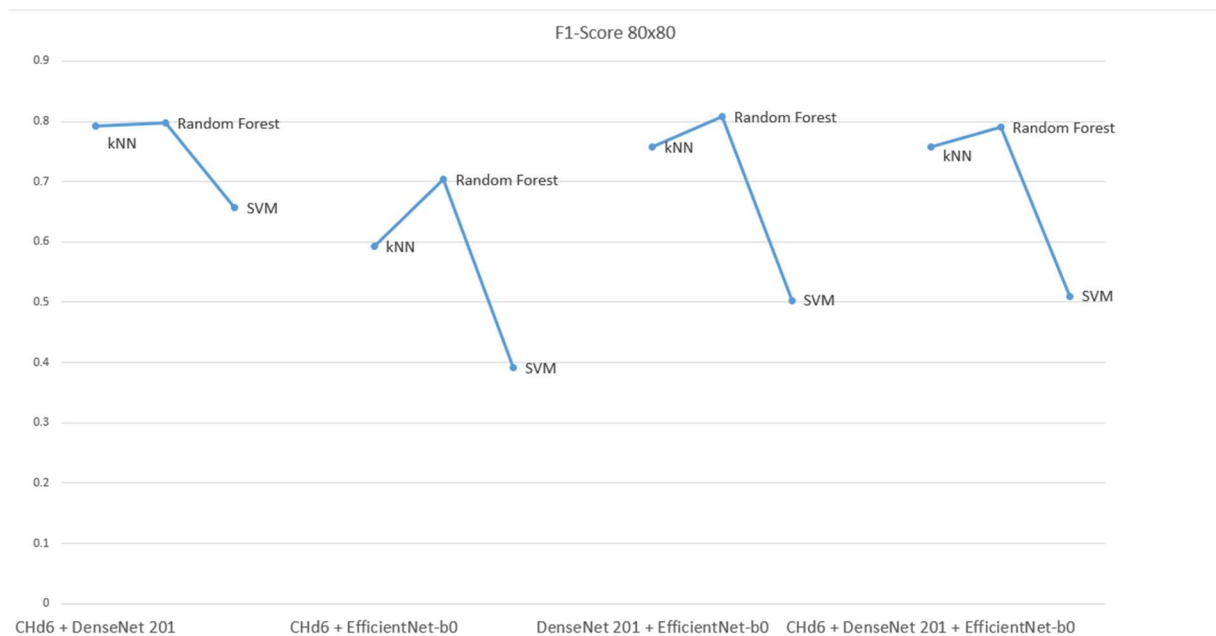


Figura 4.15: Grafico a linee che raffigura l’F-Score di ogni classificatore per ogni feature combinata di immagini 80x80.

Per ambedue i dataset, il classificatore che ha svolto al meglio il suo lavoro è stata il Random Forest.

4.4 Discussione

Dai risultati ottenuti, si può evincere che le feature handcrafted abbiano avuto il rendimento più scarso nel completare il task prefissato. In particolare, le feature di colore hanno permesso di ottenere soltanto valori di F-Score minori dell’80%. Addirittura, nel color histogram è possibile scorgere il valore di F-Score più basso in assoluto, utilizzato per l’allenamento del classificatore SVM, pari al 16%.

Anche i momenti di Zernike non hanno portato ad un ottimo risultato, anch’essi con valori inferiori all’80%. Le performance migliorano con le feature di texture, coi momenti di Legendre e coi momenti di Chebishev di prima e seconda specie, dove tale soglia viene superata, arrivando anche a superare l’80%.

Per questo genere di feature, i classificatori più proficui sono stati il Random Forest, l’Ensemble e, per tutti i momenti invarianti, l’SVM. Fine Tree e kNN hanno, invece, ottenuto sempre risultati inferiori all’80%.

L’efficienza dell’Ensemble non si ripresenta, tuttavia, nella classificazione delle feature deep. I valori di F-Score ottenuti con tale classificatore sono tutti inferiori al 10%, ad eccezione di GoogleNet e Inception-v3, in cui è lievemente maggiore. Difatti, questo è il classificatore che ha avuto il rendimento peggiore per questo genere di feature. Al contrario, Random Forest e SVM, sono stati i più produttivi, menzione speciale anche a kNN e Fine Tree, che hanno visto netti miglioramenti rispetto alle loro prestazioni per le feature handcrafted. Per il resto, i valori superano anche il 90%, notevole aumento rispetto alle precedenti.

Una volta che le feature più efficienti sono state combinate fra loro, si sono ottenuti valori ancora più elevati. Tutti i valori di F-Score superano il 90%, tranne il CH2 + EfficientNet-b0 con il classificatore kNN, che è lievemente inferiore. Il raffinamento che si intendeva fare è stato eseguito con successo.

Per quanto concerne la sperimentazione cross-dataset, si sono ottenuti discreti valori di F-Score per il dataset con immagini 120x120, compresi tra l'86% e il 92%; mentre per quello con immagini 80x80, l'SVM non ha riscontrato un grande successo; infatti, si notano percentuali di piuttosto mediocri: 65%, 50% o addirittura 39%. Differentemente, i rimanenti due classificatori sono riusciti a sfiorare l'80% in quest'ultimo dataset.

I metodi che hanno portato ai risultati più soddisfacenti in assoluto, a livello di feature, sono la tripla combinazione CHd6 + DenseNet 201 + EfficientNet-b0 e la coppia DenseNet 201 + EfficientNet-b0. Il titolo di miglior classificatore, invece, spetta al Random Forest.

Si propone, quindi, un approccio basato su Random Forest addestrato mediante le seguenti feature, per avere una rappresentazione fedele dei problemi presi in esame e ottenere performance di alto livello:

- per il primo task (classificazione di immagini 160x160):
 - o le feature handcrafted CH, CH2 e LM, avendo ottenuto sempre valori maggiori dell'80%;
 - o le feature deep DarkNet 53, DenseNet 201, EfficientNet-b0 ed Inception-ResNet-v2, avendo portato a valori maggiori del 90%;
 - o le feature combinate ottenute mediante la concatenazione tra i momenti CH2 e le due migliori feature deep DenseNet 201 ed EfficientNet-b0, per ottenere prestazioni ancora più elevate;
- per il secondo task (sperimentazione cross-dataset): la combinazione tra DenseNet 201 ed EfficientNet-b0, avendo ottenuto un rendimento maggiore dell'80% per le immagini 80x80 e maggiore del 90% per immagini 120x120.

Capitolo 5

CONCLUSIONE

Questa tesi aveva come fine quello di ottimizzare l'accuratezza, la precisione e altre caratteristiche della classificazione di immagini istopatologiche del dataset GasHisSDB. Rispetto alla ricerca effettuata dagli autori dell'articolo [1] che è stato preso come riferimento e che ha fornito il dataset oggetto dello studio, i risultati della classificazione sono complessivamente migliori. Nello specifico, si evidenzia come, senza l'utilizzo di classificatori appartenenti alla categoria Deep Learning, sia stato possibile progettare e realizzare un sistema che, mediante training su una risoluzione di immagini, sia in grado di ottenere risultati del tutto comparabili con quelli ottenuti dagli autori, sebbene con l'esclusivo utilizzo di tecniche deep. A titolo di esempio, attraverso il nostro metodo, è possibile classificare immagini di risoluzione 120x120 e 80x80 con un'accuratezza dell'80% e del 90%, rispettivamente, utilizzando modelli addestrati su immagini di risoluzione 160x160. In quest'ultimo caso, tali modelli hanno raggiunto un'accuratezza massima del 95%, superando di 10 punti percentuali il miglior risultato ottenuto dagli autori nel lavoro esaminato.

Si può notare, dai paragrafi precedenti, come i valori di F-Score siano notevoli anche per i dataset di immagini 120x120 e 80x80, il che significa che i modelli utilizzati per quello con immagini 160x160 siano consistenti ed efficienti anche per immagini istopatologiche di dimensioni differenti.

La classificazione con l'Ensemble nelle feature deep, purtroppo, non ha ripetuto le performance dei risultati con feature handcrafted, pur avendo precedentemente notato come in queste ultime sia stato parecchio remunerativo.

Tuttavia, a livello generale e volendo trarre un'indicazione complessiva, la classificazione ha ottenuto risultati migliori tramite la combinazione tra le reti neurali DenseNet 201 e EfficientNet-b0 con l'utilizzo del classificatore Random Forest. Per quanto concerne le immagini 160x160 l'ulteriore combinazione con le feature handcrafted dei polinomi di Chebyshev di seconda specie di sesto grado ha ancora ottimizzato il rendimento, raggiungendo quasi il 96% di F-Score e di Accuracy, con l'utilizzo del classificatore SVM.

I modelli utilizzati per questo esperimento sono stati notevolmente ottimizzati.

Si può dire, perciò, che l'obiettivo è stato raggiunto con successo.

Nonostante i risultati fin qui ottenuti, i lavori futuri che possono essere realizzati a partire da questo sono molteplici. In particolare, si estenderà la classificazione su ulteriori feature e classificatori, andando a includere Vision Transformer di nuova generazione. Ulteriori sviluppi riguardanti la problematica cross-dataset, ugualmente, saranno realizzati per mezzo di test e eventuali raffinamenti su nuovi dataset e differenti tipologie di colorazione.

RIFERIMENTI BIBLIOGRAFICI

- [1] A Comparative Study of Gastric Histopathology Sub-size Image Classification: from Linear Regression to Visual Transformer. <https://arxiv.org/abs/2205.12843>, 2022
- [2] An Application of Transfer Learning an Ensemble Learning Techniques for Cervical Histopathology Image Classification. <https://ieeexplore.ieee.org/document/9107128>, 2019
- [3] GasHisSDB: A new gastric histopathology image dataset for computer aided diagnosis of gastric cancer. <https://arxiv.org/abs/2106.02473>, 2021
- [4] Local Binary Pattern. <http://www.ce.unipr.it/~medici/geometry/node134.html>
- [5] Feature di Haar. <http://www.ce.unipr.it/~medici/geometry/node129.html>
- [6] Help Center MathWorks. VGG 19. <https://it.mathworks.com/help/deeplearning/ref/vgg19.html>.
- [7] GLCM Texture Features. https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_glcml.html
- [8] Binary Shape Clustering via Zernike Moments by Stephen Yoo and Michael Vorobyov
- [9] Chebyshev moments as a new set of moments for image reconstruction. <https://ieeexplore.ieee.org/document/938829>
- [10] Help Center MathWorks. AlexNet. <https://it.mathworks.com/help/deeplearning/ref/alexnet.html>.
- [11] Help Center MathWorks. Inceptionresnetv2. <https://it.mathworks.com/help/deeplearning/ref/inceptionresnetv2.html>.
- [12] Help Center MathWorks. Inceptionv3. <https://it.mathworks.com/help/deeplearning/ref/inceptionv3.html>.
- [13] Help Center MathWorks. Xception. <https://it.mathworks.com/help/deeplearning/ref/xception.html>.
- [14] Support-Vector Machine. <https://www.developersmaggioli.it/blog/support-vector-machine/>
- [15] Random Forest. <https://www.ibm.com/cloud/learn/random-forest>
- [16] Help Center MathWorks. GoogleNet. <https://it.mathworks.com/help/deeplearning/ref/googlenet.html>.
- [17] Help Center MathWorks. Efficientnetb0. <https://it.mathworks.com/help/deeplearning/ref/efficientnetb0.html>.
- [18] Help Center MathWorks. Densenet201. <https://it.mathworks.com/help/deeplearning/ref/densenet201.html>.
- [19] Stonemeadow Photography. <https://www.youtube.com/watch?v=fS43iwDJmk>
- [20] Matrice di confusione. <https://www.andreaminini.com/ai/machine-learning/matrice-di-confusione>

- [21] Rekhil M Kumar et al, «A Survey on Image Feature Descriptors,» International Journal of Computer Science and Information Technologies, vol. 5, n. 6, pp. 7668-7673, 2014.
- [22] Cos'è il Machine Learning? <https://www.oracle.com/it/artificial-intelligence/machine-learning/what-is-machine-learning/>
- [23] Ensemble Modeling: come migliorare il Machine Learning <https://blog.alleanzia.com/it/ensemble-modeling-come-migliorare-il-machine-learning>
- [24] Caratteristiche degli alberi di decisione (decision tree) in informatica. <https://vitolavecchia.altervista.org/caratteristiche-degli-alberi-di-decisione-decision-tree-in-informatica/>
- [25] Perché le reti neurali sono importanti? <https://it.mathworks.com/discovery/neural-network.html>
- [26] Cosa rende le CNN così utili? <https://it.mathworks.com/discovery/convolutional-neural-network-matlab.html>
- [27] Balanced Accuracy. When Should You Use It? <https://neptune.ai/blog/balanced-accuracy>
- [28] Introduzione alle reti neurali – pesi, bias e attivazione <https://www.diariodiunanalista.it/posts/introduzione-alle-reti-neurali-pesi-bias-e-attivazione>
- [29] Rethinking the Inception Architecture for Computer Vision. <https://arxiv.org/abs/1512.00567>, 2015
- [30] A simple interpretations of Zernike Polynomials. <https://www.quickguide.org/post/a-simple-interpretation-of-zernike-polynomials>
- [31] Commissione Internazionale per l'illuminazione. https://it.wikipedia.org/wiki/Commissione_internazionale_per_l'illuminazione

Elenco delle figure

2.1 Esempio di rete neurale convoluzionale.....	5
3.1 Esempio di immagine istopatologica normale.....	6
3.2 Esempio di immagine istopatologica anormale.....	6
3.3 Esempio di Local Binary Pattern.....	7
3.4 Polinomi di Zernike.....	8
3.5 Colori RGB delle immagini digitali all'occhio umano.....	10
3.6 Esempio di istogramma del colore.....	11
3.7 Istogramma del colore applicato a un'immagine istopatologica anormale.....	11
3.8 Esempio di feature di Haar.....	12
3.9 Esempio di codice Matlab per allenamento classificatore.....	15
3.10 Esempio di Support Vector Machine.....	17
3.11 Struttura di una matrice di confusione.....	18
4.7 Grafico a linee per F-Score di ogni classificatore-feature handcrafted.....	25
4.9 Grafico a linee per F-Score di ogni classificatore-feature deep.....	28
4.11 Grafico a linee per F-Score di ogni classificatore-feature combinata per immagini 160x160.....	29
4.13 Grafico a linee per F-Score di ogni classificatore-feature combinata per immagini 120x120.....	31
4.15 Grafico a linee per F-Score di ogni classificatore-feature combinata per immagini 80x80.....	32

Elenco delle tabelle

4.1 Valori ottenuti per ogni coppia di feature handcrafted del colore e classificatore.....	20
4.2 Valori ottenuti per ogni coppia di feature handcrafted della texture e classificatore.....	21
4.3 Valori ottenuti per ogni coppia di feature handcrafted dei momenti ZM e classificatore.....	22
4.4 Valori ottenuti per ogni coppia di feature handcrafted dei momenti di CH e classificatore.....	23
4.5 Valori ottenuti per ogni coppia di feature handcrafted CH2 e classificatore.....	24
4.6 Valori ottenuti per ogni coppia di feature handcrafted LM e classificatore.....	24
4.8 Valori ottenuti per ogni coppia di feature deep e classificatore.....	27
4.10 Valori ottenuti per ogni coppia di feature combinata e classificatore per dataset 160x160.....	29
4.12 Valori ottenuti per ogni coppia di feature combinata e classificatore per dataset 120x120.....	30
4.14 Valori ottenuti per ogni coppia di feature combinata e classificatore per dataset 80x80.....	31

Ringraziamenti

I miei più sentiti ringraziamenti vanno alla professoressa Cecilia Di Ruberto e al Dottor Andrea Loddo, che hanno accettato di essere i miei relatori per questa tesi; e per la loro grande disponibilità, cordialità, professionalità e il loro grande supporto.

Menzioni speciali vanno alla mia famiglia per essermi sempre stata vicino nei momenti di difficoltà ed avermi spronato ad andare avanti; e al mio collega Mirco per aver studiato ed aver affrontato tanti ostacoli nell'ambito universitario assieme.

Ringrazio tutti i miei amici che mi sono stati accanto per avermi dato la possibilità di svagarmi nei momenti di pausa.

Ringrazio infine tutti i professori del corso di laurea in informatica di Unica per avermi insegnato tantissime cose di cui farò sicuramente tesoro negli anni a venire.

Auguro a tutti loro di realizzarsi nella vita e ottenere tante soddisfazioni, così come io ne ho ottenuto una oggi, molto importante.