POLITECNICO DI TORINO

Collegio di Ingegneria Gestionale

Tesi di Laurea Magistrale in Ingegneria Gestionale

Progettazione e sviluppo di una metodologia data-driven per la classificazione di immagini.
Caso di studio: immagini istopatologiche del colon

Relatori

Prof.ssa Tania Cerquitelli (Politecnico di Torino)

Dr. Cheng-Bang Chen (University of Miami)

Candidato

Giuseppe Sciacca

Anno Accademico 2021/2022

Ringraziamenti

Prima di procedere con la trattazione, vorrei spendere due parole per ringraziare tutte le persone che mi sono state vicine e che mi hanno supportato durante questo percorso accademico durato cinque anni.

Grazie alla mia famiglia, che mi è sempre stata vicino e che mi ha aiutato sia moralmente che economicamente.

Grazie a tutti i miei fantastici colleghi del Politecnico di Torino, studiare insieme a voi ha reso più leggeri e divertenti anche i momenti più stressanti.

Grazie a tutte le persone che ho conosciuto durante il mio Erasmus a Parigi, insieme abbiamo trascorso dei momenti indimenticabili e abbiamo vissuto un'esperienza che ha cambiato per sempre le nostre vite e ampliato i nostri orizzonti.

Grazie alla professoressa Tania Cerquitelli che ha accettato fin da subito di seguirmi durante questo progetto di tesi all'estero e di essere la mia relatrice.

Grazie alla University of Miami e al professor Cheng-Bang Chen, che mi hanno accolto per svolgere il mio progetto di ricerca senza chiedere nulla in cambio, e grazie a tutti i colleghi che ho conosciuto durante la mia permanenza a Miami, mi avete fatto sentire a casa anche dall'altra parte del mondo.

Un grande grazie allo Stato Italiano e al Politecnico di Torino, che mi hanno permesso di ricevere un'educazione eccellente in maniera quasi del tutto gratuita.

Sommario

Figure

Capitolo 1. Introduzione

Questo lavoro di tesi è stato svolto in collaborazione con il dipartimento di Industrial Engineering della University of Miami in Florida, negli Stati Uniti. Per questo motivo molti dei dati sono riferiti alla situazione americana. L'obiettivo principale di questo lavoro di tesi è stato quello di sviluppare e sperimentare un nuovo metodo di machine learning per la classificazione di immagini mediche basato sull'applicazione della teoria dei grafi e dell'analisi delle ricorrenze. Il software, sviluppato in ambiente Matlab, può essere una base per il successivo sviluppo di sistemi CAD per supportare i medici nella diagnosi di diversi tipi di tumori e di altre patologie, ovviamente dopo la relativa fase di training.

Con il termine imaging biomedico, o diagnostica per immagini, si intende l'insieme di tecniche e procedure con le quali è possibile conoscere, esplorare, esaminare e monitorare aree del corpo umano (o parti di organi e/o tessuti) non visibili ad occhio nudo dall'esterno. L'output di questi processi è un'immagine, reale o ricostruita, di una parte del corpo. Ad oggi sono moltissime le tecniche di imaging utilizzate in medicina: radiografia a raggi X, risonanza magnetica, ultrasuoni, endoscopia, elastografia, termografia e tecniche di imaging nucleare come la tomografia a emissione di positroni (PET) o la SPECT. L'istologia, invece, è la branca della biologia che studia i tessuti animali e vegetali, anche conosciuta come anatomia microscopica o microanatomia. L'istopatologia, o istologia medica, è la branca dell'istologia che si occupa di studiare, individuare e identificare delle patologie attraverso lo studio di tessuti malati. Gli esami istolopatologici sono uno strumento di fondamentale importanza nell'individuazione e diagnosi di tumori e di molte altre patologie. Lo strumento fondamentale per analizzare un tessuto biologico è il microscopio, che riesce a fornire un'immagine fortemente ingrandita del tessuto. Prima di essere passati al microscopio, i tessuti devono essere trattati e preparati con delle tecniche istologiche. Alcune delle tecniche istologiche più utili e diffuse sono la fissazione, per prevenire la decomposizione del tessuto, l'inclusione, ossia l'inserimento in materiali più resistenti, la disidratazione, per eliminare la componente acquosa, e il sezionamento, ossia la suddivisione del tessuto in sezioni molto sottili per permettere alla luce del microscopio di attraversarlo. La maggioranza dei microscopi utilizzati per le analisi istologiche al giorno d'oggi sono digitali, ossia permettono di osservare l'immagine del tessuto che si sta analizzando attraverso un monitor. Questi strumenti consentono anche di salvare le immagini ottenute ed esse solitamente sono ad altissima risoluzione.

Con il termine CAD (Computer-aided diagnosis) si intende la diagnosi assistita da calcolatore, ovvero una tecnica con cui i medici si avvalgono di software specifici per avere una seconda opinione di conferma per arrivare ad una diagnosi finale, che resta comunque dipendente dall'osservazione diretta da parte del medico. Con la tecnica CAD, le prestazioni del software non devono essere paragonabili o migliori di quelle dei medici, ma devono essere complementari. Con il termine ACD (automated computer diagnosis), invece, si intende la diagnosi automatizzata, in cui l'output del software di analisi rappresenta la diagnosi finale. Per ovvie ragioni di sicurezza e affidabilità l'utilizzo di quest'ultima modalità è poco diffusa, e ad oggi praticamente tutti i sistemi disponibili si pongono come aiuto all'operatore sanitario e non come sostituto dello stesso. L'uso di strumenti di analisi quantitativa delle immagini, insieme all'esperienza del medico, può migliorare la sensibilità e la specificità diagnostica e ridurre il tempo di interpretazione necessario per arrivare ad una diagnosi. Attualmente il focus su questo settore è abbastanza alto ed il mercato globale dei sistemi CAD sfiorerà i 2 miliardi di dollari americani nel corso del 2022 e i campi di applicazione si stanno diversificando. Nel 2014 il principale ambito di applicazione era la diagnostica mammografica, ma oggi i sistemi CAD supportano i medici anche nell'individuazione del cancro al polmone, alla prostata, al fegato, al colon/retto e dei tumori cerebrali e muscoloscheletrici. La tipica architettura di un sistema CAD è composta da pre-processamento dell'immagine, definizione delle regioni d'interesse, estrazione e selezione delle caratteristiche utili ed infine classificazione dell'immagine. Una delle peggiori criticità nello sviluppo dei sistemi CAD basati su algoritmi di classificazione è la fase di training del modello. L'utilizzo di un dataset non molto ampio e/o non rappresentativo di tutte le situazioni cliniche possibili o contenente immagini di bassa qualità può portare a valori di precisione e accuratezza non soddisfacenti. L'utilizzo di strumenti informatici per l'analisi di immagini medicali è in rapida crescita ed espansione da molti anni. Già alla fine degli anni '50 , con l'avvento dei primi computer, i ricercatori in campo biomedico iniziarono a valutare la possibilità di utilizzare questi strumenti tecnologici per analizzare e risolvere problemi in biologia e medicina e come aiuto nel processo di diagnosi di patologie. Questi primi sistemi CAD utilizzavano diagrammi di flusso, riconoscimento di pattern tipici e teoria della probabilità e usavano come input i valori di laboratorio del paziente e i suoi sintomi. Ovviamente, la tecnologia presente in quegli anni non consentiva ancora di processare le immagini con dei software complessi. Ben presto emersero tutte le criticità e tutti i limiti di questo nuovo metodo, ma fu subito chiara l'importanza e l'impatto che avrebbe avuto in futuro. Durante gli anni '60, per la prima volta un gruppo di radiologi cominciò a sviluppare una forma di CAD per individuare anormalità nelle immagini mediche, in particolare per

individuare il tumore alle ossa (Gwilym S. Lodwick, 1963). Oggi, la radiologia diagnostica e l'imaging medicale rappresentano il più importante campo di ricerca e applicazione della tecnologia CAD.
Inoltre, in riferimento alla situazione americana, le cause per negligenza medica sono aumentate drammaticamente a partire dagli anni '80, e conseguentemente sono aumentati i costi delle assicurazioni di responsabilità dei medici, fatto che a sua volta ha causato un aumento generale dei costi della sanità. Questo è uno dei motivi principali che hanno spinto ad investire nella ricerca e sviluppo in campo CAD, con l'obiettivo di ridurre la probabilità di errore da parte dei medici.

Il metodo proposto in questa tesi è stato testato su un dataset di immagini istologiche di tessuto del colon, la metà delle quali presentava segni di adenocarcinoma e l'altra metà riferita a soggetti sani. Le immagini di tessuti con adenocarcinoma presentano delle caratteristiche strutture circolari e/o allungate che le contraddistinguono da quelle di tessuti sani. Un adenocarcinoma è un carcinoma, quindi un tumore maligno, che ha origine da cellule epiteliali ghiandolari, presenti negli organi ghiandolari esocrini (pancreas, mammelle, prostata, ecc) e in generali nei tessuti con proprietà secretorie, per esempio la mucosa di rivestimento dell'esofago, dello stomaco e del colon. L'adenocarcinoma al colon è un tipo di cancro che ha origine nel colon, che è il tratto finale dell'apparato digestivo umano, e che solitamente colpisce in età avanzata, anche se in rari casi può comparire anche in soggetti giovani. Solitamente ha inizio con dei grumi piccoli e benigni di cellule, chiamati polipi, che si formano a causa della crescita incontrollata di cellule della mucosa del colon. Con il passare del tempo, questi polipi possono diventare maligni dando origine al tumore del colon-retto vero e proprio. Inizialmente la presenza di polipi intestinali non produce nessun sintomo al soggetto interessato, e per questa ragione il modo più efficace per individuare e contrastare il cancro al colon è attraverso screening medici periodici. I medici consigliano di iniziare lo screening quando si superano i 50 anni di età ma nelle persone ad alto rischio, ad esempio nel caso di precedenti casi di tale cancro in famiglia, è bene iniziare prima. Intervenire tempestivamente durante le prime fasi della malattia aumenta sensibilmente la possibilità di sopravvivenza. I fattori di rischio per il cancro al colon-retto sono numerosi, tra cui l'età, la presenza di malattie intestinali infiammatorie croniche come il morbo di Crohn, una dieta povera di fibre e ricca di grassi, uno stile di vita sedentario, diabete, obesità, fumo e un uso eccessivo di bevande alcoliche. Con il progredire della malattia iniziano a manifestarsi alcuni sintomi tipici di questa neoplasia tra cui diarrea o costipazione, presenza di sangue nelle feci, malessere addominale persistente (crampi, dolore, gas eccessivo), stanchezza, perdita di peso inaspettata e sensazione che l'intestino non si svuoti completamente durante l'evacuazione. La progressione di questo tipo di tumore è categorizzata in

4 stadi, da 1 a 4, in ordine di gravità e tenendo conto delle dimensioni e profondità della massa cancerosa, del livello di espansione nei linfonodi vicini e dell'eventuale diffusione in altri organi al di fuori dell'intestino. Diversi trattamenti sono disponibili per contrastare la malattia, sia attraverso la chirurgia che con l'utilizzo di farmaci. Gli interventi chirurgici che si possono effettuare dipendono dallo stadio in cui si trova la malattia, e se questa si trova nello stadio iniziale è possibile intervenire in maniera poco invasiva e poco debilitante per il paziente, ad esempio con la polipectomia, ossia la rimozione e asportazione dei polipi con appositi strumenti durante la colonscopia. In fase avanzata, invece, è necessario ricorrere a tecniche chirurgiche più invasive e debilitanti per il paziente, come la colectomia parziale, ossia l'asportazione di una parte del colon e riconnessione delle parti ancora sane, o addirittura con la creazione di una stomia, ossia un'apertura artificiale nell'addome per permettere la fuoriuscita delle feci senza l'attraversamento del colon. Le terapie farmacologiche comunemente utilizzate sono la chemioterapia, soprattutto per eliminare le eventuali cellule rimaste dopo un intervento chirurgico di asportazione, l'immunoterapia, che si avvale di speciali farmaci per stimolare il sistema immunitario a combattere il cancro, e la terapia mirata, ovvero l'utilizzo di farmaci in grado di attaccare unicamente le cellule tumorali.

Il tumore del colon-retto rappresenta uno dei maggiori problemi sanitari che affliggono la società al giorno d'oggi. E' attualmente il terzo tipo di cancro più diffuso negli Stati Uniti (10% di tutti i tumori) ma è quello che causa il secondo più alto numero di morti annuali, subito dopo il tumore ai polmoni. Il cancro al colon-retto è il terzo tipo di cancro più diffuso negli uomini (dopo polmoni e prostata) e il secondo più diffuso nelle donne (dopo il tumore al seno) (Sara P. Oliveira, 2021). Fortunatamente, la percentuale di individui americani over 50 che effettua periodicamente lo screening è andata crescendo costantemente nel corso degli anni, dal 38% nel 2000 al 66% nel 2018, secondo i dati del National Center for Health Statistics (NHIS). Il tasso di sopravvivenza a 5 anni dalla diagnosi è stato del 65% nel periodo 2011-2017. Il tasso d'incidenza annuale di nuovi casi è di 37.8 casi ogni 100.000 individui e il tasso di mortalità annuale è di 13.4 morti ogni 100.000 individui. Si stima che circa al 4,1% della popolazione totale sarà diagnosticato il cancro al colon-retto nel corso della loro vita. Nel 2018 erano presenti negli Stati Uniti d'America circa 1,4 milioni di persone affette dal cancro al colon-retto.

Capitolo 2. Literature review

L'idea di utilizzare i grafi per rappresentare immagini è stata già studiata e affrontata da parte di diversi ricercatori e in diversi ambiti. L'uso di grafi per l'image processing and analysis consente di estrarre dall'immagine in questione dei modelli strutturali rappresentati gli oggetti che compongono l'immagine (A. Sanfeliu, 2000) . Uno dei problemi principali riscontrati inizialmente e che ha fortemente limitato l'utilizzo di queste tecniche è la complessità computazione degli algoritmi proposti. Infatti, iterare ciascun nodo con tutti gli altri nodi del grafo richiede del tempo polinomiale. Nonostante ciò, di recente i nuovi sviluppi nell'ambito di algoritmi approssimativi hanno consentito di ridurre fortemente il tempo necessario per arrivare a delle soluzioni sub-ottimali. Una buona parte della teoria che è alla base di questo lavoro di tesi è stata proposta e sviluppata dal Dr. Cheng-Bang Chen, professore presso il dipartimento di Industrial Engineering della University of Miami e prolifico ricercatore nel campo della teoria dei grafi e dei network applicata alle immagini. In particolare, sono due i paper scientifici d'interesse per questo progetto. Il primo (Cheng-Bang Chen, Recurrence network modeling and analysis of spatial data, 2018) propone un nuovo metodo per la rappresentazione e la visualizzazione di ricorrenze in dati spaziali, come ad esempio un'immagine. Questo primo metodo è applicabile solamente alle immagini a scala di grigi, in cui ciascun pixel ha un valore che va da 0 a 255, ovvero dal nero al bianco.

Un grafo è un insieme di elementi chiamati nodi o vertici che possono essere collegati fra loro da delle connessioni dette archi o spigoli. Più precisamente, si definisce grafo una coppia ordinata $G = (V, E)$, dove V è l'insieme dei nodi ed E è l'insieme degli archi. Due nodi uniti da un arco vengono chiamati estremi dell'arco. Un grafo si dice non orientato se ciascun arco è identificato da una coppia di vertici indipendentemente dal loro ordine. Un grafo si dice non pesato se a tutti gli archi che lo compongono non è associato un valore numerico.

Un arco prende il nome di cappio o self-loop se ha due estremi coincidenti.

Nella metodologia proposta ciascun pixel dell'immagine è rappresentato con un nodo di un grafo non orientato, non pesato e senza self-loops. Dunque, il grafo riferito all'immagine sarà costituito da un numero di nodi pari al numero di pixel dell'immagine.

Successivamente, per la definizione degli archi viene presa in considerazione sia la similarità in termini di colore che la distanza geometrica che separa i due pixel. Nella Figura 1 sono evidenziati due pixel, ciascuno di essi identificato da un vettore di due valori, la coordinata x e la coordinata y.

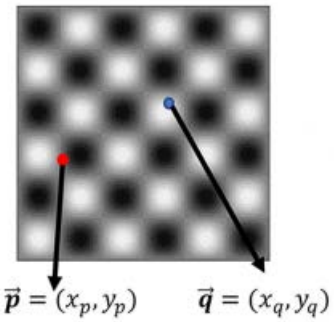

Figura 1 – Esempio di due pixel

Il peso dell'arco che collega il nodo corrispondente al pixel in rosso con quello corrispondente al pixel in blu è dato dalle seguenti equazioni:

(1) $w_{p,q} = I_{p,q} * D_{p,q}$

(2) $I_{p,q} = 1 - \frac{\vec{S}_p - \vec{S}_q}{max\{\|\vec{S}.\|\} - min\{\|\vec{S}.\|\}}$

(3) $D_{p,q} = \frac{\phi(\vec{p} - \vec{q})}{\phi(\|0\|)}$

Il peso dell'arco è rappresentato con $w_{p,q}$ ed è il prodotto di $I_{p,q}$ (che dipende dalla similarità di intensità di grigio tra i due pixel) per $D_{p,q}$ (che dipende dalla correlazione spaziale tra i due pixel). Nell'equazione (2) viene illustrato come viene ponderata la differenza di intensità di grigio tra i due pixel. $\vec{S}_p$ ed $\vec{S}_q$ indicano rispettivamente l'intensità di grigio del primo e del secondo pixel e la differenza tra di essi viene normalizzata considerando il massimo valore di intensità di grigio presente nell'immagine e il minimo valore. Dunque, $I_{p,q}$ è un valore compreso tra [0,1] e se $I_{p,q}$ = 1 allora i due pixel hanno la stessa identica intensità di grigio, mentre se $I_{p,q}$ = 0 allora i due pixel hanno intensità di grigio opposte. L'equazione (3) illustra come la distanza spaziale tra i due pixel viene computata. Viene utilizzata una funzione di utilità $\phi(\cdot)$. In generale, la funzione di utilità che viene utilizzata deve avere le seguenti proprietà: localmente supportata, non negativa e monotona decrescente in funzione della distanza spaziale $\varphi = \vec{p} - \vec{q}$. La funzione scelta in questo caso è la funzione di utilità gaussiana , $\phi(x|\Sigma) = (2\pi|\Sigma|)^{-0.5} \exp\{-(1/2)x^T\Sigma x\}$. Il valore di $D_{p,q}$ è compreso

tra [0,1). Esso è inversamente proporzionale alla distanza geometrica tra due pixel, dunque se $\vec{p} - \vec{q} < \vec{p} - \vec{r}$ allora $D_{p,q} > D_{p,r}$.

Il valore di $w_{p,q}$ è anch'esso compreso tra 0 e 1 e dopo aver calcolato questo valore per ogni coppia di pixel è possibile costruire la matrice delle adiacenze del grafo con l'equazione:

$$\boldsymbol{A}_{p,q} = \Theta\,(w_{p,q} \geq \xi\,) - \Delta_{p,q}$$

ξ è un valore soglia che viene deciso prima di iniziare il processo.

Θ è la funzione gradino di Heaviside, che assume il valore 1 se $w_{p,q}$ è maggiore o uguale al valore di soglia ξ e 0 altrimenti.

$\Delta_{p,q}$ è il delta di Kronecker, che assume un valore pari a 1 se $p = q$ e 0 altrimenti. Esso è utilizzato per evitare i self-loops, ovvero i casi in cui un arco connette un nodo a sè stesso.

Come si evince dall'equazione, $\boldsymbol{A}_{p,q}$ è pari a 1 se $w_{p,q}$ è maggiore o uguale al valore di soglia e 0 altrimenti.

La matrice delle adiacenze o matrice delle connessioni è una matrice quadrata, ovvero con lo stesso numero di righe e colonne, utilizzata per rappresentare in forma numerica un grafo finito. La matrice ha come indici di righe e colonne il nome dei nodi del grafo. Nel posto (i, j) della matrice si trova il valore 1 se esiste un arco che connette i nodi i e j e 0 se non esiste tale arco.

Nel caso di cui sopra la diagonale principale della matrice di adiacenze è sempre formata da soli valori 0, perché i self-loops non sono ammessi e dunque alla posizione (i, i) si trova sempre il valore 0, per ogni i .

Una volta ottenuto il grafo corrispondente all'immagine in analisi è possibile estrarre le proprietà statistiche topologiche del grafo. Prima, però, è necessario spiegare alcuni concetti fondamentali riguardanti i grafi.

1. Grado

Il grado di un nodo è il numero di archi che connettono il nodo in questione ad altri nodi.

In un grafo non diretto il grado di un nodo generico i è indicato come $k_i = \sum_{j=1}^{n} A_{i,j}$.

$A_{i,j}$ è il valore presente nella matrice delle adiacenze in posizione (i, j), come già visto in precedenza.

Analizzando la distribuzione statistica di $\ll k_i \gg$ è possibile estrarre informazioni importanti circa la distribuzione delle ricorrenze in un'immagine.

2. Cammino

Un cammino tra i e j è una rotta, intesa come successione di archi, che viene attraversata per raggiungere il nodo j partendo dal nodo i e viceversa, visto che si sta sempre parlando di grafi non orientati. La lunghezza di un cammino è il numero di archi che compongono il cammino. Nel metodo appena analizzato un cammino caratterizza come una specifica ricorrenza connette i pixel dell'immagine. Se sono presenti dei cammini molto lunghi significa che ci sono dei pattern molto frequenti nell'immagine.

3. Distanza

La distanza tra due nodi è il cammino più breve possibile (cammino minimo) tra di essi. Una distanza maggiore di 1 implica che i due nodi non sono direttamente connessi.

4. Lunghezza media del cammino

La lunghezza media del cammino, in inglese "average path length", è la media delle distanze di tutte le coppie di nodi. Questa statistica fornisce informazioni sulla distribuzione degli eventuali pattern nell'immagine. L'equazione matematica dell'average path length è $L = \frac{1}{n(n+1)} \sum_{i \neq j} \frac{1}{D_{i,j}}$, dove $D_{i,j}$ è la distanza tra il nodo i e il nodo j.

5. Betweenness centrality

La betweenness centrality è una misura di centralità e quantifica il numero di cammini minimi che passano per un nodo. La betweenness centrality indica quanto è importante un nodo nella comunicazione tra diverse parti del grafo. E' definita matematicamente come $BC(i) = \sum_j \sum_k \frac{p(i,j,k)}{p(j,k)} \; con \; i \neq j \neq k$, dove $p(j,k)$ è il numero di cammini minimi che collegano il nodo j al nodo k e $p(i,j,k)$ è il numero di cammini più brevi tra j e k che passano anche per il nodo i.

6. Transitività o coefficiente di clustering

Il coefficiente di clustering misura la probabilità che due nodi connessi con un nodo i siano a loro volta connessi tra di loro, ovvero formando un triangolo. E' cioè una misura di quanto i nodi di un grafo tendono a formare clusters, ovvero insiemi di nodi con delle connessioni molto fitte tra di loro. Il coefficiente di clustering di un nodo i può essere espresso come $C_i = \frac{2\Delta_i}{k_i(k_i-1)}$, dove Δ_i è il numero di triangoli centrati sul nodo i.

Il paper contiene anche i risultati della fase sperimentale, dove il metodo proposto è stato applicato a 3 immagini diverse: un'immagine di rumore casuale, ovvero formata da pixel ognuno dei quali con intensità di grigio casuale, un'immagine a strisce ognuna con un certo livello di intensità di grigio e un'immagine a scacchiera. I risultati hanno mostrato che le misure sopraelencate sono nettamente diverse per ognuno dei tre grafi corrispondenti alle immagini, dimostrando così che il metodo riesce a identificare e distinguere tra diversi pattern.

Il secondo paper scientifico (Cheng-Bang Chen, Heterogeneous recurrence analysis of spatial, 2019) che ha fornito un punto di partenza teorico per questo lavoro di tesi presenta e descrive un nuovo metodo di identificazione delle ricorrenze eterogenee in dati spaziali in due dimensioni (immagini) o in tre dimensioni. Questo metodo, al contrario del precedente, viene applicato alle immagini a colori espressi in RGB. Questa nuova metodologia integra e utilizza, in versione modificata, i diagrammi di ricorrenza e le statistiche RQA.

Un diagramma di ricorrenza, o "recurrence plot" in lingua inglese, è uno strumento molto efficiente e ampiamente utilizzato per lo studio e la visualizzazione di ricorrenze in dati unidimensionali, come ad esempio le serie temporali (Norbert Marwan, 2006). La ricorrenza è una proprietà fondamentale di molti sistemi dinamici e di molti processi presenti in natura. Un sistema dinamico può essere rappresentato matematicamente in uno spazio delle fasi. Lo spazio delle fasi è uno spazio in cui tutti i possibili stati del sistema possono essere rappresentati, con ogni possibile stato che corrisponde ad un unico punto nello spazio. Ad esempio, se ogni stato del sistema dinamico in considerazione è descritto da 3 variabili di stato allora lo spazio delle fasi è uno spazio tridimensionale in cui ogni asse rappresenta una variabile. Si ha una ricorrenza quando un sistema dinamico si trova all'istante t_2 nello stesso (o molto simile) stato in cui si trovava all'istante t_1, con $t_1 \neq t_2$, ovvero quando i due stati occupano lo stesso punto nello spazio delle fasi.
Un recurrence plot, da qui in avanti indicato con RP, è un diagramma di forma quadrata in cui entrambi gli assi rappresentano il tempo ed esso è formato da punti bianchi e neri. Un RP è costruito seguendo l'equazione:

$$\mathbf{R}_{i,j} = \Theta\left(\varepsilon - \left\|\vec{x}_i - \vec{x}_j\right\|\right), \qquad \vec{x}_i \in \mathbb{R}^m, \quad i,j = 1 \dots N$$

dove i e j rappresentano due istanti di tempo, $\vec{x}_i$ è il vettore di valori che descrive lo stato del sistema all'istante i, ε è un valore di soglia e Θ è la funzione di Heaviside, già accennata in precedenza. Dunque, se la differenza tra i due stati del sistema nei due istanti di tempo è inferiore

al valore di soglia allora $\mathbf{R_{i,j}} = 1$ altrimenti $\mathbf{R_{i,j}} = 0$.

Se $\mathbf{R_{i,j}} = 1$ allora nel RP in posizione (i, j) è presente un punto nero, altrimenti il punto è bianco.

La Figura 2, riportata di seguito, è un esempio di RP.

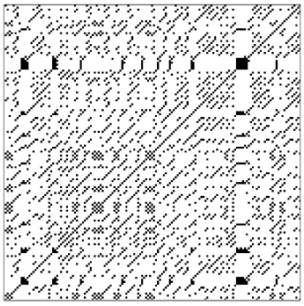

Figura 2 – Esempio di Recurrence Plot

In ogni RP è presente una diagonale che collega l'estremo inferiore sinistro con quello superiore destro. Questa linea è chiamata "linea di identità", in inglese "line of identity" (LOI). Su questa linea $i = j$ sempre, e quindi $\mathbf{R_{i,j}} = 1$. Un RP è simmetrico rispetto a questa linea per definizione, infatti se $\mathbf{R_{i,j}} = 1$ allora anche $\mathbf{R_{j,i}} = 1$. Alcuni elementi grafici di un RP possono fornire informazioni importanti. Le presenza di linee diagonali, di lunghezza l, implica che la traiettoria che descrive l'evoluzione del sistema dinamico nello spazio delle fasi rivisita la stessa regione dello spazio ma in tempi differenti. Le linee diagonali e verticali ,di lunghezza v, indicano dei lassi di tempo in cui lo stato del sistema non cambia o cambia molto lentamente. Come già accennato in precedenza, un RP è simmetrico rispetto alla linea di identità e dunque se è presente una linea verticale in uno dei due triangoli allora è presente una linea orizzontale di uguale lunghezza nell'altro triangolo, e viceversa. Analizzando la distribuzione delle lunghezze delle linee diagonali $P(l)$ e delle linee verticali/orizzontali $P(v)$ è possibile estrarre informazioni importanti sul comportamento del sistema dinamico in questione. Questo metodo prende il nome di "Analisi di quantificazione delle ricorrenze", in inglese "Recurrence quantification analysis" o semplicemente RQA. Prima di tutto, però, è necessario definire delle lunghezze minime l_{min} e v_{min} . Queste devono essere più piccole possibili ma abbastanza grandi da escludere strutture simili a linee ma che in realtà rappresentano stati non ricorrenti, che può succedere quando si sceglie un valore di soglia troppo alto o se i dati sono stati arrotondati troppo prima di calcolare i valori per costruire il RP. Di seguito sono elencate le misure di RQA per sistemi dinamici con dimensionalità d , che in questo caso è pari a 1.

- **Recurrence Rate** $RR = \frac{1}{N^{2d}} \sum_{\vec{i},\vec{j}}^{N} R_{\vec{i},\vec{j}}$

Esso esprime la percentuale di stati ricorrenti nel sistema, ovvero indica la probabilità di ricorrenza di ogni stato

- **Determinismo** $\mathrm{DET}_{HS} = \frac{\sum_{l=l_{min}}^{N} l\, P(l)}{\sum_{\vec{i},\vec{j}}^{N} R_{\vec{i},\vec{j}}}$

E' la percentuale di punti nel RP che formano linee diagonali e questo valore fornisce informazioni sulla prevedibilità del sistema

- **Laminarità** $\mathrm{LAM}_{HS} = \frac{\sum_{v=v_{min}}^{N} v\, P(v)}{\sum_{v=1}^{N} v\, P(v)}$

E' la percentuale di punti del RP che formano linee verticali.

- **Trapping size** $\mathrm{TT}_{HS} = \frac{\sum_{v=v_{min}}^{N} v\, P(v)}{\sum_{v=v_{min}}^{N} P(v)}$

Questo valore è riferito alla grandezza dell'area in cui il sistema non cambia

Il metodo RP può essere applicato a sistemi con qualsiasi dimensionalità d, ma la dimensionalità del RP è $n = 2xd$, dunque se il sistema dinamico ha dimensionalità maggiore di 1 allora il RP risultante può essere usato per applicare la RQA ma non può più essere visualizzato. In questo caso nel RP non è presente una linea di identità ma bensì un'ipersuperficie di identità. Allo stesso modo non sono presenti linee verticali e orizzontali ma ipersuperfici verticali e orizzontali.

Come già accennato, il secondo paper del Dr. Chen amplia il metodo RQA per analizzare le ricorrenze in modo eterogeneo, e la metodologia proposta prende il nome di "Analisi quantitativa delle ricorrenze eterogenee", ossia "Heterogeneous recurrence quantification analysis" o HRQA. Fino a questo punto, infatti, le ricorrenze sono state trattate in modo omogeneo mentre ora vengono

trattate in modo eterogeneo, ossia discriminando tra diversi tipi di transizioni di stato. La metodologia HRQA viene applicata ai dati spaziali, come le immagini, e si articola in 5 fasi in sequenza. Nella Figura 3 viene riassunto e illustrato l'intero processo.

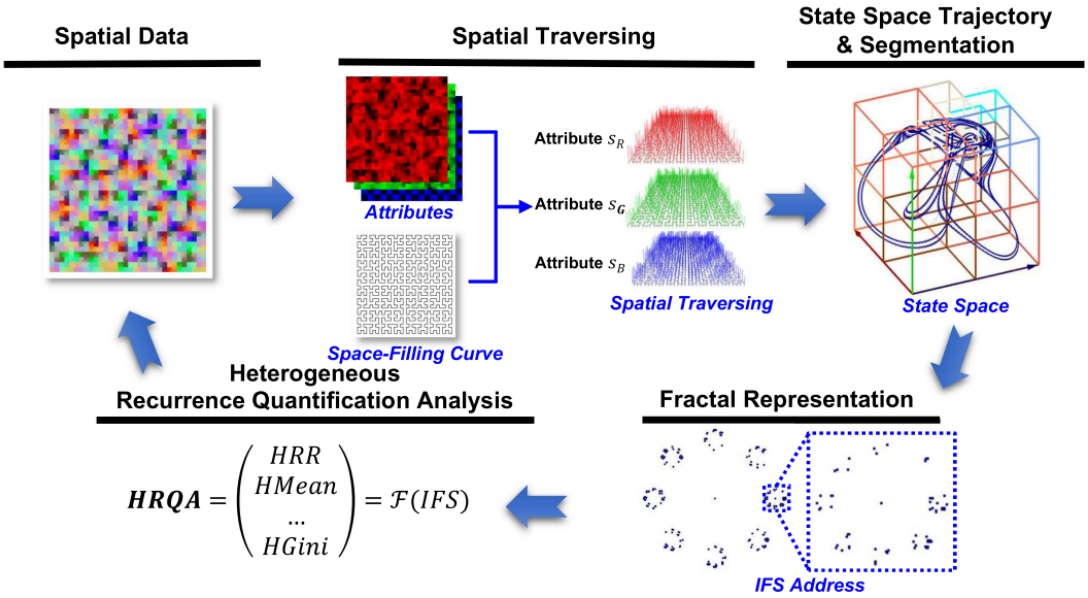

Figura 3 – Processo HRQA

- **Attraversamento spaziale**

  Nella prima fase viene utilizzata una Space-Filling Curve (SFC) per attraversare l'immagine in modo da preservare la vicinanza spaziale dei vari pixel e che aiuta a trasformare l'immagine in delle serie di attributi che sono utilizzati per costruire lo spazio dello stato e proseguire con il processo. Una Space-Filling Curve è una curva auto-simile e ricorsiva che attraversa lo spazio. Nella Figura 4 sono illustrati tre diversi tipi di SFC.

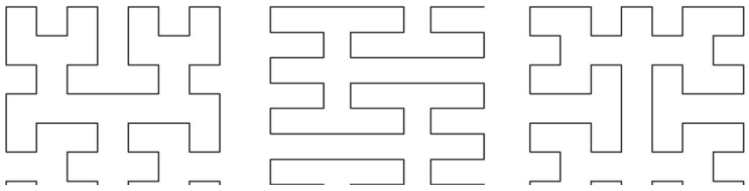

Figura 4 – Tre diverse Space-filling Curves

  In questa occasione è stata utilizzata la curva di Hilbert che è sembrata essere la migliore per preservare la vicinanza spaziale tra gli elementi ed inoltre è più bilanciata nelle direzioni che attraversa (nord, est, sud, ovest). Ciò non esclude che possano essere utilizzati anche altri tipi di SFC che magari si prestano meglio in alcune determinate applicazioni. La curva di

Hilbert è un modo per attraversare uno spazio d-dimensionale e la curva segue delle regole che la rendono ricorsiva. Per generare una curva di Hilbert sono necessari due parametri: il livello l, da cui dipende la dimensione della curva, e l'orientamento v. Una curva di Hilbert di livello l attraversa uno spazio d-dimensionale attraverso 2^{dxl} regioni. L'attraversamento dello spazio può avvenire in 2^d diversi orientamenti. Se due pixel sono vicini nell'immagine allora sono vicini anche sulla curva.

- **Segmentazione dello spazio degli stati**

  Dopo che le curve di Hilbert attraversano tutto lo spazio, in questo caso tutta l'immagine, gli attributi dei pixel vengono convertiti in serie storiche multivariate che possono essere accorpate in un vettore multidimensionale i cui elementi formano una traiettoria nello spazio degli stati, che ha come assi le variabili R,G e B, che indicano l'intensità dei colori rosso verde e blu. Ogni punto s_i della traiettoria rappresenta un pixel ben preciso che è posizionato nello spazio a seconda delle sue intensità di rosso verde e blu. La lunghezza della traiettoria che separa due punti in questo spazio è proporzionale al loro ordine di attraversamento da parte della curva di Hilbert. Per individuare le ricorrenze eterogenee e transizioni stocastiche si segmenta lo spazio degli stati al fine di partizionare la traiettoria in regioni locali, e ad ogni regione è assegnata una variabile locale. La segmentazione, oltre a facilitare l'individuazione di pattern diversi, riduce anche lo sforzo computazionale richiesto. Per partizionare lo stato si utilizza la "Hyperoctree Aggregate Segmentation" o HAS, che è un metodo di partizione basato sugli alberi, dove ogni regione di spazio è un ramo dell'albero che può essere ancora divisa in sottoalberi. Questo metodo partiziona ogni regione ricorsivamente fino a quando il numero di stati in ogni regione è inferiore ad un certo limite. Se una regione supera il limite allora è ulteriormente partizionata in 2^m sotto-regioni, dove m è la dimensionalità dello spazio degli stati. Con la partizione, quindi, ad ogni stato s_i della traiettoria viene assegnato un valore categorico. La funzione di segmentazione, quindi, trasforma la traiettoria in una serie di valori categorici. Tutto questo processo è illustrato e sintetizzato nella Figura 5 all'inizio della prossima pagina.

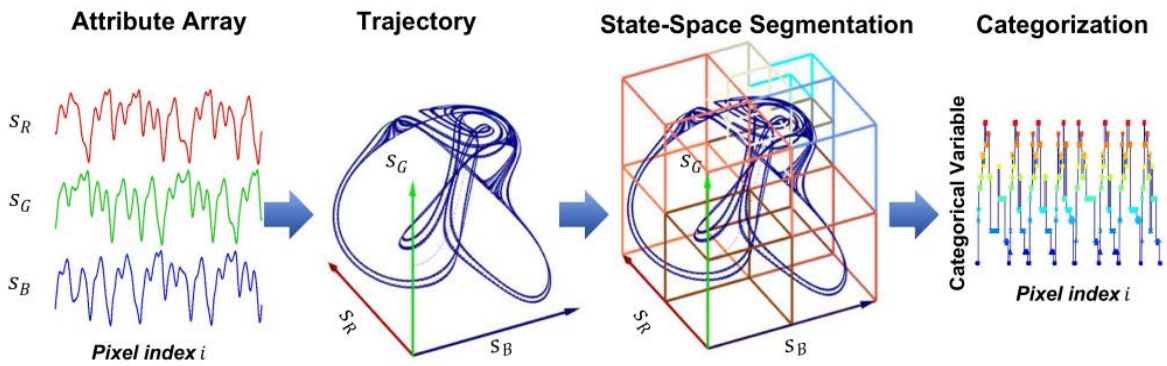

Figura 5 – Segmentazione dello spazio degli stati

- **Rappresentazione frattale**

  Il passo successivo è definire un sistema di funzioni iterate, che è un metodo usato per costruire i frattali. Un frattale è una figura geometrica dotata di omotetia interna, cioè la sua forma si ripete ricorsivamente e quindi qualsiasi porzione di questa figura ha una forma uguale all'intera figura. In questo caso viene usato il sistema di funzioni iterate di trasformazione circolare, che è una funzione biunivoca che mappa sequenzialmente le serie categoriche in rappresentazione frattale. Come già accennato prima, la funzione di segmentazione trasforma la traiettoria in una serie di variabili categoriche e il sistema di funzioni iterate mappa ogni variabile categorica k, assegnata allo stato s_i , in un unico indirizzo nel cerchio frattale. Il sistema di funzioni iterate crea una struttura frattale che incorpora l'informazione di tutti gli stati precedenti fino a quel punto. La Figura 6 riporta un esempio della rappresentazione frattale appena descritta.

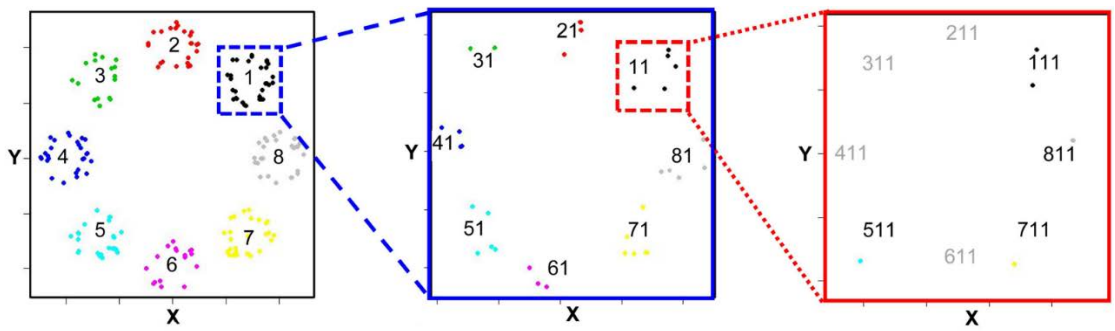

Figura 6 – Rappresentazione in frattale delle ricorrenze eterogenee

Nell'esempio è mostrata la rappresentazione frattale di dati spaziali in termini di regioni

individuali, transizione di 2 regioni, transizione di 3 regioni.

- **Recurrence plot eterogeneo**

  Questo passo non è presente nella Figura 3 , ma è utile per visualizzare le ricorrenze eterogenee presente nell'immagine. Un RP eterogeneo, in inglese "Heterogeneous recurrence plot" o HRP, è una matrice di dimensioni $|i|x|i|$ che è descritta dalla seguente equazione:

$$\mathbf{HRP}_{i,j} = \mathcal{L}(s_i) * \delta_{\mathcal{L}(s_i),\mathcal{L}(s_j)}$$

$$= \begin{cases} 0, se\ \mathcal{L}(s_i) \neq \mathcal{L}(s_j) \\ \mathcal{L}(s_i), se\ \mathcal{L}(s_i) = \mathcal{L}(s_j) \end{cases} \qquad \forall\, i,j\ \in \{1,2,\dots,I\}$$

  dove $\mathcal{L}(s_i) = k \in \{1,2,\dots,K\}$ con $K =$ numero di regioni dello spazio degli stati, $k =$ valore univoco usato per indicare l'appartenenza ad una specifica regione e $\delta_{(.)} =$ delta di Kronecker, con $\delta_{(i,j)} = 1$ se $i = j$ e 0 altrimenti. La costruzione di un HRP è molto simile a quella di un RP tradizionale, ma in quel caso il valore di $RP_{i,j}$ può essere soltanto 1 o 0, corrispondenti al colore nero e bianco. In un HRP, invece, il numero di colori presenti usati per rappresentare i punti è uguale al numero di regioni in cui è stato segmentato lo spazio degli stati. Ad ogni valore univoco, usato per indicare in maniera univoca una regione, viene assegnato un colore usato esclusivamente per quel valore. Si supponga che sia i che j appartengano alla stessa regione di spazio, indicata con il valore univoco 3, allora in questo caso $HRP_{i,j} = 3$. Si supponga che a $k = 3$ sia stato assegnato il colore univoco verde, allora nel HRP in posizione (i,j) e (j,i) è presente un punto di colore verde. La Figura 7 mostra un esempio di HRP.

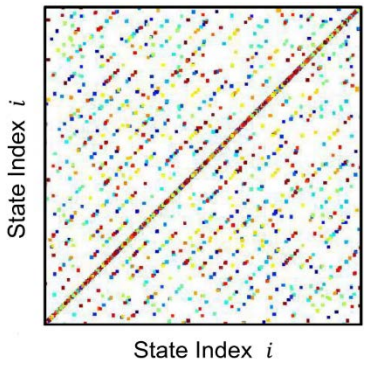

Figura 7 – Esempio di Heterogeneous Recurrence Plot

- **Analisi di quantificazione delle ricorrenze eterogenee**

  L'analisi di quantificazione delle ricorrenze eterogenee, in inglese "Heterogeneous recurrence quantification analysis" o HRQA, è un metodo per quantificare e analizzare le ricorrenze eterogenee in dati spaziali, in questo caso sempre un'immagine. Il metodo è un'estensione del già citato RQA tradizionale, che invece quantifica le ricorrenze trattate in modo omogeneo, senza distinzioni. Il primo passo di questo nuovo metodo è identificare i set di stati che appartengono alla stessa regione o se ci sono transizioni tra regioni multiple nella rappresentazione frattale. Visto che il sistema di funzioni iterative raggruppa i set di stati con indirizzi univoci nei cerchi del frattale, vengono definiti questi set di ricorrenze eterogenee $H_{k_1,k_2,\dots,k_N}$ come:

$$H_{k_1,k_2,\dots,k_N} = \{w(k_1|k_2,\dots,k_N) : \mathcal{L}(s_i) \to k_1, \mathcal{L}(s_{i-1}) \to k_2,\dots,\mathcal{L}(s_{i-N+1}) \to k_N,\ \forall k_i \in K\}$$

  dove $k_1, k_2, \dots, k_N$ una sequenza di transizione di *N*-regioni. Ad esempio, H_{k_1} rappresenta il set di ricorrenze nelle regioni individuali, H_{k_1,k_2} rappresenta il set di ricorrenze tra due regioni, tale che $\{\mathcal{L}(s_i) \to k_1,\ \mathcal{L}(s_{i-1}) \to k_2\}$, $H_{k_1,k_2,\dots,k_N}$ indica il set di ricorrenze tra *N*-regioni tale che $\{\mathcal{L}(s_i) \to k_1,\ \mathcal{L}(s_{i-1}) \to k_2,\dots,\mathcal{L}(s_{i-N+1}) \to k_N\}$. Le statistiche di HRQA forniscono informazioni importanti circa la quantità e il tipo di ricorrenze presenti nell'immagine che si sta analizzando. Esse sono:

- **Heterogeneous recurrence rate (HRR)**

$$HRR(k_1,k_2,\dots,k_N) = \left(\frac{\overline{\overline{H}}}{L}\right)^2$$

  Esso quantifica la proporzione di una specifica sequenza di transizione $k_1, k_2, \dots, k_N$.

  $\overline{\overline{H}}$ è la cardinalità di $H_{k_1,k_2,\dots,k_N}$ e L è la lunghezza del percorso attraversato dalla curva di Hilbert. HRR fornisce informazioni circa le densità delle ricorrenze di pattern di transizione multi-regione nello spazio.

- **Misure basate sulla distanza**

  Il sistema di funzioni iterate a trasformazione circolare fornisce un indirizzo unico ad ogni stato nella rappresentazione frattale e ogni indirizzo è determinato dallo stato attuale e dallo storico degli stati precedenti. La distribuzione degli indirizzi dei set di ricorrenze eterogenee dipende dall'eterogeneità delle transizioni di stato. E' possibile estrarre le distanze tra due indirizzi i e j, $D_{k_1,k_2,\dots,k_N}(i,j)$, per ogni set di ricorrenze eterogenee $H_{k_1,k_2,\dots,k_N}$. Basandosi sulla distribuzione delle misure di distanza, è possibile calcolare la "Heterogeneous recurrence mean" (**HRmean**), "Heterogeneous recurrence variance" (**HVar**), "Heterogeneous recurrence skewness" (**HSkew**), "Heterogeneous recurrence kurtosis" (**HKurtosis**).

$$HMean(k_1,k_2,\dots,k_N) = \frac{\sum_{i=1}^{\bar{\bar{H}}}\sum_{j=i+1}^{\bar{\bar{H}}} D_{k_1,k_2,\dots,k_N}(i,j)}{\bar{\bar{H}}(\bar{\bar{H}}-1)/2}$$

$$HVar(k_1,k_2,\dots,k_N) = \sum_{i=1}^{\bar{\bar{H}}} \frac{\sum_{j=i+1}^{\bar{\bar{H}}}(D_{k_1,k_2,\dots,k_N}(i,j) - HMean)^2}{\bar{\bar{H}}(\bar{\bar{H}}-1)/2}$$

$$HSkew(k_1,k_2,\dots,k_N) = \frac{\sum_{i=1}^{\bar{\bar{H}}} \frac{\sum_{j=i+1}^{\bar{\bar{H}}}(D_{k_1,k_2,\dots,k_N}(i,j) - HMean)^3}{\bar{\bar{H}}(\bar{\bar{H}}-1)/2}}{HVar^{\frac{3}{2}}}$$

$$HKurtosis(k_1,k_2,\dots,k_N) = \frac{\sum_{i=1}^{\bar{\bar{H}}} \frac{\sum_{j=i+1}^{\bar{\bar{H}}}(D_{k_1,k_2,\dots,k_N}(i,j) - HMean)^4}{\bar{\bar{H}}(\bar{\bar{H}}-1)/2}}{HVar^2}$$

Inoltre, è possibile quantificare l'entropia e il coefficiente di Gini delle ricorrenze eterogenee, chiamate "Heterogeneous recurrence entropy" (**HENT**) e "Heterogeneous Gini Index" (**HGini)**. HENT è la misura dell'entropia di Shannon basata sulla distribuzione di probabilità di $D_{k_1,k_2,\dots,k_N}(i,j)$. L'istogramma della matrice delle distanze $D_{k_1,k_2,\dots,k_N}$ viene diviso in B bins uguali, da 0 a $\max(D)$, e la probabilità $p(b)$ è calcolata con la formula:

$$p(b) = \frac{1}{\bar{\bar{H}}(\bar{\bar{H}} - 1)} \# \left\{ \frac{b-1}{B} \max(D) < D_{k_1, k_2, \ldots, k_N(i,j)} \right.$$

$$\left. \leq \frac{b}{B} \max(D) \right.$$

dove $b = 1,2, \ldots, B$. HENT e HGini forniscono informazioni generali sulla incertezza della ricorrenza di una sequenza di transizione di N-regioni.

$$HENT(k_1, k_2, \ldots, k_N) = -\sum_{b=1}^{B} \Pr(b) * \ln(\Pr(b))$$

$$HGini(k_1, k_2, \ldots, k_N) = 1 - \sum_{b=1}^{B} \Pr(b)^2$$

Capitolo 3. Stato dell'arte per la classificazione di immagini

Quando si parla di classificazione ci si riferisce all'utilizzo di particolari algoritmi di machine learning per assegnare un elemento descritto da un insieme di variabili ad una certa categoria. Il Machine Learning è definito come "programmare software per ottimizzare un criterio di performance utilizzando dati di esempio o dati di esperienze passate" (Alpaydin, 2010). Il Machine Learning è uno dei campi tecnologici in cui attualmente vengono investite più risorse e si calcola che nel 2025 verranno investiti oltre 100 miliardi di dollari in questo campo. Gli algoritmi di apprendimento si dividono in due grandi categorie: supervisionati e non supervisionati.

L'apprendimento non supervisionato si avvale dell'utilizzo di algoritmi per analizzare dati senza etichette, ossia non suddivisi in categorie già in precedenza. Gli algoritmi di questo tipo spesso sono in grado di scovare dei pattern nascosti, ovvero non rilevabili con un'osservazione diretta da parte dell'uomo. L'apprendimento non supervisionato è utilizzato prevalentemente per tre scopi:

- **Clustering:** è una tecnica di data mining per raggruppare dei dati senza etichette in insiemi chiamati cluster, basandosi sulla similarità e sulle differenze dei vari elementi su cui si sta

operando. Un algoritmo di clustering molto performante ed utilizzato è il K-means, che suddivide gli elementi del dataset in K cluster minimizzando la varianza totale intra-gruppo. Il numero di cluster viene deciso dall'utente e per questo motivo devono essere svolte diverse prove per arrivare al risultato ottimale. Tornando all'argomento oggetto di questa tesi, l'algoritmo K-means è molto utilizzato per processare immagini digitali, in particolare per la segmentazione di immagini, cioè la partizione di un'immagine in varie componenti. In particolare, l'algoritmo è utilizzato per il riconoscimento dei vari oggetti che compongono l'immagine. Si consideri, ad esempio, l'immagine di una macchia nera sulla pelle di una persona bianca. In questo caso si può applicare l'algoritmo k-means con k=2 per separare la macchia, che è la parte di interesse dell'immagine (ROI- region of interest), dallo sfondo (la pelle).

- **Associazione:** è un altro metodo di apprendimento non supervisionato utilizzato per provare a trovare delle relazioni tra gli elementi di un certo dataset. Le regole di associazione sono molto usate nel mondo del business, l'esempio più classico è quello degli acquisti consigliati quando si compra un oggetto su un sito di e-commerce. Un sito di questo genere solitamente estrae le regole di associazione analizzando tutte le transazioni passate e poi seleziona quelle più significative in base al Support e alla Confidence per proporre ulteriori acquisti al cliente. Si considerino due elementi A e B, il Support (A→B) è il numero di transazioni contenenti sia A che B diviso il numero totale di transazioni, mentre la Confidence (A→B) è il numero di transazioni contenenti sia A che B sul totale di transazioni contenenti A. Una regola di associazione, per essere interessante, deve avere un Support e una Confidence superiori a dei valori minimi impostati dall'utente che sta svolgendo l'analisi. Quando si parla di regole di associazione, un itemset è un insieme di elementi, nell'esempio precedente l'itemset di riferimento era {A,B}. Esistono diversi algoritmi per estrarre le regole di associazione e i popolari sono Apriori e FP-Growth. Apriori è un algoritmo iterativo che come primo step considera ogni singolo oggetto del dataset come un itemset formato da un solo elemento e calcola i vari Support, poi si selezionano gli elementi che hanno ottenuto un Support superiore ad un livello minimo stabilito e si passa ad analizzare tutte le coppie possibili contenenti gli elementi che hanno superato il primo step, e così via. FP-Growth, invece, rappresenta gli elementi in una struttura ad albero chiamata FP-Tree, che è utile per mantenere l'informazione di associazione tra i vari elementi. Dopo aver create l'FP-Tree, esso viene suddiviso in un set di FP-Trees condizionali

per ogni elemento frequente. Un set di FP-Trees condizionali può essere ancora scomposto. I ricercatori hanno provato ad applicare le regole di associazione per estrarre informazioni importanti da immagini digitali e identificare pattern caratterizzanti. Diversi paper scientifici sono stati pubblicati a tal proposito e in uno di essi (Maria-Luiza Antonie, 2002) si è provato a creare un modello di classificazione per immagini mediche basato su regole di associazione. Il metodo è stato provato e testato su immagini mammografiche che, come si è già spiegato in precedenza, sono il campo di applicazione più diffuso per quanto riguarda l'analisi di immagini mediche tramite machine learning. Il metodo proposto classifica le immagini mammografiche digitali in tre categorie: normale, benigno e maligno. Un'immagine è effettivamente "normale" quando è riferita ad una paziente sana, l'etichetta "benigno" è usata per le immagini contenenti dei noduli non cancerosi e l'etichetta "maligno" è usata in presenza di un tumore maligno. Generalmente, la maggior parte degli errori da parte dei medici quando analizzano immagini mammografiche è relativa alla confusione tra noduli benigni e maligni. Prima di analizzare le immagini con questo metodo è necessaria una fase di pre-processing per separare la regione d'interesse dal background e per equiparare le immagini in termini di luminosità e intensità di grigio. Successivamente si passa alla fase di estrazione delle feature visive, che poi vanno a formare il database transazionale dal quale estrarre le regole di associazione. Le caratteristiche estratte e usate nel metodo sono quattro parametri statistici: media, varianza, skewness e curtosi. Questi parametri non vengono estratti dall'intera immagine, ma essa viene suddivisa in quattro riquadri simmetrici e da ognuno di essi vengono estratti i parametri statistici. I quattro riquadri vengono denominati con NW (Nord-Ovest), NE (Nord-Est), SW (Sud-Ovest), SE (Sud-Est). Successivamente si passa alla fase di creazione del database. Il paper propone due modi per creare due database transazionali. Il primo database contiene tutte le feature estratte da tutti i quadranti di tutte le immagini, sia quelle riferite a quadranti contenenti cellule cancerose che quelle riferite a quadranti senza segni di tumore. Il secondo database viene creato in questo modo: se un'immagine mammografica non contiene segni di tumore allora tutte le feature estratte da tutti i quattro quadranti vengono aggiunte al database, mentre se l'immagine contiene un tumore allora solo le feature estratte dal quadrante contenente il tumore vengono aggiunte al database. Dunque, le feature estratte dalle immagini sono gli items sui quali applicare l'algoritmo di associazione. Il metodo usa l'algoritmo Apriori per individuare le regole di associazione tra le feature e per individuare, grazie ad esse, la

categoria alla quale l'immagine appartiene. Le regole di associazione sono vincolate in modo tale che l'antecedente delle regole sia composto da una congiunzione di feature dell'immagine mammografica mentre il conseguente della regola è sempre la categoria alla quale appartiene l'immagine. In altre parole, una regola descrive gli insiemi frequenti di feature per le varie categorie. Il paper descrive due diversi modelli di classificazione basati sulle regole di associazione. Il primo modello si ottiene quando tutte le regole di associazione sono estratte dal database contenente tutte le feature, sia quelle relative a immagini senza noduli che quelle relative ai noduli benigni e maligni. In questo modello regole di associazione rappresentano *de facto* il classificatore. L'input di questo modello è un set di oggetti O_i nella forma $\{cat_i, f_1, f_2, \dots, f_n\}$, dove cat_i è la categoria ("normale", "benigno" o "maligno") di O_i e $f_1, f_2, \dots, f_n$ sono le feature visive dell'oggetto. Poi viene definito un limite inferiore per considerare le regole interessanti e infine si estraggono le regole di associazione nella forma f1 ∧ f2 ∧ ... ∧ fn → cat_i, cioè la presenza delle feature $f_1, f_2, \dots, f_n$ implica che l'immagine appartiene alla categoria i e questo in pratica è il processo di classificazione. Il secondo metodo si basa sull'estrazione di regole di associazione da tre database diversi. Il primo database contiene tutte le feature estratte da immagini mammografiche relative a donne sane, il secondo contiene tutte le feature estratte da immagini mammografiche relative a donne con noduli benigni e infine il terzo contiene tutte le feature estratte da immagini relative a donne con noduli maligni. Dal primo database vengono estratte le regole di associazione per la categoria 1 ("normale"), dal secondo quelle per la categoria 2 ("benigno") e dal terzo quelle per la categoria 3 ("maligno"). Quando si deve analizzare e classificare una nuova immagine mammografica, si estraggono le feature visive e poi si fa una comparazione automatica con tutte le regole di associazione estratte durante la fase di training del modello e la categoria con cui ha il maggior numero di regole di associazione in comune è la categoria alla quale appartiene l'immagine.

- **Riduzione della dimensionalità:** è una tecnica utilizzata quando il numero di variabili (dimensioni) di un certo dataset è troppo alto. Il numero di dimensioni viene ridotto per rendere i dati più facilmente elaborabili e per rendere gli algoritmi di apprendimento più performanti. I vantaggi ottenuti con la riduzione della dimensionalità sono molteplici. Per prima cosa, si riducono il tempo e la potenza computazionale necessari per elaborare i dati e si migliora la performance. Gli algoritmi di machine learning che impiegano troppe feature,

infatti, possono risultare estremamente lenti a tal punto da rendere la loro applicazione di fatto impossibile. Inoltre, in un dataset in cui gli elementi sono descritti con molte variabili, e cioè un dataset i cui elementi sono rappresentabili come punti in uno spazio con molte dimensioni, i vari elementi sono solitamente molto distanti gli uni dagli altri. Come conseguenza di ciò, gli algoritmi fanno fatica ad allenarsi efficientemente con dati di questo tipo e la performance non è ottimale. In ambito machine learning questo problema è noto come "curse of dimensionality", cioè maledizione della dimensionalità. La riduzione della dimensionalità serve anche a ridurre il problema dell'overfitting, perché se gli elementi di un dataset fossero caratterizzati da troppe variabili allora un modello allenato su un dataset di questo tipo potrebbe risultare troppo complesso e troppo sensibile ai dati di training. La riduzione della dimensionalità è molto utile, ovviamente, anche per la visualizzazione dei dati. Infatti, se il numero di dimensioni è ridotto a due o tre è possibile visualizzare gli elementi del dataset come punti in uno spazio 2D o 3D. La riduzione della dimensionalità riduce anche il problema della multicollinearità tra le varie feature, che è un problema che si verifica quando una variabile è altamente correlata ad una o più delle altre variabili. La riduzione della dimensionalità può essere usata per ottenere meno variabili ma assolutamente indipendenti le une dalle altre. La riduzione della dimensionalità è molto utile anche per la l'analisi dei fattori. Un fattore è una variabile latente che non è direttamente presente ma che è composta, cioè dipende, da molteplici variabili tra quelle presenti. La tecnica della riduzione della dimensionalità può anche essere usata per ridurre il rumore nei dati, ovvero per eliminare quelle variabili ridondanti e superflue che non forniscono informazioni importanti. Per quanto riguarda le immagini digitali, infine, la riduzione della dimensionalità è utilizzata per comprimere le immagini, cioè per minimizzare la dimensione in bytes di un'immagine e al tempo stesso preservare, per quanto possibile, la sua qualità. I pixel di un'immagine possono essere considerati come le dimensioni (variabili) di un elemento (immagine). Esistono numerosi metodi di riduzione della dimensionalità che si dividono in due grandi gruppi. Un gruppo è formato da tutti quei metodi che mantengono soltanto le variabili più importanti ed eliminano quelle ridondanti e prive di significato, ma senza applicare nessuna trasformazione al set di variabili. Alcuni esempi di questo tipo sono la Backward elimination e Forward selection. I metodi del secondo gruppo, invece, applicano un processo di trasformazione alle variabili per ottenere delle nuove feature che non sono effettivamente presenti nel dataset. Questi metodi, a sua

volta, si suddividono in lineari e non lineari. I metodi lineari più diffusi sono la Principal Component Analysis (PCA), Factor Analysis (FA), Linear Discriminant Analysis (LDA) e Truncated Singular Value Decomposition (SVD). Invece, alcuni esempi di metodi non lineari sono la Kernel PCA, t-distributed Stochastic Neighbor Embedding (t-SNE), Multidimensional Scaling (MDS) e Isometric Mapping (Isomap).

L'apprendimento supervisionato, anche conosciuto come Supervised Machine Learning, si avvale dell'uso di dataset formati da elementi etichettati, cioè ad ogni elemento corrisponde una categoria ben precisa. Le tecniche di apprendimento supervisionato sono utilizzate prevalentemente per classificare dati o per predire qualcosa con una certa accuratezza. L'accuratezza e le altre misure di performance dipendono da vari fattori, come:

- **Algoritmo implementato:** esistono molteplici algoritmi di apprendimento supervisionato e ognuno di essi presenta vantaggi e svantaggi. E' necessario scegliere con cura l'algoritmo da implementare in base al tipo di dati sui quali si deve operare e all'obiettivo che si vuole raggiungere. Bisogna anche valutare i costi e i benefici in termini di risultati ottenuti a fronte del tempo e della potenza computazionale necessari.
- **Qualità dei dati di input:** i dati etichettati che vengono usati come input del processo possono contenere errori umani o contenere dei bias, dunque come conseguenza il modello predittivo creato utilizzando l'algoritmo di apprendimento potrebbe fornire dei risultati distorti.
- **Qualità dei dati sui quali viene applicato il modello:** i dati utilizzati durante la fase di testing del modello e i dati sui quali il modello viene applicato successivamente devono essere simili ai dati utilizzati come input di training del modello, cioè devono essere descritti con le stesse variabili dei dati di training.

Gli algoritmi di apprendimento supervisionato sono usati prevalentemente per due scopi:

- **Classificazione:** un modello di classificazione ha come obiettivo quello di assegnare un'etichetta, cioè una categoria, agli elementi sui quali viene applicato il modello. Gli elementi sono descritti da diverse variabili e il modello valuta tutte queste variabili per stabilire l'etichetta. La classificazione, essendo l'argomento primario di questo progetto di ricerca, varrà trattata più in dettaglio di seguito.

- **Regressione:** gli algoritmi di regressione sono usati per identificare le relazioni tra variabili dipendenti e indipendenti. La regressione è molto utilizzata in ambito business, ad esempio per fare proiezioni delle vendite future di un'azienda. Gli algoritmi di regressione più popolari sono la regressione lineare, la regressione logistica e la regressione polinomiale. La regressione lineare è usata per identificare la relazione tra una variabile dipendente con una o più variabili indipendenti. Quando c'è solamente una variabile indipendente e una variabile dipendente si parla di regressione lineare semplice, mentre se il numero di variabili indipendenti è maggiore di uno si parla di regressione lineare multipla. Per ciascun tipo di regressione lineare, l'obiettivo è quello di riuscire a tracciare una linea retta, chiamata linea di regressione o di best fit, attraverso il metodo dei minimi quadrati. La regressione logistica si usa quando le variabili dipendenti non sono continue ma categoriche binarie, cioè queste variabili dipendenti possono assumere solamente due valori. La regressione logistica è usata principalmente per risolvere problemi di classificazione binaria. In pratica, l'obiettivo del modello è quello di calcolare la probabilità con cui la variabile binaria assume un valore piuttosto che l'altro.

Gli algoritmi di apprendimento supervisionato al giorno d'oggi sono utilizzati per moltissimi scopi e hanno numerose applicazioni in ambito business. Alcuni degli usi principali sono:

- **Object recognition:** gli algoritmi di questo tipo possono essere usati per localizzare, isolare e identificare gli oggetti presenti in un'immagine digitale o in un video, e in generale la computer vision è forse il campo di applicazione più importante per il futuro e su cui si sta investendo di più.
- **Previsioni:** gli algoritmi possono essere usati per fare previsioni di ogni tipo, ad esempio possono essere usati per stimare la domanda futura di un certo bene o servizio, aiutando quindi le imprese a prendere le giuste decisioni.
- **Analisi del Sentiment:** tramite l'implementazione di alcuni algoritmi di supervised machine learning è possibile estrarre importanti informazioni da grossi volumi di dati con l'intervento umano ridotto al minimo. L'applicazione più classica in questi casi è l'estrazione del feedback generale su un prodotto o servizio a partire da una grande quantità di feedback, ad esempio recensioni, lasciati da molti utenti.
- **Rilevamento di Spam e frodi:** gli algoritmi di apprendimento supervisionato sono molto utili nella lotta contro lo spamming e contro le frodi. Ad esempio, si potrebbe usare un grande

database di email, alcune etichettate come “normali” e altre etichettate come “spam”, come set di training di un modello di classificazione e successivamente utilizzare tale modello per filtrare le email in arrivo.

Creare un modello di classificazione significa applicare un algoritmo ad una grande quantità di dati etichettati con lo scopo di allenare il modello a distinguere un elemento con una certa etichetta da un altro elemento con un’etichetta diversa in base alle variabili dei due elementi, in modo da poter poi essere applicato per classificare (etichettare) dei nuovi dati. In poche parole, un modello di classificazione prende delle decisioni, cioè assegna una categoria, per analogia in base alle esperienze precedenti, ossia in base ai dati utilizzati per il training. Ogni elemento del dataset di training è descritto da numerose variabili, chiamate variabili predittive, che vengono prese in considerazione durante la fase di allenamento del modello. Successivamente il modello di classificazione analizza le variabili del nuovo elemento a cui deve assegnare un’etichetta e prende una decisione in base ad esse.
Gli algoritmi di classificazione sono, al giorno d’oggi, estremamente importanti ed utilizzati in svariati campi. Ad esempio, si pensi ad una società che eroga prestiti e che etichetta ogni cliente con un livello di rischio che può essere “basso”, “medio”, “alto”, dunque le etichette possibili sono tre. Tale società potrebbe utilizzare un modello di classificazione per definire il livello di rischio relativo ad un nuovo cliente, allenando il modello con i dati (reddito, età, ecc) dei clienti passati e le relative etichette (livello di rischio). Bisogna però prestare grande attenzione ai dati che vengono usati come variabili predittive durante la fase di training di un modello di classificazione, perché il modello risultante potrebbe avere dei bias. Si supponga che una grossa azienda informatica, dove lavorano migliaia di ingegneri prevalentemente giovani uomini, voglia sviluppare un modello di classificazione binaria per assegnare un’etichetta (Idoneo, Non-idoneo) ai nuovi candidati per le posizioni lavorative e che quest’azienda utilizzi come dataset di training del modello i dipendenti attuali e i loro dati come variabili predittive, compresi l’età e il sesso. Il modello di classificazione risultante potrebbe essere distorto, ossia potrebbe aver appreso che un candidato per essere idoneo deve essere un giovane uomo, discriminando così verso le donne o le persone più grandi. I principali algoritmi per la creazione di modelli di classificazione sono:

- **Naive Bayes:** i classificatori bayesiani sono una famiglia di classificatori probabilistici basati sul teorema di Bayes. Un classificatore probabilistico è un modello di classificazione che è in

grado di prevedere, dopo numerose osservazioni di dati di input di training, la distribuzione probabilistica di un set di variabili, invece che limitarsi semplicemente ad assegnare un'etichetta. Il teorema di Bayes, che prende il nome da Thomas Bayes, esprime la probabilità di un evento basandosi sulla conoscenza pregressa delle condizioni che causano l'evento. Esistono tre tipi di classificatori bayesiani: Naive Bayes multinomiale, Bernoulli Naive Bayes e Naive Bayes gaussiano. L'ipotesi fondamentale dei classificatori Naive Bayes è che ogni feature fornisce un contributo uguale e indipendente all'outcome finale. Dunque, si assume che ogni variabile sia indipendente dalle altre e che ogni variabile abbia lo stesso peso, ovvero la stessa importanza. Questi due elementi, come conseguenza, limitano i possibili utilizzi dei metodi Naive Bayes. Nonostante ciò, questi algoritmi presentano anche numerosi vantaggi, ad esempio sono estremamente veloci e quindi possono essere usati per fare previsioni quasi in tempo reale e richiedono, a parità di performance, meno dati di training rispetto ad altri metodi. Uno degli utilizzi più comuni è nel text mining e nella classificazione di testo.

- **Support Vector Machines:** è un metodo di apprendimento supervisionato molto popolare, sviluppato dal matematico russo Vladimir Vapnik e dai suoi colleghi presso i laboratori di AT&T Bell. E' usato sia per la classificazione che per la regressione. SVM si basa sulla creazione di un iperpiano dove la distanza tra due classi di dati è massimizzata. Questo iperpiano è noto come "confine di decisione", in inglese "decision boundary", e separa gli elementi del dataset in esame in classi (categorie, etichette) a seconda che si trovino da una parte o dall'altra dell'iperpiano. In altre parole, SVM usa un algoritmo che crea una linea o iperpiano, a seconda del numero di variabili che descrivono gli elementi del dataset, che separa gli elementi in classi. Un iperpiano in uno spazio euclideo n-dimensionale è un subset di tale spazio che è piano e con n-1 dimensioni e divide lo spazio in due parti disconnesse. Per riuscire a comprendere meglio il funzionamento di SVM è necessario procedere con un esempio e con una ipotetica rappresentazione visiva. Si supponga di avere un dataset formato da elementi descritti da due variabili e che siano presenti soltanto due etichette, quindi ciascun elemento del dataset appartiene ad una delle due categorie. Questi elementi possono essere rappresentati come punti in un grafico dove ognuno dei due assi è una delle due variabili. In questo caso l'iperpiano necessario per separare tutti gli elementi di una categoria con quelli dell'altra categoria è semplicemente una linea. E' possibile tracciare diverse linee per separare i due gruppi di punti e l'obiettivo dell'algoritmo è quello di trovare

la linea di separazione migliore. La linea migliore è quella che massimizza il margine, che è la somma della distanza del punto della categoria 1 più vicino alla linea dalla linea stessa e la distanza del punto della categoria 2 più vicino alla linea dalla linea stessa. La definizione dell'iperpiano è in pratica la fase di training e successivamente il modello risultante può essere utilizzato per classificare nuovi elementi. Continuando con l'esempio precedente, si supponga di voler classificare un nuovo elemento di cui non si conosce l'etichetta. Questo elemento può essere rappresentato come un punto nel grafico precedente in base ai valori delle sue due variabili. Il modello di classificazione assegnerà un'etichetta all'elemento a seconda che esso si trovi da una parte o dall'altra della linea tracciata in precedenza, durante la fase di training. Le applicazioni della metodologia SVM sono le più disparate, dal riconoscimento e classificazione di testo, all'analisi di dati scientifici fino alla classificazione di immagini, ad esempio per il riconoscimento e classificazione di espressioni visive. Delle applicazioni del SVM nel campo della classificazione di immagini si tratterà meglio in seguito.

- **K-Nearest Neighbor:** è un algoritmo non parametrico che classifica gli elementi basandosi sulla loro prossimità e associazione ad altri elementi. Questo algoritmo assume che gli elementi simili, cioè appartenenti alla stessa categoria, possono essere trovati vicini gli uni con gli altri. Per comprendere fino in fondo il funzionamento di questo algoritmo è necessario immaginare gli elementi, descritti con una serie di variabili, come punti in uno spazio geometrico in cui ogni variabile è una dimensione. L'algoritmo calcola la distanza tra i vari elementi del dataset, solitamente con la distanza euclidea, e successivamente assegna un'etichetta basandosi sulla categoria più frequente. In altre parole, l'algoritmo prova a determinare l'etichetta di un elemento considerando l'etichetta degli elementi in sua prossimità. Si pensi ad un dataset usato per il training in cui gli elementi sono suddivisi in due categorie, "A" e "B", e per semplicità si ipotizzi che gli elementi siano descritti da due variabili e che quindi possano essere rappresentati come punti in un grafico bidimensionale, in cui ogni asse è una variabile. Si pensi ad un nuovo elemento di cui non si conosce l'etichetta, e che questo elemento venga rappresentato come un punto nello spazio insieme a tutti gli elementi del dataset di training. L'algoritmo, per assegnare l'etichetta "A" o "B" al nuovo elemento, analizza i punti nelle sue vicinanze e se la maggioranza di essi appartiene alla categoria "A" allora l'algoritmo assegnerà tale etichetta al nuovo elemento, altrimenti assegnerà l'etichetta "B". K-Nearest Neighbor è considerato un algoritmo "pigro", infatti in inglese è definito "lazy learner", perché non crea un vero e proprio modello di classificazione,

ovvero non ha come output della fase di training una funzione discriminante con cui analizzare i nuovi elementi, ma semplicemente memorizza un insieme di elementi del dataset di riferimento e ogni volta confronta i nuovi elementi da classificare con tali elementi. K-NN è un algoritmo molto semplice ma il suo tempo di processamento di un nuovo elemento cresce al crescere del numero di elementi del dataset di training, proprio perché l'algoritmo deve svolgere tanti più calcoli di distanze tanti più sono i punti da usare come riferimento. Gli altri algoritmi di classificazione, quindi, richiedono relativamente tanto tempo per la fase di training per creare il modello (funzione) da applicare ad altri elementi ma poi la fase "predittiva" di nuovi elementi è relativamente rapida. K-NN, al contrario, non ha una vera e propria fase di training perché si tratta semplicemente di memorizzare dei dati da usare come riferimento, ma al contrario la fase di di applicazione per classificare nuovi elementi è relativamente lunga perché, per ogni nuovo elemento, l'algoritmo deve calcolare tutte le distanze con gli elementi di riferimento per trovare gli elementi "vicini" e decidere l'etichetta in base alle etichette dei "vicini". K-NN è molto utilizzato per i motori di raccomandazione e per il riconoscimento di immagini.

- **Random Forest:** è un altro algoritmo di apprendimento supervisionato estremamente diffuso e utilizzato sia per la regressione che per la classificazione. Si tratta di un metodo "ensemble", cioè un metodo d'insieme che utilizza un gran numero di altri elementi per arrivare a fare una previsione. Il metodo Random Forest si basa su un gran numero, deciso dall'utente che implementa tale metodo, di alberi di decisione (in inglese "decision tree") indipendenti. Un albero di decisione è un modello di classificazione, creabile con diversi algoritmi, che arriva ad assegnare un'etichetta attraverso una serie di decisioni di tipo if-then-else. E' chiamato così perché è rappresentabile sotto forma di albero con numerosi nodi di decisione riferiti a variabili del dataset e da cui partono due ramificazioni (nel caso di una variabile numerica) o più ramificazioni (nel caso di una variabile categorica) che a loro volta portano ad altri nodi di decisione, e così via fino ad arrivare alle ramificazioni finali che invece portano ad un'etichetta. Il primo nodo di decisione, ovvero la radice dell'albero, è riferito all'attributo più selettivo per il processo di classificazione. La profondità di un albero è il numero di ramificazioni da attraversare per arrivare del primo nodo di decisione alle foglie dell'albero, cioè alle etichette, più distanti possibili. Solitamente, aumentando la profondità dell'albero aumenta anche la complessità del modello e la sua accuratezza nel predire le etichette di nuovi elementi, ma aumentando troppo la profondità si rischia

l'overfitting, ovvero si rischia di creare un modello troppo sensibile ai dati usati per il training e non efficace nel classificare nuovi elementi. Il Random Forest, come già detto, aggrega le previsioni di molti alberi di decisione per arrivare ad una previsione più accurata. Gli algoritmi di Random Forest sono utili nella gestione di dataset molto grandi, con una dimensionalità elevata e con attributi eterogenei, cioè alcuni numerici e altri categorici, ma presentano anche degli svantaggi. Gli algoritmi di Random Forest, ad esempio, sono di tipo black box, ovvero è molto difficile guardare dentro al modello e comprendere appieno le ragioni dietro le sue decisioni. Inoltre, possono richiedere molto tempo per la fase di training, testing e applicazione.

Classificare e distinguere gli oggetti presenti in immagini è una mansione relativamente facile per gli esseri umani, ma che si è rivelata essere una sfida complessa per i calcolatori e fin da subito sono stati investiti grandi risorse ed energie nel settore della computer vision.

La classificazione di immagini consiste nell'associare alle immagini delle etichette associate a delle classi predefinite. La classificazione può essere di due tipi: binaria e multiclasse. Nella classificazione binaria le etichette possibili sono solamente due, mentre nella classificazione multiclasse ci sono n etichette possibili. I processi di classificazione binaria di immagini sono relativamente meno complessi, perché basta accertarsi che un'immagine non appartenga ad una categoria per assegnarla all'altra categoria. Come già detto, questo processo è relativamente semplice per l'intelletto umano, anche se a volte richiede competenze specifiche, ma può richiedere molto tempo, specie se il numero di immagini da analizzare e classificare è molto alto. Negli ultimi anni, l'utilizzo di tecniche di classificazione di immagini è aumentato enormemente e le sue applicazioni pratiche, come l'image recognition, cioè l'abilità di distinguere e identificare oggetti in un'immagine, sono già le più svariate e l'importanza di queste tecniche continuerà a crescere in futuro. Ad esempio, il mercato dell'Image Recognition è stato di 1,7 miliardi di dollari americani nel 2020 e supererà i 5 miliardi nel 2026, con un tasso CAGR del 24.82% durante il quinquennio 2021-2026. L'Image Recognition può essere applicata in molti settori, tra cui la sicurezza e sorveglianza, scanner biometrici, advertising e soprattutto nell'automotive, in particolare in ambito delle auto a guida autonoma. Un'auto a guida autonoma, infatti, è dotata di sistemi tecnologici in grado di acquisire la visuale dell'intorno del veicolo, identificare e distinguere i vari oggetti presenti nell'ambiente circostante e di classificarli in tempo reale.

Ad oggi, sono disponibili molti algoritmi e metodologie per creare un modello di classificazione di immagini, alcuni dei quali sono già stati in parte trattati in questo capitolo. Recentemente, con l'avvento del Deep Learning, in combinazione con i progressi in ambito hardware e GPU, è stato possibile raggiungere dei livelli di precisione e accuratezza superiori a quelli ottenuti con i metodi tradizionali. Il Deep Learning, o "apprendimento profondo", è una famiglia di metodi di machine learning che si basano sull'implementazione di reti neurali e sul feature learning. Una rete neurale artificiale, in inglese "Artificial Neural Network" o ANN, è un insieme di nodi, chiamati neuroni artificali, interconnessi. Il nome deriva dal fatto che la struttura di una rete neurale artificiale mima la struttura del cervello umano, che è formato da un insieme di neuroni connessi tra di loro. Le reti neurali artificiali hanno una struttura stratificata, ovvero sono composte da diversi strati di nodi, tra cui uno strato di input, uno o più strati nascosti ("hidden layers") e uno strato di output. Ogni nodo, o neurone artificiale, è collegato agli altri nodi dello strato successivo. Ogni nodo o neurone riceve un segnale, lo processa e può eventualmente mandare un segnale ai neuroni ai quali è connesso. Questi "segnali" sono dei numeri reali. I neuroni e le connessioni hanno dei pesi che si modificano e si aggiustano man mano che il processo di apprendimento va avanti. Il peso aumenta o diminuisce il numero reale che attraversa una connessione. I neuroni possono avere un valore di threshold, in modo tale che il neurone manda un segnale di output solo se questo supera il valore di threshold. In caso contrario, il nodo non si attiva e nessun segnale è passato ai nodi successivi. Più in dettaglio, i nodi possono essere considerati come modelli di regressione lineare, composti da dati di input, pesi, un valore di bias o threshold e un output. Dopo aver definito i nodi di input, vengono assegnati dei pesi. Questi pesi aiutano a determinare l'importanza di una data variabile, con quelle più importanti che contribuiscono in maniera più significativa all'output. Tutti i valori di input vengono poi moltiplicati per i rispettivi pesi e poi sommati. Successivamente, il valore ottenuto viene usato come variabile in una funzione di attivazione per ottenere un altro output. Se l'output ottenuto è maggiore del valore di soglia (threshold) allora il nodo in questione viene attivato e l'output viene passato ai nodi ai quali è collegato. Di conseguenza, l'output di un nodo diventa l'input di un altro nodo e così via. Come esempio per capire facilmente il funzionamento di un singolo nodo utilizzando valori binari, si supponga un neurone relativo alla decisione di uno sciatore di andare a sciare in uno specifico luogo di una montagna oppure no (Si:1, No:0). La decisione di andare o non andare è l'outcome da prevedere, indicato con $\hat{y}$. Si supponga che ci siano tre fattori che influenzano la scelta:

1. La neve presente è sufficiente? (Si:1, No:0)
2. Il luogo è facilmente raggiungibile? (Si:1, No:0)

3. Qualche altro sciatore ha avuto un incidente in quel luogo di recente? (Si:0, No:1)

Si supponga che ci siano le seguenti condizioni: $X_1 = 1$ (è presente abbastanza neve), $X_2 = 0$ (il luogo è difficile da raggiungere) e $X_3 = 1$ (nessuno sciatore ha avuto un incidente in quel preciso luogo). Adesso è necessario assegnare dei pesi per determinare l'importanza di queste tre variabili. Maggiore è il peso e maggiore è l'importanza della variabile. Si supponga di utilizzare i seguenti pesi: $W_1 = 3$, perché è raro trovare un luogo con sufficiente neve, $W_2 = 2$, perchè è meno importante la comodità nello raggiungere il luogo rispetto alla quantità di neve, $W_3 = 4$, perché la sicurezza è molto importante. Come valori di threshold e bias vengono scelti rispettivamente 3 e -3. Di conseguenza, la formula per calcolare ŷ è $\hat{y} = (3 * 1) + (2 * 0) + (4 * 1) - 3 = 4$. Il risultato è maggiore del valore di soglia, quindi il nodo si attiva e l'informazione "lo sciatore va a sciare in quel luogo", cioè il valore 1, viene trasmesso agli altri nodi connessi, che useranno questo valore come una delle variabili delle loro rispettive funzioni. Come si può notare dall'esempio, basta modificare i valori dei pesi o del valore di soglia e l'output cambia totalmente. Nell'esempio considerato, le variabili erano binarie (Si/No, 1/0) e un neurone che opera con valori di questo tipo è chiamato percettrone, ma le moderne reti neurali usano i neuroni sigmoidi, che operano su valori compresi tra 0 e 1, aumentando la complessità totale del sistema. I valori dei pesi non vengono definiti in modo casuale, ma vengono assegnati e modificati durante la fase di training della rete neurale. Nel caso di apprendimento supervisionato, ossia tramite l'utilizzo di elementi etichettati per il training, durante la fase di allenamento viene calcolato il valore di una funzione di costo che deve essere minimizzata, chiamata funzione MSE (Mean Squared Error) o errore quadratico medio. La funzione è $MSE = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y} - y)^2$, dove m è il numero di campioni, ŷ il risultato atteso e y è il valore effettivo. L'obiettivo è quello di minimizzare il risultato di questa equazione e per fare ciò il sistema modifica e aggiusta i vari pesi e bias durante la fase di training per avvicinarsi il più possibile al punto di minimo locale della funzione. La logica e la matematica dietro le reti neurali sono in realtà molto datate, anche se solo di recente è stato possibile sviluppare e applicare sistemi di questo tipo su larga scala, grazie al miglioramento delle tecnologie disponibili. Frank Rosenblatt è considerato il padre del deep learning e delle reti neurali ed è stato il primo, nel lontano 1958, a ideare e sviluppare il funzionamento di un percettrone (Rosenblatt, 1958).

Esistono diversi tipi di reti neurali e in base all'ambito di applicazione e all'obiettivo che si vuole raggiungere è meglio implementare un modello piuttosto che un altro. I modelli di reti neurali più popolari sono:

- **Feed-Forward Artificial Neural Network:** è il modello più semplice ed intuitivo ed è quello che è stato spiegato e usato come esempio fino ad ora, il nome deriva dal fatto che le informazioni, cioè i valori, si muovono in un'unica direzione che va dai nodi dello strato di input ai nodi dello strato di output. Le reti Feed-Forward sono formate da uno strato di input, uno o più strati nascosti e uno strato di output. Le reti neurali sono in grado di imparare qualsiasi funzione non lineare e perciò sono anche chiamate "Universal Function Approximators", ovvero approssimatore universali di funzioni. Le reti neurali, come già detto, imparano i vari pesi per mappare gli input in output durante la fase di training. Il motivo principale dietro l'approssimazione universale è proprio la presenza della funzione di attivazione. Le funzioni di attivazione introducono proprietà non lineari nella rete e questo aiuta la rete a individuare la relazione, anche se complessa, tra gli input e l'output. I modelli di questo tipo possono essere applicati sia su dati tabulari che immagini, ma nel secondo caso sono presenti diversi svantaggi. Ad esempio, è necessario convertire le immagini, che sono bidimensionali, in dei vettori unidimensionali prima di iniziare la fase di training e ovviamente la lunghezza di questi vettori cresce drasticamente al crescere delle dimensioni delle immagini. Ciascun valore di questi vettori è una variabile che la rete neurale considera durante il training. Si pensi a delle immagini di dimensioni 224 x 224, i vettori risultanti dopo la scomposizione hanno una lunghezza di 602.112, il che significa che il modello deve tenere in considerazione 602.112 parametri durante la fase di training. Inoltre, trasformando un'immagine in un vettore si perdono tutte le informazioni spaziali. Un altro problema comune a questo tipo di reti è, in inglese, quello del "Vanishing and Exploding Gradient", ovvero gradiente evanescente ed esplosivo. Questo problema è associato all'algoritmo di retropropagazione, che aggiorna i pesi dei neuroni con un meccanismo bottom-up, e se la rete neurale è troppo profonda, cioè presenta un gran numero di strati nascosti, porta il gradiente a ridursi fino a quasi annullarsi o ad aumentare drasticamente. Inoltre, i modelli di questo tipo non riescono a catturare le informazioni sequenziali, il che è fondamentale se si opera su dati sequenziali come le serie storiche o i dati testuali.
- **Recurrent Neural Networks:** il nome deriva dal fatto che i nodi degli strati nascosti presentano delle ricorrenze, ovvero delle connessioni con se stessi. Questi loop permettono di catturare le informazioni sequenziali come input. Questo tipo di reti è, di conseguenza, ideale per i dati sequenziali come le serie storiche, dati testuali o dati audio. Catturare le informazioni sequenziali significa, per esempio durante l'analisi di un testo, catturare la

dipendenza che c'è tra una singola parola e le altre parole della stessa frase, ovvero di considerare una parola anche in funzione delle altre parole della frase. I neuroni che compongono questo tipo di reti, chiamati neuroni ricorrenti, applicano una formula che considera sia il nuovo input che il neurone sta ricevendo ma anche lo stato precedente. Lo stato attuale si ottiene in funzione dello stato precedente e dell'input attuale, cioè $h_t = f(h_{t-1}, x_t)$, in cui h_t è il nuovo stato, h_{t-1} è lo stato precedente e x_t è il nuovo input. Come si evince dalla formula, viene tenuto in considerazione lo stato precedente e non direttamente l'input precedente, poiché questo è stato trasformato dalla funzione specifica del neurone. La funzione è quindi iterativa e ogni iterazione prende il nome di time step. Si supponga che un neurone ricorrente abbia come funzione di attivazione la funzione $\tanh(x)$, come peso dello stato precedente W_{hh} e come peso dell'input corrente W_{xh}. Si supponga che la funzione dello stato attuale sia $h_t = \tanh(W_{hh} * h_{t-1} + W_{xh} * x_t)$. Questa funzione viene iterata tante volte quanto richiesto dal problema, ad esempio se si sta considerando una frase il numero di iterazioni è il numero di parole che compongono la frase. Quando viene calcolato lo stato finale, cioè quando finiscono le iterazioni, è possibile poi calcolare l'output finale con la formula $y_t = W_{hy} * h_t$. Uno dei vantaggi delle reti neurali ricorrenti è la condivisione dei parametri tra diversi time step e di conseguenza si ha un numero minore di parametri da definire durante la fase di training, con conseguente riduzione della capacità computazionale richiesta. Sono presenti, però, anche diversi svantaggi. Ad esempio, le reti neurali ricorrenti profonde, cioè con un gran numero di iterazioni, soffrono del problema del gradiente evanescente ed esplosivo, già descritto in precedenza.

- **Convolutional Neural Network:** le reti neurali convoluzionali sono quelle di maggior interesse per questo ambito di ricerca, cioè la classificazione di immagini, perché vengono utilizzate principalmente per processare immagini e video, anche se è possibile ottenere ottimi risultati anche con i dati sequenziali. Una rete neurale convoluzionale è un tipo di rete neurale che usa dei filtri per estrarre caretteristiche dalle immagini e lo fa in una maniera tale che anche l'informazione spaziale è mantenuta. Il nome deriva dal fatto che vengono usate delle convoluzioni. Una convoluzione è un'operazione matematica applicata ad una matrice. E' possibile rappresentare un'immagine digitale in bianco e nero di dimensioni $n\ x\ m$ con una matrice di dimensioni $n\ x\ m$ in cui ogni cella rappresenta uno specifico pixel dell'immagine il quale si trova nella medesima posizione. Il valore numerico presente in

ciascuna cella, invece, è un numero intero compreso tra [0,255] e rappresenta il livello di intensità di grigio del pixel, poiché si è detto che si tratta di un'immagine in bianco e nero. Se fosse stata un'immagine a colori, invece, sarebbe stato possibile rappresentarla come una matrice con il modello a colori RGB. In tal caso, la matrice avrebbe avuto dimensioni $n\ x\ m\ x\ 3$ perché sarebbe stati presenti i tre strati di colori rosso, verde e blu. Questo modo di rappresentare le immagini come matrici sarà approfondito nel capitolo successivo. La convoluzione, quindi, è applicata alle matrici che rappresentano delle immagini per estrarre le caratteristiche visive. I filtri delle reti neurali convoluzionali sono chiamati "kernels", e i kernel sono usati per estrarre le caratteristiche visive dall'immagine tramite le operazioni convoluzionali. Le reti di questo tipo "imparano" automaticamente i filtri da applicare durante la fase di training, di conseguenza imparano ad estrarre le caratteristiche visive più importanti e determinanti durante la fase di training. Come si è già detto, le reti di questo tipo riescono anche a mantenere le informazioni spaziali di un'immagine. Le informazioni spaziali sono il modo come i pixel sono disposti in un'immagine e la relazione che c'è tra di essi. Mantenere queste informazioni è fondamentale per riuscire ad identificare un oggetto in maniera accurata, per identificare la sua posizione nell'immagine e la sua relazione con altri oggetti. Nelle reti neurali convoluzionali è anche presente il concetto di condivisione dei parametri, proprio come nelle reti ricorrenti. Un singolo filtro o kernel, infatti, è applicato a differenti parti dell'immagine per estrarre le feature. Le reti di questo tipo riescono a ridurre le immagini in una forma più piccola e facile da processare, senza però perdere le feature visive fondamentali. Questo è molto utile nel caso in cui si abbia a che fare con immagini ad altissima risoluzione, ad esempio 8K (7680 x 4320). Processare immagini di questo tipo con altri metodi di classificazione richiederebbe troppo potenza computazionale e tempo. Per capire meglio il funzionamento, si supponga di avere un'immagine a colori di dimensioni 6x6. Questa può essere rappresentata, con il modello RGB, come una matrice di dimensioni 6x6x3. Per semplicità, si consideri per il momento soltanto uno strato di colore, ad esempio il verde. Si ha quindi una matrice di dimensioni 6x6x1 e su questa viene applicato il filtro. Bisogna prima decidere le dimensioni del filtro, cosa che viene fatta durante la fase di creazione del modello. Si supponga di scegliere un filtro K di dimensioni 3x3. Si supponga anche che lo Stride Value, che indica il modo in cui si sposta il filtro sull'immagine, sia 1. I valori di questo filtro vengono utilizzati per fare delle operazioni di moltiplicazione matriciale con delle porzioni di dimensioni 3x3 della matrice di volta in volta diverse. Il filtro, quindi,

viene applicato prima sulla porzione 3x3 (righe 1->3, colonne 1->3) in alto a sinistra della matrice poi si sposta a destra di 1, poiché si è detto che lo Stride Value è 1, su un'altra porzione 3x3 (righe 1->3, colonne 2->4) e viene ripetuta la moltiplicazione e così via fino alla porzione di matrice in alto a destra (righe 1->3, colonne 4->6). Successivamente il filtro scende di una riga, poiché lo Stride Value è 1, e si ripete l'operazione sulla porzione di matrice che va dalle righe 2 a 4 e dalle colonne 1 a 3, e così via fino ad applicare il filtro all'intera matrice, per un totale di 16 applicazioni. Di volta in volta il risultato dell'operazione di moltiplicazione matriciale viene salvato in una matrice in modo tale che il risultato ottenuto con la prima applicazione (righe 1->3, colonne 1->3) sia in posizione (1,1), il risultato ottenuto con la seconda (righe 1->3, colonne 2->4) sia in posizione (1,2) e così via. La matrice dei risultati avrà quindi dimensioni 4x4. In questo caso, per semplicità, si è considerato solo uno strato di colore ma il filtro K ha la stessa profondità della matrice alla quale deve essere applicato, quindi ha dimensioni 3x3x3. In particolare, lo strato i-esimo del filtro viene applicato allo strato i-esimo della matrice che rappresenta l'immagine, quindi per ogni porzione di immagine viene moltiplicato il primo strato del filtro per il primo strato di colore, il secondo strato del filtro per il secondo strato di colore e il terzo strato del filtro per il terzo strato di colore, per un totale di tre risultati diversi. I tre risultati di ogni porzione vengono sommati tra di loro e con un bias e il risultato finale viene salvato in una matrice dei risultati. Questa matrice dei risultati altro non è che la matrice delle caratteristiche visive estratte dall'immagine, che in inglese prende il nome di "convolved feature", e ogni valore presente nella matrice è una feature visiva dell'immagine. Nel caso di immagini troppo grandi le convolved feature potrebbero comunque essere troppo grandi per essere gestite in modo efficiente, cioè senza richiedere troppo tempo o potenza computazionale, e la fase successiva del processo, ovvero il pooling, aiuta a ridurre questo problema. La fase di pooling è un processo di riduzione della dimensionalità delle caratteristiche estratte e per riuscire a individuare le caratteristiche dominanti. Esistono due tipi di pooling: il Max Pooling e l'Average Pooling. Il Max Pooling, tramite un processo simile a quello descritto prima, considera porzioni della matrice delle caratteristiche di volta in volta diverse, seleziona il valore massimo presente e lo salva in un'altra matrice. Come esempio, si consideri una matrice convolved feature di dimensioni 5x5 e si applichi un filtro di pooling di dimensioni 3x3. Come primo step si considera la porzione della matrice convolved feature in alto a sinistra (righe 1->3, colonne1->3) , si seleziona il valore massimo e si salva in una nuova

matrice in posizione (1,1) e si applica il filtro di pooling ad un'altra porzione di matrice. Alla fine si ottiene una matrice di dimensioni 3x3 con i valori massimi delle varie porzioni di matrice convolved feature. L'Average Pooling funziona alla stessa maniera ma invece che selezionare il valore massimo di una porzione di matrice si calcola la media di tutti i valori della porzione. Il Max Pooling è preferibile rispetto all'Average Pooling perché oltre a ridurre la dimensionalità della matrice convolved feature riduce anche il rumore. L'output ottenuto dopo il processo di pooling viene successivamente usato come input di una rete neurale con l'obiettivo di classificare l'immagine. La matrice, quindi, viene convertita in un vettore e usata come input di una rete neurale Feed-Forward. Solitamente si usano i fully-connected layer, ovvero strati completamente connessi. Un fully-connected layer è uno strato in cui ogni neurone è connesso a tutti i neuroni dello strato successivo.Durante la fase di training,tutto il processo appena descritto (applicazione dei filtri kernel, pooling, input di una rete neurale) viene ripetuto su tutte le immagini del dataset e durante il training il modello riesce ad apprendere quali sono le feature visive più importanti e determinanti per classificare le immagini, utilizzando la tecnica della classificazione Softmax. La tecnica prende il nome dalla funzione softmax, che viene appunto implementata per arrotondare i valori delle varie classi in valori positivi normalizzati, in modo tale che la perdita di entropia incrociata può essere applicata. Le reti neurali convoluzionali, come già detto, sono ad oggi il campo di ricerca nella classificazione di immagini più in forte espansione, perché tramite la loro implementazione è possibile ottenere ottimi risultati e sono presenti anche altri vantaggi. Ad esempio, la necessità di applicare un processo di preprocessing è molto minore rispetto ad altre metodologie di classificazione di immagini. Le reti neurali convoluzionali sono state ideate su modello del cervello umano e in particolare della corteccia visiva.

I singoli neuroni umani, infatti, rispondono agli stimoli solo in una regione ristretta del campo visivo nota come campo ricettivo. Una raccolta di tali campi si sovrappone per coprire l'intera area visiva. Nella tabella (Figura 8) della pagina successiva sono riassunte tutte le principali caratteristiche dei tre tipi di reti neurali illustrate fino ad ora.

	Feed Forward	RNN	CNN
Ideale per	Dati tabulari	Dati sequenziali	Immagini
Connessioni ricorrenti	No	Si	No
Condivisione dei parametri	No	Si	Si
Informazioni spaziali	No	No	Si
Vanishing and exploding gradient	Si	Si	Si

Figura 8 – Sintesi delle caratteristiche dei vari tipi di reti neurali

Ad oggi, le reti neurali artificiali, ed in particolare le reti convoluzionali, rappresentano lo stato dell'arte per la classificazione di immagini. Ciò significa che sono i metodi che forniscono i risultati migliori e su cui si sta investendo maggiormente.

Capitolo 4. Metodologia proposta

La metodologia proposta in questo progetto di tesi è frutto di un intenso lavoro di ricerca teorica e di sviluppo software. Il software è stato sviluppato in ambiente Matlab con l'aggiunta dei seguenti toolbox:

- **Deep Learning Toolbox**: fornisce una serie di strumenti e funzioni per la progettazione e l'implementazione di reti neurali profonde con algoritmi, modelli pre-addestrati e app. Consente di utilizzare reti neurali convoluzionali (ConvNet, CNN) e reti Long Short-Term Memory (LSTM) per implementare la classificazione e la regressione su immagini, serie storiche e dati testuali. Consente inoltre di progettare architetture di rete come reti generative avversarie (GAN) e reti siamesi utilizzando la differenziazione automatica, cicli di addestramento personalizzati e pesi condivisi. Con l'app Deep Network Designer è possibile progettare, analizzare e addestrare reti in forma grafica. L'app Experiment Manager aiuta a gestire più esperimenti di deep learning, a tenere traccia dei parametri di training, ad analizzare i risultati e a confrontare il codice di diverse implementazioni.
- **Wavelet Toolbox**: mette a disposizione degli utenti applicazioni e funzioni per l'analisi e la sintesi di segnali e immagini. Consente di rilevare eventi quali anomalie, punti di cambiamento e transitori e rimuovere il rumore dai dati. Wavelet e altre tecniche multiscala possono essere utilizzate per analizzare i dati a diverse risoluzioni di tempo e frequenza e scomporre segnali e immagini in diverse componenti. Le tecniche Wavelet possono essere

utilizzate per ridurre la dimensionalità ed estrarre caratteristiche discriminanti da segnali e immagini per la fase di training di modelli di machine learning e deep learning. Questo toolbox inoltre permette di rimuovere in modo interattivo il rumore dai segnali, eseguire analisi multirisoluzione e wavelet e generare codice MATLAB. Il toolbox include algoritmi per l'analisi wavelet continua e discreta, l'analisi tramite pacchetti wavelet, l'analisi multirisoluzione, il wavelet scattering e altre analisi multiscala.

- **Signal processing Toolbox**: fornisce all'utente funzioni e applicazioni per l'analisi, la pre-elaborazione e l'estrazione di caratteristiche da segnali campionati in maniera uniforme o non uniforme. Il toolbox include strumenti per progettare e analizzare dei filtri, per ricampionare, per linearizzare, per rimuovere il trend e per stimare lo spettro di potenza. Mette a disposizione dell'utente anche funzioni per l'estrazione di caratteristiche specifiche come changepoint e inviluppi, la quantificazione delle somiglianze nei segnali e l'esecuzione di misurazioni come SNR e distorsione. Con la Signal Analyzer App è possibile pre-elaborare e analizzare multipli segnali simultaneamente nel dominio del tempo, della frequenza e tempo-frequenza.
- **Image processing Toolbox**: fornisce un set completo di algoritmi standard di riferimento e app per lavorare con l'elaborazione delle immagini, l'analisi, la visualizzazione e lo sviluppo di algoritmi. È possibile effettuare la segmentazione, correzione e registrazione di immagini, ridurre il rumore, eseguire trasformazioni geometriche e l'elaborazione di immagini in 3D. Le app di Image Processing Toolbox consentono di automatizzare comuni flussi di lavoro per l'elaborazione di immagini. È possibile segmentare interattivamente dati immagine, comparare tecniche di registrazione immagini ed elaborare in batch grandi set di dati. Le funzioni e le app di visualizzazione consentono di esplorare immagini, volumi in 3D e video, regolare il contrasto, creare istogrammi e manipolare le regioni d'interesse (ROI).
- **Statistics and Machine Learning Toolbox**: fornisce funzioni e strumenti per la descrizione, l'analisi e la modellazione dei dati. Si possono utilizzare statistiche descrittive, visualizzare i dati e raggrupparli, eseguire l'analisi esplorativa dei dati, adattare le distribuzioni di probabilità ai dati, generare numeri casuali per le simulazioni Monte Carlo ed eseguire test di verifica d'ipotesi. Vengono forniti degli algoritmi di regressione e classificazione per costruire modelli predittivi sia interattivamente, usando la Classification and Regression learner app, che programmaticamente, usando autoML. Per l'analisi ed estrazione di caratteristiche di dati multidimensionali il toolbox consente di effettuare la PCA, di

regolarizzare, di ridurre la dimensionalità e consente anche di identificare le variabili con il miglior potere predittivo.

- **Computer Vision Toolbox**: fornisce algoritmi, funzioni e app per la progettazione e il test di sistemi di visione artificiale, visione 3D e di elaborazione video. È possibile eseguire rilevamento e tracking di oggetti, nonché rilevamento, estrazione e confronto di feature.

In Matlab, un'immagine di dimensioni $n\ x\ m$ pixels viene trattata come una matrice e le dimensioni della matrice dipendono dal tipo di immagine, se a colori o a scala di grigi. Un'immagine a scala di grigi, o grayscale in inglese, è un'immagine in bianco e nero in cui ogni pixel è dotato di uno specifico livello di intensità di grigio nel range [0,255] , ovvero dal nero al bianco. Un'immagine di questo tipo viene elaborata in Matlab con una matrice di dimensioni $n\ x\ m$ dove ciascuna cella rappresenta un pixel dell'immagine ed il valore presente nella cella è il livello di intensità di grigio del pixel.
I colori di un'immagine vengono trattati in Matlab con il modello RGB. Il modello di colori RGB è un modello additivo in cui i colori primari rosso (R-red), verde (G-green), e blu (B-blue) sono sovrapposti per riprodurre un'ampia gamma di colori. La maggior parte dei dispositivi elettronici gestisce i colori con questo modello, anche se è un modello "device-dependent", nel senso che dispositivi diversi visualizzano e riproducono gli stessi valori di RGB in modo diverso tra di loro.
Un'immagine a colori viene salvata in Matlab con una matrice di dimensioni $n\ x\ m\ x\ 3$, dove n e m sono le dimensioni dell'immagine e 3 è il numero di strati di colore, uno per il rosso, uno per il verde e uno per il blu. Dunque, il pixel che nell'immagine si trova in posizione (x, y) viene rappresentato nella matrice con un trio di valori e in posizione $(x, y, 1)$ è presente il valore di intensità di rosso del pixel, in $(x, y, 2)$ il valore di intensità di verde e in $(x, y, 3)$ quello di intensità di blu. Tutti e tre i valori sono nel range [0,255].

Il dataset utilizzato in questo progetto di ricerca è composto da 10.000 immagini istopatologiche a colori di tessuto del colon, di cui 5000 etichettate con "colonaca", ossia con adenocarcinoma, e 5000 etichettate con "colonn", ossia senza adenocarcinoma. Tutte le immagini sono hanno una dimensione di 768x768 e sono in formato jpeg. La Figura 9 mostra due esempi di immagini presenti nel dataset. Nell'immagine a sinistra non è presente l'adenocarcinoma mentre in quella a destra è presente.

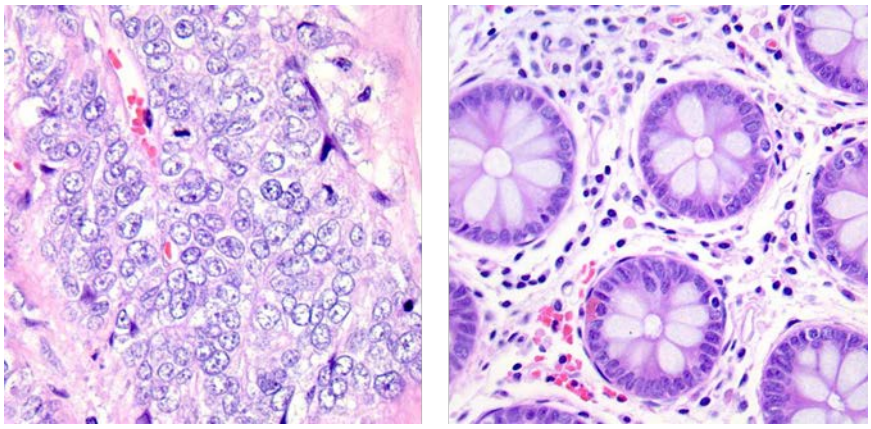

Figura 9 – Esempio di due immagini presenti nel dataset

L'obiettivo principale di questo progetto era quello di creare, allenare e testare un modello classificatore in grado di classificare immagini istopatologiche di tessuto del colon in due categorie: adenocarcinoma o non-adenocarcinoma. I risultati finali sono stati eccellenti e hanno superato la performance dei principali metodi di classificazione binaria di immagini di questo ambito. La notazione usata per valutare la performance del modello è la seguente:

- Positive: "colonaca", adenocarcinoma
- Negative: "colonn", non-adenocarcinoma
- P: numero di elementi che realmente sono Positive
- N: numero di elementi che realmente sono Negative
- True positive TP: numero di immagini predette come "colonaca" che effettivamente lo sono
- True negative TN: numero di immagini predette come "colonn" che effettivamente lo sono
- False positive FP: numero di immagini predette come "colonaca" che in realtà sono "colonn"
- False negative FN: numero di immagini predette come "colonn" che in realtà sono "colonaca"

I risultati finale sono stati valutati in termini di:

- **Accuracy**: è la percentuale di elementi predetti correttamente rispetto al totale degli elementi

$$\frac{TP+TN}{P+N}$$

- **Specificity:** è la percentuale di elementi TN rispetto al totale di elementi N

$$\frac{TN}{N}$$

- **Precision:** è la percentuale di TP rispetto al totale di elementi predetti come Positive

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

- **Recall:** è la percentuale di elementi TP rispetto al totale di elementi P.

$$\frac{TP}{P}$$

- **F-measure:** è la media armonica di precision e recall

$$2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall}$$

- **G-mean:** la media geometrica è una misura di performance molto utile quando la distribuzione delle etichette nel dataset non è equa, cioè quando un'etichetta è molto più presente rispetto all'altra.

$$\sqrt{\frac{TP}{P}\ x \frac{TN}{N}}$$

Inizialmente sono stati sviluppati e testati due metodi molto diversi tra di loro, ognuno con una base teorica indipendente da quella dell'altro. Questi due metodi mettono in pratica alcuni degli elementi teorici già citati nel capitolo *Literature review* , con le opportune modifiche del caso, e li integrano con degli elementi originali e innovativi. Entrambi i metodi hanno fornito dei risultati eccellenti e superiori a quelli delle tecniche di classificazione più comunemente utilizzate. Successivamente si è provato ad accorpare le due metodologie sviluppate in un unico metodo, e i risultati finali ottenuti con quest'ultimo sono stati migliori di quelli dei due metodi usati singolarmente.

Capitolo 4.1. Metodologia n°1 (Recurrence Network)

Il primo metodo si basa sull'utilizzo di grafi costruiti con una procedura innovativa per rappresentare le ricorrenze presenti nelle immagini e da questi grafi vengono poi estratte delle statistiche che successivamente vengono utilizzate come variabili di predizione durante il processo di classificazione. Lo sviluppo di tutto il processo ha richiesto molto tempo perché, dopo aver programmato e testato la prima versione, sono stati aggiunti vari elementi innovativi e sono state apportate delle modifiche con lo scopo di ottenere risultati migliori.

La versione finale del processo di questo metodo è illustrata nella Figura 10 a pagina successiva.

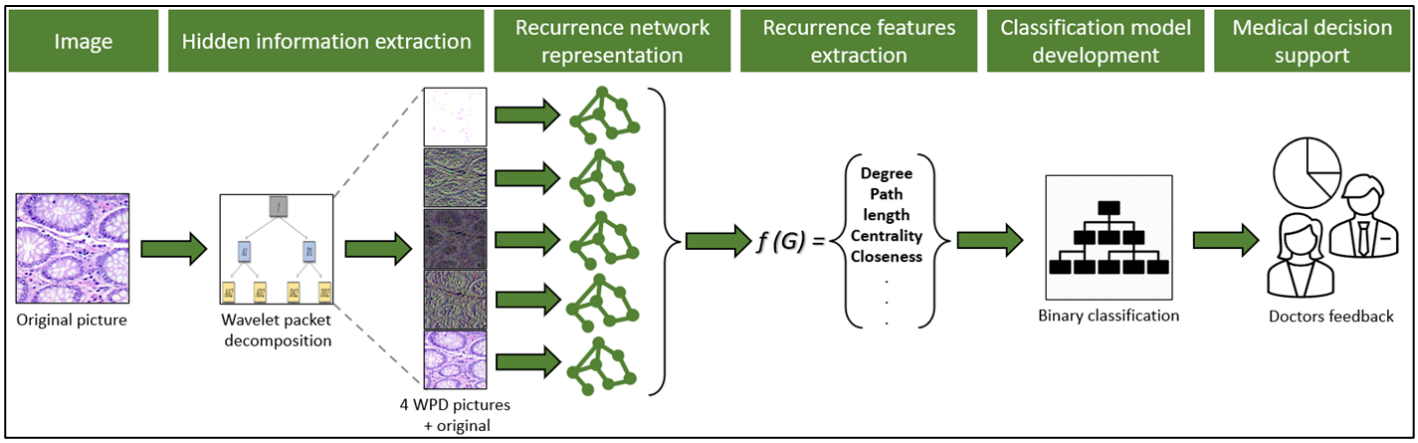

Figura 10 – Processo del metodo basato su Recurrence Network

L'illustrazione è riferita ad una singola immagine, ma durante la fase di raccolta dei dati per il training del modello questo processo deve essere ripetuto iterativamente. La descrizione dettagliata che è riportata di seguito è riferita alla fase di raccolta di dati e training del modello.

Step 1: Un'immagine del dataset, a colori e di dimensioni 768x768, viene aperta in Matlab e salvata sottoforma di matrice di dimensioni 768x768x3 contenente i valori dei colori secondo il modello RGB già esplicato in precedenza. L'etichetta ("colonaca" o "colonn") relativa all'immagine viene salvata in un vettore.

Step 2: L'immagine viene scomposta nelle sue frequenze con la Wavelet packet decomposition (WPD). La WPD è una trasformata wavelet dove il segnale (campionato) a tempo discreto è passato attraverso più filtri rispetto alla trasformata wavelet discreta (DWT – Discrete Wavelet Transform). Per implementare la DWT si usano dei filtri discreti per ottenere coefficienti wavelet discreti. Nella DWT ogni livello è calcolato passando i precedenti coefficienti di approssimazione wavelet attraverso dei filtri QMF (Quadrature Mirror Filter) a passo alto e basso. Nella WPD sia il dettaglio che i coefficienti di approssimazione sono scomposti per creare un albero binario. Per n livelli di decomposizione la WPD produce 2^n set di coefficienti (o nodi) al contrario di $n + 1$ set con la DWT. Comunque, a causa del processo di downsampling il numero di coefficienti è lo stesso e non c'è ridondanza. Matlab mette a disposizione una funzione chiamata "*wpdec2*" per applicare la WPD ad un'immagine (o segnale) ed ottenere dei coefficienti. Si è scelto di impostare come parametro della funzione $l = 1$ per ottenere quattro coefficienti, che poi sono stati applicati all'immagine per ottenere 4 nuove immagini. Il risultato finale sono cinque immagini, l'originale più le quattro ottenute con la WPD, ovvero cinque matrici di dimensioni 768x768x3.

Step 3: Le cinque immagini vengono ridimensionate e ridotte a 64x64 per diminuire drasticamente la potenza computazionale e il tempo necessari ad elaborarle. Diversi esperimenti eseguiti durante la fase di sviluppo del codice hanno dimostrato che la perdita di informazioni causata dal ridimensionamento è soltanto marginale ai fini di questa metodologia. Si procede poi con la creazione di un grafo non orientato e non pesato per ognuna delle cinque immagini, per un totale di cinque grafi. Ogni grafo ha un numero di nodi pari al numero di pixel dell'immagine ridimensionata, in questo caso 64 x 64 = 4096 nodi, e ogni nodo rappresenta un pixel. Successivamente si iterano tutte le coppie di nodi possibili e in base alla loro similarità di colore e alla loro distanza geometrica si crea o meno un arco che unisce i due pixel. Le equazioni per decidere se tra i nodi p e q deve essere creato un arco sono le seguenti:

(1) $w_{p,q} = I_{p,q} * D_{p,q}$

(2) $I_{p,q} = 1 - \frac{\vec{s}_p - \vec{s}_q}{max\{\|\vec{s}_.\|\} - min\{\|\vec{s}_.\|\}}$

(3) $D_{p,q} = \frac{\phi(\vec{p} - \vec{q})}{\phi(\|0\|)}$

Già ampiamente discusse nei capitoli precedenti, ma implementate con alcune sostanziali differenze. Nel paper scientifico le immagini trattate erano tutte in bianco e nero e dunque la differenza di colore tra due pixel era una semplice sottrazione tra i due diversi valori di intensità di grigio dei due punti, mentre in questo caso le immagini utilizzate sono tutte a colori. Il colore di ogni pixel è salvato in Matlab con un trio di valori RGB e quindi la differenza di colore tra due pixel è la distanza geometrica di due punti in uno spazio tridimensionale dove le variabili degli assi sono R,G e B.

Si considerino due pixel p e q che si trovano in posizione (p_x,p_y) e (q_x,q_y) di un'immagine, che viene poi aperta e salvata in Matlab sotto forma di matrice. Nella matrice i valori dei colori del pixel p si trovano in posizione $(p_x, p_y, 1)$, $(p_x, p_y, 2)$ e $(p_x, p_y, 3)$, le cui celle contengono rispettivamente i valori R_p, G_p, B_p, ovvero i livelli di intensità di rosso, verde e blu. Stessa cosa per il pixel q.

La differenza di colore tra questi due pixel è quindi $\sqrt{(R_p - R_q)^2 + (G_p - G_q)^2 + (B_p - B_q)^2}$.

Si è poi pensato a come risolvere il problema della ricerca del massimo e del minimo per normalizzare la differenza tra colori. Nel caso di immagini in scala di grigio questo processo è abbastanza semplice perché, trattandosi di valori singoli, basta individuare il pixel, ovvero la cella della matrice, con il livello di intensità di grigio più alto e più basso. Si è provato in un primo momento a calcolare le differenze di colore tra tutte le coppie di pixel e poi scegliere come massimo e minimo i due pixel riferiti alla differenza maggiore riscontrata, ma questo procedimento allungava troppo i tempi di elaborazione. Si è scelto successivamente di effettuare tre ricerche indipendenti, una per ognuno dei tre colori. Si ricercano in tutta la matrice i valori massimi di rosso, verde e blu, che verosimilmente sono riferiti a pixel diversi, e si salvano come M_R, M_G, M_B. Si individuano anche i valori minimi dei tre colori e si salvano come m_R, m_G, m_B. Successivamente può essere calcolata la differenza tra massimo e minimo che è uguale a $\sqrt{(M_R - m_R)^2 + (M_G - m_G)^2 + (M_B - m_B)^2}$.
Questo modo di affrontare il problema si è rivelato essere molto meno dispendioso in termini di tempo e l'efficacia era praticamente uguale al primo metodo. La ricerca dei massimi e dei minimi viene fatta prima di iterare tutte le coppie di nodi (pixel) di un'immagine. Per calcolare i vari $D_{p,q}$ si è implementata la funzione di utilità di Gauss già citata. Diversi esperimenti effettuati hanno dimostrato che il valore di sigma ottimale è circa 1/3 delle dimensioni dell'immagine, quindi in questo caso il valore di sigma utilizzato è stato 21. Come valore di soglia per confrontare $w_{p,q}$ si è scelto di utilizzare 0.85, che è un valore medio per questo tipo di analisi. Se si usa un valore di soglia troppo alto, ad esempio 0.95, si rischia che vengano creati troppi pochi archi e che quindi non si riesca a rappresentare tutte le ricorrenze presenti nell'immagine. Invece, se si usa un valore di soglia troppo basso, ad esempio 0.8, si rischia di creare troppi archi di connessione tra i vari nodi e che quindi tutto il metodo perda di significato.
Per ogni coppia di nodi p e q si calcola il $w_{p,q}$ e se questo è maggiore o uguale a 0.85 allora si crea un arco tra i due nodi.
Tutto il procedimento appena descritto va iterato per tutte le coppie di nodi e inizialmente si era pensato di utilizzare un semplice ciclo *for* annidato dentro un altro ciclo *for,* ma i tempi necessari per processare una singola immagine rendevano impossibile l'utilizzo di questo metodo su grandi quantità di immagini. Si è provato allora ad utilizzare gli strumenti di cui Matlab è dotato per ridurre i tempi. In particolare, si è pensato di assegnare un codice univoco ad ogni nodo e di creare una matrice di dimensioni *N* x 12, dove le colonne 1 e 2 riportano dei codici univoci di nodi, le colonne 3 e 4 riportano le coordinate x e y del pixel corrispondente al nodo il cui codice si trova in colonna 1, le colonne 5,6 e 7 riportano i valori di R,G e B del pixel corrispondente al nodo il cui codice si trova

in colonna 1, le colonne 8 e 9 riportano le coordinate x e y del pixel corrispondente al nodo il cui codice si trova in colonna 2, le colonne 10,11 e 12 riportano i valori di R,G e B del pixel corrispondente al nodo il cui codice si trova in colonna 2.

N è il numero di coppie diverse di nodi possibili, nel senso che si evitano le ripetizioni. Ad esempio, se in una riga della matrice si trova la coppia di nodi con codici 10 e 44, rispettivamente in colonna 1 e 2, allora non c'è nessuna riga dove sono presenti i valori 44 in colonna 1 e 10 in colonna 2.

Step 4: Dopo la creazione dei 5 grafi si procede con l'estrazione delle statistiche topologiche da ognuno di essi. Alcune delle metriche utilizzate e implementate sono le stesse già discusse nel paper scientifico, anche se con alcune modifiche. Altre metriche invece sono totalmente nuove e sono frutto di una lunga e attenta ricerca e selezione. Le metriche che si sono rivelate utili ai fini di questa metodologia sono:

- **Degree:** per ogni nodo viene calcolato il suo grado, ovvero il numero di archi che lo connettono ad altri nodi. Il grado di un nodo generico i è calcolato come $k_i = \sum_{j=1}^{n} A_{i,j}$, dove $A_{i,j}$ è il valore presente nella matrice delle adiacenze.

- **Average path length:** per ogni nodo viene calcolato il cammino minimo medio. Per ottenere questo valore si calcolano tutti i cammini minimi che connettono il nodo a tutti gli altri nodi e poi si fa una media di tutti i valori ottenuti. L'equazione matematica implementata è la seguente $L = \frac{1}{n(n+1)} \sum_{i \neq j} \frac{1}{D_{i,j}}$, dove $D_{i,j}$ è la distanza tra il nodo i e il nodo j mentre n è in numero di nodi presenti nel grafo, in questo caso 4096

- **Eigenvector centrality:** questa metrica misura l'importanza di un nodo prendendo in considerazione l'importanza dei suoi vicini, ovvero dei nodi connessi direttamente al nodo in esame con un arco. Questa metrica è chiamata anche indice di importanza relativa di un nodo i ed è calcolata come $x_i = \frac{1}{\lambda} \sum_{j \in M(x)} x_j$, dove x_i è l'importanza relativa del nodo i, x_i è l'importanza relativa del nodo j, $M(x)$ è l'insieme di nodi vicini di i e $\boldsymbol{\lambda}$ è una costante. Può anche essere espressa in notazione vettoriale come $\mathbf{Ax} = \boldsymbol{\lambda}\mathbf{x}$**,** dove **A** è la matrice delle adiacenze del grafo. L'importanza di un nodo, come si evince dalla formula, dipende sia dalla quantità di vicini che dall'importanza relativa dei vicini.

- **PageRank centrality:** si basa su un algoritmo sviluppato da Larry Page e Sergei Brian, i fondatori di Google, per ordinare le pagine web. E' una variante dell'Eigenvector centrality. L'importanza relativa di un nodo è calcolata secondo la formula
  $x_i = \frac{1}{\lambda}\sum_{j \in M(x)} \frac{x_j}{degree\ (j)} + \beta$, dove β è un fattore di attenuazione ($\beta < \frac{1}{\lambda}$)

- **Betweenness centrality:** è un'altra misura di importanza di un nodo all'interno della rete. Misura quanto spesso un nodo compare nei cammini minimi tra le varie coppie di nodi del grafo. La betweenness centrality di un nodo i è calcolata come
  $BC(i) = \sum_j \sum_k \frac{p(i,j,k)}{p(j,k)}\ con\ i \neq j \neq k$, dove $p(j,k)$ è il numero di cammini minimi che collegano il nodo j al nodo k e $p(i,j,k)$ è il numero di cammini più brevi tra j e k che passano anche per il nodo i

- **Closeness centrality:** questa misura di centralità quantifica quanto un nodo è vicino agli altri nodi del grafo. La closeness centrality di un nodo i si calcola come $CC(i) = \frac{1}{\sum_1^n d(i,j)}$, ovvero come l'inverso della somma di tutti i cammini minimi che collegano il nodo i a tutti gli altri nodi.

- **Clustering coefficient:** Il coefficiente di clustering di un grafo misura la probabilità che due nodi connessi con un nodo i siano a loro volta connessi tra di loro, ovvero formando un triangolo. E' una misura di quanto i nodi del grafo tendono a formare clusters, ovvero insiemi di nodi con delle connessioni molto fitte tra di loro. Il coefficiente di clustering di un nodo i viene calcolato come $C_i = \frac{2\Delta_i}{k_i(k_i-1)}$, dove Δ_i è il numero di triangoli centrati sul nodo i.

- **Average neighbors degree:** per ogni nodo si calcola il grado medio di tutti i suoi vicini. L'equazione implementata è $ANDeg(i) = \frac{\sum_{j \in N(i)} degree(j)}{N(i)}$, dove $N(i)$ è l'insieme di tutti i vicini del nodo i. Un vertice viene definito popolare quando il suo grado è maggiore del grado medio di tutti i suoi vicini, ovvero $degree(i) > ANDeg(i)$. Il paradosso dell'amicizia dice che nella maggior parte dei grafi la percentuale di nodi i per cui

$degree(i) < ANDeg(i)$ è più del 50%.

- **S Metric:** è una misura di quanto sono interconnessi tra di loro i nodi con un alto valore di grado. La S Metric di un grafo G è la somma dei prodotti dei gradi di tutte le coppie di nodi collegate tra di loro con un arco, ovvero $S(G) = \sum_{(i,j)\in E(G)} degree(i) * degree(j)$, dove $E(G)$ è l'insieme di archi del grafo.

Di tutte queste statistiche soltanto due (clustering coefficient e S Metric) sono dei valori singoli riferiti a tutto il grafo. Tutte le altre sono dei valori riferiti ai singoli nodi, quindi l'output del software per ognuna di esse è un vettore di lunghezza 4096 (numero di nodi) e ogni cella del vettore è riferita ad uno specifico nodo. Ad esempio, si consideri il vettore Degree e si supponga che in posizione i-esima ci sia il valore 150. Questo significa che il nodo contraddistinto con il codice univoco i ha un grado pari a 150. Per ovviare a questo problema e avere dei valori riferiti a tutto il grafo si è scelto di estrarre delle statistiche dai vettori e utilizzarle come valori di riferimento per le successive analisi. Le statistiche estratte da ogni vettore sono: valore massimo, valore minimo, media, primo quartile, secondo quartile (mediana), terzo quartile, deviazione standard, indice di asimmetria (skewness in inglese), curtosi.
Si ottengono 9 statistiche per ognuno dei 7 vettori, per un totale di 63 valori riferiti a tutto il grafo.
A questi si devono anche aggiungere i due valori singoli (clustering coefficient e S Metric) già ottenuti in precedenza, per un totale di 65 valori per ogni grafo. Va ricordato che ogni immagine originale del dataset è scomposta, durante il processo, in 5 immagini e per ognuna di queste viene creato un grafo da cui sono estratte le statistiche, perciò alla fine del processo si ottengono 65x5=325 valori riferiti all'immagine originale. Tutti questi valori vengono salvati in una riga di una matrice e si può procedere ad analizzare una nuova immagine del dataset, ripetendo tutto il processo.

L'output finale del processo, dopo aver analizzato tutte le immagini del dataset, è una matrice di n righe (una per ogni immagine) e 325 colonne (una per ogni statistica). In questo caso $n = 10.000$.
Un altro output del processo è un vettore colonna di dimensioni 10.000x1 contenente tutte le etichette delle immagini. C'è una corrispondenza tra la posizione di un'etichetta nel vettore e l'immagine alla quale è riferita. Ad esempio, se in posizione 100 del vettore colonna si trova la stringa "colonaca" significa che l'immagine i cui valori sono riportati nella riga 100 della matrice dei valori ha come etichetta "colonaca".

Step 5: la matrice dei valori di dimensioni 10.000x325 ed il vettore colonna con le etichette vengono utilizzati come input per la fase di training e testing di un modello di classificazione binaria. Durante la fase di sviluppo sono stati testati e messi a confronto diversi algoritmi di classificazione, come il Random Forest, K-NN, SVM, discriminant analysis, neural network, adaboost. I risultati migliori sono stati ottenuti con il metodo Random Forest, che è implementabile in Matlab tramite la funzione "fitcensemble". Il metodo Random Forest è un algoritmo di machine learning supervisionato, ovvero che utilizza delle etichette durante la fase di training, al contrario dei metodi non supervisionati che identificano pattern in dati che non sono etichettati. che fornisce un output di predizione combinando gli output di un grande numero di alberi di decisione (decision tree). Ciascun albero è costruito indipendentemente dagli altri

La sintassi della funzione è *Mdl=fitcensemble(Tbl,Y),* dove il primo parametro di input *Tbl* è una matrice in cui ogni colonna è una variabile predittiva, il secondo parametro *Y* è il vettore con tutte le etichette e l'output *Mdl* è il modello di classificazione. Questa funzione usa di default un numero di alberi di decisione (decision trees) pari a 100. Quando nel vettore di etichette sono presenti solamente due valori distinti, ossia quando si tratta di classificazione binaria, la funzione *fitcensemble* usa di default il metodo *LogitBoost.*
LogitBoost è un algoritmo di boosting creato da Jerome Friedman, Trevor Hastie, and Robert Tibshirani (Jerome Friedman, 2000), ed è correlato al metodo AdaBoost. Il termine "boosting" fa riferimento ad una famiglia di algoritmi che convertono dei classificatori deboli in un classificatore forte. La principale differenza è che AdaBoost minimizza la perdita esponenziale, mentre LogitBoost minimizza la perdita logistica.

Il processo di training e testing del modello di classificazione è stato ripetuto 30 volte per avere dei risultati più affidabili e per riuscire a calcolare la varianza dei risultati ottenuti. Per ogni ripetizione sono state selezionate in maniera del tutto casuale l'80% delle righe della matrice dei dati di input e le corrispondenti etichette e questi dati sono stati usati per il training del modello. Successivamente il restante 20% dei dati è stato utilizzato per la fase di testing, dunque ad ogni ripetizione i dati usati per il training e testing sono stati diversi e ogni volta i dati utilizzati per il testing non erano utilizzati per il training, e viceversa. Alla fine di ogni ripetizione i risultati ottenuti sono stati salvati in una matrice di risultati. In Matlab, l'output finale dopo tutte le iterazioni del processo di classificazione è stata una matrice di 30 righe (una per ogni ripetizione del processo) e 7 colonne (accuracy, sensitivity, specificity, precision, recall, F measure, G mean). Successivamente

è stata calcolata la media e la deviazione standard per ogni colonna. I risultati finali sono stati eccellenti e sono riportati nella Figura 11.

	Media	**Deviazione Standard**
Accuracy	**98.08%**	**0.45%**
Specificity	**98.53%**	**0.42%**
Precision (Positive)	**98.50%**	**0.42%**
Recall (Positive)	**97.63%**	**0.50%**
F Measure (Positive)	**98.07%**	**0.46%**
G Mean	**98.08%**	**0.45%**

Figura 11 – Risultati ottenuti con il metodo RN

Step 6: Dopo che si sono completate le fasi di training e testing, il modello può essere eventualmente utilizzato da parte di un medico per analizzare un'immagine istopatologica del colon e avere una seconda opinione che sia di aiuto nella diagnosi finale. Per raggiungere questo scopo, però, il modello dovrebbe essere integrato in un'interfaccia grafica user-friendly e facilmente utilizzabile da parte di attori esterni.

Capitolo 4.2. Metodologia n°2 (Heterogeneous Recurrence Quantification Analysis)

Il secondo metodo si basa sull'analisi delle ricorrenze eterogenee delle immagini e sull'estrazione di statistiche HRQA che vengono poi utilizzate come input nel processo di classificazione. La Figura 12, a pagina successiva, riporta uno schema del processo.

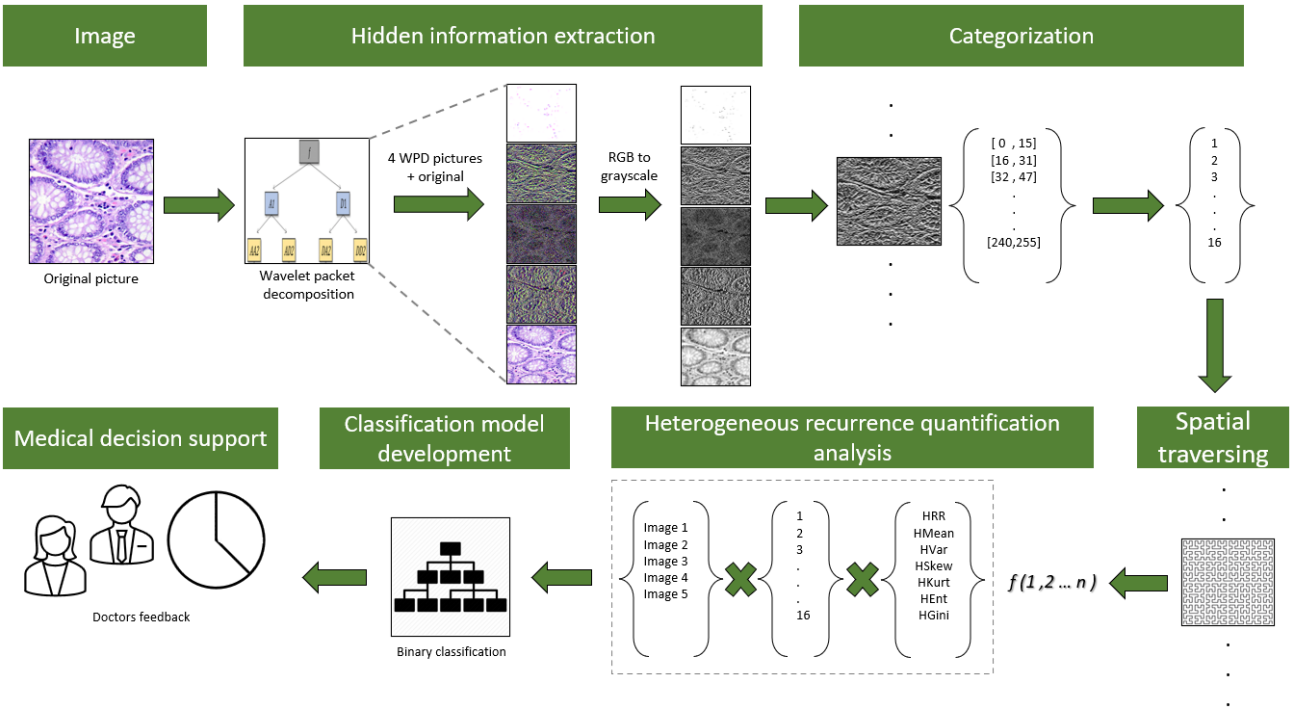

Figura 12 – Processo del metodo basato su HRQA

Anche in questo caso l'illustrazione è riferita ad una singola immagine, ma durante la fase di raccolta dei dati per il training del modello questo processo deve essere ripetuto iterativamente. Di seguito sono descritti in dettaglio tutte le fasi che caratterizzano questo metodo.

Step 1: questo step è uguale a quello del metodo n°1 già discusso. Un'immagine del dataset, a colori e di dimensioni 768x768, viene aperta in Matlab e salvata sottoforma di matrice di dimensioni 768x768x3 contenente i valori dei colori secondo il modello RGB. L'etichetta ("colonaca" o "colonn") relativa all'immagine viene salvata in un vettore.

Step 2: anche in questo caso l'immagine viene scomposta nelle sue frequenze con la Wavelet Packet Decomposition (WPD). Il risultato finale sono cinque immagini, l'originale più le quattro ottenute con la WPD, ovvero cinque matrici di dimensioni 768x768x3.

Step 3: anche in questo caso le immagini vengono ridotte di dimensioni e portate a 64x64 per ridurre la potenza computazionale e il tempo richiesti. Successivamente, le cinque immagini vengono convertite in bianco e nero (grayscale) e vengono salvate come cinque matrici di dimensioni 64x64 in cui ogni cella contiene il valore di intensità di grigio di un pixel. Questa fase non è presente nel metodo teorico descritto nel paper scientifico, che utilizza le immagini a colori, ma si è deciso di procedere in questo modo per ridurre la complessità dell'algoritmo e il tempo necessario per

l'elaborazione poiché si è scoperto che le informazioni ai fini di questo processo vengono mantenute anche nelle immagini in scala di grigio.

Step 4: successivamente si passa alla fase di categorizzazione. I valori presenti nelle cinque matrici vengono trasformati e categorizzati e il valore di ogni cella viene sostituito dal valore numerico della categoria corrispondente. Si è scelto di utilizzare 16 categorie, numerate da 1 a 16. Tutti i pixel nel range [0,15] presenti nelle matrici vengono sostituiti con il valore 1, tutti i pixel nel range [16,31] vengono sostituiti con il valore 2 e così via, fino ad arrivare ai pixel nel range [240,255] che vengono sostituiti con il valore 16. L'output finale sono cinque matrici contenenti valori nel range [1,16].

Step 5: ognuna delle cinque immagini viene attraversata da una curva di Hilbert con livello l=6 e orientamento v=1, che corrisponde al Nord. Prima di procedere con l'attraversamento vengono creati cinque vettori, uno per ogni immagine, inizialmente di lunghezza nulla. Ad ogni passo dell'attraversamento di un'immagine viene aggiunto al vettore il valore corrispondente al valore presente nella cella della matrice che la curva sta attraversando in quell'istante. L'output finale del processo sono cinque vettori, ognuno corrispondente ad una delle cinque immagini, di lunghezza pari a 4096 (64 x 64) , ossia pari al numero di pixel (celle) presenti in ogni immagine (matrice). Questi vettori contengono ovviamente tutti valori compresi nel range [1,16].

Step 6: da ogni vettore vengono estratte le statistiche di Heterogeneous Recurrence Quantification Analysis. Queste statistiche (HRR, HMean, HVar, HSkew, HKurt, HEnt, HGini) sono state già ampiamente descritte nei capitoli precedenti. Viene estratto un set di statistiche per ognuna delle 16 categorie, perché come già detto le ricorrenze vengono trattate in modo eterogeneo. Ad esempio, HRR1 è il valore di heterogeneous recurrence rate riferito alla categoria 1, cioè è la percentuale di ricorrenze aventi per oggetti i pixel assegnati alla categoria 1, HRR2 è il valore di heterogeneous recurrence rate riferito alla categoria 2, cioè è la percentuale di ricorrenze aventi per oggetti i pixel assegnati alla categoria 2, e così via anche per le altre metriche. Si ottengono quindi 7 x 16 = 112 valori per ognuna delle cinque immagini, per un totale di 112 x 5 = 560 valori per ogni immagine del dataset. Tutti questi valori poi vengono salvati in una riga di una matrice e si può procedere con l'analisi di una nuova immagine del dataset.

L'output finale di tutto il processo per la raccolta dei dati per il training e testing del modello è una matrice di 10.000 righe, una per ogni elemento del dataset, e 560 colonne, una per ognuna delle statistiche descrittive appena citate.

Step 7: successivamente, si può procedere con la creazione di un modello di classificazione con apprendimento supervisionato che utilizzi la matrice dei valori come input. Come nel caso precedente, si è provato ad applicare diversi metodi di classificazione ed anche in questo caso i risultati migliori sono stati ottenuti con il metodo Random Forest, implementato in Matlab con la funzione "*fitcensemble"*, usando un numero di alberi pari a 100 e usando l'algoritmo di boosting LogitBoost.

Il processo di training e testing del modello di classificazione è stato ripetuto 30 volte per avere dei risultati più affidabili e per riuscire a calcolare la varianza dei risultati ottenuti. Anche in questo caso, per ogni ripetizione sono state selezionate in maniera del tutto casuale l'80% delle righe della matrice dei valori di input e le corrispondenti etichette e questi dati sono stati usati per il training del modello. Successivamente il restante 20% dei dati è stato utilizzato per la fase di testing, dunque ad ogni ripetizione i dati usati per il training e testing sono stati diversi e ogni volta i dati utilizzati per il testing non erano utilizzati per il training, e viceversa. Alla fine di ogni ripetizione i risultati ottenuti sono stati salvati in una matrice di risultati. In Matlab, l'output finale dopo tutte le iterazioni del processo di classificazione è stata una matrice di 30 righe (una per ogni ripetizione del processo) e 7 colonne (accuracy, sensitivity, specificity, precision, recall, F measure, G mean). Successivamente è stata calcolata la media e la deviazione standard per ogni colonna.I risultati finali si sono rivelati eccellenti anche con questo metodo e sono riportati di seguito nella Figura 13.

	Media	**Deviazione Standard**
Accuracy	**97.99%**	**0.68%**
Specificity	**98.50%**	**0.61%**
Precision (Positive)	**98.48%**	**0.62%**
Recall (Positive)	**97.48%**	**0.91%**
F Measure (Positive)	**97.98%**	**0.70%**
G Mean	**97.99%**	**0.69%**

Figura 13 – Risultati ottenuti con il metodo HRQA

Step 8: una volta che il modello di classificazione viene creato esso può essere utilizzato da parte di un medico per avere una seconda opinione durante l'analisi istopatologica del colon, anche se per essere effettivamente utile il modello dovrebbe essere integrato all'interno di un software di tipo CAD (Computer-aided diagnosis) con interfaccia user-friendly.

Capitolo 4.3. Metodologia n°3 (RN+HRQA)

I risultati ottenuti con i due metodi proposti si possono ritenere soddisfacenti e si sono rivelati superiori a quelli ottenuti con i metodi di classificazione più comunemente utilizzati. Successivamente, si è deciso di provare ad ottenere dei risultati ancora migliori combinando i due metodi ed effettivamente i risultati finali ottenuti sono stati superiori a quelli ottenuti con i metodi usati singolarmente. Con questo nuovo metodo ogni immagine del dataset è stata analizzata dapprima seguendo il metodo n°1 (RN) e sono stati salvati i valori delle statistiche estratte, poi è stato applicato il metodo n°2 (HRQA) e sono stati salvate le statistiche ottenute. L'output finale di tutto il processo è stata una matrice di valori di 10.000 righe, una per ogni immagine del dataset, e 885 colonne, di cui 325 per le statistiche estratte con il primo metodo e 560 per quelle estratte con il secondo metodo. Ovviamente durante il processo iterativo sono state anche salvate tutte le etichette delle immagini in un vettore. Successivamente si è provato a creare un modello di classificazione applicando il metodo Random Forest tramite la funzione "*fitcensemble"* di Matlab, anche stavolta con numero di alberi di default pari a 100 e LogitBoost come algoritmo di boosting, e anche stavolta il procedimento è stato riprodotto 30 volte. Ad ogni ripetizione sono state selezionate in maniera casuale l'80% delle righe della matrice di valori e le rispettive etichette e sono state usate per il training del modello, mentre il restante 20% è stato utilizzato solamente per la fase di testing del modello. Alla fine è stata calcolata la media e la deviazione standard per tutti i valori ottenuti. La Figura 14 mostra i risultati finali ottenuti con questa metodologia.

	Media	Deviazione Standard
Accuracy	**99.39%**	**0.18%**
Specificity	**99.71%**	**0.19%**
Precision (Positive)	**99.70%**	**0.19%**
Recall (Positive)	**99.07%**	**0.30%**
F Measure (Positive)	**99.38%**	**0.19%**
G Mean	**99.39%**	**0.19%**

Figura 14 – Risultati ottenuti con il metodo finale (RN + HRQA)

Capitolo 4.4 L'importanza della Wavelet Packet Decomposition

Inizialmente i tre metodi sono stati sviluppati senza includere la fase di decomposizione delle immagini con la Wavelet Packet Decomposition, che è stata un'idea implementata in un secondo momento per provare a migliorare i risultati dei modelli. Il metodo n°1 (Recurrence Network) originariamente prevedeva la costruzione di un unico grafo relativo all'immagine in analisi ridotta a 64x64 e l'estrazione delle relative statistiche del grafo. L'output finale del processo applicato ad un'immagine del dataset era un set di 65 statistiche che venivano poi utilizzate come variabili predittive per il processo di classificazione. Anche in questo caso il processo di training e testing veniva ripetuto 30 volte come già spiegato nel capitolo precedente. I risultati ottenuti con il metodo RN senza l'implementazione della WPD sono riportati di seguito, nella Figura 15.

	Media	Deviazione Standard
Accuracy	**95.16%**	**1.32%**
Specificity	**95.64%**	**1.41%**
Precision (Positive)	**95.54%**	**1.44%**
Recall (Positive)	**94.68%**	**1.42%**
F Measure (Positive)	**95.11%**	**1.34%**
G Mean	**95.16%**	**1.32%**

Figura 15 – Risultati con il metodo RN senza WPD

Di seguito, nella Figura 16, viene riportata anche la comparazione dei risultati ottenuti con il primo metodo con e senza implementazione di WPD

	Media senza WPD	Media con WPD	Dev. Std senza WPD	Dev. Std con WPD
Accuracy	**95.16%**	**98.08%**	**1.32%**	**0.45%**
Specificity	**95.64%**	**98.53%**	**1.41%**	**0.42%**
Precision (Positive)	**95.54%**	**98.50%**	**1.44%**	**0.42%**
Recall (Positive)	**94.68%**	**97.63%**	**1.42%**	**0.50%**
F Measure (Positive)	**95.11%**	**98.07%**	**1.34%**	**0.46%**
G Mean	**95.16%**	**98.08%**	**1.32%**	**0.45%**

Figura 16 – Comparazione metodo RN con e senza WPD

Come si può facilmente notare, i risultati ottenuti dopo l'implementazione della Wavelet Packet Decomposition sono superiori a quelli ottenuti con il metodo originario di circa tre punti percentuali. Anche le deviazioni standard presentano grosse differenze, con quelle relative al metodo senza WPD che sono circa il triplo di quelle ottenute con il metodo contenente l'implementazione della WPD.
Inizialmente anche il secondo metodo, quello che si basa sull'heterogeneous recurrence quantification analysis, era stato implementato senza la decomposizione tramite WPD. L'output finale del processo era, per ogni immagine, un set di 112 statistiche che venivano salvate in una matrice per essere poi utilizzate come variabili predittive durante il processo di classificazione. I risultati ottenuti con il secondo metodo senza la WPD sono riportati nella Figura 17.

	Media	Dev. Standard
Accuracy	**94.15%**	**0.86%**
Specificity	**95.82%**	**0.93%**
Precision (Positive)	**95.69%**	**0.89%**
Recall (Positive)	**92.48%**	**1.27%**
F Measure (Positive)	**94.05%**	**0.86%**
G Mean	**94.13%**	**0.86%**

Figura 17 – Risultati con HRQA senza WPD

La Figura 18 riporta la comparazione dei risultati ottenuti con il secondo metodo con e senza implementazione di WPD.

	Media senza WPD	**Media con WPD**	**Dev. Std senza WPD**	**Dev. Std con WPD**
Accuracy	**94.15%**	**97.99%**	**0.86%%**	**0.68%**
Specificity	**95.82%**	**98.50%**	**0.93%**	**0.61%**
Precision (Positive)	**95.69%**	**98.48%**	**0.89%**	**0.62%**
Recall (Positive)	**92.48%**	**97.48%**	**1.27%**	**0.91%**
F Measure (Positive)	**94.05%**	**97.98%**	**0.86%**	**0.70%**
G Mean	**94.13%**	**97.99%**	**0.86%**	**0.69%**

Figura 18 – Comparazione metodo HRQA con e senza WPD

Come si nota dalla figura a sinistra, i risultati ottenuti senza WPD sono stati di gran lunga inferiori a quelli ottenuti con l'implementazione della WPD. In questo caso, la differenza tra le varie coppie di risultati è ancora più netta di quella del metodo precedente, dove la differenza era di circa tre punti percentuali per ognuna delle metriche di performance. Al contrario del metodo precedente, però, la differenza tra le deviazioni standard è più contenuta.

Successivamente, si è provato a unificare i due metodi per migliorare i risultati ottenuti implementando i due metodi contemporaneamente, ottenendo in questo modo un set di 177 statistiche (65 dal primo metodo e 112 dal secondo) per ogni immagine del dataset. Si è applicato poi il classico processo di training e testing di un modello di classificazione ripetuto 30 volte. La Figura 19 riporta i risultati ottenuti con il terzo metodo senza l'implementazione della Wavelet Packet Decomposition. La Figura 20 mostra la differenza tra i risultati ottenuti prima e dopo l'implementazione della Wavelet Packet Decomposition.

	Media	Dev. Standard
Accuracy	**97.12%**	**0.65%**
Specificity	**97.88%**	**0.59%**
Precision (Positive)	**97.85%**	**0.58%**
Recall (Positive)	**96.36%**	**0.96%**
F Measure (Positive)	**97.10%**	**0.65%**
G Mean	**97.11%**	**0.65%**

Figura 19 – Risultati con RN+HRQA senza WPD

	Media senza WPD	Media con WPD	Dev. Std senza WPD	Dev. Std con WPD
Accuracy	**97.12%**	**99.39%**	**0.65%**	**0.18%**
Specificity	**97.88%**	**99.71%**	**0.59%**	**0.19%**
Precision (Positive)	**97.85%**	**99.70%**	**0.58%**	**0.19%**
Recall (Positive)	**96.36%**	**99.07%**	**0.96%**	**0.30%**
F Measure (Positive)	**97.10%**	**99.38%**	**0.65%**	**0.19%**
G Mean	**97.11%**	**99.39%**	**0.65%**	**0.19%**

Figura 20 – Comparazione metodo RN+HRQA con e senza WPD

I risultati ottenuti originariamente con il terzo metodo originale, ovvero senza la WPD, erano ottimi ma sono risultati essere inferiori a quelli ottenuti successivamente di circa 2 punti percentuali per ciascuna misura di performance. D'altro canto, la differenza riscontrata tra le varie deviazione standard ottenute è stata molto ampia, con quelle riferite al metodo senza WPD che sono risultate essere circa tre volte maggiori di quelle ottenute dopo l'implementazione della WPD.

Capitolo 4.5. LogitBoost vs AdaBoost

Come precedentemente descritto, i risultati migliori sono stati ottenuti con l'utilizzo del metodo Random Forest implementato in Matlab tramite la funzione "*fitcensemble*". Questa funzione usa come algoritmo di boosting, di default, il LogitBoost. Si è provato a ripetere tutti gli stessi processi utilizzando come algoritmo di boosting AdaBoost, per riuscire a comprendere quanto l'algoritmo di boosting utilizzato influenzasse i risultati finali. I processi di raccolta dei dati da utilizzare come input

per il training del modello erano identici a quelli già descritti perciò sono state riutilizzate le matrici dei valori ottenute con i metodi precedenti, che erano state salvate nel workspace di Matlab. L'unica differenza con i metodi precedenti è stata nell'implementazione della funzione "*fitcensemble"*, perché in questo caso si è implementato l'algoritmo AdaBoost aggiungendo un parametro facoltativo alla funzione. I risultati ottenuti ripetendo il metodo n°1 (Recurrence Network) con questa modifica sono riportati nella Figura 21. La Figura 22, invece, mostra la differenza tra i risultati ottenuti con AdaBoost e quelli ottenuti con LogitBoost.

	Media	**Deviazione standard**
Accuracy	**94.96%**	**0.47%**
Specificity	**95.51%**	**0.57%**
Precision (Positive)	**95.41%**	**0.56%**
Recall (Positive)	**94.40%**	**0.63%**
F Measure (Positive)	**94.90%**	**0.46%**
G Mean	**94.95%**	**0.46%**

Figura 21 – Risultati ottenuti con il metodo RN con AdaBoost

	Media con AdaBoost	**Media con LogitBoost**	**Dev. Std con AdaBoost**	**Dev. Std con LogitBoost**
Accuracy	**94.96%**	**98.08%**	**0.47%**	**0.45%**
Specificity	**95.51%**	**98.53%**	**0.57%**	**0.42%**
Precision (Positive)	**95.41%**	**98.50%**	**0.56%**	**0.42%**
Recall (Positive)	**94.40%**	**97.63%**	**0.63%**	**0.50%**
F Measure (Positive)	**94.90%**	**98.07%**	**0.46%**	**0.46%**
G Mean	**94.95%**	**98.08%**	**0.46%**	**0.45%**

Figura 22 – Comparazione metodo RN con AdaBoost e LogitBoost

La performance ottenuta con il primo metodo utilizzando AdaBoost invece di LogitBoost è stata sensibilmente inferiore, con una differenza tra le varie coppie di valori mediamente superiore a tre punti percentuali. Le deviazioni standard, invece, sono risultate essere abbastanza simili, anche se alcune di quelle ottenute con AdaBoost sono state leggermente maggiori.

I risultati ottenuti con il metodo n°2 (HRQA) con AdaBoost sono riportati nella Figura 23. La Figura 24 mostra la differenza tra i risultati ottenuti con AdaBoost e LogitBoost.

	Media	**Deviazione Standard**
Accuracy	**96.58%**	**0.49%**
Specificity	**97.24%**	**0.57%**
Precision (Positive)	**97.20%**	**0.54%**
Recall (Positive)	**95.92%**	**0.81%**
F Measure (Positive)	**96.55%**	**0.50%**
G Mean	**96.57%**	**0.49%**

Figura 23 – Risultati ottenuti con il metodo HRQA con AdaBoost

	Media con AdaBoost	**Media con LogitBoost**	**Dev. Std con AdaBoost**	**Dev. Std con LogitBoost**
Accuracy	**96.58%**	**97.99%**	**0.49%**	**0.68%**
Specificity	**97.24%**	**98.50%**	**0.57%**	**0.61%**
Precision (Positive)	**97.20%**	**98.48%**	**0.54%**	**0.62%**
Recall (Positive)	**95.92%**	**97.48%**	**0.81%**	**0.91%**
F Measure (Positive)	**96.55%**	**97.98%**	**0.50%**	**0.70%**
G Mean	**96.57%**	**97.99%**	**0.49%**	**0.69%**

Figura 24 – Comparazione metodo HRQA con AdaBoost e LogitBoost

Anche in questo caso i risultati ottenuti sono inferiori a quelli ottenuti utilizzando l'algoritmo LogitBoost. Le deviazioni standard, però, sono risultate inferiori a quelle ottenute con LogitBoost. Questo è stato l'unico caso in cui le deviazioni standard ottenute implementando AdaBoost sono state inferiori alle altre. Successivamente si è provato ad eseguire il processo di training e testing del terzo metodo (RN + HRQA) utilizzando la funzione "*fitcensemble"* con AdaBoost come parametro e i risultati ottenuti sono riportati nella Figura 25. La Figura 26 mostra la differenza tra i risultati ottenuti con la metodologia definitiva (RN+HRQA) con AdaBoost e LogitBoost.

	Media	Deviazione Standard
Accuracy	**98.52%**	**0.28%**
Specificity	**98.73%**	**0.40%**
Precision (Positive)	**98.72%**	**0.40%**
Recall (Positive)	**98.30%**	**0.34%**
F Measure (Positive)	**98.51%**	**0.28%**
G Mean	**98.52%**	**0.28%**

Figura 25 – Risultati ottenuti con il metodo RN+HRQA con AdaBoost

	Media con AdaBoost	Media con LogitBoost	Dev. Std con AdaBoost	Dev. Std con LogitBoost
Accuracy	**98.52%**	**99.39%**	**0.28%**	**0.18%**
Specificity	**98.73%**	**99.71%**	**0.40%**	**0.19%**
Precision (Positive)	**98.72%**	**99.70%**	**0.40%**	**0.19%**
Recall (Positive)	**98.30%**	**99.07%**	**0.34%**	**0.30%**
F Measure (Positive)	**98.51%**	**99.38%**	**0.28%**	**0.19%**
G Mean	**98.52%**	**99.39%**	**0.28%**	**0.19%**

Figura 26 – Comparazione metodo RN+HRQA con AdaBoost e LogitBoost

I risultati ottenuti con l'algoritmo AdaBoost sono stati comunque eccellenti, anche se inferiori a quelli ottenuti con LogitBoost, e la differenza media è inferiore ad un punto percentuale. Le deviazioni standard ottenute sono state superiori ma comunque nel complesso abbastanza basse.
Come si è dimostrato, l'algoritmo di boosting LogitBoost si è rivelato migliore per tutti e tre i metodi.

Capitolo 4.6. Risultati con metodi tradizionali

Si è provato ad applicare qualcuno dei metodi più comunemente utilizzati per la classificazione binaria di immagini per confrontare la performance del metodo elaborato durante questo progetto di tesi con quella dei metodi tradizionali e si è scoperto che il metodo descritto in questa tesi non solo è molto innovativo, ma è anche più preciso degli altri. Per avere un confronto valido sono state utilizzate le stesse 10.000 immagini istopatologiche del colon già usate in precedenza.

Convolutional Neural Network: si è provato a sviluppare e testare un modello CNN e a tal proposito Matlab mette a disposizione dell'utente una comoda applicazione chiamata "Deep Network Designer". Per prima cosa sono state importate le immagini del dataset all'interno dell'applicazione e l'80% di esse, scelte in maniera casuale, sono state impostate come training set, mentre il restante 20% delle immagini sono state usate per la validazione. Il design del modello CNN in Matlab è stato il seguente:

1. ImageInputLayer: questo elemento processa e prepara le immagini da utilizzare come input ed in questo caso si è scelto di ridurre le immagini a 64x64, come si è fatto con il metodo già descritto
2. Convolution2DLayer: questo elemento applica dei filtri convoluzionali all'input.
3. BatchNormalizationLayer: questo elemento normalizza un mini-batch di dati in tutte le osservazioni per ciascun canale in modo indipendente. E' un elemento molto utile da usare tra il Convolution2DLayer e il ReluLayer per accelerare il processo di training della rete neurale convoluzionale e ridurre la sensitività nella fase di inizializzazione del network.
4. ReluLayer: questo elemento trasforma in 0 ogni valore dell'input che è minore di 0.
5. FullyConnectedLayer: tutti i valori di input vengono moltiplicati per una matrice di pesi e poi viene sommato un vettore di bias.
6. SoftmaxLayer: questo elemento applica una funzione softmax (funzione esponenziale normalizzata) all'input
7. ClassificationLayer: valuta la performance ottenuta con il processo in termini di accuratezza

Per il training, il numero di Epoch è stato fissato a 5 e il valore di dimensione del mini-batch è stato fissato a 128. L'accuratezza ottenuta con l'applicazione del CNN è stata del **90.10%**, quindi ben inferiore al risultato ottenuto dal metodo definitivo (RN+HRQA), anzi è addirittura inferiore ai risultati dei primi due metodi implementati singolarmente.

Support Vector Machine: si è provato a sviluppare, allenare e testare un classificatore binario SVM per comparare la performance con quella del metodo innovativo descritto in questa tesi. Creare un modello SVM che operi su immagini piuttosto che su elementi caratterizzati da variabili numeriche è piuttosto complesso. Per fare ciò sono state utilizzate delle funzioni messe a disposizione da Matlab e implementabili tramite il toolbox "Computer Vision". Come input del processo sono state

usate le solite 10.000 immagini istopatologiche del colon, 50% con adenocarcinoma e 50% senza, ridotte a 64x64 per avere una comparazione più attendibile con quella del metodo n°3 sviluppato in questo progetto, visto che anche in quel caso sono state usate le immagini ridotte a 64x64.
Successivamente sono state selezionate l'80% delle immagini del dataset, in maniera del tutto casuale, da usare come set di training del modello e il restante 20% del dataset è stato poi usato per la fase di testing. E' stata poi utilizzata la funzione "*bagOfFeatures"* di Matlab che riceve come input un set di immagini, in questo caso il set di training formato dall'80% delle immagini, e ne estrae delle caratteristiche visive. Queste feature visive sono chiamate, in inglese, "visual words", ossia parole visive. Proprio come la disposizione di certe lettere forma una specifica parola anche la disposizione di certi pixel forma una feature visiva specifica. In questa funzione viene usato l'algoritmo SURF (Speeded Up Robust Features) per l'identificazione di feature visive. La funzione "*bagOfFeatures"* restituisce come output un insieme di visual words. Successivamente è stata usata la funzione "*trainImageCategoryClassifier"* per creare e allenare un classificatore SVM usando come input il set di training e l'insieme di feature visive. Infine il classificatore è stato applicato sul set di testing per provare a prevederne le etichette. La Figura 27 riporta la matrice di confusione ottenuta.

	Previsione	
Effettivo	colonaca	colonn
colonaca	87.70%	12.30%
colonn	10.80%	89.20%

Figura 27 – Risultati ottenuti con SVM

Anche in questo caso i risultati ottenuti sono stati largamente inferiori a quelli ottenuti con il metodo innovativo proposto in questo progetto di ricerca.

Capitolo 5. Conclusione e sviluppi futuri

Questo progetto di ricerca è riuscito nell'intento che era stato prefissato all'inizio, ovvero quello di creare un nuovo metodo di classificazione di immagini con un'efficacia comparabile, se non addirittura superiore, a quelli dei metodi di classificazione di immagini più diffusi al giorno d'oggi. Il caso di studio su cui è stato applicato e testato il nuovo metodo è un tema estremamente attuale e sul quale si stanno investendo grandi somme. Come già accennato, il proseguimento naturale di questa ricerca potrebbe essere la creazione di un software CAD (Computer-aided diagnosis) che

integri al suo interno il modello di classificazione sviluppato in questo progetto. Tale software, che dovrebbe essere molto user-friendly per poter essere facilmente utilizzato da parte parte del personale sanitario (medici, biologi, ecc), avrebbe come obiettivo quello di aiutare l'operatore che svolge l'analisi istopatologica del tessuto del colon a formulare la diagnosi, fornendo una seconda opinione circa la presenza o meno di adenocarcinoma. Un software di questo tipo potrebbe essere venduto o fornito come SaaS (software as a service) agli ospedali e ai laboratori che eseguono analisi istopatologiche del colon.

Il passo successivo potrebbe essere l'applicazione della metodologia sviluppata in questo progetto di ricerca anche ad altri tipi di immagini mediche. Per altre applicazioni potrebbe essere necessario integrare altre tecnologie nel processo. Ad esempio, si potrebbe usare la metodologia proposta per classificare immagini di macchie sulla pelle per decidere se si tratta di un melanoma o di una macchia benigna e fornire una seconda opinione al medico che esegue l'analisi e aiutarlo a formulare una diagnosi definitiva. Questo tipo di analisi sicuramente necessita di un'ulteriore step prima di procedere con la Wavelet Packet Decomposition per riuscire a individuare e delimitare la regione di interesse ed eliminare lo sfondo, in questo caso la pelle tutta intorno alla macchia, che non fornisce nessuna informazione. Si tratterebbe, quindi, di applicare un processo di image segmentation, ovvero segmentazione dell'immagine, per selezionare solo i pixel di interesse, ovvero i pixel relativi alla macchia sulla pelle, ed eliminare tutti gli altri pixel relativi alla pelle intorno alla macchia. I metodi di image segmentation sono già molto utilizzati nei software di analisi di immagini mediche, ne esistono di diversi tipi e ognuno di essi presenta dei vantaggi e degli svantaggi. I metodi più comuni al giorno d'oggi per la segmentazione di immagini mediche sono (Dzung L. Pham, 2020):

- **Thresholding:** viene definito un livello limite di intensità di colori e la segmentazione è ottenuta selezionando e reggruppando tutti i pixel con un'intensità di colore superiore (o inferiore) al limite. In questo modo si ottengono due classi di pixel, quelli che formano la regione di interesse e quelli che formano lo sfondo. E' possibile ampliare il metodo selezionando più di un threshold in modo tale da ottenere tre o più classi di pixel. Questo è necessario quando in un'immagine ci sono più oggetti da segmentare.
- **Region growing:** questo metodo richiede, nella sua versione più semplice, che venga selezionato un "punto seme", ovvero un pixel iniziale, che faccia parte della regione di interesse e da cui far partire il processo di segmentazione. Successivamente, l'algoritmo analizza i pixel nelle immediate vicinanze del punto seme e se questi hanno un'intensità di

colore molto simile al punto seme allora vengono selezionati e la regione di interesse viene ampliata. Il processo viene poi iterato anche sui nuovi pixel selezionati. Il punto seme può essere selezionato manualmente oppure automaticamente tramite un algoritmo. Ad esempio, si supponga di dover applicare tale metodo ad un'immagine di una macchia scura su una pelle molto chiara per selezionare la macchia ed eliminare la pelle. Come punto seme viene selezionato un pixel della macchia scura e l'algoritmo seleziona tutti i pixel che formano la macchia, riuscendo quindi a individuare la regione di interesse.

- **Clustering:** i metodi di clustering possono essere utilizzati per raggruppare i pixel simili tra di loro, ovvero con dei livelli di rosso, verde e blu molto simili nel caso di immagini a colori o con livello di intensità di grigio molto simile nel caso di immagini in bianco e nero. Si consideri ancora l'esempio di un'immagine di una macchia molto scura su una pelle molto chiara. Per dividere la macchia dalla pelle basterebbe applicare un algoritmo di clustering, ad esempio il K-means con numero di cluster k=2, e il risultato sarebbe sicuramente molto soddisfacente, poiché i pixel della pelle hanno dei livelli di colore molto simili tra di loro e lo stesso vale anche per i pixel della macchia scura. Nel caso di immagini più complesse o che contengono rumore, però, gli algoritmi di clustering possono facilmente fornire dei risultati non corretti. Gli algoritmi di clustering, infatti, non tengono conto della vicinanza spaziale dei pixel ma soltanto dei loro valori di intensità di colori.
- **Markov Random Field:** non è esattamente un metodo di segmentazione ma piuttosto un modello statistico. Il metodo del campo casuale di Markov modella le interazioni spaziali che ci sono tra pixel molto vicini tra di loro. Queste correlazioni locali forniscono un meccanismo per modellare una varietà di proprietà dell'immagine. Questo metodo viene usato nella segmentazione di immagini poiché solitamente un pixel appartiene alla stessa classe, ovvero allo stesso oggetto, dei pixel vicini. Nell'esempio di prima ci sarebbe, ovviamente, un'alta correlazione tra i vari pixel che formano la macchia scura.
- **Artificial Neural Networks:** le reti neurali artificiali, già ampiamente discusse nei capitoli precedenti, possono essere usate per la segmentazione di immagini sia come modelli di classificazione che come modelli di clustering.
- **Modelli deformabili:** questi modelli cercano di delineare i confini della regione di interesse utilizzando curve parametriche chiuse o superfici che si deformano sotto l'influenza di forze interne ed esterne. Per delineare il confine di un oggetto in un'immagine, una curva o una superficie chiusa deve essere prima posizionata vicino al confine desiderato e quindi lasciata

subire un processo di rilassamento iterativo. Le forze interne vengono calcolate dall'interno della curva o della superficie per mantenerla liscia durante tutta la deformazione. I principali vantaggi dei modelli deformabili sono la loro capacità di generare direttamente curve o superfici parametriche chiuse dalle immagini e la loro incorporazione di un vincolo di levigatezza che fornisce robustezza al rumore e ai bordi spuri. Uno svantaggio è che richiedono l'interazione manuale per posizionare un modello iniziale e scegliere i parametri appropriati.

Un'altra naturale evoluzione della metodologia descritta in questa tesi di laurea potrebbe essere l'applicazione del modello per la classificazione multiclasse e non più solamente classificazione binaria. A tal proposito, si sta già provando ad applicare la metodologia ad un dataset di 10015 immagini dermatoscopiche relative a persone di sesso ed età diverse, da utilizzare per allenare un modello di classificazione multiclasse. Il dataset, noto come HAM10000 (Human against Machine 10000), è uno dei dataset più ampi e più utilizzati nel mondo accademico per le tecniche di machine learning relative a immagini dermatologiche (Philipp Tschandl, 2018). E' formato da 10015 immagini dermoscopiche contenenti le più comuni lesioni pigmentate che sono: cheratosi attinica, carcinoma basocellulare, lesioni cheratosiche benigne, dermatofibroma, melanoma, nevi melanociti e lesioni vascolari. I metadati del dataset Il dataset è fortemente sbilanciato, con più del 60% delle immagini relative a pazienti con nevi melanocitici, che è il tipo di lesione più presente nel dataset. A seguire ci sono il melanoma e le lesioni cheratosiche benigne, con circa 1000 immagini ciascuna. Infine, ci sono solamente 115 immagini relative a pazienti con dermatofibroma, che è la lesione meno presente nel dataset. Per ogni immagine si conosce, oltre all'etichetta relativa al tipo di lesione, anche l'età del paziente, il sesso, la localizzazione sul corpo della lesione e il modo con cui è stata effettuata la diagnosi. Entrambi i sessi sono rappresentanti nel dataset in misura uguale. Più della metà delle immagini è relativa a pazienti che hanno dai 35 ai 55 anni, pochissimi hanno meno di 20 anni e il 28% ha più di 65 anni. La localizzazione delle lesioni è abbastanza variegata e circa il 20% delle immagini è relativo a lesioni che si trovano sulla schiena, che è la zona del corpo più presente nel dataset. La zona del corpo meno rappresentata sono i genitali, con solo 48 immagini. Per quanto riguarda il metodo con cui si è arrivati alla diagnosi, circa il 50% delle lesioni sono confermate tramite l'esame istopatologico, mentre il resto delle lesioni sono confermate dall'osservazione diretta da parte di un medico, esame follow-up o microscopia confocale in vivo. Usare le immagini di questo dataset come dataset di training per creare un modello di classificazione multiclasse con 7 classi con la metodologia descritta in questo progetto di ricerca si è rivelato più complesso del previsto, poiché

è necessario pre-processare le immagini per selezionare la regione di interesse, cioè la lesione, ed ignorare la pelle circostante. La segmentazione deve essere effettuata in modo del tutto automatizzato perché ovviamente è impossibile pensare di applicare dei processi di segmentazione manuali ad una tale quantità di immagini. I metodi di segmentazione automatizzati, però, commettono spesso errori, soprattutto con quei tipi di lesione che non hanno dei confini ben definiti con la pelle circostante o che sono localizzate in punti del corpo molto particolari. Inoltre, l'eventuale presenza di peli rende più difficile sia la segmentazione che il processo di apprendimento vero e proprio.

Fino ad ora si è parlato solo di classificazione di immagini mediche, ma la metodologia sviluppata in questo progetto di ricerca può essere applicata per classificare immagini di qualunque tipo, con le dovute modifiche e integrazione di nuove tecnologie. La metodologia può avere diverse applicazioni in ambito industriale e di conseguenza è possibile sviluppare dei prodotti software che implementano tale metodologia da vendere o fornire come SaaS alle aziende manifatturiere e alle società di consulenza. A tal proposito, si sta già provando ad utilizzare tale metodologia per la classificazione binaria di immagini microscopiche di finiture superficiali ottenute con processo UPM per decidere se una superficie metallica è sufficientemente liscia, quindi accettabile, o troppo ruvida, quindi non accettabile. L'UPM (Ultra-Precision Machining) è un processo manifatturiero in cui si utilizzano utensili diamantati a cristallo singolo per il taglio su scala nanometrica di pezzi in metallo. Il processo UPM è ampiamente utilizzato nei settori industriali moderni (semiconduttori, aerospaziale, ecc) per finiture superficiali di altissima precisione. La rugosità delle superfici ottenute è misurata con un apposito strumento chiamato profilometro, dotato di una punta diamantata che scorre sulla superficie metallica e ne registra la rugosità in scala micrometrica. Le misure di rugosità sono quattro: deviazione media aritmetica (Ra), altezza massima del picco (Rp), radice media al quadrato (Rq) e profondità massima della valle (Rv). Una superficie è considerata accettabile o meno in base al rispetto di limiti massimi per ciascuna di queste misure e/o in base alla loro combinazione. Se analizzate al miscoscopio, le superfici sufficientemenete lisce e le superfici ruvide forniscono delle immagini abbastanza differenti e con caratteristiche visive uniche. Di seguito, nella Figura 28, vengono riportati gli esempi di due immagini microscopiche di superfici metalliche, una (quella a

sinistra) relativa a una superficie sufficientemente liscia e l'altra (quella a destra) relativa a una superficie troppo ruvida, quindi non accettabile.

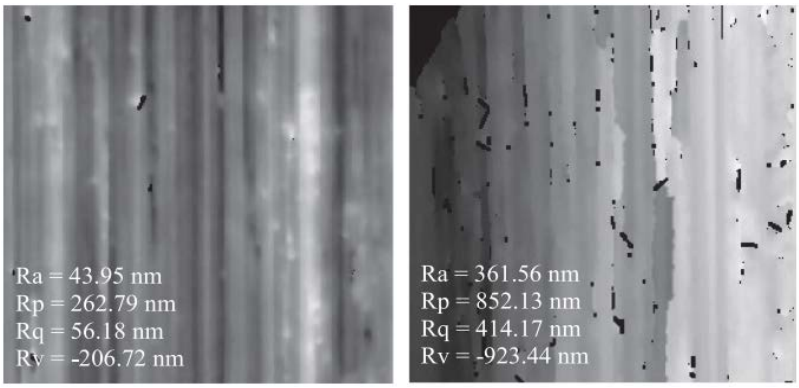

Figura 28 – Due superfici metalliche con diversa rugosità

Come si evince osservando le due immagini, le loro caratteristiche visive rendono facilmente intuibile l'appartenenza ad una categoria (liscia/accettabile) o all'altra (ruvida/non accettabile) e per questo motivo si è deciso di provare a sviluppare un modello di classificazione binaria basato sulla metodologia di questo progetto di tesi. Un grande vantaggio rispetto alla possibile applicazione descritta in precedenza, ovvero la classificazione di immagini dermoscopiche, è che non è necessario effettuare alcun tipo di segmentazione, poiché la regione di interesse è l'immagine intera. Il più grande ostacolo incontrato fino ad ora è stata la mancanza di un database abbastanza ampio di immagini microscopiche di superfici metalliche con le relative etichette (liscia/ruvida). Quando si troverà o si creerà un dataset ideale si potrà usare per la fase di training di un modello di classificazione con la procedura descritta nel capitolo precedente. Successivamente, si potrebbe procedere con lo sviluppo di un software per usi industriali che integri al suo interno tale modello di classificazione. Questo software insieme ad un microscopio potrebbero essere usati in una linea produttiva alla fine del processo UPM per valutare l'accettabilità del prodotto, in alternativa all'analisi con profilometro.

La principale limitazione della metodologia proposta in questo progetto di ricerca è che il modello di classificazione può essere applicato soltanto ad immagini dello stesso identico tipo di quelle usate durate la fase di training. Ad esempio, si supponga che un laboratorio di analisi istologiche decida di acquistare l'ipotetico software CAD che implementa il modello di classificazione per immagini istopatologiche del colon sviluppato con la metodologia descritta e in cui sono state utilizzate le 10.000 immagini del dataset citato per la fase di training. Prima di procedere con l'affare, è necessario assicurarsi che il laboratorio disponga di un microscopio che consente di ottenere immagini dello stesso identico tipo di quelle mostrate nella Figura 9. Qualsiasi differenza (intensità

dei colori, contrasto, dimensioni) renderebbe il modello di classificazione di fatto inapplicabile. Ciò significa che un'eventuale azienda produttrice di software CAD per le analisi istopatologiche di tessuto del colon che implementa la metodologia descritta in questo progetto dovrebbe fornire dei software personalizzati rispetto alle esigenze dei vari centri di analisi e dei vari tipi di microscopi che esistono in commercio. Quindi, si supponga che un laboratorio di analisi utilizzi un tipo di microscopio che non consente di ottenere lo stesso tipo di immagini di quelle presenti nel dataset usato in questo progetto di ricerca. L'azienda produttrice del software CAD, in questo caso, dovrebbe trovare un nuovo e ampio dataset di immagini istopatologiche di tessuto del colon etichettate e dello stesso tipo di quelle ottenute con il microscopio usato dal potenziale cliente e usare questo dataset per la fase di training e testing di un nuovo modello di classificazione da implementare nel software che verrebbe acquistato dal laboratorio o fornito come SaaS. La ricerca del dataset idoneo, oltre a richiedere del tempo, potrebbe risultare infruttuosa e quindi non sarebbe possibile sviluppare un modello di classificazione adatto alle esigenze di quel particolare cliente. In questo caso, una soluzione intelligente potrebbe essere quella di creare un nuovo dataset mettendo insieme tutte le precedenti immagini ottenute da quel laboratorio, se sono state salvate in precedenza, e le relative etichette.

Bibliografia

A. Sanfeliu, R. A. (2000). Graph-based representations and techniques for image.

Alpaydin, E. (2010). Introduction to Machine Learning.

Cheng-Bang Chen, H. Y. (2018). Recurrence network modeling and analysis of spatial data.

Cheng-Bang Chen, H. Y. (2019). Heterogeneous recurrence analysis of spatial.

Dzung L. Pham, C. X. (2020). Current methods in medical image segmentation. *Annual review of Biomedical Engineering*.

Gwilym S. Lodwick, C. L. (1963). Computer Diagnosis of Primary Bone Tumors.

Jerome Friedman, T. H. (2000). Additive logistic regression: a statistical view of boosting.

Maria-Luiza Antonie, O. R. (2002). Associative Classifiers for Medical Images. *Lecture Notes in Computer Science*.

Norbert Marwan, J. K. (2006). Generalised recurrence plot analysis for spatial data.

Philipp Tschandl, C. R. (2018). The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Sci Data*.

Rosenblatt, F. (1958). The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*.

Sara P. Oliveira, P. C. (2021). CAD systems for colorectal cancer from WSI are still not ready for clinical acceptance.