



POLITECNICO DI TORINO

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA BIOMEDICA

Tesi di Laurea Magistrale

**SVILUPPO DI UN SISTEMA BASATO SULL'INTELLIGENZA
ARTIFICIALE PER L'IDENTIFICAZIONE ED IL GRADING
DEL TUMORE ALLA MAMMELLA IN IMMAGINI
ISTOLOGICHE**

Relatore: Prof. Filippo Molinari

Correlatore: Massimo Salvi

Candidato: Andrea Ferraris

Anno accademico 2020-2021

PREFAZIONE

Il carcinoma alla mammella è la neoplasia più frequente nel mondo femminile con un tasso di mortalità che si attesta intorno al 17%. Il tasso di mortalità è strettamente legato alla tempestività con cui si effettua la diagnosi, tempi minori corrispondono a prognosi migliori. In ambito tecnologico, lo sviluppo del deep learning e in particolare delle CNN (convolutional neural network) ha fornito un ottimo strumento per la classificazione e la segmentazione di immagini. In questo lavoro di tesi si è sviluppato un sistema automatico per la caratterizzazione del carcinoma alla mammella in immagini istologiche basato su CNN capace di classificare il tessuto tra quattro classi: a) Normal b) Benign c) In-Situ d) Invasive. La classificazione di una eventuale neoplasia su più classi rispetto alla più semplice classificazione binaria (carcinoma/non-carcinoma), permette di poter utilizzare sul paziente trattamenti clinici più mirati e specifici, con una ripercussione positiva sull'esito della terapia. In questo progetto si è deciso di adottare una strategia in particolare, aumentare le prestazioni del classificatore migliorando la qualità e il contenuto informativo delle immagini che vengono fornite in ingresso. La fase di pre-processing è composta da due step: la normalizzazione dello stain dell'immagine tramite un modello cycle-GAN e una estrazione smart delle aree (patch) che presentano un maggior contenuto informativo per ogni immagine. Il classificatore vero e proprio è stato ottenuto aggregando 3 differenti architetture: DenseNet201, DenseNet169 e InceptionResNetV2; i risultati ottenuti dalla classificazione delle patch delle tre reti sono stati combinati ottenendo un valore per ogni patch e in una fase di post-processing è stato

ottenuto il label dell'immagine di provenienza. Infine, il sistema produce una rappresentazione in cui è possibile valutare grading e probabilità di appartenenza a una classe per ogni patch estratta da ciascuna immagine. Il sistema ha ottenuto un'accuratezza sulla classificazione in 4 classi pari al 90% superando le performance degli altri metodi presenti in letteratura sullo stesso dataset.

INDICE

CAPITOLO I

INTRODUZIONE

1.1 Carcinoma alla mammella	pag. 1
1.1.1 Fattori di rischio	pag. 4
1.2 Biopsia ed esame istologico	pag. 6
1.3 Colorazione ematossilina-eosina	pag. 10
1.4 Digital pathology	pag. 11
1.5 Cos'è una rete neurale artificiale	pag. 12
1.6 CNN (<i>Convolutional Neural Network</i>)	pag. 14
1.6.1 Padding	pag. 20
1.6.2 Pooling	pag. 21
1.6.3 Fully-connected	pag. 22
1.6.4 Classificazione	pag. 23
1.7 Analisi dello stato dell'arte	pag. 23

CAPITOLO II

MATERIALI E METODI

2.1 Descrizione Dataset	pag. 30
2.2 Stain Normalization	pag. 32
2.2.1 Cos'è una rete GAN?	pag. 33

2.2.1.1 Cycle-GAN	pag. 34
2.2.1.2 Allenamento Cycle-GAN	pag. 36
2.2.1.3 Quando terminare l'allenamento	pag. 39
2.2.2 Ricostruzione immagine normalizzata	pag. 39
2.3 Estrazione <i>smart</i>	pag. 45
2.3.1 Algoritmo estrazione smart	pag. 51
2.4 Allenamento reti	pag. 55
2.4.1 Grafici allenamento	pag. 61
2.5 Testing	pag. 63
CAPITOLO III	
RISULTATI	pag. 66
CAPITOLO IV	
CONCLUSIONI	pag. 69
BIBLIOGRAFIA	pag. 71

CAPITOLO I

INTRODUZIONE

1.1 Carcinoma alla mammella

Il cancro alla mammella è la principale causa di decessi per cancro nelle donne. Secondo l'Organizzazione mondiale della salute (OMS), ogni anno 2,1 milioni di donne ricevono una diagnosi di cancro al seno in tutto il mondo: nel 2018 sono morte circa 627.000 donne, che rappresentano circa il 15% di tutte le morti per cancro nelle donne; in Italia pertanto è il tumore che si presenta con maggior frequenza nel sesso femminile e colpisce una donna su 9 con un aumento di incidenza rispetto al passato, in particolare tra i 35-55 anni [1].

Il tumore del seno è una formazione di tessuto costituito da cellule che crescono in modo anomalo e incontrollato all'interno della ghiandola mammaria. Esistono diversi tipi di cancro della mammella, con differenti tassi di crescita e risposta alle terapie, e sia prognosi che trattamento sono influenzati dallo stadio in cui la neoplasia si trova al momento della diagnosi. La maggior parte dei carcinomi della mammella sono costituiti da tumori epiteliali che hanno origine dalle cellule che rivestono lobuli o dotti, mentre sono meno frequenti i tumori non epiteliali dello stroma di sostegno. Il carcinoma mammario è suddivisibile in due principali categorie: carcinoma in situ o di tipo invasivo [2].

Il carcinoma in situ è una proliferazione di cellule tumorali all'interno di lobuli o dotti, senza invasione del tessuto stromale. Il carcinoma di tipo in situ si presenta principalmente in due forme: duttale o lobulare.

Carcinoma duttale in situ (DCIS): circa l'85% dei carcinomi in situ è di questo tipo. Il carcinoma duttale in situ è una forma iniziale di tumore al seno (detto anche precancerosi, preinvasivo, intraduttale) in cui le cellule tumorali si sviluppano all'interno dei dotti, rimangono confinati al loro interno e non si estendono al di fuori, riversandosi nel tessuto circostante o in altre parti del corpo. Questa caratteristica rende la prognosi del DCIS favorevole rispetto a una forma di neoplasia più invasiva.

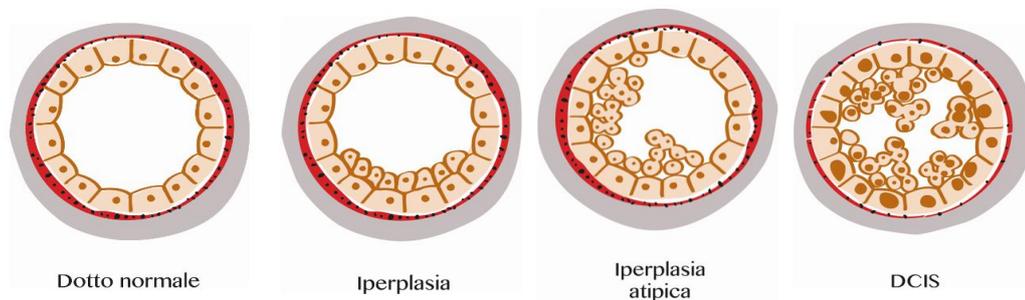


Fig.1.1 esempio di sviluppo di carcinoma alla mammella in situ Fonte: www.infoseno.org

Generalmente, non essendoci sintomi, il DCIS viene scoperto in occasione di una mammografia.; questa è la ragione per cui il DCIS è stato diagnosticato con maggior frequenza da quando sono iniziati i programmi di *screening* mammografico, infatti circa il 20% di tutte le diagnosi nei programmi di *screening* sono valutazioni di DCIS. Nella mammografia, il DCIS si può presentare come accumuli di calcio (micro-calcificazioni) che appaiono come piccole macchie bianche. Per arrivare a una diagnosi definitiva è

comunque necessaria una biopsia che in caso di nodulo ha come ultimo step l'esame istologico. Se il DCIS non viene trattato in tempi brevi le sue cellule possono estendersi al di fuori del dotto e iniziare a proliferare nel tessuto circostante, o in altre parti del corpo, e diventare un tumore duttale invasivo.

Carcinoma lobulare in situ (LCIS): è una precancerosi in cui le cellule tumorali si sviluppano all'interno dei lobuli e rimangono confinate all'interno delle ghiandole mammarie. Il LCIS non si estende al di fuori del lobulo e non invade il tessuto circostante o altre parti del corpo, per questo è classificato come un carcinoma in situ. Tale caratteristica, rende la prognosi del LCIS molto buona; il carcinoma lobulare è più frequente tra i 40-50 anni, nel 40-85% dei casi è multifocale (cioè forma più focolai di cellule tumorali nello stesso seno) e bilaterale (nel 30% dei casi può interessare tutte e due i seni). Il carcinoma lobulare in situ non è maligno ma aumenta il rischio di sviluppare il carcinoma invasivo in entrambe le mammelle, infatti si associa alla presenza di un tumore invasivo solo nel 5% delle pazienti, ma aumenta da 7 a 10 volte il rischio di uno sviluppo in futuro di tumore invasivo. Questo tipo di neoplasia non presenta sintomi e la sua diagnosi è per lo più casuale in seguito a esami di *screening*; la sua effettiva presenza può solo essere confermata dall'analisi del tessuto istologico a seguito di una biopsia.

Il carcinoma invasivo è in genere un adenocarcinoma. Circa l'80% è di tipo duttale, mentre la maggior parte della parte rimanente è lobulare. Il carcinoma duttale infiltrante viene così chiamato perchè origina dai dotti mammari ed oltrepassa la struttura della mammella diffondendosi anche al di fuori di essa, intaccando linfonodi e altri organi.

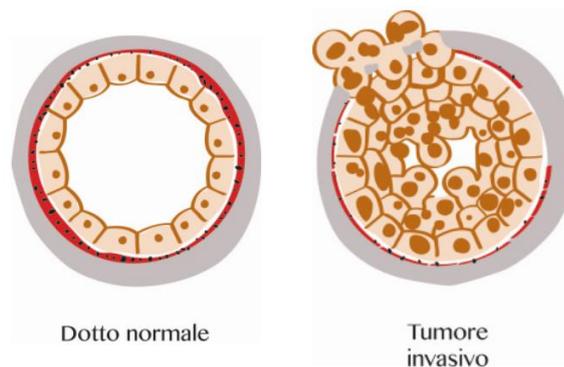


Fig.1.2 esempio di carcinoma alla mammella invasivo. Fonte www.infoseno.org

Il carcinoma lobulare infiltrante ha origine dalle cellule dei lobuli; si presenta con un'incidenza del 10-15% di tutti i tumori al seno, posizionandosi dunque al secondo posto tra i tumori invasivi più frequenti.

In alcuni casi può interessare più di una regione dello stesso seno o colpire entrambi i seni e pertanto, rispetto agli altri tipi di tumore, presenta un rischio leggermente superiore di ripresentarsi nel seno controlaterale a distanza di tempo dalla prima diagnosi.

1.1.1 Fattori di rischio del carcinoma

I fattori di rischio sono i seguenti [3]:

-Età: l'età è uno dei fattori di rischio più incisivi. Pur potendo colpire a qualsiasi età, è sicuramente più probabile in donne al di sopra dei 30 anni ed il rischio aumenta con il passare degli anni.

Il rischio più elevato incorre al superamento dei 50 anni: più del 75% dei casi di tumore del seno colpisce queste donne, mentre è meno frequente prima della menopausa.

-Anamnesi familiare: la presenza di un parente di primo grado affetto da carcinoma alla mammella può aumentare fino a tre volte il rischio di essere colpiti da tale patologia, mentre se i parenti di primo grado affetti da carcinoma al seno sono due o più, il rischio aumenta fino a sei volte.

-Mutazione del gene del carcinoma mammario: dal 5 al 10% delle donne colpite da cancro alla mammella, presentano una mutazione in uno dei due geni BRCA1 o BRCA2. Questi due geni sono responsabili del controllo del ciclo cellulare e pertanto fungono da oncosoppressori. La mutazione di uno di questi geni aumenta la possibilità di formazione di cellule figlie con mutazioni del DNA, ciò significa suscettibilità alla formazione di neoplasie.

-Anamnesi personale: aver avuto precedentemente un carcinoma alla mammella in situ o invasivo aumenta il rischio di formazione di una nuova neoplasia.

-Anamnesi ginecologica: menarca precoce, menopausa e gravidanza tardiva costituiscono dei fattori di rischio. Le donne che hanno avuto la prima gravidanza dopo i 30 anni hanno un rischio più alto rispetto a coloro che non hanno partorito.

-Uso di contraccettivi orali: il rischio aumenta lievemente soprattutto durante gli anni di utilizzo, ma si riduce durante i 10 anni dopo l'interruzione.

-Radioterapia: l'esposizione a radioterapia prima dei 30 anni incrementa il

rischio. Essere soggetti a una radioterapia a mantello per il linfoma di Hodgkin ad esempio, quadruplica il rischio di sviluppo di un carcinoma alla mammella per i successivi 20-30 anni.

-Stile di vita: uso di fumo e alcol può contribuire ad un maggior rischio di sviluppare cancro al seno.

Data l'elevata incidenza con cui il tumore alla mammella si presenta nella popolazione femminile, si è univocamente d'accordo sull'importanza dello *screening*. Questa procedura consente di individuare precocemente eventuali forme di tumore alla mammella, trattarle ed ampliare il numero di tumori identificati ai primi stadi di sviluppo della malattia, quando il trattamento ha maggiori probabilità di essere efficace e meno invasivo.

Le modalità con cui si effettua lo *screening* sono:

- Mammografia (comprese quelle digitale e tridimensionale)
- RM (indicata per i pazienti ad alto rischio)
- Autopalpazione del seno mensile

1.2 Biopsia ed esame istologico

Nel momento in cui venga identificata una massa o un tessuto sospetto occorre procedere con la biopsia per avere certezze sull'esistenza della neoplasia ed eventualmente, su tipo e grado. La biopsia al tessuto mammario consiste nel prelievo di materiale tissutale, prelevato dall'area del seno in cui è stata individuata un'anomalia o una sospetta formazione neoplasica.

Il tessuto prelevato viene poi analizzato istologicamente; lo scopo di questa analisi è quello di definire con maggiore precisione e attendibilità la natura e le caratteristiche istopatologiche del campione prelevato [4].

La biopsia del seno viene eseguita sotto guida ecografica o mammografica, introducendo nel seno un ago oppure praticando un'incisione fino a raggiungere la formazione sospetta, da cui si preleva il campione da analizzare in laboratorio.

In base all'esito dell'esame istologico, il medico specialista potrà programmare ulteriori accertamenti o impostare il percorso terapeutico più adeguato al paziente.

L'istologia è la branca della biologia che si occupa dello studio della morfologia dei tessuti e le cellule che li compongono, sia da un punto di vista morfologico che funzionale.

Le tecniche istologiche costituiscono quell'insieme di tecniche e processi che permettono la preparazione di un campione prelevato attraverso biopsia, per effettuare l'esame al microscopio.

Di seguito vengono riportate le operazioni necessarie per poter effettuare un'analisi istologica al microscopio [5]:

Fissazione: questo step serve per poter prevenire la decomposizione del campione destinato all'analisi microscopica dal momento che una volta asportati dall'organismo di appartenenza, i tessuti perdono velocemente le loro proprietà chimiche e fisiche. Questo accade sia a causa della variazione di condizioni "ambientali" (come variazione di pH e temperatura) sia per l'azione dei microrganismi che, una volta asportato il tessuto, immediatamente attaccano e invadono il materiale biologico.

Attraverso il processo di fissazione si riesce a ritardare la decomposizione e il deterioramento; perciò i tessuti prelevati vengono trattati con composti chimici come alcoli e aldeidi che ne impediscono la degenerazione [6].

Disidratazione: operazione propedeutica all'inclusione. Prima di procedere all'inclusione è necessario estrarre la componente acquosa utilizzando dell'alcol, anch'esso insolubile in paraffina: una volta allontanata l'acqua, si procede alla rimozione dell'alcol attraverso sostanze quali benzene o xilene; quest'ultimo processo è chiamato diafanizzazione, in quanto il tessuto immerso in xilene diventa trasparente [7].

Inclusione: i tessuti precedentemente asportati perdono la consistenza necessaria al loro sostegno; ciò che si fa è quindi inserirli in materiali più resistenti che possano fungere da sostegno. Nella microscopia elettronica lo spessore deve essere dell'ordine degli Angstrom e si utilizzano resine acriliche, fluide a temperatura ambiente e solidificate per mezzo di un agente polimerizzante (calore o raggi UV).

Sezionamento: affinché un tessuto possa essere osservato al microscopio ottico, deve essere abbastanza sottile da permettere alla luce di attraversarlo. I tessuti vengono perciò tagliati in sezioni sottilissime, tramite un microtomo o ultramicrotomo da microscopia elettronica.

I moderni microtomi permettono di ottenere sezioni di spessore non superiore ai 20-30 μm .

Il vantaggio di poter ottenere campioni di tessuto così sottili è che lo spessore di 20-30 μm è circa lo spessore di una cellula ed è possibile ottenere un campione in cui è presente un singolo strato di cellule, evitando così che la sovrapposizione di più strati cellulari possa disturbare la visione.

Colorazione: nella maggior parte dei casi i tessuti animali sono incolore (perché costituite in gran parte di acqua e prive di pigmenti) e trasparenti tanto da risultare pressoché invisibili al microscopio ottico; pertanto, fin dalla nascita dell'istologia scientifica, sono state scoperte e realizzate una serie di sostanze coloranti capaci di colorare le cellule, o le diverse parti di essa, in modo da renderle immediatamente visibili e distinguibili.

Sono note diverse sostanze di questo tipo, che in base ai meccanismi con cui si legano ai diversi componenti cellulari possono essere divise in due categorie:

- coloranti acidi: si legano alle molecole con caratteristiche basiche
- coloranti basici: si legano alle molecole con caratteristiche acide

Normalmente, nelle analisi istologiche vengono utilizzate coppie di coloranti basici/acidi che colorano in modo diverso le diverse parti cellulari: la più usata è la colorazione con ematossilina/eosina: l'ematossilina che è basica, colora il nucleo in blu mentre l'eosina, acida, colora il citoplasma in rosa.

Oltre ai coloranti tradizionali, negli ultimi anni si sono diffuse anche le tecniche della immunistochemica per individuare i diversi componenti cellulari: queste tecniche prevedono l'uso di anticorpi opportunamente trattati in modo da legarsi determinate proteine, lipidi o carboidrati.

1.3 Colorazione ematossilina–eosina (ee)

La colorazione Ematossilina-Eosina è il tipo di colorazione più usato in istologia grazie alla sua semplicità d'uso.

Questa colorazione consente di osservare e analizzare morfologicamente un determinato organo o tessuto di un qualsiasi organismo animale e identificare eventuali alterazioni patologiche.

Questo tipo di colorazione è definita bicromica o combinata poiché, vengono utilizzati i due coloranti più comuni in campo istologico, l'Eosina e l'Ematossilina. La colorazione ha le seguenti caratteristiche [7]:

- i nuclei assumono una colorazione che varia dal blu scuro al viola scuro
- il citoplasma, le sostanze intercellulari e le fibre muscolari, hanno una colorazione che varia dal rosa al rosso
- I fluidi presenti nei tessuti, o negli spazi interstiziali, si colorano debolmente, assumendo la forma di ampi spazi bianchi (sangue, linfa, etc.)
- I lipidi e i grassi non si colorano.

Nonostante il fatto che i vari coloranti si leghino opportunamente ai vari contenuti strutturali in un campione conferendo ai vari elementi quali nuclei, stroma e lipidi colorazioni caratteristiche, di fatto nella pratica il colore viene influenzato anche dallo spessore e dai tempi di esposizione del campione con i coloranti chimici creando una profonda differenza di tonalità tra vari campioni. Questo costituisce un fattore di disturbo per un patologo ed è un ostacolo non banale per un algoritmo automatico di detezione e classificazione dei tessuti.

1.4 Digital pathology

Negli ultimi anni si è notato con grande evidenza una correlazione elevata tra la qualità e la precocità della diagnosi con la bontà della prognosi. I patologi costituiscono quella figura professionale deputata a visionare i campioni e le immagini istologiche con l'arduo compito di classificare correttamente il tipo di tumore e associargli un grado di pericolosità. Una diagnosi accurata si traduce in un trattamento e una strategia di cura migliore con una conseguente prognosi più favorevole. Tuttavia, effettuare una buona diagnosi è estremamente difficoltoso e dispendioso per il patologo dal momento che occorrono moltissime ore di lavoro con una elevata concentrazione. Questo non sarebbe un problema se non fosse che a causa dell'elevata quantità di stress, del tipo di occupazione e del senso di responsabilità elevato ci sia un numero di patologi limitato rispetto alla domanda. Questo causa un inevitabile ritardo nell'effettuare le diagnosi con una conseguente e drammatica diminuzione della aspettativa di vita e peggioramento della prognosi del paziente.

Per questo con il passare degli anni la Digital pathology si è candidata come una possibile soluzione efficace e rapida per coadiuvare il patologo nell'analisi dei campioni istologici.

La Digital pathology si concentra sulla gestione dei dati in base alle informazioni generate dai vetrini dei campioni digitalizzati. Le *slice* di tessuto nei vetrini vengono convertite in *slice* digitali (WSI) che possono essere visualizzate, gestite, condivise e analizzate sul monitor di un computer. Questo produce alcuni chiari vantaggi immediati come la facile condivisione del materiale per ottenere secondi pareri (fondamentali in questo campo) o vantaggi più diretti come segmentazione di aree di interesse,

conteggio delle cellule (operazione che in condizioni normali è estremamente dispendiosa per il patologo) o classificazioni di intere porzioni di WSI.

L'approccio che si è rivelato più congeniale per questa applicazione è l'intelligenza artificiale in particolare il deep learning e le reti neurali convoluzionali. Questo lavoro di tesi in particolare si pone come obiettivo quello di costruire un sistema automatico per la classificazione delle immagini istologiche che possa essere utile ad un patologo per un'analisi più rapida e accurata dei tumori alla mammella.

1.5 Cos'è una rete neurale artificiale

Le reti neurali artificiali sono un popolare strumento di *machine learning* che cerca di simulare il fenomeno dell'apprendimento negli esseri viventi. Il sistema nervoso umano è composto da cellule connesse tra loro da assoni e dendriti che comunicano tramite sinapsi.

All'interno del cervello umano vi sono perciò miliardi di connessioni, tuttavia esse non sono tutte uguali tra loro ed il contributo di ciascuna connessione e la sua forza dipende dagli stimoli esterni. Esattamente come funziona per un cervello umano, in cui l'apprendimento di un'informazione è costituita dall'elaborazione di uno stimolo, per una ANN (*artificial neural network*) funziona allo stesso modo. Una rete artificiale si propone di imitare questo fenomeno tramite neuroni e pesi, esattamente come nel parallelo biologico i pesi di una rete sono molteplici e ciascuno può fornire un maggior o minor contributo.

In fase di apprendimento viene fornito un insieme di dati alla rete, ad ogni iterazione una porzione di questi dati (ad esempio un'immagine) attraverso

tutta l'architettura della rete fino a raggiungere *l'output*. *L'output* non è altro che una predizione che l'ANN effettua sulla base dei pesi che possiede in quel momento; la rete neurale successivamente confronta tale *output* con il *ground truth* associato a quei dati e calcola l'errore che ha compiuto nella predizione. In base all'errore commesso la rete cambia i pesi dei vari parametri (connessioni della rete) cercando di minimizzare l'errore. Addestrare una rete di questo tipo non è altro che un problema di minimizzazione in quanto ad ogni iterazione l'errore diventa sempre minore, e ad un minor errore corrispondono predizioni più accurate. [8]

Lo scopo di una rete neurale artificiale è quello di generalizzare, ovvero trasferire la conoscenza acquisita in fase di *training* su un certo set di dati visti su un altro insieme di dati dello stesso tipo, ma che la rete non ha mai analizzato. La bontà di una rete artificiale risiede prettamente nella sua capacità di generalizzare e quindi di effettuare giuste predizioni su dati ad essa sconosciuti.

Effettuare valutazioni sui dati utilizzati durante il *training* viene da sé che sia una considerazione priva di robustezza in quanto su quei dati la rete ha effettuato un'operazione di minimizzazione dell'errore.

Tuttavia, ottenere un modello che mostri buone *performance* su quelli che vengono definiti *unseen data* non è un compito banale, ma bensì ricco di ostacoli.

I maggiori aspetti che influenzano le *performance* di un modello sono:

- Numerosità dei dati
- Numero di parametri dell'architettura della rete
- Profondità del modello della rete

- Qualità e omogeneità dei dati del *training* rispetto ai dati di validazione e test
- Strategia di allenamento della rete

Ognuno di questi aspetti gioca un ruolo importante nell'esito finale delle performance della rete; tuttavia, prima di snocciolare nel profondo ognuno di queste problematiche, è necessario fare un *focus* sul campo applicativo in questione e sul tipo di tecnologia che si userà dal momento che, per diversi campi applicativi, questi aspetti meritano considerazioni differenti.

1.6 CNN (*Convolutional Neural Network*)

Entrando più nel dettaglio, il tipo di rete neurale utilizzato per la classificazione e il riconoscimento delle immagini è la CNN. Le reti neurali convoluzionali sono reti specializzate nel processamento di *input* composti da dati che presentano una struttura a griglia con forti dipendenze spaziali in alcune regioni locali. Un tipico esempio di informazione che presenta una organizzazione di questo tipo è un'immagine, poiché ogni *pixel* rappresenta un elemento di una griglia a due dimensioni.

Le CNN, come detto sopra, risultano molto performanti su dati con struttura a matrice; ciò è dovuto al fatto che al loro interno mantengono questa struttura spaziale grazie all'operazione di convoluzione. Una CNN è composta da un'alternanza di *layer* in cui ognuno di essi mantiene una struttura a tre dimensioni caratterizzata da altezza larghezza e profondità.

Il termine profondità in questo caso identifica il numero di canali di un *layer* e non va confuso con la profondità della rete.

Una CNN opera a grandi linee come una rete *feed-forward* eccetto per il fatto che le operazioni che svolge sono organizzate nello spazio e vi è una connessione meno consistente ma più significativa tra i vari *layer* rispetto alle reti tradizionali.

La CNN ha come operazione fulcro la convoluzione ma è composta anche da altri due *layer* fondamentali che sono il *layer* di *pooling* e il *layer ReLU*, oltre che ovviamente i *layer* di *input* e *output* [B]. Rispetto alle reti tradizionali l'uso delle operazioni di convoluzione e *pooling* garantisce alla rete di estrarre dalle immagini le caratteristiche essenziali e utili a classificare un'immagine.

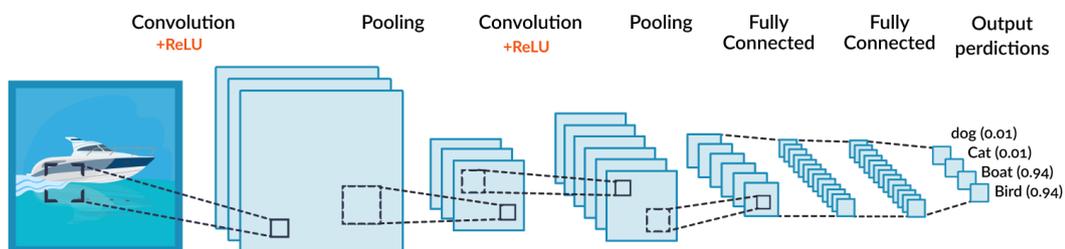


Fig.1.3 esempio di CNN completa

Tuttavia per quale motivo una *feed-forward neural net* sarebbe sconveniente?

Dal momento che un'immagine è una serie di *pixels*, si potrebbe pensare di appiattare tutti i *pixels* in un vettore e utilizzare questo gigantesco *array* in ingresso a una rete *perceptron* multilivello.

Questa strategia potrebbe andare bene per immagini bidimensionali, semplici e di piccole dimensioni. Per immagini di medie dimensioni (come ad esempio la dimensione che riceve in *input* una famosa rete convoluzionale chiamata *InceptionV3*) di $299 \times 299 \times 3$, tale strategia darebbe origine a un vettore di 268.203 elementi associato ad ogni immagine. Un approccio di questo tipo

sarebbe computazionalmente troppo dispendioso.

Le *convnet* hanno il grande pregio di diminuire notevolmente la dimensionalità del problema, senza diminuire le informazioni contenute in un'immagine.

Un ulteriore vantaggio delle CNN è che lasciando intatta la struttura a tre dimensioni del problema possono evidenziare dipendenze spaziali e *feature* nella disposizione dei *pixels* che andrebbero perse. Per concludere questa parte introduttiva, il compito di una CNN è di ridurre le immagini in una forma più facile da elaborare senza perdere contenuto informativo, ottenendo un metodo scalabile per gigantesche quantità di dati.

Supponiamo di dover classificare un'immagine codificata RGB di dimensione 299x299. Il primo *layer* che l'immagine incontra attraversando la rete è un *layer* convoluzionale; l'elemento coinvolto nello svolgimento dell'operazione di convoluzione nella prima parte di uno strato convoluzionale è chiamato kernel. Il kernel o filtro è una matrice che contiene i parametri allenabili dalla rete ed ha dimensioni nettamente inferiori rispetto a quelle dell'immagine. Il filtro scorre sull'immagine e ad ogni spostamento viene eseguita l'operazione di convoluzione.

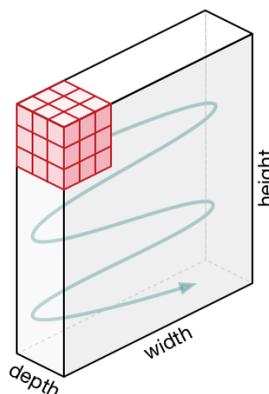


Fig.1.4 immagine illustrativa del movimento del Kernel [3]

Il kernel può spostarsi di un elemento alla volta o di più elementi, questo parametro viene chiamato *stride*.

Considerando per un momento un'immagine bidimensionale, l'operazione della convoluzione produce come risultato un'altra matrice chiamata "mappa di attivazione" che contiene le caratteristiche estratte in questo strato di convoluzione. I valori sono ottenuti come illustrato nell'immagine a seguire:

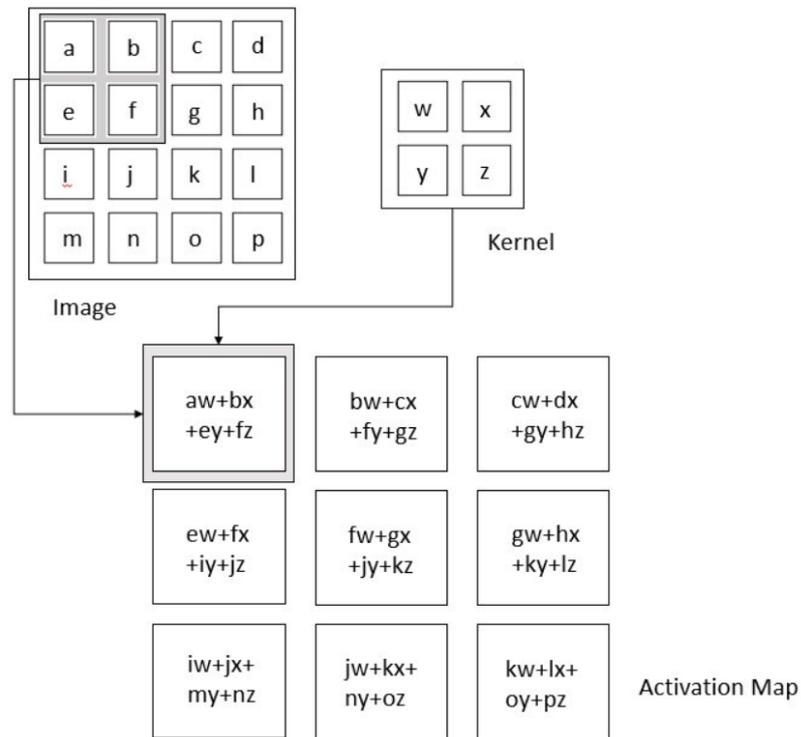


Fig.1.5 Meccanismo operazione di convoluzione [11]

Questa mappa di attivazione sarà l'*input* di altri *layer* di convoluzione che estrarranno caratteristiche di più alto livello, utili ai fini della comprensione profonda dell'immagine.

A seconda delle dimensioni della matrice in *input* W (nel primo *layer* l'immagine, in quelli successivi le mappe di attivazione), della dimensione del kernel F , dello *stride* S e del *padding* P , viene definita la dimensione della mappa di attivazione in uscita secondo la formula (eq.1.1):

$$W_{\text{out}} = \frac{W - F + 2P}{S} + 1 \quad \text{eq.1.1}$$

Normalmente per ogni *layer* convoluzionale non viene applicato un solo filtro. Supponendo di avere un'immagine (in questo caso bidimensionale) di 299×299 applicando un filtro di dimensione 5×5 con *stride* pari a 1 e *padding* ai bordi nullo si ottiene, secondo la formula sopra, una mappa di dimensione 294×294 .

Se invece applicassimo un numero di filtri pari a d_f otterremo d_f mappe di attivazione, dunque concatenandole avremmo una dimensione $294 \times 294 \times d_f$. Si dice che il *layer* ha profondità pari a d_f .

Estendendo il discorso al caso di immagini RGB non cambia pressochè nulla, se non che i filtri al posto di essere bidimensionali sono volumetti con 3 canali sulla terza dimensione.

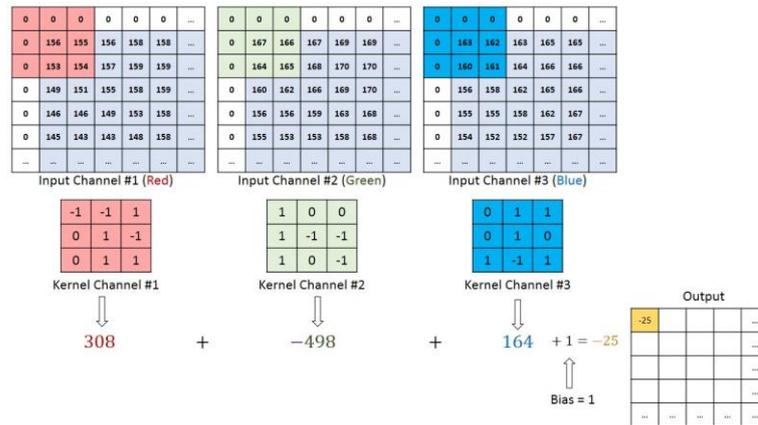


Fig.1.6 immagine illustrativa dell'operazione di convoluzione su un'immagine RGB [10]

Il fatto che si abbiano 3 dimensioni non influisce sulla profondità dei vari *layer*; ad ogni filtro corrisponde la produzione di una sola mappa di attivazione. Per ottenere una profondità maggiore di uno in un certo strato devo necessariamente applicare più filtri.

1.6.1 Padding

La riduzione di dimensionalità, nonostante sia un vantaggio enorme che caratterizza il successo delle CNN in ambito di *deep learning*, porta con sé uno svantaggio. Nell'operazione di convoluzione con la riduzione delle dimensioni si ottiene anche una perdita dell'informazione ai bordi; per limitare questo effetto si esegue una operazione di *padding* ai bordi dell'immagine o della mappa di attivazione; il *padding* consiste nell'aggiungere un certo numero di zeri intorno ai bordi così che l'operazione di convoluzione non elimini troppo il contributo informativo che si trova ai confini dell'immagine. [9]

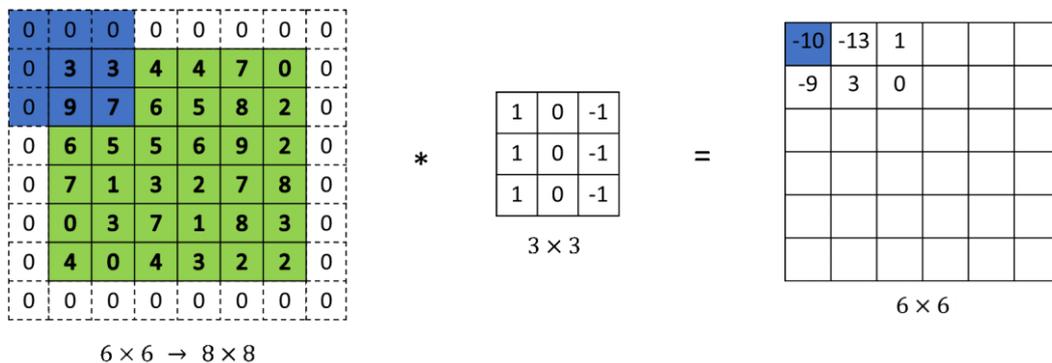


Fig.1.7 esempio di padding su un'immagine o mappa di attivazione

Fonte: <http://datahacker.rs/what-is-padding-cnn/>

1.6.2 Pooling

Nel momento in cui si applica la convoluzione, si ha già una riduzione della dimensionalità che diventa tanto più consistente all'aumentare dello *stride*. La funzione deputata puramente alla riduzione della dimensionalità delle mappe di attivazione è pertanto l'operazione di *pooling*.

Il *layer di pooling* serve per diminuire la potenza di calcolo richiesta per elaborare i dati attraverso la riduzione della dimensionalità. Inoltre, è utile per estrarre le caratteristiche dominanti che sono invarianti rotazionali e posizionali, così che il processo di allenamento risulti efficace e la rete possa riconoscere caratteristiche geometriche base, o più profonde, anche in condizioni differenti.

Come per l'operazione di convoluzione, anche l'operazione di *pooling* necessita dell'utilizzo di un kernel che scorra sulla mappa di attivazione. Principalmente esistono due tipi di *pooling*: il *Max Pooling* che restituisce il valore massimo dalla porzione della matrice coperta dal kernel e l'*Average Pooling* che invece fornisce un valore medio pesato sulla porzione di matrice su cui si trova il kernel.

Se si ha una mappa di attivazione di dimensione $W \times W$, un kernel di *pooling* di dimensione F e stride S , la dimensione del volume di *output* può essere determinata dalla seguente formula (eq.1.2):

$$W_{\text{out}} = \frac{W-F}{S} + 1 \quad \text{eq.1.2}$$

Il tipo di *pooling* più utilizzato solitamente è il *Max Pooling* in quanto funge anche da soppressore del rumore eliminando completamente le caratteristiche

rumorose; nell' *Average* invece queste hanno un peso nella restituzione del valore finale.

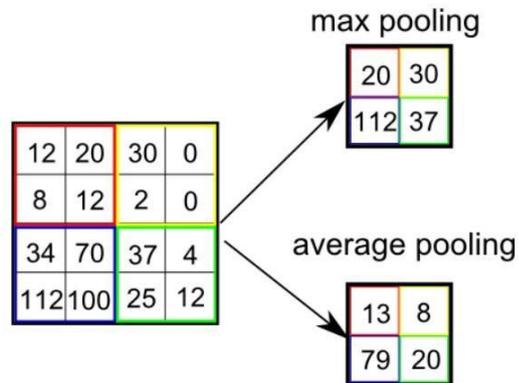


Fig.1.8 differenti tipi di pooling [10]

Lo strato convoluzionale e lo strato di *pooling*, insieme formano l'i-esimo strato di una rete neurale convoluzionale [10]. Questo blocco può essere ripetuto in maniera modulare formando reti più o meno profonde.

In reti più profonde vengono apprese caratteristiche più complesse, tuttavia queste potrebbero essere troppo caratterizzanti per immagini semplici e quindi il rischio è di ottenere una rete poco performante su dati di test, oltre che ad aumentare notevolmente la potenza computazionale.

1.6.3 Fully-connected

In questo strato, che viene posto appena prima del *layer* di classificazione, vengono posizionati dei neuroni completamente connessi con tutte le mappe di attivazione dello strato precedente, esattamente come accade in una *fully connected neural network* tradizionale.

Questo passaggio serve ad introdurre un contributo di non linearità.

1.6.4 Classificazione

Nell'ultimo *layer* della rete viene posto il livello di classificazione. Lo strato è composto da un numero di neuroni pari al numero di classi della *task* completamente connessi allo strato precedente e forniscono in uscita una percentuale di appartenenza a ogni classe.

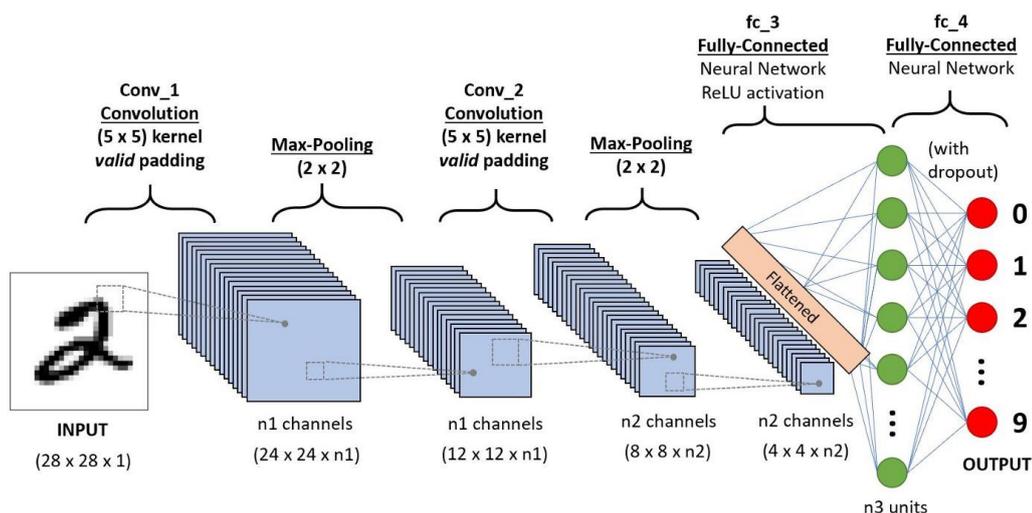


Fig.1.9 esempio di CNN completa [10]

1.7 Analisi stato dell'arte

I recenti successi delle CNN in diversi ambiti come la computer vision o per gli strumenti di visione artificiale per il riconoscimento automatico di oggetti hanno dato una importante spinta anche al settore BHIA (*Breast Histopatological Image Analysis*). Nel momento in cui si consulta la letteratura per avere una visione dei metodi utilizzati per sviluppare un sistema automatico per il *detect* dei tumori in immagini istologiche è evidente come la tecnologia delle CNN sia il pilastro sopra il quale questi software vengono costruiti [12].

Per quanti riguarda il *task* della classificazione dei tumori in istologia si possono individuare centinaia di *paper* sull'argomento, ognuno dei quali si distingue per la metodologia applicata alle immagini che si hanno a disposizione. Le prestazioni finali sono influenzate da molte scelte che riguardano sia come vengono pre-processate le immagini e sia quali architetture scegliere e come esse vengono allenate.

Uno dei primi studi nella classificazione è stato quello svolto da Araùjo *et al.* [13] che con un classificatore *custom made* basato su CNN ha ottenuto un'accuratezza di classificazione del 77%.

Nello studio di Chennamsetty *et al.* [14] si afferma che non esiste un architettura di CNN o un algoritmo di pre-processing che permette di ottenere le migliori performance in assoluto. Per questo motivo viene presentato un metodo che propone di combinare le uscite di più CNN per ottenere una classificazione finale migliore rispetto alle reti prese singolarmente.

Entrando più nel dettaglio, l'autore decide di utilizzare le immagini del dataset reso disponibile dalla ICIAR2018 *grand challenge* nel loro formato originale, effettuando prima un *padding* per eguagliare le due dimensioni, e poi effettuando un *resize* riducendole alla grandezza di 224x224 pixels.

In seguito, applica come pre-processing alle immagini un algoritmo di normalizzazione chiamato z-score. Questo algoritmo consiste nel sottrarre all'immagine il valor medio dei pixel di un certo dataset e dividere per la deviazione standard dei valori dei pixel dello stesso dataset. Vengono successivamente generati due differenti dataset originati dall'applicazione della normalizzazione z-score sul dataset BACH, utilizzando come valori di media e deviazione standard i valori ricavati dai dataset Imagenet e dal dataset di train della BACH challenge.

Per creare la struttura combinata delle reti sono state allenate tre reti: Densenet161 su entrambi i dataset normalizzati e ResNet-101 solo sul dataset normalizzato Imagenet. Sono state utilizzate reti pre-allenate alle quali è stato modificato l'ultimo layer rimuovendo i 1000 neuroni corrispondenti alle 1000 classi dell'Imagenet e sostituendoli con un dense layer con 4 neuroni.

La struttura del classificatore finale è illustrata nell'immagine a seguire:

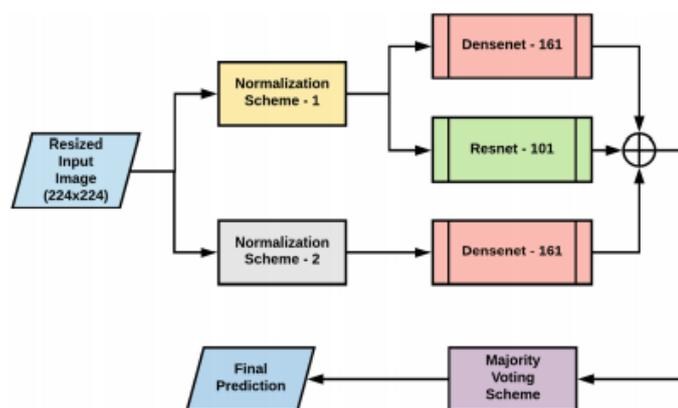


Fig.1.10 pipeline di testing proposta da Chennamsetty et al [14]

La struttura proposta ha ottenuto un'accuratezza del 97,5% sul test set estratto dal training set, mentre ha ottenuto un'accuratezza dell'87% sul vero test set della competizione.

Un approccio differente è stato utilizzato da Roy *et al.* [15], che ha proposto un classificatore basato su patch estratte dalle immagini della BACH challenge. Nell'articolo pubblicato sostiene che in situazioni in cui il numero di immagini è limitato, l'estrazione di patch dalle immagini consente di aumentare notevolmente il numero di campioni da utilizzare per

l'allenamento della rete. La pipeline proposta inizia con l'applicazione sull'intero dataset dell'algoritmo di Macenko [16], che è una tecnica di normalizzazione della colorazione, prosegue con l'estrazione delle patch dalle immagini (viene estratta ogni porzione di ogni immagine), applicazione del *data augmentation* (rotazione a step di 45°), allenamento di un'architettura CNN *custom-made* e infine valutazione delle performance della rete.

Per quanto riguarda la costruzione della rete custom sono state valutate molte architetture ed è stata scelta quella con le performance migliori; una simile analisi è stata svolta anche sulla dimensione delle patch da utilizzare. I risultati ottenuti mostrano come l'accuratezza del classificatore dipenda molto dalla grandezza delle patch: per patch di 512x512 pixels è stato registrato un *learning accuracy* dell'85.8% mentre scendendo con la dimensione a 128x128 e 64x64 si ottengono performance peggiori con accuratezze rispettivamente del 58.8% e 51%.

Al termine dell'allenamento con patch 512x512 si ottiene una rete che determina lo score di appartenenza a una classe per ogni patch. Per ottenere la classificazione *image-wise* sono stati provati diversi metodi, il più performante è stato il *majority voting*. Ogni patch appartenente ad una immagine concorre con il suo *label* alla classificazione dell'immagine di appartenenza secondo il criterio di maggioranza. L'accuratezza registrata a livello patch è stata del 77% ; mentre sul test set splittato dal train set a livello *image-wise* si è raggiunto il 90%. Successivamente è stato valutato il metodo anche sul test set della competizione registrando un'accuratezza dell'87%.

In un differente studio Awan *et al.* [17] mostra come l'utilizzo delle patch di dimensione 512x512 ma con un *overlap* del 50% permettano di aumentare ulteriormente il dataset. In questo caso è stato utilizzato l'algoritmo di Reinhard per la normalizzazione dello *stain* delle immagini del dataset. La scelta sull'architettura della rete è ricaduta sulla ResNet50 preallentata sul dataset Imagenet, in questo caso però la classificazione non avviene a valle della rete. L'autore ha utilizzato la ResNet50 per ricavare le *features* legate alle patch, successivamente ha applicato a queste una PCA per poi effettuare la classificazione vera e propria con un classificatore SVM. In fine, sono stati ottenuti i label *image-wise* applicando il majority voting agli score delle patch.

A differenza degli studi riportati precedentemente è stato scelto di effettuare data augmentation durante le varie iterazioni dell'allenamento della rete. Le accuratze registrate *image-wise* in questo lavoro di ricerca sono state del 90% sul test ottenuto dal dataset di train, mentre è stato ottenuto un valore di 75% su un dataset di test esterno.

Una strategia particolarmente interessante è stata adottata da Golatkar *et al.* [18], secondo cui si deve estrarre le patch solo nelle regioni a più alta densità di nuclei eliminando quindi le zone con pochi nuclei o che rappresentano tessuto non epiteliale. In questo modo si scartano le zone prive di informazioni utili ai fini della classificazione. Sempre secondo l'autore l'utilizzo delle immagini per intero (a cui però si applica *resize*) comporterebbe l'incapacità della rete di generalizzare dato il numero limitato di elementi nel dataset di train. Vengono quindi estratte le patch con un *overlap* del 50% e con dimensione 299x299 pixels. Tale scelta è giustificata dal fatto che l'autore utilizza come rete per la classificazione l'architettura preallentata Inception-V3 che richiede come input-size immagini di

dimensione 299x299. Scegliendo patch di tali dimensioni si evita l'utilizzo di algoritmi di *resize* che possono causare errori dovuti alle interpolazioni bilineari o cubiche alla base degli algoritmi di riscalamento. Come detto in precedenza però non tutte le patch vengono selezionate; la selezione viene effettuata confrontando i canali del blu e del rosso della codifica RGB. Siccome la colorazione ematossilina-eosina tende a colorare di blu i nuclei e di violaceo lo stroma è possibile, tramite un rapporto soglia ricavato empiricamente, ottenere una maschera dei nuclei da utilizzare per la selezione delle patch. A valle della selezione viene effettuato data augmentation sulle patch aumentando così la numerosità del dataset di train di 8 volte. Infine, la rete viene allenata tramite il metodo del fine-tuning e la classificazione *image-wise* viene effettuata tramite *majority voting*. Il classificatore ha registrato un'accuratezza a livello patch del 79% e dell'85% a livello *image-wise* su immagini del train set utilizzate come test set.

Da questa analisi dello stato dell'arte emerge come siano molti gli aspetti determinanti che incidono sul risultato finale e sulle performance del classificatore. I punti critici a cui la maggior parte degli autori dedica attenzione sono sia problemi comuni nell'ambito della classificazione delle immagini istologiche come la variabilità nella colorazione, sia problemi che affliggono in particolare questo dataset come la numerosità ridotta delle immagini con conseguenti problemi di overfitting [19] o la elevata omogeneità dei dati del train rispetto al dataset di test fornito dalla competizione. In questo lavoro di tesi viene proposto un metodo che pone molta attenzione sul miglioramento della qualità dei dati da fornire al classificatore, in quanto si ritiene che insistendo su un pre-processing accurato si possano raggiungere performance migliori. Il *deep learning* e le reti CNN per classificazione sono strumenti con un potenziale enorme ma

purtroppo poco manipolabili direttamente. L'obiettivo di questa tesi, oltre che di ottenere buone performance sul dataset BACH, è quello di dimostrare che con un meticoloso e strutturato *pre-processing* e *post processing* è possibile guidare il classificatore verso prestazioni superiori.

CAPITOLO II

MATERIALI E METODI

2.1 Descrizione Dataset

In questo lavoro di tesi si è deciso di utilizzare un dataset pubblico in modo tale da poter confrontare l'accuratezza del nostro algoritmo con i valori presenti in letteratura. In particolare, la scelta è ricaduta sul dataset BACH2018 utilizzato nel corso di una challenge, ciò ha permesso di poter effettuare un confronto anche sulle prestazioni riguardanti il test set.

Il dataset è composto da due parti, un train set e un test set. Il train set è formato da 400 immagini di dimensione (2048×1536) etichettate con quattro classi (normale, benigno, invasive e in situ) mentre il test set è composto da 100 immagini di cui non si conosce la classe. In fase di addestramento della rete è necessario l'utilizzo di un validation set con la funzione di monitorare l'accuratezza della rete durante il suo allenamento. Il validation set è stato costruito prendendo in modalità casuale cinque immagini per ogni classe dal dataset di train.

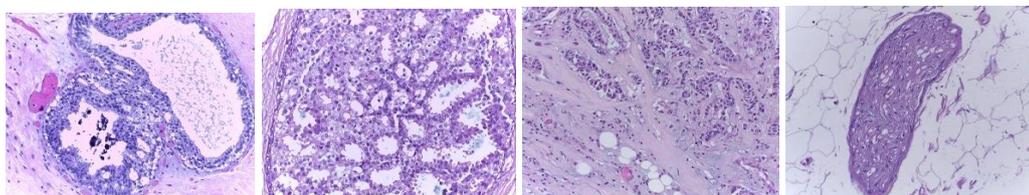


Fig.2.1 Da sinistra a destra un immagine Benigna, Insitu, Invasiva, Normale del dataset di train della BACH challenge

Di seguito viene riportata una tabella in cui si mostra la suddivisione dei dati:

	TRAIN	VALIDATION	TEST
Normal	95	5	?
Benign	95	5	?
InSitu	95	5	?
Invasive	95	5	?
totale	380	20	100

I label del test set sono sconosciuti poiché oggetto di valutazione delle prestazioni dei classificatori iscritti alla competizione ICIAR2018. Anche successivamente al rilascio della classifica i label non sono stati resi pubblici. Questo è il motivo per cui la maggior parte degli autori utilizza come test set una parte delle immagini del training set. Bisogna però tenere in conto che il training set della BACH è un dataset molto omogeneo, perciò le prestazioni ottenute sul test set come *split* dal train possono essere molto più elevate rispetto a quelle che si otterrebbero sul vero test set. Per effettuare un confronto consistente con la classifica stilata a seguito della chiusura della BACH challenge è stato chiesto ad un patologo di classificare le 100 immagini del test set della competizione. In questo modo il confronto delle prestazioni verrà effettuato solo rispetto alla classifica stilata a seguito del termine della *challenge*.

Di seguito viene illustrato nei dettagli l'intero pre-processing, attraverso il quale si normalizza lo stain delle immagini e si effettua la *smart extraction*.

2.2 Stain Normalization

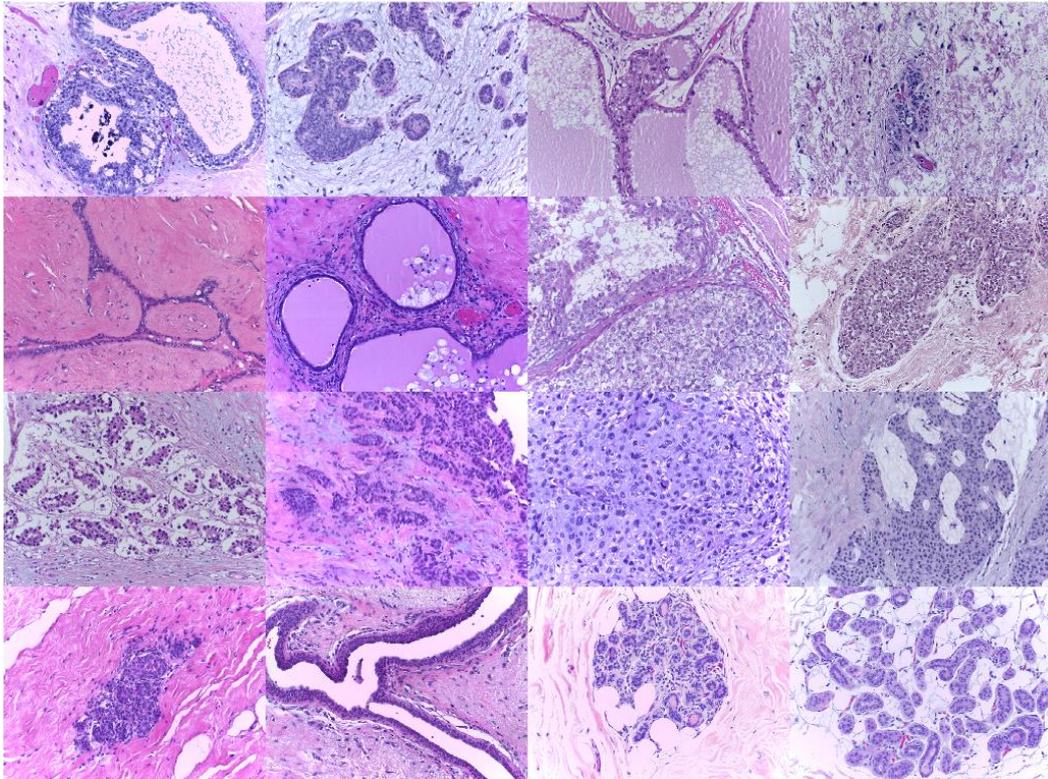
Dal momento che si lavora con immagini istologiche provenienti da WSI differenti, queste presentano colorazioni molto diverse tra loro; ciò rappresenta sicuramente un elemento di disturbo per il patologo e potrebbe rappresentare, al fine di classificare le immagini istologiche, un elemento deleterio anche per l'intelligenza artificiale.

In letteratura è possibile trovare diversi algoritmi di normalizzazione da utilizzare per ridurre la differenza di colorazione tra le diverse immagini che costituiscono il dataset, tuttavia, all'interno della comunità scientifica, non si ritiene di aver ancora trovato una soluzione efficace a questo tipo di problema.

Questo aspetto non rappresenta un semplice dettaglio ma un elemento chiave per ottenere un classificatore accurato; immaginare che una rete convoluzionale possa apprendere caratteristiche tali da poter bypassare questa problematica è in parte corretto: sicuramente si può affermare che anche in mancanza di uno step di normalizzazione il classificatore può ottenere buone performance, tuttavia, è altrettanto logico che una maggiore uniformità porta a risultati migliori. Questo problema è strettamente legato al livello di uniformità nella colorazione delle immagini che si hanno a disposizione, tanto più esse sono differenti nello stain, tanto più sarà marcato il contributo che si ottiene da uno step di normalizzazione.

Il dataset della BACH challenge contiene una discreta variabilità di stain tra le varie immagini, come mostrato di seguito nella figura Fig.2.2

Sebbene gli algoritmi maggiormente utilizzati in letteratura sono Macenko, Vahadane [20], e Reinhard [21], in questo lavoro di tesi è stato scelto un approccio più promettente e innovativo: le reti GAN.



*Fig.2.2 rappresentazione della variabilità della colorazione all'interno del dataset
BACH*

2.2.1 Cos'è una rete GAN?

Le reti GAN (Generative Adversarial Networks) sono un modello di generazione di dati capace di generare dati sintetici addestrando due reti in maniera competitiva. La rete chiamata generatore prende in ingresso del

rumore Gaussiano e produce un output sintetico che è il più possibile simile ai dati forniti al generatore come target. La rete discriminatore, il quale tipicamente è un classificatore probabilistico, ha il compito di distinguere i dati sintetici dai dati reali.

Il generatore tenta così di costruire dati sintetici che sembrano più reali possibili, cercando di ingannare il discriminatore che invece tenta di riconoscere il generatore individuandone i dati artificiali. In conclusione, si può affermare che l'obiettivo dell'addestramento della rete generativa è di aumentare l'errore della rete discriminativa al punto tale che essa non riesca più a distinguere dati reali da artificiali; quando ciò avviene allora la GAN ha raggiunto un punto di equilibrio soddisfacente [22].

2.2.1.1 Cycle-GAN

Le GAN sono una grande categoria di cui fanno parte modelli leggermente diversi tra loro, ognuno con i propri punti di forza. Il processo di normalizzazione delle immagini può essere riconducibile a un trasferimento di stile da applicare a un insieme di immagini rispetto a un altro insieme di immagini che hanno una colorazione ritenuta ottimale. Per assolvere a questo compito lo strumento più robusto all'interno di tutta la famiglia delle GAN è la cycle-GAN.

La cycle-GAN è uno strumento particolarmente abile nel riuscire a catturare le caratteristiche di un insieme di immagini e capire come queste caratteristiche potrebbero essere trasferite nell'altro insieme di immagini, il

tutto in assenza di immagini accoppiate. Di seguito viene brevemente illustrato il motivo della scelta:

Supponiamo di avere due domini di immagini X e Y e di voler allenare la GAN per applicare una trasformazione $G: X \rightarrow Y$ tale che l'output $\hat{y} = G(x)$ con $x \in X$ sia indistinguibile da $y \in Y$. Quello che una normale GAN cerca di fare è trovare la trasformazione G ottimale al fine di generare un dominio \hat{Y} la cui distribuzione di dati sia il più possibile simile a Y [23].

Tale trasformazione non garantisce tuttavia che un singolo input x e un output y siano accoppiati in modo significativo; ciò porta al noto problema chiamato mode collapse, dove tutte le immagini di input generano la stessa immagine di uscita e l'ottimizzazione non riesce a fare progressi.

Se un generatore produce un output particolarmente plausibile, il generatore può imparare a produrre solo quell'output o un piccolo insieme di output.

Questo problema viene dunque evitato nelle cycle-gan aggiungendo un generatore e un discriminatore che si occupano del processo inverso, quindi ad ogni iterazione si hanno due generatori e due discriminatori che vengono allenati simultaneamente in questo modo:

- $G: X \rightarrow Y$ tale che l'output $\hat{y} = G(x)$ con $x \in X$ sia indistinguibile da $y \in Y$
- $F: Y \rightarrow X$ tale che l'output $\hat{x} = F(y)$ con $y \in Y$ sia indistinguibile da $x \in X$

Il valore aggiunto delle cycle-GAN non è solo quello di svolgere l'operazione inversa della trasformazione, ma anche quella di valutare la consistenza

dell'intero ciclo, calcolando l'errore che intercorre tra le immagini sintetiche e quelle originali e dei rispettivi domini.

Se le trasformazioni F e G sono robuste nel svolgere il loro compito allora deve valere che:

- $F(G(x)) \approx x$ forward cycle-consistency loss
- $G(F(y)) \approx y$ backward cycle-consistency loss

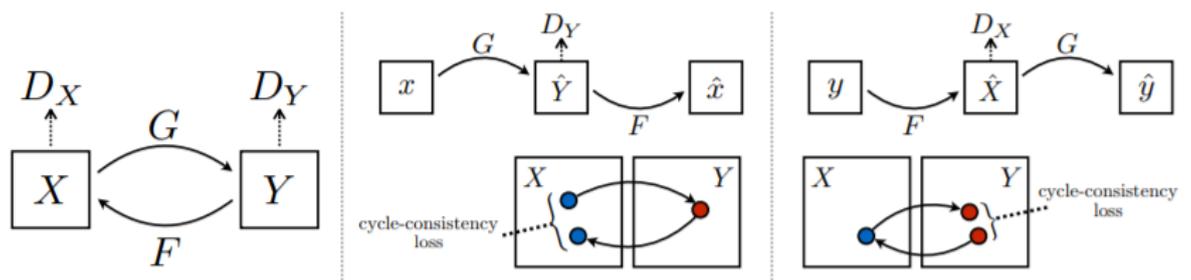


Fig.2.3 schema di funzionamento della cycle-GAN [23]

2.2.1.2 Allenamento Cycle-GAN

Dopo una breve introduzione teorica in merito a questa tecnologia relativamente nuova, illustrerò di seguito come questa è stata applicata in questo ambito e con queste immagini.

Prima di tutto è necessario costruire i due domini X e Y . Lo scopo di questo step di preprocessing è quello di avere un dataset il più omogeneo possibile; ciò che si deve ottenere è dunque una trasformazione in cui, una qualunque immagine istologica con un qualsiasi stain in ingresso, produce la stessa immagine in uscita con la colorazione voluta. Il dominio Y contiene dunque tutte le immagini target (con lo stain voluto) mentre il dominio X tutte le altre immagini del dataset.

E' molto importante che questo processo non modifichi il contenuto dell'immagine e che non riduca l'informazione contenuta in essa, altrimenti il classificatore finale sarà ostacolato anziché aiutato da questo processo di normalizzazione. Questo aspetto tuttavia non è semplice da gestire poiché può essere rispettato solo se nel dominio Y sono presenti immagini che siano rappresentative del dominio X.

La cycle-GAN produce buoni risultati anche con dataset non accoppiati in cui tutte le immagini contenute in X sono diverse dalle immagini in Y; tuttavia il contenuto informativo legato alle strutture anatomiche deve essere presente in entrambi i domini. Se ad esempio in X ho immagini con molti nuclei e in Y solo immagini con pochi nuclei, la cycle-GAN nel tentare di trasferire lo stile di Y a X andrà a forzare la trasformazione elidendo i nuclei meno marcati.

Affinchè la GAN non forzi troppo la trasformazione è necessario che i due domini siano fortemente correlati e che in Y sia presente un'ampia casistica di ciò che è raffigurato nelle immagini in X. Questo aspetto è fondamentale ed è anche difficilmente rispettabile quando si dispone di un numero limitato di immagini. Nel mio caso sono state necessarie molte prove per riuscire a capire come ottenere una cycle-GAN performante senza che essa modificasse il contenuto strutturale dell'immagine.

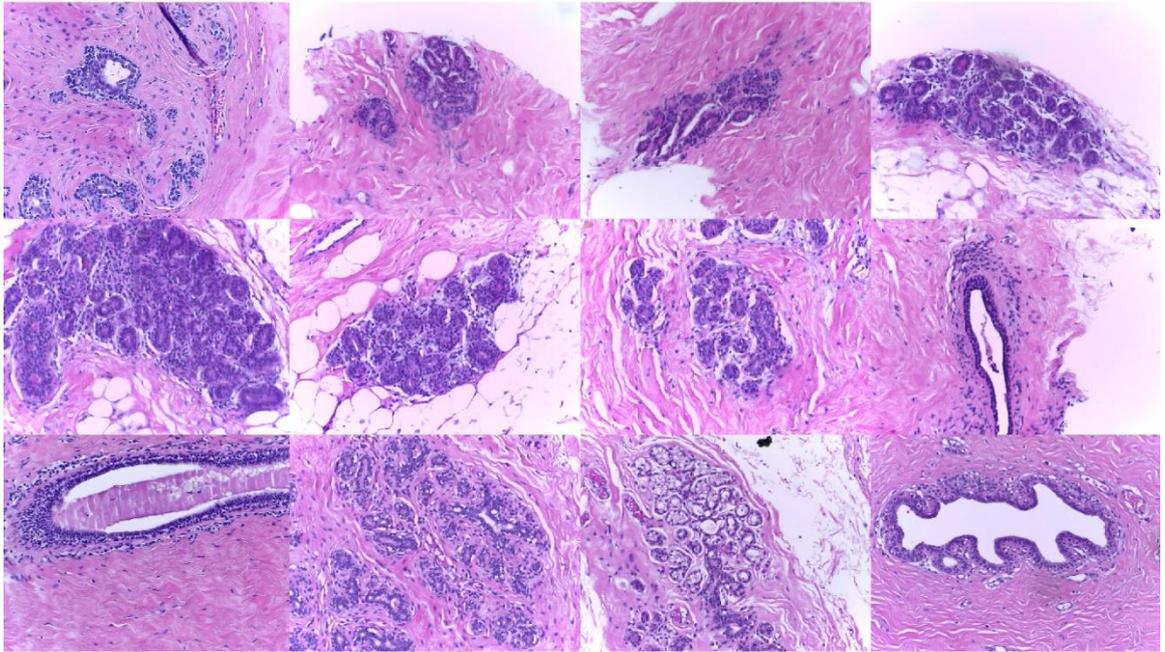


Fig.2.4 immagine che mostra le immagini del dataset BACH che fanno parte del dominio Y (target)

Una volta costruiti i due domini è possibile procedere con l'allenamento della rete. In questo caso le immagini sono state suddivise in patch più piccole al fine di aumentare il numero di immagini dei due domini e quindi di costruire una rete più robusta; oltre a ciò, immagini di grosse dimensioni, non sarebbero potute essere allenate per intero anche a causa di motivi tecnici. La cycle-GAN è un modello che richiede molta potenza computazionale e molto spazio di memoria grafica e, avendo a disposizione una NVIDIA 2070, sarebbe comunque stato necessario suddividere l'immagine in patch più piccole.

Le patch sono state estratte facendo scorrere una finestra di dimensione 512x512 pixels sull'intera immagine con un overlap del 50%; per ciascuna immagine sono state quindi estratte 35 patch.

Riassumendo, i due domini X e Y contengono le patch estratte dalle immagini appartenenti al dataset di train della bach challenge. In Y sono state inserite le patch delle immagini con lo stain desiderato che si vorrebbe trasferire sull'intero dataset, mentre in X tutte le patch delle immagini rimanenti. All'interno del dominio X durante l'allenamento non sono state inserite immagini del test set della challenge.

2.2.1.3 Quando terminare l'allenamento

Comprendere il momento in cui terminare un allenamento di una cycle-gan non è semplice, in quanto gli unici parametri che possiamo valutare sono le varie loss che vengono calcolate durante le diverse iterazioni; tuttavia dal momento che la GAN applica stain transfer tra immagini che possono essere molto differenti, le loss non potranno mai convergere a un valore molto piccolo o terminare in un plateau. Ciò che si verifica è un andamento oscillatorio delle varie loss intorno a un certo valore che progressivamente diminuisce fino a che la GAN non trova un punto di equilibrio.

2.2.2 Ricostruzione Immagine Normalizzata

Una volta terminato l'allenamento della cycle-GAN si dispone di un modello in grado di applicare trasformazioni a patch o immagini di 512x512 pixels; pertanto non si ha la possibilità di normalizzare per intero le immagini del dataset (dimensione 2048x1536 pixels).

Se si vuole ottenere l'immagine originale da 2048x1536 pixels normalizzata bisogna ottenere le patch suddividendo l'immagine, applicare la cycle-GAN a ognuna e ricomporle nel giusto ordine. Questa procedura risulta tuttavia

lenta e macchinosa, oltre che parzialmente inefficace. Nella procedura di ricomposizione dell'immagine si verificano artefatti nei punti di congiunzione delle patch dovuti a difetti che la GAN genera nei bordi.

A questo proposito è stata utilizzata una strategia per ottenere una ricostruzione di buona qualità [24]:

Step 1: generazione immagine estesa

Il primo passo di questa strategia prevede la generazione di una immagine di dimensioni maggiori estendendola ai bordi applicando un padding di dimensione opportuna. In questo caso il padding applicato è un mirroring di 256 pixel su ogni bordo. La nuova immagine estesa ha una dimensione pari a 2560x2048 pixel mentre quella di partenza era di 2048x1536 pixel.

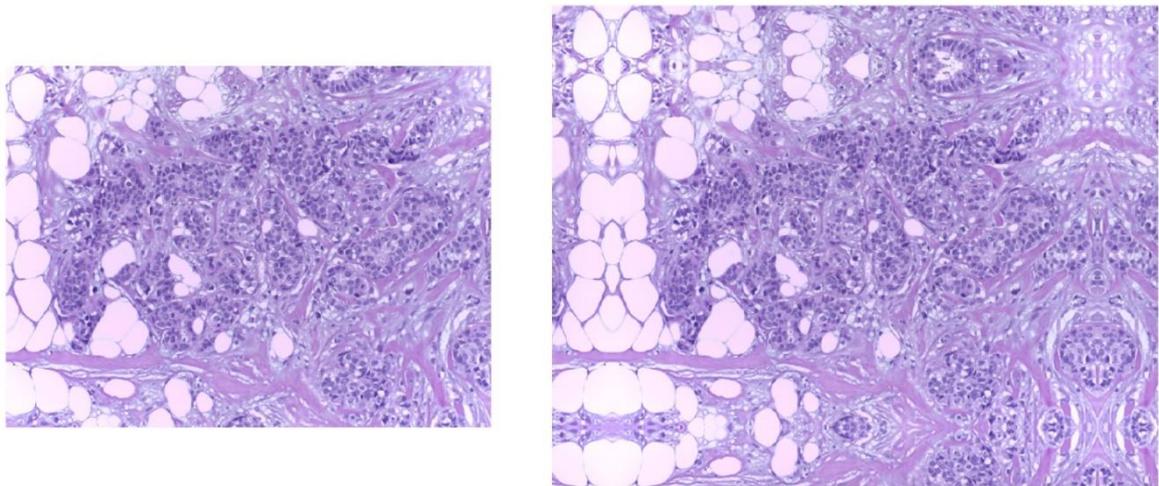


Fig.2.5 generazione dell'immagine estesa

Step 2: applicazione cycle-gan online e crop della parte centrale

Come detto in precedenza, il modello a disposizione è applicabile per patch di 512x512 pixels pertanto si genera una finestra di dimensione 512x512 che scorre sull'immagine estesa con un overlap del 50%. Ad ogni spostamento della finestra viene effettuato un crop della porzione di immagine in cui si trova, sulla quale viene applicata la cycle-GAN. Una volta applicata la trasformazione viene effettuato un crop della parte centrale di dimensione 256x256 pixels che verrà usata per costruire la nuova immagine normalizzata. L'immagine estesa non viene alterata in questo processo, tuttavia per dare un supporto visivo ho deciso di riportare alcune immagini di debug che possano aiutare a chiarire come opera questo algoritmo di ricostruzione. Nell'immagine a seguire (Fig.2.5) viene mostrata in arancione la finestra scorrevole che scorre sull'immagine, mentre in rosso è stata evidenziata la porzione di immagine che è stata estratta e normalizzata.

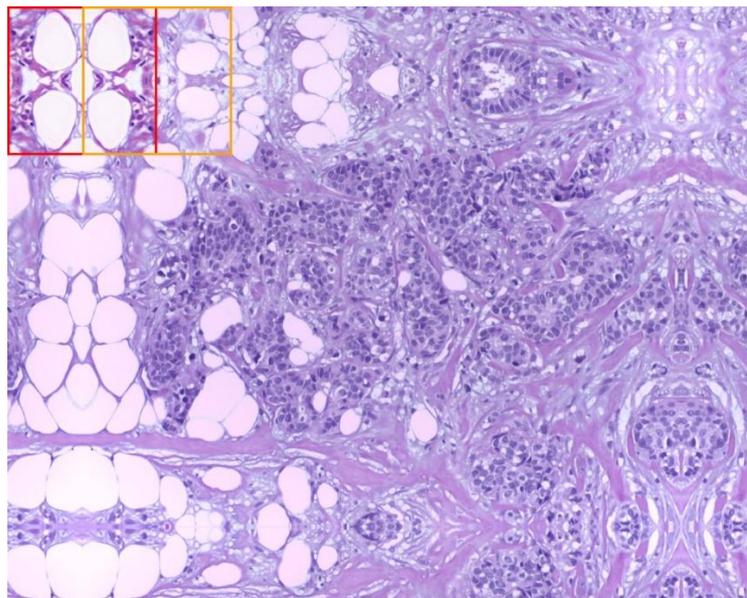


Fig.2.6 processo di normalizzazione dell'immagine

Come detto in precedenza, di ogni patch normalizzata viene stratta solo la parte centrale di dimensione 256x256 (parte in verde nell'immagine), in questo modo viene scartata la porzione di immagine che può dar origine a difetti ai bordi.

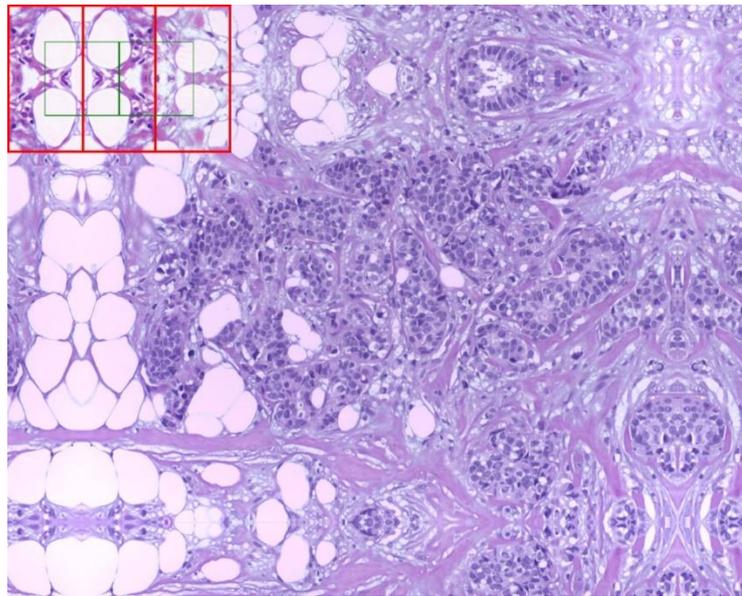


Fig.2.7 crop della parte centrale delle patch normalizzate

Quando la finestra scorrevole ha terminato, l'immagine ricostruita con le parti centrali delle patch normalizzate appare come nella figura Fig.2.8

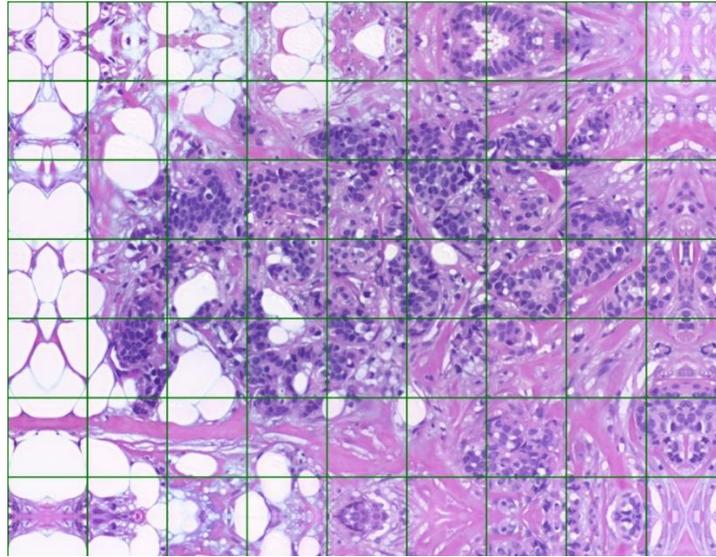


Fig.2.8 immagine normalizzata formata dalle parti centrali delle patch

Come si può notare non ci sono segni di discontinuità o difetti dovuti al fatto di aver ricompattato l'immagine a partire da pezzi molto più piccoli. Risulta evidente però che l'immagine mostrata sopra ha ancora una parte di tessuto in eccesso. Ciò è dovuto al fatto che effettuando un padding attraverso una simmetria ai bordi di un valore pari a N pixels e prendendo di ogni finestra solo la parte centrale, manteniamo un valore in eccesso pari a $N/2$ pixels.

L'immagine originale aveva una dimensione pari a 2048×1536 pixels, in seguito al processo di padding l'immagine estesa aveva una dimensione pari a 2560×2048 pixels in quanto sono stati aggiunti 256 pixels ad ogni bordo. L'immagine ha una dimensione di 2304×1792 pixels discostandosi dall'originale di 128 pixels su ogni bordo.

L'ultimo step di questo processo di ricostruzione è quello di eliminare tale eccesso; in questa ultima fase viene eliminata la parte rimanente del padding iniziale pari a $N/2$ pixels con N pari al pad-size.

Risultato finale:

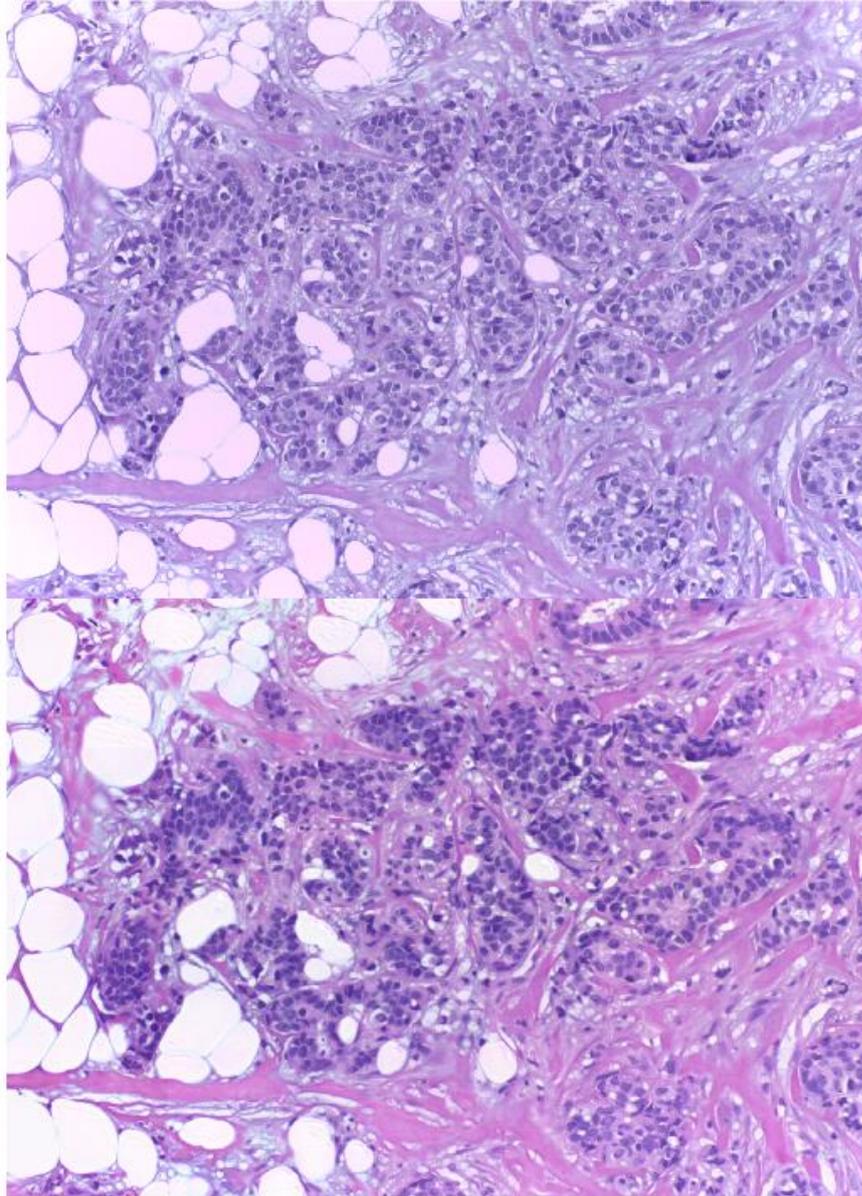


Fig.2.9 comparazione tra l'immagine di partenza (in alto) e il risultato del processo di normalizzazione (in basso)

Come si evince dal confronto tra l'immagine di partenza e l'immagine normalizzata ricostruita vi è un netto miglioramento della qualità. Lo stain della versione normalizzata presenta la colorazione desiderata ed è conforme a quello delle immagini target. E' importante inoltre sottolineare che con questo algoritmo di ricostruzione non si sono verificati artefatti dovuti alla ricongiunzione delle patch. Il modello appare molto stabile nell'applicare stain transfer alle varie patch poiché se così non fosse l'immagine ricostruita sarebbe caratterizzata da un marcato effetto a "scacchiera".

2.3 Estrazione *smart*

In questo capitolo viene presentato il processo di estrazione *smart* che costituisce il cuore di questo progetto di tesi.

Tale procedimento permette di estrarre dalle immagini le informazioni utili ai fini della classificazione rendendola più accurata. Tuttavia, prima di illustrare nel dettaglio questa procedura è necessario introdurre alcuni concetti antecedenti che aiuteranno a giustificare le scelte fatte.

In letteratura si trovano molti articoli che trattano la costruzione di algoritmi automatici per la classificazione di immagini istologiche, in cui ciascun autore illustra le proprie metodologie. Se si effettua una suddivisione macroscopica dei metodi per il *training* del classificatore si possono individuare due grosse categorie: gli allenamenti con immagini usate in formato intero e gli allenamenti svolti utilizzando porzioni dell'immagine originale chiamate patch.

Il vantaggio di utilizzare le immagini in formato intero è che, poiché esse hanno un ampio FOV (*field of view*), si ha la certezza che l'immagine

contenga l'informazione utile a determinare la classe della lesione. Questo tipo di approccio però ha principalmente tre svantaggi:

1. *Dataset* limitato: spesso per questo tipo di applicazioni vengono utilizzati *dataset* pubblici che hanno il limite di essere ridotti numericamente; in ambito di sistemi di intelligenza artificiale la numerosità dei dati di cui si dispone è un aspetto essenziale, maggiore è la dimensione del *dataset* maggiore sarà la capacità del modello di generalizzare su dati mai visti (*dataset* di test). Un modello incapace di generalizzare è un modello completamente fallimentare.
2. *Resize* dell'immagine: per allenare un modello con immagini di elevate dimensioni bisogna disporre di elevatissima potenza computazionale, pertanto la soluzione maggiormente adottata è quella di effettuare un *resize* dell'immagine e rimpicciolirla. In questo processo si passa da dimensioni di migliaia di *pixels* all'ordine delle centinaia, riducendo di un fattore che va da dieci a cento volte l'area dell'immagine; così facendo, anche gran parte dell'informazione viene compressa e in parte persa.
3. Mancanza di *post-processing*: allenare una rete con immagini intere significa non aver possibilità di elaborarne il risultato in quanto la rete associa già un label a livello image-wise.

Gli elementi di svantaggio sopra elencati costituiscono invece punti di forza nell'utilizzo del metodo a *patch*:

1. *Dataset* esteso: suddividere l'immagine originale in *patch* ha come diretta conseguenza il fatto che il numero di immagini del *dataset* aumenta. L'incremento dipende dalla dimensione della *patch*, presenza o meno di *overlap* e dal metodo di estrazione.
2. *Resize* dell'immagine: il *resize* della *patch* è molto limitato o potrebbe essere inesistente; le dimensioni di una *patch* sono confrontabili e simili a quelle solitamente usate per allenare una rete, quindi la compressione dell'informazione è nulla o trascurabile.
3. *Post-processing*: impostare la *pipeline* sulle *patch* significa che all'uscita del classificatore si ottiene la classe di *ogni patch*; questo è un grosso vantaggio perché il *label* dell'immagine intera sarà dato dall'integrazione dei *grading* delle *patch*. In questo modo si ha l'opportunità di eliminare/attenuare alcuni errori e non considerare la rete neurale come una “*black box*”.

Tuttavia, mentre in un'immagine di grosse dimensioni si ha la certezza che essa contenga l'informazione per classificare la lesione, suddividendola in *patch* ci saranno *patch* ad alto contenuto informativo, altre con meno e altre ancora che per la rete saranno addirittura rumore o fonte di errore.

Un altro aspetto da considerare nel momento in cui si decide di usare la strategia della suddivisione in *patch* è che non tutte le *patch* contengono un'informazione coerente con il *label* dell'immagine generale, anche se

rappresentano contenuti anatomici di interesse. Per fare un esempio, se si ha un'immagine classificata maligna non è detto che in ciascuna porzione dell'immagine vi sia alterazione dei tessuti, basta una piccola porzione in cui vi sia tessuto con caratteristiche anomale per considerarlo globalmente maligno.

Si consideri la seguente immagine a titolo illustrativo (Fig.2.10):

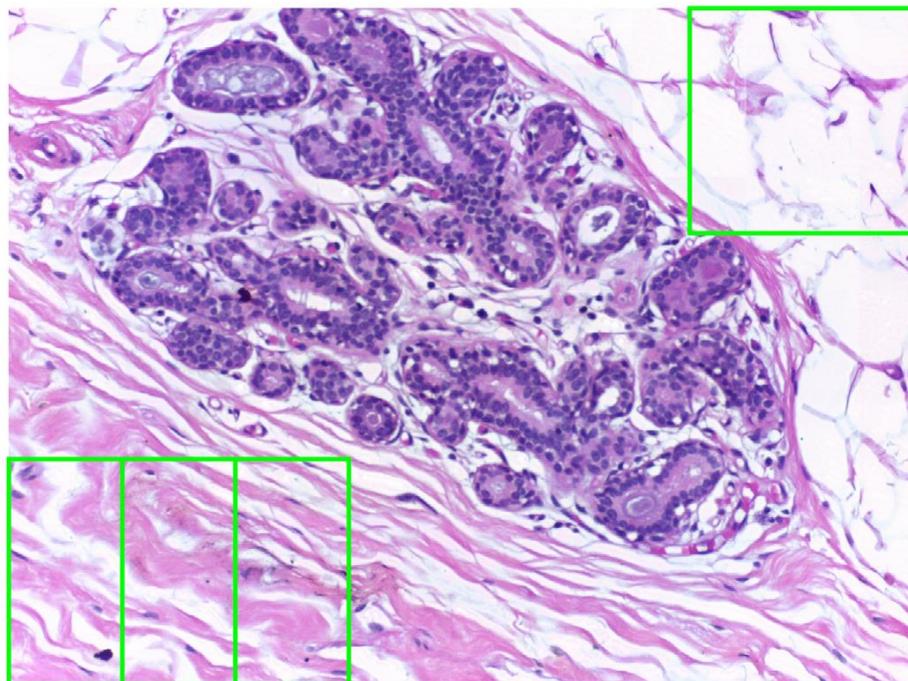


Fig.2.10 esempio di porzioni di immagini in cui non vi è contenuto informativo utile (riquadri in verde)

Appare evidente come le singole *patch* all'interno dei rettangoli verdi non siano interessanti ai fini della classificazione. Analizzando le *patch* singolarmente neanche i patologi più esperti potrebbero associarvi un *label* e tantomeno potrebbero trarvi da queste conclusioni sulla classe dell'immagine

dalla quale sono state tratte.

Alla stessa maniera sarebbe errato chiedere a una rete neurale di svolgere questo compito e, ancora peggio, sarebbe deleterio fornirle alla CNN durante un allenamento.

Nella fase di *training*, una rete neurale associa una classe all'immagine (in questo caso *patch*) in base ai pesi dei parametri che ha in quel momento, poi confronta la classe che ha predetto con quella reale della *patch* (che ha la stessa classe dell'immagine da cui è stata estratta) e successivamente le compara; nel caso in cui queste siano differenti modifica i pesi dei parametri di una certa percentuale rispetto all'errore che ha commesso, al fine di non sbagliare più in una simile situazione in una futura iterazione. Nel momento in cui si chiede alla rete di prendere una decisione su una *patch* su cui non si può inferire nulla, probabilmente sbaglierà andando a forzare i pesi dei suoi parametri ed a prendere decisioni che causeranno confusione nella classificazione delle future iterazioni.

Risulta ancor più evidente come sia deleterio includere nel *dataset* di *train/test patch* come quella rappresentata nel riquadro verde dell'immagine (Fig.2.11), nella quale non è presente nemmeno una piccola parte di tessuto e solo sfondo bianco.

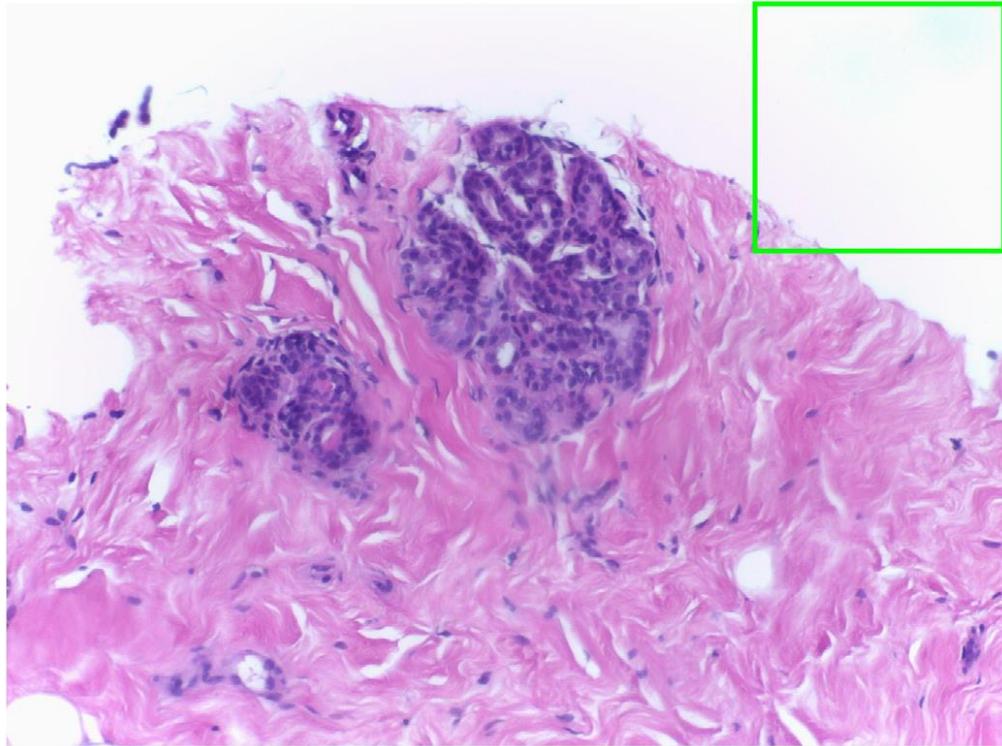


Fig.2.11 esempio di porzioni di immagini in cui non vi è contenuto informativo utile ma solo sfondo (riquadri in verde)

Appare evidente che se si vuole utilizzare la metodologia a *patch* è necessario utilizzare un approccio intelligente ed effettuare una selezione di cosa sia ritenuto di interesse da cosa non lo sia.

Per questo lavoro di tesi, si è deciso di utilizzare la strategia che prevede l'uso delle *patch* e di gestire gli svantaggi ad essa legati attraverso un approccio *smart*. Nel paragrafo successivo viene descritto quella che è stata definita estrazione *smart*, attraverso la quale si cerca di ottenere dalle varie immagini solo le porzioni che contengono l'informazione utile alla rete per poter apprendere e classificare in maniera corretta.

2.3.1 Algoritmo estrazione *smart*

Per poter comprendere come impostare il problema è necessario conoscere dove risiede l'informazione utile alla classificazione. La migliore strategia in questi casi è cercare di simulare il metodo che utilizzerebbe un patologo per svolgere tale compito.

I ragionamenti e le considerazioni fatte da uno specialista tengono conto di molti fattori e aspetti; tuttavia per l'estrazione *smart* è sufficiente sapere dove risiede l'informazione per poterla estrarre, piuttosto che conoscere quali attributi del tessuto siano importanti per il patologo. La maggior parte delle proprietà che forniscono informazioni utili sul tessuto sono da ricercarsi nei nuclei; sono molto importanti dimensioni e forme dei nuclei, numerosità, organizzazione nel tessuto e disposizione spaziale, deviazione *standard* della colorazione intra-nucleo, tessuto circostante ai nuclei, caratteristiche dello stroma nella vicinanza dei nuclei e caratteristiche del tessuto nello spazio inter-nuclei.

In definitiva, l'estrazione guidata delle patch deve avvenire nelle zone in cui vi è una elevata presenza di nuclei ed escludere le zone che hanno in prevalenza sfondo o tessuto privo di interesse.

Per strutturare l'algoritmo di estrazione è necessario distinguere i tessuti presenti in un'immagine e separare accuratamente i nuclei dallo stroma e dallo sfondo. Per fare ciò è stato utilizzato come strumento un sistema automatico per la segmentazione di immagini istologiche in possesso del gruppo di ricerca del Politecnico di Torino.

Questo sistema è anch'esso basato su reti neurali e permette, per ogni immagine istologica, di produrre due maschere binarie in cui nella prima viene effettuata una segmentazione dei nuclei, mentre nella seconda vengono

segmentati i bianchi dell'immagine corrispondenti allo sfondo o alle zone di grasso che costituiscono tessuto di interesse nullo.

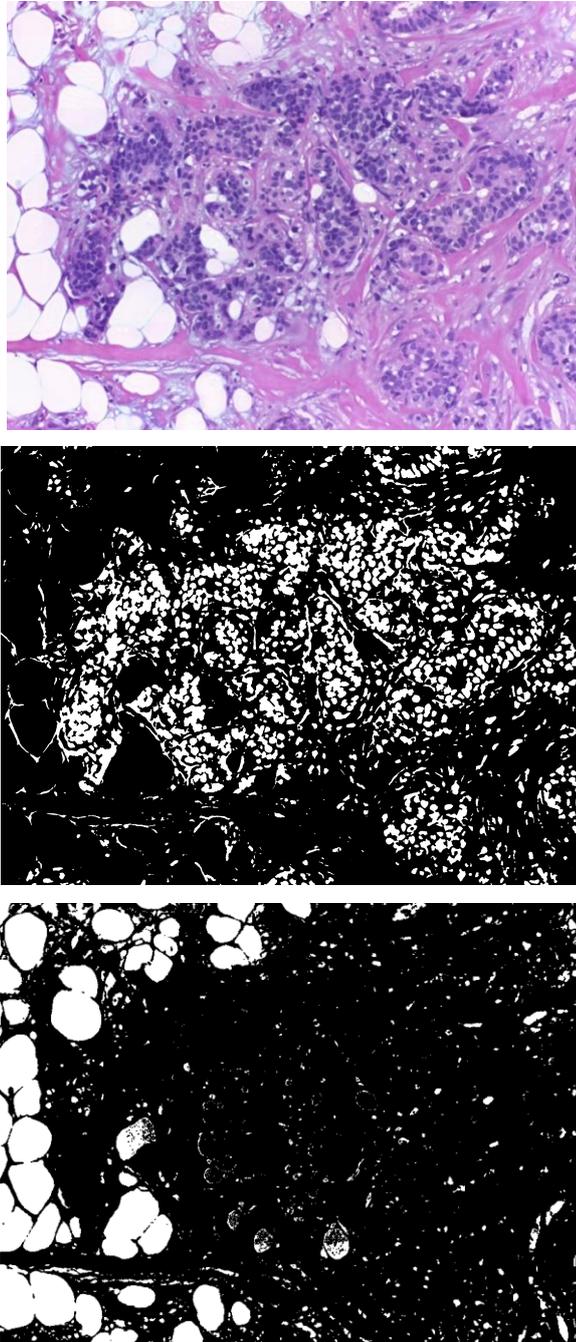


Fig.2.12 maschere di segmentazione dei nuclei(in centro) e dei bianchi (in basso)

Una volta ottenute le due maschere binarie è possibile partire con la vera e propria estrazione. Tale processo prevede l'utilizzo di una *sliding window* che scorre sull'immagine originale e sulle due maschere.

La dimensione della finestra deve essere pensata di dimensione sufficiente ad avere un elevato FOV, tenendo anche conto che a dimensioni maggiori della *sliding window* corrisponderanno un minor quantitativo di *patch* estratte e differenti strategie per la selezione della *patch*. Dunque vi è un *trade-off* di cui fare attenzione.

Nell' articolo di [15] è stato dimostrato che per questo *dataset* la dimensione ottimale della *patch* è 512x512 *pixels* in quanto, per dimensioni minori, a parità di modello e rete neurale utilizzata si ha un significativo *drop* delle *performance*.

Per ottenere una dimensione del *dataset* significativa si è deciso di adottare un *overlap* del 50%, spostando la finestra di 256 *pixels* alla volta.

Ad ogni spostamento della finestra sull'immagine RGB normalizzata corrisponde un egual spostamento sulle due maschere dalle quali vengono calcolati due parametri, *area_out* (sulla maschera dei bianchi) e numero di nuclei *ns* (sulla maschera dei nuclei).

Il parametro *area_out* viene calcolato come valore percentuale della quantità di *pixel* bianchi della maschera divisa per la quantità di *pixel* nella *patch*.

Il numero dei nuclei invece viene calcolato attraverso un metodo chiamato GLOG (*Generalized Laplacian of Gaussian*) [25]. Il GLOG permette un rilevamento automatico ed accurato dei nuclei attraverso un insieme di kernel normalizzati costruiti su diverse scale e orientamenti.

Dopo aver costruito questo set di kernel, vengono convoluti con l'immagine per ottenere quelle che nell'articolo vengono definite mappe di risposta. All'interno di queste è possibile determinare la presenza dei nuclei trovando i massimi locali. Per eliminare la possibilità di interpretare come massimi locali dei punti di risposta massima sul *background* delle mappe, viene calcolato il prodotto tra l'immagine e la maschera dei nuclei che si ha a disposizione.

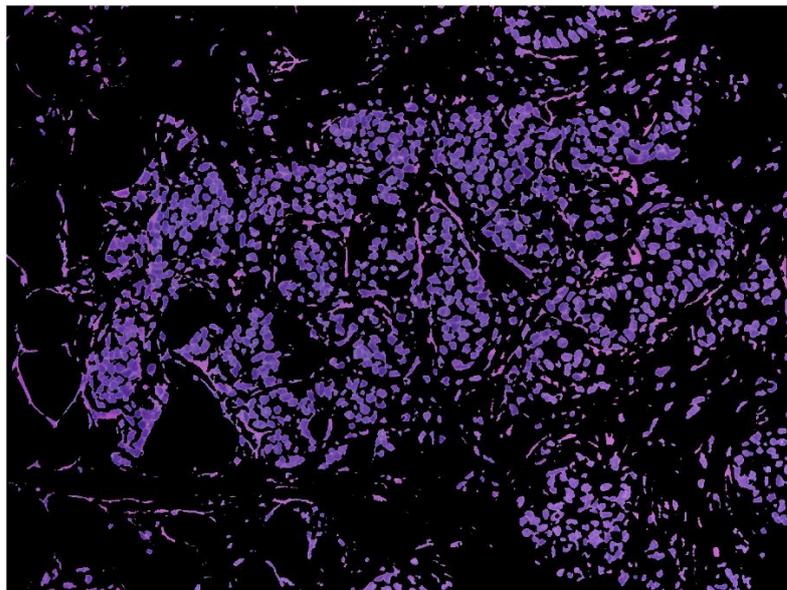


Fig.2.13 maschera di segmentazione dei nuclei moltiplicata per l'immagine

In seguito, viene eseguita una operazione di filtraggio sul numero dei nuclei. I nuclei con una variazione elevata di colore al loro interno possono presentare più massimi locali, tuttavia si è ovviamente in presenza di un singolo nucleo. Per risolvere questa problematica i punti di massimo locale entro una certa distanza vengono aggregati in *cluster*, dopo di che all'interno dei *cluster* il punto che presenta la risposta massima viene identificato come nucleo.

Tornando al criterio di selezione, questo è articolato nel seguente modo: se una patch ha un numero elevato di nuclei e supera la soglia impostata viene selezionata in qualsiasi caso, altrimenti se non si ricade in questo caso viene confrontato il numero dei nuclei con il parametro *area_out* della patch in modo tale che se una porzione dell'immagine ha un numero di nuclei non elevato ma significativo viene presa solo se non presenta una quantità di bianchi elevata. Le soglie sono state ricavate empiricamente attraverso un numero elevato di prove successive.

L'algoritmo di estrazione smart viene applicato a tutto il dataset di train (train e validation set) e al dataset di test. In questo modo si è "ripulito" sia il dataset di train per allenare la rete sia il dataset di test testare il metodo.

2.4 Allenamento reti

Terminata la fase di pre-processing costituita da normalizzazione della colorazione ed estrazione smart, si è giunti alla fase conclusiva di costruzione del classificatore finale. Nel capitolo precedente non si è accennato al processo di data augmentation, operazione che viene effettuata da quasi tutti i partecipanti della challenge. Questo è dovuto al fatto che, si è deciso di eseguire data augmentation in real-time durante le varie iterazioni dell'allenamento.

Questa scelta appare inusuale, però è figlia di determinate considerazioni; effettuando data augmentation durante l'allenamento ogni immagine del training set viene modificata ad ogni epoca in quanto prima di attraversare la rete convoluzionale subisce una trasformazione secondo valori e parametri decisi a monte. Questa strategia viene praticata poco, soprattutto quando si

dispone di dataset numericamente limitati, tuttavia porta con sé il vantaggio di passare alla rete ad ogni iterazione di ogni epoca immagini leggermente diverse. Nel caso specifico le CNN sono state allenate per un numero elevato di epoche (100) e con *learning rate* molto bassi (0.00003) così che la rete potesse apprendere caratteristiche differenti vedendo molte immagini differenti.

Per quanto riguarda le trasformazioni applicate alle immagini sono state usate:

- Rotazioni per un valore casuale da 0° a 180°
- *Flip* verticale e orizzontale
- *Shift* del 15% in altezza e larghezza

Essendo le patch quadrate applicando queste trasformazioni si ottengono immagini che presentano delle parti in cui non c'è più tessuto e si vede uno sfondo nero. L'ambiente *tensorflow* permette di riempire questi "buchi" in diversi modi, in questo caso è stato deciso di usare l'opzione *fill_mode=reflect* che riempie il vuoto con il tessuto circostante applicando una simmetria rispetto al bordo.

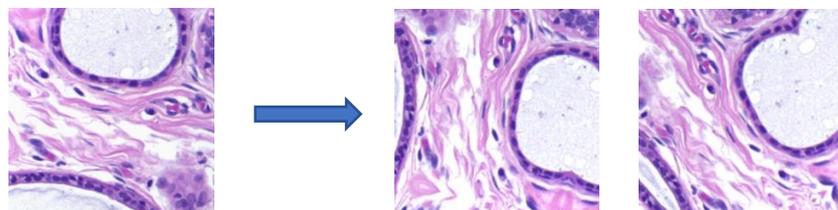


Fig.2.14 esempio di data augmentation

Per quanto riguarda la scelta della/e reti da utilizzare, questa è stata fatta in parte a seguito di attente analisi teoriche e in parte attraverso una lunga serie

di osservazioni e prove empiriche; esistono diverse architetture disponibili per applicazioni come questa, tuttavia non è possibile stabilire a priori quale garantisca le migliori prestazioni. In aggiunta a ciò, bisogna anche considerare che esistono diversi modi di allenare una rete e la scelta dipende anche dalla potenza computazionale di cui si dispone e dalla quantità di dati che si possiede. Ho scelto di elencare di seguito le principali possibilità a disposizione insieme a vantaggi e svantaggi che le caratterizzano:

- Rete custom-made: il vantaggio di questo tipo di rete è che è controllabile nella sua struttura ma per essere allenata necessita di una elevata quantità di dati nel caso si costruisca con un'elevata profondità, mentre se si abbassano il numero di parametri si rischia di avere una rete che non è in grado di apprendere caratteristiche utili per classificare correttamente a causa della sua bassa capacità. Con questo tipo di approccio è difficile trovare un compromesso se si hanno pochi dati [26].
- Rete già costruita e allenarla da zero: in questo caso rispetto a prima si ha la certezza di allenare una rete che potenzialmente può garantire ottime performance ma allenare una rete *from scratch* è una strada perseguibile solo se si dispone di un'enorme quantità di dati e di molta potenza computazionale dato l'enorme numero di parametri che caratterizza questi modelli. Una rete come la famosa InceptionV3 ha 23.851.784 parametri allenabili mentre la ResNet152V2 ad esempio dispone di 60.380.348 parametri; viene da sé come riuscire a ottimizzare reti così profonde necessiti di dati e potenza di calcolo in grandi quantità.
- Transfer learning: questo approccio consente di allenare una rete sul proprio set di dati caricando però una rete pre-allenata di cui si

conservano i pesi. Anche se le due *task*, quella dell'allenamento precedente e quella per cui si vuole allenare la rete sono molto differenti, si ha la certezza di utilizzare una rete che ha già imparato a riconoscere molti pattern spaziali, perciò è più semplice da ottimizzare anche con dataset di minore dimensione, risparmiando anche molta potenza computazionale. Inoltre, è possibile anche scegliere di mantenere i pesi di alcuni layer, allenando solo una parte dei parametri della rete. Solitamente si sceglie questa strada, si effettua il *freeze* dei primi strati che estraggono features a più alto livello per le quali non è necessario essere troppo specifici, andando a ottimizzare solo gli ultimi n strati in cui si trovano le caratteristiche più profonde e specifiche.

Oltre alla strategia di allenamento delle reti ci sono molti altri parametri da stabilire che influiscono sulle prestazioni del classificatore. Questi sono chiamati iperparametri e sono:

- **Batch size:** attraverso questo iperparametro si controlla il numero di campioni da analizzare prima che i parametri interni del modello vengano aggiornati durante l'allenamento.
- **Learning rate:** è un iperparametro che controlla quanto cambiare i pesi del modello in risposta all'errore stimato ad ogni iterazione. La scelta del *learning rate* è impegnativa in quanto un valore troppo piccolo può portare ad allenamenti troppo lunghi o addirittura bloccare il processo di ottimizzazione in un ottimo locale sub-ottimo, mentre un valore troppo grande può portare ad un processo di apprendimento instabile.

- Numero di strati allenabili (se si adotta il fine tuning)
- Scelta dell'algoritmo di ottimizzazione e dei suoi parametri

Esattamente come non è possibile stabilire a priori le architetture, allo stesso modo è complicato fare previsioni precise sugli iperparametri. Di conseguenza si è deciso di adottare l'approccio del *transfer learning* con tutte le reti pre-allenate disponibili su Keras, utilizzando per ognuna di queste diverse combinazioni di iperparametri. Le reti sono state allenate sulle patch estratte in modo smart dal dataset di train e le diverse prove sono state valutate sulle patch smart del validation set durante le varie epoche dell'allenamento. In base alle prestazioni ottenute sul validation set sono state scartate le reti più propense a raggiungere il fenomeno dell'overfitting o che non performavano bene.

Le reti che hanno ottenuto le prestazioni migliori sono state oggetto di un vero e proprio tuning dei parametri. Dal momento che si è deciso di costruire un classificatore in cui si combinano le uscite di più reti, in base alle prestazioni sul validation set sono state scelte le reti: DenseNet201, Densenet169 e InceptionResNetV2 tutte pre-allenate sul dataset Imagenet.

I parametri con cui le reti hanno ottenuto le migliori performance sono:

- Batch size = 6
- Learning rate = 0.00003
- Algoritmo di ottimizzazione: Adam optimization algorithm con i parametri $\beta_1=0.9$ $\beta_2=0.999$ $\epsilon=0.1$ (valori standard) [27]

Per quanto riguarda i layer a cui sono stati aggiornati i pesi durante l'allenamento sono stati gli ultimi 80 per le reti DenseNet169 e DenseNet201 mentre per l'architettura InceptionResNetV2 sono stati aggiornati i pesi degli ultimi 200 layer. Questa diversità è dovuta al fatto che le reti DenseNet e la InceptionResNetV2 hanno profondità molto diverse dunque è stato necessario sbloccare un numero differente di layer. Ovviamente a tutte e tre le reti è stato rimosso il layer con 1000 neuroni che serviva a classificare le 1000 classi del dataset Imagenet e sostituito con un layer a 4 neuroni. Per limitare il fenomeno dell'overfitting è stato posto un layer di dropout [28] prima del layer finale di classificazione. Per le due reti DenseNet è stato scelto un dropout di 0.5 mentre per la InceptionResNetV2 un layer di dropout con probabilità 0.8. Il layer di dropout agisce spegnendo in modo random alcune connessioni tra due layer in modo tale che durante le varie iterazioni i pesi diventino maggiormente indipendenti gli uni dagli altri, in questo modo si ottiene un modello più robusto che ha una maggior capacità di generalizzare.

2.4.1 Grafici allenamento

Di seguito vengono riportati i grafici delle reti allenate:

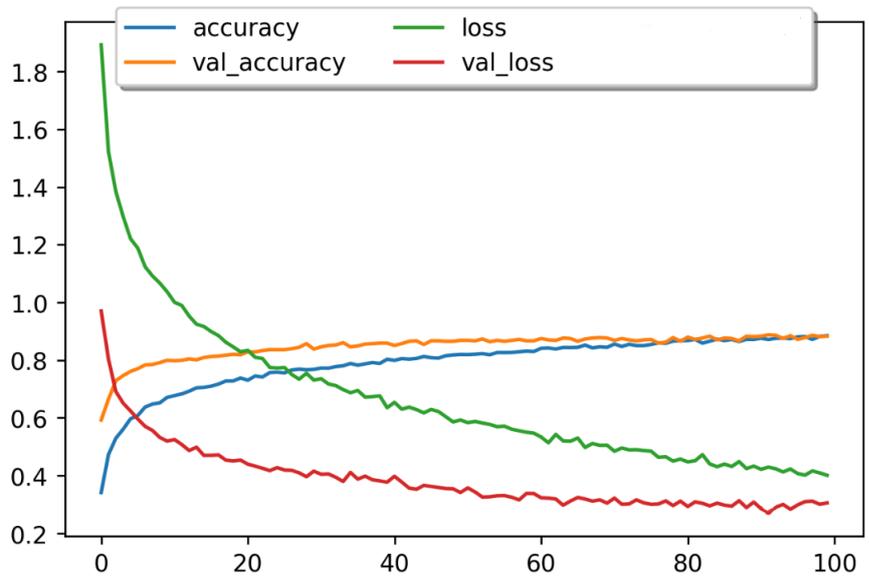


Fig.2.15 grafico dell'allenamento rete DenseNet169

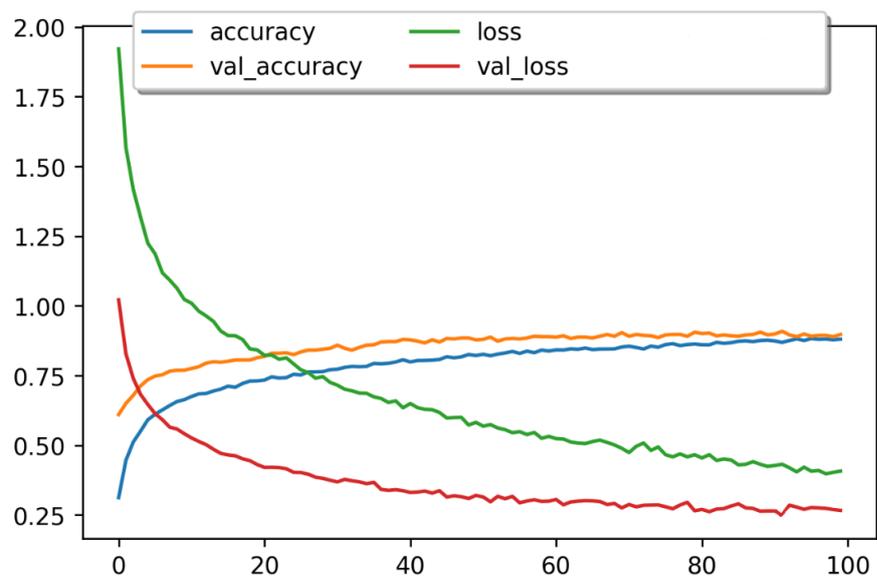


Fig.2.16 grafico dell'allenamento rete DenseNet201

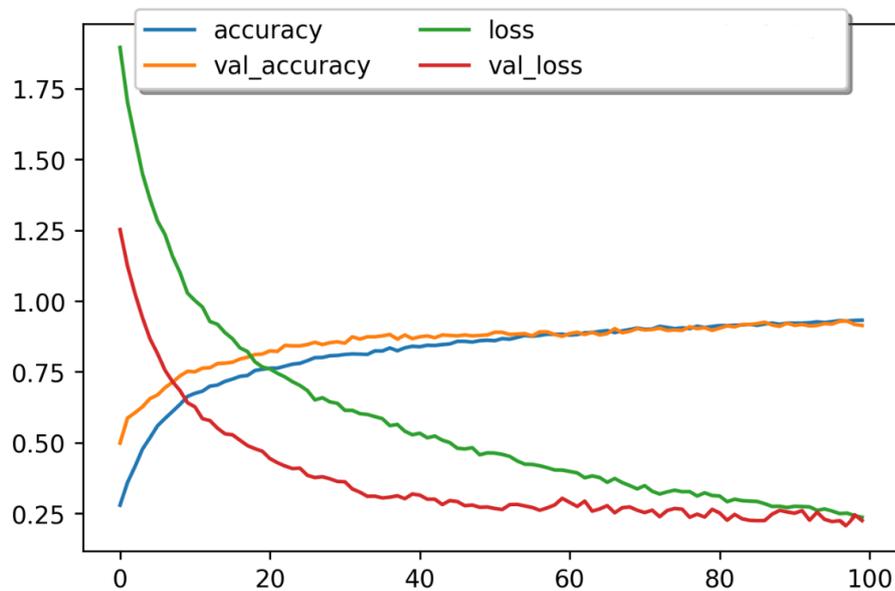


Fig.2.17 grafico dell'allenamento rete InceptionResNetV2

Dai grafici è evidente come il validation che è stato estratto casualmente dall'originale dataset di train sia relativamente semplice da classificare per le tre reti in quanto la curva dell'accuratezza del validation è sempre sopra quella dell'accuracy del train. Ciò significa che per una certa epoca le caratteristiche apprese dalla rete performano meglio sul validation che sul train. Solitamente la curva del validation è sotto la curva del train in quanto è sul train che si minimizza l'errore e si identifica la presenza di overfitting quando la differenza di accuracy tra train e validation supera una certa soglia. Tuttavia anche in questo caso è presente overfitting, ciò succede quando la curva dell'accuracy del train continua a salire per diverse epoche mentre quella del validation rimane piatta, quando ciò accade vuol dire che la rete sta apprendendo solo caratteristiche utili a classificare il train e non è più in grado di generalizzare poiché si sta specializzando eccessivamente sui dati dell'allenamento. Dal momento che l'obiettivo di un classificatore è

soprattutto quello di classificare bene gli *unseen data* è necessario scegliere le epoche del test prima che le reti vadano in overfitting.

2.5 Testing

Dopo aver allenato le reti si procede con la costruzione dell'architettura finale del classificatore. Per creare una struttura più robusta si è deciso di scegliere due modelli corrispondenti a due epoche differenti per ognuna delle tre architetture e di combinarle tra loro mediando le softmax delle epoche corrispondenti alla stessa architettura; dopo di che si effettua un'ulteriore media tra le tre medie ricavate precedentemente (questa volta si fa la media delle medie delle tre architetture). Successivamente attraverso questa softmax si classificano le patch e attraverso il principio del majority voting si attribuisce il label finale all'immagine. Di seguito in figura Fig.2.17 viene illustrata la pipeline del classificatore combinato.

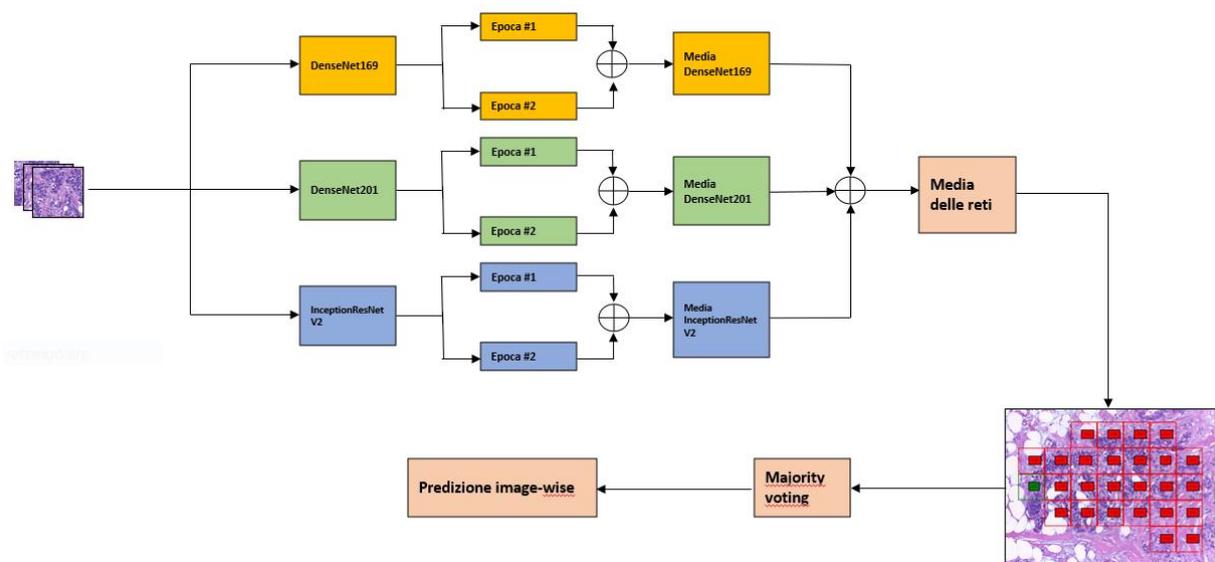


Fig.2.18 pipeline classificatore

Il motivo per cui si è deciso di calcolare le medie si giustifica nei seguenti 2 punti:

1. Durante il processo di apprendimento di una rete i pesi vengono aggiornati, dunque tra i modelli di due diverse epoche vi è una differenza nel classificare le patch; se si confrontano le softmax si può notare che epoche differenti classificano meglio alcune classi piuttosto che altre. Mediando le softmax delle patch ottenute con due modelli di due epoche differenti di una stessa rete è possibile ottenere delle softmax più robuste per quella determinata rete in quanto si abbattono parzialmente predizioni errate e si fanno emergere le predizioni corrette. Semplificando un po' il concetto, mediando due epoche è come effettuare una sorta di soppressione del rumore dove per rumore si intende predizioni sbagliate che la rete effettua ad una determinata epoca.
2. Per quanto riguarda la media delle architetture invece si sfrutta il fatto che reti differenti basano la loro predizione su *features* diverse, combinando le softmax si ottiene un classificatore con una miglior capacità di generalizzare in quanto nell'attribuire a una patch uno score questo passa attraverso differenti contributi e caratteristiche.

Per avere anche un riscontro visivo sul comportamento del classificatore combinato è stato realizzato un algoritmo che genera una sorta di *heatmap* sull'immagine sovrapponendo gli score delle patch e i label tramite codice colore sulle varie zone.

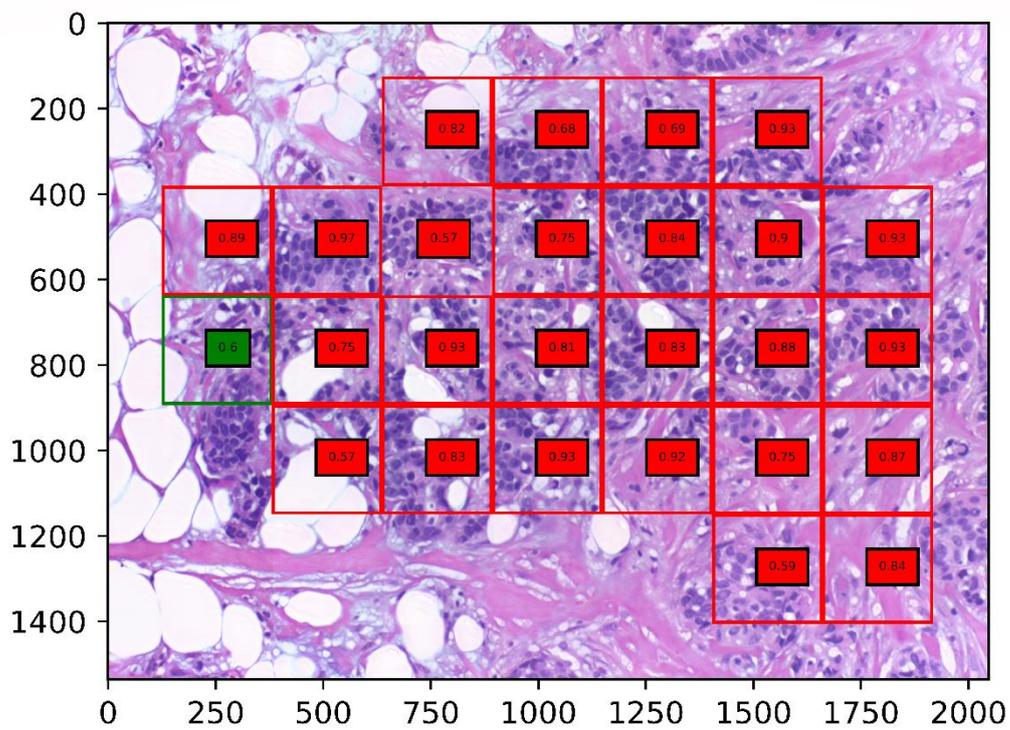


Fig.2.19 heatmap generata sull'immagine, codice colore: Normal verde, Benign giallo, Insitu arancione, Invasive rosso.

CAPITOLO III

RISULTATI

Il testing è stato effettuato sulle patch del dataset di train (train e validation riaccorpati) sia sul test set della challenge. Di seguito vengono riportate le confusion matrix per entrambi i dataset:

TEST SET:

- a) Confusion Matrix relativa alla classificazione image-wise del dataset di test su quattro classi:

		PREDIZIONE			
		Benign	InSitu	Invasive	Normal
LABEL	Benign	15	0	2	2
	InSitu	2	24	1	0
	Invasive	1	1	25	0
	Normal	1	0	0	26

- b) Confusion Matrix relativa alla classificazione patch-wise del dataset di test su quattro classi:

		PREDIZIONE			
		Benign	InSitu	Invasive	Normal
LABEL	Benign	302	84	96	54
	InSitu	63	586	114	43
	Invasive	36	121	750	6
	Normal	37	5	1	554

- c) Confusion Matrix relativa alla classificazione image-wise del dataset di test su due classi:

		PREDIZIONE	
		Carcinoma	Non-Carcinoma
LABEL	Carcinoma	51	3
	Non-Carcinoma	2	44

TRAIN SET:

- a) Confusion Matrix relativa alla classificazione image-wise del dataset di train su quattro calssi:

		PREDIZIONE			
		Benign	InSitu	Invasive	Normal
LABEL	Benign	100	0	0	0
	InSitu	0	100	0	0
	Invasive	0	3	97	0
	Normal	0	0	0	100

- b) Confusion Matrix relativa alla classificazione patch-wise del dataset di train su quattro classi:

		PREDIZIONE			
		Benign	InSitu	Invasive	Normal
LABEL	Benign	2479	19	10	60
	InSitu	22	2434	23	22
	Invasive	3	69	3084	0

Normal	29	1	0	2047
---------------	----	---	---	------

c) Confusion Matrix relativa alla classificazione image-wise del dataset di train su due classi:

		PREDIZIONE	
		Carcinoma	Non-Carcinoma
LABEL	Carcinoma	200	0
	Non-Carcinoma	0	200

Il classificatore del metodo proposto registra un accuracy image-wise sulle 4 classi nel test pari al 90% rispetto all'87% dei vincitori della BACH challenge; anche sulla classificazione binaria a livello image-wise sul test set si sono ottenuti ottimi risultati con un accuracy del 95%.

CAPITOLO IV

CONCLUSIONI

In questo progetto di tesi ci si è proposti di costruire un sistema automatico per la classificazione di carcinomi alla mammella in immagini istologiche. Il metodo proposto si basa su un elaborato *pre-processing* e *post-processing* a supporto di un classificatore composto da tre reti CNN: DenseNet-169, DenseNet-201, InceptionResNetV2. Le reti sono state allenate sul dataset di train pubblico della BACH challenge ed il metodo è stato valutato sul test set della competizione. Le performance ottenute con il metodo proposto, come visibile in tabella Tab.4.1, sono migliori rispetto a quelle registrate dai *team* che hanno partecipato alla *challenge*.

METODO	ACCURATEZZA 4 CLASSI	POSIZIONE CLASSIFICA
Proposto	90%	
Chennamsetty et al., 2018 [14]	87%	1
Kwok, 2018 [29]	87%	1
Brancati et al., 2018 [30]	86%	3
Marami et al., 2018 [31]	84%	4
Kohl et al., 2018 [32]	83%	5

Tab.4.1 *Confronto del metodo proposto con i migliori metodi proposti nella challenge*

Gran parte del merito riguardo alla bontà dei risultati raggiunti è da attribuire alla qualità del pre-processing. Per dimostrare ciò sono stati proposti altri due approcci con la stessa *pipeline* del metodo definitivo ma rimuovendo nel primo la *stain normalization* e nel secondo sia la *stain normalization* che l'*estrazione smart* estraendo le patch in ogni zona dell'immagine.

Nella tabella Tab.4.2 sono riportati i risultati sulle accuratezze valutate sul test set.

METODO	ACCURATEZZA 4 CLASSI	ACCURATEZZA 2 CLASSI
Proposto completo	90%	95%
Proposto senza stain normalization	86%	93%
Proposto senza stain normalization ed estrazione smart	85%	93%

Tab.4.2 *Confronto del metodo proposto con i due metodi alternativi proposti*

In base ai risultati raggiunti, il sistema automatico costruito in questo progetto si candida come un buon punto di partenza per ottenere in futuro uno strumento che possa aiutare i patologi a classificare le immagini istologiche per il carcinoma alla mammella.

BIBLIOGRAFIA

- [1] <https://www.humanitas.it/malattie/tumore-della-mammella/>
- [2] <https://www.fondazioneveronesi.it/magazine/tools-della-salute/glossario-delle-malattie/tumore-al-seno>
- [3] <https://www.humanitas.it/news/dalla-chirurgia-alla-radioterapia-curare-il-tumore-al-seno/>
- [4] <https://www.humanitas.it/visite-ed-esami/agobiopsia-mammaria/>
- [5] <https://www.britannica.com/science/histology>
- [6] <https://www.leicabiosystems.com/knowledge-pathway/an-introduction-to-specimen-processing/>
- [7] <http://www.istologia.unige.it>
- [8] Charu C. Aggarwal, Neural Networks and Deep Learning, Springer, pp. 1-10 (2018)
- [9] Charu C. Aggarwal, Neural Networks and Deep Learning, Springer, , pp. 315-328 (2018)
- [10] <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [11] Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, pp. 330 (2016)
- [12] Zhou, Xiaomin, Li, Chen, Rahaman, Md Mamunur, Yao, Yudong, Ai, Shiliang, Sun, Changhao, Wang, Qian, Zhang, Yong, Li, Mo, Li, Xiaoyan, Jiang, Tao, Xue, Dan, Qi, Shouliang, and Teng, Yueyang. "A Comprehensive Review for Breast Histopathology Image Analysis Using

Classical and Deep Neural Networks." IEEE Access 8 (2020): 90931-0956.
Web.

[13] Araújo T, Aresta G, Castro E, Rouco J, Aguiar P, Eloy C, et al. (2017) Classification of breast cancer histology images using Convolutional Neural Networks. PLoS ONE 12(6): e0177544.

<https://doi.org/10.1371/journal.pone.0177544>

[14] Chennamsetty, Sai Saketh, Safwan, Mohammed, and Alex, Varghese. "Classification of Breast Cancer Histology Image Using Ensemble of Pre-trained Neural Networks." Image Analysis and Recognition (2018): 804-11.
Web.

[15] K. Roy, D. Banik, D. Bhattacharjee, and M. Nasipuri. Patch-based system for Classification of Breast Histology images using deep learning. Computerized Medical Imaging and Graphics, 71:90–103, 2019.

[16] Macenko, M.: A method for normalizing histology slides for quantitative analysis. In: IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2009. IEEE (2009)

[17] R. Awan, N. A. Koohbanani, M. Shaban, and et al. Context-aware learning using transferable features for classification of breast cancer histology images. In International Conference Image Analysis and Recognition, pages 788–795. Springer, 2018.

[18] A. Golatkar, D. Anand, and A. Sethi. Classification of Breast Cancer Histology Using Deep Learning. arXiv: Breast Cancer Histology Challenge 2018, 2018.

[19] Charu C. Aggarwal, Neural Networks and Deep Learning, Springer, , pp. 24-28 (2018)

- [20] Vahadane, A., et al.: Structure-preserved color normalization for histological images. In: 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI). IEEE (2015)
- [21] Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. *IEEE Comput. Graph. Appl.* 21(5), 34–41 (2001)
- [22] Charu C. Aggarwal, *Neural Networks and Deep Learning*, Springer, Yorktown Heights NY USA, pp. 45 (2018)
- [23] Jun-Yan Zhu, Taesung Park, Isola, Phillip, and Efros, Alexei A. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks." 2017 IEEE International Conference on Computer Vision (ICCV) (2017): 2242-251. Web.
- [24] Salvi, Massimo, Mogetta, Alessandro, Meiburger, Kristen M, Gambella, Alessandro, Molinaro, Luca, Barreca, Antonella, Papotti, Mauro, and Molinari, Filippo. "Karpinski Score under Digital Investigation: A Fully Automated Segmentation Algorithm to Identify Vascular and Stromal Injury of Donors' Kidneys." *Electronics (Basel)* 9.10 (2020): 1. Web.
- [25] Hongming Xu, Cheng Lu, Berendt, Richard, Jha, Naresh, and Mandal, Mrinal. "Automatic Nuclei Detection Based on Generalized Laplacian of Gaussian Filters." *IEEE Journal of Biomedical and Health Informatics* 21.3 (2017): 826-37. Web.
- [26] Iesmantas, Tomas, and Alzbutas, Robertas. "Convolutional Capsule Network for Classification of Breast Cancer Histology Images." *Image Analysis and Recognition* (2018): 853-60. Web.

- [27] Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980v9, 30 January 2017
- [28] Nitish S, Geoffrey H, Alex K, Ilya S, Ruslan S. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res.* 2014;15(1):1929–58.
- [29] Kwok, S., 2018. Multiclass classification of breast cancer in whole-slide images. In: Campilho, A., Karray, F., ter Haar Romeny, B. (Eds.), *Image Analysis and Recognition*. Springer International Publishing, Póvoa de Varzim, pp. 931–940. doi:10.1007/978-3-319-93000-8_106.
- [30] Brancati, N., Frucci, M., Riccio, D., 2018. Multi-classification of Breast cancer histology images by using a fine-tuning strategy. In: Campilho, A., Karray, F., ter Haar Romeny, B. (Eds.), *Image Analysis and Recognition*. Springer International Publishing, Póvoa de Varzim, pp. 771–778. doi:10.1007/978-3-319-93000-8_87
- [31] Marami, B., Prastawa, M., Chan, M., Donovan, M., Fernandez, G., Zeineh, J., 2018. Ensemblenetwork for region identification in breast histopathology slides. In: Campilho, A., Karray, F., ter Haar Romeny, B. (Eds.), *Image Analysis and Recognition*. Springer International Publishing, Póvoa de Varzim, pp. 861–868. doi:10.1007/978-3-319-93000-8_98.
- [32] Kohl, M., Walz, C., Ludwig, F., Braunewell, S., Baust, M., 2018. Assessment of breast cancer histology using densely connected convolutional networks. In: G. Aresta, T. Araújo and S. Kwok et al. / *Medical Image Analysis* 56 (2019) 122–139. Campilho, A., Karray, F., ter Haar Romeny, B. (Eds.), *Image Analysis and Recognition*. Springer

International Publishing, Póvoa de Varzim, pp. 903–913. doi:10. 1007/978-3-319-93000-8_103.