

**Università degli Studi di degli Studi di Padova**

---

FACOLTÀ DI INGEGNERIA  
Corso di Laurea Triennale in Ingegneria Biomedica

# **Analisi di Texture di immagini di Endomicroscopia Confocale**

Texture Analysis of Confocal Endomicroscopy Images

Candidato:  
**Gloria Zampiero**  
Matricola 562982

Relatore:  
**Prof. Enrico Grisan**  
Correlatore:  
**Dott.ssa Elisa Veronese**



## Sommario

L'analisi di texture è un processo che permette -scegliendo tra varie tipologie di metodi- la caratterizzazione di diverse superfici ed oggetti tramite l'individuazione di loro precise proprietà statistiche. Attraverso rigorose tecniche di acquisizione delle immagini, è possibile ottenere una texture relativa ad una determinata superficie che individua univocamente la sua struttura a seconda dell'illuminazione e dell'intensità catturate dall'acquisizione. Da questa è possibile estrarre -tramite adeguata formulazione matematica- delle caratteristiche o *feature* che permettono la caratterizzazione effettiva dell'immagine di texture.

In questo elaborato si applica tale tecnica di analisi allo scopo di implementare degli algoritmi di riconoscimento che permettano la discriminazione di immagini sane o contenenti diversi gradi di alterazioni dell'epitelio esofageo. Si approfondisce, nello specifico, la patologia pre-cancerosa dell'esofago di Barrett. Le immagini vengono acquisite tramite la tecnica dell'endomicroscopia confocale, di cui sono messi in evidenza l'alto grado di risoluzione e le modalità di esecuzione dell'esame.



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Utilità dell'analisi di texture . . . . .	2
1.2	Obiettivi dell'elaborato . . . . .	3
<b>2</b>	<b>Esofago di Barrett</b>	<b>5</b>
2.1	La patologia . . . . .	5
2.2	Endomicroscopia confocale . . . . .	7
<b>3</b>	<b>Analisi di texture</b>	<b>11</b>
3.1	Estrazione di feature . . . . .	12
3.1.1	Matrici di co-occorrenza [12] . . . . .	13
3.1.2	Metodo delle matrici di run-length (RLM) [13] . . . . .	14
3.1.3	Local binary patterns (LBP) [10][14][15] . . . . .	18
3.2	Classificazione . . . . .	21
3.2.1	Tecniche supervised . . . . .	22
<b>4</b>	<b>Implementazione dell'algoritmo</b>	<b>25</b>
<b>5</b>	<b>Risultati</b>	<b>29</b>
5.1	Valutazione dei risultati ottenuti . . . . .	29
<b>6</b>	<b>Conclusioni</b>	<b>33</b>
6.1	Sviluppi futuri . . . . .	33
<b>A</b>		<b>35</b>
<b>B</b>		<b>37</b>



# Capitolo 1

## Introduzione

Una texture può essere definita come un'immagine bidimensionale che rappresenta informazioni circa la struttura di un particolare oggetto o superficie. Le texture sono variazioni d'intensità (o di colore) tipicamente originate dalla rugosità delle superfici degli oggetti colpiti da una sorgente luminosa. Per una texture ben definita, le variazioni d'intensità mostrano normalmente sia regolarità che casualità e, per questo motivo, l'analisi di texture richiede un'attenta progettazione di misure statistiche.

Secondo Tang X. [1] una *texture* è un termine usato per caratterizzare la superficie di un dato oggetto o regione ed è una delle principali caratteristiche utilizzate nell'elaborazione delle immagini e nel riconoscimento di pattern.

Il sistema visivo umano percepisce le scene interpretandole come un insieme di oggetti con molteplici relazioni fra loro: è per questo motivo che riusciamo a distinguere i vari oggetti anche se vengono colpiti da un'illuminazione non omogenea. Considerazioni totalmente diverse devono essere fatte per il riconoscimento di texture al calcolatore. In questo caso, l'illuminazione degli oggetti gioca un ruolo fondamentale poichè, al suo variare otterremo diverse condizioni di riflettività che si traducono in un aspetto diverso delle texture associate ad uno stesso oggetto, creando differenti texture. Questa proprietà nasce a causa della ruvidezza della superficie, che tende a disperdere la luce casualmente, così da migliorare o ridurre la riflettanza locale nella direzione di ispezione.

In conclusione si può affermare che un qualsiasi oggetto colpito da una particolare illuminazione e visto da una certa distanza e angolazione, può determinare l'acquisizione di immagini differenti. Di conseguenza, le texture computate da tali acquisizioni saranno inevitabilmente dipendenti dalle caratteristiche di illuminazione dell'immagine così acquisita. E' chiara dunque la difficoltà di un'analisi automatica di immagini in cui sono presenti texture diverse poiché, modificando anche una sola di queste caratteristiche, si



Figura 1.1: Esempi di texture. A seconda dell'illuminazione e dell'angolazione dell'immagine si possono creare pattern radicalmente differenti.

possono ottenere texture radicalmente differenti.

## 1.1 Utilità dell'analisi di texture

Nelle immagini digitali, le caratteristiche di una texture vengono rilevate per mezzo di variazioni di intensità o colori. Sebbene in genere non ci siano informazioni sulle cause della variazione, le differenze nei pixel delle immagini forniscono un mezzo pratico di analisi delle proprietà texturali degli oggetti. Il modo più comune di classificare texture in relazione al loro aspetto è quello di dividere le texture in due principali categorie, stocastiche e deterministiche [12]. Altri autori [2, 3] hanno invece identificato aspetti differenti per arrivare ad una classificazione di texture come ad esempio la linearità, rugosità, omogeneità, contrasto o semplicità/complessità del motivo.

La nozione di texture è molto importante in diversi ambiti, sia industriali che biomedici, infatti, conoscendo le caratteristiche superficiali di diversi oggetti è possibile effettuare un loro riconoscimento e quindi una classificazione di questi. Applicazioni importanti includono ispezioni di superfici industriali o biomedicali; per esempio per malformazioni o malattie, *remote sensing* e segmentazione di immagini aeree o dal satellite e segmentazione di regioni di texture nell'analisi di documenti. In campo biomedico, l'analisi di texture è lo strumento che viene utilizzato per il riconoscimento di tessuti di diversa natura.

Lo scopo del presente elaborato è quello di riconoscere -attraverso l'estrazione di determinate caratteristiche o *feature*- varie tipologie di tessuto, e quindi, di classificarli come fisiologici o patologici.



## 1.2 Obiettivi dell'elaborato

Le finalità del presente elaborato sono quelle di verificare le capacità, di un sistema automatico di analisi, di discriminare immagini di metaplasia da quelle di neoplasia. Le immagini vengono acquisite tramite la tecnica dell'endomicroscopia confocale.

Per quanto riguarda l'analisi, una prima scrematura è possibile distinguendo semplicemente il tessuto sano da quello patologico, procedendo poi -secondo criteri più fini- ad una differenziazione più minuziosa che tiene conto del grado e dello stato di avanzamento della patologia.

Il prossimo capitolo ha lo scopo di contestualizzare il lavoro svolto nell'ambito della patologia trattata, facendo quindi una panoramica dell'area esofagea interessata dall'*esofago di Barrett*. Verranno dati dei rudimenti di fisiologia e anatomia specificando con maggior attenzione la sintomatologia e i disturbi derivanti da tale patologia. Nel capitolo 3 verrà dato un quadro delle metodologie di analisi applicate nell'elaborato. Nel fare ciò non si ha la pretesa di essere esaurienti sull'argomento, ma quantomeno di delineare alcune delle possibilità offerte per l'analisi di texture e l'estrazione di *feature*. Il capitolo 4 contiene il lavoro vero e proprio scorporato in tutti i passi di avanzamento; nel quinto e sesto capitolo, si illustrano i risultati ottenuti con l'applicazione dell'algoritmo evidenziando le possibili migliorie.



## Capitolo 2

# Esofago di Barrett

L'esofago è un organo muscolare cavo situato nella zona toracica e controllato da uno sfintere superiore e da uno inferiore. La sua lunghezza è di circa 25-30 cm con variazioni individuali a seconda di costituzione ed altezza [4]. L'esofago può essere distinto in cervicale, toracico e addominale: in quest'ultimo tratto l'esofago s'inserisce nello stomaco ed il tessuto esofageo assume pertanto la connotazione di tessuto gastrico.

Il corpo dell'esofago è costituito da quattro tonache: esterna, muscolare, sottomucosa e mucosa; l'approfondimento delle caratteristiche di quest'ultima risulta più pertinente ai fini dell'elaborato in quanto è su questo tessuto che si localizza la patologia di seguito approfondita. La tonaca mucosa è rivestita da un epitelio squamoso pluristratificato organizzato in pliche longitudinali. Il passaggio dall'epitelio liscio, pallido e squamoso dell'esofago (di colore rosa) all'epitelio cilindrico dello stomaco (di colore rosso intenso) forma una linea netta di demarcazione, con andamento a zig-zag, detta *linea Z*. Una zona di epitelio giunzionale è interposta tra l'epitelio dell'esofago e la mucosa gastrica. Essa è costituita da cellule colonnari ed è resistente all'acidità gastrica ed alla digestione peptica [4].

### 2.1 La patologia

L'esofago di Barrett è una complicanza patologica dell'esofago a carico dell'epitelio esofageo distale. A causa di prolungati stimoli lesivi il tessuto citato tende a modificarsi a livello istologico, sostituendo le cellule del tratto esofageo con cellule proprie dell'epitelio gastrico o addirittura intestinale. E' stato appurato che tale patologia si manifesta con una probabilità del 8-20% [20] nei pazienti affetti da reflusso gastroesofageo (RGE), il quale provoca sintomi quali pirosi e rigurgito. Il contenuto dello stomaco assieme ai

suoi succhi gastrici risalgono nel tratto esofageo distale e, venendo a contatto con la mucosa esofagea, ne infiammano le pareti provocando -nei soggetti affetti- dolori cronici. A seguito di un contatto prolungato con le mucose, i succhi gastrici dello stomaco (acido cloridrico e bile) causano una modificazione dell'epitelio esofageo il quale tenta di difendersi dalla corrosione. L'epitelio modificato ha tutte le caratteristiche del tessuto gastrico, e, nello specifico, epitelio colonnare. Specifichiamo inoltre che, nelle patologie più gravi, il tessuto si presenta con caratteristiche istologiche intestinali. L'importanza della metaplasia di Barrett è legata alla sua potenziale evoluzione in adenocarcinoma esofageo, neoplasia la cui incidenza è quadruplicata negli ultimi 20 anni nel mondo occidentale. La presenza di una mucosa di Barrett comporta un incremento del rischio, compreso tra lo 0,3 e l'1% annuo, di sviluppare un adenocarcinoma esofageo [4]; per tale motivo è indicata la sorveglianza endoscopica per l'identificazione della displasia, che ne costituisce la lesione precancerosa.

L'esame endoscopico detto *esofago-gastro-duodenoscopia* o EGDS si effettua inserendo una sonda (il gastroscopio) di circa 6-10 mm [19] di diametro all'interno della cavità orofaringea, spingendola poi attraverso l'esofago, lo stomaco e il duodeno. La sonda è munita al suo interno di una telecamera e di fibre ottiche all'estremità, così da permettere una buona osservazione delle cavità prese in esame. L'EGDS riscontra la presenza di erosioni o ulcere della tonaca mucosa esofagea e consente -contemporaneamente all'osservazione tramite telecamera- l'esecuzione di una biopsia nei punti che l'operatore considera a rischio di patologie. Questo permette di evitare al paziente un'ulteriore esame invasivo in ogni caso necessario se vengono riscontrate displasie di qualche grado. L'esame istologico della biopsia permette -in un secondo momento- di discriminare la natura benigna o maligna di una lesione potenzialmente sospetta come un'ulcera o una stenosi, e di diagnosticare la metaplasia di Barrett e le sue complicanze. Le innovazioni tecnologiche degli ultimi anni hanno permesso di apportare delle modifiche strutturali alla sonda utilizzata nell'EGDS, rendendolo uno strumento più potente, accurato e sofisticato. La tecnica dell'*endomicroscopia confocale* è il frutto di queste evoluzioni poiché coniuga uno strumento endoscopico ad alta risoluzione ad un microscopio confocale laser, incrementando allo stesso tempo capacità risolutiva e precisione dello strumento. Tale tecnologia viene presentata nel seguente paragrafo.

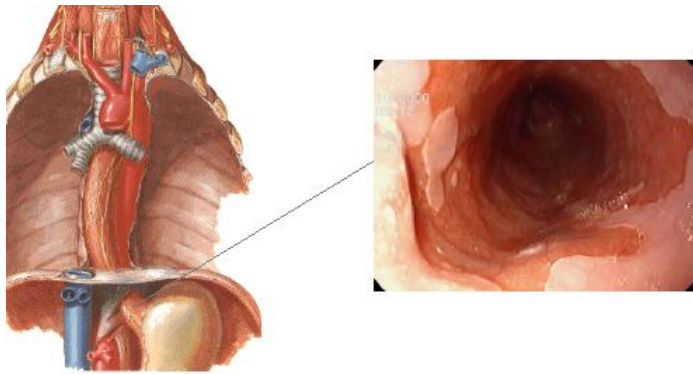


Figura 2.1: Anatomia del tratto esofageo distale con ingrandimento relativo alla morfologia tissutale in presenza di esofago di Barrett durante l'esame endoscopico.

## 2.2 Endomicroscopia confocale

L'endomicroscopia confocale laser (CLE) è una recente tecnologia endoscopica sviluppata per consentire l'analisi in vivo della struttura dei tessuti a livello cellulare. La principale innovazione apportata da tale tecnica è che, per l'analisi dei tessuti, viene utilizzato un microscopio confocale che permette un incremento notevole della risoluzione spaziale durante l'acquisizione [7]. Il termine *confocale* si riferisce al comportamento del sistema che, in fase d'ispezione, garantisce il contemporaneo allineamento su di un unico piano focale, del sistema di illuminazione e di quello di convogliamento. Tale attributo assicura l'assenza di fenomeni di *blurring* o sovrapposizione di rumore dovuto a rifrazioni provenienti da altri piani focali.

Nell'endomicroscopia confocale, un laser a bassa potenza è focalizzato (tramite lenti obiettive) su di un unico punto di un campione tissutale reso fluorescente dalla somministrazione di un mezzo di contrasto. Il laser scansiona il campione in modo sequenziale su specifici punti modellizzati da un pattern a griglia, con lo scopo di tracciare una mappa completa del campione [8]. Il fascio laser attraversa un pinhole (foro stenopeico) e colpisce il campione. La luce emessa dai fluorocromi eccitati dal laser viene catturata dall'obiettivo, attraversa lo specchio dicroico (che permette di eliminare le riflessioni date da piani *out-of-focus*) e raggiunge il fotomoltiplicatore. Questo trasforma l'intensità luminosa in un segnale elettrico di intensità proporzionale (Figura 2.2). Tra lo specchio dicroico ed il fotomoltiplicatore, il fascio luminoso attraversa un diaframma, o pinhole, che impedisce alla luce proveniente dalle zone fuori fuoco di raggiungere il fotomoltiplicatore [21]. La dimensione del pinhole è direttamente proporzionale alla capacità del

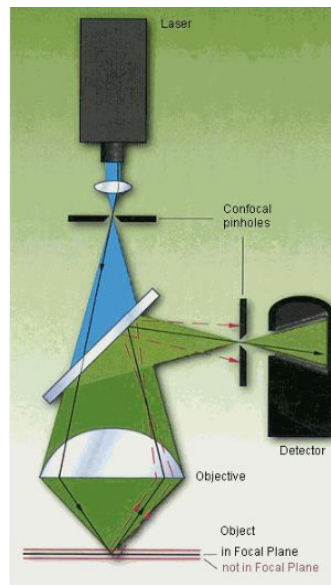


Figura 2.2: Schema dei principi basilari dell'endoscopia confocale. Sono evidenziati il fascio laser, i pinhole (uno per l'uscita del fascio e uno per l'ingresso nel detettore) e lo specchio diroico.

sistema di eliminare le riflessioni provenienti da altri piani focali. Il segnale elettrico in uscita dal fotomoltiplicatore viene quindi digitalizzato ed inviato ad un computer che registra i valori di intensità misurati in ogni punto ricreando l'immagine in scala di grigi [21]. Attraverso degli attuatori il piano di focalizzazione può essere settato a diverse profondità (fino a  $250 \mu m$  [6]) così da permettere l'acquisizione sequenziale di diversi piani di osservazione ottenendo una ricostruzione tridimensionale della struttura del tessuto (o *biopsia virtuale*).

L'estremità distale del tubo endoscopico integra le componenti del microscopio confocale laser ed uno strumento endoscopico ad alta risoluzione, generando simultaneamente l'immagine confocale e videoendoscopica standard [7]. Proprio per queste sue caratteristiche, la tecnica dell'endoscopia confocale permette, in corso di endoscopia, di ottenere immagini microscopiche in tempo reale.

### **Tecnica di acquisizione delle immagini e peculiarità dei sistemi endoscopici**

L'unità confocale del sistema endoscopico è collegata ad una sorgente laser monocromatica blu ( $\lambda = 488-600 \text{ nm}$ ), necessaria per l'eccitamento del fluoroforo presente

nel tessuto a seguito della somministrazione dello stesso per via endovenosa. La sonda confocale cattura il riflesso della luce focalizzata su piani paralleli della mucosa mediante una lente, e la ritrasmette ad un detettore d'immagine. La somministrazione endovenosa del mezzo di contrasto permette di acquisire immagini che mappano la distribuzione di fluoresceina contenuta nel tessuto analizzato. La fluoresceina sodica -largamente utilizzata in oftalmologia e angiografia- è uno dei mezzi di contrasto più comunemente utilizzati negli studi effettuati, sia per il costo contenuto che per l'assenza di effetti mutageni, e viene somministrata per via endovenosa in soluzione al 10% [6].

### Caratteristiche del tessuto analizzato tramite CLE

Durante l'acquisizione delle immagini endomicroscopiche il terminale dello strumento, o della sonda confocale, deve essere appoggiato delicatamente sulla mucosa da indagare e va mantenuto ben fermo per ridurre gli artefatti legati al movimento. La visualizzazione dell'architettura della mucosa avviene per piani paralleli alla sua superficie e non per piani ortogonali. Il sistema acquisisce e mostra sul monitor 0.8 o 1.6 immagini endomicroscopiche per secondo. Queste possono essere registrate singolarmente in un file d'immagini nel data-base dell'apparecchiatura.

L'area di mucosa esplorata è di  $500\mu\text{m} \times 500\mu\text{m}$  con una risoluzione di  $0.7\mu\text{m}$ . La scansione può essere spinta dalla superficie della mucosa sino a  $250\mu\text{m}$  in profondità, con piani di scansione confocale dello spessore dell'ordine dei  $\mu\text{m}$ . L'esame endomicroscopico dell'esofago mostra la presenza di cellule poligonali di uguale dimensione e anse capillari intrapapillari regolari dirette verso la superficie endoluminale. Nei pazienti con malattia da reflusso gastroesofageo si osservano un maggior spazio intercellulare e la presenza di capillari lievemente dilatati [6].

L'esofago di Barrett, invece, è caratterizzato dalla presenza di un pattern ghiandolare di tipo villiforme con occasionali *goblet cell* e un pattern vascolare con capillari a struttura regolare. Tali strutture risultano visibili nelle scansioni superficiali subito al di sotto dell'epitelio. La neoplasia, infine, si distingue per la presenza di capillari irregolari con spandimento di fluoresceina e di ghiandole irregolari con componente cellulare scura.





## Capitolo 3

# Analisi di texture

Nel capitolo introduttivo sono state descritte le texture come immagini che riportano univocamente (a seconda dell'illuminazione che le determina) la struttura delle superfici che descrivono. Questa proprietà consente di utilizzarle all'interno di un processo di codifica di immagini che ha come obbiettivo l'estrazione di caratteristiche statistiche matematicamente definite anche dette *feature*. Le *feature* così estratte permettono la caratterizzazione della specifica texture analizzata; questa fase rappresenta il punto di partenza per il riconoscimento (e una successiva classificazione) di set di immagini di texture. Uno dei principali obiettivi dell'analisi di texture è pertanto quello di compiere una modellazione della struttura delle texture per poi poterne fare un'elaborazione da cui si ricavano le *feature* desiderate [9]. Il sistema -adattato all'elaborazione di immagini digitali- può essere schematizzato come in Figura 3.1.

Come basi fondamentali per tutte le altre applicazioni texture-related, l'analisi di texture cerca in generale di ricavare descrizioni di texture efficienti e quantitativamente compatte, cosicché i vari operatori matematici possano essere usati per alterare e confrontare le texture.

Gli algoritmi di analisi più sviluppati inglobano l'estrazione di *feature* in modo da presentare uno schema dell'immagine attraverso un processo di codifica. Questo è peculiare a seconda della metodologia di estrazione utilizzata in base alle caratteristiche statistiche riscontrate nell'immagine di texture. Gli algoritmi possono differire sia da come le texture sono state estratte sia da come sono presentate nella descrizione. Per esempio, un approccio statistico descrive una texture per mezzo di segnali statistici di un'immagine che riflettono proprietà aleatorie della distribuzione spaziale delle intensità dell'immagine. Altri approcci possono essere per esempio il metodo spettrale, che estrae *feature* dal dominio spettrale, o l'approccio strutturale, che considera una texture come

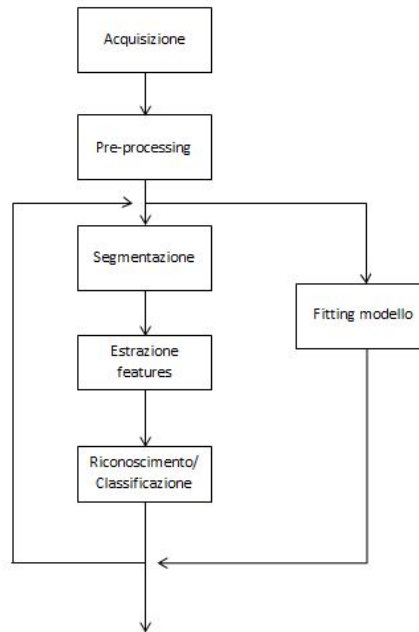


Figura 3.1: Schematizzazione dell'analisi di immagini digitali.

una disposizione gerarchica di primitive di texture ben definite.

Nei prossimi paragrafi vengono presentati vari metodi per l'estrazione di *feature* e viene data una panoramica generale riguardante il processo di classificazione con particolare attenzione al metodo poi implementato nel codice.

### 3.1 Estrazione di feature

Negli ultimi decenni sono state sviluppate molteplici tecniche per l'estrazione di *feature* impiegate nell'analisi di texture; tali metodi hanno lo scopo di mettere in evidenza -a seconda delle informazioni che si vogliono ottenere- particolari caratteristiche della texture presa in esame. Una separazione indicativa può essere fatta distinguendo metodi statistici, stocastici o legati alla frequenza spaziale.

I metodi statistici si basano sulle distribuzioni statistiche dei livelli di grigio dei pixel che compongono l'immagine; le misure che si ricavano da queste distribuzioni sono utilizzate per caratterizzare le diverse texture rispetto a specifici parametri scelti preliminarmente.

Un approccio stocastico permette di prendere in considerazione un insieme discreto di pixel attorno ad un determinato pixel, per valutare le proprietà che mettono in relazione il singolo pixel con gli altri che lo circondano.

Infine, per quanto riguarda i metodi legati alla frequenza spaziale, questi prendono in considerazione la texture valutando com'è caratterizzata l'immagine nel dominio delle frequenze di modo da poterne estrarre dei parametri che la rendono riconoscibile [11][12].

Nei prossimi paragrafi vengono esaminati con più attenzione specifici metodi per tutti gli approcci all'analisi presentati finora; i metodi delle matrici di co-occorrenza e di run length per l'approccio statistico, il metodo delle local binary patterns (LBP) per l'approccio stocastico e, infine, il metodo dei filtri di Gabor per tecniche di analisi legate alle frequenze spaziali.

### 3.1.1 Matrici di co-occorrenza [12]

Supponiamo che l'area da analizzare per una texture abbia  $N_c$  celle di risoluzione nella direzione orizzontale,  $N_r$  celle di risoluzione nella direzione verticale, e che i livelli di grigio che compongono l'immagine siano quantizzati in un numero pari a  $N_g$ . Allora  $L_c = \{1, 2, \dots, N_c\}$  sarà il dominio spaziale nella direzione orizzontale,  $L_r = \{1, 2, \dots, N_r\}$  sarà il dominio nella direzione verticale e  $G = \{1, 2, \dots, N_g\}$  sarà l'insieme degli  $N_g$  livelli di grigio quantizzati. L'immagine  $I$  può essere pertanto rappresentata come una funzione che assegna dei valori di grigio in  $G$  ad ogni cella del dominio o paio di coordinate  $L_r \times L_c$ ;  $I : L_r \times L_c \rightarrow G$ .

Le occorrenze dei livelli di grigio possono essere specificate in una matrice di frequenze relative  $A_{ij}$  con le quali due celle di risoluzione attigue separate da una distanza  $d$  occorrono nell'immagine, una con un livello di grigio  $i$  e l'altra con livello di grigio  $j$ . Tale matrice è una matrice di dimensioni  $N_g \times N_g$  detta *matrice di co-occorrenza*.

La funzione  $P$  che associa alla coppia di coordinate  $N_g \times N_g$  un determinato valore, è scelta a discrezione delle relazioni che si vogliono ricavare dall'immagine e può quindi agire secondo diverse orientazioni; le più utilizzate sono  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  e  $135^\circ$ . Di seguito viene riportato un esempio di matrice di co-occorrenza che si può ricavare dall'immagine a  $N_g = 4$  livelli di grigio in Figura 3.2.

Nell'esempio la funzione  $P$  conta il numero di transizioni calcolate lungo la direzione orizzontale ( $0^\circ$ ), associando alla cella di coordinate  $(0,0)$  della matrice di co-occorrenza il numero di coppie che - passando da destra a sinistra e viceversa- contengono 1 all'interno della loro cella e sono attigue ad un'altra cella contenente 1. Procedimenti analoghi

$$I = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 3 & 3 & 3 \\ 3 & 3 & 4 & 4 \end{bmatrix}$$

Figura 3.2: Immagine generica  $I$  da cui si ricava la matrice di co-occorrenza  $A_0$ .

vengono seguiti per la cella di coordinate  $(i, j)$  per la quale sarà conteggiato il numero di coppie contenente  $i$  nella propria cella adiacenti ad una cella  $j$ .

$$A_0 = \begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

### 3.1.2 Metodo delle matrici di run-length (RLM) [13]

Una *run* di livelli di grigio è definita come un *insieme di pixel consecutivi aventi lo stesso livello di grigio*. La lunghezza di una run è quindi pari al numero di pixel in essa contenuti. Per una data immagine, possiamo elaborare una *matrice di run length* a livello di grigio considerando differenti orientazioni. L'elemento  $(i, j)$  della matrice di run length specifica il numero di volte che l'immagine contiene -per una determinata orientazione- una run di lunghezza  $j$  composta da pixel aventi tutti livello di grigio  $i$  (o che sono compresi in un range di livelli di grigio  $gl$ ). La composizione delle matrici di run length è molto semplice, e il numero di calcoli da fare è direttamente proporzionale al numero di punti dell'immagine. Il metodo computazionale per ricavare una matrice di run length viene descritto con un semplice esempio prendendo come punto di partenza la matrice (immagine)  $B$ . Da questa ricaviamo, per quattro differenti orientazioni ( $0^\circ, 45^\circ, 90^\circ$  e  $135^\circ$ ), quattro diverse matrici di run length, e che chiamiamo rispettivamente:  $B_0, B_{45}, B_{90}, B_{135}$ .

$$B = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 1 & 2 & 2 \\ 3 & 0 & 2 & 0 \\ 2 & 2 & 3 & 1 \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 3 & 0 & 0 & 0 \end{bmatrix} \quad B_{45} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 \end{bmatrix} \quad B_{90} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 \end{bmatrix} \quad B_{135} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 \end{bmatrix}$$

Molte *feature* possono essere ricavate dalla matrice di run length [16], e parte di questi vengono riassunti in Appendice A; qui di seguito viene data una spiegazione del significato di ognuno di questi e delle convenzioni usate nella notazione.

1. ***Gray Level Run-Length Pixel Number Matrix (GLRLPNM)***

$$p_p(i, j) = p(i, j) \cdot j$$

Ogni elemento della matrice  $p_p$  rappresenta il numero di pixel di run length  $j$  e livello di grigio  $i$ . Confrontata alla matrice originale, questa matrice dà uguale importanza a tutte le lunghezze delle run dell'immagine.

2. ***Gray Level Run-Number Vector (GLRNV)***

$$p_g(i) = \sum_{j=1}^N p(i, j)$$

$p_g$  è un vettore che rappresenta la distribuzione del numero di run length  $j$  aventi livello di grigio  $i$ .  $N$  è il massimo valore di run length.

3. ***Run-Length Run-Number Vector (RLRNV)***

$$p_r(j) = \sum_{i=1}^M p(i, j)$$

$p_r$  è un vettore che rappresenta la distribuzione del numero di run length  $j$ .

4. ***Gray Level Run-Length-One Vector (GLRLOV)***

$$p_o(i) = p(i, 1)$$

$p_o$  è un vettore che rappresenta la distribuzione del numero di run aventi livello di grigio  $i$  e lunghezza pari a 1. Questa *feature* è molto importante in quanto la

maggior parte dell'informazione di una matrice di run length è contenuta nelle run di lunghezza unitaria.

5. **Short Run Emphasis (SRE)**

$$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{p(i,j)}{j^2}$$

Misura la distribuzione di run corte. E' altamente dipendente dall'occorrenza delle run corte e ci si aspetta che sia consistente nelle texture fini.  $n_r$  rappresenta il numero di run totali,  $i$  il numero di livelli di grigio dell'immagine (aventi valore massimo  $M$ ) e  $j$  il numero delle run length dell'immagine (il cui valore massimo è  $N$ ).

6. **Long Run Emphasis (LRE)**

$$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N p(i,j) \cdot j^2$$

Misura la distribuzione di run lunghe. E' altamente dipendente dall'occorrenza delle run lunghe e ci si aspetta che sia consistente nelle texture che presentano ruvidità.  $n_r$  rappresenta il numero di run totali,  $i$  il numero di livelli di grigio dell'immagine (aventi valore massimo  $M$ ) e  $j$  il numero delle run length dell'immagine (il cui valore massimo è  $N$ ).

7. **Gray-Level Nonuniformity (GLN)**

$$\frac{1}{n_r} \sum_{i=1}^M \left( \sum_{j=1}^N p(i,j) \right)^2$$

Misura la somiglianza dei livelli di grigio lungo l'immagine. Ci si aspetta che sia poco consistente se i livelli di grigio è simile lungo l'immagine. Le notazioni utilizzate sono identiche a quelle delle feature precedentemente spiegate.

8. *Run-Length Nonuniformity (RLN)*

$$\frac{1}{n_r} \sum_{j=1}^N \left( \sum_{i=1}^M p(i, j) \right)^2$$

Misura la somiglianza della lunghezza delle run lungo l'immagine. Ci si aspetta che sia poco consistente se la lunghezza delle run è simile lungo l'immagine.

9. *Run Percentage (RP)*

$$\frac{n_r}{n_p}$$

Misura l'omogeneità e la distribuzione delle run di un'immagine per una specifica direzione. Ci si aspetta che sia più consistente quando la lunghezza delle run è 1 per tutti i livelli di grigio di una specifica direzione.  $n_r$  rappresenta il numero di run totali mentre  $n_p$  il numero totale di pixel dell'immagine.

10. *Low Gray-Level Run Emphasis (LGRE)*

$$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{p(i, j)}{i^2}$$

Misura la distribuzione dei livelli di grigio bassi. Ci si aspetta che sia più consistente per un'immagine con bassi livelli di grigio.

11. *High Gray-Level Run Emphasis (HGRE)*

$$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N p(i, j) \cdot i^2$$

Misura la distribuzione dei livelli di grigio alti. Ci si aspetta che sia più consistente per un'immagine con alti livelli di grigio.

12. *Short Run Low Gray-Level Emphasis (SRLGE)*

$$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{p(i, j)}{i^2 \cdot j^2}$$

Misura la distribuzione congiunta delle run corte e dei livelli di grigio bassi. Ci si aspetta che sia più consistente per un'immagine con molte run corte e bassi livelli di grigio.

13. *Short Run High Gray-Level Emphasis (SRHGE)*

$$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{p(i, j) \cdot i^2}{j^2}$$

Misura la distribuzione congiunta delle run corte e dei livelli di grigio alti. Ci si aspetta che sia più consistente per un'immagine con molte run corte e alti livelli di grigio.

14. *Long Run Low Gray-Level Emphasis (LRLGE)*

$$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{p(i, j) \cdot j^2}{i^2}$$

Misura la distribuzione congiunta delle run lunghe e dei livelli di grigio bassi. Ci si aspetta che sia più consistente per un'immagine con molte run lunghe e bassi livelli di grigio.

15. *Long Run High Gray-Level Emphasis (LRHGE)*

$$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N p(i, j) \cdot i^2 \cdot j^2$$

Misura la distribuzione congiunta delle run lunghe e dei livelli di grigio alti. Ci si aspetta che sia più consistente per un'immagine con molte run lunghe e alti livelli di grigio.

### 3.1.3 Local binary patterns (LBP) [10][14][15]

L'operatore LBP è stato inizialmente introdotto come misurazione complementare al contrasto locale dell'immagine; successivamente è stato valutato come un operatore locale avente significato proprio. La sua prima implementazione è stata fatta usando otto pixel attigui ad un pixel centrale che veniva usato come soglia: la codifica LBP avveniva moltiplicando i valori soglia con dei pesi caratteristici per ogni posizione dei pixel del vicinato. Sommando i vari addendi così ottenuti si ricava la codifica cercata. In Figura 3.2 viene riportato un esempio di codifica utilizzando l'operatore LBP.

L'algoritmo appena descritto può essere migliorato per ottenere una codifica che presenta due importanti e utili proprietà:

- invarianza dell'operatore rispetto alla scala di grigi,



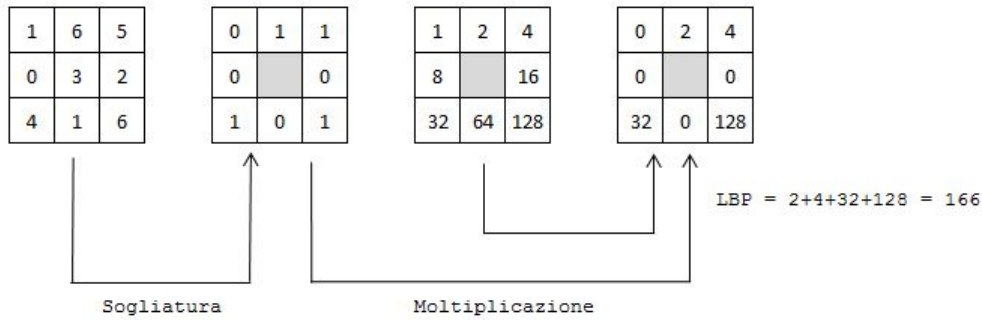


Figura 3.3: Codifica attraverso local binary pattern di un'immagine digitalizzata sotto forma di una matrice  $3 \times 3$  esemplificativa.

- invarianza dell'operatore rispetto alla rotazione.

L'ottimizzazione di seguito presentata si basa sul metodo adottato da Ojala et al. [15]; questi propongono di partire da una distribuzione di valori di grigio di un intorno circolare simmetrico di pixel, e di ricavarne un operatore invariante, per definizione, alle trasformazioni monotoniche di livelli di grigio. L'importanza della formulazione risiede inoltre nel concetto di pattern uniforme: questa è una delle proprietà locali fondamentali di una texture, che permette di modellare un operatore LBP caratterizzato dalle proprietà sopra espresse.

### Derivazione operatore LBP uniforme e invariante alla rotazione

Definiamo una texture  $T$  in un intorno di un'immagine monocromatica come la distribuzione dei valori di grigio dei pixel  $P$  ( $P > 1$ ) dell'immagine:

$$T = t(g_c, g_0, \dots, g_{P-1}), \quad (3.1)$$

dove  $g_c$  corrisponde al livello di grigio del pixel centrale dell'intorno, e  $g_p$  ( $p = 0, \dots, P-1$ ) corrisponde ai livelli di grigio di  $P$  equamente distanti su un cerchio di raggio  $R$  ( $R > 0$ ) che forma un intorno circolare simmetrico di pixel. Pertanto, se le coordinate di  $g_c$  sono  $(0, 0)$ , le coordinate di  $g_p$  sono date da  $(-R \sin(2\pi p/P), R \cos(2\pi p/P))$ . I parametri  $P$  ed  $R$  controllano rispettivamente la quantizzazione dello spazio angolare e

la risoluzione spaziale. In formule l'operatore  $LBP_{P,R}$  è definito come:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p. \quad (3.2)$$

Dove  $s(x)$  è:

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

L'operatore  $LBP_{P,R}$  produce  $2^P$  pattern differenti; quando l'immagine è rotata, i livelli di grigio  $g_p$  si muovono lungo il perimetro del cerchio. Per rimuovere gli effetti della rotazione e assegnare un unico identificatore per ogni pattern, il local binary pattern invariante alla rotazione è definita come segue:

$$LBP_{P,R}^{ri} = \min\{ROR(LBP_{P,R}, i) \mid i = 0, 1, \dots, P-1\} \quad (3.3)$$

Dove  $ROR(x, i)$  è una rotazione verso destra del bit  $P$ -esimo del numero  $x$   $i$  volte. E' stato osservato che certi local binary pattern sono proprietà fondamentali della texture. Questi pattern fondamentali sono detti uniformi e hanno una struttura circolare uniforme che contiene pochissime transizioni spaziali da 0 a 1. Per definire formalmente questi modelli introduciamo la misura di uniformità  $U$ , che corrisponde al numero di transizioni spaziali (cambiamenti da 0 a 1) nel pattern. Per esempio, il pattern  $00000000_2$  o  $11111111_2$  hanno valore  $U$  pari a 0 poichè hanno 0 transizioni da 0 a 1 al loro interno. Per convenzione, definiamo pattern uniformi quelli che hanno  $U \leq 2$ . Si definisce quindi l'operatore invariante alla rotazione e alla scala di grigi  $LBP_{P,R}^{riu2}$ :

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c), & \text{se } U(LBP_{P,R}) \leq 2 \\ P + 1 & \text{altrimenti} \end{cases} \quad (3.4)$$

dove

$$U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|. \quad (3.5)$$

Secondo le equazioni appena riportate ci sono esattamente  $P + 1$  pattern uniformi in un intorno circolare simmetrico di un insieme di  $P$  pixel; la (3.4) assegna un'unica etichetta ad ognuno di loro. L'istogramma dei local binary pattern uniformi può essere usato per

misurare somiglianze e diversità delle texture.

## 3.2 Classificazione

Con il termine classificazione si intende una procedura statistica che permette di associare ciascun oggetto (che sia un'immagine, un pattern, un dato numerico...), appartenente a un generico spazio multidimensionale, a una o più etichette, corrispondenti alle possibili classi di cui può far parte un oggetto; si parla di classificazione esclusiva quando ciascun oggetto può appartenere a una sola classe, o di classificazione continua (o *fuzzy*) se un oggetto può appartenere, con un certo grado di probabilità, a più classi. L'obiettivo della classificazione è quindi, più in generale, quello di selezionare la categoria più appropriata per un oggetto sconosciuto avendo a disposizione un set di categorie conosciute. Una formalizzazione del problema di classificazione è la seguente: partendo da una serie di dati di addestramento  $(x_1, y), (x_2, y), \dots, (x_n, y)$  si deve produrre un classificatore  $h : X \rightarrow Y$  che realizzi il mappaggio di nuovi elementi  $x \in X$  sulle etichette  $y \in Y$  [22].

La classificazione riguarda la costruzione di una procedura di decisione basata su una sequenza di casi e può essere considerata come una delle maggiori applicazioni della *machine learning* in base a cui i classificatori sono costruiti basandosi sulla conoscenza osservata dagli esempi.

I classificatori vengono usualmente separati in due categorie:

- classificatori parametrici;
- classificatori non-parametrici.

La differenza sta nel fatto che i primi fanno ipotesi certe sulle distribuzioni delle *feature*, mentre i classificatori non-parametrici utilizzano distribuzioni di *feature* arbitrarie senza fare ipotesi circa le forme di densità sottostanti. Entrambe le categorie hanno bisogno di un qualche tipo di conoscenza dei dati, siano questi campioni di addestramento o parametri delle distribuzioni di *feature* assunte.

Per ottenere una buona classificazione si usano le informazioni riguardanti alcuni tratti salienti (*feature*, caratteristiche) degli oggetti in esame, e li si confronta, in un apposito spazio multidimensionale, con quelli di un *training set*: se questo è etichettato -e quindi le possibili classi sono note e ogni campione nel set è già associato alla classe di appartenenza- si parla di apprendimento *supervised*; in caso contrario -cioè quando le possibili classi vanno direttamente ricavate dai dati stessi- abbiamo a che fare con

apprendimento *unsupervised*. Possiamo pertanto fare un'ulteriore distinzione discriminando le tecniche che tengono conto delle informazioni fornite a priori dai dati da quelle che non ne tengono conto; rispettivamente chiameremo le prime *tecniche supervised* e le seconde *tecniche unsupervised*.

La trattazione proseguirà nella presentazione della prima tipologia di tecniche e, in particolare, della SVM in quanto strumento utilizzato nel presente elaborato.

### 3.2.1 Tecniche supervised

Nella classificazione *supervised* si insegna ad un classificatore a riconoscere l'appartenenza di un oggetto a delle classi di interesse note a priori (etichettate). A questo scopo, l'insieme di campioni da etichettare nel processo di classificazione viene suddiviso in due insiemi disgiunti: il *training set* ed il *test set*. Il primo ha il compito di addestrare il classificatore fornendogli vari esempi di campioni correttamente classificati, mentre il *test set* contiene l'insieme dei campioni usati per stimare la probabilità di errore del classificatore. Poichè il *training set* è impiegato per valutare numerosi parametri, di solito, il numero di campioni contenuti nel *training set* è maggiore o uguale al numero di campioni presenti nel *test set*. La capacità di generalizzazione di un classificatore *supervised* è la capacità di classificare un *test set* arbitrario con accuratezza simile a quella ottenuta classificando il *training set* con cui il classificatore è addestrato. Un'assunzione importante è che i campioni del *training set* siano indipendenti e disgiunti da quelli presenti nel *test set*; da questa ipotesi, infatti, dipendono sia la buona capacità di generalizzazione del classificatore sia la sua accuratezza nella classificazione [17].

Vi sono numerose metodologie per partizionare i set di campioni che vanno negli insiemi di *training* e di *test*, ma quello utilizzato in questo elaborato è il *leave-one-out*, sostanzialmente un caso particolare della *k-fold cross validation*. Quest'ultima procedura viene usata per migliorare la stima empirica della probabilità di errore del classificatore, e consiste nella suddivisione di  $N$  campioni pre-etichettati in  $k$  sottoinsiemi disgiunti (ciascuno composto da  $M = \frac{N}{k}$  campioni). A rotazione si usa come *test set* ciascuno dei  $k$  insiemi, stimando la probabilità di errore per ognuno di essi. Se  $N$  è molto grande, spesso si usa  $k = 2$  con divisione casuale, a volte senza neanche ruotare sui due possibili *test set* [17].

Nel *leave-one-out* il principio è essenzialmente identico a quello della *k-fold cross validation*; l'unica differenza è rappresentata dal fatto che si utilizzano  $N-1$  campioni per l'addestramento del classificatore, e il campione escluso dal *training* viene utilizzato per

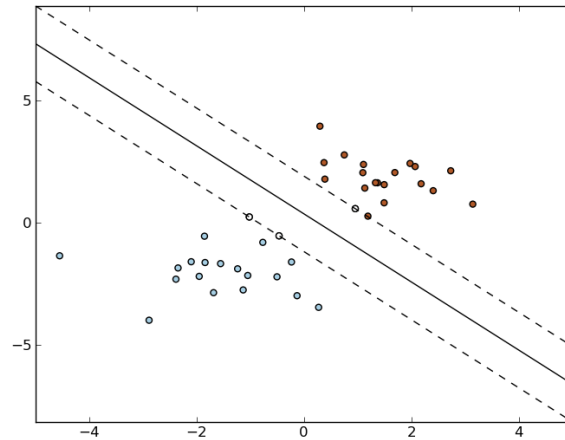


Figura 3.4: Esempio di ricerca di iper-piano di separazione ottimo con SVM.

il *testing*. Tale procedura viene compiuta ripetutamente per ognuno degli  $N$  campioni presenti nel data set.

### Support Vector Machine (SVM)

SVM è un classificatore binario utile in molti campi della ricerca; se comparato con buona parte dei classificatori tradizionali esso è capace di generalizzare con un buon grado di accuratezza campioni sconosciuti o non visti. Ciò che caratterizza questo algoritmo è che invece di tentare di stimare le densità di probabilità delle classi, risolve direttamente il problema di interesse, ovvero di trovare le superfici decisionali che delimitano le classi (*classification boundaries*). L'idea sottostante è la seguente: date due classi di pattern multidimensionali, SVM determina l'iper-piano in grado di separare al meglio le classi (ovvero che classifica correttamente pattern di entrambe massimizzando la distanza della separazione marginale tra le classi).

E' da mettere in evidenza che quando si trattano classi linearmente separabili non sorgono particolari problemi, mentre quando le classi non sono linearmente separabili, ovvero non esiste un iper-piano in grado di separare le classi, si devono apportare delle modifiche all'algoritmo. Nella sostanza si cerca l'iper-piano di separazione in spazi di dimensionalità superiore: infatti, se due classi non sono linearmente separabili la probabilità che diventino linearmente separabili aumenta se aumentiamo la dimensionalità dello spazio [18].



## Capitolo 4

# Implementazione dell'algoritmo

L'obiettivo di questo capitolo è quello di assemblare le conoscenze finora introdotte finalizzandole nell'implementazione di un algoritmo di riconoscimento.

### Descrizione dei dati

L'insieme globale delle immagini utilizzate -che chiameremo blocco (1)- è stato raccolto e organizzato in blocchi di dati a seconda del grado di patologia riscontrata a seguito della biopsia, distinguendo fra immagini sane, metaplasiche o neoplasiche. Il set di immagini (1) presenta una distribuzione delle diagnosi piuttosto sbilanciata, infatti contiene 23 immagini di tessuto sano, 262 immagini di tessuto affetto da esofago di Barrett e 53 immagini di tessuto neoplasico.

Il secondo blocco -che chiameremo (2)- deriva da una selezione di immagini di (1) ed è stato elaborato ottenendo delle matrici di dimensioni  $27 \times 16384 \times 64$ . La prima dimensione (27) indica il numero delle *feature* calcolate (tenendo presente che le prime tre forniscono semplicemente informazioni riguardo riga, colonna e indice del pixel analizzato), la seconda (16384) indica il numero di pixel per ogni sottoimmagine, la terza (64) il numero di sottoimmagini calcolate a partire dall'immagine di partenza ciascuna di dimensioni (128x128). Il blocco (2) presenta un numero piuttosto limitato di dati (17) e una distribuzione delle diagnosi che discrimina solamente tra due classi: quella delle immagini affette da esofago di Barrett (7 matrici) e quella di immagini neoplasiche (10 matrici). Il data set (2) è composto da un numero di campioni di molto inferiore a (1) poichè le matrici di cui è costituito risultano molto pesanti in termini di byte di memoria occupati; ciascuna matrice è infatti dell'ordine delle centinaia di megabyte.

### Feature estratte

L'algoritmo sviluppato si basa sull'implementazione del concetto di matrice di run length ottenuta a partire dalle immagini esofagee acquisite.

A livello computazionale (sia per il blocco (1) che per il blocco (2)), la scelta è stata quella di suddividere la dimensione delle immagini e ottenere quindi delle regioni più piccole da processare: questo per abbattere i tempi di elaborazione e, soprattutto, per compensare la variabilità intra-immagine data dalla possibile presenza di tessuti con aspetto differente all'interno della stessa immagine. Nello specifico, per il blocco (1), si è partiti da immagini 1024x1024 pixel da cui sono state ricavate delle sottimmagini o *region of interest (ROI)* ciascuna da 32x32 pixel. Il partizionamento non è stato compiuto solo a livello spaziale agendo sui pixel, ma anche sul piano delle intensità quantizzando i livelli di grigio delle immagini; queste infatti erano composte di 256 valori di grigio che, dopo la quantizzazione, sono divenuti 8. Per la quantizzazione dei livelli di grigio vale la stessa ratio compiuta con il partizionamento in regioni delle immagini, infatti, raggruppando 32 livelli di grigio sotto uno solo, si abbassa il costo dell'elaborazione in termini di tempo senza rinunciare al contributo di informazione offerto da ciascuna immagine.

Dopo aver normalizzato l'immagine in ingresso ( $x_{im}$ ) se ne ricavano le ROI e quindi la matrice di run length (indicata come  $rl$ ). Tale matrice contiene per ognuno dei livelli di grigio quantizzati ( $N_{gl}$ ), l'istogramma della lunghezza delle *run*.

Avendo ottenuto nel modo descritto la matrice di run length, possiamo utilizzarla per l'estrazione delle *feature* che ci interessano maggiormente. Quelle individuate come le più significative sono: Short Run Emphasis (SRE), Long Run Emphasis (LRE), Short Run Low Gray-level Emphasis (SRLGE), Long Run Low Gray-level Emphasis (LRLGE), Short Run High Gray-level Emphasis (SRHGE), Long Run High Gray-level Emphasis (LRHGE).

L'uscita (per il blocco di (1)) è costituita da matrici di dimensioni 32x32x6 per ciascuna delle immagini contenute nel file di dati caricato. Plottando i diversi piani delle matrici, si possono visualizzare in uscita le diverse texture corrispondenti alle sei diverse *feature* estratte. In Figura 4.1 si riporta un'immagine di esofago neoplastico.

Ai fini dell'addestramento del classificatore, e della valutazione degli errori, associamo il valore -1 alle immagini che non presentano patologie, 0 a quelle affette da esofago di Barrett e +1 alle immagini neoplastiche: combinando in un unico vettore tali valori si ottiene il vettore di classificazione vera che definiamo *target*.

La matrice  $F$  e il vettore di classificazione *target*, inseriti nella Support Vector Machine, permetteranno l'addestramento del classificatore, e quindi produrranno il vettore



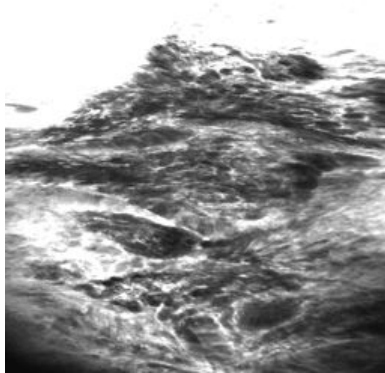


Figura 4.1: Immagine endoscopica di esofago affetto da neoplasia appartenente al data set (1).

di *training* per l'algoritmo. Nella pratica, l'addestramento del classificatore viene effettuato utilizzando la libreria `libsvm-3.11` [23], che offre delle function per la stima delle distribuzioni differenti da quelle presenti nel toolbox di MATLAB. Grazie a questa libreria è possibile settare parametri quali  $C$  e  $\gamma$ . Modificando il primo determiniamo la misura del rischio atteso della classificazione, mentre, agendo sul secondo, modifichiamo parametri relativi alla funzione che modella l'iperpiano in grado di separare al meglio le classi.

Per quanto riguarda il blocco di dati denominato (2) l'approccio è stato identico, e la differenza è data dalle *feature* utilizzate. Si è partiti da immagini (sotto forma di matrici a tre dimensioni) già suddivise in sottoimmagini dalle quali è stata estratta una particolare *feature* (la quarta) successivamente salvata in un vettore. Quest'ultimo (come nell'approccio per il blocco (1)) caratterizza l'immagine dato che al suo interno troviamo l'istogramma dei livelli di grigio per ogni sottoimmagine di quella determinata immagine (e quindi di quella particolare tipologia). Poichè il contributo d'informazione più rilevante si trova mediamente nei primi valori dell'istogramma, la scelta è stata quella di salvare i primi dieci valori (valore arbitrario) del vettore di dimensioni  $128 \times 128$  campionandoli logicamente sull'intera lunghezza del vettore.

Analogamente a prima tutti i vettori così raccolti (uno per ognuna delle sottoimmagini) sono stati inseriti nella matrice di classificazione  $F$ , mentre ad ogni sottoimmagine (che presenta esofago di Barrett o neoplasia) è stato assegnato rispettivamente il valore numerico 0 o 1. Alla fine di questa operazione si ottiene -come prima- il vettore di classificazione vera *target*. Allo stesso modo possiamo dunque addestrare la Support Vector Machine come è stato fatto per il blocco di dati (1).

**Cross validation**

La cross validation è l'ultimo step degli algoritmi che vengono presentati e permette, in una sola fase, sia la classificazione dei set di dati sia la valutazione della robustezza dell'algoritmo.

Come esposto nel paragrafo 3.2.1, la cross validation segue la logica *leave-one-out* che consiste nel partizionare l'insieme di dati da classificare, etichettando  $N - 1$  di questi come facenti parte di un set detto *training set* utile all'addestramento del classificatore, e quello che rimane fuori viene utilizzato per il test del classificatore.

## Capitolo 5

# Risultati

Per stimare i risultati ci serviamo delle iterazioni compiute durante la *cross validation*. Il classificatore SVM ha il compito di fornire la classificazione per ogni sottoimmagine (di cui si calcola l'accuratezza attraverso il leave-one-out), così da ottenere in uscita -per ognuna delle iterazioni- la percentuale di accuratezza corrispondente ad ogni ciclo.

Al fine di valutare con maggior rigore i risultati ottenuti dall'algoritmo ci siamo serviti delle ROC (*Receiver Operating Characteristic*) ovvero delle curve di predizione che valutano la bontà dell'algoritmo di riconoscimento. Per classificare un'intera immagine, si guarda la frazione di sottoimmagini classificate come metaplasia e quelle classificate come neoplasia. Al variare della soglia su tale frazione -che utilizziamo per assegnare l'intera immagine ad una classe- varia la sensibilità e specificità del sistema. Per determinare la curva ROC abbiamo posto la *sensibilità* del test in funzione del valore  $(1 - \textit{specificità})$ . Per *sensibilità* si intende la percentuale con cui si determina correttamente se l'immagine analizzata presenta neoplasie se questo è vero (*veri positivi*), mentre la *specificità* rappresenta la percentuale con cui viene determinata neoplasia quando questa non c'è (*falsi positivi*).

### 5.1 Valutazione dei risultati ottenuti

I risultati ottenuti durante la *cross validation* appaiono in questa forma:

$$\textit{Accuratezza} = pp\% \quad (n/s_{im})$$

dove *pp* è la percentuale di precisione riscontrata nella classificazione rappresentata dal rapporto espresso come *n* (numero di sottoimmagini classificate correttamente) sul totale delle sottoimmagini disponibili *s<sub>im</sub>* per quella determinata immagine. La classifi-

cazione finale sarà quindi il risultato di due step di classificazione 'annidata': la prima, a livello di sottoimmagine, determina il numero di sottoimmagini classificate correttamente (veri positivi per immagini sane o affette da metaplasia/displasia) sul numero totale delle sottoimmagini di cui un'immagine è composta, mentre la seconda determina la percentuale di immagini classificate correttamente sull'intero set di dati. Quest'ultima è stata effettuata fissando una soglia all'80% di accuratezza nel primo step di classificazione, e conteggiando tutti i risultati che risultavano uguali o superiori a tale soglia. La seconda fase è quindi la più significativa in quanto determina la bontà degli algoritmi implementati.

Utilizzando l'insieme di *feature* raccolte con il blocco di dati (1) abbiamo una buona precisione per la classificazione, infatti si riscontra circa il 77.5% di classificazione corretta. Si noti inoltre che tutte le immagini che presentano la patologia dell'esofago di Barrett sono state riconosciute in modo eccellente ( $\simeq 99\%$  per ognuna delle immagini).

Per quanto riguarda il blocco (2), la precisione della classificazione è abbiamo una discreta precisione nella classificazione, infatti si ha per circa il 64.7% dei casi un riconoscimento corretto ( $Accuracy \geq 80\%$ ). Ciò dipende sia dal settaggio dei parametri per il classificatore SVM, sia dall'aver preso i primi dieci valori in scala logaritmica del vettore che poi è stato utilizzato per la creazione della matrice di classificazione  $F$ . Infine, si riportano in Figura 5.1 e 5.2 i plottaggi delle ROC ottenute rispettivamente dai due algoritmi.

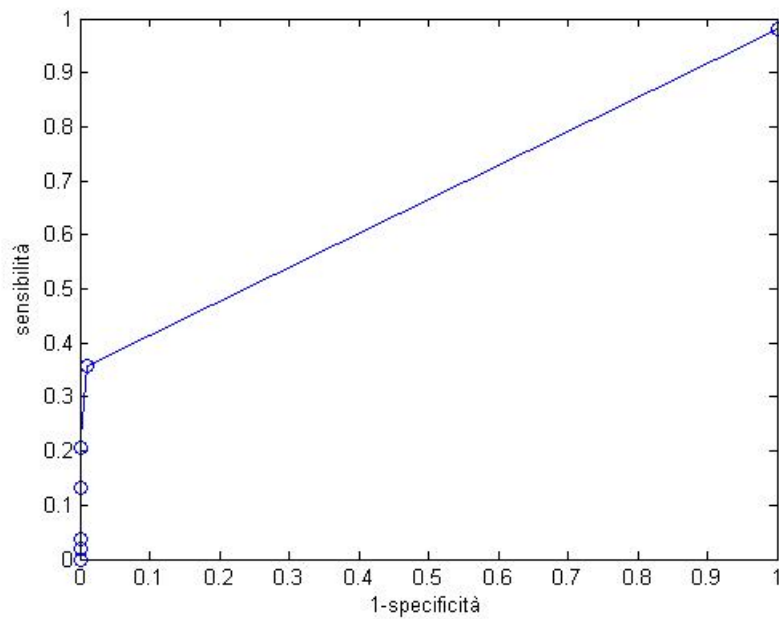


Figura 5.1: Grafico della ROC ottenuta da main1.

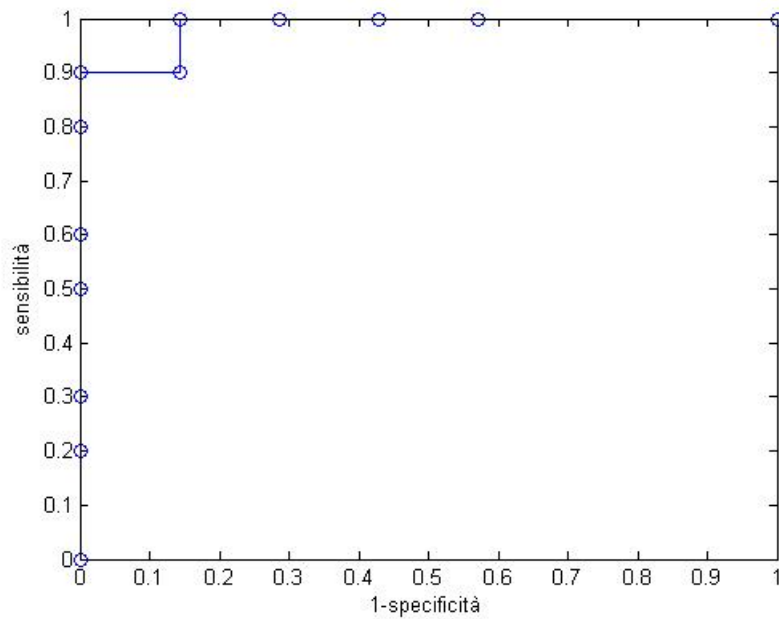


Figura 5.2: Grafico della ROC ottenuta da main2.



## Capitolo 6

# Conclusioni

Attraverso quest'elaborato, si è voluta valutare la capacità di un sistema automatico di analisi di distinguere le immagini di tessuti metaplasici da quelle di tessuti neoplasici. Tale obiettivo è stato raggiunto attraverso l'implementazione di due differenti algoritmi basati sull'estrazione di matrici di run length.

In entrambi i casi, la classificazione delle immagini è stata possibile mediante l'utilizzo di classificatori a supporto vettoriale settati diversamente in base ai data set di partenza. La bontà e l'efficienza della classificazione è stata stimata con l'aiuto di tecniche di cross validation e delle curve ROC ottenute a seguito del processo di classificazione stesso.

I risultati ottenuti dimostrano che gli algoritmi `main1` e `main2` consentono un buon grado di riconoscimento tra le classi etichettate con diversi valori di target (vedi pag. 26-27). Un confronto tra i due algoritmi permette di evidenziare come `main1` sia più efficiente di `main2` nella classificazione; ciò è legato alla notevole discrepanza tra il numero di dati presenti in ciascuno dei due data set.

### 6.1 Sviluppi futuri

#### Main 1

Per `main1` vi sarebbe la possibilità di modificare il codice di costruzione delle matrici di run length così da selezionare differenti feature o privilegiarne alcune piuttosto di altre per mettere in evidenza aspetti diversi delle immagini da analizzare. Per quanto riguarda l'alleggerimento del costo totale dell'algoritmo, dato che la parte di caricamento è già efficiente, l'unico aspetto su cui possiamo agire è la costruzione del modello. Questa è infatti la parte più pesante, sia come tempo di elaborazione sia come spazio in memoria occupato per il salvataggio dei dati. Per intervenire sulla costruzione del

modello potremmo settare in maniera differente i valori di gamma ( $-g$ ) e dei pesi di C ( $-w2$ ) che costruiscono il modello di addestramento del classificatore, così da avere una classificazione più precisa per ognuno dei data set utilizzati. In questo caso il problema è sempre quello di ricavare l'iperpiano che separa nel modo migliore le classi sacrificando magari un po' della precisione nella classificazione di una classe al fine però di aumentarla nelle altre. Infine, nella parte che riguarda il plottaggio della ROC potremmo studiare dei metodi di thresholding più adeguati per selezionare e discriminare tra veri positivi e veri negativi, basandoci ad esempio su varianza e correlazione.

## Main 2

Per quanto riguarda questo algoritmo è piuttosto difficile ridurre il costo temporale del caricamento dei dati: questo perchè le matrici caricate sono piuttosto pesanti. Si potrebbe comunque intervenire a priori sulla costruzione delle matrici di feature selezionando un minor numero di feature o processando un numero superiore di sottoimmagini. In questo modo si avrebbe il vantaggio di ottenere sottoimmagini di dimensione inferiore e acquisire pertanto una maggior rapidità nel caricamento.

Sul versante del miglioramento dell'efficienza nella classificazione si potrebbe intervenire principalmente aumentando il numero di campioni da classificare così da fornire all'SVM un numero maggiore di esempi sul quale basare il riconoscimento. Questa scelta però aumenterebbe in ogni caso il costo totale dell'algoritmo in quanto avremmo un tempo maggiore in fase di caricamento dei dati.

Per entrambi i due algoritmi bisogna in ogni caso sottolineare che i due data set utilizzati sono evidentemente sbilanciati. Questo sia quantitativamente che qualitativamente: infatti `main1` contiene complessivamente 338 immagini, mentre `main2` possiede 17 file. Inoltre, nello specifico, ognuno dei set ha al suo interno un numero non omogeneo di dati per tipologia di immagine. Pertanto, al fine di migliorare l'efficienza e la bontà del confronto tra i due algoritmi, sarebbero da bilanciare sia il numero dei dati per algoritmo -per esempio fornendo un maggior numero di immagini a `main2`- sia le varie tipologie di immagini all'interno dello stesso algoritmo.



# Appendice A

Vengono di seguito riportate le *feature* descritte nel paragrafo 3.1.2, ognuna delle quali con la rispettiva formulazione matematica. Le *feature* si intendono riferite per una matrice di run length  $p(i, j)$  avente  $M$  livelli di grigio e  $N$  il massimo valore di run length.

Feature	Formulazione
GLRLPNM	$p_p(i, j) = p(i, j) \cdot j$
GLRNV	$p_g(i) = \sum_{j=1}^N p(i, j)$
RLRNV	$p_r(j) = \sum_{i=1}^M p(i, j)$
GLRLOV	$p_o(i) = p(i, 1)$
SRE	$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{p(i, j)}{j^2}$
LRE	$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N p(i, j) \cdot j^2$
GLN	$\frac{1}{n_r} \sum_{i=1}^M \left( \sum_{j=1}^N p(i, j) \right)^2$
RLN	$\frac{1}{n_r} \sum_{j=1}^N \left( \sum_{i=1}^M p(i, j) \right)^2$
RP	$\frac{n_r}{n_p}$

Feature	Formulazione
LGRE	$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{p(i, j)}{i^2}$
HGRE	$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N p(i, j) \cdot i^2$
SRLGE	$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{p(i, j)}{i^2 \cdot j^2}$
SRHGE	$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{p(i, j) \cdot i^2}{j^2}$
LRLGE	$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{p(i, j) \cdot j^2}{i^2}$
LRHGE	$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N p(i, j) \cdot i^2 \cdot j^2$

## Appendice B

Si riportano di seguito gli script degli m-file utilizzati.

BARR\_RL:

```
function [rl,Xhist,Glev] = BARR_RL(xim,Ngl,Nscale,dbf);
```

```
dfactor = 256/Ngl;
```

```
Glev = 0:Ngl;
```

```
xred = double(xim)/max(xim(:));
```

```
xred = round(xred*256/dfactor);
```

```
if(Nscale^2>100)
```

```
    Xhist = linspace(1,Nscale^2,100);
```

```
else
```

```
    Xhist = 1:Nscale^2;
```

```
end;
```

```
for ct = 1:(Ngl+1),
```

```
    xbin = (xred==Glev(ct));
```

```
    xgl = bwlabel(xbin);
```

```
    bl = [];
```

```
    for ctlab = 1:max(xgl(:)),
```

```
        bl(ctlab) = sum(xgl(:)==ctlab);
```

```
    end;
```

```
    rl(ct,:) = hist(bl,Xhist);
```

```
end;
```

```
BARR_RL_tex:
```

```
function [rltex,rl,Xhist,Glev] = BARR_RL_tex(xim,Ng1,Nscale,dbf);
```

```
[rl,Xhist,Glev] = BARR_RL(xim,Ng1,Nscale,dbf);
```

```
sz = size(rl);
```

```
Lemph = ones(sz(1),1)*(Xhist.^2);
```

```
Gemph = (Glev.^2)'*ones(1,sz(2),1);
```

```
Nfact = sum(rl(:));
```

```
ShR_Emph = sum(rl(:)./Lemph(:))/Nfact;
```

```
LoR_Emph = sum(rl(:).*Lemph(:))/Nfact;
```

```
ShR_LoGL_Emph = sum(rl(:)./Lemph(:)./(1+Gemph(:)))/Nfact;
```

```
LoR_LoGL_Emph = sum(rl(:).*Lemph(:)./(1+Gemph(:)))/Nfact;
```

```
ShR_HiGL_Emph = sum(rl(:).*Lemph(:).*Gemph(:))/Nfact;
```

```
LoR_HiGL_Emph = sum(rl(:).*Lemph(:).*Gemph(:))/Nfact;
```

```
rltex = [ShR_Emph,LoR_Emph,ShR_LoGL_Emph,LoR_LoGL_Emph,ShR_HiGL_Emph,LoR_HiGL_Emph];
```

```
testBARR_RL:
```

```
clear all
```

```
close all
```

```
dbf = 0;
```

```
NScale = 32; %% Dimension of ROI for RL texture extraction
```

```
Ng1 = 8; %% Number of gray levels
```

```

load('DataGas');

for ct = 1:size(x05,2)
    im = x05(ct).image;
    imname = x05(ct).file;
    imname = imname(1:end-4);
    impat = x05(ct).patcode;

    passo_col = size(im,1)/NScale;
    passo_row = size(im,1)/NScale;
    xtex = zeros(passo_row,passo_col,6);
    for ctx = 1:NScale
        for cty = 1:NScale
            xroi = im((ctx-1)*passo_col+1:ctx*passo_col,(cty-1)*passo_row+1:cty*passo_row);
            [rltex,rl,Xhist,Glev] = BARR_RL_tex(xroi,Ngl,NScale,dbf);
            xtex(ctx,cty,:) = rltex;
        end;
    end;

    strtosave = sprintf('TextRLM_%03i_%s', impat,imname);
    save(strtosave, 'xtex')
end;

main1:

clear all, close all, clc,

%% Parameters

n_bar = 6;
rows = 32;
N_sub = 1024;
im_tot = 1024*338;
patient = zeros(1,im_tot);

```

```

cty = 1;

%% DataGas

s = what('DataGas');
matfiles = s.mat;
F1 = zeros(rows*rows*numel(matfiles),n_bar);
target1 = zeros(1,size(F1,1));
sz1 = size(matfiles,1);

for j = 1:numel(matfiles),
    t(j) = load(char(matfiles(j)));
    for i = 1:(rows*rows),
        patient(cty) = j;
        cty = cty+1;
    end;
end;

for z = 1:numel(matfiles),
    xtex = t(z).xtex;
    for m = 1:n_bar,
        for k = 1: rows,
            for i = 0: rows-1,
                T1(i+(rows*k-(rows-1)),m) = xtex(k,(i+1),m);
            end;
        end;
    end;
    if(sum(sum(F1))==0)
        F1 = T1;
    else
        F1 = [F1;T1];
    end;
end;

for ctx = 1:size(target1,2),
    target1(ctx) = -1;

```

```

end;

%% DataBar

s = what('DataBar');
matfiles = s.mat;
F2 = zeros(rows*rows*numel(matfiles),n_bar);
target2 = zeros(1,size(F2,1));
sz2 = size(matfiles,1);

for j = 1:numel(matfiles),
    t(j) = load(char(matfiles(j)));
    for i = 1:(rows*rows),
        patient(cty) = j+sz1;
        cty = cty+1;
    end;
end;

for z = 1:numel(matfiles),
    xtex = t(z).xtex;
    for m = 1:n_bar,
        for k = 1: rows,
            for i = 0: rows-1,
                T2(i+(rows*k-(rows-1)),m) = xtex(k,(i+1),m);
            end;
        end;
    end;
    if(sum(sum(F2))==0)
        F2 = T2;
    else
        F2 = [F2;T2];
    end;
end;

for ctx = 1:size(target2,2),
    target2(ctx) = 0;

```

```

end;

%% DataNpl

s = what('DataNpl');
matfiles = s.mat;
F3 = zeros(rows*rows*numel(matfiles),n_bar);
target3 = zeros(1,size(F3,1));

for j = 1:numel(matfiles),
    t(j) = load(char(matfiles(j)));
    for i = 1:(rows*rows),
        patient(cty) = j+sz1+sz2;
        cty = cty+1;
    end;
end;

for z = 1:numel(matfiles),
    xtex = t(z).xtex;
    for m = 1:n_bar,
        for k = 1: rows,
            for i = 0: rows-1,
                T3(i+(rows*k-(rows-1)),m) = xtex(k,(i+1),m);
            end;
        end;
    end;
    if(sum(sum(F3))==0)
        F3 = T3;
    else
        F3 = [F3;T3];
    end;
end;

for ctx = 1:size(target3,2),
    target3(ctx) = 1;
end;

```



```

F = [F1;F2;F3];
target = [target1,target2,target3];

%% Cross validation

ind = 1;
for ct = 1:max(patient),

    sel = find(patient~=ct);
    tmp = rand(length(sel),1);
    [dummy,sind] = sort(tmp);
    ttrain = target(sel(sind(1:2000)));
    ftrain = [F(sel(sind(1:2000)),:)];
    nn = find(patient==ct);

    if(~isempty(nn))
        ttest = target(nn);
        ftest = [F(nn,:)];
        model = svmtrain(double(ttrain>0.5)',ftrain,'-t 2 -q -w2 1 -g 0.0000001');
        pl = svmpredict(ttest',ftest,model);
        pn = sum(pl==1);
        pb = sum(pl==0);
        res(ct,:) = [ttest(1),pn>pb,pn,pb];

        Nim = length(ttest)/N_sub;
        for ctz = 1:Nim,
            for ctx = 1:rows,
                for cty = 1:rows,
                    imres(ctx,cty) = pl((ctz-1)*N_sub+(ctx-1)*rows+cty,:);
                end;
            end;
            nt = sum(imres(:)==1);
            nf = sum(imres(:)==0);
            resim(ind,:) = [ttest(1),nt>nf,nt,nf];
            ind = ind+1;
        end;
    end;
end;

```

```

        end;
    else
        res(ct,:) = [0,0,0,0];
    end;
end;

nc = resim(:,3)./N_sub;
th = 0:0.05:1;
for ctt = 1:length(th),
    TP = sum(resim(:,1)==1 & nc>th(ctt));
    TN = sum(resim(:,1)==0 & nc<th(ctt));
    sens(ctt) = TP/sum(resim(:,1)==1);
    spec(ctt) = TN/sum(resim(:,1)==0);
end;
plot(1-spec,sens,'o-')

main2:

clear all, close all, clc,

s = struct([]);
load('VolTextRLM_001_esofago00011'); s(1).vol = vol_textRLM;
load('VolTextRLM_001_esofago00013'); s(2).vol = vol_textRLM;
load('VolTextRLM_002_esofago00072'); s(3).vol = vol_textRLM;
load('VolTextRLM_004_esofago00246'); s(4).vol = vol_textRLM;
load('VolTextRLM_020_esofago00088'); s(5).vol = vol_textRLM;
load('VolTextRLM_021_esofago00197'); s(6).vol = vol_textRLM;
load('VolTextRLM_058_esofago00030'); s(7).vol = vol_textRLM;
load('VolTextRLM_001_esofago00008'); s(8).vol = vol_textRLM;
load('VolTextRLM_001_esofago00009'); s(9).vol = vol_textRLM;
load('VolTextRLM_001_esofago00012'); s(10).vol = vol_textRLM;
load('VolTextRLM_002_esofago00002'); s(11).vol = vol_textRLM;
load('VolTextRLM_002_esofago00003'); s(12).vol = vol_textRLM;
load('VolTextRLM_002_esofago00004'); s(13).vol = vol_textRLM;
load('VolTextRLM_003_esofago00016'); s(14).vol = vol_textRLM;
load('VolTextRLM_003_esofago00018'); s(15).vol = vol_textRLM;

```

```

load('VolTextRLM_004_esofago00034'); s(16).vol = vol_textRLM;
load('VolTextRLM_004_esofago00037'); s(17).vol = vol_textRLM;

%% Parameters

n_Barr = 7;
n_bar = 10;
n_righe = size(vol_textRLM,3)*size(s,2);
F = zeros(n_righe,n_bar);
target = zeros(1, n_righe);
patient = zeros(1, n_righe);
dim_sub = 128;
N_rows = 1024/dim_sub;
N_sub = (1024/dim_sub)^2;

%% Matrice e vettore di classificazione (F,target)

ctx = 1;
for k = 1:n_Barr,
    im = s(k).vol;
    for j = 1:size(vol_textRLM,3),
        v = hist(squeeze(im(4,:,j)),logspace(0,3.61,n_bar));
        for c = 1:n_bar,

            F(ctx,c) = v(c);

        end;
        patient(ctx) = k;
        target(ctx) = 0;
        ctx = ctx+1;
    end;
end;

for k = (n_Barr+1):size(s,2),
    im = s(k).vol;
    for j = 1:size(vol_textRLM,3),

```

```

v = hist(squeeze(im(4,:,j)),logspace(0,3.61,n_bar));
for c = 1:n_bar,

    F(ctx,c) = v(c);

end;
patient(ctx) = k;
target(ctx) = 1;
ctx = ctx+1;
end;
end;

%% Cross validation

ind = 1;

for ct = 1:max(patient),

    sel = find(patient~=ct);
    ttrain = target(sel);
    ftrain = [F(sel,:)];
    nn = find(patient==ct);

    if(~isempty(nn))
        ttest = target(nn);
        ftest = [F(nn,:)];

        model = svmtrain(double(ttrain>0.5)',ftrain,'-t 2 -q -w1 0.7 -g 0.000001');
        pl = svmpredict(ttest',ftest,model);
        pn = sum(pl==1);
        pb = sum(pl==0);
        res(ct,:) = [ttest(1),pn>pb,pn,pb];
        Nim = 1;

    for ctz = 1:Nim,
        for ctx = 1:N_rows,

```

```

        for cty = 1:N_rows,
            imres(ctx,cty) = pl((ctz-1)*N_sub+(ctx-1)*N_rows+cty,:);
        end;
    end;
    nt = sum(imres(:)==1);
    nf = sum(imres(:)==0);
    resim(ind,:) = [ttest(1),nt>nf,nt,nf];
    ind = ind+1;
end;
else
    res(ct,:) = [0,0,0,0];
end;
end;

nc = resim(:,3)./N_sub;
th = 0:0.05:1;
for ctt = 1:length(th),
    TP = sum(resim(:,1)==1 & nc>th(ctt));
    TN = sum(resim(:,1)==0 & nc<th(ctt));
    sens(ctt) = TP/sum(resim(:,1)==1);
    spec(ctt) = TN/sum(resim(:,1)==0);
end;
plot(1-spec,sens,'o-')

```



# Bibliografia

- [1] Tang X., (1998) , *Texture Information in Run-Length Matrices*, IEEE Transaction in Image Processing, 7.
- [2] Tamura H., Mori S., Yamawaki T., (1978), *Textural features corresponding to visual perception*, IEEE Transactions on Systems, Man and Cybernetics 8: 460–473.
- [3] Rao A., Lohse G., (1993), *Identifying high-level features of texture perception*, Models Image Process, 57(3): 218–233.
- [4] Dionigi R., (2006) , *Chirurgia, basi teoriche e chirurgia generale*, Masson Editore, Quarta edizione.
- [5] *Confocal Laser Endomicroscopy*, Gastrointestinal endoscopy journal, ASGE, Vol.70.
- [6] Trovato C. et al., (2011) , *Endomicroscopia: presente e futuro*, Giornale italiano endoscopia digitale, 34:213-219.
- [7] Hoffman A. et al., (2006) , *Confocal laser endomicroscopy: technical status and current indications*, Endoscopy, Vol.38:1275-1283.
- [8] Matloff J. et al., (2011), *High-resolution and optical molecular imaging for the early detection of colonic neoplasia*, Grastrointestinal Endoscopy, vol 73, 6.
- [9] Davies E.R., *Handbook of texture analysis*, Imperial College Press.
- [10] Mäenpää T., (2003), *The Local Binary Pattern Approach to Texture Analysis - Extensions and Applications*.
- [11] Hill P.R., (2002) , *Wavelet Based Texture Analysis and Segmentation for Image Retrieval and Fusion*.
- [12] Haralick R.M., (1979) , *Statistical and Structural Approaches*, IEEE Proceedings, 4.

- [13] Galloway M.M., (1975), *Texture analysis using gray level run lengths*, Comput. Graphics Image Process, 4:172-179.
- [14] Hung C.C, Pham M., Arasteh S., Kuo B.C., Coleman T., (2006) , *Image Texture Classification Using texture Spectrum and Local Binary Pattern*, IEEE International Conference on Geoscience and Remote Sensing Symposium.
- [15] Ojala T. et al., (2002) , *Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns*, IEEE Transactions on pattern analysis and machine intelligence, Vol.24.
- [16] Xu D. et al., *Run length encoding for volumetric texture*, Intelligent Multimedia Processing Laboratory, School of Computer Science, Telecommunications and Information Systems, Chicago.
- [17] Serpico S.B., *Probabilità di errore di un classificatore supervisionato*, Dipartimento di Ingegneria Biofisica ed Elettronica, Università di Genova.
- [18] Maltoni D., (2010), *Classificazione supervisionata*, Visione Artificiale, Scienze dell'Informazione, Università di Bologna.
- [19] *Endoscopia* [online]. Disponibile da: <http://it.wikipedia.org/wiki/Endoscopia> [consultato il 16 gennaio 2012].
- [20] Ancona E. et al., *Tumore dell'esofago* [online]. Disponibile da: <http://www.startoncology.net> [consultato il 17 gennaio 2012].
- [21] Tanzi F., *Microscopia confocale* [online]. Disponibile da: [http://www-3.unipv.it/webtanzi/13\\_Microscopia%20confocale.pdf](http://www-3.unipv.it/webtanzi/13_Microscopia%20confocale.pdf) [consultato il 23 marzo 2012].
- [22] Devescovi D., (2007) *Impiego di classificatori nell'analisi di immagini* [online]. Disponibile da: <http://www.devedeve.com/wp-include/files/Impiego%20di%20Classificatori.pdf> [consultato il 24 marzo 2012].
- [23] *Libreria per Support Vector Machine* [online]. Disponibile da: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/> [consultato il 3 febbraio 2012].