



UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Corso di Laurea triennale in
INGEGNERIA BIOMEDICA

Elaborazione di Immagini di Lesioni Cutanee per
l'Estrazione di Caratteristiche Utili Alla Classificazione
Automatica

Processing images of skin lesions to useful features extraction to
automatic classification

Relatore:
Prof. Francesco Piva

Tesi di Laurea di:
Luigi Gabriel Troconis

A.A. 2019/2020

Indice

Introduzione	3
1. Melanoma Cutaneo e diagnosi precoce	4
1.1 Il melanoma Cutaneo	4
1.1.1 Fattori di rischio	6
1.1.2 I sintomi	7
2. Applicazioni algoritmiche	9
2.1.1 Algoritmo ABCD	9
2.1.2 Modello di funzionalità ibrido	12
2.1.3 Miglioramento della diagnostica tramite le immagini digitali	13
3. Algoritmo	15
3.1 Image Pre-Processing	16
3.2 Image Segmentation	21
3.3 Feature Extraction	26
3.3.1 Caratteristica n° 1	29
3.3.2 Caratteristica n° 2	29
3.3.3 Caratteristica n° 3	29
3.3.4 Caratteristica n° 4	31
3.3.5 Caratteristica n° 5	33
3.3.6 Caratteristica n° 6	34
3.3.7 Caratteristiche dalla n° 7 alla n° 9	34
Conclusione	36
Bibliografia	37

Introduzione

L'incremento dei casi di melanomi e la complessità di diagnosi portano spesso gli specialisti a dover usufruire di strumenti di supporto alla diagnosi.

Ovviamente la presenza di questi macchinari è di supportare e di facilitare la diagnosi dello specialista in casi particolarmente complessi di lesioni cutanee.

Tra le molte possibilità di supporto nella classificazione di lesioni cutanee ci si è concentrato negli algoritmo di classificazione automatica dei nevi.

Lo scopo di questa Tesi è elaborare un algoritmo che possa essere di supporto nella classificazione automatica di lesioni cutanee tramite l'estrazioni di caratteristiche utili che rendano possibile l'aiuto necessario allo specialista per una possibile diagnosi di melanoma cutaneo.

Il primo ed il secondo capitolo dell'elaborato forniscono al lettore alcune delle principali nozioni di base riguardanti le lesioni cutanee, sulle loro caratteristiche e sulla diagnosi; in particolare il secondo di applicazioni algoritmiche già esistenti.

Infine l'ultimo capitolo contiene la spiegazione dell'algoritmo proposto in questa Tesi.

Capitolo 1

1. Melanoma Cutaneo e diagnosi precoce

Una neoplasia o un tumore è una massa anomala di tessuto che cresce in eccesso e in modo incontrollato rispetto ai tessuti normali. Ogni tumore viene generalmente classificato, in base alla propria aggressività, in:

- Tumore **benigno**, caratterizzato da cellule che mantengono, almeno in parte, le caratteristiche morfologiche del tessuto originario e restano confinate all'interno di un tessuto connettivo; così facendo comprimono i tessuti circostanti senza distruggerli direttamente.
- Tumore **maligno**, anche detto cancro, presenta cellule che tendono a diffondersi all'interno dei tessuti e degli organi vicini, propagandosi attraverso il sistema linfatico o i vasi sanguigni e generando spesso delle metastasi.

Le metastasi tumorali derivano dalla crescita di cellule maligne distaccatesi dal tumore originario, ma sono situate in diversi siti.

Nonostante questa classificazione ci sono comunque situazioni intermedie che presentano caratteristiche dell'una e dell'altra classe.

1.1 Il melanoma Cutaneo

Il melanoma cutaneo è un tumore maligno che ha origine nei melanociti, le cellule epiteliali responsabili della produzione di melanina, perciò si origina nella cute o, più raramente, negli occhi o nelle mucose.

Inoltre i melanociti sono i responsabili, in condizioni normali, della presenza di macchioline scure sulla superficie della pelle conosciute come nei, definite in gergo clinico nevi.

Il melanoma può manifestarsi su una pelle integra, oppure da nevi situati sulla cute che possono essere presenti sin dalla nascita o dalla prima infanzia (congeniti) oppure comparsi nel corso della vita (acquisiti).

Clinicamente si possono distinguere quattro tipologie di melanoma cutaneo:

1. *melanoma lentiginoso acrale* : è il tipo di melanoma più raro nelle persone con carnagione chiara e si manifesta alle estremità degli arti, specialmente nelle sedi palmo-plantari e subungueali. Questo particolare melanoma ha una fase intraepidermica rapida e la comparsa di una regione nodulare rispecchia l'inizio di una crescita verticale del tumore.
2. *melanoma di tipo lentigo maligna* : è poco frequente (circa il 5-10% dei casi) e insorge soprattutto nelle persone anziane; tende a comparire in aree fotoesposte e danneggiate da un'esposizione solare cronica e pertanto si manifesta principalmente sul volto, dove assume l'aspetto di una macchia asimmetrica e piana che va dal bruno pallido al bruno nerastro. La sua evoluzione è piuttosto lenta e solo in fase avanzata possono comparire noduli sulla superficie.
3. *melanoma nodulare* : è una tipologia di melanoma caratterizzata da una crescita radiale scarsa o addirittura assente e da una crescita verticale sin dall'inizio; per questo motivo è molto aggressivo e spesso presenta metastasi alla diagnosi. Si riscontra nel 10-15% dei casi di melanoma e specialmente nei maschi intorno ai 50-60 anni; la sua diagnosi è difficile in quanto non dà sintomi e a volte non presenta la caratteristica colorazione tumorale.
4. *melanoma a diffusione superficiale* : è il melanoma più comune, circa il 60-70% dei casi, e spesso ha un decorso bifasico: dapprima ha una crescita orizzontale stabile e lenta che si manifesta come una lesione maculare; successivamente interviene una crescita verticale che corrisponde ad un'invasione in profondità e ha un'evoluzione della lesione in placca con aree policromatiche o talvolta chiare.

1.1.1 Fattori di rischio

Il rischio di sviluppare un melanoma è legato sia a fattori ambientali che a fattori genetici propri dell'individuo.

I raggi del sole non sono l'unica causa del melanoma cutaneo, è importante prestare attenzione anche ad altri fattori di rischio. La possibilità di sviluppare un melanoma è strettamente dipendente dall'interazione tra fattori di rischio genetico-costituzionali e ambientali.

Vengono riconosciuti quali cause che possono favorire lo sviluppo del melanoma in maniera indipendente e statisticamente significativa:

- **fototipo e pigmentazione cutanea:** carnagione chiara, capelli e occhi chiari, presenza di lentiggini (fototipo 1-2). I soggetti con fototipo 1-2 hanno un rischio di melanoma doppio rispetto ai soggetti con pelle scura/olivastra e con occhi, capelli scuri/neri e fototipo 4
- **familiarità:** l'appartenenza a una famiglia in cui si è verificato almeno un altro caso di melanoma. Il 10% dei pazienti affetti da melanoma riferisce almeno un altro caso nell'ambito della propria famiglia
- **genetica:** la mutazione nel gene CDKN2A. La probabilità di ereditare la mutazione aumenta in maniera proporzionale all'aumento del numero dei membri della famiglia affetti da melanoma
- **numero nei:** la presenza di numerosi nei acquisiti, oltre 100, o la presenza di 5 o più nei atipici, cioè di diametro superiore a 6 mm, bordo irregolare, colore variegato
- **aver già avuto in precedenza un melanoma:** la probabilità cumulativa di sviluppare un secondo melanoma a 5 e 10 anni di distanza è rispettivamente del 2,8% e del 3,6%
- **scottature solari** durante l'infanzia e l'adolescenza
- **esposizione** al sole intense ed intermittenti.
- **presenza di un neo congenito gigante**

Classificazione dei diversi fototipi:

Fototipo 1: Capelli biondi o rossi, occhi chiari, carnagione molto chiara con efelidi, estremamente sensibile al sole, si scotta sempre e non si abbronzia.

Fototipo 2: Capelli biondi o castano chiaro, occhi chiari, carnagione chiara, spesso con efelidi, sensibile al sole, si scotta con facilità e si abbronzia con difficoltà.

Fototipo 3: Capelli castani, occhi chiari o marroni, carnagione bruno-chiara, può scottarsi, ma si abbronzia.

Fototipo 4: Capelli castano scuro o neri, occhi scuri, carnagione olivastra o scura, si scotta di rado e si abbronzia con facilità.

Fototipo 5: Capelli neri, occhi scuri, carnagione bruno-olivastra si abbronzia intensamente.

Fototipo 6: Capelli neri, occhi neri, carnagione nera, non si scotta mai.

1.1.2 I sintomi

Come detto precedentemente, i melanomi possono formarsi sia da nevi congeniti che da nevi acquisiti. Il primo passo per distinguere un nevo da un melanoma è sintetizzato nella regola dell'ABCDE:

- **Asimmetria nella forma** : un neo è generalmente circolare o tondeggiante, mentre un melanoma ha una forma più irregolare
- **Bordi irregolari** : un melanoma, a differenza di un neo benigno, presenta bordi frastagliati
- **Colore disomogeneo** : i nei sono generalmente caratterizzati da una colorazione uniforme contrariamente ai melanomi che invece presentano colorazioni diverse al proprio interno
- **Diametro superiore a 6 mm** : generalmente le dimensioni massime di un neo si aggirano intorno ai 6 mm di diametro, mentre un melanoma è caratterizzato da uno sviluppo, sia in larghezza che in spessore
- **Evoluzione** : a differenza dei nei, i melanomi cambiano aspetto in un tempo piuttosto breve.

Altri campanelli d'allarme che devono essere valutati da un medico sono un neo che sanguina, che prude o che `e circondato da un nodulo o da un'area arrossata.

Chiaramente la presenza di tutte queste caratteristiche agevola il riconoscimento e la diagnosi di un melanoma, tuttavia quando il melanoma `e nelle fasi iniziali o quando solo alcuni caratteri clinici dell'ABCDE vengono riscontrati in un nevo la diagnosi può risultare complessa.

La difficoltà della valutazione clinica può provocare sia un alto tasso di inutili rimozioni di nevi, sia una sottostima della diagnosi. Per questa ragione, al fine di incrementare la sensibilità diagnostica, lo specialista può ricorrere all'utilizzo di strumenti che consentono osservazioni migliori dell'osservazione diretta ad occhio nudo.

Capitolo 2

2. Applicazioni algoritmiche

Si cerca la possibilità, in modo oggettivo, di diagnosticare il tumore cutaneo tramite lo sviluppo ed il miglioramento di algoritmi per la determinazione di caratteristiche più importanti per la categorizzazione della lesione cutanea. Tale si possa avere metodi diagnostici computerizzati, che possono essere basati su fotografie dermatologiche (facilitano il processo di diagnosi del melanoma), ma la maggior parte di questi algoritmi sono basati su caratteristiche topografiche superficiali, che possono essere limitanti nella caratterizzazione del tumore. Di seguito si descrive l'algoritmo ABCD, basato con caratteristiche superficiali; ed un modello di funzionalità ibrido, composto da un insieme di funzionalità di alto livello (HLIF) e di basso livello (LLF).

2.1.1 Algoritmo ABCD

La regola della demoscopia ABCD(E) è stato il primo algoritmo di dermatoscopia creato per aiutare a differenziare i tumori melanocitici benigni da quelli maligni. Algoritmo sviluppato per rispondere quantitativamente alla questione cruciale nella demoscopia se una lesione cutanea melanocitica sotto esame è benigna, sospetta o maligna. Basata solo su quattro criteri, rendendolo facile da imparare e applicare. I criteri che si combinano per creare la regola ABCD(E) della demoscopia sono: “**A**simmetria”, “**B**ordo”, “**C**olore”, “**s**trutture **D**ifferenziali” infine, se è possibile, “**E**voluzione” della lesione. Per utilizzare questi criteri, è stato sviluppato un sistema di punteggio per calcolare il “Total Dermascopy Score”, (TDS) usando un equazione lineare. Tramite questo TDS, vi è la possibilità di classificare le lesioni rispetto la probabilità che le lesioni in esame siano maligne.

Asimmetria

Le lesioni vengono divise in due da due assi ortogonali posizionati in modo tale da dividere la lesione nel suo piano più simmetrico, ottenendo così il punteggio più basso possibile di asimmetria. L'asimmetria tiene conto della distribuzione dei colori e delle strutture su entrambi gli assi all'interno della lesione. Il punteggio di asimmetria è 0, se sono presenti simmetrie in entrambi gli assi all'interno della lesione. Se c'è asimmetria in un asse il punteggio dato è 1; in entrambi gli assi 2. Il punteggio di asimmetria deve essere moltiplicato per il fattore 1,3 per calcolare il TDS. (è importante rendersi conto che molte lesioni che appaiono clinicamente simmetriche possono essere dermoscopicamente asimmetriche rivelando una distribuzione asimmetrica di colori e strutture).

Bordo

la valutazione del punteggio limite si basa sul se vi è un taglio netto e brusco del modello di pigmento alla periferia della lesione o un taglio graduale e indistinto. Ai fini dell'analisi, le lesioni sono divise in otto. Il punteggio massimo del bordo è 8, viene dato quando l'intero bordo (cioè tutti gli otto segmenti) della lesione rivela un taglio retto. Se il bordo della lesione in tutti e otto i quadranti ha un taglio indistinto, viene dato il punteggio minimo di 0. Il punteggio è compreso tra 8 e 0 ed il suo fattore è di 0,1 per ottenere il TDS.

Colore

Nella determinazione del punteggio del colore, i seguenti sei colori sono considerati importanti. La presenza di ciascun colore vale 1 punto (bianco, rosso, marrone chiaro, marrone scuro, grigio bluastrò, nero) ed il fattore è 0,5 per il TDS.

Quattro colori corrispondono alla distribuzione della melanina (cioè marrone chiaro e scuro, melanina nello strato superficiale; il nero, melanina nello strato corneo o in tutti gli strati dell'epidermide; il grigio bluastrò, melanina nel derma papillare). Il colore bianco corrisponde alle aree di regressione ed il colore rosso riflette il grado di infiammazione o neovascolarizzazione. Il bianco è considerato se rispetto al colore adiacente della pelle normale è nettamente più chiaro (più bianco).

Strutture dermatoscopiche

Per la valutazione vengono considerate cinque caratteristiche principali. Per la presenza di ciascuna struttura viene attribuito un valore di 1 (aree senza struttura, rete di pigmenti, striature ramificate (rete atipica), punti, globuli). Il fattore di moltiplicazione è 0,5.

In generale, i melanomi mostrano più strutture dermoscopiche rispetto ai nevi. Il novanta per cento dei nevi melanocitici rivela tre o meno componenti strutturali. Al contrario, il 73% dei melanomi rivela quattro o più componenti strutturali.

È importante essere consapevoli di alcuni avvertimenti riguardanti la regola ABCD:

- Innanzitutto, i punti e i globuli di qualsiasi colore (incluso il rosso) sono considerati significativi.
- In secondo luogo, le aree senza struttura sono considerate presenti se comprendono almeno il 10% della superficie della lesione. (Queste aree senza struttura possono essere iperpigmentate.)
- In terzo luogo, soltanto se ce ne sono più di due punti e/o striature ramificate vengono conteggiati, inoltre i globuli vengono contati se n'è presente almeno uno all'interno della lesione.

La combinazione dei criteri implica la determinazione del punteggio totale della dermatoscopia.

Calcolo del punteggio totale della dermatoscopia (TDS)

Una distinzione accurata tra lesioni melanocitiche benigne e maligne può essere fatta se i singoli punteggi di asimmetria, bordo, colore e strutture dermoscopiche (i componenti della regola ABCD della dermatoscopia) sono moltiplicati per i coefficienti 1.3, 0.1, 0.5 e 0.5, rispettivamente, producendo un TDS.

Per calcolare una formula per un TDS che discrimina in modo affidabile le lesioni melanocitiche benigne da quelle maligne si è fatto uso di una rete neurale artificiale, provata con 157 casi istologicamente provati che sono stati equamente e casualmente divisi tra un set di allenamento e test. Ad una soglia di 5,45, è stata ottenuta una precisione diagnostica del 94% (L'accuratezza diagnostica misura il numero di diagnosi corrette di melanoma rispetto a tutti i casi di melanoma e tutti i casi diagnosticati erroneamente come melanoma). Nel set di allenamento, la specificità era del 90% e la sensibilità era del 100%, il che indica che tutti i melanomi nel set di allenamento avevano un punteggio di dermatoscopia maggiore di 5,45 e che solo circa il 10% dei nevi melanocitici era classificato erroneamente come maligno. Nel set di test, l'accuratezza diagnostica era del 92%, la sensibilità era del 98% e la specificità era del 90%, a conferma dell'elevato valore diagnostico della regola ABCD della dermatoscopia. Nel tentativo di mantenere la massima sensibilità per la diagnosi del melanoma, la regola ABCD è stata ulteriormente perfezionata classificando come "sospette" tutte le lesioni il cui punteggio di dermatoscopia cade tra 4,75 e 5,45. Queste lesioni sospette devono essere esportate o monitorate molto attentamente con una soglia bassa di escissione in caso di cambiamenti.

Eccezioni

Sono presenti eccezioni alla regola ABCD della dermatoscopia. I nevi melanocitici con una componente lentiginosa potrebbero non essere classificati in modo affidabile tramite esame clinico o dermoscopico e queste lesioni sembrano spesso sospette sotto la dermatoscopia. TDS falsi positivi sono comuni anche quando si valutano nevi melanocitici con globuli o nevi con una superficie papillomatosa. Inoltre, il nevo di Spitz/Reed, il nevo melanocitico congenito, il nevo di spago, il nevo agminato, il nevo melanocitico ricorrente e il lentigo a macchia di inchiostro possono avere un punteggio TDS elevato. In questi casi la visione e la diagnosi tipiche della "Gestalt" sono superiori alla diagnosi basata sul calcolato TDS. Le lesioni localizzate su viso, palmi, soles e mucose non possono essere analizzate in modo affidabile usando la regola ABCD.

2.1.2 Modello di funzionalità ibrido

ha la finalità di caratterizzare una lesione cutanea tramite la combinazione di funzionalità di basso livello ed alto livello; aumentandole con una serie di funzionalità fisiologiche estratte da fotografie dermatologiche usufruendo un modello non lineare (che sia il più vicino possibile) per migliorare le prestazioni di classificazione. Le caratteristiche fisiologiche estratte dalla lesione per il modello ibrido includono quelle basate su:

- i) concentrazioni di eumelanina
- ii) concentrazione di feomelanina
- iii) saturazione di ossigeno nel sangue.

L'analisi del melanoma

è basato su tre categorie principali di funzionalità:

- i) funzionalità intuitive di alto livello (HLIF)

L'utilizzo di funzionalità di livello superiore che sono state specificamente progettate per l'applicazione della valutazione di una fotografia dermatologica di una lesione in base ai criteri ABCD ha dimostrato di migliorare la sensibilità di classificazione e ha anche dimostrato di popolare efficacemente lo spazio delle caratteristiche.

- ii) funzionalità di basso livello (LLF)

Utilizzando le tecniche di elaborazione delle immagini esistenti, è possibile estrarre e combinare numerose funzionalità per fornire una valutazione obiettiva di una lesione cutanea rispetto ai criteri ABCD.

- iii) funzionalità fisiologiche.

Il melanoma cutaneo è caratterizzato dalla crescita cancerosa di cellule che producono melanina chiamate melanociti. I melanociti esistono nello strato basale dell'epidermide. Nell'anatomia cutanea, la "melanina" si riferisce comunemente ai sottotipi eumelanina e feomelanina, che contribuiscono al colore della pelle di un individuo. L'eumelanina è responsabile della pigmentazione marrone-nera, mentre la feomelanina è responsabile della pigmentazione rosa-rossa. Una maggiore densità complessiva della melanina provoca una pelle più scura, poiché la struttura molecolare della melanina è particolarmente adatta ad assorbire la luce ultravioletta e visibile. Normalmente, la distribuzione della melanina è omogenea su una piccola distanza, ottenendo un colore uniforme. Il melanoma, d'altra parte, può provocare una densità eterogenea

della melanina dovuta alla crescita locale dei melanosomi. Inoltre, a causa dell'ampia raccolta di cellule presenti nel melanoma, si verifica angiogenesi che provoca un massiccio assorbimento di sangue ossigenato nell'area. Per cui si può considerare anche le concentrazioni di ossigeno nel sangue, oltre che le concentrazioni di eumelanina e di feomelanina.

2.1.3 Miglioramento della diagnostica tramite le immagini digitali

Quest'anno il melanoma interesserà circa 73 mila nuovi casi e causerà 9 mila decessi, ma una diagnosi precisa rimane un problema molto serio. Senza una diagnosi precoce e cure preventive, il melanoma può rapidamente diffondersi fino a diventare fatale. A causa della non esistenza di un biomarcatore per il melanoma nell'uso clinico provoca che gli attuali criteri diagnostici per le lesioni cutanee rimangano soggettivi ed imprecisi. Un'accurata diagnosi del melanoma si basa su un standard istopatologico; cioè l'escissione della lesione cutanea melanocitica.

Esiste un'elevata morbilità nella diagnosi errata del melanoma, ad esempio la progressione della malattia in un falso negativo contro i rischi di un intervento chirurgico (non necessario) per un falso positivo.

Attualmente, il "Dermatoscopic imaging" manuale è lo standard usato nella selezione delle lesioni cutanee atipiche per la biopsia e raggiunte una sensibilità del 90% ma solo il 59% di specificità quando è eseguita da un dermatologo esperto.

Inoltre molti algoritmi di visione artificiale ("computer vision", CV) fanno prestazioni migliori rispetto ai dermatologi nella classificazione delle lesioni cutanee, sebbene non in modo significativo nella pratica clinica. Ad esso si affianca la pratica delle tecniche del deep learning (DL), in particolare quelle open source, riuscendo ad ottenere una percentuale di errore inferiore del 5% per la classificazione delle immagini digitali. Tuttavia, per ottenere un'elevata precisione, gli algoritmi DL devono essere addestrati su un vasto set di dati di immagini classificate.

Una diagnosi clinica accurata del melanoma può essere impegnativa

Tuttavia, l'accuratezza diagnostica del melanoma da parte dell'occhio non aiutato è deludente. La resa del melanoma sulla biopsia di lesioni sospette è solo 1 su 36.

La dermatoscopia facilita la visualizzazione di caratteristiche morfologiche che non sono distinguibili dall'esame ad occhio nudo e consente una diagnosi migliore rispetto all'occhio non aiutato, con un miglioramento nella sensibilità diagnostica del 10-30%.

Però, la dermoscopia può effettivamente ridurre l'accuratezza diagnostica nelle mani dei dermatologi inesperti, poiché questo metodo richiede molta esperienza per differenziare le lesioni cutanee. Gli esperti ottengono una sensibilità del 90% e 59 % di specificità, mentre questa

performance peggiora significativamente con l'inesperienza e scende al 62% -63% per i medici generici.

Gli algoritmi di visione artificiale (CV) attualmente disponibili funzionano solo leggermente meglio nella pratica senza alcun miglioramento significativo nella diagnosi rispetto a un dermatologo. L'istopatologia rimane il miglior standard per un'accurata diagnosi del melanoma sebbene il tasso di letture discordanti tra i patologi possa essere elevato: quando 11 patologi esperti hanno esaminato 37 lesioni melanocitiche "classiche"; solo il 30% dei casi ha raggiunto un accordo totale; altri studi riportano un tasso di discordanza fino al 50% tra i patologi. Pertanto l'accuratezza diagnostica del melanoma rimane problematica indipendentemente dal metodo utilizzato per la diagnosi.

Il Deep Learning

Il concetto Deep learning (DL) è nato dal desiderio di applicare l'attività neurale del cervello ad un computer. Un neurone artificiale è programmato per funzionare:

- i) Ad una specifica soglia d'attivazione (o bias)
- ii) Per input pesati in modo differenziale

Tuttavia, per l'interpretazione di segnali e schemi complessi c'è il bisogno di modelli sofisticati di neuroni computazionali che possano incatenarsi tra loro per generare segnali molto simile al sistema visivo nel cervello, per interpretare cognitivamente i segnali luminosi (retina), al fine di classificare gli oggetti. Oggi, i modelli di rete neurale più usati sono composti da numerosi neuroni artificiali multistrato che sono composti da diversi bias ed input differenziali che richiedono un set con elevato numero di dati per la stima. Un volta che queste reti sono state addestrate su un set di dati classificati sufficientemente grande e di alta qualità, generalmente superano gli altri metodi di apprendimento automatico. Il processo intensivo dal punto di vista computazionale di stima accurata dei loro parametri attraverso la formazione su enormi set di dati costituisce il paradigma del DL. Inoltre, la crescita esponenziale del potere computazionale e la recente emergenza del calcolo della GPU, insieme all'abbondanza di grandi set di dati su cui allenarsi rende più l'applicazione DL pratico ora che mai.

Capitolo 3

3. Algoritmo

Tenendo in considerazioni le diverse possibilità di caratterizzare l'algoritmo per la classificazione delle immagini di lesioni cutanee, si è scelto di basarsi in particolare su quello ABCD.

Però a causa delle difficoltà riscontrate per l'estrazione di informazioni relative al punto D, che consiste nel determinare le dimensioni della lesione in mm oppure il numero di strutture pigmentarie. Ciò ha implicato l'assenza del calcolo del TDS (precedentemente descritto nel paragrafo dedicato), ma si sono ugualmente estratte le caratteristiche relative al punto A, B e C della regola dell'ABCD.

Nell'elaborazione dell'algoritmo si è usufruito anche dell'ambiente di programmazione di applicazioni presente in MATLAB, il quale verrà descritto in un paragrafo dedicato.

L'algoritmo scritto è basato principalmente:

- Image Pre-Processing
- Image Segmentation
- Feature Extraction
- Classifier (Rete Neurale)

3.1 Image Pre-Processing

L'immagine a colori *RGB* viene inserita nell'algoritmo tramite la funzione di base presente in MATLAB, una volta caricata, con *'imread'* (1), viene convertita in scala di grigio, con *'rgb2gray'* (2) e migliorati i contrasti. Si aumenta il contrasto per poter rimuovere la peluria dall'immagine e rendere l'“Image Segmentation” e la “Feature Extraction” ottimali.

Per cui si applica un doppio aumento di contrasto dell'immagine in scala di grigio, prima con *'imadjust'* (3), funzione che satura 1% superiore ed inferiore del valore dei pixel; poi si applica *'adapthisteq'* (4), la quale migliora il contrasto dell'immagine in scala di grigio trasformando i valori usando l'equalizzazione dell'istogramma adattivo con contrasto limitato.

Ottenuta la migioria dei contrasti si applica un filtraggio mediano dell'immagine in due dimensioni, con *'medfilt2'* (5), cioè ogni pixel in output contiene il valore mediano dei pixel costituenti una matrice 3x3 attorno al pixel corrispondente nell'immagine in input della funzione.

Fatto ciò si può procedere ulteriormente nella rimozione della peluria secondo il *'thresholding'*, basandosi su funzioni di *'Morphological structuring element'*.

Si è usata la funzione *'strel'* (6) che rappresenta un elemento strutturante morfologico 2D, che è una parte essenziale delle operazioni di dilatazione ed erosione morfologica; in questo caso si è desiderata un'operazione di erosione dell'immagine per ottenere i peli ben differenziati rispetto lo sfondo.

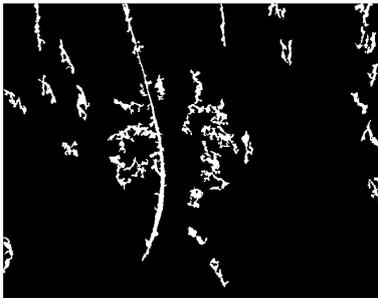
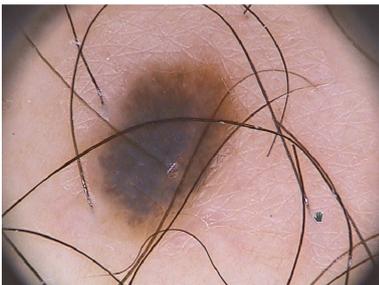
Giunti a questo punto si tiene in considerazione che la peluria può essere di colorazione scura o chiara per cui si deve poter applicare l'output di *'strel'* per entrambi i casi di colore, per far ciò si sfruttano le funzioni *'imbothat'* (7), per elementi da rimuovere di tonalità scure, e *'imtophat'* (8), per elementi da rimuovere di tonalità chiare.

All'output delle funzioni *'imbothat'* e *'imtophat'* (nella seconda colonna della tabella), che sono dei filtri morfologici applicati all'immagine in scala di grigi in uscita dal *'medfilt2'*, si applica un ulteriore miglioramento di contrasto con *'imadjust'*.

A questo punto si deve immettere una soglia per ottenere l'immagine in bianco e nero degli elementi da voler rimuovere dall'immagine, la scelta del valore per il quale l'immagine diverrà costituita da elementi solo di 1 e 0 è calcolato tramite la funzione *'adaptthresh'* (9), che computa localmente una soglia adattiva per immagini in scala di grigio 2D.

Allora si applica la soglia all'immagine in uscita della funzione *'imbothat'* e *'imtophat'* con contrasto migliorato tramite la funzione *'imbinarize'* (10) che scrive l'immagine binaria sfruttando il valore calcolato dalla funzione *'adaptthresh'*, infine si applica la funzione *'bwareafilt'* (11), che

estrae tutti i componenti connessi presenti nell'immagine binaria. In questo caso si è deciso di prendere i 30 elementi più grandi (si evidenzia che la soglia è calcolata sia per l'immagine con i peli scuri che per i peli chiari così come l'immagine binaria e l'acquisizione dei 30 elementi connessi più grandi).

Immagini esempio	Rilevamento elementi da rimuovere
<p>n° 1</p>  <p><small>Gr. Cir. 2015;82:167-11</small></p>	<p>n° 1</p>  <p><small>Gr. Cir. 2015;82:167-11</small></p>
<p>n° 2</p> 	<p>n° 2</p> 
<p>n° 3</p> 	<p>n° 3</p> 

- Nella tabella si può notare come vengano rilevati gli elementi da rimuovere, le immagini n° 1 e n° 3 avendo elementi di tonalità scura è usata la funzione *'imbothat'*; invece per l'immagine n° 2 avente un elemento chiaro si è usata la funzione *'imtophat'*.

» Codice :

- Per ottenere gli elementi da rimuovere:

```

function rimozione_peli_scuri(app,im)
    B = adapthisteq(imadjust(rgb2gray(im)));
    im_gray_Better = medfilt2(B,[3,3]);
    se = strel('diamond',10);
    Q_bot =imadjust(imbothat(im_gray_Better,se));
    soglia_bot=adaptthresh(Q_bot0.2,'ForegroundPolarity','bright');
    app.L = bwareafilt(imbinarize(Q_bot,soglia_bot),30);
    immagine_colorata(app,app.L)
end

```

```

function rimozione_peli_chiari(app,im)
    B = adapthisteq(imadjust(rgb2gray(im)));
    im_gray_Better = medfilt2(B,[3,3]);
    se = strel('diamond',10);
    Q_top =imadjust(imtophat(im_gray_Better,se));
    soglia_top=adaptthresh(Q_top0.2,'ForegroundPolarity','bright');
    app.L = bwareafilt(imbinarize(Q_top,soglia_top),30);
    immagine_colorata(app,app.L)
end

```

~(il codice presente è dell'ambiente di programmazione di applicazioni di MATLAB)

Ottenute le immagini binarie si devono applicare all'immagine in RGB data in input all'algoritmo, per far ciò si deve passare l'immagine da un formato 'uint8' a 'double', tramite la funzione 'double' (12) così facendo si può moltiplicare l'immagine in RGB per l'immagine binaria negata calcolata costituente dagli elementi da rimuovere (cioè dai peli chiari o scuri) per ogni canale di colore ottenendo un'immagine in RGB con la presenza di parti nere al posto degli elementi rimossi; ecco per cui si esegue un ulteriore processo in cui i pixel completamente neri nei tre i canali di colore possano essere rimpiazzati con i colori dei pixel adiacenti.

» Codice :

```

function immagine_colorata(app,L)
    originale = double(app.im);
    [rows,columns,colour] = size(app.im);
    for p = 1 : colour
        RGB_Prova(:,:,p) = originale(:,:,p) .* ~L;
    end
    RGB_Prova = uint8(RGB_Prova);
    . . . (continua successivamente)

```

Ottenendo finalmente l'immagine con la rimozione degli elementi appartenenti all'immagini binarie calcolate, sostituiti con il colore dei pixel adiacenti.

Immagini con l'eliminazione degli elementi	Immagini con la ricostruzione dei pixel eliminati
<p>n° 1</p> 	<p>n° 1</p> 
<p>n° 2</p> 	<p>n° 2</p> 
<p>n° 3</p> 	<p>n° 3</p> 

Nelle figure presenti nella tabella si può evincere che, non sempre, la ricostruzione è perfetta per tutte le immagini.

» Codice :

```

. . . (continuo della funzione immagine_colorata(app,L))
N=1; M=N;
for idx = 1: rows
    for jdy = 1: columns

```

```

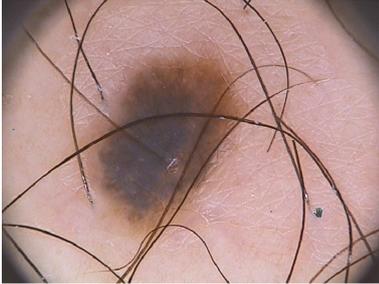
if RGB_Prova(idx,jdy) ~= 0
    N = idx; M = jdy;
    ER(idx, jdy, :) = app.im(idx, jdy, :);
else
    ER(idx, jdy, :) = app.im(N ,M, :);
end
end
end
end (della function immagine_colorata(app,L))

```

~(il codice presente è dell'ambiente di programmazione di applicazioni di MATLAB)

Si noti l'efficienza della rimozione degli elementi delle foto inserite nell'algoritmo con quelle in uscita dal 'Pre-processing'.

Immagini in ingresso all'algoritmo	Immagine dopo il 'Pre-processing'
<p>n° 1</p>  <p><small>Ch Cr, 2015,83:107-11</small></p>	<p>n° 1</p>  <p><small>Ch Cr, 2015,83:107-11</small></p>
<p>n° 2</p> 	<p>n° 2</p> 

Immagini in ingresso all'algoritmo	Immagine dopo il 'Pre-processing'
<p>n° 3</p> 	<p>n° 3</p> 

3.2 Image Segmentation

Per ottenere una ottimale segmentazione della lesione cutanea c'è bisogno di poter scegliere per ciascuna foto una soglia, il valore può dipendere dall'intensità luminosa dei pixel caratterizzando quindi intervalli numerici per cui è preferibile scegliere un valore rispetto l'altro per segmentare l'immagine. Si è provato a renderlo del tutto automatico ma con scarsi risultati quindi si è optato nel creare un'applicazione usufruendo l'ambiente di programmazione per applicazioni presente in MATLAB; implementando quindi un modo di scelta della soglia 'soggettiva' ,per l'operatore che sta usando l'applicazione, ma relativa a ciascuna immagine (giungendo ad un'immagine binaria che tenga conto solo della lesione cutanea indipendente dall'intensità dei pixel, perché l'operatore interviene nel gestire il valore per cui l'immagine RGB diventa binaria).

L'immagine in uscita dall' "Image pre-processing", in RGB, viene nuovamente convertita in scala di grigio con *'rgb2gray'* in seguito viene migliorata l'immagine con l'incremento dei contrasti per far sì che la scelta della soglia da parte dell'operatore modifichi troppo l'immagine binaria.

Con l'uso dell'ambiente di programmazione per applicazione si è potuto lavorare nella scelta della soglia dipendendo dalla tipologia di miglioramento del contrasto, perché dipendendo dall'immagine da elaborare si può scegliere il contrasto migliore. Le funzioni usate sono *'adapthisteq'* (4), *'histeq'* (14), e *'imadjust'* (3), ciascuna ha una determinata tipologia di miglioramento; in particolare la maggior parte dell'immagine la miglior funzione per migliorare il contrasto è *'imadjust'* (3), ma per foto poco illuminate le funzioni *'histeq'* (14) e *'adapthisteq'* (4) si comportano meglio.

» Codice :

```
function contrasti(app,grayscale)

    app.contrasto_1 = adapthisteq(grayscale);
    app.contrasto_2 = histeq(grayscale);
    app.contrasto_3 = imadjust(grayscale);

end
```

Per definire l'immagine binaria rispetto la soglia decisa dall'operatore si è usato:

```
function E(app,level)

    app.BW1 = app.contrasto_1 < level;
    app.BW2 = app.contrasto_2 < level;
    app.BW3 = app.contrasto_3 < level;

end
```

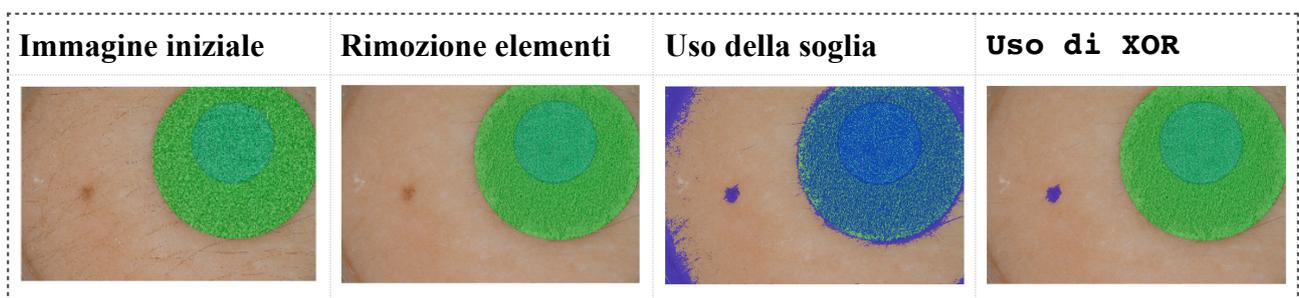
~(il codice presente è dell'ambiente di programmazione di applicazioni di MATLAB)

La funzione *'E(app,level)'* pone in bianco tutto ciò che è minore al valore, chiamato *'level'*, invece i pixel che sono maggiore del valore sono messi in nero. L'intervallo di soglia è compreso tra 0 e 255.

Allora ottenuto l'incremento di contrasto e la soglia si ha l'immagine binaria della lesione con la possibile presenza di ulteriori elementi da rimuovere, che possono essere le ombre presenti negli angoli dell'immagine, per eliminare queste 'impurità' si utilizza la funzione *'bwareafilt'* (11) con

l'accortezza delle dimensioni degli elementi da rimuovere i quali debbono essere minori rispetto le dimensioni della lesione.

In caso ci siano elementi più grandi rispetto la lesione cutanea si applica la sovrapposizione delle immagini binarie con la funzione logica XOR così facendo si eliminano gli elementi di dimensioni più elevate rispetto la lesione cutanea.



Inoltre prima della funzione *'bwareafilt'* si applica *'imfill'* (22), per eliminare possibili pixel neri all'interno della sezione da segmentare, che è composta da pixel bianchi.

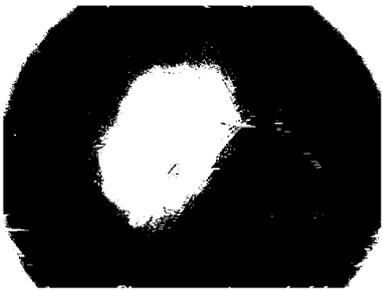
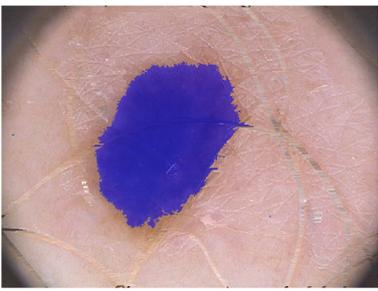
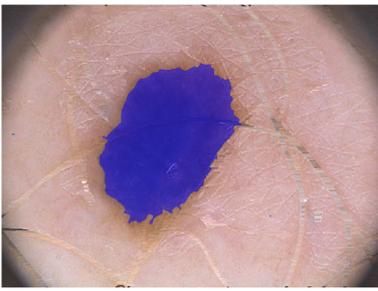
Una importante considerazione è l'aleatorietà dei possibili valori di soglia per una medesima immagine, perché varierebbe da operatore ad operatore; per cui si è pensato di inserire la funzione *'activecontour'* (13) algoritmo iterativo di segmentazione dell'immagine che aumenta la regione, con la finalità di rendere la scelta della soglia meno 'soggettiva'. In un primo momento sembrava essere la soluzione al problema però per poter applicarla si deve inserire il numero di iterazione dell'aumento delle regioni selezionate tramite la soglia ma essendo le immagini prese in considerazione per l'addestramento della rete di dimensioni molto grandi, implicherebbe un numero di iterazioni altrettanto grande per cui si è preferito lasciare l'immagine scelta così com'è senza l'applicazione della funzione *'activecontour'*.

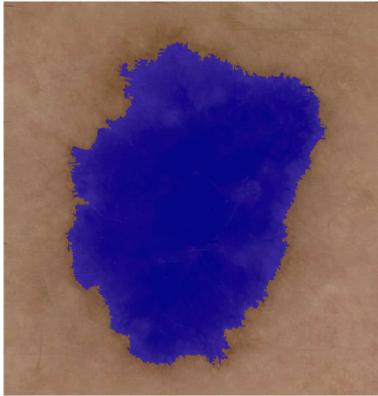
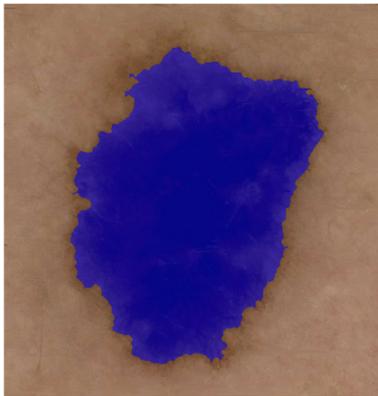
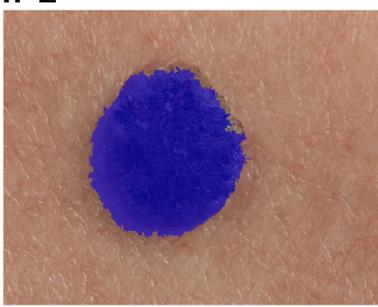
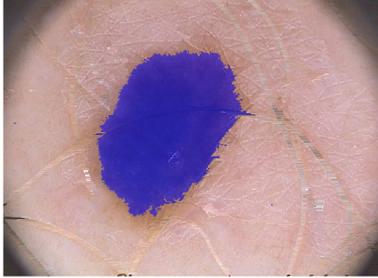
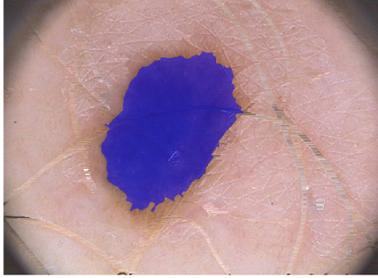
Per calcolare l'immagine binaria con l'uso di *'activecontour'* è stata usato il seguente codice:

» Codice :

```
function contorno(app,mask)
    app.active = activecontour(app.GRAY,mask,200,'edge');
end
```

~(il codice presente è dell'ambiente di programmazione di applicazioni di MATLAB)

Immagine binaria	Uso di <i>'imfill'</i> e <i>'bwareafilt'</i>	Uso di <i>'activecontour'</i>
		

Senza l'uso di <i>'activecontour'</i>	Uso di <i>'activecontour'</i>
<p data-bbox="161 819 240 851">n° 1</p> 	<p data-bbox="807 819 887 851">n° 1</p> 
<p data-bbox="161 1290 240 1321">n° 2</p> 	<p data-bbox="807 1290 887 1321">n° 2</p> 
<p data-bbox="161 1671 240 1702">n° 3</p> 	<p data-bbox="807 1671 887 1702">n° 3</p> 

- nella tabella si è posto in confronto la stessa sovrapposizione dell'immagine binaria acquisita tramite la soglia senza e con l'uso della funzione *'activecontour'*.

Infine si moltiplica la matrice RGB in uscita dal 'pre-processing' per l'immagine binaria ottenendo solo l'immagine della lesione scelta a colori.

» Codice :

```
function Segmented_Image(app,immagine,BW)
    immagine = double(immagine);
    [app.rows, app.columns] = size(BW);
    for i = 1:3
        app.Segment(:, :, i) = immagine(:, :, i) .* BW;
    end
    app.Segment = uint8(app.Segment);
end
```

~(il codice presente è dell'ambiente di programmazione di applicazioni di MATLAB)

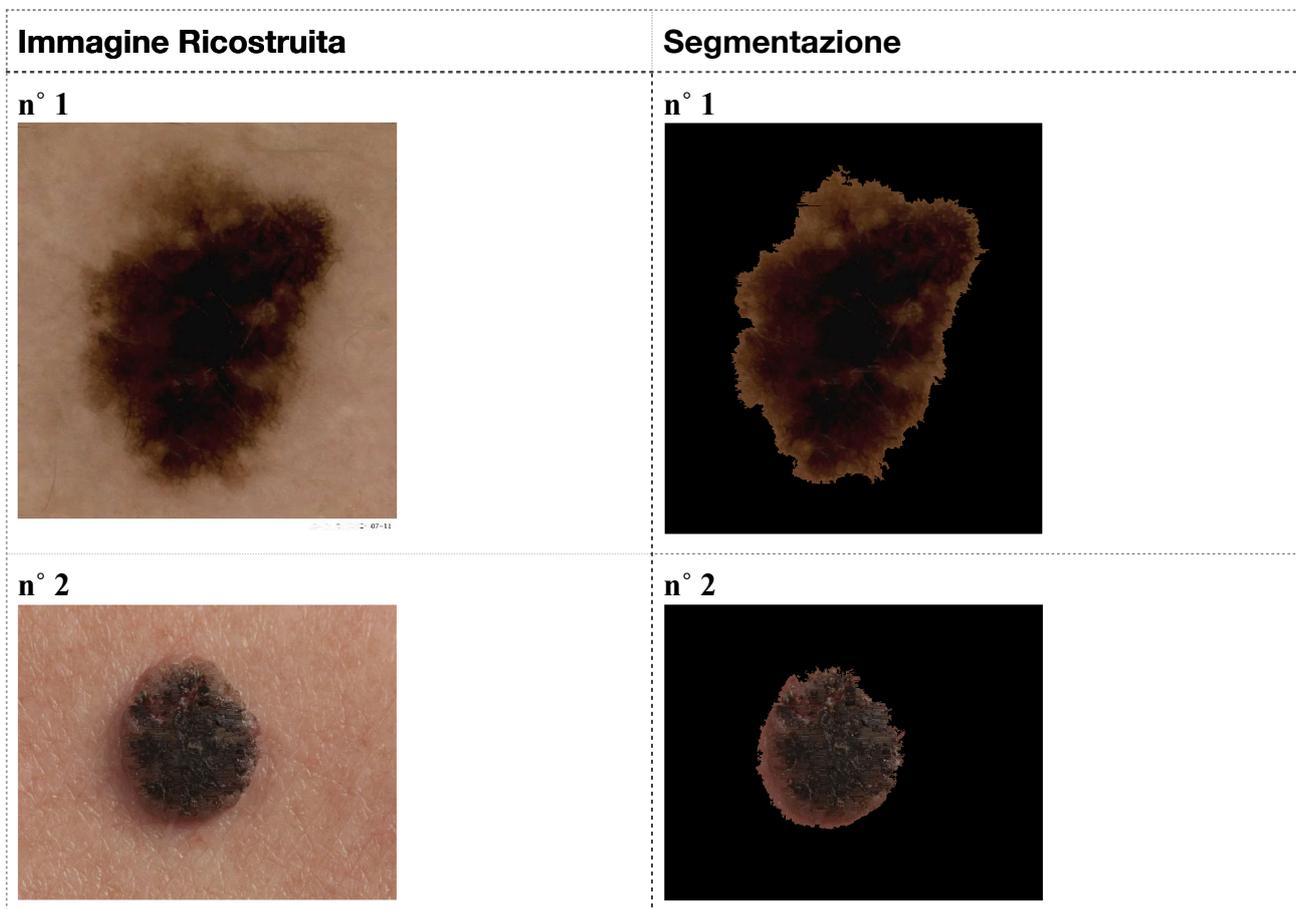


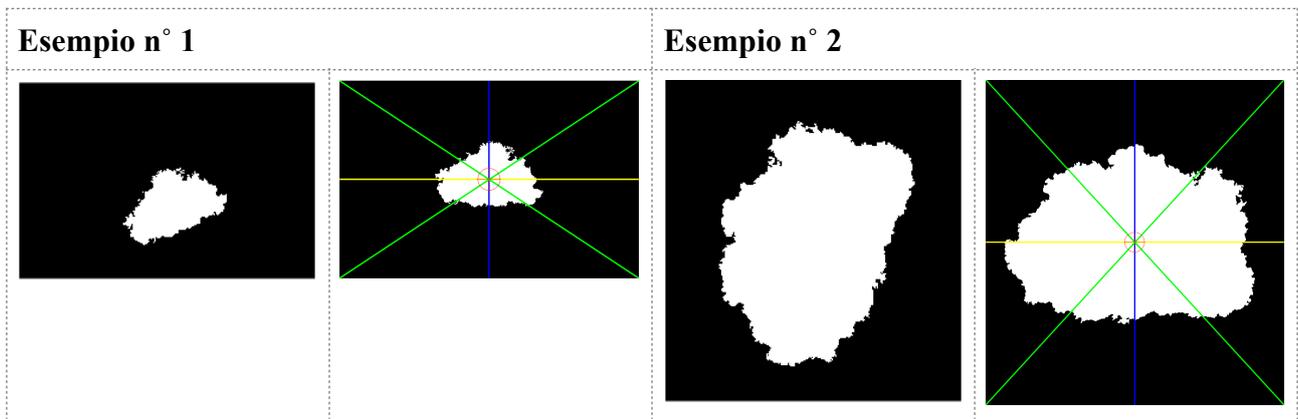
Immagine Ricostruita	Segmentazione
n° 3 	n° 3 

3.3 Feature Extraction

Passaggio fondamentale per l'algoritmo, perché costituisce l'acquisizione dei dati dell'immagini, i quali sono elaborati dalla rete neurale giungendo così ad una probabile diagnosi, quindi si dedicherà una profonda descrizione delle funzioni scritte per estrarre le 9 caratteristiche inserite nell'algoritmo presentato.

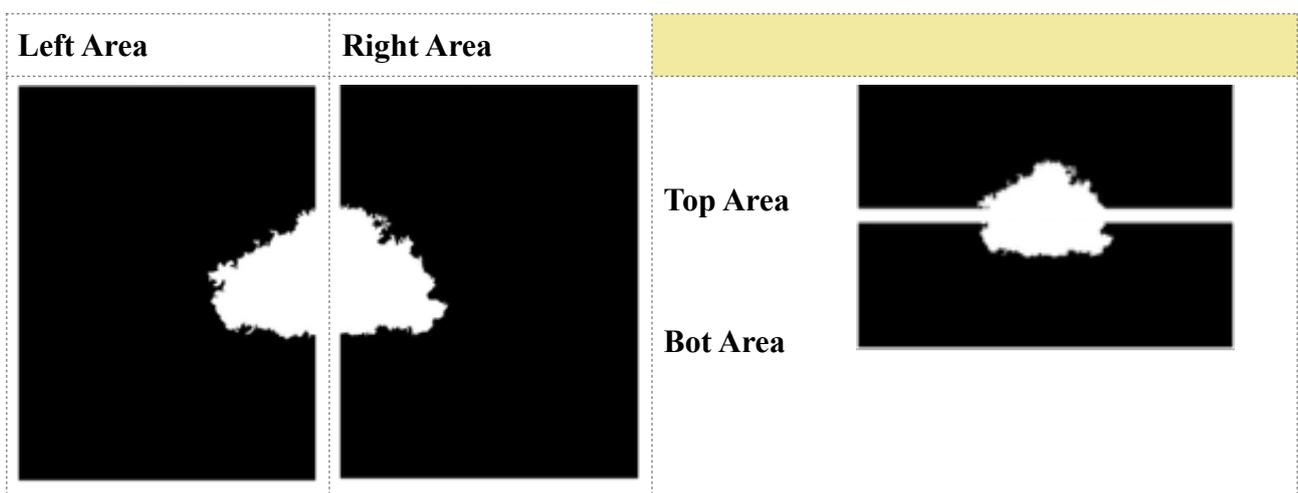
Per l'estrazione delle prime 3 delle 9 caratteristiche si è sfruttata la funzione '*regionprops*' (15) per calcolare 2 proprietà della immagine binaria della segmentazione: i centroidi, che sono le coordinate in pixel del centro dell'immagine binaria, e l'orientazione dei pixel costituenti l'immagine.

Calcolati questi valori si centra la lesione al centro dell'immagine, in modo tale che i centroidi corrispondano alle coordinate del punto centrale dell'immagine. Alcuni esempi:

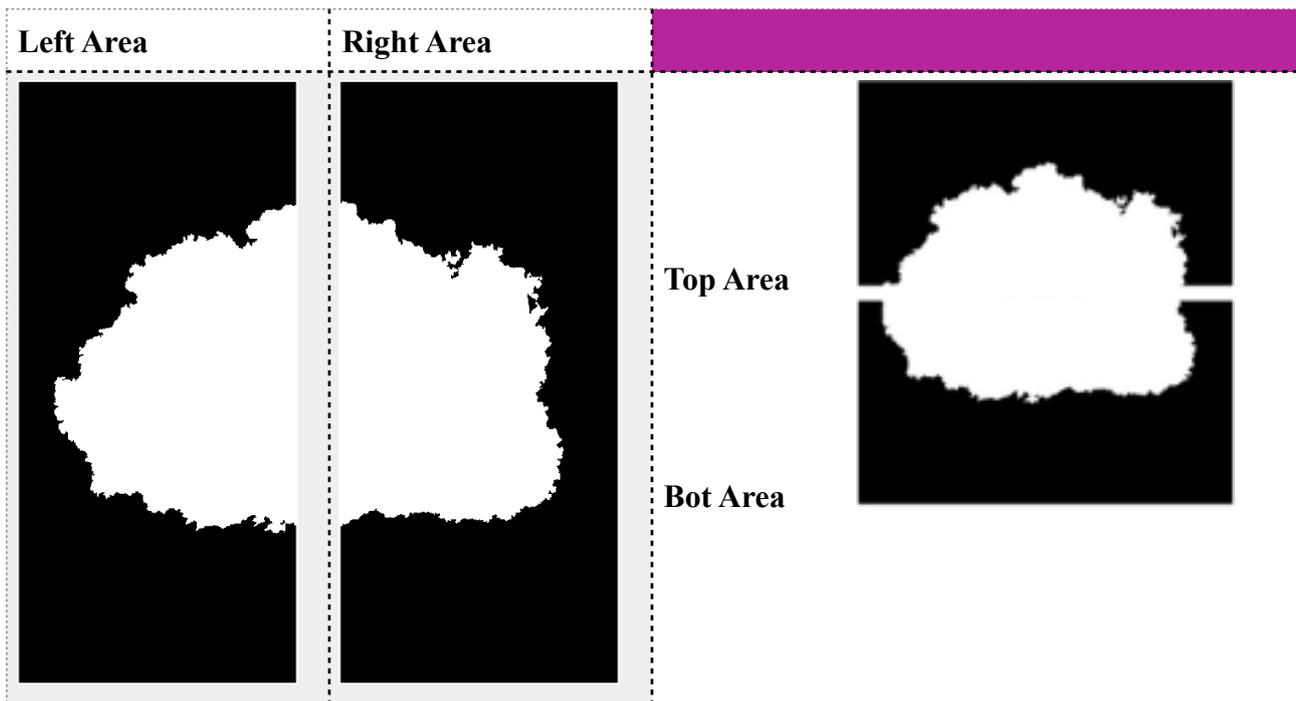


Così facendo si può facilmente suddividere l'immagine binaria della lesione. Essa viene suddivisa sia rispetto l'asse orizzontale che rispetto l'asse verticale, ottenendo 4 suddivisioni ma che corrispondono a 2 divisioni dell'immagine binaria ma rispetto assi ortogonali.

- Esempio n° 1



- Esempio n° 2



Allora si è scelto di sfruttare ‘*non Over-lapping Area*’, cioè si sovrappongono le suddivisioni della lesione rispetto lo stesso asse e si acquisiscono i pixel che non si sovrappongono; applicando la funzione logica XOR.

» Codice :

```
%rispetto l'asse orizzontale
```

```
botarea = flip(rotatedImage(app.rows/2+1:app.rows,:),1);
```

```
toparea = rotatedImage(1:app.rows/2,:);
```

```
%rispetto l'asse verticale
```

```
leftarea = rotatedImage(:,1:app.columns/2);
```

```
rightarea = flip(rotatedImage(:,app.columns/2+1:app.columns),2);
```

```
A_r = bwarea(rotatedImage); %area totale della lesione
```

```
A_1 = xor(botarea,toparea);
```

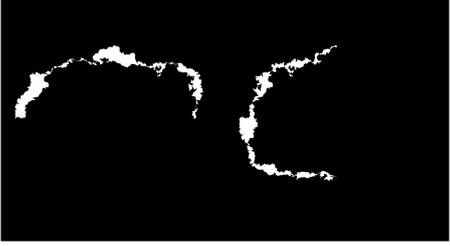
```
A_2 = xor(leftarea,rightarea);
```

```
%si ottengono le matrici dati dalla sovrapposizione
```

```
A_1area = bwarea(A_1);
```

```
A_2area = bwarea(A_2);
```

~(il codice presente è dell’ambiente di programmazione di applicazioni di MATLAB)

Esempio n° 1 ' <i>non Over-lapping Area</i> '	Esempio n° 2 ' <i>non Over-lapping Area</i> '
	

Inoltre si deve tenere in considerazione che le caratteristiche delle foto non devono dipendere dal numero dei pixel che compongono l'immagine; bensì da valori “*adimensionali*”, costituiti da rapporti tra valori dipendenti dalle dimensioni dell'immagine in pixel.

3.3.1 Caratteristica n° 1

- Area_verticale/Area_ToT

Il parametro definito ‘Area_verticale’ è l’area di ‘*non Over-lapping*’ riferita all’asse verticale dell’immagine, cioè la sovrapposizione delle immagini binarie definite precedentemente come ‘Left Area’ e ‘Right Area’. Inoltre il calcolo del numero di pixel che costituiscono l’area verticale dell’immagine è dato dalla funzione ‘*bwarea*’ (16).

3.3.2 Caratteristica n° 2

- Area_orizzontale/Area_ToT

Il parametro definito ‘Area_orizzontale’ è l’area di ‘*non Over-lapping*’ riferita all’asse orizzontale dell’immagine, cioè la sovrapposizione delle immagini binarie definite precedentemente come ‘Top Area’ e ‘Bot Area’. Inoltre il calcolo del numero di pixel che costituiscono l’area orizzontale dell’immagine è dato dalla funzione ‘*bwarea*’ (16).

Inoltre per la caratteristica n° 3 si è preso in considerazione il rapporto della somma delle due ‘*non Over-lapping Area*’ diviso due volte l’area totale:

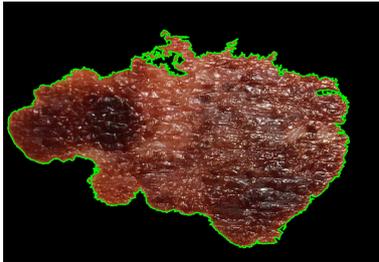
3.3.3 Caratteristica n° 3

- $(\text{Area_verticale} + \text{Area_orizzontale}) / (2 * \text{Area_ToT})$

Riguardo la caratteristica numero 4 si è deciso di prendere in considerazione la variazione del perimetro della lesione (l’irregolarità del bordo), per ottenere ciò si è usufruito della funzione

'*bwboundaries*' (17) la quale calcola il bordo esterno dell'oggetto dell'immagine binaria. Ottenuto il bordo si calcolano: la distanza tra il bordo ed il centroide della lesione e la deviazione standard del vettore che ha come componenti tutte le distanze di ogni coordinata costituente il perimetro.

Uso della funzione 'bwboundaries'



» Codice :

```
boundaries = bwboundaries(BW);
numberOfBoundaries = size(boundaries, 1);
for k = 1 : numberOfBoundaries
    thisBoundary = boundaries{k};
    a = thisBoundary(:,2)-xcentroide;
    b = thisBoundary(:,1)-ycentroide;
    distances = sqrt((a).^2 + (b).^2);
    SDdistances(k) = std(distances);
end
```

~(il codice presente è dell'ambiente di programmazione di applicazioni di MATLAB)

Allora si divide in 8 parti il perimetro, usando la funzione '*round*' (18) per avere intervalli costituiti da valori interi; ma ciò comporta, nel caso che la dimensione del perimetro non sia divisibile per 8, ad avere un bordo più grande o più piccolo rispetto a quello originale quindi all'ottavo intervallo si toglie o si aggiunge un numero pari all'errore dell'approssimazione di '*round*'.

» Codice :

```
[lunghezza,~] = size(distances);
divisione_distances = round(lunghezza/8);
eccesso = divisione_distances * 8 - lunghezza;
valore_soglia = (SDdistances * 3/8); %% ----- Valore Soglia

% si definiscono gli 8 intervalli:
primo = distances( 1 : divisione_distances);
```

```

secondo = distances( divisione_distances+1 : 2*divisione_distances);
terzo = distances(2* divisione_distances+1 : 3*divisione_distances);
quarto = distances(3*divisione_distances+1 : 4*divisione_distances);
quinto = distances(4*divisione_distances+1 : 5*divisione_distances);
sesto = distances(5*divisione_distances+1 : 6*divisione_distances);
settimo = distances(6*divisione_distances+1 : 7*divisione_distances);
ottavo = distances(7*divisione_distances+1 : 8*divisione_distances-eccesso);

```

~(il codice presente è dell'ambiente di programmazione di applicazioni di MATLAB)

Infine si calcola la deviazione standard di ogni intervallo di suddivisione del bordo e si confronta con un valore. Se la deviazione standard, calcolato con 'std' (19), è minore del valore soglia il bordo non è molto irregolare, invece nel caso contrario significa che il bordo ha grandi irregolarità perché ha una variazione elevata della distanza causando una elevata deviazione standard. Ogni valore di ciascun intervallo, che possono essere 0 o 1 (dipendendo se la deviazione standard è maggiore o no al valore soglia), vengono sommati costituendo la feature numero 4.

» Codice :

```

if std(INTERVALLO_IESIMO) < valore_soglia
    punto(I) = 0;
else
    punto(I) = 1;
end

```

~(il codice presente è dell'ambiente di programmazione di applicazioni di MATLAB)

Si calcola la deviazione standard per ognuno degli 8 intervalli definiti nel codice precedente e si confronta con il valore definito come 'valore_soglia' e si confronta per ogni intervallo se il valore della deviazione standard del bordo i-esimo è maggiore o minore alla soglia.

3.3.4 Caratteristica n° 4

- sum(punto)

Viene usata la funzione 'sum' (20), per sommare ciascuna componente del vettore punto, (che ha come componenti i valori ottenuti dal confronto delle irregolarità del bordo precedentemente descritto).

Il vettore punto ha dimensioni [8,1], in cui ogni riga può avere un valore pari a 0 o 1, dipendendo se il valore della deviazione standard presente nell'intervallo i-esimo, (suddivisione del bordo), ha un valore minore o maggiore rispetto il valore soglia.

Le successive caratteristiche, invece, si concentrano nel calcolare il numero di pixel di determinate colorazioni di colore le quali sono: bianco, nero, rosso, marrone chiaro, marrone scuro e blue-gray. Il conteggio viene eseguito separando l'immagine segmentata RGB in 3 immagini in cui siano presenti in ciascuna solo 1 canale di colore, per cui solo rosso, verde e blu; in seguito ogni pixel di ciascun canale viene confrontato con soglie che costituiscono i 6 colori precedentemente elencati. Certamente si è deciso di applicare degli intervalli sufficientemente larghi per poter ottenere più sfumature dello stesso colore (che possono cambiare a causa della luminosità presente e dalla qualità dell'immagine stessa).

» Codice :

```
%Matrici Red, Green e Blue
Red = RGB; Red(:,:, [2 3]) = 0;
Green = RGB; Green(:,:, [1 3]) = 0;
Blue = RGB; Blue(:,:, [1 2]) = 0;
    for idx = 1 : rows %si genera una matrice [M,N,1]
        for jdy = 1 : columns
            if Red(idx,jdy,1) ~= 0
                RR(idx, jdy) = Red(idx,jdy,1);
            else
                RR(idx,jdy) = 0;
            end
        end
    end
end
```

~(il codice presente è dell'ambiente di programmazione di applicazioni di MATLAB)

Il procedimento definito dal codice è eseguito per i tre canali di colore, ottenendo 3 matrici binarie. Inoltre per calcolare il numero di pixel che definiscono un colore dei 6 elencati:

» Codice :

```
for idx = 1 : rows
    for jdy = 1 : columns
        if(RR(idx,jdy)> !! && GG(idx,jdy)> !! && BB(idx,jdy)> !!)
            if (RR(idx,jdy)< !! && GG(idx,jdy)< !! && BB(idx,jdy)< !!)
                colore_iesimo = colore_iesimo+ 1;
            end
        end
    end
end
end
```

~(il codice presente è dell'ambiente di programmazione di applicazioni di MATLAB)

In cui al posto dei '!!' Sono definiti dei valori che rappresentano l'intervallo del "colore_iesimo" per poter essere conteggiato.

3.3.5 Caratteristica n° 5

- Percentage(i)

'Percentage' è il nome della variabile utilizzata nel algoritmo, è un vettore avente come componenti il rapporto in percentuale tra il numero di pixel corrispondente all'i-esimo colore rispetto il numero totale dei pixel colorati della lesione.

» Codice :

```
Percentage = [ (Num_Black/NumeroTotale)*100;  
              (Num_White/NumeroTotale)*100;  
              (Num_Red/NumeroTotale)*100;  
              (Num_LightBrown/NumeroTotale)*100;  
              (Num_DarkBrown/NumeroTotale)*100;  
              (Num_BlueGray/NumeroTotale)*100      ];
```

~(il codice presente è dell'ambiente di programmazione di applicazioni di MATLAB)

Inoltre, nel caso in cui un colore sia presente più del 5% viene conteggiato ed inserito in un'ulteriore vettore (chiamato nell'algoritmo 'Score'), ove è composto da 1 o 0 (1 se il colore i-esimo è presente più del 5% dell'area della lesione 0 invece se è meno del 5%). Si sommano tutte le componenti caratterizzando l'11-esima caratteristica.

» Codice :

```
for i = 1 : length(Percentage)  
    if round(Percentage(i,1)) > 5  
        Score(i,1) = 1;  
    else  
        Score(i,1) = 0;  
    end  
end
```

~(il codice presente è dell'ambiente di programmazione di applicazioni di MATLAB)

3.3.6 Caratteristica n° 6

- `sum(Score)`

Infine le ultime 3 delle 9 caratteristiche estratte sono costituite dalla deviazione standard di ogni canale di colore dell'immagine RGB della lesione segmentata.

Cioè costituisce quanto è il valore della deviazione standard totale per ogni canale di colore, (Rosso, Verde e Blu).

Per calcolare queste caratteristiche viene usata la funzione '`std2`' (21), calcola la deviazione standard degli elementi di una matrice.

3.3.7 Caratteristiche dalla n° 7 alla n° 9

- `std2(RGBDouble(: , : , i))`

'RGBDouble' è la rappresentazione in '`double`' dell'immagine in RGB segmentata in '`uint8`', e si calcola la deviazione standard di ogni canale di colore i-esimo (rosso, verde, blu) ottenendo, quindi, la variazione di ogni canale nell'immagine.

» Codice :

```
RGBDouble = double(RGB_Segmentata);  
i = 1;  
for q = 7 : 1 : 9  
    o(i,1) = std2(RGBDouble(:, :, i));  
    i = i + 1;  
end
```

~(il codice presente è dell'ambiente di programmazione di applicazioni di MATLAB)

Le tonalità di illuminazione della lesione possono incidere ad eventuali errori di intensità luminosità dei pixel costituenti la lesione cutanea, influenzando , quindi, sia le caratteristiche del paragrafo '3.3.5' e '3.3.7'.

L'influenza della tonalità di luminosità è maggiore nelle caratteristiche presenti nel paragrafo '3.3.5' perché essendo una caratteristica che tiene in considerazione la presenza di 6 distinti colori, (descritti nel paragrafo '3.3.5'), può far variare i valori dei canali rosso, verde, blu che definiscono un determinato colore dei 6 applicando un errore sulla effettiva presenza dei colori.

Invece per quanto riguarda l'influenza della tonalità di luminosità nelle caratteristiche del paragrafo '3.3.7' è minore visto che sono il risultato del calcolo della deviazione standard di ogni canale di colore (rosso, verde, blu) nel loro complesso e non per un determinato colore. Cioè l'influenza della tonalità di colore viene eliminata perché essendo applicata in tutta l'immagine la deviazione standard non ne risente in grande valore.

Conclusione

L'algoritmo trattato in questa Tesi è basato sul '*Thresholding*', cioè si applica un valore di soglia e si costituisce l'immagine binaria, ciò comporta ad una difficoltà di generazione di un'ottima segmentazione automatica.

Cioè che si possa calcolare un'immagine binaria che presenti soltanto la lesione cutanea (avendo eroso l'immagine dagli elementi da rimuovere nel 'pre-processing').

Per cui si pensa di elaborare un miglior procedimento di analisi dell'immagini per evitare possibili errori in immagini con differenti luminosità o con variazione minime di colore (ad esempio: pelle chiara con lesione cutanea di colore rosa).

Si è anticipato durante la descrizione dell'algoritmo l'uso dell'ambiente di programmazione di applicazioni di MATLAB, da cui sono estratti i codici presenti in questo elaborato, ciò ha dato la possibilità di poter eseguire un algoritmo interattivo con l'operatore per evitare errori provenienti da una segmentazione automatica, precedentemente discusso.

In vista di sperimentazioni future si desidera applicare ad una rete neurale i dati estratti tramite i codici presenti nel paragrafo '3.3 Feature Extraction' per testare la qualità dell'estrazione delle caratteristiche e l'impatto che esse hanno nell'elaborazione da parte della rete neurale di definire se la lesione cutanea in studio possa essere un semplice nevo oppure un probabile caso di melanoma.

Infine si sta pensando ad un modo di poter definire la presenza di strutture pigmentarie all'interno della lesione, procedura che al momento non si è riuscita ad implementare per la difficoltà e complessità di realizzazione del codice.

Bibliografia

- <http://www.airc.it/cancro/tumori/melanoma-cutaneo>
- <http://www.humanitas.it/malattie/melanoma>
- <http://www.stanganelliignazio.it/>
- http://www.salute.gov.it/portale/salute/p1_5.jsp?lingua=italiano&id=48&area=Tumori
- Dermoscopedia “ABCD rule” (Editor: Wilhelm Stolz; Author(s): Michael Kunz, Wilhelm Stolz)
- “Enhanced classification of malignant melanoma lesions via the integration of physiological features from dermatological photographs” (Authors: Shahid Haider, Daniel Cho, Robert Amelard, Alexander Wong, David A. Clausi)
- “Precision Diagnosis of melanoma and other skin lesions from digital images” (Authors: Abhishek Bhattacharya, Albert Young, Andrew Wong, Simone Stalling; Maria Wei, Dexter Hadley)

1. <https://it.mathworks.com/help/matlab/ref/imread.html>
2. <https://it.mathworks.com/help/matlab/ref/rgb2gray.html>
3. <https://it.mathworks.com/help/images/ref/imadjust.html>
4. <https://it.mathworks.com/help/images/ref/adapthisteq.html>
5. <https://it.mathworks.com/help/images/ref/medfilt2.html>
6. <https://it.mathworks.com/help/images/ref/strel.html>
7. <https://it.mathworks.com/help/images/ref/imbothat.html>
8. <https://it.mathworks.com/help/images/ref/imtophat.html>
9. <https://it.mathworks.com/help/images/ref/adaptthresh.html>
10. <https://it.mathworks.com/help/images/ref/imbinarize.html>
11. <https://it.mathworks.com/help/images/ref/bwareafilt.html>
12. <https://it.mathworks.com/help/matlab/ref/double.html>
13. <https://it.mathworks.com/help/images/ref/activecontour.html>
14. <https://it.mathworks.com/help/images/ref/histeq.html>
15. <https://it.mathworks.com/help/images/ref/regionprops.html>
16. <https://it.mathworks.com/help/images/ref/bwarea.html>
17. <https://it.mathworks.com/help/images/ref/bwboundaries.html>
18. <https://it.mathworks.com/help/matlab/ref/round.html>
19. <https://it.mathworks.com/help/matlab/ref/std.html>
20. <https://it.mathworks.com/help/matlab/ref/sum.html>
21. <https://it.mathworks.com/help/images/ref/std2.html>
22. <https://it.mathworks.com/help/images/ref/imfill.html>