

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Corso di Laurea Magistrale in Matematica

Recupero di immagini dermatologiche mediante persistenza e formule di Viète

Tesi di Laurea in Topologia Algebrica

Relatore:
Prof.
Massimo Ferri

Presentata da:
Alessia Angeli

Correlatore:
Dott.
Ivan Tomba

Seconda Sessione
Anno Accademico 2016/2017

*A tutti coloro che mi sono sempre stati vicini in passato,
che mi sono vicini oggi,
e che mi saranno sempre vicini in futuro...*

Indice

Introduzione	1
1 Il melanoma cutaneo	3
1.1 Introduzione	3
1.2 Principali fattori di rischio	4
1.3 Sintomi	5
1.4 Diagnosi	5
1.5 Epidemiologia	6
1.6 Prevenzione	7
1.7 Supporto computerizzato alla diagnosi	7
1.8 Esempi di immagini di lesioni cutanee	9
2 La matematica alla base dell’algoritmo	11
2.1 L’omologia persistente	11
2.2 Le formule di Viète	20
3 L’algoritmo attuale ed il nuovo algoritmo	23
3.1 L’algoritmo attuale	23
3.1.1 Introduzione al funzionamento della macchina	23
3.1.2 I passaggi principali dell’algoritmo	25
3.1.3 Il calcolo delle distanze	26
3.2 Il nuovo algoritmo	27
3.2.1 L’obiettivo	27
3.2.2 Introduzione al nuovo algoritmo	27

3.2.3	Il nuovo algoritmo in dettaglio	30
4	Prove effettuate e risultati ottenuti	37
4.1	Primo database	40
4.2	Nuovo database PH2	46
4.3	Ulteriore prova con il primo database	61
4.4	Osservazione relativa ai costi computazionali degli algoritmi	63
4.5	Risultati globali	65
4.6	Ulteriore confronto tra risultati globali	72
	Conclusioni e lavoro futuro	74
	Conclusioni e lavoro futuro	74
A	Codice Python	79
A.1	PolyParte1	80
A.2	PolyParte2	82
	Bibliografia	87

Introduzione

Il melanoma cutaneo è un tumore maligno della pelle. Considerato fino a pochi anni fa un tipo di tumore raro, oggi è in crescita costante in tutto il mondo. Dagli studi effettuati negli ultimi decenni emerge, oltre al fatto che l'incidenza del melanoma cutaneo è in aumento, il fatto che la prevenzione ed una diagnosi precoce risultano essere fondamentali per fare in modo che questo non diventi pericoloso. Purtroppo però la diagnosi clinica di melanoma è generalmente difficoltosa ed è condizionata direttamente dall'esperienza del medico. Per questi motivi un'analisi computerizzata può rivelarsi un valido strumento di supporto in questo ambito.

Da qualche anno è in fase di realizzazione e sviluppo un progetto per la costruzione di una macchina per il supporto medico nell'individuazione di lesioni melanocitiche. Il progetto è a cura della ditta CA-MI S.r.l., con la supervisione dell'Università di Bologna, dell'Università di Parma e dell'IRST di Meldola. L'obiettivo è quello di creare una macchina contenente un database di immagini di nei e melanomi già diagnosticati che, una volta acquisita una nuova immagine di una lesione cutanea, restituisca le 10 immagini più "vicine" a quella in esame tra tutte quelle contenute nel database. Un aspetto fondamentale da sottolineare è che, con tale macchina, non si vuole assolutamente fornire una diagnosi che possa sostituire quella del dermatologo, ma si vuole cercare di agevolare ed aiutare il lavoro di quest'ultimo. Il principio che sta alla base del funzionamento di tale macchina sfrutta principalmente la teoria dell'omologia persistente.

In questo progetto di tesi si è modificato, in parte, l'algoritmo che attualmente è presente all'interno della macchina. In particolare è stata presa in esame la parte di algoritmo in cui vengono calcolate le distanze tra le diverse immagini del database con l'obiettivo di ridurre il tempo di esecuzione del programma al fine di riuscire a lavorare con

database di maggiori dimensioni e potere quindi confrontare tra loro un numero maggiore di immagini. Con il nuovo algoritmo si cerca inoltre di tralasciare meno informazioni possibili dalle immagini che vengono acquisite e poi analizzate dalla macchina. Come verrà spiegato in seguito nel dettaglio, attualmente queste distanze vengono calcolate utilizzando la distanza di bottleneck tra diagrammi di persistenza relativi rispettivamente alle immagini che si stanno considerando, data una certa funzione filtrante. Nel nuovo algoritmo, invece, verranno costruiti dei polinomi complessi relativi a tali diagrammi di persistenza e le distanze saranno calcolate come distanze tra i coefficienti di tali polinomi. In questo modo si spera di riuscire ad ottenere dei buoni risultati e di dare un maggiore supporto allo specialista.

Nel primo capitolo viene introdotto il melanoma cutaneo riportandone i principali fattori di rischio ed illustrandone i principali sintomi. Successivamente vengono poi descritte le principali caratteristiche legate alla diagnosi di questo tipo di tumore, all'importanza della prevenzione e di una diagnosi precoce, arrivando poi ad introdurre un possibile supporto computerizzato alla diagnosi stessa.

Nel secondo capitolo vengono riportate le principali nozioni teoriche per comprendere al meglio la matematica che è alla base dell'algoritmo che andremo poi a modificare ed utilizzare: in particolare verrà introdotta la teoria dell'omologia persistente e verranno illustrate le formule di Viète, per il calcolo dei coefficienti di un polinomio date le rispettive radici, le quali saranno fondamentali per il nuovo algoritmo.

Nel terzo capitolo viene illustrato il funzionamento generale della macchina e vengono spiegati, in dettaglio, l'algoritmo attuale ed il nuovo algoritmo mettendo in evidenza analogie e differenze.

Nel quarto capitolo vengono riportate una descrizione accurata della fase sperimentale ed un'analisi dei risultati ottenuti dalle prove effettuate. Tutto questo illustrando, in dettaglio, il ragionamento che ci ha portato a preferire un certo codice piuttosto che un altro.

In appendice viene, infine, riportata una versione dei codici utilizzati in questo progetto.

Capitolo 1

Il melanoma cutaneo

In questo capitolo introduciamo alcune definizioni al fine di capire che cosa è ed in che modo si forma il melanoma cutaneo. Vedremo inoltre quali sono i principali fattori di rischio ed i principali sintomi legati ad esso e cercheremo di capire perché sono così importanti la prevenzione ed una diagnosi precoce.

1.1 Introduzione

La *pelle*, o *cute*, è l'organo più esteso del corpo umano ed è un organo dinamico, che cambia costantemente. La pelle consiste di tre strati principali, disposti uno sull'altro: epidermide, derma ed ipoderma.

I *melanociti* sono cellule situate nella parte inferiore dell'epidermide. Tali cellule hanno il compito di produrre un pigmento, chiamato *melanina*, che conferisce la colorazione alla pelle, ai capelli e ad alcune parti dell'occhio. La melanina, inoltre, protegge dagli effetti dannosi dei raggi solari.

In condizioni normali i melanociti possono dare luogo ad agglomerati scuri visibili sulla superficie della pelle e comunemente chiamati *nei* (in letteratura medica *nevi*). Questo accade quando i melanociti si accumulano formando una piccola massa insieme al tessuto circostante. Clinicamente i nei si manifestano con lesioni in chiazza o nodulari a limiti netti e relativamente stabili nel tempo.

Il *melanoma cutaneo* è un tumore maligno che deriva dalla trasformazione tumorale dei melanociti. Questo può avere origine da un neo preesistente oppure insorgere su una parte di pelle integra.

In generale un *tumore* (dal latino tumor, “rigonfiamento”) è una massa di tessuto anormale la cui crescita avviene in modo eccessivo e scoordinato rispetto agli altri tessuti, in seguito alla proliferazione patologica delle cellule di una certa zona del corpo, e che persiste in questo stato dopo la cessazione degli stimoli che hanno indotto tale processo. Un tumore viene anche chiamato *neoplasia* (dal greco νέος *nèos*, “nuovo”, e πλάσις *plásis*, “formazione”).

Una comune classificazione in base all’aggressività del tumore stesso è quella tra *tumori benigni* e *tumori maligni*. La principale differenza biologica tra essi è nel tipo di riproduzione cellulare. In entrambi i casi la riproduzione non è controllata, però, mentre nel caso di un tumore benigno le cellule proliferate mantengono, almeno in parte, le caratteristiche morfologiche tipiche del tessuto originario e restano limitate all’organo in cui sono nate, nel caso di un tumore maligno le cellule mutano ancora più profondamente invadendo tessuti circostanti ed estendendosi agli organi vicini (fenomeno conosciuto come diffusione delle metastasi).

1.2 Principali fattori di rischio

Le possibili cause del melanoma non sono ancora state del tutto chiarite. Tuttavia molti fattori di rischio del melanoma sono conosciuti. Alcuni di questi vengono trasmessi dai genitori ai figli attraverso i geni (*fattori di rischio endogeni*), altri sono invece di tipo ambientale o correlabili alle attività svolte dalle persone stesse (*fattori di rischio esogeni*). Avere uno o più fattori di rischio non significa che si svilupperà il melanoma e d’altra parte il melanoma può svilupparsi anche in soggetti che non presentano alcun fattore di rischio. Da tenere in considerazione che un qualsiasi traumatismo su un neo benigno non ne cambia la natura e non aumenta quindi il rischio di trasformazione maligna.

I principali fattori di rischio conosciuti per lo sviluppo del melanoma cutaneo sono i seguenti:

- radiazioni ultraviolette (UV): possono danneggiare il DNA delle cellule della pelle ed innescare la trasformazione tumorale;
- nei atipici o nei presenti in numero significativo;
- carnagione chiara, capelli e occhi chiari, presenza di lentiggini;
- familiarità (a presentare questa caratteristica sono circa il 10% dei casi di melanoma cutaneo);
- stato di immunosoppressione (alcune patologie e alcuni farmaci indeboliscono il sistema immunitario).

1.3 Sintomi

Il sintomo principale del melanoma cutaneo è il cambiamento nell'aspetto di un neo o la comparsa di uno nuovo. Le caratteristiche di un neo che possono indicare l'insorgenza di un melanoma possono essere riassunte grazie alla *regola dell'ABCDE*:

- *A* *simmetria* nella forma (generalmente un neo è circolare o comunque tondeggiante, un melanoma è più irregolare);
- *B* *ordi* irregolari e indistinti;
- *C* *olore* variabile e disomogeneo o nero molto intenso;
- *D* *imensioni* in aumento, sia in larghezza sia in spessore (solitamente i nei hanno un diametro non superiore ai 6 *mm*);
- *E* *voluzione* progressiva in un tempo piuttosto breve del neo, con cambiamenti cromatici o altri cambiamenti morfologici.

1.4 Diagnosi

La diagnosi precoce del melanoma cutaneo non dipende solo dal medico: un autoesame periodico della propria pelle permette in molti casi di identificare cambiamenti o nei sospetti e di rivolgersi per tempo al dermatologo.

Lo specialista effettua in primo luogo una visita completa al paziente nella quale valuta la sua storia familiare e la presenza, o meno, di segni e sintomi tipici del melanoma cutaneo. La diagnosi di un melanoma risulta agevolata quando sono presenti tutte le caratteristiche descritte nella regola dell'ABCDE. Può invece risultare complessa quando il melanoma è nelle fasi iniziali o quando solo alcuni caratteri clinici dell'ABCDE vengono riscontrati nella lesione considerata. Da tenere presente che una diagnosi errata può portare sia ad asportazioni inutili di nei sia ad una sottovalutazione della gravità di melanomi cutanei. Per fare in modo che questo accada il meno possibile lo specialista può quindi ricorrere all'utilizzo di strumenti, che consentono di effettuare osservazioni migliori rispetto all'osservazione diretta ad occhio nudo, come ad esempio il dermatoscopio manuale, il videodermatoscopio oppure tecniche come ad esempio la tecnica dell'epiluminescenza che, in particolare, utilizza un mezzo di contrasto.

La diagnosi certa di melanoma cutaneo necessita però di una biopsia, in cui un campione di tessuto della lesione considerata viene prelevato e poi analizzato al microscopio.

Inoltre, grazie a specifiche analisi sul campione di tessuto prelevato, è possibile identificare la presenza di mutazioni molecolari tipiche di alcune forme di melanoma cutaneo e utili per definire prognosi e trattamento. Esami di diagnostica per immagini come raggi x del torace, TAC, PET e risonanza magnetica sono utili per definire se e dove la malattia si è estesa.

1.5 Epidemiologia

Il melanoma cutaneo risulta essere rarissimo prima della pubertà e colpisce prevalentemente soggetti di età compresa tra i 30 ed i 60 anni.

Il melanoma cutaneo è stato considerato fino a pochi anni fa un tipo di tumore raro, mentre oggi è in crescita costante in tutto il mondo. La sua incidenza è cresciuta con un incremento di circa il 15% nell'ultimo decennio, mentre il tasso di mortalità ha subito incrementi più modesti. I tassi di incidenza più elevati si riscontrano nelle aree molto soleggiate e abitate da popolazioni di ceppo nordeuropeo, con la pelle particolarmente chiara. Si stima che nell'ultimo decennio ogni anno nel mondo vengano diagnosticati circa 100.000 nuovi casi di questo tipo di tumore [1].

1.6 Prevenzione

I dati sopra riportati testimoniano che l'incidenza del melanoma cutaneo è effettivamente aumentata, in particolare nelle popolazioni europee. Tuttavia la quasi stabilità del tasso di mortalità è ascrivibile ai primi risultati favorevoli ad una diagnosi precoce.

Inoltre si è riscontrato un aumento dei tassi di incidenza dei melanomi sottili rispetto a quelli più spessi e anche questo porta a ritenere che i progressi diagnostici siano in parte alla base degli aumenti di incidenza, soprattutto nell'ambito degli ultimi decenni.

Infatti, negli ultimi 20 anni, le modalità di trattamento non sono cambiate in modo molto significativo, mentre è aumentato notevolmente il numero dei melanomi asportati in fase precoce. Questo aspetto risulta particolarmente evidente in quelle aree geografiche in cui sono state condotte delle campagne di prevenzione secondaria.

Il melanoma cutaneo ha infatti una prognosi strettamente dipendente dallo spessore raggiunto nella pelle al momento della sua diagnosi ed asportazione. Se il melanoma, nel momento in cui viene diagnosticato, è rimasto ancora confinato agli strati cutanei superficiali, la prognosi è generalmente buona, con guarigione del paziente. Viceversa, se il melanoma ha già raggiunto gli strati più profondi, perché ha avuto molto tempo di accrescersi prima della sua diagnosi ed asportazione, i rischi di vita per il paziente sono molto elevati. Individuare il melanoma quanto più precocemente possibile rappresenta quindi la principale arma per tentare di ridurre la mortalità.

Grazie, quindi, a campagne di prevenzione secondaria, adesso la quota di melanomi scoperti quando la prognosi può essere ancora favorevole è arrivata al 60%-70% circa. Il risultato principale è stato soprattutto il miglioramento della sopravvivenza: nel 1960 solo circa la metà dei malati di melanoma era ancora in vita 5 anni dopo la prima diagnosi, mentre oggi lo è circa l'80% di essi, quindi un 30% in più [1].

1.7 Supporto computerizzato alla diagnosi

Da quanto detto precedentemente l'incidenza del melanoma cutaneo è in aumento e in particolare emerge il fatto che la prevenzione ed una diagnosi precoce sono fondamentali per fare in modo che il melanoma cutaneo non diventi pericoloso. Purtroppo però la

diagnosi clinica di melanoma è generalmente difficoltosa ed è condizionata direttamente dall'esperienza del medico con sensibilità oscillante tra il 50% e l'85% [2].

Da qualche anno è in fase di realizzazione e sviluppo un progetto per la costruzione di una macchina per il supporto medico nell'individuazione di lesioni melanocitiche e attualmente è stato prodotto un primo tipo di macchina che è pronto per essere venduto. Il progetto è a cura del Dott. Ivan Tomba e dell'Ing. Andrea Visotti della ditta CAMI S.r.l., con la supervisione del Prof. Massimo Ferri dell'Università di Bologna e la collaborazione del Prof. Ignazio Stanganelli dell'Università di Parma il quale collabora anche con l'IRST (Istituto Scientifico Romagnolo per lo Studio e la Cura dei Tumori) di Meldola. L'obiettivo è stato quello di creare una macchina contenente un database di immagini di nei e melanomi già diagnosticati che, una volta acquisita una nuova immagine di una lesione, restituisca le 10 immagini più "vicine" a quella in esame tra tutte quelle contenute nel database. In questo modo il medico può vedere le diagnosi delle lesioni più "vicine" a quella che sta esaminando in quel momento.

Un aspetto che è fondamentale sottolineare è che non si vuole assolutamente fornire una diagnosi che possa sostituire quella del dermatologo, ma si vuole cercare di agevolare ed aiutare il lavoro di quest'ultimo attraverso l'utilizzo del recupero di immagini già diagnosticate.

Il principio che sta alla base del funzionamento di tale macchina sfrutta la teoria dell'omologia persistente e verrà spiegato nei capitoli successivi dove, oltre a questo, verrà illustrata la modifica che si è pensato di fare all'attuale algoritmo al fine di rendere il funzionamento di tale macchina più veloce cercando però di non perdere in "bontà" dei risultati. Questa modifica è poi quella che è stata studiata e sviluppata in questo progetto di tesi.

Per il lettore interessato ad approfondire quanto introdotto in relazione al melanoma cutaneo consigliamo la lettura del libro [3]. Consigliamo inoltre la lettura di [4] e [5] per chi volesse conoscere altri progetti di natura matematica effettuati al fine di migliorare le funzionalità di sistemi di recupero di immagini dermatologiche.

1.8 Esempi di immagini di lesioni cutanee

Riportiamo di seguito alcune immagini di lesioni cutanee per dare un'idea di come possano essere i nei ed i melanomi tipici, ovvero quelli con caratteristiche morfologiche spesso caratteristiche di quella particolare classe di lesioni cutanee, ed altre per mostrare come alcune volte, invece, nei e melanomi possano avere caratteristiche molto simili tra loro. Questo per potere osservare parte di quello spiegato in questo capitolo e per poi capire al meglio il problema di cui ci andremo ad occupare in questo progetto.

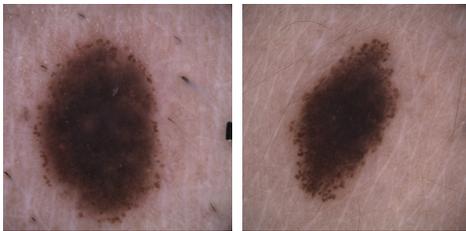


Figura 1.1: *Nei*.

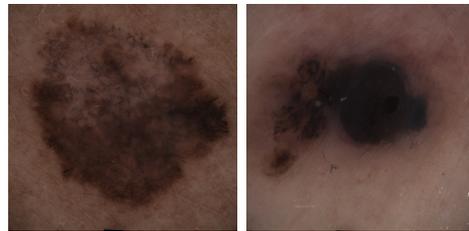


Figura 1.2: *Melanomi*.



Figura 1.3: *Coppia neo-melanoma*.

Possiamo notare che nelle lesioni cutanee raffigurate nelle prime immagini è possibile riconoscere molte delle caratteristiche tipiche rispettivamente di nei e melanomi spiegate precedentemente.

Nel caso, invece, di questa particolare coppia neo-melanoma anche i medici specialisti hanno dato un voto di 3/3 per la somiglianza “visivo-clinica” tra le due immagini. In casi come questo è praticamente impossibile anche per un esperto capire quale dei due sia il neo e quale il melanoma. In questo caso l'immagine di sinistra è un neo e l'immagine di destra un melanoma e questo è stato accertato solo dopo l'esame istopatologico.

Capitolo 2

La matematica alla base dell'algoritmo

Nel seguente capitolo introduciamo i concetti matematici che saranno alla base dell'algoritmo che verrà illustrato nei capitoli successivi della tesi. In particolare avremo un primo paragrafo dedicato ad alcune nozioni dell'omologia persistente ed un secondo paragrafo incentrato su una relazione che esiste tra soluzioni e coefficienti di un dato polinomio, formalizzata dalle formule di Viète.

2.1 L'omologia persistente

Consideriamo la seguente domanda: "Sono più simili un uomo in piedi ed un uomo seduto oppure un uomo seduto ed un panda seduto?". Per certi scopi può essere ragionevole considerare più simili i primi due, in quanto, ad esempio, sono lo stesso essere vivente, mentre per altri può essere opportuno considerare più importante la posizione e ritenere più simili due esseri viventi diversi però entrambi seduti. Quindi per rispondere adeguatamente ad una domanda del genere, se chiamiamo A e B i protagonisti del confronto, non dobbiamo confrontare A e B così come sono, ma dobbiamo cercare di farlo studiando A e B in base alla caratteristica che ci interessa confrontare in quel momento.

Questo è quello che fa l'omologia persistente con gli spazi topologici. Infatti, dati due spazi topologici X e Y , l'omologia persistente cerca di andare oltre al fatto di studiare X e Y così come sono, e cerca di farlo studiando le coppie (X, f) e (Y, f) dove f è una funzione filtrante, la quale oltre a darci l'idea della forma ci dà informazioni anche sul punto di

vista dell'osservatore. Quindi possiamo dire che l'omologia persistente cerca di rispondere alla domanda: "Quanto si somigliano due spazi topologici X e Y guardati attraverso una certa funzione filtrante f ?". Il diverso punto di vista si può quindi concretizzare nella scelta della funzione filtrante.

Questo viene dal fatto che se uno deve analizzare, classificare o dare un giudizio a delle immagini raffiguranti ad esempio pezzi meccanici, veicoli o oggetti rigidi, allora si può procedere utilizzando la geometria e questo risulta essere più che sufficiente. Sull'immagine stessa, la teoria delle matrici ci fornisce la trasformazione per sovrapporre l'oggetto dell'immagine ad un modello considerato. Inoltre, molte volte, immagini di questo tipo sono caratterizzate da vettori di misure geometriche, ognuno dei quali è associato ad una particolare caratteristica dell'oggetto rappresentato nell'immagine. In questi casi il riconoscimento di un'immagine, la rilevazione di un difetto dell'oggetto rappresentato, il confronto tra più immagini o un'operazione di questo genere avviene, più semplicemente, su tali vettori, rispettivamente per ogni caratteristica considerata.

Questo però, solitamente, non risulta essere più valido se l'oggetto raffigurato nell'immagine è di origine naturale. In questi casi la rigidità della geometria diventa un ostacolo. Riconoscere ad esempio la somiglianza tra un uomo seduto e un uomo in piedi è difficile e la sfida diventa ancora più difficile quando si iniziano a considerare immagini biomediche e quando il contesto è essenziale per capire i dati. In questo caso si può utilizzare la topologia in quanto esiste una trasformazione topologica che permette ad esempio di sovrapporre un uomo in piedi e un uomo seduto (vedremo in seguito che si dirà che questi sono tra loro omeomorfi), mentre non esiste alcuna matrice in grado di fare questo.

In realtà non è facile capire se due oggetti sono omeomorfi tra loro, però la topologia ci può aiutare in quanto ad ogni spazio topologico vengono associati i propri valori di invarianti topologici (per esempio i numeri di Betti), in modo che valga il seguente risultato: oggetti che sono omeomorfi tra loro hanno gli stessi invarianti topologici, però, sfortunatamente, il viceversa non vale sempre.

Rimane comunque un problema: se la geometria è troppo rigida, la topologia lo è troppo poco in quanto gli spazi tra loro omeomorfi possono risultare molto differenti da un punto di vista intuitivo (vedremo ad esempio che una tazza con manico è omeomorfa

ad una ciambella). Questa è la ragione per cui la topologia persistente, in particolare l'omologia persistente, può fornire dei nuovi descrittori topologici (per esempio le funzioni dei numeri di Betti persistenti e i diagrammi di persistenza) i quali preservano alcune caratteristiche selezionate attraverso delle funzioni filtranti e rendono possibile distinguere due oggetti anche se essi sono omeomorfi tra loro.

Questa impostazione conferisce un'estrema applicabilità alla teoria: si ottengono funzioni dei numeri di Betti persistenti non solo da immagini, ma anche, ad esempio, da suoni o da sequenze video.

Si parla di topologia, omologia “persistente” proprio perché l'informazione più rilevante è insita nei cicli con “vita” più lunga. Infatti i cicli poco persistenti rappresentano “rumore” e questi si riconoscono perché producono triangoli piccoli nelle funzioni PBN e punti vicini alla “diagonale” $u = v$ nei diagrammi di persistenza.

Definiamo adesso alcuni dei concetti principali dell'omologia persistente (tra questi quelli nominati in precedenza) con lo scopo di rendere più chiaro il motivo dell'utilizzo di determinati concetti matematici all'interno dell'algoritmo pensato nello specifico per una macchina di supporto medico, in particolare in campo dermatologico.

Definizione 2.1. *Omeomorfismo.*

Siano X e Y due spazi topologici e sia $\varphi : X \rightarrow Y$ funzione. Se φ è continua con inversa continua, allora φ si dice *omeomorfismo* da X a Y . Se esiste una funzione con tali caratteristiche allora i due spazi X e Y si dicono *omeomorfi*.

Definizione 2.2. *Funzione filtrante.*

Sia X uno spazio topologico. Una *funzione filtrante* f è una funzione $f : X \rightarrow \mathbb{R}$ tale che f sia una funzione continua.

Definizione 2.3. *Numeri di Betti.*

Sia k un numero intero non negativo. Si definisce *k -esimo numero di Betti* relativo allo spazio topologico X , $\beta_k(X)$, la dimensione del k -esimo spazio vettoriale di omologia $\mathcal{H}_k(X)$, ovvero il numero di generatori indipendenti (le classi di omologia dei k -cicli) di tale spazio. Intuitivamente $\beta_0(X)$ conta il numero di componenti connesse per archi di X , $\beta_1(X)$ conta i buchi (il numero di 1-cicli indipendenti) di X , $\beta_2(X)$ conta gli spazi vuoti (il numero di 2-cicli indipendenti) che compongono X .

Notazione: in questo capitolo, da ora in avanti, con k indichiamo un numero intero non negativo.

Definizione 2.4. *Insieme di sottolivello.*

Consideriamo la coppia (X, f) , dove X spazio topologico e f funzione filtrante. Sia $l \in \mathbb{R}$. L'insieme di sottolivello relativo a l è l'insieme $X_l = \{x \in X / f(x) \leq l\}$.

Notazione: in questo capitolo, da ora in avanti, ogni volta che scriviamo una coppia del tipo (X, f) intendiamo X spazio topologico e $f : X \rightarrow \mathbb{R}$ funzione continua, cioè f funzione filtrante.

Definizione 2.5. *Funzioni dei numeri di Betti persistenti.*

Per ogni $u, v \in \mathbb{R}$, $u < v$, l'inclusione $i^{u,v} : X_u \rightarrow X_v$ è una funzione continua ed induce, per ogni k , una trasformazione lineare $i_*^{u,v} : \mathcal{H}_k(X_u) \rightarrow \mathcal{H}_k(X_v)$. Le funzioni dei numeri di Betti persistenti in grado k (k -PBN) associano ad ogni coppia (u, v) il numero relativo a $\dim(\text{Im } i_*^{u,v})$, in seguito denominato anche $\beta_k^{u,v}(X, f)$, ovvero il numero di classi di k -cicli di $\mathcal{H}_k(X_u)$ che “sopravvivono” in $\mathcal{H}_k(X_v)$.

Definizione 2.6. *Diagramma di persistenza.*

Le funzioni corrispondenti ai k -PBN sono caratterizzate dalla posizione di determinati punti e rette di discontinuità, chiamati rispettivamente *cornerpoints* e *cornerlines* (o *cornerpoints* all'infinito). Dato un cornerpoint di coordinate (u, v) , u e v rappresentano rispettivamente il livello di “nascita” e “morte” di un generatore. Data una cornerline di ascissa w , w rappresenta il livello della “nascita” di un generatore che “non muore mai”. La persistenza di un cornerpoint (u, v) è la differenza $v - u$ delle proprie coordinate. Il k -esimo diagramma di persistenza $\mathcal{D}_k(X, f)$ è formato dai cornerpoints e dalle cornerlines che caratterizzano la funzione corrispondente al k -PBN.

Più precisamente cornerpoints e cornerlines vanno contati con la propria molteplicità. Introduciamo le seguenti notazioni per definire la molteplicità di un cornerpoint e dare una seconda versione della definizione di diagramma di persistenza.

Notazioni: $\Delta^+ = \{(u, v) \in \mathbb{R}^2 : u < v\}$, $\Delta = \{(u, v) \in \mathbb{R}^2 : u = v\}$, $\bar{\Delta}^+ = \Delta^+ \cup \Delta$.

Definizione 2.7. *Molteplicità di $(u, v) \in \Delta^+$.*

Sia k un numero intero non negativo e sia $(u, v) \in \Delta^+$. La *molteplicità* $\mu_k(u, v)$ di (u, v) è il numero finito non negativo definito dal seguente limite:

$$\lim_{\epsilon \rightarrow 0^+} (\beta_k^{u+\epsilon, v-\epsilon}(X, f) - \beta_k^{u-\epsilon, v-\epsilon}(X, f) - \beta_k^{u+\epsilon, v+\epsilon}(X, f) + \beta_k^{u-\epsilon, v+\epsilon}(X, f)).$$

Diamo adesso un'altra definizione di diagramma di persistenza.

Definizione 2.8. *Diagramma di persistenza.*

Il k -esimo diagramma di persistenza $\mathcal{D}_k(X, f)$ è l'unione tra l'insieme di tutti i punti $(u, v) \in \Delta^+$ tali che $\mu_k(u, v) > 0$, contati con la loro molteplicità, e i punti di Δ , contati con molteplicità infinita. I punti di un diagramma di persistenza appartenenti a Δ^+ spesso vengono chiamati *punti propri*.

In questa tesi tutti i k -esimi diagrammi di persistenza che andremo a considerare sono formati da un numero finito di punti $(u, v) \in \Delta^+$ tali che $\mu_k(u, v) > 0$.

Definizione 2.9. *Distanza di bottleneck o distanza di matching.*

Date le coppie (X, f) e (Y, g) tali che i rispettivi diagrammi di persistenza $\mathcal{D}_k(X, f)$ e $\mathcal{D}_k(Y, g)$ abbiano un numero finito di punti $(u, v) \in \Delta^+$ tali che $\mu_k(u, v) > 0$, consideriamo tutti i possibili confronti tra i cornerpoints di $\mathcal{D}_k(X, f)$ e i cornerpoints di $\mathcal{D}_k(Y, g)$ e tra i cornerpoints e le rispettive proiezioni sulla "diagonale" $u = v$. Il peso di questo confronto è il max tra le distanze delle coppie di punti considerati. La *distanza di bottleneck* tra $\mathcal{D}_k(X, f)$ e $\mathcal{D}_k(Y, g)$ è il min tra i pesi di tutti i possibili confronti. Traducendo il tutto con un formalismo matematico otteniamo il seguente risultato.

Considerati i diagrammi di persistenza $\mathcal{D}_k(X, f)$ e $\mathcal{D}_k(Y, g)$ relativi, rispettivamente, alle coppie (X, f) e (Y, g) , la distanza di bottleneck $d_B(\mathcal{D}_k(X, f), \mathcal{D}_k(Y, g))$ è definita come segue:

$$d_B(\mathcal{D}_k(X, f), \mathcal{D}_k(Y, g)) = \min_{\sigma} \max_{P \in \mathcal{D}_k(X, f)} \hat{d}(P, \sigma(P))$$

dove σ varia tra tutte le possibili biezioni tra $\mathcal{D}_k(X, f)$ e $\mathcal{D}_k(Y, g)$ e, dati $(u, v) \in \mathcal{D}_k(X, f)$ e $(u', v') \in \mathcal{D}_k(Y, g)$,

$$\hat{d}((u, v), (u', v')) = \min \left\{ \max \{|u - u'|, |v - v'|\}, \max \left\{ \frac{v - u}{2}, \frac{v' - u'}{2} \right\} \right\}.$$

Per capire meglio in che modo la distanza di bottleneck confronta due diagrammi di persistenza si può fare riferimento all'immagine riportata di seguito.

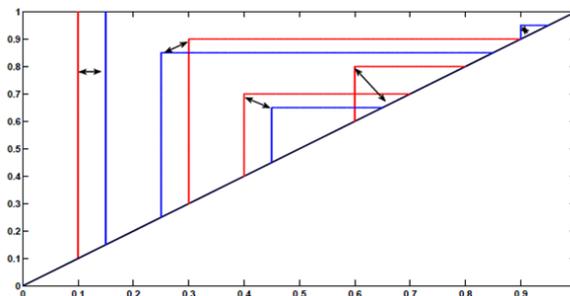


Figura 2.1: La distanza di bottleneck tra due diagrammi di persistenza si ottiene cercando la migliore corrispondenza tra i due insiemi di cornerpoints considerati.

Definizione 2.10. *Pseudodistanza naturale.*

Considerate le coppie (X, f) e (Y, g) tali che X e Y siano omeomorfi tra loro, il peso di un dato omeomorfismo $\varphi : X \rightarrow Y$ è il $\sup_{x \in X} |g(\varphi(x)) - f(x)|$. La *pseudodistanza naturale* tra (X, f) e (Y, g) , $d((X, f), (Y, g))$, è l'inf tra i pesi di tutti i possibili omeomorfismi tra X e Y . Questo significa che:

$$d((X, f), (Y, g)) = \inf_{\varphi} \sup_{x \in X} |g(\varphi(x)) - f(x)|,$$

dove φ varia tra tutti i possibili omeomorfismi tra X e Y .

Riportiamo di seguito la definizione di grafo di taglia il quale risulta essere uno strumento molto utile nelle applicazioni come punto di partenza per costruire un determinato diagramma di persistenza, relativo ad un'immagine considerata, rispetto ad una data funzione filtrante.

Definizione 2.11. *Grafo di taglia.*

Sia (G, ψ) una coppia dove G è un grafo e $\psi : \mathcal{V}(G) \rightarrow \mathbb{R}$ una funzione definita sui vertici di G a valori reali. Allora la coppia (G, ψ) viene chiamata *grafo di taglia*.

Per il lettore interessato ad approfondire la teoria dell'omologia persistente e, più in generale, della topologia persistente, consigliamo la lettura di alcune referenze tecniche quali [6] e i lavori di P. Frosini e dei suoi collaboratori [7], [8], [9].

Esempio 1. Vediamo adesso un esempio per cercare di capire al meglio e per cercare di visualizzare, per quanto possibile, alcuni dei concetti appena definiti.

Consideriamo i seguenti oggetti: una tazza con manico e una ciambella.

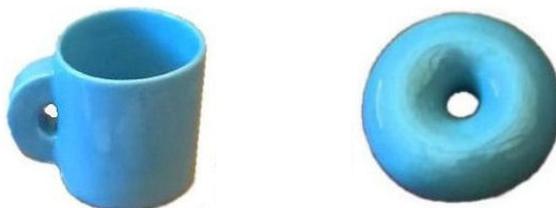


Figura 2.2: *Tazza con manico e ciambella.*

Si può verificare che i due oggetti sono omeomorfi, ovvero che, denotando con X la tazza e con Y la ciambella, esiste un omeomorfismo $\varphi : X \rightarrow Y$. Dal punto di vista visivo l'idea può essere data dall'immagine seguente, dove si può vedere che con deformazioni continue si può passare da un oggetto all'altro.



Figura 2.3: *Trasformazione da tazza a ciambella.*

Dato che i due oggetti considerati sono omeomorfi tra loro, allora i numeri di Betti di X e Y coincidono, in particolare:

$$\beta_0(X) = \beta_0(Y) = 1, \quad \beta_1(X) = \beta_1(Y) = 1, \quad \beta_2(X) = \beta_2(Y) = 0.$$

Infatti, entrambi gli oggetti hanno una componente connessa per archi, un 1-ciclo e zero 2-cicli.

Andiamo adesso a studiarne gli insiemi di sottolivello. Per fare questo possiamo pensare di mettere la tazza X (rovesciata) e la ciambella Y (in posizione verticale) su uno stesso piano, ed usare come funzione filtrante f la quota. Quindi avremo, rispettivamente, $f_X : X \rightarrow \mathbb{R}$ e $f_Y : Y \rightarrow \mathbb{R}$ e andremo a considerare le coppie (X, f_X) e (Y, f_Y) . Supponiamo, senza ledere la generalità, che entrambi gli oggetti abbiano quota minima a e quota massima c , e che il buco del manico della tazza e il buco della ciambella abbiano inizio e fine rispettivamente a quote uguali. Sia $\epsilon \in \mathbb{R}$, $\epsilon > 0$, ϵ piccolo a piacere e andiamo a considerare $a - \epsilon$ che sarà quindi un numero poco più piccolo di a . Otteniamo allora i seguenti risultati:

$$X_{a-\epsilon} = Y_{a-\epsilon} = \emptyset, \quad X_c = X, \quad Y_c = Y.$$

Per avere un'idea più concreta di quanto appena detto, si può fare riferimento all'immagine riportata di seguito.

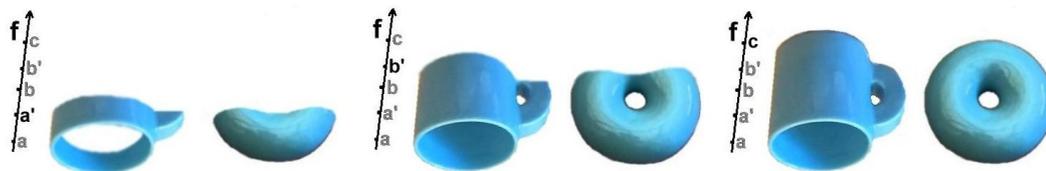


Figura 2.4: *Insiemi di sottolivello per la tazza e per la ciambella.*

Se applichiamo, inoltre, l'omologia agli insiemi di sottolivello intermedi, allora possiamo distinguere la tazza dalla ciambella. Infatti abbiamo che:

$$\beta_1(X_{a'}) = 1 \neq 0 = \beta_1(Y_{a'}), \quad \beta_1(X_{b'}) = 2 \neq 1 = \beta_1(Y_{b'}),$$

con $a < a' < b < b' < c$.

Una volta visti gli insiemi di sottolivello di X e Y , cerchiamo di capire come sono fatte in questi casi le funzioni dei numeri di Betti persistenti in grado k , in particolare ci concentreremo sul caso $k = 1$. Questa funzione nel caso $k = 1$ è illustrata nell'immagine seguente, rispettivamente per la tazza e la ciambella, dove $c' = c - (\text{spessore del fondo della tazza})$.

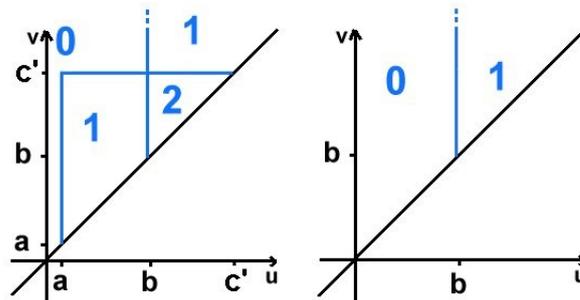


Figura 2.5: Funzioni 1-PBN, rispettivamente, per la tazza e per la ciambella.

Per quanto riguarda la tazza possiamo osservare che il trapezio di vertici (a, a) , (b, b) , (b, c') , (a, c') ci dice che l'1-ciclo presente negli insiemi di sottolivello $X_{a'}$, con $a \leq a' < b$, “sopravvive” negli $X_{b'}$ con $a' < b' < c'$ fino a “morire” in $X_{c'}$ (“ucciso” dal fondo della tazza). Il triangolo adiacente, invece, ci dice che i due 1-cicli di un $X_{b'}$ con $b \leq b' < c'$ continuano ad essere presenti anche in tutti gli $X_{b''}$ se $b' < b'' < c'$, ma non se $b'' \geq c'$, poiché in questo caso solo l'1-ciclo del manico “sopravvive”. Per quanto riguarda la ciambella possiamo vedere che c'è un 1-ciclo che “nasce” al livello b e che poi non “muore” mai.

Queste stesse informazioni possono essere lette anche nei rispettivi diagrammi di persistenza degli oggetti considerati. Costruendo i diagrammi di persistenza, relativi alla tazza ed alla ciambella, nel caso $k = 1$, otteniamo i risultati riportati di seguito.

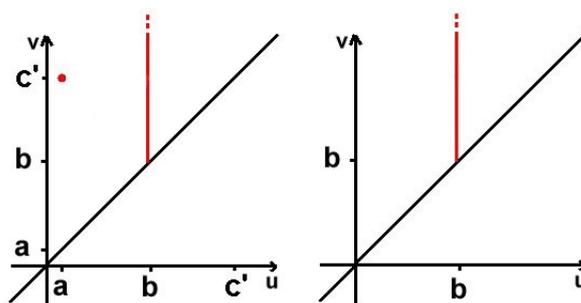


Figura 2.6: $\mathcal{D}_1(X, f_X)$ e $\mathcal{D}_1(Y, f_Y)$.

Leggendo questi diagrammi otteniamo le stesse informazioni dateci dalle 1-PBN, infatti, nel caso della tazza, il cornerpoint in posizione (a, c') ci indica che c'è un 1-ciclo che

“nasce” al livello a fino a “morire” al livello c' , mentre la cornerline con origine nel punto (b, b) ci dice che ci sono degli 1-cicli che “nascono” al livello b e che poi non “muoiono” più. Mettendo insieme le informazioni date sia dal cornerpoint che dalla cornerline allora capiamo che ci sono due 1-cicli che “nascono” al livello a di cui uno “muore” al livello c' mentre l'altro non “muore” mai. Nel caso della ciambella, invece, abbiamo unicamente la cornerline con origine nel punto (b, b) , la presenza della quale significa che abbiamo un 1-ciclo che “nasce” al livello b e che poi non “muore” più.

Quanto spiegato in questo paragrafo si può trovare negli articoli di M. Ferri [10], [11] e in parte nell'articolo [12].

2.2 Le formule di Viète

Proposizione 2.1. (*Principio di identità dei polinomi*).

Sia K campo infinito. Si considerino due polinomi $p(t), p'(t) \in K[t]$ della forma

$$p(t) = b_0 t^n + b_1 t^{n-1} + \dots + b_{n-1} t + b_n$$

$$p'(t) = b'_0 t^m + b'_1 t^{m-1} + \dots + b'_{m-1} t + b'_m,$$

con $b_0 \neq 0$ e $b'_0 \neq 0$.

Se le funzioni polinomiali relative a $p(t)$ e $p'(t)$ coincidono, allora $m = n$ ed i coefficienti relativi a t^j sono uguali per ogni $j = 0, 1, \dots, n$, cioè $b_0 = b'_0, b_1 = b'_1, \dots, b_{n-1} = b'_{n-1}, b_n = b'_n$.

Proposizione 2.2. (*Formule di Viète*).

Sia $p(t) = b_0 t^n + b_1 t^{n-1} + \dots + b_{n-1} t + b_n$ un polinomio di grado $n \geq 1$ a coefficienti complessi ($b_0, b_1, \dots, b_{n-1}, b_n \in \mathbb{C}$ e $b_0 \neq 0$) e siano z_1, z_2, \dots, z_n le sue radici, ripetute con la loro molteplicità. Le formule di Viète affermano che:

$$z_1 + z_2 + \dots + z_n = -\frac{b_1}{b_0}$$

$$(z_1 z_2 + z_1 z_3 + \dots + z_1 z_n) + (z_2 z_3 + z_2 z_4 + \dots + z_2 z_n) + \dots + (z_{n-1} z_n) = \frac{b_2}{b_0}$$

...

$$(z_1 z_2 \dots z_n) = (-1)^n \frac{b_n}{b_0}.$$

Queste formule possono essere riscritte nel modo seguente:

$$\sum_{1 \leq i_1 \leq i_2 \leq \dots \leq i_j \leq n} z_{i_1} z_{i_2} \dots z_{i_j} = (-1)^j \frac{b_j}{b_0}$$

per ogni $j = 1, 2, \dots, n$.

Dimostrazione. Per il Teorema fondamentale dell'algebra sappiamo che $p(t)$ ha n radici in \mathbb{C} : z_1, z_2, \dots, z_n . Per il Teorema di Ruffini sappiamo che vale la seguente uguaglianza

$$p(t) = (t - z_1)(t - z_2) \dots (t - z_n).$$

Applicando adesso il Principio di identità dei polinomi si ottengono le formule richieste. \square

Nota 2.1. Nei capitoli successivi utilizzeremo le formule di Viète per calcolare i coefficienti di un polinomio monico date le rispettive radici e verrà illustrato in che modo abbiamo utilizzato tutto questo per calcolare delle distanze tra diagrammi di persistenza. Da sottolineare il fatto che per convenzione parleremo sempre di coefficienti complessi, relativi ai polinomi monici costruiti, calcolati utilizzando le formule di Viète, però in realtà noi andremo a considerare sempre e solo le “funzioni simmetriche elementari” delle radici, cioè i coefficienti calcolati a meno di un segno pari a $(-1)^j$ per il coefficiente relativo al monomio di grado $n - j$, con $j = 1, 2, \dots, n$, se ipotizziamo di lavorare con un polinomio monico di grado n .

Ipotizzando di volere costruire un polinomio monico avente come radici i numeri complessi z_1, \dots, z_n applichiamo le formule di Viète ed otteniamo i coefficienti $\frac{b_1}{b_0}, \dots, \frac{b_n}{b_0}$ e quindi il polinomio monico:

$$p(t) = t^n + \frac{b_1}{b_0} t^{n-1} + \dots + \frac{b_{n-1}}{b_0} t + \frac{b_n}{b_0}.$$

In seguito, quando parleremo di coefficienti complessi calcolati, relativi ai polinomi monici costruiti, non faremo quindi riferimento ai coefficienti veri e propri $\frac{b_1}{b_0}, \dots, \frac{b_n}{b_0}$ del polinomio costruito (non consideriamo il coefficiente relativo al monomio di grado maggiore in quanto essendo polinomi monici questo sarà sempre uguale ad 1) ma alle

“funzioni simmetriche elementari” delle radici che possiamo identificare con a_1, \dots, a_n dove

$$a_j = (-1)^j \frac{b_j}{b_0} = \sum_{1 \leq i_1 \leq i_2 \leq \dots \leq i_j \leq n} z_{i_1} z_{i_2} \dots z_{i_j}$$

per ogni $j = 1, 2, \dots, n$.

Considerando a_1, \dots, a_n , il polinomio monico costruito può quindi essere riscritto come:

$$p(t) = t^n - a_1 t^{n-1} + \dots + (-1)^j a_j t^{n-j} + \dots + (-1)^n a_n.$$

Capitolo 3

L'algoritmo attuale ed il nuovo algoritmo

In questo capitolo verrà illustrato l'algoritmo che attualmente è contenuto nella macchina, pensata e costruita per il supporto medico in campo dermatologico, introdotta nel primo capitolo. In particolare ci soffermeremo su una parte di questo algoritmo che è quella che è stata presa in esame e modificata in questo progetto di tesi. Successivamente verranno spiegati, in dettaglio, il motivo e le caratteristiche delle modifiche apportate.

3.1 L'algoritmo attuale

3.1.1 Introduzione al funzionamento della macchina

Come introdotto brevemente nel primo capitolo l'obiettivo dei ricercatori è stato quello di creare una macchina per il supporto medico, in particolare in campo dermatologico, contenente un database di immagini di nei e melanomi già diagnosticati che, una volta acquisita una nuova immagine di una lesione cutanea, restituisca le 10 immagini più "vicine", rispetto a quella presa in esame, tra tutte quelle contenute nel database. Da sottolineare il fatto che con tale macchina non si vuole assolutamente fornire una diagnosi che possa sostituire quella del dermatologo, ma si vuole cercare di agevolare e aiutare il lavoro di quest'ultimo attraverso l'utilizzo del recupero di immagini già diagnosticate.

In realtà, inizialmente, nel momento in cui la macchina viene prodotta e venduta, questa contiene un database ridotto cioè formato da poche immagini. Tali immagini sono immagini di lesioni cutanee già diagnosticate ed acquisite da una macchina dello stesso tipo e modello di quella considerata. Questo database verrà poi aggiornato nel momento in cui la macchina avrà acquisito altre immagini di lesioni cutanee di cui si conosce la diagnosi corretta. In questo modo il database che si viene a costruire contiene immagini tutte acquisite da macchine dello stesso tipo e modello e questo porta ad avere (ed in seguito a poter confrontare) immagini con dei parametri alla base che sono gli stessi, come ad esempio la luminosità o il fattore di ingrandimento. Da questo segue che, nel momento in cui si andranno a confrontare immagini di lesioni cutanee tra loro, un cambiamento di luminosità tra le lesioni o una differenza di dimensione tra queste sarà sicuramente dovuto ad una reale differenza tra le lesioni considerate e non a differenze meccaniche, o di costruzione, della macchina che si sta utilizzando rispetto alla macchina che si è utilizzata per acquisire le immagini del database. Questo è fondamentale per poter costruire un buon database, se poi come caratteristiche di confronto si utilizzano parametri che a priori potrebbero dipendere dal tipo di macchina. Caratteristiche come queste ne esistono molte come ad esempio quelle appena nominate oppure tonalità e intensità di un certo colore.

Il confronto tra le immagini viene effettuato principalmente sfruttando la teoria dell'omologia persistente. In particolare, date due immagini, queste vengono confrontate considerando 19 caratteristiche, quindi è come se inizialmente guardassimo le immagini da 19 differenti punti di vista. Di queste caratteristiche 11 vengono considerate come funzioni filtranti e quindi verrà utilizzata l'omologia persistente per lo studio e il confronto di queste, mentre le altre 8 sono parametri standard e quindi verranno utilizzate altre tecniche per effettuare il confronto.

Le 11 funzioni filtranti sono relative a: luminosità su tutta l'immagine segmentata, canale del Blu, canale del Verde, canale del Rosso, canale dell'Excess Blue, canale dell'Excess Green, canale dell'Excess Red (un'idea di come possono essere calcolati i valori degli Excess Blue, Excess Green ed Excess Red è illustrata in [13]), filtrazione inversa della luminosità, luminosità sulla zona periferica della lesione, bordo (distanza dal baricentro) e filtrazione inversa della distanza dal baricentro. Gli 8 parametri standard

sono relativi a: istogramma dei colori, fattore di forma, circolarità di Haralick, simmetria media, ellitticità, eccentricità, diametro ed entropia del colore.

3.1.2 I passaggi principali dell'algoritmo

Più nello specifico, il lavoro svolto dall'algoritmo che è contenuto all'interno della macchina può essere descritto attraverso alcuni passaggi principali.

Dopo essere state acquisite, le immagini vengono prima di tutto private degli eventuali peli presenti. In questo modo si eliminano alcune delle possibili fonti di rumore per i passi successivi. L'immagine viene poi segmentata, ovvero viene determinato il bordo della lesione, in modo da estrapolare la lesione vera e propria su cui andranno a lavorare i passi ancora successivi. A questo punto, per ogni immagine, in relazione ad ogni singola caratteristica, viene calcolato il grafo di taglia in base alla quantità di pixel che si decide di considerare (questa varia in base alla caratteristica data) [14]. Da questi grafi di taglia si costruiscono gli 0-diagrammi di persistenza da cui si può ricavare la lista dei cornerpoints e calcolare poi le distanze. Prima vengono calcolate singolarmente le 19 distanze, ognuna riferita ad una specifica caratteristica, e poi vengono calcolate le distanze globali tra le varie immagini. Dopo aver fatto tutto questo viene effettuato il retrieval. Questo si differenzia a seconda che la macchina stia lavorando "real time" o "off line". Nel momento "real time", cioè nel caso in cui viene acquisita una nuova immagine nel sistema, la quale viene poi confrontata attraverso i passi precedenti con tutte le immagini già presenti nel database, il retrieval restituisce in output le 10 immagini più "vicine" a quella presa in esame, ovvero le 10 immagini corrispondenti alle 10 distanze globali più piccole dall'immagine considerata. Il momento "off line", invece, lavora su tutte le immagini contenute nel database ed in particolare prende tutti i valori calcolati attraverso i passi precedenti, che rappresentano tutte le distanze reciproche tra le immagini (confrontate due a due), e li raccoglie creando in output le matrici delle distanze. Il calcolo di queste matrici risulta particolarmente utile anche ai fini di un successivo momento "real time". Infatti, l'aver già calcolato tutti i vari dati relativi alle immagini del database permette al momento "real time" di evitare alcuni confronti tra le immagini, in modo da abbassare il costo computazionale e, di conseguenza, il tempo di esecuzione dell'algoritmo. Inoltre,

le matrici delle distanze sono particolarmente interessanti in quanto, a partire da esse, è possibile esprimere una valutazione sul sistema di recupero [15].

3.1.3 Il calcolo delle distanze

Vediamo adesso, un po' più nel dettaglio, come vengono calcolate le distanze tra le immagini nell'algoritmo attuale, in quanto sarà quella la parte di codice che andremo a considerare e a modificare.

Per quanto detto in precedenza, possiamo dire che per calcolare le distanze si parte dai grafi di taglia delle immagini. Per ognuno di questi, utilizzando un particolare algoritmo illustrato nell'articolo [16], vengono calcolati i corrispondenti 0-diagrammi di persistenza, ognuno per ogni funzione filtrante. Quindi ad ogni immagine vengono associati 11 diagrammi di persistenza. A questo punto viene usata la distanza di bottleneck fra questi per calcolare le prime 11 matrici delle distanze. Tutto questo vale solo per le prime 11 matrici in quanto sono quelle relative ai diagrammi di persistenza, relativi a loro volta alle prime 11 caratteristiche descritte precedentemente. Per quanto riguarda le altre distanze, invece, le otteniamo come segue: la distanza fra due istogrammi dei colori viene calcolata utilizzando un metodo spiegato nell'articolo [14], mentre per le altre si ha che la distanza fra due immagini rispetto ad un dato parametro standard è data dal valore assoluto della differenza fra i due valori del parametro considerato relativi alle immagini che si stanno confrontando. Una volta calcolate queste distanze vengono costruite anche le altre 8 matrici delle distanze.

Dato N_{db} il numero di immagini da confrontare tra loro, queste 19 matrici delle distanze avranno dimensione $N_{db} \times N_{db}$ e il posto (i, j) di ogni singola matrice conterrà la distanza tra l'immagine i e l'immagine j rispetto alla caratteristica relativa alla matrice considerata. Da questo e dalle proprietà di una distanza segue che tali matrici sono simmetriche con la diagonale principale formata da tutti 0.

La distanza globale tra due immagini viene poi calcolata come media pesata delle 19 distanze calcolate, una per ogni caratteristica.

Come appena descritto, qui il confronto tra i diagrammi di persistenza viene effettuato utilizzando la distanza di bottleneck in quanto si tratta di uno strumento stabile. Questo è dovuto al modo in cui viene definita tale distanza (Definizione 2.9). Appros-

simativamente, piccoli cambiamenti nei valori delle funzioni filtranti producono piccoli cambiamenti nei diagrammi di persistenza relativi.

3.2 Il nuovo algoritmo

Rispetto all'algoritmo attuale, nel nuovo algoritmo cambia il modo in cui vengono calcolate le distanze, tra le immagini considerate, rispetto alle singole funzioni filtranti. La differenza tra gli algoritmi riguarda quindi il calcolo delle singole distanze e, in particolare, il calcolo delle distanze relative alle prime 11 caratteristiche. Il calcolo delle distanze relative ai parametri standard ed il calcolo delle distanze globali rimane invariato, come il resto dell'algoritmo.

3.2.1 L'obiettivo

L'obiettivo di questo cambiamento è ridurre il costo computazionale e, di conseguenza, il tempo di esecuzione del programma in quanto la bottleneck ha un costo computazionale abbastanza elevato. Questo al fine di riuscire a lavorare con database di maggiori dimensioni e potere quindi confrontare tra loro un numero maggiore di immagini. Si cerca poi di fare tutto questo senza perdere in "bontà" dei risultati. In questo modo si spera di riuscire a dare un maggiore supporto allo specialista. Inoltre, utilizzando il metodo dei polinomi contenuto in questo nuovo algoritmo, si cerca di andare a considerare il contributo di tutti i cornerpoints dei diagrammi di persistenza, per poi andare ad effettuare un confronto tra questi, in modo da tralasciare meno informazioni possibili anche se questo potrebbe però portare ad una minore stabilità dell'algoritmo.

3.2.2 Introduzione al nuovo algoritmo

In questo progetto tutti i diagrammi di persistenza che andremo a considerare sono 0-diagrammi di persistenza con un numero finito di cornerpoints ed un'unica cornerline (l'oggetto studiato e rappresentato nell'immagine considerata ha sempre un'unica componente connessa per archi).

Come detto precedentemente, attualmente la distanza tra i diagrammi di persistenza viene calcolata con la distanza di bottleneck. Da prove effettuate dai ricercatori in questi anni tale metodo porta buoni risultati, infatti questo è contenuto nell'algoritmo realmente presente nella macchina. C'è però un problema: la distanza di bottleneck soffre di "esplosione combinatoria", quindi diventa difficile, in termini di tempo di esecuzione dell'algoritmo, applicarla a diagrammi di persistenza con un numero molto grande di cornerpoints, oppure quando il numero di confronti da effettuare è elevato. Approssimazioni, organizzazioni dei dati particolarmente intelligenti, applicazioni progressive di diversi classificatori possono aiutare, ma il problema viene alleggerito, non risolto. Un'altra caratteristica della distanza di bottleneck, che si vorrebbe cercare di migliorare, è il fatto che per come viene definita (Definizione 2.9) va a considerare un minimo tra dei particolari massimi e quindi è come se, dati due diagrammi di persistenza, considerasse solo alcuni cornerpoints, rispettivamente dell'uno e dell'altro, per poi andare ad effettuare un confronto tra questi.

Un nuovo modo di procedere è frutto di un'idea di Claudia Landi: rappresentare i diagrammi di persistenza come l'insieme delle radici complesse di un polinomio. A questo punto il confronto può essere fatto sui coefficienti dei polinomi complessi così costruiti.

Due problemi però si presentano inizialmente: uno, relativo alla natura stessa dei diagrammi di persistenza, è che nella situazione reale ci sono molti punti vicino alla "diagonale", spesso dovuti al rumore e quindi meno importanti nella rappresentazione dell'immagine, ma con un grande impatto sui coefficienti dei polinomi; un altro problema, questo relativo alla teoria dei polinomi, è che piccole distanze tra le radici implicano piccole distanze tra i coefficienti però non vale il viceversa. Per quanto riguarda il primo problema, ovvero l'esistenza dei punti vicino alla "diagonale", si potrebbe considerare una trasformazione che prende tutti i punti (u, v) della retta $u = v$ e li porta in $(0, 0)$. In questo modo i punti vicino alla "diagonale" verranno raggruppati intorno allo zero. Portando i punti corrispondenti al rumore attorno allo zero si diminuisce il contributo che essi danno ai coefficienti dei polinomi (questo segue dalle formule di Viète). Per il secondo problema, cioè per il fatto che, in riferimento ai polinomi, coefficienti vicini non implicano radici vicine, si può pensare di utilizzare il confronto tra polinomi in una fase di preprocessing in modo da scartare velocemente oggetti molto diversi da quello in

esame, in modo da utilizzare poi la distanza di bottleneck solo su un insieme ridotto di candidati.

Quanto studiato e sviluppato in questo progetto di tesi parte da questo nuovo modo di procedere [12]. In parte consideriamo come punti di partenza risultati già ottenuti dalle prime prove effettuate con questa nuova idea e, in parte, apporteremo ulteriori modifiche per cercare di migliorare l'intero algoritmo. Tutti i risultati che riporteremo nel capitolo successivo riguardano unicamente il nuovo metodo e non la collaborazione di questo con la distanza di bottleneck. Tali risultati verranno poi confrontati con i risultati ottenuti utilizzando, invece, solo la distanza di bottleneck.

Al momento, l'approccio per convertire i diagrammi di persistenza in vettori complessi può essere applicato solo se non vengono considerate le cornerlines. Noi abbiamo studiato un possibile modo per tenere conto anche di queste, in modo da tralasciare meno informazioni possibili. Non considerando le cornerlines si perde la conoscenza del minimo e questa potrebbe essere, invece, importante. Ad esempio, se come funzione filtrante consideriamo quella relativa alla tonalità del colore marrone, date due immagini di lesioni cutanee entrambe aventi un colore abbastanza omogeneo all'interno della lesione stessa però una sul marrone chiaro ed una sul marrone molto scuro tendente al nero queste potrebbero avere diagrammi di persistenza molto simili tra loro a parte per la cornerline.

Inoltre i test preliminari effettuati dai ricercatori in questi anni con questo metodo suggeriscono che i coefficienti dei polinomi costruiti relativi ai monomi di grado maggiore siano più importanti rispetto agli altri. Nelle prove che abbiamo effettuato in questo progetto teniamo conto anche di questo e vediamo come variano i risultati al variare del numero di coefficienti complessi considerati.

Per quanto detto spostiamo quindi il problema di confrontare i diagrammi di persistenza da un confronto diretto attraverso la distanza di bottleneck ad un confronto tra vettori complessi le cui componenti sono i coefficienti, a meno di un segno (Nota 2.1), dei polinomi che hanno come radici i numeri complessi associati ai cornerpoints dei diagrammi di persistenza considerati.

Vediamo adesso, più nel dettaglio, come si ottengono questi vettori a partire dai diagrammi di persistenza e come si procede poi con il calcolo delle singole distanze.

3.2.3 Il nuovo algoritmo in dettaglio

Dato un diagramma di persistenza, per prima cosa definiamo una trasformazione che porta i punti di questo in numeri complessi. Successivamente andremo poi a costruire un polinomio a coefficienti complessi con la proprietà di avere come radici i numeri complessi appena calcolati. Tornando alla trasformazione, per quanto detto precedentemente, cerchiamo una trasformazione che mandi i punti (u, v) tali che $u = v$ in $(0, 0)$ in modo da dare meno importanza ai cornerpoints considerati relativi al rumore. In questo progetto noi prenderemo in considerazione tre trasformazioni di questo tipo: la prima è già stata pubblicata in un articolo di Barbara Di Fabio e Massimo Ferri [12]; la seconda e la terza, invece, sono una composizione di una trasformazione pubblicata nel medesimo articolo e di una pensata da noi durante il lavoro a questo progetto.

Consideriamo inizialmente le trasformazioni già pubblicate:

$$T : \bar{\Delta}^+ \rightarrow \mathbb{C}, \quad T(u, v) = \frac{v-u}{2}(\cos(\alpha) - \text{sen}(\alpha) + i(\cos(\alpha) + \text{sen}(\alpha))); \quad (3.1)$$

$$S : \bar{\Delta}^+ \rightarrow \mathbb{C}, \quad S(u, v) = \begin{cases} \frac{v-u}{\alpha\sqrt{2}}(u + iv), & \text{se } (u, v) \neq (0, 0); \\ (0, 0), & \text{altrimenti;} \end{cases}$$

dove $\alpha = \sqrt{u^2 + v^2}$.

Sia

$$\hat{t} = \arctan\left(\frac{\text{Re}(S(u, v))}{\text{Im}(S(u, v))}\right)$$

e siano

$$t_1 = 2\pi\left(1 - \sqrt{1 - \frac{4}{\pi}\hat{t}}\right), \quad t_2 = 2\pi\left(\frac{1 - \sqrt{-15\left(\frac{4}{\pi}\hat{t}\right)^2 - \frac{4}{\pi}\hat{t} + 16}}{4}\right),$$

definiamo adesso le due nuove trasformazioni:

$$NT_1 : \bar{\Delta}^+ \rightarrow \mathbb{C}, \quad NT_1(u, v) = \begin{cases} \alpha_1(\text{sen}(t_1)) + i\alpha_1(\cos(t_1)), & \text{se } (u, v) \neq (0, 0); \\ (0, 0), & \text{altrimenti;} \end{cases} \quad (3.2)$$

$$NT_2 : \bar{\Delta}^+ \rightarrow \mathbb{C}, \quad NT_2(u, v) = \begin{cases} \alpha_1(\text{sen}(t_2)) + i\alpha_1(\cos(t_2)), & \text{se } (u, v) \neq (0, 0); \\ (0, 0), & \text{altrimenti;} \end{cases} \quad (3.3)$$

dove $\alpha_1 = \sqrt{\operatorname{Re}(S(u, v))^2 + \operatorname{Im}(S(u, v))^2}$.

Le tre trasformazioni che utilizzeremo sono: T , NT_1 e NT_2 .

Le trasformazioni utilizzate sono tutte funzioni continue e, in particolare, le trasformazioni NT_1 e NT_2 sono iniettive su Δ^+ . Tali trasformazioni non sono però iniettive su $\bar{\Delta}^+$ in quanto tutti i punti del tipo (u, u) ($u \in \mathbb{R}$) vengono portati in $(0, 0)$.

Definiamo la molteplicità di un numero complesso appartenente all'immagine di T , NT_1 o NT_2 come la somma delle molteplicità dei punti (cornerpoints) appartenenti alla controimmagine del numero complesso considerato rispetto alla trasformazione che si sta utilizzando. Questo è necessario data la non iniettività di T , anche se in realtà nel nostro caso è molto difficile che la controimmagine di un numero complesso contenga più di un cornerpoint.

In tutte queste trasformazioni, come detto precedentemente, i punti vicino alla “diagonale”, relativi principalmente al rumore, vengono portati in un intorno di $(0, 0)$ in modo che, andando a calcolare somme e somme di prodotti (formule di Viète), venga ridotto il loro impatto sui coefficienti dei polinomi che andremo a costruire. Inoltre, queste trasformazioni hanno in comune il fatto che tutte portano i cornerpoints a distanza dall'origine $(0, 0)$ uguale alla loro distanza originaria dalla “diagonale” $u = v$.

Le principali differenze tra le trasformazioni considerate sono le seguenti:

- la trasformazione T , oltre a portare i cornerpoints a distanza dall'origine $(0, 0)$ uguale alla loro distanza originaria dalla “diagonale” $u = v$, li ruota in modo tale da cercare di portare alcuni punti anche negli altri quadranti del piano cartesiano (in modo che i punti trasformati abbiano almeno una coordinata negativa) per cercare di ridurre ulteriormente il loro impatto sui coefficienti dei polinomi che andremo a costruire. Può succedere però che con questa rotazione un punto ruoti di un angolo maggiore rispetto ad un angolo giro quindi questo può portare ad una situazione in cui i trasformati di cornerpoints distanti tra loro siano invece punti vicini tra loro.
- la trasformazione NT_1 agisce come la trasformazione T con la differenza che la rotazione che produce dilata la metà del quadrante che va dal semiasse positivo delle ordinate alla “diagonale” $u = v$ (tutti i cornerpoints sono contenuti in quella

parte di piano cartesiano) in modo tale che i punti con coordinate $(0, v)$ ($v \in \mathbb{R}$) rimangano sul semiasse positivo delle ordinate, mentre i punti di coordinate (u, u) ($u \in \mathbb{R}$) vengano ruotati sul semiasse positivo delle ordinate. Questo per evitare che succeda quanto descritto per la trasformazione T . Infatti in questo caso gli unici cornerpoints lontani i cui trasformati, attraverso la rotazione, sono vicini sono quelli che stanno sulle semirette che delimitano la metà del quadrante di cerchio considerato, mentre per il resto i cornerpoints non vengono “mescolati” tra loro. Applicando poi la trasformazione completa, anche per i cornerpoints appena considerati, viene ad esserci una differenza nella distanza dall'origine a cui vengono portati, per quanto detto precedentemente. Inoltre l'angolo di rotazione a cui viene sottoposto il cornerpoint aumenta al diminuire della distanza di quest'ultimo dalla retta $u = v$. Questa proprietà dell'angolo di rotazione è stata pensata sempre per cercare di dare una maggiore importanza ai cornerpoints che sono più lontani dalla retta $u = v$.

- la trasformazione NT_2 agisce come la trasformazione NT_1 con la differenza che questo fatto che l'angolo di rotazione a cui viene sottoposto il cornerpoint aumenta al diminuire della distanza di quest'ultimo dalla retta $u = v$ è ancora più accentuato.

Dato un diagramma di persistenza \mathcal{D} , siano $c_1 = (u_1, v_1), \dots, c_h = (u_h, v_h)$, con $c_1, \dots, c_h \in \mathbb{R}^2$, i cornerpoints (punti propri) relativi con molteplicità, rispettivamente, r_1, \dots, r_h . Per il momento non consideriamo la cornerline. Chiamiamo z_1, \dots, z_h i numeri complessi ottenuti da c_1, \dots, c_h attraverso una delle trasformazioni T, NT_1, NT_2 . Associamo quindi al diagramma di persistenza \mathcal{D} considerato il polinomio complesso

$$p_{\mathcal{D}}(t) = \prod_{j=1}^h (t - z_j)^{r_j}.$$

Calcoliamo poi i coefficienti complessi di tale polinomio utilizzando le formule di Viète.

Ricordiamo che (Nota 2.1) per convenzione parleremo sempre di coefficienti complessi, relativi ai polinomi monici costruiti, calcolati utilizzando le formule di Viète, però in realtà noi andremo a considerare sempre e solo le “funzioni simmetriche elementari” delle radici.

Una volta calcolati i coefficienti complessi a_1, \dots, a_n possiamo scrivere

$$p_{\mathcal{D}}(t) = t^n - a_1 t^{n-1} + \dots + (-1)^j a_j t^{n-j} + \dots + (-1)^n a_n.$$

Da osservare che i cornerpoints relativi ai diagrammi di persistenza che consideriamo in questo progetto hanno molteplicità 1 (come c'è da aspettarsi se si lavora con immagini raffiguranti oggetti di origine naturale) quindi avremo $h = n$.

Consideriamo adesso due diagrammi di persistenza \mathcal{D} e \mathcal{D}' e siano

$$p_{\mathcal{D}}(t) = t^n - a_1 t^{n-1} + \dots + (-1)^j a_j t^{n-j} + \dots + (-1)^n a_n$$

e

$$p_{\mathcal{D}'}(t) = t^m - a'_1 t^{m-1} + \dots + (-1)^j a'_j t^{m-j} + \dots + (-1)^m a'_m$$

i polinomi complessi ad essi associati. Vediamo ora come potere risolvere il primo problema dato dalla possibilità che i due polinomi abbiano grado diverso, cioè che $m \neq n$ (supponiamo, senza ledere la generalità, $m < n$).

Visto il modo in cui abbiamo costruito questi polinomi si preferisce procedere confrontando tra loro i coefficienti relativi allo stesso indice piuttosto che i coefficienti relativi a monomi con lo stesso grado in t .

Questo problema lo possiamo quindi gestire aggiungendo $n - m$ coefficienti nulli a $p_{\mathcal{D}'}(t)$, cioè moltiplicando $p_{\mathcal{D}'}(t)$ per t^{n-m} che significa aggiungere come radice il numero complesso zero con molteplicità $n - m$. In questo modo possiamo costruire due vettori di numeri complessi (a_1, \dots, a_n) e (a'_1, \dots, a'_n) della stessa lunghezza e siamo quindi pronti per calcolare la distanza tra essi. Per la continuità delle formule di Viète radici vicine implicano coefficienti vicini.

A questo punto andiamo a calcolare la distanza tra i due vettori di numeri complessi appena costruiti. Come detto in precedenza, da alcuni test risulta che i coefficienti relativi ai monomi di grado maggiore siano più importanti rispetto agli altri, per questo si può pensare a delle distanze (tra i polinomi complessi) che tengano conto anche di questo. Possibili distanze da poter utilizzare sono le seguenti:

$$d_1 = \sum_{j=1}^n |a_j - a'_j|, \quad d_2 = \sum_{j=1}^n \frac{|a_j - a'_j|}{j}, \quad d_3 = \sum_{j=1}^n |a_j - a'_j|^{1/j} \quad (3.4)$$

Di queste distanze d_1 considera tutti i coefficienti con la stessa importanza, mentre d_2 e d_3 attribuiscono ai coefficienti un grado di importanza che decresce al crescere dell'indice di tali coefficienti.

In questo progetto abbiamo scelto di considerare inizialmente un'unica distanza, in particolare la distanza d_2 , e di variare altre componenti all'interno dell'algoritmo. Ovviamente altri tipi di prove ed osservazioni possono essere fatte tenendo costanti le altre componenti e variando il tipo di distanza che si va ad utilizzare.

A questo punto, una volta calcolate le distanze per ogni caratteristica considerata, riusciamo a costruire le matrici delle distanze.

Prima di passare alle prove effettuate ed ai risultati ottenuti, spieghiamo in che modo andremo a considerare le cornerlines in tutto questo procedimento appena illustrato. Ricordiamo che per ogni diagramma di persistenza considerato, in questo caso, abbiamo un'unica cornerline. Il problema delle cornerlines è che sono cornerpoints con ordinata infinita e quindi non riusciamo a portarle in numeri complessi attraverso le trasformazioni che abbiamo visto. Abbiamo quindi pensato di "travestire" le cornerlines da cornerpoints in modo da trattarle poi nello stesso modo. La trasformazione che viene utilizzata è la seguente: considerato il diagramma di persistenza \mathcal{D} con cornerline di ascissa w , questa viene trasformata nel punto

$$(w, \max_j v_j), \quad j \in \{1, \dots, N\} \quad (3.5)$$

dove N è il numero dei cornerpoints di \mathcal{D} e v_j , al variare di j , l'ordinata del j -esimo cornerpoint di \mathcal{D} .

Durante il lavoro a questo progetto abbiamo pensato anche ad ulteriori modifiche da apportare al nuovo algoritmo, sempre al fine di dare più importanza ai coefficienti complessi relativi ai monomi di grado maggiore, rispetto agli altri. Inizialmente la modifica apportata è stata la seguente: dati due polinomi da confrontare, dopo averli portati, come descritto precedentemente, ad avere lo stesso grado n , si divide il coefficiente di indice j , relativo quindi al monomio di grado $n - j$ in t , di ogni polinomio considerato, per il coefficiente binomiale $\binom{n}{j}$ con $j \in \{1, \dots, n\}$. Siano, rispettivamente, (a_1, \dots, a_n) e (a'_1, \dots, a'_n) i vettori dei coefficienti complessi dei polinomi considerati, questi vengono

trasformati nel modo seguente:

$$\begin{aligned} a_j &\longmapsto \frac{a_j}{\binom{n}{j}}, & j \in \{1, \dots, n\} \\ a'_j &\longmapsto \frac{a'_j}{\binom{n}{j}}, & j \in \{1, \dots, n\} \end{aligned} \tag{3.6}$$

In questo modo al crescere di j il coefficiente binomiale considerato cresce finché $j \leq \lfloor \frac{n}{2} \rfloor$ e quindi, rimanendo entro i limiti in cui tale condizione risulta essere verificata, il coefficiente complesso relativo al monomio di grado $n - j$ in t (al crescere di j il grado in t di questo monomio, $n - j$, decresce) viene diviso per un numero sempre più grande e quindi gli viene data minore importanza. Non analizziamo i valori di j per cui, al crescere di j , tale coefficiente binomiale decresce poiché questo metodo lo useremo insieme al fatto che andremo a considerare solo un certo numero k ($k \in \mathbb{N}$) di coefficienti complessi e non andremo mai a considerare $k > \lfloor \frac{n}{2} \rfloor$. Rispetto alla divisione che si effettua nel calcolo della distanza d_2 questo metodo al crescere dell'indice j (con $j \leq \lfloor \frac{n}{2} \rfloor$) diminuisce di molto l'importanza da dare al coefficiente complesso relativo a quell'indice. Questo è dovuto al fatto che al denominatore della distanza d_2 abbiamo l'indice j mentre in questo caso al denominatore mettiamo il coefficiente binomiale scritto precedentemente il quale al crescere di j (con $j \leq \lfloor \frac{n}{2} \rfloor$) cresce molto più rapidamente. Tale trasformazione è stata pensata anche per cercare di evitare un possibile overflow dovuto al fatto che, applicando le formule di Viète, si vanno a calcolare molte somme e somme di prodotti tra i numeri complessi associati ai cornerpoints dei diagrammi di persistenza considerati. Sono state poi eseguite delle prove con tale modifica e, come illustrato nel capitolo successivo, abbiamo osservato che questo poteva non essere il modo ottimale per cercare di ottenere il risultato sperato. A questo punto abbiamo quindi applicato al nuovo algoritmo una nuova idea di modifica, sempre al fine di dare più importanza ai coefficienti complessi relativi ai monomi di grado maggiore rispetto agli altri e, come verrà spiegato meglio nel capitolo successivo, per cercare di togliere la dipendenza dei coefficienti complessi dal numero di cornerpoints dei diagrammi di persistenza relativi. Tale modifica consiste in questo: dati due polinomi da confrontare, dopo averli portati, come descritto precedentemente, ad avere lo stesso grado n , si applica ad ogni coefficiente complesso di entrambi i polinomi la trasformazione riportata di seguito. Siano, rispettivamente, (a_1, \dots, a_n) e (a'_1, \dots, a'_n) i vettori dei coefficienti complessi dei polinomi considerati, questi vengono trasformati nel

modo seguente:

$$\begin{aligned} a_j &\longmapsto \frac{|a_j|^{\frac{1}{j}}}{N} e^{i \arg(a_j)}, & j \in \{1, \dots, k\} \\ a'_j &\longmapsto \frac{|a'_j|^{\frac{1}{j}}}{M} e^{i \arg(a'_j)}, & j \in \{1, \dots, k\} \end{aligned} \quad (3.7)$$

dove N e M sono, rispettivamente, il numero di cornerpoints dei diagrammi di persistenza corrispondenti ai polinomi complessi considerati e k è, come detto in precedenza, il numero di coefficienti complessi che andiamo a considerare per poi effettuare il confronto tra i polinomi. Nel caso in cui applichiamo questa modifica appena illustrata andiamo a considerare come distanza tra i polinomi la distanza d_1 , illustrata precedentemente, per non avere un'ulteriore differenza di importanza tra i coefficienti in base al grado del monomio relativo e per vedere al meglio come lavora la nuova trasformazione, e non la distanza d_2 cioè quella considerata fino ad ora.

Nel capitolo successivo verranno illustrate le diverse prove che abbiamo effettuato utilizzando questo nuovo algoritmo applicandolo ad un database di immagini di lesioni dermatologiche già diagnosticate e verranno poi riportati ed analizzati i risultati ottenuti.

Capitolo 4

Prove effettuate e risultati ottenuti

Nelle prove effettuate utilizzando il nuovo algoritmo, di cui riporteremo in seguito i risultati ottenuti, vedremo come variano i risultati variando:

- il database iniziale, in particolare andremo a considerare due diversi database:
 - il primo contenente 295 immagini di cui 67 melanomi e 228 nei;
 - il secondo contenente 200 immagini di cui 40 melanomi e 160 nei;
- la risoluzione delle immagini del database iniziale, in particolare andremo a considerare:
 - per il primo database una risoluzione a 200 pixel ed una a 300 pixel;
 - per il secondo database una risoluzione a 200 pixel;
- la trasformazione che porta i cornerpoints in numeri complessi, in particolare andremo a considerare le trasformazioni illustrate nel capitolo precedente T , NT_1 , NT_2 ;
- il fatto di considerare o meno la cornerline;
- il numero di coefficienti complessi da considerare per calcolare la distanza tra i polinomi, in particolare chiameremo tale parametro k ed andremo a considerare inizialmente $k = 10, 20, 50$ e successivamente anche $k = 5$;

- il fatto di applicare o meno la divisione ai coefficienti complessi per i termini binomiali (3.6);
- il fatto di applicare o meno la nuova trasformazione ai coefficienti complessi (3.7).

Cercheremo poi di capire quale insieme di tali operazioni e parametri sia quello che porta ad avere risultati migliori ed andremo a confrontare questi con quelli ottenuti con l'algoritmo attuale (con la distanza di bottleneck) per vedere le differenze sia in termini di "bontà" dei risultati sia in termini di costo computazionale e tempo di esecuzione dell'algoritmo.

In generale, per ogni funzione filtrante, la parte di algoritmo che stiamo considerando prende in input i file, ognuno riferito ad un'immagine del database, contenenti ognuno la lista dei cornerpoints del diagramma di persistenza relativo a quella funzione filtrante e ad una particolare immagine. Come output, invece, viene restituita la matrice delle distanze relative sempre a quella funzione filtrante. A questo punto, non avendo a disposizione un giudizio di rilevanza proveniente da dermatologi, utilizziamo la diagnosi istologica, ovvero la classificazione neo-melanoma. Quindi per confrontare i risultati con quelli ottenuti con la distanza di bottleneck, consideriamo un algoritmo scritto da Ivan Tomba che, presa in input la matrice delle distanze relative ad una singola funzione filtrante e, conoscendo le informazioni relative al database che si sta utilizzando, applicando un kNN seguito da altre operazioni (maggiori dettagli sul funzionamento di questo algoritmo verranno riportati in seguito), restituisce in output i valori relativi a:

- *Sensibilità*: percentuale di melanomi individuati correttamente sul totale nel database dal sistema;
- *Specificità*: percentuale di nei individuati correttamente sul totale nel database dal sistema;
- *Accuratezza globale*: percentuale di lesioni individuate correttamente sul totale nel database dal sistema.

Questo in particolare è riferito ad una funzione filtrante data e quindi alle distanze relative ad essa, però possiamo avere lo stesso tipo di valori relativi alle distanze globali.

Per fare questo si calcola la matrice delle distanze globali come media pesata delle matrici delle distanze singole e si considera questa come input dell'algoritmo appena descritto. Come output in questo caso avremo i valori di sensibilità, specificità e accuratezza globale relativi alle immagini viste attraverso tutte le caratteristiche considerate. Nel caso si vogliano poi calcolare i risultati globali di tutto l'algoritmo allora bisogna tenere conto non solo delle informazioni relative alle 11 funzioni filtranti ma anche di quelle relative agli 8 parametri standard utilizzati.

Da specificare, inoltre, il fatto che il programma utilizzato per la parte del nuovo algoritmo analizzata prende in input da file le liste dei cornerpoint relativi ad ogni immagine, data una certa funzione filtrante, e come output principale, al fine di poter procedere con il resto dell'algoritmo per il funzionamento completo della macchina, restituisce la matrice delle distanze singole relative alla funzione filtrante considerata. In questo progetto però, dato che consideriamo in particolare questa parte del nuovo algoritmo, facciamo restituire al programma anche degli output intermedi in modo da poter osservare particolari passaggi dell'algoritmo. Più in dettaglio noi lavoriamo con due codici (una versione di tali codici è riportata nell'Appendice A): il primo (*PolyParte1*) per ogni immagine del database, data una certa funzione filtrante, prende in input da file la lista dei cornerpoints relativi a quell'immagine e restituisce come output un file contenente i cornerpoints trasformati attraverso una delle trasformazioni T , NT_1 , NT_2 , in base alle prove che andremo ad effettuare. Il secondo codice (*PolyParte2*), invece, prende come input l'output del codice precedente e restituisce come output un file contenente i $k + 1$ coefficienti complessi relativi ai $k + 1$ monomi di grado maggiore del polinomio relativo all'immagine e alla funzione filtrante considerate (in realtà restituisce k coefficienti in quanto stiamo considerando polinomi monici e il coefficiente relativo al monomio di grado maggiore, essendo uguale per tutti i polinomi, non viene restituito) ed un file contenente la matrice delle distanze singole relative alla funzione filtrante considerata.

Ricordiamo che (Nota 2.1) per convenzione parleremo sempre di coefficienti complessi, relativi ai polinomi monici costruiti, calcolati utilizzando le formule di Viète, però in realtà noi andremo a considerare sempre e solo le “funzioni simmetriche elementari” delle radici.

4.1 Primo database

Inizialmente abbiamo considerato un database, costruito da Ignazio Stanganelli utilizzando immagini di lesioni cutanee acquisite presso il centro di oncodermatologia di Ravenna, contenente 295 immagini di cui 67 melanomi e 228 nei. In questo database sono presenti numerose immagini di nei molto simili a melanomi e immagini di melanomi molto simili a nei quindi possiamo dire che, rispetto alle caratteristiche analizzate, non è presente un'evidente clusterizzazione delle due classi. Per un sistema computerizzato questo potrebbe complicare un po' le cose poiché se un sistema computerizzato afferma che un neo ed un melanoma sono simili ed effettivamente le immagini di questi si assomigliano da un punto di vista "oggettivo", rispetto alle caratteristiche analizzate, questo porta a commettere un errore importante se si valuta il risultato nel contesto del problema che stiamo considerando però non vuole dire che l'algoritmo all'interno del sistema non funzioni correttamente. In ogni caso noi andremo a confrontare i risultati ottenuti con il nuovo algoritmo con quelli ottenuti con l'algoritmo attuale e quindi questo risulta essere un problema secondario rispetto alla valutazione dei risultati ottenuti.

Considerando le immagini di tale database con RISOLUZIONE 200 ed utilizzando l'algoritmo attuale si ottengono i risultati, in termini di sensibilità, specificità ed accuratezza globale, riportati nella tabella seguente.

	acc.	sens.	spec.
LUM	71.86441	50.746269	78.070175
BLUE	72.203392	53.731346	77.631584
GREEN	70.847458	49.253731	77.192978
RED	69.830505	49.253731	75.877197
EXCB	55.932205	61.194031	54.385967
EXCG	65.762711	59.701492	67.543861
EXCR	64.406784	49.253731	68.85965
IL	58.305084	40.298508	63.596493
LBORD	72.542374	50.746269	78.947372
BORD	61.35593	37.313431	68.421051
IBORD	63.728817	41.791046	70.175438

Tabella 4.1: *Risultati bottleneck primo database, R200.*

Osservando questi primi risultati possiamo affermare che in questo caso in generale l'algoritmo attuale sulle singole funzioni filtranti ottiene risultati discreti ma non troppo soddisfacenti.

Vediamo ora, invece, il comportamento del nuovo algoritmo rispetto a questo database iniziale. Le prime prove che abbiamo deciso di fare sono state quelle relative appunto all'applicazione del nuovo algoritmo alle immagini di questo database considerando:

- RISOLUZIONE 200 per le immagini;
- solo cornerpoints e NO CORNERLINE;
- TRASFORMAZIONE T per passare da cornerpoints a numeri complessi;
- CON DIVISIONE dei coefficienti complessi per i termini binomiali, quindi SENZA trasformazione nuova ai coefficienti complessi.

Abbiamo deciso di fare queste scelte per iniziare a vedere come lavora effettivamente questo nuovo metodo per confrontare diagrammi di persistenza utilizzando i polinomi, in particolare abbiamo quindi deciso per un primo momento di non considerare le cornerlines e di considerare la divisione dei coefficienti complessi per i termini binomiali sia per dare più importanza ai coefficienti relativi ai monomi di grado maggiore, sia per evitare un possibile overflow dovuto al fatto che, applicando le formule di Viète, si vanno a calcolare molte somme e somme di prodotti tra i numeri complessi associati ai cornerpoints dei diagrammi di persistenza considerati. Abbiamo inoltre scelto di utilizzare inizialmente la trasformazione T (3.1) descritta nel capitolo precedente e solo in un momento successivo le altre trasformazioni. Per quanto riguarda il parametro k abbiamo deciso di considerare $k = 10, 20, 50$ in modo da osservare come variano i risultati al variare di k , per poi cercare di capire se i risultati dipendono molto o no da questo parametro.

I risultati ottenuti, in termini di sensibilità, specificità ed accuratezza globale, sono riportati nella tabella seguente.

	$k = 10$		
	acc.	sens.	spec.
LUM	58.305084	47.761192	61.403507
BLUE	55.254238	40.298508	59.64912
GREEN	60.338985	41.791046	65.789474
RED	53.220337	46.268658	55.263157
EXCB	62.372883	58.208954	63.596493
EXCG	58.64407	52.238804	60.526318
EXCR	51.86441	40.298508	55.263157
IL	58.983051	44.776119	63.15789
LBORD	58.64407	43.283581	63.15789
BORD	58.305084	46.268658	61.84211
IBORD	55.932205	43.283581	59.64912

Tabella 4.2: *Risultati polinomi primo database, R200, NCL, T, D.*

Osservando i risultati ottenuti possiamo notare che:

- in generale il nuovo algoritmo con le caratteristiche considerate porta ad ottenere risultati peggiori rispetto all'algoritmo attuale;
- all'aumentare del valore del parametro k non si riscontrano variazioni sui valori dei risultati ottenuti, quindi in questo caso i dati non mostrano dipendenza dal parametro k considerato. Per questo in tabella abbiamo riportato solamente i risultati relativi ad un valore del parametro k tra quelli considerati.

Andando ad osservare i tempi di esecuzione del programma si nota che i tempi relativi al nuovo algoritmo sono di ordini di grandezza minori rispetto ai tempi relativi all'algoritmo attuale. Non è però ragionevole andare ad esaminare tali tempi se con il nuovo algoritmo non si ottengono dei buoni risultati o almeno risultati comparabili a quelli ottenuti con l'algoritmo attuale.

Una volta effettuate queste prime prove siamo andati ad osservare i valori dei coefficienti complessi di alcuni dei polinomi costruiti ed abbiamo notato che al variare delle immagini e delle funzioni filtranti alcuni di questi tra i primi 10 relativi ai monomi di grado maggiore hanno parte reale ed immaginaria in valore assoluto dell'ordine di 10^{-10} o 10^{-15} mentre molti altri sono dell'ordine di 10^{-20} e, andando a guardare tra i primi 20, ce ne sono anche dell'ordine di 10^{-40} ed alcuni dell'ordine di 10^{-60} . Quindi abbiamo

deciso di effettuare delle prove analoghe a quelle appena fatte senza però applicare la divisione per i termini binomiali ai coefficienti complessi. Questo poiché, osservando i cornerpoints trasformati, molti di questi hanno coordinate (ovvero parte reale e parte immaginaria dei numeri complessi ad essi relativi) in valore assoluto minori di 1 e quindi applicando le formule di Viète non è assolutamente detto che si vada incontro ad un aumento rapido degli ordini di grandezza dei risultati ottenuti.

Abbiamo quindi effettuato le prove considerando:

- RISOLUZIONE 200 per le immagini;
- solo cornerpoints e NO CORNERLINE;
- TRASFORMAZIONE T per passare da cornerpoints a numeri complessi;
- SENZA DIVISIONE dei coefficienti complessi per i termini binomiali e SENZA trasformazione nuova ai coefficienti complessi.

Per quanto riguarda il parametro k abbiamo deciso di considerare $k = 5, 10, 20, 50$. Dato che nelle prove precedenti i risultati non hanno subito cambiamenti al variare del valore di k , abbiamo deciso di provare anche con $k = 5$ poiché se in questo modo, ipotizzando di ottenere dei buoni risultati con $k = 10$, i risultati non cambiassero in modo considerevole ma il tempo di esecuzione sì allora questo valore sarebbe da preferire rispetto agli altri.

Andando ad esaminare i risultati relativi a queste prove le osservazioni che si possono fare sono dello stesso tipo di quelle fatte per le prove precedenti. In questo caso all'aumentare del valore del parametro k alcuni valori per alcune funzioni filtranti, anche se di poco, variano in positivo o in negativo però in generale non ci sono grandi differenze. Questo vuol dire che anche se alcuni valori dei risultati ottenuti sono cambiati al variare di k questi non sembrano avere una forte dipendenza da tale parametro. Per quanto riguarda la divisione dei coefficienti complessi per il termine binomiale, invece, sembrerebbe, almeno in questo caso è così, non avere un grande impatto sulla validità o meno del nuovo algoritmo in generale.

Per quanto riguarda i tempi computazionali si osserva che sostanzialmente questi non variano tra $k = 5$ e $k = 10$ quindi si possono fare, in ogni caso, ulteriori prove con $k = 5$

ma nel caso in cui anche queste vadano a confermare quanto appena detto allora conviene utilizzare almeno $k = 10$ in modo da andare a considerare più informazioni riguardanti i polinomi considerati per poi calcolare le distanze.

Abbiamo poi deciso di provare ad aggiungere la cornerline alla lista dei cornerpoints relativi ad ogni immagine, data una certa funzione filtrante, applicando alla cornerline la trasformazione (3.5) illustrata nel capitolo precedente. Questo poiché la cornerline potrebbe fornire informazioni utili per poi distinguere due immagini tra loro. Ad esempio, come riportato anche nel capitolo precedente, immaginando di considerare come caratteristica la tonalità del colore marrone, date due immagini di lesioni cutanee entrambe aventi un colore abbastanza omogeneo all'interno della lesione stessa però una sul marrone chiaro ed una sul marrone molto scuro tendente al nero queste potrebbero avere diagrammi di persistenza molto simili tra loro a parte per la cornerline.

Abbiamo quindi ripetuto le prove considerando:

- RISOLUZIONE 200 per le immagini;
- cornerpoints CON CORNERLINE;
- TRASFORMAZIONE T per passare da cornerpoints a numeri complessi;
- SENZA DIVISIONE dei coefficienti complessi per i termini binomiali e SENZA trasformazione nuova ai coefficienti complessi.

Per quanto riguarda il parametro k , visti i risultati precedenti, abbiamo deciso di considerare solo due valori di k in particolare $k = 5, 10$.

Sfortunatamente però, osservando i risultati ottenuti, anche in questo caso non si notano particolari differenze. Aggiungendo le cornerlines alcuni valori dei risultati relativi ad alcune funzioni filtranti variano leggermente o in positivo o in negativo però comunque non in modo significativo. Quindi, almeno in questo caso, sembrerebbe che anche il fatto di considerare o meno la cornerline non abbia un grande impatto sulla validità o meno del nuovo algoritmo in generale.

In realtà andremo poi successivamente ad analizzare anche i risultati ottenuti dall'algoritmo globale utilizzato dalla macchina ovvero mettendo insieme le informazioni fornite dai calcoli relativi alle 11 funzioni filtranti considerate (con l'algoritmo attuale o con il nuovo algoritmo) e le informazioni fornite dai calcoli relativi agli 8 parametri standard considerati per cercare di raccogliere più informazioni sulle diverse immagini. In questo modo, in generale, i risultati dovrebbero migliorare in quanto le diverse caratteristiche analizzate (funzioni filtranti e parametri standard) in un certo senso cooperano tra loro. Questo tipo di risultati li esamineremo in seguito.

Utilizzando sempre questo database, dato che i cambiamenti apportati all'algoritmo tra le varie prove effettuate non hanno portato grandi cambiamenti nei risultati, abbiamo deciso, dato che i tempi computazionali non superano in questi casi i 60 *secondi* per ogni funzione filtrante per $k = 5, 10$ e i 120 *secondi*, sempre per ogni funzione filtrante, per $k = 20, 50$, di provare a considerare una risoluzione maggiore per le immagini del database in quanto questa potrebbe portare ad avere maggiori dettagli nelle immagini e quindi maggiori informazioni su cui basarsi per poi effettuare il confronto tra queste. Abbiamo quindi considerato una RISOLUZIONE 300 per le immagini del database iniziale ed abbiamo poi ripetuto tutte le prove effettuate per la risoluzione 200 appena illustrate.

I risultati ottenuti con risoluzione 300 non si discostano in modo significativo da quelli ottenuti con risoluzione 200. Quindi o la risoluzione delle immagini non è rilevante per questo modo di risolvere il problema considerato oppure c'è un elemento dell'algoritmo che fa in modo che questo cambiamento non influisca sui risultati che si ottengono. Da sottolineare che in questo caso i tempi maggiori di esecuzione del nuovo algoritmo relativi ad una data funzione filtrante arrivano ai 220 *secondi* circa quindi più alti rispetto a quelli relativi a risoluzione 200, come era ragionevole aspettarsi. Questi però continuano comunque ad essere tempi buoni se consideriamo il fatto che le prove con risoluzione 300 e l'algoritmo attuale non sono state fatte in quanto i tempi di esecuzione di questo algoritmo erano già alti nel caso della risoluzione 200 (circa 8 *ore* per tutte le 11 funzioni filtranti quindi in media circa 43 *minuti* per ogni funzione filtrante) e tempi maggiori di questi sarebbero stati difficili da gestire all'interno di una macchina di questo tipo. Anche in questo caso però non è ragionevole andare ad esaminare tali tempi se non si ottengono dei buoni risultati.

4.2 Nuovo database PH2

Visti i risultati ottenuti con il primo database considerato, abbiamo voluto testare il nuovo algoritmo anche su un altro database per cercare di capire se i risultati ottenuti con le prove precedenti sono meno soddisfacenti rispetto a quelli ottenuti con l'algoritmo attuale a causa del database utilizzato (con tale database abbiamo visto che neanche l'algoritmo attuale riesce ad ottenere buoni risultati), di un errore di concetto commesso in una trasformazione o in una delle modifiche apportate al nuovo algoritmo nelle diverse prove effettuate oppure se è proprio il nuovo metodo pensato per confrontare i diagrammi di persistenza a non essere adatto per risolvere il problema considerato.

Come nuovo database abbiamo deciso di considerare un database che si trova online [17]. Questa scelta può comportare svantaggi e vantaggi. Uno svantaggio è che all'interno di questo database possono essere presenti immagini acquisite con diverse macchine e quindi immagini che a priori possono avere livello di luminosità, fattore di ingrandimento e caratteristiche di questo genere diverse tra loro ed inoltre, nel caso specifico di questo database, in alcune immagini è rappresentata solo parte della lesione cutanea e questo non aiuta ad esempio la fase di segmentazione dell'immagine e di conseguenza rende più difficile esaminarne le caratteristiche. Un vantaggio, invece, è che, dato che tale database si trova online, sono presenti anche risultati, in termini di sensibilità e specificità, ottenuti da altri gruppi di ricerca con altri metodi, quindi una volta effettuate le prove applicando il nuovo algoritmo alle immagini di questo database potremo calcolare i risultati globali, relativi all'algoritmo globale utilizzato dalla macchina e non alle singole funzioni filtranti, che si ottengono in questi casi ed andare a confrontarli non solo con quelli globali relativi all'utilizzo dell'algoritmo attuale ma anche con quelli degli altri gruppi di ricerca.

Il nuovo database *PH2* contiene 200 immagini di cui 40 melanomi e 160 nei. Questo database rispetto al primo database utilizzato presenta una maggiore clusterizzazione delle due classi di lesioni cutanee considerate rispetto alle caratteristiche analizzate. Da questo possiamo quindi dire che è lecito aspettarsi risultati, in generale, non peggiori rispetto a quelli ottenuti utilizzando il primo database.

Osserviamo innanzitutto che considerando le immagini del nuovo database *PH2* con RISOLUZIONE 200 ed utilizzando l'algoritmo attuale si ottengono i risultati, in termini

di sensibilità, specificità ed accuratezza globale, riportati nella tabella seguente.

	acc.	sens.	spec.
LUM	73.5	62.5	76.25
BLUE	75.5	52.5	81.25
GREEN	74.5	60	78.13
RED	79.5	75	80.63
EXCB	91.5	87.5	92.5
EXCG	79.5	90	76.88
EXCR	84	87.5	83.13
IL	82.5	72.5	85
LBORD	79.5	65	83.13
BORD	84.5	80	85.63
IBORD	82	67.5	85.63

Tabella 4.3: *Risultati bottleneck nuovo database PH2, R200.*

Da tali risultati si nota subito che questi sono migliori rispetto a quelli ottenuti dallo stesso algoritmo attuale con le immagini del primo database. In questo caso molti valori raggiungono l'80%, cosa che non succedeva nel caso precedente. Con questo nuovo database possiamo dire che l'algoritmo attuale ottiene dei buoni risultati.

Dobbiamo ora vedere come lavora il nuovo algoritmo con queste immagini. Iniziamo allora effettuando le prime prove su questo nuovo database *PH2* utilizzando il nuovo algoritmo e considerando:

- RISOLUZIONE 200 per le immagini;
- solo cornerpoints e NO CORNERLINE;
- TRASFORMAZIONE T per passare da cornerpoints a numeri complessi;
- SENZA DIVISIONE dei coefficienti complessi per i termini binomiali e SENZA trasformazione nuova ai coefficienti complessi.

Abbiamo deciso di fare queste scelte per iniziare a vedere come lavora il nuovo algoritmo sul nuovo database *PH2*. Abbiamo deciso di non effettuare la divisione dei coefficienti complessi per i termini binomiali dalle osservazioni fatte precedentemente sul

primo database utilizzato. Per quanto riguarda il valore di k abbiamo deciso di considerare $k = 5, 10, 20, 50$ in modo da osservare bene come variano i risultati al variare di tale parametro.

I risultati ottenuti, in termini di sensibilità, specificità ed accuratezza globale, sono riportati nelle tabelle seguenti.

	$k = 5$			$k=10$		
	acc.	sens.	spec.	acc.	sens.	spec.
LUM	74.5	90	70.625	76	87.5	73.125
BLUE	69	80	66.25	68	75	66.25
GREEN	74.5	87.5	71.25	76.5	87.5	73.75
RED	79.5	85	78.125	79.5	80	79.375
EXCB	87	85	87.5	87	85	87.5
EXCG	77	90	73.75	77	90	73.75
EXCR	85	85	85	85	85	85
IL	78.5	85	76.875	81	82.5	80.625
LBORD	82.5	72.5	85	81.5	70	84.375
BORD	76.5	80	75.625	76.5	80	75.625
IBORD	78	82.5	76.875	78	82.5	76.875

	$k = 20$			$k=50$		
	acc.	sens.	spec.	acc.	sens.	spec.
LUM	75.5	90	71.875	75	90	71.25
BLUE	67.5	72.5	66.25	68	72.5	66.875
GREEN	76	82.5	74.375	76	82.5	74.375
RED	81.00	82.50	80.63	82	82.5	81.875
EXCB	87	85	87.5	87	85	87.5
EXCG	77	90	73.75	77	90	73.75
EXCR	85	85	85	85	85	85
IL	80.5	82.5	80	80	82.5	79.375
LBORD	81	70	83.75	81.5	67.5	85
BORD	76.5	80	75.625	76.5	80	75.625
IBORD	78	82.5	76.875	78	82.5	76.875

Tabella 4.4: *Risultati polinomi nuovo database PH2, R200, NCL, T, ND.*

Osservando i risultati riportati possiamo notare che:

- come l'algorithmo attuale anche il nuovo algorithmo restituisce risultati migliori sul nuovo database rispetto al primo database utilizzato e in generale sono risultati buoni;

- con le caratteristiche considerate il nuovo algoritmo porta ad ottenere risultati in generale simili a quelli ottenuti con l'algoritmo attuale anche se leggermente peggiori soprattutto se si vanno a guardare i valori relativi alla specificità;
- anche in questo caso all'aumentare del valore del parametro k alcuni valori per alcune funzioni filtranti, anche se di poco, variano in positivo o in negativo però in generale non ci sono grandi differenze.

Andando ad osservare i tempi di esecuzione del programma si nota che, anche utilizzando il nuovo database *PH2*, i tempi relativi al nuovo algoritmo sono di ordini di grandezza minori rispetto ai tempi relativi all'algoritmo attuale. I tempi relativi a queste prove appena descritte sono riportati nelle tabelle seguenti. Abbiamo riportato i tempi relativi a *PolyParte1* e *PolyParte2* che sono i due codici che utilizziamo per fare le prove descritti precedentemente in questo capitolo (una versione di tali codici è riportata nell'Appendice A). I valori dei tempi sono in *secondi*. Da notare il fatto che al codice *PolyParte1* è associato un solo tempo in quanto questo codice non dipende dal valore del parametro k considerato. Nella tabella seguente, come informazione aggiuntiva, abbiamo riportato anche il numero massimo di cornerpoints delle immagini del nuovo database *PH2* rispetto ad ogni funzione filtrante considerata.

	cp number max.	<i>PolyParte1</i>	<i>PolyParte2</i>			
	/		/	k=5	k=10	k=20
LUM	1811	4.34399986267	16.0809998512	17.1270000935	19.3009998798	25.2029998302
BLUE	1678	4.04500007629	15.3770000935	11.3440001011	11.9659998417	17.3440001011
GREEN	1746	3.82700014114	15.9479999542	17.0680000782	19.2239999771	25.5360000134
RED	1953	4.34399986267	16.373000145	17.4210000038	19.4000000954	25.8400001526
EXCB	1275	2.78500008583	8.06599998474	8.8259999752	10.7910001278	15.9819998741
EXCG	1644	3.15700006485	8.59200000763	10.1540000439	18.9249999523	25.5539999008
EXCR	1136	1.51799988747	14.3860001564	14.8740000725	16.4739999771	20.2709999084
IL	1814	3.83999991417	16.1510000229	17.2720000744	18.9300000668	25.118999958
LBORD	2993	5.60599994659	17.7239999771	19.2309999466	22.2509999275	30.9800000191
BORD	576	3.16299986839	7.2039999619	7.57199978828	8.96799993515	12.882999897
IBORD	576	1.51799988747	14.3860001564	14.8740000725	16.4739999771	20.2709999084

Tabella 4.5: *Tempi (in secondi) polinomi nuovo database PH2, R200, NCL, T, ND.*

Osservando i valori dei tempi di queste prove effettuate possiamo notare che, data una certa funzione filtrante e considerando uno dei valori di k riportati, mettendo insieme

i tempi dei due codici *PolyParte1* e *PolyParte2* in questo caso non si superano i 60 *secondi*. Quindi per calcolare le 11 matrici delle distanze relative alle 11 funzioni filtranti considerate non si superano gli 11 *minuti* mentre con l'algoritmo attuale considerando il nuovo database *PH2* e risoluzione 200 per le immagini si impiegano circa 5.5 *ore* per calcolare tali matrici.

A questo punto, in questo caso, possiamo dire di aver ottenuto dei buoni risultati anche con il nuovo algoritmo però vorremmo vedere se attuando una qualche modifica tra quelle prese in considerazione nelle prove precedenti eseguite utilizzando il primo database si ottengano, sempre con il nuovo algoritmo, dei risultati migliori.

Abbiamo quindi deciso, come fatto anche per il primo database, di considerare anche la cornerline oltre che ai cornerpoints per ogni diagramma di persistenza esaminato. Abbiamo quindi effettuato le prove considerando:

- RISOLUZIONE 200 per le immagini;
- cornerpoints CON CORNERLINE;
- TRASFORMAZIONE T per passare da cornerpoints a numeri complessi;
- SENZA DIVISIONE dei coefficienti complessi per i termini binomiali e SENZA trasformazione nuova ai coefficienti complessi.

Osservando i risultati ottenuti, anche in questo caso non si notano particolari differenze. Aggiungendo le cornerlines, anche utilizzando il nuovo database *PH2*, alcuni valori dei risultati relativi ad alcune funzioni filtranti variano leggermente, o in positivo o in negativo, però comunque non in modo significativo. Quindi continua a sembrare che il fatto di considerare o meno la cornerline non abbia un grande impatto sulla validità o meno del nuovo algoritmo in generale.

Abbiamo quindi deciso di provare a cambiare la trasformazione utilizzata per portare i cornerpoints in numeri complessi. In tutte le prove eseguite fino ad ora abbiamo sempre considerato la trasformazione T . Nelle prove seguenti andremo invece a considerare le trasformazioni NT_1 (3.2) e NT_2 (3.3) le cui caratteristiche sono state illustrate in dettaglio nel capitolo precedente.

Abbiamo ripetuto le prove considerando:

- RISOLUZIONE 200 per le immagini;
- cornerpoints CON CORNERLINE;
- TRASFORMAZIONE NT_1 per passare da cornerpoints a numeri complessi;
- SENZA DIVISIONE dei coefficienti complessi per i termini binomiali e SENZA trasformazione nuova ai coefficienti complessi.

E poi considerando:

- RISOLUZIONE 200 per le immagini;
- cornerpoints CON CORNERLINE;
- TRASFORMAZIONE NT_2 per passare da cornerpoints a numeri complessi;
- SENZA DIVISIONE dei coefficienti complessi per i termini binomiali e SENZA trasformazione nuova ai coefficienti complessi.

In questi due casi, dato che da tutte le prove fino ad ora eseguite risulta non esserci una grande dipendenza dei risultati ottenuti al variare del valore di k con $k \in \{5, 10, 20, 50\}$, abbiamo deciso di considerare un solo valore di k in particolare $k = 10$.

Anche in questi casi però non otteniamo differenze rilevanti nei risultati, cioè ci sono sempre alcuni valori dei risultati rispetto ad alcune funzioni filtranti più alti rispetto a quelli delle prove precedenti ed altri più bassi. In questi casi andando a guardare nello specifico i risultati potremmo dire che in generale questi sono peggiorati un po' rispetto a quelli delle prove precedenti, però non in modo così significativo.

Visto che anche con questa modifica non abbiamo ottenuto grandi differenze nei risultati siamo andati a vedere qualche valore dei coefficienti complessi calcolati relativi ai polinomi costruiti (utilizzando il nuovo algoritmo considerando risoluzione 200 per le immagini, cornerpoints e cornerlines, trasformazioni T , NT_1 o NT_2 per passare da cornerpoints a numeri complessi, senza divisione dei coefficienti complessi per i termini binomiali e senza trasformazione nuova ai coefficienti complessi) ed abbiamo osservato che alcuni di questi, tra i primi 10 relativi ai monomi di grado maggiore, riferiti ad una

certa immagine, data una certa funzione filtrante, hanno parte reale e parte immaginaria in valore assoluto di un ordine di grandezza maggiore o uguale di 10^3 mentre altri riferiti ad un'altra immagine però sempre relativi alla stessa funzione filtrante, hanno parte reale e parte immaginaria in valore assoluto di un ordine di grandezza minore o uguale di 10^{-3} , quindi con almeno 6 ordini di grandezza di differenza. Questo concettualmente non è una cosa positiva perché poi questi coefficienti andranno confrontati tra loro e se ci sono differenze di questo tipo tra di essi allora una volta effettuati tutti i confronti il risultato dipenderà praticamente solo dai confronti tra coefficienti di ordini di grandezza molto diversi perché tali differenze saranno molto più elevate rispetto a quelle tra coefficienti di ordini di grandezza simili e, anche se nella distanza utilizzata in questo caso, d_2 (3.4), la differenza tra i coefficienti j -esimi (cioè i coefficienti relativi ai monomi di grado $n-j$ dove n è il grado dei polinomi costruiti) viene divisa per j (in questi casi particolari $k = 10$ quindi $1 \leq j \leq 10$), una volta sommate tutte le differenze per calcolare la distanza tra i due polinomi considerati andranno a prevalere le differenze di un ordine di grandezza più elevato. Abbiamo quindi deciso di provare ad utilizzare le nuove trasformazioni NT_1 e NT_2 tornando però a considerare la divisione dei coefficienti complessi per i termini binomiali con la speranza che questa divisione vada ad attenuare questo fenomeno appena illustrato.

Abbiamo quindi ripetuto le prove considerando:

- RISOLUZIONE 200 per le immagini;
- cornerpoints CON CORNERLINE;
- TRASFORMAZIONE NT_1 per passare da cornerpoints a numeri complessi;
- CON DIVISIONE dei coefficienti complessi per i termini binomiali, quindi SENZA trasformazione nuova ai coefficienti complessi.

E poi considerando:

- RISOLUZIONE 200 per le immagini;
- cornerpoints CON CORNERLINE;
- TRASFORMAZIONE NT_2 per passare da cornerpoints a numeri complessi;

- CON DIVISIONE dei coefficienti complessi per i termini binomiali, quindi SENZA trasformazione nuova ai coefficienti complessi.

Osservando i risultati relativi a queste prove si può notare che sono presenti sia valori corrispondenti ad alcune funzioni filtranti che sono minori rispetto a quelli ottenuti dalle prove eseguite precedentemente considerando le stesse modifiche del nuovo algoritmo senza però calcolare la divisione dei coefficienti complessi per i termini binomiali, sia valori maggiori. Più in dettaglio si può osservare, in generale, un miglioramento nei valori relativi a sensibilità ed un peggioramento nei valori relativi a specificità e ad accuratezza globale. Osservando alcuni coefficienti complessi dei polinomi calcolati, tra i primi 10 relativi ai monomi di grado maggiore, si può notare che in alcuni casi considerando una specifica funzione filtrante certi hanno parte reale e parte immaginaria in valore assoluto di un ordine di grandezza maggiore o uguale di 10^{-23} mentre altri hanno parte reale e parte immaginaria in valore assoluto di un ordine di grandezza minore o uguale di 10^{-30} , quindi anche eseguendo tale divisione rimane una buona differenza tra i possibili ordini di grandezza dei coefficienti complessi dei polinomi che saranno poi confrontati.

Dopo tutte queste prove ed osservazioni abbiamo quindi voluto cercare di creare una trasformazione da applicare ai coefficienti complessi in modo da renderli maggiormente confrontabili tra loro, ovvero in modo da restringere il range di ordini di grandezza in cui questi si trovano, senza però andare a creare problemi nella struttura globale del nuovo algoritmo e cercando di continuare a considerare con maggiore importanza i coefficienti complessi relativi ai monomi di grado maggiore. In più, cosa più importante, abbiamo cercato di capire quale fosse la causa di questa differenza. Da precisare che con “questa differenza” intendiamo la differenza tra gli ordini di grandezza di coefficienti complessi relativi a monomi di grado simile tra loro perché il fatto che coefficienti relativi a monomi di grado molto diverso tra loro possano avere ordini di grandezza distanti non va ad influire sul problema del calcolo delle differenze spiegato precedentemente ed è completamente giustificato dalle formule di Viète. Dopo avere osservato diversi valori di coefficienti complessi relativi a diverse immagini, diverse funzioni filtranti tra quelle considerate ed a diverse prove tra quelle effettuate siamo giunti alla conclusione che una delle principali cause di questa differenza possa essere la dipendenza del calcolo di tali coefficienti dal numero di cornerpoints del diagramma di persistenza di ogni immagine

fissata una certa funzione filtrante. Questo perché, anche se per confrontare al meglio i polinomi, per ogni funzione filtrante, noi li portiamo ad avere lo stesso grado n dove n non è altro che il numero massimo di cornerpoints su tutti i diagrammi di persistenza relativi alle immagini del database rispetto alla funzione filtrante considerata, non è assolutamente detto che tutte le altre immagini abbiano un diagramma di persistenza relativo alla stessa funzione filtrante con un numero di cornerpoints vicino ad n . Ad esempio considerando il nuovo database *PH2* e la funzione filtrante *IL* alcune immagini hanno un numero di cornerpoints (considerando anche la cornerline trasformata) uguale a 47, 77 o 156 mentre altre uguale a 1444, 1247 o al massimo 1815. Questo ha molto peso quando si vanno ad applicare le formule di Viète poiché la sommatoria per calcolare il coefficiente j -esimo avrà $\binom{n}{j}$ addendi però di questi addendi molti potrebbero essere nulli se il diagramma di persistenza relativo all'immagine ed alla funzione filtrante analizzate in quel momento ha un numero di cornerpoints che è molto minore rispetto ad n . Inoltre, questo potrebbe spiegare anche il perché quando si va ad applicare la divisione dei coefficienti complessi per i termini binomiali questo non vada a diminuire di molto la differenza tra gli ordini di grandezza dei coefficienti complessi (più precisamente quando parliamo di ordini di grandezza in riferimento ai coefficienti complessi intendiamo sempre gli ordini di grandezza della parte reale e della parte immaginaria dei coefficienti complessi considerate in valore assoluto) ed in particolare il perché calcolando tale divisione si ottengono coefficienti complessi molto piccoli anche dell'ordine di 10^{-30} come osservato anche precedentemente (questi valori sono riferiti all'utilizzo del nuovo database *PH2*). Questo potrebbe succedere in quanto i termini binomiali per cui andiamo a dividere i coefficienti complessi di un polinomio di grado n sono $\binom{n}{j}$ con $j \in \{1, \dots, k\}$ però niente ci assicura che la sommatoria di prodotti con cui sono stati calcolati quei determinati coefficienti abbia $\binom{n}{j}$ (per il j -esimo coefficiente) addendi non nulli, questi potrebbero essere molto meno e quindi in questo caso si andrebbe ad ottenere un numero piccolo rispetto a quello per cui si divide dato dal coefficiente binomiale appena descritto.

Detto questo abbiamo quindi pensato ad una trasformazione da applicare ai coefficienti complessi che lavori con le radici j -esime con $j \in \{1, \dots, k\}$. In questo modo si dovrebbe riuscire a compensare la differenza del numero di termini non nulli nelle sommatorie di prodotti descritti dalle formule di Viète portando quindi tutti i coefficienti

ad avere ordini di grandezza simili ed a considerare con maggiore importanza i coefficienti relativi ai monomi di grado maggiore rispetto agli altri. Dopo avere applicato la radice j -esima ai coefficienti j -esimi abbiamo poi deciso di dividere per N , dove N è il numero di cornerpoints di ogni immagine rispetto alla funzione filtrante considerata, in modo da andare a togliere la dipendenza dal numero di cornerpoints di cui avevamo parlato precedentemente. Dal punto di vista formale abbiamo quindi deciso di applicare ai coefficienti complessi la trasformazione descritta in dettaglio nel capitolo precedente e riportata anche di seguito:

$$a_j \quad \longmapsto \quad \frac{|a_j|^{\frac{1}{j}}}{N} e^{i \arg(a_j)}, \quad j \in \{1, \dots, k\}$$

dove a_j è il coefficiente complesso j -esimo e, come già detto precedentemente, N è il numero di cornerpoints del diagramma di persistenza corrispondente al polinomio complesso considerato e k è il numero di coefficienti complessi che andiamo a considerare per poi effettuare i vari confronti tra i diversi polinomi. Inoltre, come detto anche nel capitolo precedente, quando applichiamo tale trasformazione ai coefficienti complessi andremo poi ad utilizzare come distanza tra i polinomi costruiti la distanza d_1 (3.4). Questo per non aumentare ulteriormente la differenza nell'importanza da attribuire ai coefficienti in base al grado del monomio relativo e per vedere meglio come lavora la nuova trasformazione.

Da osservare che, applicando questa nuova trasformazione ai coefficienti complessi senza considerare per un momento la divisione per il numero di cornerpoints, non sempre si va a diminuire il valore del modulo di tali coefficienti poiché se questi hanno modulo minore o uguale ad 1 applicando la radice j -esima ($j \in \{1, \dots, k\}$) esso non può che rimanere inalterato o aumentare. Quindi tale trasformazione potrebbe non garantire il fatto che noi vorremmo dare più importanza ai coefficienti relativi ai monomi di grado maggiore rispetto agli altri. Tuttavia questa trasformazione sembra rimanere molto valida al fine di "livellare" gli ordini di grandezza dei vari coefficienti complessi in quanto, appunto, applicando una radice j -esima ($j \in \{1, \dots, k\}$) ad un numero con modulo maggiore di 1 ne andiamo a diminuire il valore mentre applicando una stessa radice ad un numero in modulo minore di 1 il valore di questo aumenta, come detto anche precedentemente.

Abbiamo quindi deciso di ripetere le prove considerando:

- RISOLUZIONE 200 per le immagini;

- cornerpoints CON CORNERLINE;
- TRASFORMAZIONE T per passare da cornerpoints a numeri complessi;
- CON trasformazione nuova ai coefficienti complessi.

I risultati ottenuti, in termini di sensibilità, specificità ed accuratezza globale, sono riportati nella tabella seguente.

	$k = 10$		
	acc.	sens.	spec.
LUM	77.5	80	76.875
BLUE	76.5	70	78.125
GREEN	73.5	72.5	73.75
RED	74.5	77.5	73.75
EXCB	91	95	90
EXCG	81.5	87.5	80
EXCR	89.5	92.5	88.75
IL	80.5	70	83.125
LBORD	82	75	83.75
BORD	77.5	87.5	75
IBORD	77.5	85	75.625

Tabella 4.6: *Risultati polinomi nuovo database PH2, R200, CL, T, NTC.*

In questo caso, dato le considerazioni fatte in seguito alle prove precedenti sulla dipendenza dei risultati dal valore del parametro k considerato, abbiamo deciso di utilizzare un solo valore di k ed in particolare $k = 10$.

Osservando i risultati relativi a queste prove si può notare che questi sono dei buoni risultati e confrontabili con quelli ottenuti utilizzando l'algoritmo attuale (con distanza di bottleneck) su questo nuovo database *PH2*: l'algoritmo che ottiene risultati migliori sembra dipendere da quale funzione filtrante viene usata. Fino ad ora questi sono stati i migliori risultati ottenuti.

Inoltre, andando ad osservare alcuni coefficienti complessi dei polinomi costruiti, tra i primi 10 relativi ai monomi di grado maggiore, si può notare che coefficienti relativi a monomi dello stesso grado hanno parte reale e parte immaginaria che in valore assoluto sono di un ordine di grandezza compreso tra 10^{-3} e 10^{-5} , con prevalenza di 10^{-3} , o tra 10^{-2} e 10^{-3} quindi, almeno per quanto riguarda i coefficienti osservati, sembra che tale

trasformazione abbia realmente ridotto il range del possibili ordini di grandezza relativi ai coefficienti complessi che andiamo a considerare per poi calcolare le differenze.

A questo punto abbiamo deciso di ripetere le prove considerando le stesse modifiche della prova appena fatta tranne che per la trasformazione utilizzata per portare i cornerpoints in numeri complessi. Abbiamo deciso di provare a considerare la trasformazione NT_2 per vedere se e come variano i risultati che si ottengono.

Abbiamo quindi ripetuto le prove considerando:

- RISOLUZIONE 200 per le immagini;
- cornerpoints CON CORNERLINE;
- TRASFORMAZIONE NT_2 per passare da cornerpoints a numeri complessi;
- CON trasformazione nuova ai coefficienti complessi.

Per quanto riguarda il valore del parametro k , per motivi analoghi a quelli spiegati per la prova precedente, abbiamo deciso di considerare $k = 10$.

In questo caso però osservando i risultati ottenuti questi sono, in generale, peggiori rispetto a quelli appena riportati. Di conseguenza, almeno in questo caso, sembrerebbe funzionare meglio la trasformazione T rispetto alla trasformazione NT_2 per passare da cornerpoints a numeri complessi.

Dai risultati ottenuti nelle prove precedenti possiamo quindi notare che, almeno per quanto riguarda il nuovo database *PH2*, con le nuove trasformazioni NT_1 e NT_2 non si sono mai ottenuti, in generale, risultati migliori rispetto a quelli ottenuti utilizzando la trasformazione T e questo, da un punto di vista teorico, non era quello che ci saremmo aspettati dal modo in cui queste trasformazioni operano sui diversi cornerpoints. Come analizzato in dettaglio nel capitolo precedente le nuove trasformazioni dovrebbero trasformare i cornerpoints in modo più “ordinato” rispetto alla trasformazione T cercando di dare ancora meno importanza ai cornerpoints molto vicini alla “diagonale”. A questo punto, per cercare di capire il perché di questi risultati, siamo andati a costruire ed osservare alcuni dei diagrammi di persistenza relativi ad alcune immagini del database utilizzato, data una certa funzione filtrante, e i grafici relativi ai cornerpoints dei diagrammi di

persistenza considerati dopo avere applicato una delle trasformazioni sopra richiamate: prima utilizzando la trasformazione T e poi utilizzando le nuove trasformazioni NT_1 e NT_2 . Per poter commentare adeguatamente grafici di questo tipo in relazione alle prove effettuate ed ai risultati ottenuti sarebbe necessario analizzare i diagrammi di persistenza, prima e dopo avere applicato una delle trasformazioni considerate, relativi a tutte immagini del database utilizzato e questo per ogni funzione filtrante considerata. Dato che il nuovo database *PH2* contiene 200 immagini e dato che le funzioni filtranti utilizzate sono 11 abbiamo deciso di osservare tali diagrammi di persistenza, sempre prima e dopo avere applicato una delle trasformazioni considerate, per alcune immagini e per un paio di funzioni filtranti. Osservando tali grafici ci siamo accorti che non c'erano grandi differenze sul modo in cui agivano le diverse trasformazioni sui diagrammi di persistenza al variare dell'immagine considerata, data una certa funzione filtrante e quindi abbiamo deciso di riportare solamente alcuni di questi grafici: in particolare di seguito sono riportati prima i grafici relativi alle immagini *IMD435Size* e *IMD427Size* considerando la funzione filtrante *Bord* e poi i grafici relativi alle immagini *IMD398Size* e *IMD423Size* considerando la funzione filtrante *Luminosità*.

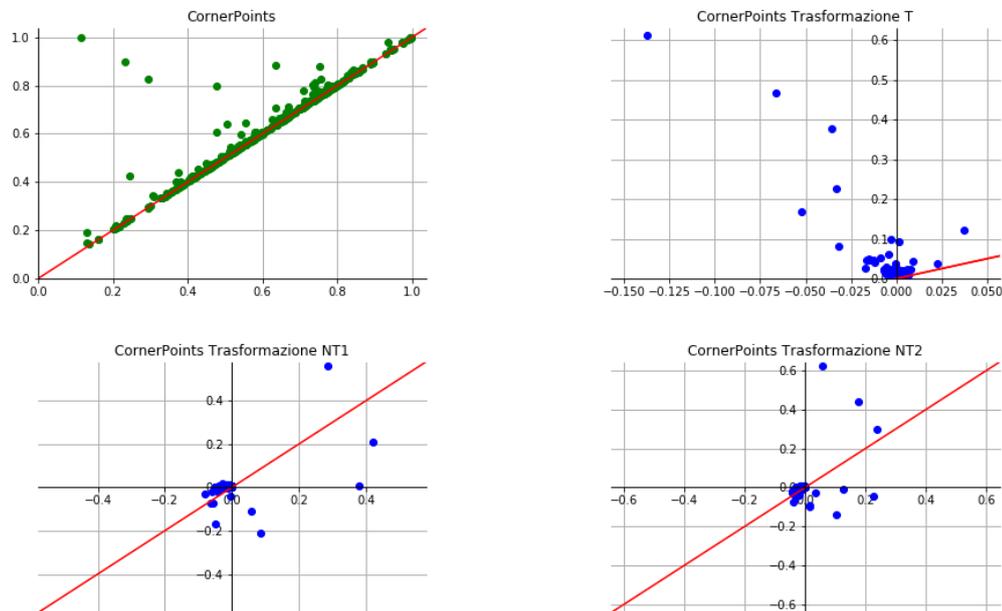


Figura 4.1: Database *PH2*, immagine *IMD435Size*, funzione filtrante *Bord*.

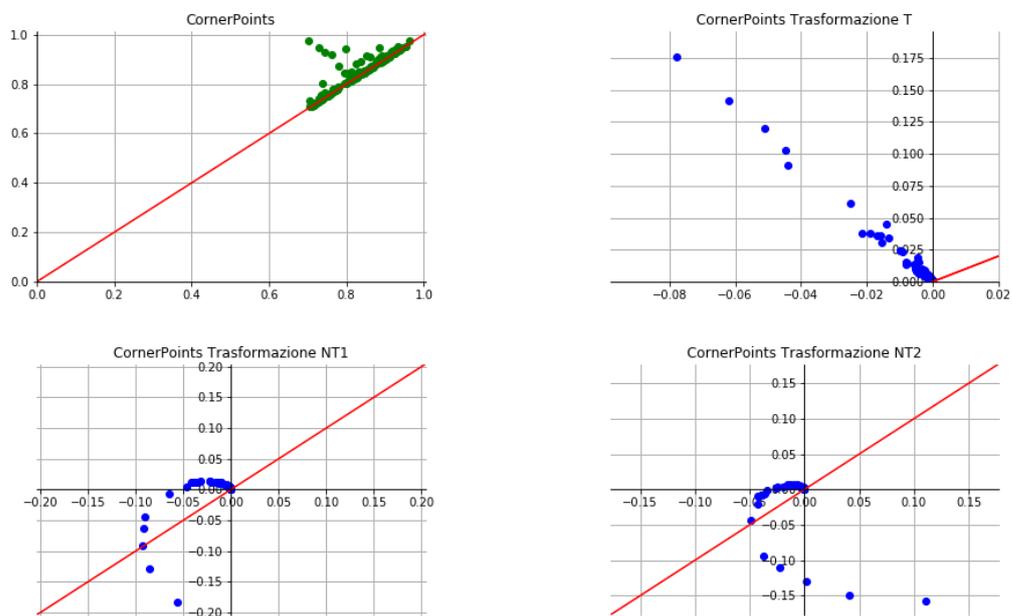


Figura 4.2: Database PH2, immagine IMD427Size, funzione filtrante Bord.

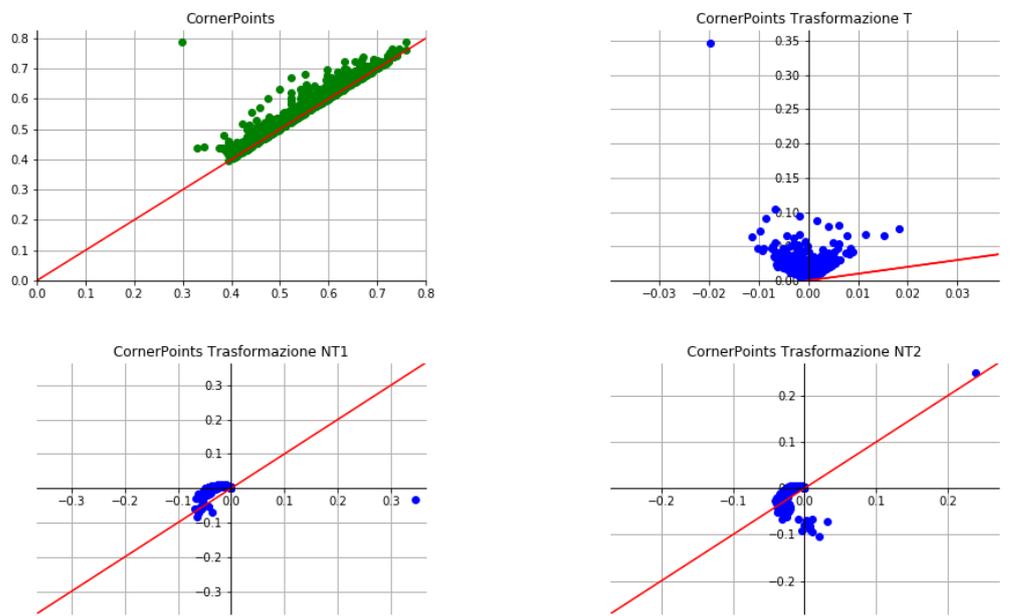


Figura 4.3: Database PH2, immagine IMD398Size, funzione filtrante Luminosità.

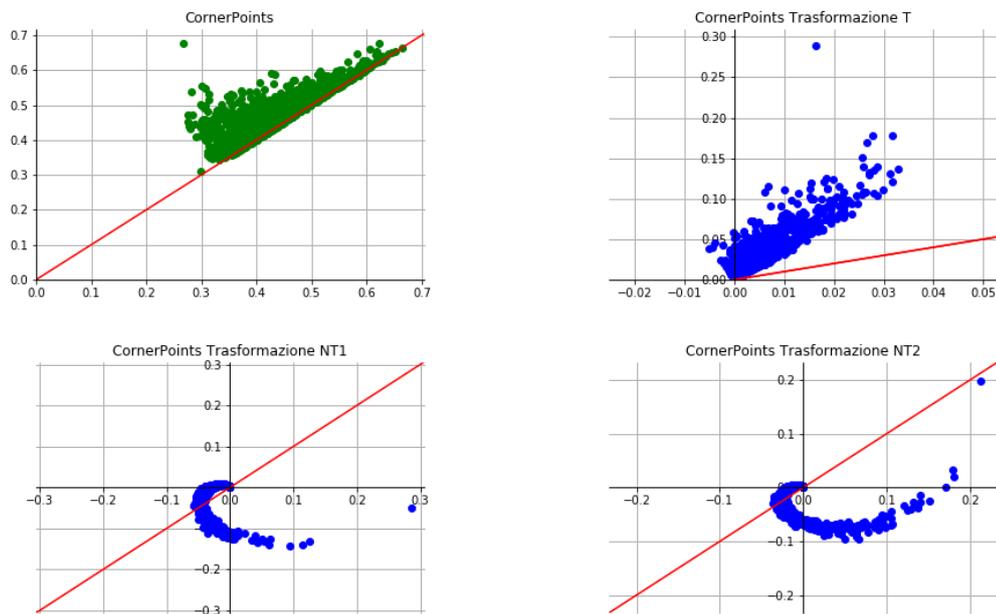


Figura 4.4: *Database PH2, immagine IMD423Size, funzione filtrante Luminosità.*

Dai grafici riportati possiamo osservare che, nei casi in cui siamo andati a costruire tali grafici, applicando la trasformazione T i cornerpoints non vengono mai portati in un insieme di punti che raggiunge più di due quadranti e sembra che, a differenza di quello che ci si poteva aspettare, questi non vengano “mescolati” tra loro anche se per affermare questo con certezza sarebbero opportune ulteriori analisi per capire in quale punto viene trasformato ogni singolo cornerpoint dei diagrammi di persistenza considerati. Questo sempre in relazione a questi casi particolari osservati poiché considerando la trasformazione T può succedere che l’insieme di punti in cui vengono trasformati i cornerpoints di un diagramma di persistenza raggiunga più di due quadranti o che l’angolo di rotazione a cui vengono sottoposti i cornerpoints sia maggiore di un angolo giro. In questi casi analizzati i valori delle coordinate iniziali dei cornerpoints saranno tali da non portare ad angoli di rotazione di tali ampiezze. Osservato questo, il fatto che con le nuove trasformazioni NT_1 e NT_2 si ottengano sempre risultati non migliori rispetto a quelli ottenuti con la trasformazione T potrebbe essere collegato ai cornerpoints molto vicini alla “diagonale”, quelli che vengono considerati come rumore. Una prima ipotesi potrebbe essere che in realtà sia importante considerare anche tali cornerpoints con mag-

giore rilievo e quindi, in questo caso, le nuove trasformazioni darebbero risultati peggiori perchè considerano poco tali punti in quanto li portano in punti molto vicini all'origine e disposti in maniera tale che applicando le formule di Viète questo potrebbe portare a considerare tali punti con minore importanza. Un'altra ipotesi, invece, potrebbe essere che le rotazioni effettuate nelle nuove trasformazioni portino, applicando poi le formule di Viète, a considerare con troppa poca importanza anche i cornerpoints più lontani dalla "diagonale" e quindi anche le informazioni dateci da tali punti. Quindi, per quanto detto, in questi casi specifici la trasformazione T potrebbe risultare più efficace rispetto alle altre perché potrebbe succedere che con i dati utilizzati questa non "mescoli" tra loro i cornerpoints e dia in questo modo il giusto peso sia ai cornerpoints considerati rumore che ai cornerpoints più lontani dalla "diagonale" e quindi più significativi per le immagini considerate.

Quindi rimane vero quanto osservato precedentemente cioè che i risultati relativi alle prove effettuate con la nuova trasformazione applicata ai coefficienti complessi unita alla trasformazione T applicata ai cornerpoints sono stati i migliori risultati ottenuti fino ad ora per quanto riguarda l'applicazione del nuovo algoritmo al nuovo database *PH2*. Da questo abbiamo deciso di provare ad applicare il nuovo algoritmo, tenendo conto di queste modifiche, al primo database per vedere se in questo modo si ottengono risultati migliori dato che con il primo database nessuna delle prove effettuate ci aveva restituito dei buoni risultati sulle singole funzioni filtranti.

4.3 Ulteriore prova con il primo database

Abbiamo quindi riutilizzato il primo database ed effettuato un'ulteriore prova considerando:

- RISOLUZIONE 200 per le immagini;
- cornerpoints CON CORNERLINE;
- TRASFORMAZIONE T per passare da cornerpoints a numeri complessi;
- CON trasformazione nuova ai coefficienti complessi.

Per quanto riguarda il valore del parametro k , analogamente a quanto fatto per le prove precedenti, abbiamo deciso di considerare $k = 10$.

I risultati ottenuti, in termini di sensibilità, specificità ed accuratezza globale, sono riportati nella tabella seguente.

	$k = 10$		
	acc.	sens.	spec.
LUM	61.694916	61.194031	61.84211
BLUE	59.322037	44.776119	63.596493
GREEN	57.288136	47.761192	60.087715
RED	62.033897	56.716419	63.596493
EXCB	61.35593	64.179108	60.526318
EXCG	59.661018	46.268658	63.596493
EXCR	59.661018	64.179108	58.333332
IL	63.389832	61.194031	64.035088
LBORD	62.711864	65.671638	61.84211
BORD	55.254238	46.268658	57.894737
IBORD	52.203388	43.283581	54.824562

Tabella 4.7: *Risultati polinomi primo database, R200, CL, T, NTC.*

Osservando i risultati riportati possiamo notare che in questo caso non abbiamo ottenuto dei buoni risultati. Andando a confrontarli con quelli riportati nel primo paragrafo di questo capitolo possiamo vedere che questi sono, solo per alcune funzioni filtranti, migliori rispetto a quelli ottenuti con le prime prove effettuate con il nuovo algoritmo anche se in modo non troppo significativo e, in generale, peggiori rispetto a quelli ottenuti con l'algoritmo attuale su questo database.

Quindi, anche in questo caso, continuiamo a non avere ottenuto dei buoni risultati relativi al primo database considerato. A questo punto bisognerebbe capire se con il primo database, a differenza del nuovo database *PH2*, non riusciamo ad ottenere dei buoni risultati a causa di alcune caratteristiche degli algoritmi che abbiamo utilizzato oppure se è dovuto al fatto che questo è un database in cui molti nei e melanomi sono simili tra loro e quindi un sistema computerizzato non riesce a cogliere le giuste differenze tra le diverse lesioni cutanee che si vanno ad analizzare.

4.4 Osservazione relativa ai costi computazionali degli algoritmi

Da quanto osservato, e per alcune specifiche prove anche riportato nella Tabella 4.5, i tempi computazionali di esecuzione del nuovo algoritmo sono minori rispetto a quelli dell'algoritmo attuale, una volta dato un database su cui lavorare. Andiamo quindi ad osservare i costi computazionali di questi due algoritmi per cercare di capire al meglio questa differenza. L'algoritmo attuale, il quale, data una coppia di immagini del database e considerata una certa funzione filtrante, utilizza la distanza di bottleneck per calcolare la distanza tra i diagrammi di persistenza relativi alle immagini ed alla funzione filtrante considerate, ha un costo computazionale pari a $O(N_c^{\frac{4}{3}} \log N_c)$ dove N_c rappresenta il numero massimo di cornerpoints dei diagrammi di persistenza tra quelli relativi alle immagini osservate data la funzione filtrante considerata. Inizialmente i primi algoritmi che implementavano la distanza di bottleneck tra due diagrammi di persistenza avevano un costo computazionale molto più elevato poi attraverso opportune osservazioni e modifiche alcuni ricercatori sono arrivati a scrivere algoritmi relativi a tale implementazione con un costo computazionale uguale a quello riportato [18]. Nel caso del nuovo algoritmo, invece, questo, data una coppia di immagini del database e considerata una certa funzione filtrante, utilizza una distanza tra i coefficienti dei polinomi complessi costruiti, come spiegato nel capitolo precedente, per calcolare la distanza tra i diagrammi di persistenza relativi alle immagini ed alla funzione filtrante considerate. Il costo computazionale di tale algoritmo dipende dal modo in cui vengono implementate le formule di Viète cioè le formule che calcolano i coefficienti di un certo polinomio date le rispettive radici. In questo caso abbiamo implementato le formule di Viète in modo da calcolare direttamente i primi k coefficienti relativi ai monomi di grado maggiore trascurando quello relativo al monomio di grado maggiore, in quanto, dato che lavoriamo con polinomi monici, questo è lo stesso per tutti i polinomi considerati e quindi non viene calcolato e restituito, e non in modo da calcolare tutti i coefficienti complessi e poi restituire i primi k , per evitare calcoli non necessari. Quando parliamo dei primi k coefficienti relativi ai monomi di grado maggiore di un dato polinomio intendiamo quindi, dato N_c il grado del polinomio considerato, i coefficienti relativi ai monomi di grado $N_c - 1, N_c - 2, \dots, N_c - k$. In

questo caso quindi il nuovo algoritmo ha un costo computazionale pari a $O(k(k + N_c))$ dove k rappresenta il numero di coefficienti calcolati e N_c rappresenta il numero massimo di cornerpoints dei diagrammi di persistenza tra quelli relativi alle immagini osservate data la funzione filtrante considerata. Per quanto riguarda i valori che può assumere il parametro k si ha $k \in \mathbb{N}$ e $1 \leq k \leq N_c$ anche se noi, come illustrato anche nel capitolo precedente, nel caso specifico di questo progetto abbiamo sempre utilizzato valori di k tali che $k \in \{5, 10, 20, 50\}$ i quali risultano soddisfare tutte le condizioni appena riportate. Da osservare che nelle prove effettuate spesso $k \ll N_c$ quindi in questo caso, se il valore di k risulta essere trascurabile rispetto al valore di N_c , si ha un costo computazionale pari a $O(N_c)$. Una volta scelte le immagini e la funzione filtrante da considerare il valore di k è costante ed in generale quando si effettua una prova utilizzando il nuovo algoritmo con le opportune modifiche si utilizza lo stesso valore di k per tutte le funzioni filtranti che si vanno a considerare.

I valori relativi ai tempi computazionali di esecuzione degli algoritmi considerati, ottenuti nelle prove effettuate precedentemente sono quindi in accordo con quanto scritto in questa osservazione. Sarebbe però opportuno fare un'ulteriore precisazione: quanto scritto in questa osservazione è riferito al caso dell'applicazione degli algoritmi utilizzati considerando due immagini, data una certa funzione filtrante, quindi al caso in cui gli algoritmi eseguono un unico confronto, ed in questo caso N_c , come scritto anche in precedenza, è il numero massimo di cornerpoints tra i diagrammi di persistenza relativi alle due immagini considerate, data la funzione filtrante, mentre, in realtà, nel caso dell'applicazione degli algoritmi utilizzati considerando più di due immagini (come può essere il caso in cui venga considerato un intero database), quindi nel caso in cui gli algoritmi eseguono più di un confronto, in realtà, per come sono stati pensati, c'è una piccola differenza sul modo in cui i due algoritmi considerano quello che chiamiamo N_c . Questo perchè, data una certa funzione filtrante, nel caso dell'algoritmo attuale N_c può cambiare ogni volta che cambia almeno una delle due immagini da confrontare e, dato un confronto, rimane quello descritto precedentemente quindi il numero massimo di cornerpoints tra i diagrammi di persistenza relativi alle due immagini considerate mentre, nel caso del nuovo algoritmo, N_c va ad assumere il numero massimo di cornerpoints tra i diagrammi di persistenza relativi a tutte le immagini considerate, quindi in questo

caso N_c è costante per ogni confronto da effettuare una volta fissato l'insieme delle immagini da considerare. Questo accade perchè, come spiegato in dettaglio nel capitolo precedente, una volta costruiti i polinomi complessi, ognuno riferito ad un diagramma di persistenza relativo ad una immagine considerata, data una certa funzione filtrante, questi vengono portati ad avere lo stesso grado, prima denominato con n , che non è altro che il numero massimo di cornerpoints tra i diagrammi di persistenza relativi a tutte le immagini considerate e quindi in questo caso nei diversi confronti vale sempre $N_c = n$. Questo potrebbe portare un piccolo cambiamento nei tempi di esecuzione del programma del nuovo algoritmo però, anche con tale cambiamento, i valori dei tempi computazionali del nuovo algoritmo rimangono comunque molto più bassi rispetto a quelli dell'algoritmo attuale, come osservato precedentemente. Questo poiché, anche se è vero che data una certa funzione filtrante il numero di cornerpoints può variare anche non di poco tra un'immagine ed un'altra, la differenza tra i costi computazionali osservati precedentemente è comunque maggiore rispetto alla differenza che può esserci tra i numeri di cornerpoints che stiamo considerando.

4.5 Risultati globali

In seguito alle prove eseguite fino ad ora abbiamo sempre analizzato ed in alcuni casi anche riportato i valori dei risultati, in termini di sensibilità, specificità ed accuratezza globale, per ogni singola funzione filtrante considerata. Andiamo ad osservare adesso, nei casi in cui già per le singole funzioni filtranti avevamo ottenuto dei buoni risultati, i risultati globali che si ottengono considerando il lavoro di tutte le 19 caratteristiche utilizzate dalla macchina per analizzare le diverse immagini: le 11 funzioni filtranti e gli 8 parametri standard. Non andiamo a calcolare i risultati globali relativi ai risultati singoli ottenuti con le altre prove effettuate in quanto: nel caso del primo database, questi, come osservato anche in precedenza, risultano essere peggiori rispetto ai risultati singoli ottenuti con l'algoritmo attuale e quindi non avrebbe molto significato andare a calcolare i relativi risultati globali; per quanto riguarda il nuovo database *PH2*, invece, ragionando in modo analogo andiamo a calcolare i risultati globali relativi alle prove che hanno dato i risultati singoli migliori (i quali avevamo osservato essere del tutto

paragonabili a quelli ottenuti con l'algoritmo attuale).

Prima di riportare i risultati ottenuti è importante dire che andremo ad osservare i risultati globali relativi a 10 retrieval diversi ottenuti modificando i diversi pesi relativi alle 19 caratteristiche analizzate da un retrieval all'altro. Ricordiamo che, infatti, le distanze globali vengono calcolate come media pesata delle singole distanze calcolate per ogni caratteristica considerata. I valori attribuiti a tali pesi possono essere dovuti a ragioni teoriche o sperimentali e possono anche essere cambiati in corso d'opera utilizzando un relevance feedback.

Per prima cosa riportiamo i risultati globali ottenuti utilizzando l'algoritmo attuale.

	acc.	sens.	spec.
R1	92	92.5	91.88
R2	91.5	92.5	91.25
R3	90.5	85	91.88
R4	92	87.5	93.13
R5	91.5	87.5	92.5
R6	95	92.5	95.63
R7	92.5	90	93.13
R8	94	90	95
R9	94	92.5	94.38
R10	96	95	96.25

Tabella 4.8: *Risultati globali bottleneck nuovo database PH2, R200, MV1.*

Riportiamo quindi di seguito i risultati globali relativi alla prova effettuata utilizzando il nuovo algoritmo e considerando:

- il nuovo database *PH2*;
- RISOLUZIONE 200 per le immagini;
- cornerpoints CON CORNERLINE;
- TRASFORMAZIONE T per passare da cornerpoints a numeri complessi;
- CON trasformazione nuova ai coefficienti complessi;
- $k = 10$.

	acc.	sens.	spec.
R1	90	80	92.5
R2	91	85	92.5
R3	91.5	82.5	93.75
R4	92.5	87.5	93.75
R5	92	85	93.75
R6	94.5	90	95.63
R7	93.5	87.5	95
R8	94.5	87.5	96.25
R9	97	97.5	96.88
R10	95.5	92.5	96.25

Tabella 4.9: *Risultati globali polinomi nuovo database PH2, R200, CL, T, NTC, MV1.*

Dai valori riportati possiamo osservare che questi risultano essere molto buoni e in questo caso quelli ottenuti con il nuovo algoritmo risultano essere, in generale, migliori in termini di specificità e peggiori in termini di sensibilità, anche se di poco, rispetto a quelli ottenuti con l'algoritmo attuale.

Da osservare, inoltre, che i risultati in termini di sensibilità, specificità ed accuratezza globale fino ad ora analizzati e riportati (sia quelli singoli che quelli globali), sono stati calcolati applicando un metodo standard (uno dei metodi illustrati nell'articolo [14]) i cui passi principali, considerata un'immagine del database utilizzato, sono i seguenti:

1. si utilizzano le seguenti notazioni: \mathcal{N} = numero totale dei nei nel database e \mathcal{M} = numero totale dei melanomi nel database;
2. si effettua un kNN con parametro k, in questo caso $k = 10$ (a noi interessano le 10 immagini più "vicine" a quella considerata esclusa tale immagine);
3. si conta il numero di melanomi recuperati nelle prime k posizioni e si denomina con m_r tale numero;
4. si dà poi un giudizio neo o melanoma all'immagine considerata nel seguente modo:
 - se l'immagine considerata è un neo, questa viene dichiarata neo quando

$$\frac{m_r}{k} < \frac{\mathcal{M}}{(\mathcal{N} + \mathcal{M} - 1)}$$

e melanoma altrimenti;

- se l'immagine considerata è un melanoma, questa viene dichiarata melanoma quando

$$\frac{m_r}{k} \geq \frac{(\mathcal{M} - 1)}{(\mathcal{N} + \mathcal{M} - 1)}$$

e neo altrimenti;

dove questa distinzione e i “-1” vengono unicamente dal fatto che l'immagine considerata appartiene al database e la dobbiamo togliere quando applichiamo tale metodo. Quello che viene dichiarato in relazione all'immagine dopo avere applicato tale metodo, infatti, non deve dipendere dalla diagnosi corretta dell'immagine considerata, perché nella realtà noi non conosciamo la classificazione vera dell'immagine considerata.

Questo viene poi effettuato per ogni immagine del database.

Quanto appena detto risulta essere importante poiché nel caso del nuovo database *PH2* considerato la situazione è la seguente:

- 200 immagini totali quindi $\mathcal{N} + \mathcal{M} = 200$;
- 40 melanomi quindi $\mathcal{M} = 40$;
- 160 nei quindi $\mathcal{N} = 160$.

Supponiamo ora di considerare un'immagine di tale database, di eseguire il kNN (con $k = 10$) e di ottenere $m_r = 2$, cioè tra le 10 immagini recuperate 2 sono melanomi ed 8 sono nei. Dato che, giustamente, andiamo a togliere dal database l'immagine della lesione cutanea di cui stiamo facendo il retrieval questo risulta essere un caso borderline cioè: applicando quanto scritto in precedenza, se l'immagine considerata è un neo con 8 nei su 10 nel retrieval dichiariamo “neo”, se invece l'immagine considerata è un melanoma dichiariamo “melanoma”, dando quindi un giudizio corretto in entrambi i casi. Tecnicamente applicando tale metodo non abbiamo commesso alcun errore in quanto stiamo lavorando con 199 immagini (in questo caso $\mathcal{N} + \mathcal{M} - 1 = 199$) di cui 40 melanomi in un caso e 39 nell'altro però potrebbe sembrare questo un metodo che favorisca in modo particolare il database che stiamo utilizzando.

Abbiamo quindi deciso, considerando sempre prima l'algoritmo attuale e poi il nuovo algoritmo con le particolari modifiche riportate precedentemente, di calcolare i risultati con un metodo che tiene conto della posizione, in relazione alla distanza, delle immagini recuperate dal kNN che si va ad effettuare rispetto all'immagine considerata (tale metodo viene introdotto nell'articolo [14] e, in particolare, questo viene illustrato ed utilizzato in [4]). Applicando questo metodo, quindi, alle 10 immagini recuperate dal kNN viene attribuito un valore di importanza che cresce al decrescere della posizione in cui si trovano rispetto alla "vicinanza" dall'immagine considerata.

Per quanto riguarda tale metodo i passi principali, considerata un'immagine del database utilizzato, sono i seguenti:

1. si utilizzano le seguenti notazioni: \mathcal{N} = numero totale dei nei nel database e \mathcal{M} = numero totale dei melanomi nel database;
2. si effettua un kNN con parametro k , in questo caso $k = 10$ (a noi interessano le 10 immagini più "vicine" a quella considerata esclusa tale immagine);
3. si riordinano le k immagini recuperate in ordine crescente rispetto alla distanza dall'immagine considerata e viene creato un vettore V di lunghezza $k+1$ in cui, indicata con V_i la componente i -esima del vettore V , si ha che
 - in V_i è contenuta la diagnosi corretta dell' i -esima immagine più "vicina" all'immagine considerata tra le k immagini recuperate dal kNN;
 - in V_0 è contenuta la diagnosi corretta relativa all'immagine considerata in quanto questa ha distanza 0 da sè stessa;

ed in particolare

$$V_i = \begin{cases} 0, & \text{se l'immagine corrispondente è relativa ad un neo;} \\ 1, & \text{se l'immagine corrispondente è relativa ad un melanoma;} \end{cases}$$

con $0 \leq i \leq k$;

4. si calcola

$$\lambda = \sum_{i=2}^{k+1} \sum_{j=2}^i V_j - \frac{(\mathcal{M} - V_0)k(k+1)}{2(\mathcal{N} + \mathcal{M} - 1)}$$

dove la somma in j viene fatta a partire dalla posizione $j = 2$ in quanto nella prima posizione di V (V_0) sarà sempre presente l'immagine considerata a distanza 0 da sè stessa;

5. si dà poi un giudizio neo o melanoma all'immagine considerata nel seguente modo:

- se l'immagine considerata è un neo, questa viene dichiarata neo quando $\lambda < 0$ e melanoma altrimenti;
- se l'immagine considerata è un melanoma, questa viene dichiarata melanoma quando $\lambda \geq 0$ e neo altrimenti;

dove questa distinzione viene unicamente dal fatto che l'immagine considerata appartiene al database e la dobbiamo togliere quando applichiamo tale metodo. Anche in questo caso quello che viene dichiarato in relazione all'immagine dopo avere applicato tale metodo, infatti, non deve dipendere dalla diagnosi corretta dell'immagine considerata, perché nella realtà noi non conosciamo la classificazione vera dell'immagine considerata.

Questo viene poi effettuato per ogni immagine del database.

Per prima cosa riportiamo i risultati globali ottenuti utilizzando l'algoritmo attuale.

	acc.	sens.	spec.
R1	89.5	85	90.63
R2	89.5	90	89.38
R3	90.5	85	91.88
R4	89.5	77.5	92.5
R5	90	85	91.25
R6	90	90	90
R7	91	82.5	93.13
R8	90.5	90	90.63
R9	91.5	87.5	92.5
R10	90.5	87.5	91.25

Tabella 4.10: *Risultati globali bottleneck nuovo database PH2, R200, MV2.*

Riportiamo poi di seguito i risultati globali relativi alla prova effettuata utilizzando il nuovo algoritmo e considerando:

- il nuovo database *PH2*;
- RISOLUZIONE 200 per le immagini;
- cornerpoints CON CORNERLINE;
- TRASFORMAZIONE *T* per passare da cornerpoints a numeri complessi;
- CON trasformazione nuova ai coefficienti complessi;
- $k = 10$.

	acc.	sens.	spec.
R1	88	82.5	89.38
R2	88.5	82.5	90
R3	87.5	82.5	88.75
R4	91	80	93.75
R5	90.5	87.5	91.25
R6	89	85	90
R7	87	80	88.75
R8	90.5	87.5	91.25
R9	92.5	90	93.13
R10	89	85	90

Tabella 4.11: Risultati globali polinomi nuovo database *PH2*, *R200*, *CL*, *T*, *NTC*, *MV2*.

Dai valori riportati possiamo osservare che anche in questo caso abbiamo ottenuto risultati globali buoni sia per l'algoritmo attuale che per la versione del nuovo algoritmo considerata. I valori ottenuti con i due algoritmi sono del tutto confrontabili e l'algoritmo che ottiene risultati migliori sembra dipendere da quale retrieval viene utilizzato.

Anche in questo caso siamo andati ad osservare i risultati globali relativi a 10 retrieval diversi. I pesi considerati nei vari retrieval variano da uno all'altro però i 10 retrieval considerati per questo metodo sono gli stessi considerati per il metodo precedente.

Rispetto ai risultati globali calcolati utilizzando il primo metodo spiegato precedentemente questi risultano essere peggiori e questo potrebbe essere dovuto a quanto detto

precedentemente. In ogni caso, come appena osservato, anche applicando il secondo metodo otteniamo buoni risultati e questo dà una maggiore conferma del fatto che la bontà dei risultati ottenuti con il primo metodo non era dovuta solamente a quanto succedeva applicando tale metodo al particolare database utilizzato.

4.6 Ulteriore confronto tra risultati globali

Riportiamo di seguito i migliori risultati, in termini di specificità e sensibilità, ottenuti su questo database *PH2* dal lavoro di un altro gruppo di ricerca. Tale gruppo ha ottenuto i seguenti risultati applicando una tecnica di ricostruzione 3D alle immagini di lesioni cutanee prese in considerazione [19]. In particolare, utilizzando il database *PH2* tali ricercatori hanno ottenuto i seguenti risultati:

- specificità = 97%;
- sensibilità = 96%.

Riportiamo, invece, nella seguente tabella i migliori risultati, in termini di specificità e sensibilità, ottenuti su questo database *PH2* dal lavoro fatto in questo progetto. Riportiamo i risultati migliori ottenuti applicando l' algoritmo attuale sia con il primo metodo utilizzato per calcolare i risultati globali che con il secondo metodo e analogamente per il nuovo algoritmo. Nel caso in cui non ci fosse un retrieval in cui sia specificità che sensibilità avessero risultati entrambi migliori rispetto agli altri abbiamo riportato i risultati di specificità e sensibilità corrispondenti al valore più alto di accuratezza globale.

	Primo Metodo di Valutazione		Secondo Metodo di Valutazione	
	algoritmo attuale	nuovo algoritmo	algoritmo attuale	nuovo algoritmo
specificità	96.25%	96.88%	92.5%	93.13%
sensibilità	95%	97.5%	87.5%	90%

Tabella 4.12: *Risultati globali migliori ottenuti nuovo database PH2, R200.*

Dai risultati riportati possiamo osservare che i nostri risultati sono in generale leggermente inferiori rispetto ai risultati dell'altro gruppo di ricerca. Tuttavia la differenza non è elevata e, nei casi in cui si utilizza il nuovo algoritmo, effettuate le opportune modifiche, i valori non scendono mai sotto al 90% quindi si possono considerare comunque dei buoni risultati.

Conclusioni e lavoro futuro

In questo progetto di tesi abbiamo modificato, in parte, l'algoritmo che attualmente è presente all'interno di una macchina per il supporto medico nell'individuazione di lesioni melanocitiche, dove il principio che sta alla base del funzionamento di tale macchina sfrutta principalmente la teoria dell'omologia persistente.

L'obiettivo è stato quello di cercare di ridurre il tempo di esecuzione del programma, al fine di riuscire a lavorare, in futuro, con database di maggiori dimensioni, e di cercare di tralasciare meno informazioni possibili dalle immagini che vengono acquisite e poi analizzate dalla macchina.

In particolare è stata presa in esame la parte di algoritmo in cui vengono calcolate le distanze tra le diverse immagini del database. Come è stato spiegato in precedenza, attualmente queste distanze vengono calcolate utilizzando la distanza di bottleneck tra i diagrammi di persistenza relativi rispettivamente alle immagini che si stanno considerando, data una certa funzione filtrante. Nel nuovo algoritmo, invece, vengono costruiti dei polinomi complessi relativi a tali diagrammi di persistenza e le distanze vengono calcolate come distanze tra i coefficienti di tali polinomi.

Abbiamo effettuato diverse prove utilizzando diverse versioni del nuovo algoritmo ottenute variando parametri ed apportando, o meno, ulteriori modifiche pensate al fine di migliorare ulteriormente tale algoritmo. Le prove sono state effettuate considerando due diversi database. I risultati ottenuti dalle diverse prove li abbiamo poi confrontati con i risultati ottenuti con l'algoritmo attuale dato lo stesso set di input. Con il primo database, applicando le diverse versioni del nuovo algoritmo, non abbiamo ottenuto in alcuna prova risultati migliori rispetto a quelli ottenuti utilizzando l'algoritmo attuale, i quali già non erano ottimi. Considerando, invece, il nuovo database ed applicando il

nuovo algoritmo con specifiche modifiche abbiamo ottenuto dei buoni risultati e del tutto paragonabili a quelli ottenuti utilizzando l'algoritmo attuale (l'algoritmo che ottiene risultati migliori in questa prova sembra dipendere da quale funzione filtrante viene usata), con l'ulteriore aspetto positivo che il tempo computazionale del nuovo algoritmo risulta essere notevolmente minore rispetto al tempo computazionale dell'algoritmo attuale.

Dai risultati ottenuti e dai confronti effettuati abbiamo ipotizzato che il fatto che i risultati ottenuti con il primo database siano, in generale, peggiori rispetto a quelli ottenuti con il nuovo database possa essere dovuto al fatto che nel nuovo database è presente una buona clusterizzazione delle due classi di lesioni cutanee considerate, mentre nel primo database tale clusterizzazione è praticamente assente. Questo può rendere il primo database meno "leggibile" da un sistema computerizzato rispetto al nuovo database: in questo caso un sistema computerizzato non riesce a dare un grande supporto all'analisi delle lesioni cutanee osservate. La cosa che però non è chiara, e sulla quale si potrebbe basare una possibile ricerca futura, è il perché, tralasciando il fatto che per entrambi gli algoritmi utilizzati i risultati ottenuti sul primo database sono peggiori rispetto ai risultati ottenuti sul nuovo database, nel caso del nuovo database il nuovo algoritmo riesce con opportune modifiche ad ottenere risultati del tutto paragonabili a quelli ottenuti dall'algoritmo attuale mentre, nel caso del primo database, anche considerando le stesse modifiche, il nuovo algoritmo ottiene risultati, in generale, sempre peggiori rispetto a quelli ottenuti con l'algoritmo attuale.

Un altro fatto su cui si potrebbe basare una ricerca futura è cercare di capire il motivo per cui, utilizzando il nuovo algoritmo, considerare le cornerlines, oltre ai cornerpoints, non porti ad un generale miglioramento nei risultati che si ottengono. Sarebbe opportuno capire se questo sia dovuto alla trasformazione che abbiamo utilizzato per potere considerare le cornerlines o più in generale al fatto che per considerarle abbiamo "travestito" le cornerlines da cornerpoints oppure se in generale in questo tipo di algoritmo le informazioni che potrebbero esserci date dalle cornerlines non porterebbero in ogni modo ad un miglioramento delle funzionalità del nuovo algoritmo. Dalla teoria, in questo caso, ci si aspettava un miglioramento nei risultati ottenuti in quanto le cornerlines possono contenere informazioni importanti riguardo alle relative immagini.

Un altro risultato che ci si poteva aspettare diverso da quello ottenuto, basandosi

unicamente sulla teoria, è il fatto che, considerando lo stesso set di input per il nuovo algoritmo e cambiando unicamente la trasformazione da applicare ai cornerpoints per portarli in numeri complessi, la trasformazione T dia, in generale, anche se con piccole differenze, risultati migliori rispetto alle nuove trasformazioni NT_1 e NT_2 . Dalle osservazioni teoriche fatte precedentemente emergeva il fatto che le nuove trasformazioni avrebbero trasformato i cornerpoints in modo più “ordinato” rispetto alla trasformazione T cercando di dare meno importanza ai cornerpoints molto vicini alla “diagonale”. Anche in questo caso sarebbe opportuno cercare il motivo di queste differenze. Questo potrebbe essere collegato ai cornerpoints molto vicini alla “diagonale”, quelli che vengono considerati come rumore. Un’ipotesi potrebbe essere che in realtà sia importante considerare anche tali cornerpoints con maggiore importanza e quindi, in questo caso, le nuove trasformazioni darebbero risultati peggiori perchè considerano poco tali punti. Un’altra ipotesi, invece, potrebbe essere che le rotazioni effettuate nelle nuove trasformazioni portino, applicando poi le formule di Viète, a considerare con troppo poca importanza le informazioni dateci dai cornerpoints più lontani dalla “diagonale”. La trasformazione T potrebbe risultare in questo particolare caso più efficace perchè potrebbe succedere che con i dati utilizzati tale trasformazione non vada a “mescolare” tra loro i cornerpoints, come invece potrebbe succedere con altri dati, e dia in questo modo il giusto peso sia ai cornerpoints considerati rumore che ai cornerpoints più lontani dalla “diagonale” e quindi più significativi per le immagini considerate. Anche in questo caso si potrebbero fare ulteriori prove per cercare di capire al meglio il perchè di queste differenze.

Abbiamo poi notato che all’aumentare del valore del parametro k , ovvero del numero di coefficienti complessi calcolati e poi utilizzati per calcolare le distanze singole tra i polinomi, considerata una certa funzione filtrante, i risultati non variano in modo considerevole. Questo potrebbe andare a favore dell’ipotesi secondo la quale, dato un polinomio, i coefficienti relativi ai monomi di grado maggiore siano quelli che contengono più informazioni relative al polinomio considerato.

Quindi possiamo concludere dicendo che con il nuovo database abbiamo ottenuto dei buoni risultati utilizzando il nuovo algoritmo, con le opportune modifiche, con un tempo computazionale minore rispetto al tempo computazionale dell’algoritmo attuale senza però perdere in “bontà” dei risultati.

Abbiamo poi fatto ulteriori osservazioni riguardo ai risultati ottenuti ed ai confronti effettuati tra questi sulle quali si potrebbero basare delle ricerche per un lavoro futuro con l'obiettivo di migliorare ulteriormente l'algoritmo presente all'interno della macchina considerata. Questo per cercare di migliorare il funzionamento di tale macchina per dare un maggiore supporto allo specialista durante l'analisi di lesioni cutanee.

Appendice A

Codice Python

Riportiamo di seguito una delle versioni utilizzate del codice *PolyParte1* e una del codice *PolyParte2*. In particolare riportiamo le versioni dei due codici scritte in modo tale che:

- in *PolyParte1* venga utilizzata la trasformazione T (3.1) per passare da cornerpoints a numeri complessi;
- in *PolyParte2* venga utilizzata la nuova trasformazione per i coefficienti complessi (3.7);
- il valore del parametro k utilizzato sia $k = 10$.

Abbiamo scelto di riportare queste versioni in quanto con tali modifiche apportate all'algoritmo attuale abbiamo ottenuto, utilizzando il secondo database *PH2*, i risultati migliori rispetto a quelli relativi alle diverse prove effettuate.

A.1 PolyParte1

```
import numpy as np
import matplotlib.pyplot as plt
from itertools import izip, chain
import glob
import os, sys
import errno
import time

# Questo programma prende in input la cartella con le liste di cornerpoints e applica ad ogni lista la
# trasformazione T che trasforma le coordinate di ogni punto da (x,y) a (xBM,yBM).
# Per avere tutti i file con lo stesso numero di punti, aggiunge punti di coordinate (0,0) ad ogni lista
# che ha meno righe di max_number_of_lines.
# Salva poi le nuove liste nella cartella CornerPointBM con gli stessi nomi.

tempo_iniziale = time.time()

path1='CornerPoint'

def files_in(path1):
    from os import listdir
    from os.path import isfile, join
    return [f for f in listdir(path1) if isfile(join(path1, f))]

def please_apply_1(this_fun, to_every_file, in_this_path1, and_put_the_output_file_in_this_new_directory=None):
    from os import mkdir, path
    directory = and_put_the_output_file_in_this_new_directory
    ret_value = []
    try:
        mkdir(directory)
    except Exception: pass
    for input_file in to_every_file:
        if (and_put_the_output_file_in_this_new_directory != None):
            with open(path.join(directory, input_file), "w") as to_output_file:
                on_this_file = open(path.join(in_this_path1, input_file), "r")
                this_fun(on_this_file, to_output_file)
                on_this_file.close()
                to_output_file.close()
        else:
            on_this_file = open(path.join(in_this_path1, input_file), "r")
            ret_value.append(this_fun(on_this_file))
            on_this_file.close()
    return ret_value

def count_lines(in_list):
    lines = 0
```

```
for line in in_list:
    lines += 1
return lines

max_number_of_lines = max(please_apply_1(count_lines, files_in(path1), path1))
print max_number_of_lines

def BM_function(in_list, out_file):
    BM_list=[]
    i=1
    for line in in_list:
        line = line.strip()
        columns = line.split()
        x = float(columns[0])
        y = float(columns[1])
        a = np.sqrt(x**2+y**2)
        xBM = (y-x)*(np.cos(a) - np.sin(a))/2
        yBM = (y-x)*(np.cos(a) + np.sin(a))/2
        BM_list.append(['p', i, xBM, yBM])
        out_file.write('p' + "\t" + str(i) + "\t" + str(xBM) + "\t" + str(yBM) + "\n")
        i=i+1
    cp_number=len(BM_list)
    if len(BM_list)<max_number_of_lines:
        for j in xrange(len(BM_list)+1, max_number_of_lines+1):
            BM_list.append(['p', j, 0, 0])
            out_file.write('p' + "\t" + str(j) + "\t" + str(0.0) + "\t" + str(0.0) + "\n")
    # come ultima riga del file c'è il numero di cornerpoints di quell'immagine
    out_file.write(str(cp_number)+"\n")
    return BM_list

please_apply_1(BM_function, files_in(path1), path1, 'CornerPoint_BM')

tempo_finale = time.time()
print "Impiegati", str(tempo_finale - tempo_iniziale), "secondi."
```

A.2 PolyParte2

```

from __future__ import division
import numpy as np
import math
import cmath
import matplotlib.pyplot as plt
from itertools import izip, chain, izip_longest
from decimal import *
import glob
import os, sys
import errno
import time

# Questo programma può essere lanciato solo dopo PolyParte1.py. Infatti prende in input l'output di
# PolyParte1.py, cioè la cartella con le liste di punti di coordinate (xBM,yBM), e nell'ordine gli
# applica le seguenti due funzioni.
# La prima è Vieta_function che prende ogni lista di punti BM, la trasforma in una lista di numeri
# complessi e a seconda del valore k scelto calcola la formula di Viète sulle liste e successivamente,
# attraverso la funzione new_trasformation, viene applicata la nuova trasformazione ai coefficienti
# complessi.
# Il primo output è una cartella (Coefficienti_CompleSSI) di liste di numeri complessi che sono
# i coefficienti di ogni polinomio. Da questa cartella vengono date in input alla seconda funzione
# coppie di liste per il calcolo della distanza tra vettori di coefficienti. Il tutto viene salvato
# sottoforma di matrice quadrata simmetrica nel file MatriceDistanze all'interno di una cartella Distanze
# e questo è il secondo output.

tempo_iniziale = time.time()

def files_in(path):
    from os import listdir
    from os.path import isfile, join
    if path == path2:
        return [f for f in listdir(path2) if isfile(join(path2, f))]
    if path == path3:
        return [f for f in listdir(path3) if isfile(join(path3, f))]

path2='CornerPoint_BM'

def please_apply_2(this_fun, to_every_file, in_this_path2, and_put_the_output_file_in_this_new_directory=None):
    from os import mkdir, path
    directory = and_put_the_output_file_in_this_new_directory
    ret_value = []
    try:
        mkdir(directory)
    except Exception:
        pass

```

```

for input_file in to_every_file:
    if (and_put_the_output_file_in_this_new_directory != None):
        with open(path.join(directory, input_file), "w") as to_output_file:
            on_this_file = open(path.join(in_this_path2, input_file), "r")
            this_fun(on_this_file, to_output_file)
            on_this_file.close()
            to_output_file.close()
    else:
        on_this_file = open(path.join(in_this_path2, input_file), "r")
        ret_value.append(this_fun(on_this_file))
        on_this_file.close()
return ret_value

def vieta_compute(l,k,cp_number):
    r=len(l)
    x=[1]
    for h in range(1,k+1):
        x=[a+b*1[h-1] for a,b,d,e in izip(chain([0],x), chain(x,[0]),range(h+1),range(h,-1,-1))]
        # print (x,a,b,d,e)
    y=x[0:k]
    for h in range(k+1,r+1):
        y=[a+b*1[h-1] for a,b,d,e in izip(y,chain(y[1:k],[1]),range(h-k,h),range(k,0,-1))]
        # print (y,a,b,d,e)
    y.reverse()
    y=new_trasformation(y,k,cp_number)
    return y

def new_trasformation(y,k,cp_number):
    for i in range(0,len(y)):
        t=y[i]
        n=float(cp_number)
        t1=(t.real**2+t.imag**2)**(1/(float(i)+2))
        t2=t1*(math.e**(complex(0,cmath.phase(t))))
        t3=t2/n
        y[i]=t3
    return y

def complex_list(l):
    c = [complex(i[0], i[1]) for i in l]
    return c

def Vieta_function(in_list, out_file):
    BM_Vector = []
    xBM = []
    yBM = []
    for line in in_list:
        line = line.strip()
        columns = line.split()

```

```

        if len(columns)==1:
            cp_number=columns[0]
        else:
            xBM.append(float(columns[2]))
            yBM.append(float(columns[3]))
    zBM = complex_list(izip(xBM, yBM))
    k = 10
    vBM = vieta_compute(zBM, k, cp_number)
    for i in vBM:
        out_file.write(str(i) + "\t")
        out_file.write('\n')
    return BM_Vector

please_apply_2(Vieta_function, files_in(path2), path2, 'Coefficienti_Complessi' )

path3 = 'Coefficienti_Complessi'

def please_apply_3(this_fun, to_every_file, in_this_path3, and_put_the_output_file_in_this_new_directory):
    from os import mkdir, path
    directory = and_put_the_output_file_in_this_new_directory
    try:
        mkdir(directory)
    except Exception: pass
    Distance_Matrix=np.zeros((len(to_every_file),len(to_every_file)))
    Enumerate_Matrix=[e for e in enumerate(to_every_file)]
    for (i, input_file1) in Enumerate_Matrix:
        for (j, input_file2) in Enumerate_Matrix[i:]:
            on_file1 = open(path.join(in_this_path3, input_file1), "r")
            on_file2 = open(path.join(in_this_path3, input_file2), "r")
            Distance_Matrix[i,j] = Distance_Matrix[j,i] = this_fun(on_file1, on_file2)
            on_file1.close()
            on_file2.close()
    to_output_file = open(path.join(directory, 'MatriceDistanze.txt'), 'w')
    for i in xrange(0,len(to_every_file)):
        for j in xrange(0,len(to_every_file)):
            to_output_file.write(str(Distance_Matrix[i,j]) + '\t' + '\t')
            to_output_file.write('\n')
    to_output_file.close()

def Distance_function(in_listA, in_listB, Out_file=None):
    i = 1
    DistanceVector=0
    for lineA, lineB in izip(in_listA, in_listB):
        DistanceLine = abs(complex(lineA) - complex(lineB))
        DistanceVector = DistanceLine + DistanceVector
        i += 1
    if Out_file != None : Out_file.write(str(DistanceVector))
    return DistanceVector

```

```
please_apply_3(Distance_function, files_in(path3), path3, 'Distanze')
```

```
tempo_finale = time.time()
```

```
print "Impiegati", str(tempo_finale - tempo_iniziale), "secondi."
```


Bibliografia

- [1] <http://www.epicentro.iss.it/problemi/melanoma/melanoma.asp>
- [2] Aiom *Linee guida MELANOMA*, Edizione 2016.
- [3] I. Stanganelli, *Il sole e la pelle. Dall'Emilia-Romagna un modello di ricerca e di intervento educativo*. Edizioni Minerva Medica (2015).
- [4] F. Foschi, *Valutazione di un sistema di recupero di immagini dermatologiche*, Tesi di Laurea Magistrale in Matematica, Bologna A. A. 2013/2014.
- [5] F. Di Dio, *Algoritmi di omologia persistente k -dimensionale per la diagnosi di lesioni melanocitiche*, Tesi di Laurea Magistrale in Matematica, Bologna A. A. 2010/2011.
- [6] H. Edelsbrunner, J. Harer, *Persistent Homology - a Survey*, Contemporary Mathematics **453** (2008), 257-282.
- [7] P. Frosini, C. Landi, *Size theory as a topological tool for computer vision*, Pattern Rec. and Image Analysis **9** (1999), 596-603.
- [8] M. d'Amico, P. Frosini, C. Landi, *Using matching distance in size theory: a survey*, International Journal of Imaging Systems and Technology **16** (2006), 154-161.
- [9] M. d'Amico, P. Frosini, C. Landi, *Natural pseudo-distance and optimal matching between reduced size functions*, Acta applicandae mathematicae **109** (2) (2010), 527-554.
- [10] M. Ferri, *Persistent topology for natural data analysis - A survey*, (2017). In: Holzinger A., Goebel R., Ferri M., Palade V. (eds) Towards Integrative Machine Learning and Knowledge Extraction, LNAI, **10344** (2017), 117-133.

- [11] M. Ferri, *L'incredibile ubiquità della topologia persistente*, <http://maddmaths.simai.eu/divulgazione/focus/incredibile-ubiquita-della-topologia-persistente/> (2015-2016).
- [12] B. Di Fabio, M. Ferri, *Comparing persistence diagrams through complex vectors*, Image Analysis and Processing - ICIAP 2015 Part I; V. Murino, E. Puppo Eds., LNCS 9279, Springer (2015), 294-305.
- [13] M. Guijarro, G. Pajaresb, I. Riomorosc, P. J. Herrerad, X. P. Burgos-Artizzue, A. Ribeiroe, *Automatic segmentation of relevant textures in agricultural images*, Computers and Electronics in Agriculture **75** (1) (2011), 75-83.
- [14] M. Ferri, I. Tomba, A. Visotti, I. Stanganelli, *A feasibility study for a persistent homology based k-Nearest Neighbor search algorithm in melanoma detection*, J. Math. Imaging Vis. **57** (2017), 324-339.
- [15] R. S. Patil, A. J. Agraval, *Content-based Image Retrieval System: A Survey*, Advanced in Computational Sciences and Technology, **10** (9) (2017), 2773-2788.
- [16] M. D'Amico, *A new optimal algorithm for computing size function of shapes*. Proc. CVPRIP Algorithms III, International Conference of Computer Vision, Pattern Recognition and Image Processing (2000), 107-110.
- [17] <https://www.fc.up.pt/addi/ph2%20database.html>
- [18] A. Efrat, A. Itai, M. J. Katz, *Geometry Helps in Bottleneck Matching and Related Problems*, Algorithmica **31** (1) (2001), 1-28.
- [19] T. Y. Satheesha, D. Satyanarayana, M. N. Giri Prasad, Kashyap D. Dhruve, *Melanoma Is Skin Deep: A 3D Reconstruction Technique for Computerized Dermoscopic Skin Lesion Classification*, IEEE Journal of Translational Engineering in Health and Medicine, **5** (2017), 1-17.

Ringraziamenti

Vorrei ringraziare il Prof. Massimo Ferri per avermi seguito in questo progetto di tesi, per avere risposto alle mie innumerevoli domande e il Dott. Ivan Tomba per avermi aiutato e per avere risposto anche lui a tutte le mie domande e dubbi. È stato molto interessante e penso che mi abbia insegnato molto riguardo al lavoro di ricerca in gruppo lavorare a questo progetto con persone come loro e li ringrazio molto per tutto questo. Vorrei poi ringraziare il Prof. Ignazio Stanganelli per la disponibilità nel rispondermi a domande di carattere medico.

Vorrei inoltre ringraziare tutti i professori che ho conosciuto in questi anni per tutte le conoscenze che mi hanno trasmesso.

Vorrei ringraziare la mia famiglia che è sempre stata al mio fianco, che mi ha sempre supportato ed incoraggiato anche nei momenti più difficili e che ha sempre festeggiato con me nei momenti più sereni e gratificanti.

Vorrei poi ringraziare tutti i miei compagni di viaggio, quelli che mi hanno sempre aiutato quando non capivo qualcosa o quando avevo dei dubbi, quelli che hanno condiviso con me i vari progetti e i lavori di gruppo, quelli che mi hanno ascoltato, supportato ed incoraggiato e tutti coloro che ci sono sempre stati per un confronto o anche solo per due chiacchiere quando arrivavo in dipartimento.

Vorrei poi ringraziare gli amici di sempre per il sostegno continuo che mi hanno dato e tutti gli amici con cui si riescono a condividere belle serate.

Vorrei poi ringraziare in modo particolare le persone che mi hanno supportato ed incoraggiato quando dicevo di essere indietro e di non riuscire a completare un lavoro in tempo perché, veramente, questa cosa direi che è successa diverse volte e queste persone, soprattutto alcune (e queste lo sanno), non mi hanno mai smesso di ascoltare, dare un consiglio o rispondere e mi sono sempre state vicine.

Grazie veramente a tutti voi!!!

Grazie 1000!!!