Machine Learning: principi di funzionamento e applicazioni in Medicina

Giorgio De Nunzio

Dipartimento di Matematica e Fisica "Ennio De Giorgi" - Università del Salento
Istituto Nazionale di Fisica Nucleare, Sezione di Lecce
giorgio.denunzio@unisalento.it

L'articolo espone le basi del funzionamento delle applicazioni mediche dell'Intelligenza Artificiale basate sul *Machine Learning* e illustra sinteticamente alcuni esempi dalla letteratura nei quali l'autore è o è stato coinvolto. L'approccio è essenzialmente divulgativo e pragmatico, con pochi dettagli matematici: la finalità è dare al lettore un'idea abbastanza completa della procedura tipica di realizzazione di un'applicazione di *Machine Learning* e delle difficoltà che si incontrano nel corso del lavoro.

Sistemi di Intelligenza Artificiale per la Medicina basati su *Machine Learning*

La prima parte dell'articolo descrive i sistemi di Intelligenza Artificiale per la Medicina, focalizzando in particolare l'attenzione sul *Machine Learning*. Per il *Deep Learning*, a parte una breve introduzione, si rimanda alla vasta letteratura disponibile.

Per un elenco degli acronimi di uso comune impiegati nell'articolo, si veda la lista in coda.

Intelligenza Artificiale e *pattern Recognition*

Il termine **Intelligenza Artificiale** (IA) è molto ampio e indica l'abilità di un sistema artificiale di possedere caratteristiche tipicamente umane quali il ragionamento, l'apprendimento, la pianificazione e la creatività. L'IA ha assunto forme diverse nel corso degli ultimi cinquant'anni, passando dai sistemi esperti (*software* di supporto diagnostico-decisionale costituiti da una base di conoscenza specifica, un motore inferenziale, e un'interfaccia per l'utente) a sistemi di *Pattern Recognition*, PR, ossia di riconoscimento di schemi o modelli, basati sull'apprendimento, conven-

zionali (*Machine Learning*, ML) o profondi (*Deep Learning*, DL).

Quest'articolo tratta dei sistemi di ML, dandone alcuni dettagli del funzionamento e della struttura, e accenna brevemente al DL. Utile lettura di complemento è [1].

IA in Medicina

In Medicina l'IA è finalizzata a supportare il medico nella diagnosi delle patologie, nelle scelte terapeutiche, nella prognosi, fornendo sostegno infermieristico e medico per il monitoraggio dei pazienti, assistendo durante gli interventi chirurgici per diminuire il rischio d'errore, e così via. I vantaggi di un sistema di IA sono soprattutto a favore del paziente e vanno nella direzione della Medicina personalizzata. Questa tiene conto delle differenze individuali in termini di genetica, microbioma, stile di vita, ambiente ecc. e si basa, quindi, sull'individuazione delle caratteristiche specifiche del singolo paziente, resa possibile dalla grande varietà di parametri fisiologici e patologici rilevabili grazie alle ormai disponibili tecnologie avanzate. Non sono tuttavia da trascurare le ricadute puramente economiche dell'IA sul sistema sanitario, in quanto la maggior accuratezza delle pratiche sanitarie può evitare la prescrizione di esami clinici inutili (favorendo l'appropriatezza) e riduce la probabilità di diagnosi e interventi erronei che costringano al ritorno del paziente nel percorso clinico.

Alcuni termini usuali nel campo dell'IA per la Medicina sono:

- **Radiomics** (Radiomica): termine introdotto di recente, ispirato da altre discipline `omiche`, ovvero genomica, proteomica o metabolomica; la radiomica è un approccio quantitativo all'*imaging* medicale (*Computed Tomography* o TAC, *Magnetic Resonance imaging* o Risonanza Magnetica, ...) che mira a incrementare e arricchire l'informazione presente nei dati mediante analisi matematiche avanzate. Attraverso il calcolo delle distribuzioni spaziali delle intensità del segnale e delle interrelazioni tra i *pixel*, la radiomica quantifica le informazioni presenti nell'immagine traendo indicazioni non rilevabili mediante la semplice osservazione visiva [2].
- **CADe** *(Computer-Assisted/Aided Detection)*: sistema software progettato per individuare automaticamente lesioni (ad esempio tumorali) nei tessuti e quindi avente il fine di ridurre le sviste (in particolare il tasso di falsi negativi, FN) nell'analisi visiva e nell'interpretazione delle immagini mediche (o di altro dato o segnale di interesse diagnostico). Il tipico *output* di un CADe è la posizione di potenziali lesioni (spesso tumori). Comprende solitamente una fase automatica di contornamento dei margini (ovvero segmentazione) molto sensibile (in grado cioé di individuare la totalità delle lesioni), seguita da un algoritmo di ML avente lo scopo di ridurre il numero di falsi positivi (FP). Il *focus* è sull'individuazione di lesioni.
- **CADx** *(Computer-Assisted/Aided Diagnosis)*: è progettato per diagnosticare e classificare lesioni note (quindi, ad esempio, per calcolare la probabilità che una lesione nota sia maligna). Può avere una fase di segmentazione automatica/manuale, seguita dal calcolo di *feature* radiomiche ed eventualmente da una fase di ML. Il *focus* è dunque sulla diagnosi.

Spesso l'acronimo CAD è adoperato indifferentemente per indicare i sistemi CADe e CADx, poiché le tecniche e gli algoritmi alla base sono comuni. `Radiomica` è un vocabolo di grande successo nella comunità medica e si rivolge specificamente all'analisi delle immagini mentre i CAD possono operare su dati di altro genere (come le serie temporali, ad esempio l'elettroencefalogramma o EEG). Un CAD prevede generalmente un sistema di classificazione di qualche tipo (quindi opera nel campo del ML), mentre un *software* radiomico può limitarsi al calcolo di determinati indicatori da adoperare in base a considerazioni statistiche elementari.

Pattern Recognition (PR)

Un sistema CAD è in grado di individuare *pattern*, ossia schemi comuni all'interno di dati diagnostici, e mettere in relazione tali schemi con una situazione di presenza o assenza di patologia (*detection*), o con una specifica caratterizzazione di una lesione (*diagnosis*). A ciascun *pattern*

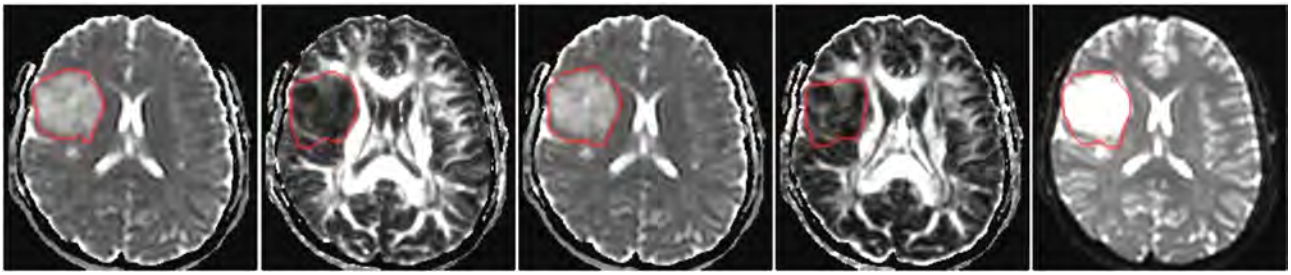

Figura 1: *Segmentazione automatica di gliomi cerebrali in immagini di Risonanza Magnetica in Tensore di Diffusione (risultati dell'autore).*

corrisponde una classe, ossia una popolazione di oggetti derivanti da osservazioni, aventi in comune delle caratteristiche di rilevanza clinica.

Formalmente, il riconoscimento di un *pattern* è un processo di assegnazione di una classe C (tessuto sano o patologico, lesione di un tipo istologico o di un altro...) a un'osservazione X (un *pixel*, un organo, una lesione). Attore principale del processo è un componente *software* denominato *classificatore*, del quale si parla ampiamente nel seguito.

Le due modalità più diffuse di *pattern recognition* sono:

- **PR supervisionato**: consiste in una tecnica di apprendimento automatico finalizzata a dedurre una funzione di classificazione da un insieme di addestramento nel quale siano state preliminarmente assegnate delle etichette (classi) che rappresentano la verità, in base alle competenze di un supervisore (generalmente un medico); ad esempio, in una classificazione binaria le classi possono essere: sano/malato, o benigno/maligno; in una classificazione multiclasse esse possono riguardare i tipi istologici di una lesione tumorale; una volta addestrato, il sistema è auspicabilmente in grado di riconoscere la classe di appartenenza di nuove osservazioni (generalizzazione);
- **PR non supervisionato**: individua e raggruppa, all'interno di un *dataset*, sottoinsiemi di osservazioni basandosi su misure di similarità; non è necessario definire a priori delle classi ed etichettare le osservazioni e anzi è proprio il classificatore che deve individuare le regolarità nei dati che permettano di dedurre come partizionare in classi il *dataset*.

In Medicina la maggioranza delle applicazioni adotta lo schema di PR supervisionato.

Oltre a stabilire se un tumore sia benigno o maligno, o se un paziente abbia o no una certa patologia, o se la prognosi sia fausta o infausta, la classificazione automatica ottenuta per apprendimento (supervisionato o non) ha anche una diversa applicazione: la `segmentazione` automatica nelle immagini diagnostiche.

La `segmentazione`, o contornamento, consiste nell'individuazione dei margini dei tessuti di interesse, quali una lesione tumorale o un organo (figura 1).

Lo scopo è per esempio la misura delle dimensioni della lesione o dell'organo (eventualmente seguendone l'evoluzione nel tempo), o il successivo calcolo di variabili radiomiche all'interno degli stessi, o ancora il contornamento di tessuti da colpire o da risparmiare nel corso del trattamento radioterapico (Piano di Trattamento Radioterapico).

La segmentazione può in realtà essere realizzata mediante procedure diverse dal PR (con tecniche più o meno avanzate di *image processing*) ma i metodi più complessi richiedono usualmente qualche meccanismo di classificazione, perché i *pixel* delle immagini sono classificati come appartenenti o non appartenenti all'organo o tessuto di interesse.

Accanto alla classificazione, i sistemi di IA possono essere finalizzati alla **regressione**. Questa affronta il problema di individuare una relazione funzionale incognita tra variabili misurate su un campione di interesse medico, di cui una o più sono considerate variabili indipendenti, e una o più sono variabili dipendenti: si tratta dunque di un'estensione del caso della classificazione: in quest'ultima la funzione da individuare ha valori discreti (le classi), mentre per la regressione il

valore può essere qualunque (una n-upla di numeri reali). La regressione è un tipo di apprendimento supervisionato: si parte da un insieme di addestramento di cui si conoscono i valori assunti sia dalle variabili indipendenti che da quelle dipendenti, e si allena il *software* a dedurne la relazione funzionale, per poi applicare il sistema a dati nuovi di cui non si conosca il valore delle variabili dipendenti.

Al fine di limitare l'estensione dell'articolo e fissare le idee, il *focus* (tranne saltuari accenni) sarà sulla classificazione con apprendimento supervisionato, in special modo sul ML, e si supporrà che il problema sia di classificazione binaria (variabili *target*: sano/malato, benigno/maligno, tipo istologico A o B...).

Inoltre, si considererà essenzialmente il caso del dato visuale bi- o tri-dimensionale (in particolare, immagini quali la radiografia, la risonanza magnetica, la tomografia computerizzata, l'ecografia, la PET...) tralasciando dati con un numero maggiore di dimensioni (spazio-temporali) e serie temporali (unidimensionali) come EEG e ECG.

Le applicazioni del ML alla Medicina condividono lo schema di sviluppo di seguito illustrato (figura 2) i cui dettagli saranno chiariti nei paragrafi successivi.

1. Il quesito clinico: per sistemi di classificazione, consiste nell'individuazione delle classi di interesse (ad esempio `lesione benigna` e `lesione maligna`); per problemi di regressione, è l'individuazione delle variabili cliniche da adoperare come *target*.
2. Acquisizione dei dati (immagini o serie temporali di biosegnali o variabili cliniche scalari) ed eventuale conversione tra formati.
3. Preprocessamento del dato, consistente nella standardizzazione di dati di origine disomogenea (se necessario), e nella pulizia dal rumore e da eventuali artefatti.
4. Eventuale *data augmentation* nel caso di *dataset* sbilanciati, eseguito a livello dei dati (immagini, serie temporali...).
5. Individuazione delle regioni di interesse da considerare (ROI, *Regions of Interest*): organi o lesioni nelle immagini, eventi salienti nelle serie temporali; eventuale approccio a finestra mobile.
6. Calcolo di variabili (dette indicatori, biomarcatori, *feature*...) nelle ROI, da scegliere tra agnostiche o semantiche, comuni o *domain specific*; eventuale *data augmentation* a livello delle *feature*.
7. Eventualmente, individuazione di una procedura di riduzione della dimensionalità dello spazio delle *feature*: selezione o estrazione.
8. Scelta di uno schema di apprendimento e validazione, con il partizionamento dei dati negli insiemi di *training*, *validation* e, possibilmente, *test*, ai fini della procedura denominata *hold-out cross validation*, o di altro schema quale il *k-fold cross validation*.
9. Scelta del modello di classificatore; ottimizzazione degli iperparametri con iterazione della procedura di allenamento e validazione, calcolando ad ogni *loop* delle appropriate figure di merito.
10. Verifica, sul *test set*, dell'avvenuta generalizzazione (ossia della capacità di fornire risposte corrette per dati non visti durante la procedura di apprendimento).
11. Trasferimento all'ambito clinico, se possibile.

La procedura è basata su un approccio iterativo che prevede di ripetere vari STEP a seconda del successo delle scelte via via operate, ottimizzando il sistema per la massima *performance*.

Lo STEP 1 è solo apparentemente semplice e banale: in realtà alle volte può non essere facile individuare una sola variabile *target* perché i medici stessi ne valutano contemporaneamente diverse, oppure usano misure fatte in momenti diversi della storia clinica del paziente, o ancora non hanno sempre la disponibilità delle variabili di interesse per l'intero campione considerato: in tal caso una discussione approfondita, allo scopo di identificare la scelta migliore, è delicata e fondamentale.

Lo STEP 2 spesso è un ostacolo non trascurabile, soprattutto nel caso delle immagini: il formato DICOM, estremamente dettagliato e personalizzabile, è uno *standard* inclusivo che permette di inserire in svariati modi, nei *file*, dati e metadati:

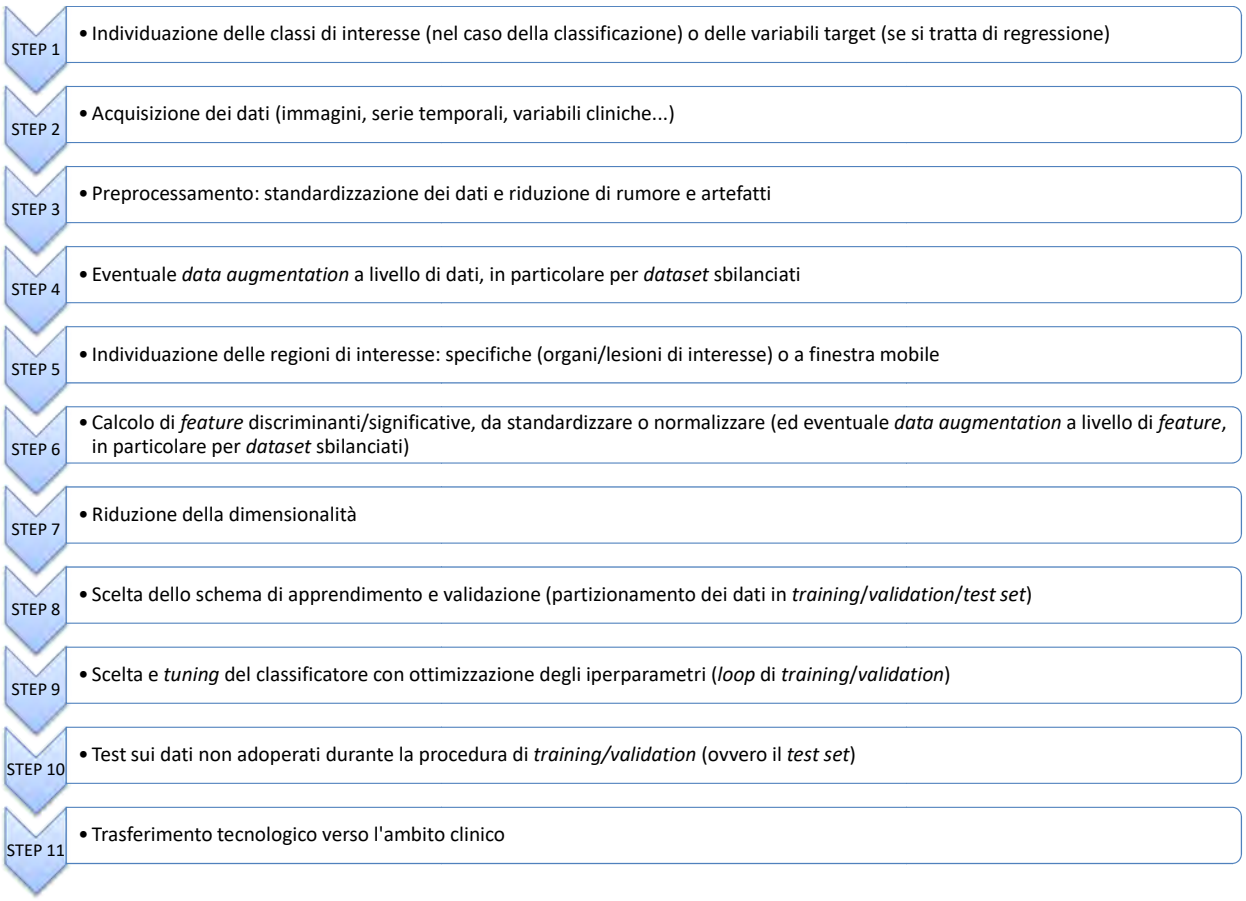

Figura 2: *Schema di sviluppo di un sistema di ML per la Medicina.*

i codici scritti per leggere e convertire le immagini in formati più agili devono spesso scontrarsi con questa versatilità, in particolare quando l'origine dei dati sia molteplice (studi multicentrici, che acquisiscano i dati da diversi presìdi ospedalieri).

Lo STEP 3 si basa su algoritmi e tecniche di trattamento di dati e immagini ed è anch'esso molto importante soprattutto nel caso di studi multicentrici: quando, infatti, i dati provengano da diverse fonti, sebbene teoricamente tra loro compatibili essi in realtà possono presentare differenze di *range* di valori acquisiti e nelle curve di calibrazione (e quindi nel significato fisico e clinico dei valori misurati, come avviene in special modo per le immagini di risonanza magnetica); nel caso delle immagini, un ulteriore possibile problema è la disparità di `voxel size` (dimensioni del volume acquisito corrispondente a un *pixel* nell'immagine) e l'eventuale presenza di `voxel` non cubici (non `isometrici` o `isotropici`); in tutti questi casi è necessario uniformare i dati applicando riscalature opportune e interpolare il dato spazialmente o temporalmente. Il preprocessamento consiste, dunque, usualmente in: normalizzazione delle intensità, omogenizzazione delle immagini per `voxel size` con eventuale isometrizzazione del `voxel` (interpolazione per portarsi a dimensioni standard quali $1\,mm \times 1\,mm \times 1\,mm$, figura 3), riduzione del rumore e degli artefatti.

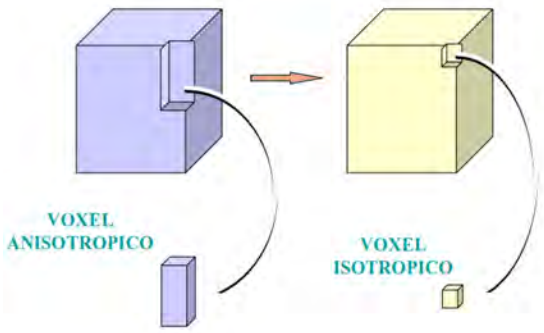

Figura 3: *Differenza tra `voxel` anisotropico e `voxel` isotropico.*

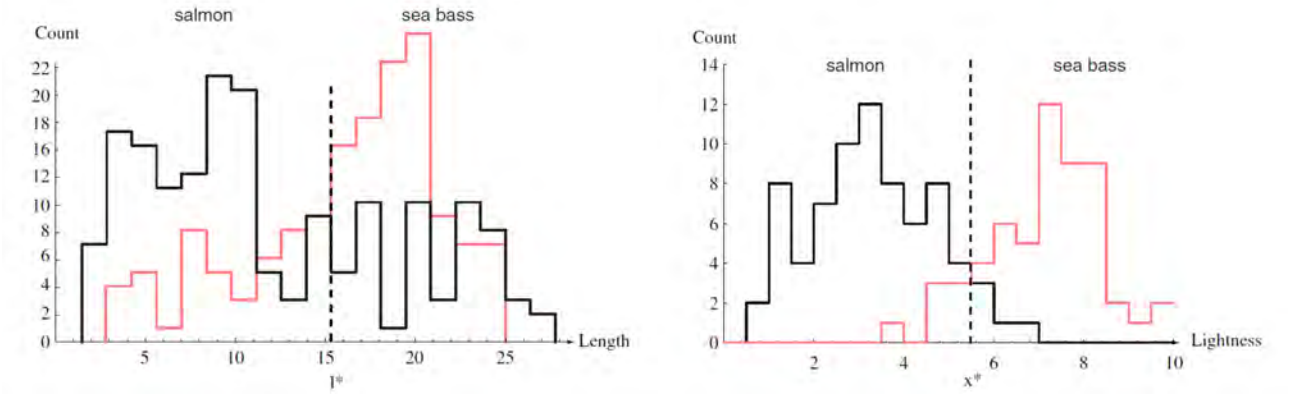

Figura 4: *A sinistra, istogrammi della lunghezza dei pesci (per specie); a destra, istogrammi della lucentezza delle scaglie. Gli istogrammi sono calcolati nel training set. Da [3].*

Lo STEP 10 può essere, per applicazioni mediche, arduo da eseguire in quanto spesso i *dataset* disponibili non sono tanto ampi da garantire la presenza di dati non sfruttati per *training* o *validation*, sui quali testare il sistema ormai ottimizzato. Dell'importanza di questo argomento si parlerà tuttavia nel paragrafo dedicato al punto 8.

Lo STEP 11 meriterebbe un trattamento esteso e quindi non sarà discusso.

Ove figurassero, accanto a dati complessi e strutturati come immagini e serie temporali, anche semplici variabili cliniche scalari, esse andranno direttamente ad arricchire il *set* di *feature* da adoperare nei calcoli di ML: in tal caso, naturalmente, alcuni degli STEP indicati perdono di significato, in particolare l'individuazione delle regioni di interesse (STEP 5) e il calcolo di *feature* al loro interno (STEP 6).

I prossimi paragrafi approfondiscono lo schema indicato, in particolare i punti da 4 a 9. Allo scopo, però, di dare un'idea immediata del funzionamento di un sistema di ML, conviene preliminarmente mostrare un esempio non specifico delle applicazioni alla Medicina, bensì estratto da un ben noto testo didattico [3] al quale si rimanda per approfondimenti. In tal modo sarà più facile in seguito comprendere i dettagli più tecnici.

Come funziona il *Machine Learning* supervisionato: un esempio

I gestori di un impianto di confezionamento di pesce desiderano automatizzare il processo di smistamento in entrata in base alla specie, in particolare distinguendo il branzino dal salmone, utilizzando il rilevamento ottico e metodi matematici [3]. Possono essere esplorate diverse misure che promettono di essere discriminanti tra i due tipi di pesci, ad esempio la lunghezza o la larghezza, la lucentezza della pelle (dovuta alla riflettanza delle scaglie), misure legate alla posizione della bocca, e così via. Il fine ultimo è elaborare una regola decisionale di classificazione che riduca al minimo il costo della procedura, a sua volta dipendente anche dal numero di errori compiuti.

È possibile immaginare una procedura basata su tre fasi: (1) preprocessamento e segmentazione (i fotogrammi registrati dal rilevatore ottico, raffiguranti il nastro trasportatore e i pesci, sono elaborati in modo da isolare i pesci l'uno dall'altro e dallo sfondo), (2) estrazione di *feature* (le immagini dei singoli pesci sono inviate a un sistema di misura, che rileva/calcola le proprietà desiderate), e (3) classificazione (i valori delle *feature* sono passati a un classificatore che valuta le informazioni, deduce il legame esistente tra i valori delle *feature* e la classe alla quale ciascun pesce appartiene, e costruisce un modello per discriminare tra le specie). Si tratta evidentemente di un approccio supervisionato perché, nella fase di creazione del modello di classificazione, a ciascun pesce è associata l'informazione sulla classe alla quale esso appartiene; il modello è costruito su una parte dei dati disponibili (*training set*) e successivamente validato su un'altra parte (*validation set*).

Per la costruzione del modello, iniziamo con-

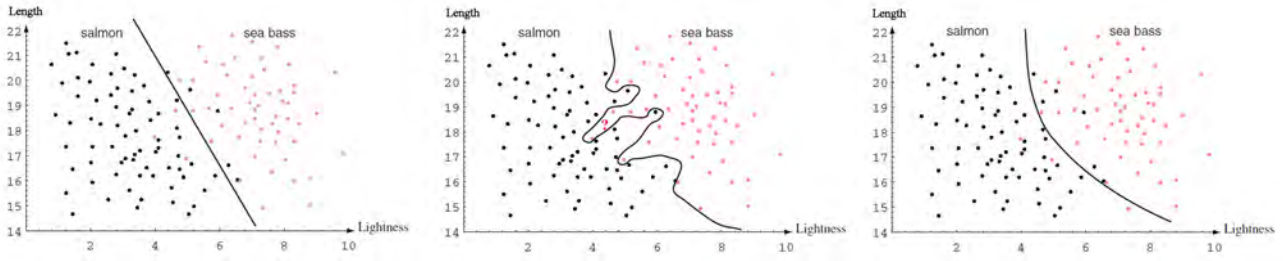

Figura 5: *Scatter plot delle due feature considerate, lunghezza dei pesci e lucentezza delle loro scaglie (per specie) (da [3]). Da sinistra verso destra, il confine di decisione è una retta, una linea curva complessa, una linea curva dolce. Gli scatter plot sono calcolati nel training set.*

siderando una singola *feature*, ad esempio la lunghezza del pesce. La conoscenza del dominio applicativo consente di affermare che un branzino è generalmente più lungo di un salmone (il branzino ha una lunghezza tipica, e questa è maggiore della lunghezza tipica del salmone). Dal punto di vista matematico, costruiamo gli istogrammi (per specie) della lunghezza dei pesci che abbiamo selezionato per il *training set* (figura 4, a sinistra) e osserviamo se i due istogrammi appaiono sufficientemente separati.

È facile immaginare un sistema elementare di decisione (una regola) basato sull'applicazione di una soglia: se la lunghezza del pesce è maggiore o uguale a un valore opportuno l^*, allora il pesce è un branzino, altrimenti è un salmone. Il valore di soglia va ottimizzato in maniera da minimizzare gli errori di classificazione. Esaminando gli istogrammi, però, ci rendiamo conto che il sistema non è ottimale perché, qualunque sia la soglia scelta, molti pesci saranno classificati in maniera erronea.

Si può ripetere il discorso su altre *feature*, ad esempio sulla lucentezza del manto di scaglie (figura 4, a destra). Le conclusioni saranno analoghe: tale *feature* da sola non basta. Possiamo allora pensare di considerare le due misure insieme, rappresentando ciascun pesce come una coppia (o vettore) di *feature*: $x = (x_1, x_2)$ dove x_1 e x_2 sono la lunghezza e la lucentezza. Poiché essa può essere vista come una coppia di coordinate che definiscono un punto nello spazio delle *feature*, ciascun oggetto (un pesce) sarà di fatto modellizzato in tal modo. Di conseguenza, classificare, ossia distinguere branzini da salmoni, sarà equivalente a tracciare nello spazio delle *feature* un confine (una linea non necessariamente retta) che separi i punti appartenenti alle due specie.

In base alla modellizzazione bidimensionale dei pesci, lo spazio delle *feature* può essere rappresentato come in figura 5, dove la linea di confine tra le classi (da sinistra a destra) è rispettivamente una retta, una curva che separi perfettamente le due classi, e infine una curva dolce in qualche modo intermedia tra i casi precedenti. La prima situazione darà un numero di errori non indifferente; la seconda non è realistica e mostra il cosiddetto *overfitting* ai dati di *training*, ovvero un eccessivo adattamento a questi dati durante il processo di apprendimento, come illustrato in figura 6; infine, la terza soluzione, pur dando luogo a un certo numero di errori risulta il miglior compromesso tra accuratezza sul *training set* e capacità di generalizzare a nuovi punti (appartenenti per esempio al *validation set*).

Compito del classificatore è trovare, a partire dai dati di *training*, la linea di separazione ottimale. Qualora tale linea sia una retta, il problema è detto `linearmente separabile`.

Generalizzando, possiamo utilizzare più di due variabili (in generale una n-pla), dimodoché un pesce sia modellizzato da un vettore (o da un punto) in uno spazio a n dimensioni e la linea di separazione tra i domini delle due classi diventi una (iper)superficie. Si osservi che alcune *feature* potrebbero essere ridondanti (a causa della correlazione tra variabili) o computazionalmente costose da ottenere, e può capitare che l'aumento eccessivo della dimensionalità possa diminuire le prestazioni. In conclusione, l'apprendimento supervisionato, tipico del ML, necessita di un *set* di dati di apprendimento (*learning set*) annotato (ovvero, etichettato). Il *dataset* si presenta usualmente sotto la forma di una matrice (figura 7) in cui le righe (più raramente, le colonne) sono

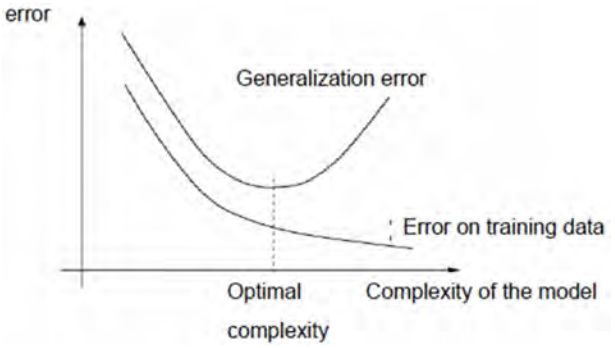

Figura 6: *Un modello che si adatti eccessivamente ai dati di training può causare overfitting, a scapito della capacità di generalizzare. La figura mostra qualitativamente come l'aumento di complessità del modello porti a una diminuzione dell'errore di classificazione sul set di training, ma un peggioramento della capacità di generalizzare (aumento dell'errore su validation set e test set)*

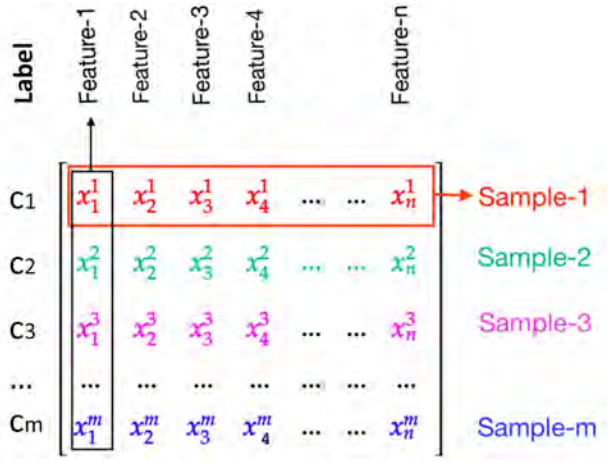

Figura 7: *Esempio di matrice di feature, con i vettori disposti per righe.*

vettori di *feature*: ciascuna riga contiene i dati di un oggetto (un campione) del *dataset* e ciascuna colonna corrisponde a una *feature*; una colonna, inoltre, spesso la prima o l'ultima, contiene le etichette, ossia la classe di appartenenza di ciascun campione: in caso di classificazione binaria di un problema diagnostico medico, detta colonna potrebbe contenere ad es. 0 o 1, rispettivamente per normale e patologico. L'etichetta è assegnata da un esperto (il supervisore) e contiene la verità o comunque lo *standard* di riferimento (*ground truth*, *golden truth*, o *gold standard*). Il classificatore ha il compito di realizzare la separazione ottimale tra i campioni nello spazio delle *feature*, come giusto compromesso tra la separazione nel *training set* e la capacità di generalizzare tale separazione a dati nuovi.

Una volta comprese le basi del funzionamento di un sistema di ML, conviene entrare nei dettagli di alcuni degli STEP della procedura introdotta con la figura 2.

STEP 4: *Dataset* sbilanciati e *Data Augmentation*

I *dataset* sbilanciati, ossia quelli nei quali una delle classi (consideriamo una classificazione binaria) è considerevolmente meno rappresentata dell'altra, sono prevalenti in una moltitudine di campi e settori, come i servizi finanziari (transazioni fraudolente vs genuine) e la diagnostica medica. Ad esempio, le lesioni maligne sono meno diffuse di quelle benigne e i pazienti malati sono in generale meno delle persone sane. Il ML in caso di *dataset* sbilanciati è particolarmente delicato perché gli algoritmi devono identificare casi rari in grandi insiemi di dati. A causa della disparità delle classi, l'algoritmo tende a categorizzare anche i casi rari nella classe maggioritaria, venendo meno al suo compito primario che è proprio l'individuazione di istanze della classe minoritaria. Inoltre molte misure di qualità tendono ad avere valori elevati nonostante il sistema di fatto non funzioni, dando la falsa sensazione di un modello altamente accurato.

Vi sono due diversi approcci al problema, di cui il primo è il più adatto per insiemi di dati poco numerosi.

- ***Data augmentation o oversampling (aumento o sovracampionamento dei dati)***. Si incrementa la numerosità della classe di minoranza aggiungendo copie leggermente modificate di dati già esistenti. Esempi: trasformazioni geometriche delle immagini come le rotazioni, aggiunta di rumore, o cancellazione casuale di parti dell'immagine. Un'alternativa è il sovracampionamento dei vettori di *feature* dopo il loro calcolo (quindi nello STEP 6) e un algoritmo noto è SMOTE (*Synthetic Minority Over-sampling Technique*). I nuovi vettori di *feature* sono da adoperarsi esclusivamente nel *training set*.
- ***Undersampling (sottocampionamento)***. Al contrario del precedente, ha lo scopo di ridurre la numerosità della classe maggioritaria, diminuendo quindi la quantità di dati

effettivamente adoperati, in modo che sia paragonabile a quella della classe di minoranza. È ottenuta selezionando casualmente una parte delle osservazioni della classe di maggioranza. È possibile effettuare un sottocampionamento `informato`, osservando la distribuzione dei dati della classe di maggioranza e selezionando opportunamente i campioni da includere. Un modo è l'uso preliminare di una tecnica di *clustering* nell'insieme maggioritario, che individui la stratificazione naturale dei dati. A questo punto, la scelta dei campioni da usare sarà fatta casualmente ma rispettando la proporzione tra i diversi cluster, mantenendo così la distribuzione originale. Anche in questo caso il bilanciamento può avvenire nello STEP 6.

In caso di *dataset* sbilanciati, ove non si intervenga con i metodi indicati, è opportuno valutare la qualità dei sistemi di classificazione in maniera specifica, ossia evitando di utilizzare la curva ROC che non è adatta a problemi affetti da squilibrio tra le classi, e adoperando piuttosto la curva Precision-Recall. Non usare, inoltre, l'accuratezza come cifra di merito (per i dettagli sui metodi di misura della qualità del sistema di ML, si veda il paragrafo corrispondente in quest'articolo).

STEP 5: Individuazione di regioni di interesse nelle immagini

Questo paragrafo è dedicato specificamente ad applicazioni del ML in immagini diagnostiche. In caso i sistemi di ML siano invece rivolti a dati come serie temporali EEG o ECG, sarà necessario adattare opportunamente il discorso, pur restando immutati i concetti di base.

Come visto nell'esempio della discriminazione tra branzini e salmoni, i calcoli delle *feature* (le variabili discriminanti in grado di differenziare tra oggetti appartenenti a classi diverse) sono condotti localmente, in `regioni di interesse`, o ROI. Le ROI possono essere, nelle applicazioni ad immagini diagnostiche mediche:

- lesioni o organi: la variabile *target* di classificazione potrebbe essere la malignità o non malignità di una lesione tumorale conclamata, o la presenza di una particolare patologia in un organo; simile è il caso in cui vi siano diverse regioni sospette (`candidati`) di cui non si conosca e si voglia stabilire la natura: il *target* potrebbe essere allora la natura eventualmente nodulare di regioni composte da *pixel* ipo- o iper-intensi (perché le caratteristiche della patologia, e la fisica sulla quale si basa la tecnica di *imaging*, garantiscono che i noduli cercati abbiano questa caratteristica); in generale, le ROI possono essere delimitate manualmente da un medico, oppure automaticamente con tecniche di processamento di immagini;
- l'interno di una piccola finestra che scansioni sistematicamente l'immagine, eventualmente con sovrapposizione parziale da una posizione all'altra: è l'approccio a `finestra mobile` (o *sliding window*) in cui la classe è attribuita come etichetta al *pixel* centrale della finestra; in fase di validazione, una volta terminata l'esplorazione esaustiva del tessuto e l'etichettatura dei *pixel* in base alla classe, l'accorpamento dei *pixel* equivalenti individua le regioni appartenenti all'una o all'altra classe; questo caso è tipico della segmentazione di un'immagine.

Una volta definite le ROI, occorre scegliere le *feature* da calcolare.

STEP 6: Calcolo di *feature*

Come detto, preliminarmente all'addestramento di un classificatore è necessario effettuare, sui dati o immagini dei pazienti e all'interno delle ROI, il calcolo di *feature* significative per il problema affrontato, ovvero di proprietà misurabili (biomarcatori, attributi) che siano in grado ad esempio di differenziare tra una lesione benigna e una maligna. L'insieme delle *feature* così calcolate può essere visto come un vettore a n componenti, o una n–upla di coordinate in uno spazio a n dimensioni. In definitiva la ROI è modellizzata da un vettore di misure e l'insieme dei dati da adoperare per il processo di addestramento costituisce una matrice con tante righe quanti sono i campioni e tante colonne quante sono le feature, più la colonna *target*.

La scelta di *feature* informative, discriminanti e indipendenti è un compito cruciale nel ML. Esse sono solitamente numeriche, ma possono

anche essere categoriali, booleane, stringhe, ecc. Le variabili numeriche sono più facili da trattare, mentre è spesso opportuno o necessario che quelle categoriali siano convertite in numeriche. Le *feature* possono essere **semantiche** o **agnostiche**: le prime sono legate al contesto e derivano dall'uso comune (in radiologia potrebbero essere misure di vascolarizzazione, di forma, o che descrivano la spiculazione di una lesione, ovvero la presenza di margini irregolari e infiltranti); le seconde colgono la natura di una ROI mediante descrittori matematici di uso generale, come le feature di Haralick (descritte nel prosieguo). Il loro calcolo è anche denominato *feature extraction*.

In letteratura è stata descritta una grande varietà di *feature* potenzialmente utilizzabili per la caratterizzazione di una ROI di un'immagine. A priori, anche esaminando il problema specifico, è spesso difficile dedurre quali siano le più adatte, per cui si preferisce procedere in maniera empirica inserendo nei sistemi di classificazione ampi *set* di variabili di diversa natura e verificandone a posteriori l'utilità, ossia la capacità di discriminazione tra oggetti di classi diverse.

Poiché esistono molti modi e formule diverse per calcolare le *feature*, si raccomanda l'adesione alle linee guida dell'*Image Biomarker Standardization Initiative* (IBSI) [6], che offrono un consenso per calcoli di *feature* radiomiche standardizzate. Un esempio di libreria molto usata in linguaggio `python` è *pyradiomics* [7].

Vi sono *feature* basate sulla *texture* delle ROI e altre basate su caratteristiche geometriche (area e forma).

Feature basate sulla tessitura

Con il termine *texture*, o `tessitura` [5], si definisce una disposizione geometrica eventualmente (grosso modo) ripetitiva dei livelli di grigio o dei colori di un'immagine (figura 8: alcuni esempi di tessiture tratte dal *database* Brodatz[1]). Nei prossimi paragrafi il riferimento sarà alle immagini in toni di grigio, le più diffuse in ambito medico.

L'analisi tessiturale è una tecnica che caratterizza una tessitura calcolandone le caratteristiche distintive con metodi matematici e statistici. L'analisi tessiturale valuta l'intensità di grigio di ogni *pixel* nella zona in esame, in relazione ai *pixel* circostanti.

[1] `http://sipi.usc.edu/database/database.php?volume=textures`, sito visitato in novembre 2021

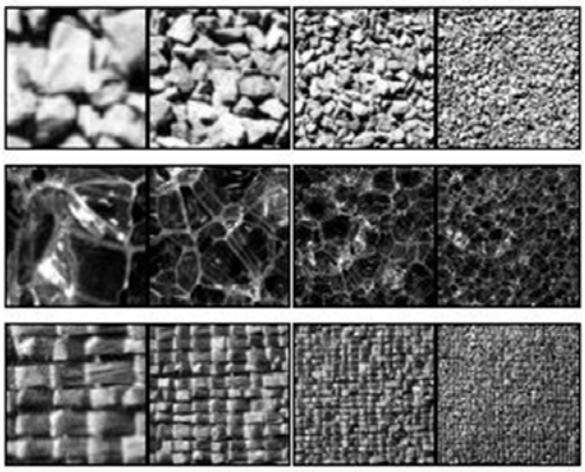

Figura 8: *Esempi di texture (tratte dal database Brodatz).*

Negli ultimi decenni sono state sviluppate molte tecniche per l'estrazione di *feature* impiegate nell'analisi di *texture*. Tra queste si elencano usualmente anche quelle dette del `primo ordine` (media, deviazione standard o varianza, *skewness* e *kurtosis* dei livelli di grigio): si parla di *feature* del `primo ordine` perché esse sono calcolabili a partire dall'istogramma dei livelli di grigio, e quindi sono legate alla probabilità di osservare un particolare valore di grigio in una locazione singola scelta a caso nella ROI. Più propriamente tessiturali sono invece quelle di `secondo ordine` (calcolate su statistiche di coppie di *pixel*, le matrici di co-occorrenza) o di ordine superiore. Come esempio, di seguito sono descritte le *feature* di Haralick (del `secondo ordine`) e i *Local Binary Pattern*.

***Feature* di Haralick.** Le *feature* di Haralick sono calcolate a partire dalle matrici di co-occorrenza (COM, *Co-Occurrence Matrix*) dei livelli di grigio, o GLCM (*Gray Level Co-occurrence Matrix*) [5]. Le GLCM permettono l'estrazione di informazioni statistiche riguardanti la distribuzione di coppie di *pixel* (relazioni tra due *pixel* distinti), pertanto sono anche denominate istogrammi di secondo ordine (o bidimensionali) dei livelli di grigio. Esse sono legate alla probabilità di osservare una coppia definita di valori di grigio nei *pixel* situati alle estremità di un segmento di lunghezza data e orientamento fissato, posto nella ROI dell'immagine in una posizione casuale. Per maggiori dettagli si faccia riferimento

all'Appendice 1.

Local Binary Pattern Il *Local Binary Pattern* (LBP) [10] è un operatore locale basato sulla tessitura, definito come segue. Si voglia calcolare il descrittore nell'intorno di un dato *pixel* di un'immagine a toni di grigio; si considerino i suoi otto primi vicini (distanza di Chebyshev pari a 1). Ai valori dei vicini è applicata un'operazione di soglia rispetto al valore del *pixel* centrale: se il livello di grigio del *pixel* è minore della soglia, ad esso è assegnato il valore 0, altrimenti 1. Si otterrà, in questo modo, un byte espresso in base 2, che verrà convertito in un valore decimale da assegnare al *pixel* dato.

La figura 9 mostra un esempio di codifica. Estendendo il concetto è possibile tenere conto di *pixel* distanti più dei primi vicini o effettuare il calcolo in 3D.

Feature basate sulla forma

Tra le *feature* basate sulla forma della ROI si citano la circolarità, l'eccentricità, i descrittori di Fourier. La figura 10, tratta da [11], ne mostra alcune.
In particolare:
Circolarità. La circolarità C_{cir} è un fattore di forma utilizzato in molti campi dell'analisi di immagini (2D), in particolare in microscopia. Esso descrive la forma di particelle e oggetti indipendentemente dalla loro dimensione. Esistono diverse definizioni di circolarità, ma una molto diffusa è il quoziente isoperimetrico, dato dalla seguente formula [5]:

$$C_{\mathrm{cir}} = \frac{4\pi A}{P^2} \, ,$$

dove A è l'area della ROI e P il suo perimetro. La circolarità di un cerchio è per definizione pari a 1 (sebbene ciò sia vero solo approssimativamente per un cerchio tracciato in un'immagine digitale) mentre, per un oggetto molto diverso da un cerchio, esso diminuisce fino a 0 (caso ideale del segmento di retta).

Eccentricità. L'eccentricità e di un'ellisse è il rapporto tra la distanza tra i fuochi e la lunghezza dell'asse maggiore. Il valore è compreso tra 0 (il caso del cerchio) e 1 (un segmento di retta). Il concetto è esteso a oggetti di forma non ellittica, ponendo e pari all'eccentricità dell'ellisse che ha momenti del secondo ordine (calcolati sulle coordinate) uguali a quelli dell'oggetto. Eccentricità e circolarità esprimono concetti simili ma non coincidenti e sono grandezze tra loro indipendenti.

Descrittori di Fourier. Forniscono una codifica della forma di un oggetto 2D tramite la trasformata di Fourier della linea che ne definisce il bordo. Per i dettagli si rimanda al testo [5].

Standardizzazione o normalizzazione delle *Feature*

Prima di essere utilizzate nella procedura di *training* e *validation*, è opportuno che le *feature* siano normalizzate, tipicamente tra 0 e 1, o standardizzate, ossia convertite mediante la trasformazione

$$z_j^k = \frac{x_j^k - \mu_j}{\sigma_j} \, ,$$

dove μ_j e σ_j sono rispettivamente la media e la deviazione standard della *feature* x_j, di cui x_j^k è il valore per il k–mo campione. Lo scopo è evitare che variabili con valore medio o *range* di variazione elevati siano erroneamente considerate dal classificatore più importanti di altre di piccola entità e varianza, compromettendo il processo ottimale di addestramento. È importante che la procedura di standardizzazione o normalizzazione sia eseguita in modo da non causare *bias* statistico (vedere il paragrafo dedicato agli schemi di apprendimento e validazione). In particolare, i parametri della procedura (minimi e massimi delle *feature*, o loro medie e deviazioni standard, a seconda del metodo che si vuole applicare) devono essere rigorosamente calcolati solo nel *training set* e con questi parametri l'operazione è applicata ai *dataset* di *training*, *validation* e *test*.

A questo punto la matrice di *feature* è pronta per essere utilizzata all'interno di uno schema di addestramento e validazione, per condizionare un classificatore in modo che esso, dopo aver imparato dai dati di *training* come distinguere i vettori di *feature* riferiti a ciascuna classe, sia poi in grado di applicare queste conoscenze a nuovi dati (*set* di validazione e poi di *test*) generalizzando quanto appreso.

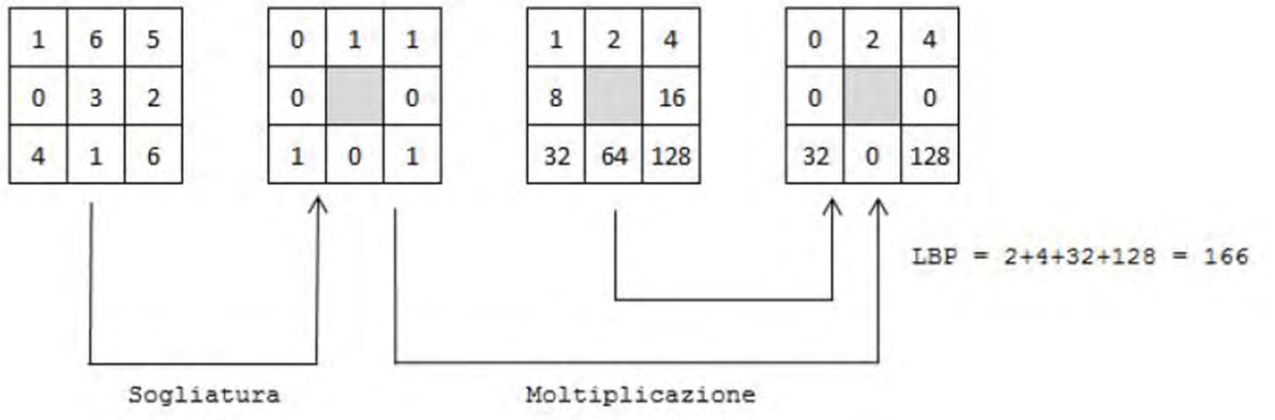

Figura 9: *Codifica attraverso Local Binary pattern del pixel centrale di una regione* 3×3.

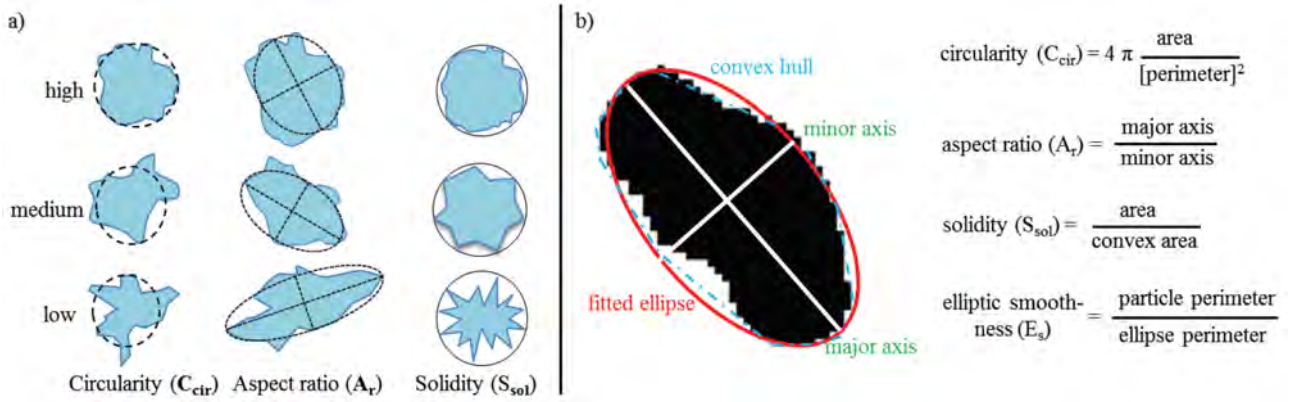

Figura 10: *a) rappresentazione schematica di Circolarità (C_{cir}), Aspect ratio (A_r) e Solidità (S_{sol}) di particelle da proiezione bidimensionale; b) immagine al microscopio di una particella con le equazioni per il calcolo delle feature geometriche. Da [11].*

Potrebbe tuttavia essere necessario uno *step* ulteriore detto di riduzione della dimensionalità, del quale si occupa il paragrafo seguente.

STEP 7: Riduzione della dimensionalità

L'uso di un gran numero di *feature* in un problema di ML ha il vantaggio di aumentare la ricchezza di informazioni e quindi la probabilità di inserire, nel *set* di variabili, quelle effettivamente discriminanti. Tuttavia l'eccessiva dimensionalità del problema può inficiare la qualità del risultato, sia perché un maggior numero di indicatori implica un tempo di calcolo maggiore, sia soprattutto perché si rischia di incorrere nella cosiddetta *curse of dimensionality*, la maledizione della dimensionalità. Il termine, definito per la prima volta da Richard E. Bellman in [12], indica il fatto che l'aumentare della dimensionalità dello spazio delle variabili, senza un contemporaneo incremento nel numero di campioni disponibili, porta a una sparsità dei dati con conseguente rischio di *overfitting* e perdita di qualità.

Allo scopo di risolvere il problema è possibile ricorrere a tecniche di riduzione della dimensionalità [4], mediante le quali si individuano e si eliminano le variabili superflue conservando solo quelle effettivamente utili (*feature selection*). In alternativa si possono applicare opportune trasformazioni allo spazio delle *feature* in modo da massimizzare l'efficacia delle variabili risultanti, di cui si conservano solo le migliori eliminando quelle poco utili (*feature extraction*).

Vantaggi della riduzione della dimensionalità sono: scongiurare la maledizione della dimensionalità evitando l'*overfitting*, accorciare il tempo di addestramento del classificatore e, nel caso della *selection*, semplificare i modelli per scopi interpretativi.

Feature Selection. Seleziona un *subset* effica-

ce e ottimale dall'insieme completo delle *feature*. Allo scopo l'approccio più semplice è valutare il potere discriminante delle singole variabili, ottenibile ad esempio con il calcolo del coefficiente di Fisher (o indice di separazione tra classi, o *Fisher's Score*) [3]:

$$F = \frac{(\mu_- - \mu_+)^2}{\sigma_-^2 + \sigma_+^2} ,$$

dove μ_- e σ_- sono rispettivamente la media e la deviazione standard di una *feature*, calcolate sui campioni negativi (classe 0), e, analogamente, μ_+ e σ_+ sono calcolate per i campioni positivi. È intuitivo che detto *score* assuma valori elevati per variabili che separino bene le due classi, e valori bassi per variabili poco discriminanti, perché massimizza la varianza interclasse allo stesso tempo minimizzando la varianza totale intraclasse.

La tecnica di selezione mediante l'approccio a variabili indipendenti può essere utile, anche perché va nella direzione della spiegabilità dei risultati, ma non è l'opzione migliore, perché non considera le relazioni tra *feature*. Dalla figura 11 appare evidente che le due *feature* mostrate sotto forma di *scatter plot* non sono singolarmente discriminanti (immaginiamo di proiettare i numeri, che rappresentano i campioni delle classi 1 e 2, sugli assi, costruendo degli istogrammi, e ce ne renderemo facilmente conto); è tuttavia evidente che le due variabili, considerate insieme, sono perfettamente discriminanti, perché permettono il passaggio, tra le due nuvole di punti, di un confine di decisione estremamente semplice (una retta).

Appare chiaro, quindi, che occorrono tecniche di selezione più elaborate che considerino le *feature* insieme. Citiamo, senza scendere nei dettagli, la `Sequential Forward Selection` (che costruisce un insieme di variabili ridotto e ottimale partendo da quella più discriminante e via via aggiungendone altre), la `Sequential Backward Selection` (che invece considera inizialmente l'insieme completo di variabili e poi via via elimina le meno utili), la selezione basata su algoritmi genetici, tecniche LASSO (*Least Absolute Shrinkage and Selection Operator*) ecc.

Feature Extraction. Applicano opportune trasformazioni allo spazio delle variabili, con la finalità di calcolarne di più utili alla soluzione del problema, trascurando quelle meno utili derivanti dalla trasformazione. Tra queste, la PCA (`Principal Component Analysis`), la LDA (`Linear Discriminant Analysis`), la ICA (`Independent Component Analysis`). In particolare, quella forse maggiormente adoperata per l'estrazione di *feature* orientata al ML è la LDA (o FLDA: `Fisher Linear Discriminant Analysis`). La FLDA è un metodo di riduzione della dimensionalità ma, nel caso di un problema a due classi, è di fatto un metodo di classificazione in sè, perché il risultato della riduzione è un'unica variabile ottimale. La FLDA applica una trasformazione lineare allo spazio delle *feature*, finalizzata a massimizzare il già citato coefficiente di Fisher. Essa definisce quindi nuovi assi di riferimento per i dati. Al termine della trasformazione, avviene la proiezione su un singolo asse (riduzione della dimensionalità ad una sola variabile) scegliendo come asse di proiezione quello a cui competa il massimo coefficiente di Fisher. Ciò porta ad una separazione ottimale tra i dati delle due classi (figura 12).

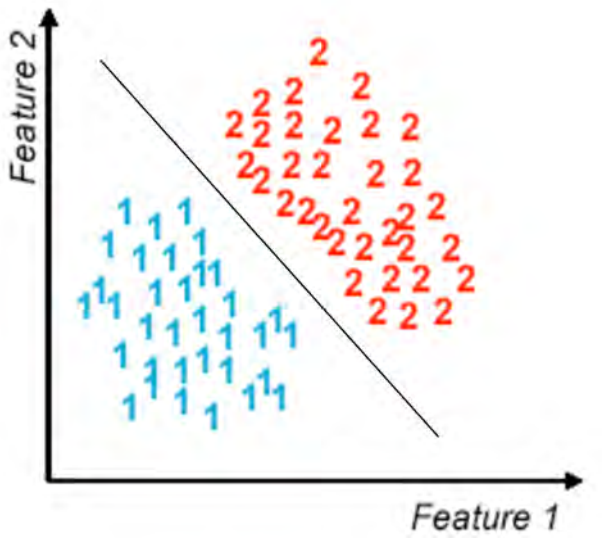

Figura 11: *Due feature singolarmente non discriminanti, ma ottimali se considerate insieme.*

STEP 8: Scelta di uno schema di addestramento e validazione, classificazione, e calcolo della qualità di un classificatore

La classificazione supervisionata, caratteristica del ML (e del DL) necessita di uno schema di addestramento e validazione incrociata, ossia una

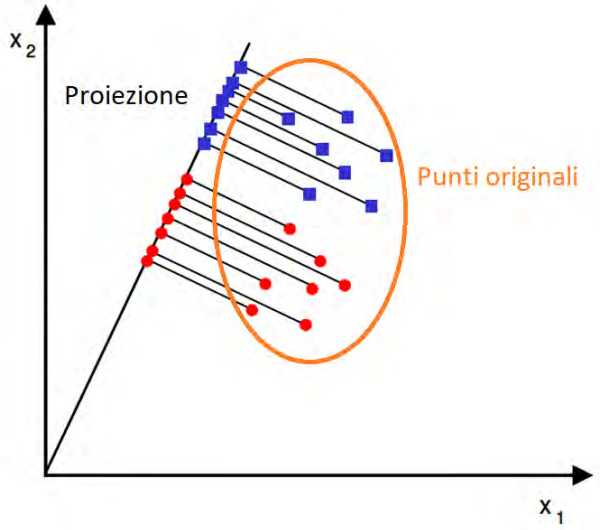

Figura 12: *Esempio di applicazione della FLDA. L'asse di proiezione individuato è ottimale e massimizza la separabilità, ora ottenibile tramite semplice sogliatura.*

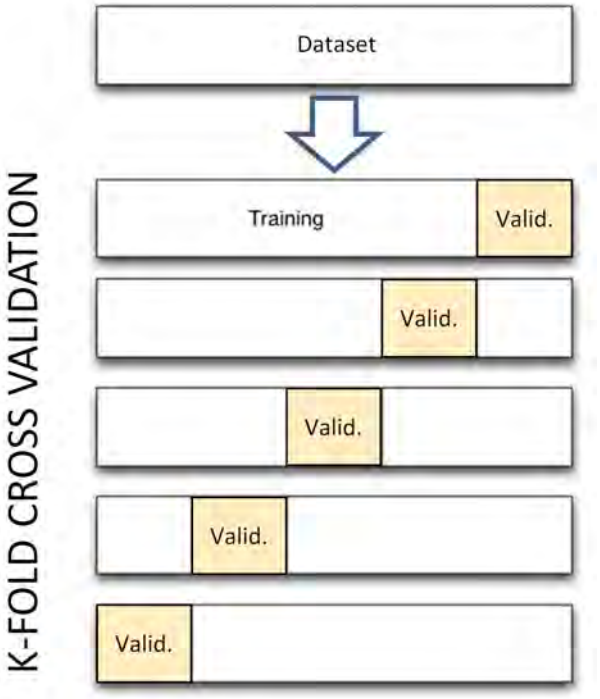

Figura 13: *Schema di k–fold cross validation con* $k = 5$.

procedura con la quale il classificatore è allenato su parte del *dataset* (*training set*) e l'apprendimento è verificato su un'altra parte di esso (*validation set*). Verificare l'apprendimento sullo stesso insieme di *training* non dà, infatti, una misura affidabile della qualità che il classificatore avrebbe su nuovi dati, che è invece la finalità ultima della procedura (capacità di *generalizzare*).

Nell'approccio più semplice, definito *hold-out cross validation* [13] i dati sono divisi in (almeno) due parti, gli insiemi di *training* e di *validation*. Il primo serve per l'ottimizzazione degli iperparametri del classificatore (per la minimizzazione dell'errore di classificazione) mentre il secondo è usato solo per la verifica (*unbiased evaluation*, ossia una valutazione non condizionata dai dati sui quali il classificatore è stato allenato) dell'apprendimento e dell'effettiva generalizzazione (capacità di classificare nuovi campioni). Tipicamente il partizionamento avviene in percentuali di 70%–30%, oppure 80%–20%. La procedura di allenamento e verifica è condotta iterativamente, ogni volta modificando leggermente gli iperparametri del modello o il tipo di classificatore sperimentato, fino a giungere al risultato migliore.

In realtà, oltre a ricavare dai dati i due insiemi di *training* e *validation*, è opportuno estrarre e mettere da parte un terzo insieme di campioni, denominati di *test*. Alla fine del processo iterativo di ottimizzazione degli iperparametri, il classificatore dovrebbe essere testato su questi dati per ottenere una misura affidabile della qualità. Infatti, con la ripetizione del ciclo di allenamento e verifica il modello addestrato diventa sempre più sbilanciato (*biased*) verso il *set* completo di apprendimento (*training* $\cup$ *validation*) finché la verifica su un insieme terzo (*final unbiased evaluation*) diventa importante. Fatto ciò, il modello non dovrebbe più essere variato in alcun modo.

Accanto al semplice *hold-out* esistono svariati altri schemi di valutazione dell'apprendimento [14]. Tra questi, il *k-fold cross validation* o *k-fold* CV, e una procedura simile detta *leave-one-out cross validation* (LOO, LOO-CV). Nel *k-fold* CV (figura 13) partizioniamo l'insieme di apprendimento in un numero k di *fold* (ovvero in k parti). Una è riservata alla validazione mentre $k - 1$ parti sono adoperate per l'addestramento (se $k = 5$ quest'ultimo corrisponde all'80% dei dati). I pronostici sono accantonati per il calcolo delle cifre di merito (e.g. la curva ROC). Quindi l'allenamento è ripetuto prendendo una parte (diversa) per la convalida e il resto per l'allenamento. Si reitera complessivamente k volte. La curva ROC o altri dati complessivi di merito sono calcolati dalle previsioni cumulate. Il LOOCV è una procedura analoga, in cui però k coincide con

il numero di campioni. Si utilizzano anche le sigle LOPO o LOSO, riferendosi specificamente a pazienti o soggetti.

Particolare attenzione va fatta, al momento del partizionamento dei dati, nel caso in cui a ciascun campione (paziente, soggetto) corrisponda più di un vettore di *feature* (è il caso per esempio di un'applicazione medica in cui vi siano misure fatte su più campioni bioptici, come nel caso delle biopsie multiple per la diagnosi del tumore della prostata): in tal caso, per evitare *bias*, si dovrà avere l'accortezza di non suddividere mai lo stesso paziente tra insieme di allenamento e insieme di validazione.

STEP 9 Classificatori: struttura, addestramento e misura della *performance*

Esiste una grande varietà di classificatori supervisionati, con una tassonomia molto complessa. Un elenco non esaustivo comprende `artificial neural networks`, `support vector machines`, `decision trees`, metodi basati su `gradient boosting` e tanti altri.

Già è stato chiarito come la procedura di classificazione si basi sulla definizione di una regola di decisione che partizioni lo spazio delle *feature* in regioni disgiunte, ognuna delle quali sia attribuibile ad una delle classi note (regioni di decisione), separate da una frontiera. I classificatori si distinguono per il metodo che adoperano per individuare tale partizionamento. In quest'articolo si darà qualche informazione sul neurone artificiale (il `percettrone`) e le reti neuronali artificiali con esso costruite (il *multi-layer perceptron*).

Neuroni biologici e artificiali

Le reti neurali artificiali sono modelli matematici composti da semplici elementi (*Processing Elements*) operanti in parallelo, il cui funzionamento trae ispirazione dai sistemi nervosi animali. Il cervello umano contiene circa 10^{11} cellule denominate neuroni. Esse sono caratterizzate da tre parti principali (figura 14): soma o corpo cellulare, dendriti, e assone. Il soma è la parte centrale della cellula, contenente il nucleo, e da esso si irradiano prolungamenti sottili detti dendriti, e l'assone. I dendriti ricevono i segnali provenienti dagli assoni degli altri neuroni e quindi costituiscono gli ingressi della cellula, mentre attraverso l'assone il neurone trasmette segnali ad altre cellule. La comunicazione tra assone e dendrite avviene alla giunzione, detta sinapsi. Ogni neurone è tipicamente connesso a un migliaio di altri neuroni e il numero totale di sinapsi nel cervello umano supera 10^{14} [15].

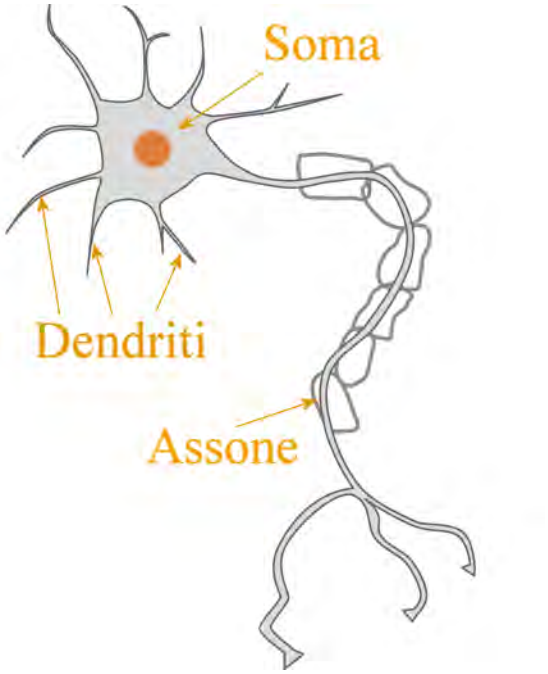

Figura 14: *Struttura di un neurone biologico.*

Semplificando, un neurone si può trovare essenzialmente in due stati: a riposo o attivo. Quando passa dallo stato di riposo a quello attivo, esso produce un impulso elettrico detto potenziale di azione (generalmente uno *spike* di circa 35 mV di picco), poi trasmesso dall'assone. Il segnale, raggiunte le sinapsi, provoca il rilascio di sostanze chimiche denominate neurotrasmettitori che attraversano la giunzione ed entrano, viaggiando nel dendrite, nel corpo del neurone connesso. Le sinapsi possono essere eccitatorie o inibitorie: nel primo caso i neurotrasmettitori aumentano la probabilità che il successivo neurone si attivi, nel caso contrario la diminuiscono. Ad ogni sinapsi è associato un peso che determina la natura e l'intensità dell'effetto eccitatorio o inibitorio. Se la somma pesata degli ingressi provenienti dagli altri neuroni supera, entro i tempi propri delle attivazioni neuroniche, una specifica soglia, allora il neurone si attiva.

Un neurone opera in tempi dell'ordine di 1 ms

e quindi è un sistema di elaborazione lento se confrontato con un *personal computer* i cui tempi tipici delle operazioni elementari vanno dai microsecondi dei primi modelli ai nanosecondi attuali. Tuttavia, il gran numero di neuroni e sinapsi che operano in parallelo rende il cervello un sistema potente, oltre a conferirgli alta tolleranza a informazioni affette da rumore e ad eventuali danni di estensione limitata (robustezza), e le facoltà di apprendimento e generalizzazione.

La struttura del neurone artificiale emula quella del neurone biologico [16]. Prendendo come esempio il `percettrone`, introdotto da Rosenblatt nel 1958 [17] partendo dalle teorie di McCulloch e Pitts [18], vediamo che esso è costituito da (figura 15):

- alcuni ingressi x_i che ricevono informazioni dall'esterno (equivalenti ai dendriti del neurone biologico); vi sono tanti ingressi quante sono le *feature* utilizzate dal modello;
- un peso w_i su ciascun ingresso; tali pesi rappresentano l'intensità del trasferimento dell'informazione tra neuroni e modellizzano le sinapsi;
- un *input* ulteriore, posto sempre a 1 e denominato *bias* (naturalmente, con significato diverso rispetto al *bias* statistico di cui si è parlato più volte nell'articolo); il peso associato all'ingresso di *bias* è indicato con b; il significato dell'*input* di *bias* sarà chiaro nel prosieguo;
- una funzione di attivazione $f()$: può assumere varie forme ma nel caso del singolo `percettrone` essa è la funzione a gradino;
- un valore di *output* y, informazione trasmessa al neurone successivo, dato dal risultato dell'applicazione di f alla somma pesata degli ingressi: $y = f(\sum_i w_i x_i + b)$.

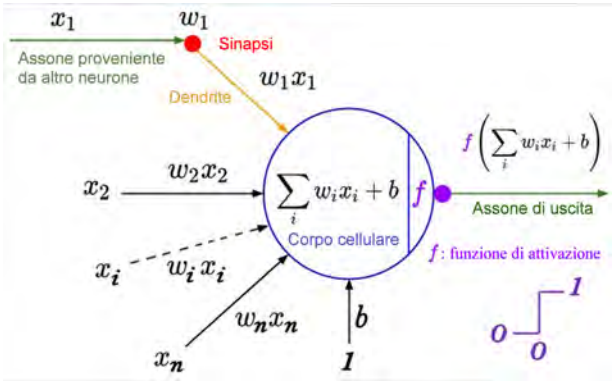

Figura 15: *Struttura di un neurone artificiale. (Adattata da: https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc, sito visitato in novembre 2021.)*

Il neurone artificiale è in grado di apprendere

In definitiva, il `percettrone` riceve un insieme di valori in *input* (il vettore di *feature*) e, in base ai pesi e a una funzione di attivazione, produce un determinato *output*. Un singolo `percettrone` può essere visto come un caso particolare di rete neurale artificiale *feedforward* (ovvero in cui l'informazione viaggia in un solo verso, dagli *input* verso l'*output*, in contrapposizione alle reti ricorsive), avente un solo strato di neuroni (composto da un solo neurone) e ad una sola uscita. Vogliamo che il neurone artificiale fornisca, come *output* dell'assone, la classe a cui appartiene il vettore (e quindi l'oggetto o il campione modellizzato da quel vettore). Senza perdere in generalità, possiamo supporre che l'*output* valga 0 o 1. È possibile dimostrare che una scelta appropriata dei pesi consente la corretta classificazione in caso di problemi linearmente separabili.

La procedura che porta a determinare il valore dei pesi è *error driven*, poiché sfrutta l'errore compiuto dal `percettrone` nella classificazione, per aggiustarne l'entità, in base alla *perceptron learning rule* introdotta da Rosenblatt. I pesi sono inizialmente assegnati in maniera *random*. A seguire, l'aggiustamento è realizzato fornendo ciclicamente e più volte come *input* i vettori dell'insieme di *training* e calcolando il valore di uscita o del neurone. Ad ogni iterazione i pesi sono aggiornati secondo la formula:

$$w_{\text{new}} = w_{\text{old}} + \eta(t - o)x_i \ ,$$

dove x_i è l'i-mo vettore di *feature*, t e o sono rispettivamente l'*output* desiderato (la classe vera alla quale appartiene il vettore di *feature* x_i) e l'*output* del neurone, mentre η è il tasso di apprendimento (o *learning rate*), relativamente piccolo per garantire la convergenza del processo iterativo.

Per una dimostrazione della capacità del `percettrone` di operare come classificatore, si

veda il *box* "Il `percettrone` come classificatore lineare".

Il `percettrone` ha capacità limitate dal fatto che il problema debba essere linearmente separabile. Per problemi più complessi si utilizzano reti di neuroni (figura 17) che prevedono uno strato (o livello) di connessioni di *input*, uno di neuroni di *output* (un solo neurone se il problema è di classificazione binaria, più neuroni in caso di classificazione multiclasse) e uno o più strati nascosti (ossia situati tra i livelli di ingresso e di uscita) popolati da neuroni (*hidden layers*). Ciascun neurone è caratterizzato da una propria funzione di attivazione che, a seconda delle applicazioni, può essere una sigmoide, una tangente iperbolica, o altre funzioni non lineari.

La regola di apprendimento in questo caso diventa più complessa. Un esempio è la regola di `backpropagation` [19], chiamata così perché l'errore si propaga all'indietro dalle unità di uscita verso lo strato di ingresso, via via modificando i coefficienti di peso, dapprima relativi al collegamento tra l'ultimo strato e il penultimo, poi a quelli tra il penultimo e il terz'ultimo, e così via. La procedura è realizzata iterativamente, inizializzando i pesi con valori casuali e poi sottoponendo più volte alla rete gli esempi del *dataset* di *training*, finché non è raggiunto un errore finale complessivo ritenuto tollerabile (o sono soddisfatti altri criteri di stop). Nel `backpropagation` sono adoperate le derivate delle funzioni di attivazione tra i vari neuroni, per cui queste devono essere derivabili.

La tipologia di problema risolvibile con una rete neurale dipende dalla sua struttura o architettura, ossia dal numero di strati nascosti. Facendo riferimento alla figura 18 si osserva come il `percettrone` sia in grado di risolvere esclusivamente problemi linearmente separabili, ossia in cui le classi possono essere separate da un iperpiano. Una rete con un singolo strato nascosto può risolvere problemi non linearmente separabili, in cui il *decision boundary* identifichi un dominio convesso, mentre un doppio strato nascosto permette la soluzione di problemi non linearmente separabili con confine arbitrario.

Qualità di un classificatore

Un classificatore, una volta addestrato, sarà in grado di mappare, nella maniera più corretta possibile anche compatibilmente con la qualità dei dati, un vettore di *feature* verso la sua classe di appartenenza. È naturalmente importante poter esprimere quantitativamente la qualità con cui avviene la mappatura, in modo da valutare l'efficacia dell'apprendimento e della capacità di generalizzare. Le prestazioni di un sistema di classificazione sono misurate attraverso l'introduzione di diversi parametri, tra i quali:

- numero di veri positivi, veri negativi, falsi positivi, falsi negativi, e la matrice di confusione
- sensibilità, specificità, precisione, recupero (*recall*), accuratezza, *F-measure*
- curva ROC e curva `precision-recall`.

Consideriamo nel prosieguo sistemi di classificazione binaria, fermo restando che le considerazioni esposte e le grandezze calcolate si estendono con facilità alla classificazione multiclasse. È opportuno, date le due classi *target*, stabilire quale considerare la classe dei positivi e quale quella dei negativi, ove questo non sia naturale, perché alcuni termini e alcune figure di merito fanno riferimento esplicito a positivi e negativi e quindi ciò semplifica l'argomentazione. Naturalmente la convenzione trae origine dal linguaggio medico dove i termini positivo e negativo si riferiscono rispettivamente a elementi patologici o sani, ed è esteso a un caso generico di oggetto di interesse (positivo) e oggetto non di interesse (negativo), o di distinzione tra qualità di pari importanza come ad esempio la discriminazione tra due tipi istologici di tumore.

Gran parte dei classificatori fornisce in uscita un valore continuo (ad esempio tra 0 e 1) e solo l'applicazione di una soglia discretizza l'*output* che così identifica la classe proposta (classificatore discreto binario, con i soli possibili valori di *output* 0 e 1). Vi sono figure di merito calcolate dopo l'applicazione della suddetta soglia, e altre che si applicano prima della sogliatura e permettono di dare un valore di qualità indipendente dalla soglia stessa.

Per meglio comprendere il significato dell'applicazione della soglia, conviene far riferimento alla figura 19 che rappresenta (qualitativamente)

Il `percettrone` come classificatore lineare

Per dimostrare che il `percettrone` è in grado di classificare, ove il problema sia linearmente separabile, consideriamo il suo *output*, pari a $y = f(a)$ dove $a = \sum_i w_i x_i + b$ e $f()$ è la funzione di attivazione (funzione a gradino).
Nel semplice caso di due variabili, questo significa che $f(a) = 1$ se e solo se $w_1 x_1 + w_2 x_2 \geq -b$, quindi (se $w_2 \neq 0$):

$$x_2 \geq -\frac{w_1}{w_2} x_1 - \frac{b}{w_2}.$$

Se identifichiamo x_1 e x_2 rispettivamente con le variabili dipendente e indipendente x e y, e poniamo $m = -(w_1)/(w_2)$ e $n = -b/w_2$, la diseguaglianza precedente diventa $y \geq mx + n$. Ciò significa che l'*output* del neurone è 1 se e solo se la coppia di coordinate, e quindi la coppia di *feature*, (x_1, x_2) identifica un punto nel semipiano superiore delimitato dalla retta $y \geq mx + n$ (figura 16). Resta quindi dimostrato che il `percettrone` possa risolvere problemi di classificazione linearmente separabili.

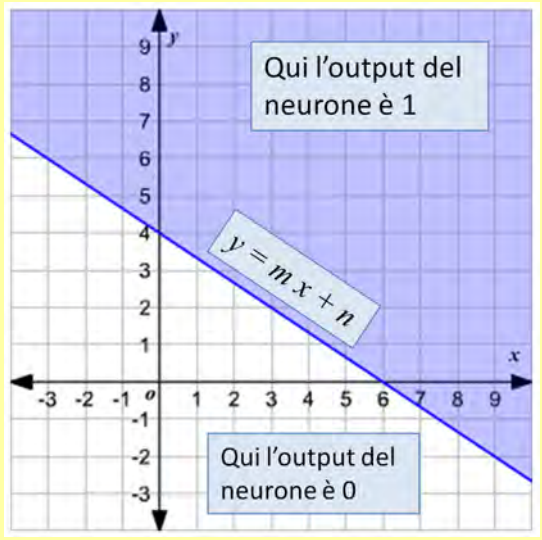

Figura 16: *Output di un* `percettrone`.

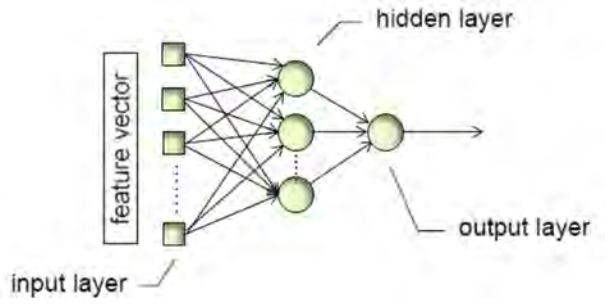

Figura 17: *Esempio di rete neurale artificiale feedforward caratterizzata da uno strato di connessioni di input (di numerosità pari al numero di feature), uno strato di neuroni di output (in questo caso con singolo neurone per la classificazione binaria) e da uno strato nascosto popolato da tre neuroni.*

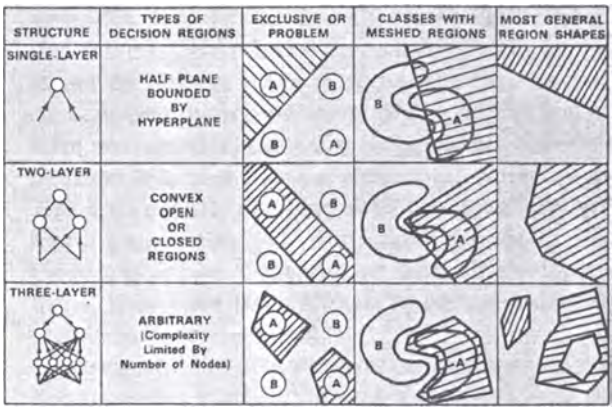

Figura 18: *Tipologia di problemi risolvibili tramite una rete neurale artificiale feedforward, in base alla sua struttura. Dall'alto verso il basso i casi raffigurati sono: rete a singolo strato (`percettrone`), rete con due strati (di cui uno nascosto) e rete con tre strati (di cui due nascosti) (da [20]).*

l'*output* del classificatore, prima dell'applicazione di una soglia, mediante istogrammi per classe normalizzati. Come spesso succede nella realtà, si è scelta una rappresentazione campaniforme senza con ciò perdere di generalità. Per chiarire meglio le idee, nel caso del `percettrone` la quantità rappresentata sarebbe $\sum_i w_i x_i$.

Supponiamo ora di applicare una soglia θ_0. In figura il valore di soglia è stato scelto come punto di intersezione tra le curve che approssimano gli istogrammi, ma questa è solo una delle scelte possibili. Consideriamo i campioni che contribuiscono alla parte indicata con a: essi sono casi effettivamente positivi che, essendo sopra soglia, anche il classificatore ha indicato come positivi. I campioni nella parte b sono invece negativi

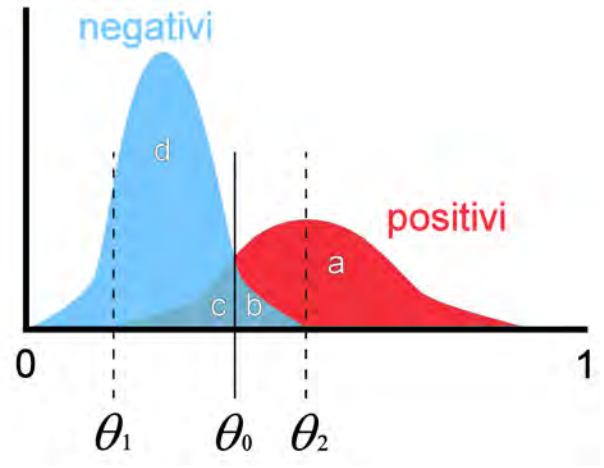

Figura 19: *Istogrammi per classe, normalizzati, del valore di uscita del classificatore, prima dell'applicazione di un'operazione di sogliatura.*

ma, essendo a destra della soglia, il classificatore li ha erroneamente classificati come positivi. Analogamente si considerino le parti c e d.

È possibile allora dare le seguenti definizioni:

- se il campione è positivo (patologico) ed è classificato come tale dalla rete neurale (*output* $\geq \theta_0$), esso costituisce un vero positivo (TP) e contribuisce alla parte etichettata come a; se invece è classificato come negativo dalla rete neurale (*output* $< \theta_0$), esso costituisce un falso negativo (FN) (caso c);
- se il campione è negativo (sano) ed è classificato come tale, esso è contato come vero negativo o TN (caso d), se è classificato come positivo, è definito falso positivo o FP (caso b).

Da queste definizioni di base si possono dedurre altre grandezze di interesse, ad esempio (indicando con FP il numero di falsi positivi, ecc.):

- **sensibilità** o *TP rate*, o recupero (*recall*): è la frazione di campioni positivi correttamente classificati): $TP/(TP+FN)$,
- **rapporto dei falsi positivi** (*FP rate*) o frequenza dei falsi allarmi: $FP/(TN+FP)$,
- **specificità** (è la frazione di campioni negativi correttamente classificati): $TN/(TN+FP)$, (complementare rispetto a 1 di *FP rate*)
- accuratezza (frazione di campioni correttamente classificati): $(TP+TN)/(TP+TN+FP+FN)$,
- **precisione**: $TP/(TP+FP)$,
- ***F-measure*** o *F-score* o *F1-score*:

$$2\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

  è la media armonica tra precisione e richiamo
- **matrice di confusione**:

$$\begin{pmatrix} TP & FP \\ FN & TN \end{pmatrix}$$

  (definizione non univoca, poiché fonti diverse definiscono la matrice di confusione come la trasposta di quella mostrata, oppure scambiano sia le righe che le colonne).

Un sistema di IA per la Medicina deve avere:

- alta sensibilità, in modo da rilevare tutte le possibili lesioni (o tutti i pazienti affetti da una patologia);
- alta specificità, per evitare falsi allarmi e inutili e dispendiosi approfondimenti diagnostici.

Si noti, sempre dalla figura 19, che la scelta della soglia influisce direttamente sui valori di sensibilità e specificità (e sulle altre figure di merito elencate). Infatti, se la soglia è $\leq \theta_1$, ossia a sinistra del valore minimo di *output* per i campioni positivi, evidentemente tutti i positivi saranno classificati come tali e la sensibilità sarà massima e pari a 1; tuttavia, la specificità sarà molto bassa. Al contrario, se la soglia è $> \theta_2$, ossia a destra del valore massimo di *output* per i campioni negativi, tutti i negativi saranno classificati come tali e la specificità sarà massima uguale a 1, a scapito di una sensibilità ridotta. Valori intermedi della soglia daranno combinazioni diverse di sensibilità e specificità, sempre tra loro in competizione e compresi nell'intervallo (0,1).

Come accennato in precedenza, nel caso di *dataset* sbilanciati è consigliabile l'uso di *F-measure* al posto dell'accuratezza, perché quest'ultima tende a dare valori eccessivamente ottimistici anche quando il classificatore favorisce la classe maggioritaria.

È opportuno introdurre un ulteriore indicatore, detto curva ROC (*Receiver Operating Characteristics*), che fornisce una misura della capacità di-

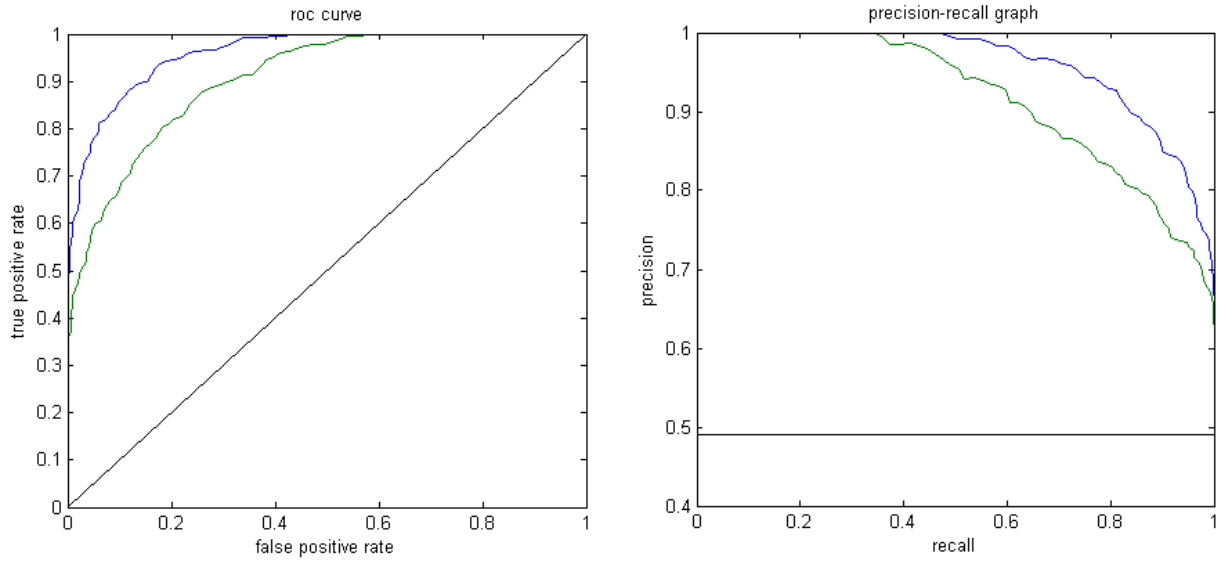

Figura 20: *Sinistra: curve ROC; destra: curve P-R. Le curve in blu sottendono un'area maggiore e quindi caratterizzano un classificatore di qualità migliore.*

scriminante del sistema indipendentemente dalla soglia applicata (ossia a partire dal suo *output* continuo), e in maniera grafica.

Lo spazio ROC è uno spazio bidimensionale in cui l'ascissa è rappresentata dal valore di *FP rate* e l'ordinata dalla sensibilità. All'interno di questo spazio, necessariamente delimitato dal quadrato $[0,1] \times [0,1]$, possono essere posti punti di coordinate $(1 - \text{specificità}, \text{sensibilità})$ che rappresentano punti di lavoro diversi del classificatore. Facendo variare la soglia di uscita del classificatore tra i valori estremi, il luogo di tali punti descriverà in generale una curva, detta curva ROC (figura 20, a sinistra), che lega la sensibilità al *FP rate*, cioè descrive la relazione fra veri e falsi allarmi. Applicando un determinato valore di soglia, la rete neurale diventa un classificatore discreto binario e lavorerà in un singolo punto definito nello spazio ROC.

Punti di lavoro diversi (definiti dalla soglia) sono scelti in base all'importanza relativa di specificità e sensibilità. Ad esempio, in un problema medico è spesso più importante riconoscere tutti i positivi, sia pure con il rischio di un numero non indifferente di FP, per cui si prediligerà un punto ad alta sensibilità. In molti casi, un punto ottimale è l'intersezione della curva con la seconda bisettrice del quadrante, dove la sensibilità è uguale alla specificità, oppure il punto della curva più vicino alle coordinate (0,1), che rappresentano il punto ideale.

Prescindendo dal valore della soglia, la capacità discriminante di un classificatore è legata all'area sottesa dalla curva ROC (*Area Under the Curve*, AUC o AUROC or ROC AUC). Nel caso di un *test* perfetto, ossia di specificità e sensibilità entrambe pari a 1, il valore di AUC corrisponde all'area dell'intero quadrato delimitato dai punti di coordinate (0,0), (0,1), (1,0), (1,1), e assume valore 1 (probabilità del 100% di una corretta classificazione). Al contrario, la ROC per un *test* assolutamente privo di valore informativo è rappresentata dalla diagonale che passa per l'origine, con AUC = 0.5, corrispondente a una scelta casuale della classificazione.

Per l'interpretazione dei valori dell'area sottostante la curva ROC è possibile riferirsi allo schema proposto in [24]:

- AUC = 0.5: sistema diagnostico non informativo;
- $0.5 < \text{AUC} < 0.7$: poco accurato;
- $0.7 \leq \text{AUC} < 0.9$: moderatamente accurato;
- $0.9 \leq \text{AUC} < 1.0$: altamente accurato;
- AUC = 1.0: perfetto.

Nel caso della *k-fold* CV o del LOO, anziché calcolare la AUC ad ogni iterazione, spesso i risultati predittivi acquisiti nelle differenti fasi di *training* e *validation* sono accumulati e la curva ROC è calcolata alla fine della procedura.

Infine, qualora si abbia a che fare con un *dataset* sbilanciato, è opportuno adoperare, al posto

della curva ROC, la curva *Precision-Recall* (P-R) (figura 20, a destra). La curva P-R visualizza il compromesso tra precisione e richiamo al variare della soglia. Un'area elevata sotto la curva rappresenta sia un *r*ecall che una *precision* elevati. la P-R AUC (o AUPR) è meno comoda da trattare in quanto il suo valore di base (legato alla *baseline* della curva e corrispondente a un test non informativo) non è fisso ma dipende dall'entità dello squilibrio.

Machine Learning vs *Deep Learning*

Il DL è una particolare architettura basata sul ML, da cui si differenzia per alcuni aspetti importanti. Esaminare in profondità il funzionamento del DL (o, quanto meno, delle reti neurali convoluzionali, o CNN, la più nota implementazione del DL) comporterebbe diverse pagine ed è al di fuori delle finalità di quest'articolo, essenzialmente divulgativo. Saranno pertanto mostrate brevemente le caratteristiche di base e saranno evidenziate le differenze con l'approccio ML, rimandando per i dettagli alla letteratura specifica (ad esempio [21]).

Il DL è nato su ispirazione del funzionamento della corteccia visiva cerebrale e i processi di visione biologica (figura 21) ed è diventato uno strumento estremamente efficace nella visione artificiale e in generale in compiti visivi (riconoscimento di forme e oggetti in immagini, in particolare in immagini naturali ma anche in immagini di diagnostica medica).

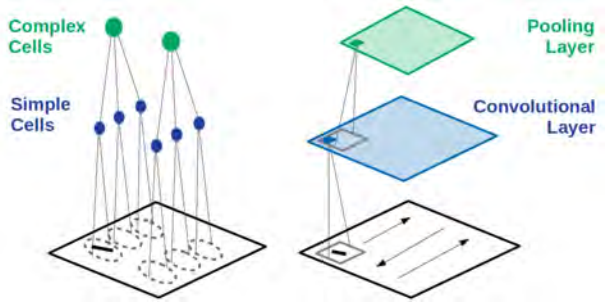

Figura 21: *Relazione tra la struttura della corteccia cerebrale visiva e i componenti base di una rete neurale convoluzionale, da [21].*

Hubel e Wiesel, due neurologi premi Nobel, scoprirono nel 1962 [22] che singoli neuroni da loro denominati `cellule semplici`, situati nella corteccia visiva del gatto (a sinistra in figura 21, in blu), rispondevano intensamente a elementi visuali di forma lineare orientati specificamente, localizzati in posizioni ben definite dell'immagine. Cellule denominate `complesse`, invece, avevano un campo ricettivo più ampio (rispondendo comunque all'orientazione, ma da elementi visuali anche localizzati in regioni vicine), perché ricevevano l'*input* dall'*output* delle cellule semplici (in verde in figura).

Nel 1980, Fukushima applicò le scoperte di Hubel e Wiesel realizzando un modello informatico funzionante del sistema visivo [23], il `Neocognitron`, precursore delle moderne CNN. Il principio è mostrato in figura 21 a destra. Il primo strato (`convoluzionale`, in blu) applica un'operazione di convoluzione all'immagine da riconoscere. In particolare, l'applicazione di un piccolo filtro di convoluzione (riquadro grigio) a ogni posizione nell'immagine (finestra mobile) crea una mappa di *feature*. Uno strato (*layer*) convoluzionale ha tante mappe di *feature* quanti sono i filtri applicati (nel caso particolare, solo uno). Conservando il valore massimo di attivazione in una piccola sezione di ciascuna mappa di *feature* (riquadro grigio) di fatto si sottocampiona l'immagine e si costruisce una risposta complessa simile a quella fornita dalle cellule *complesse* del caso biologico (in verde in figura). Quest'operazione è nota come *max-pooling* e le mappe di *feature* sottocampionate così prodotte costituiscono i livelli di *pooling*.

I primi strati delle reti convoluzionali hanno quindi il compito di estrarre dalle immagini le *feature* di basso livello, che poi si combinano tra loro strato per strato. Le *feature* non sono, dunque, ingegnerizzate manualmente come nel caso del ML, ma sono individuate in maniera automatica. Gli ultimi strati delle reti hanno il compito della classificazione vera e propria. Nelle applicazioni reali, le CNN adoperate per il DL sono complesse, come si può vedere in figura 26, nella sezione dedicata agli esempi applicativi. Nel corso degli anni si è passati da reti con un numero relativamente limitato di livelli a strutture che ne impiegano oltre cento e di tipo diverso, con un sensibile incremento delle prestazioni e quindi una diminuzione dell'errore di classificazione.

La applicazioni erano inizialmente limitate al riconoscimento delle immagini naturali (impulso notevolissimo è stato dato dalla realizzazione del

database `ImageNet`, contenente 14 milioni di immagini) poi esteso ad altri campi, tra cui le immagini diagnostiche mediche. Una rete convoluzionale ha necessità di un grandissimo numero di immagini durante la fase di *training*, il che spesso ne limita l'applicazione a favore del tradizionale ML per situazioni nelle quali la quantità di dati non sia sufficiente. Ciò accade spesso in Medicina, sia per la relativa rarità di una patologia, sia per la difficoltà di raccogliere massicciamente i dati in maniera omogenea e utile (studi multicentrici). Vengono allora in aiuto stratagemmi quali il *transfer learning* che, anziché allenare da zero una rete, parte da un sistema preallenato su immagini di natura diversa (immagini naturali, ad esempio `ImageNet`) e attua una delle seguenti strategie:

- uso delle *feature* calcolate dalla rete preallenata, e classificazione mediante un classificatore in ML (ad esempio, una rete neurale tradizionale, o *shallow* in contrapposizione alle reti *deep*, oppure una SVM o *Support Vector Machine*)
- *fine tuning*, che consiste nel modificare opportunamente in base al problema gli ultimi strati (che hanno il compito della classificazione), adattando solo i pesi di questi ultimi durante una fase di *training* sul *dataset* specifico.

Ulteriore differenza sostanziale tra DL e ML sono i tempi di calcolo, molto maggiori per il DL qualora si richieda il *training* completo della rete.

Alcuni esempi di applicazioni

La letteratura registra ormai una quantità considerevole di articoli sulle applicazioni del ML o del DL in Medicina: darne un campione rappresentativo sarebbe compito improbo oltreché improponibile per la quantità di testo necessario. È stata dunque scelta una via alternativa: la breve rassegna che segue riassume quattro articoli rappresentativi della ricerca svolta nel campo dall'autore di questo lavoro.

Preliminarmente è tuttavia utile esaminare la figura 22 che rappresenta l'evoluzione temporale e la distribuzione geografica delle pubblicazioni internazionali referenziate, censite dal database Scopus, aventi per oggetto i sistemi di ML o DL, o riferite alla radiomica, o aventi le parole chiave CADx o CADe. La figura 23 riporta le medesime informazioni ma è riferita al solo termine *radiomica*. Si nota innanzitutto che il numero di lavori globalmente dedicati all'argomento è in crescita continua e l'Italia si pone in ottima posizione (addirittura, per la parola chiave *radiomica*, è al terzo posto dopo colossi come USA e Cina). In particolare dal 2012, anno in cui uscì il lavoro seminale sulla radiomica [2], la crescita del numero di lavori che riportano questa parola chiave è stata praticamente esponenziale (tranne negli ultimi due anni in cui un assestamento ha portato la curva vicina alla linearità). Ciò dimostra, insieme ai numeri assoluti di tutto rispetto, l'importanza che le tematiche hanno nel panorama delle letteratura scientifica odierna.

Una patologia prenatale rara: l'ernia congenita diaframmatica (CDH)

Il primo esempio [25] è riportato con la finalità di porre l'accento sull'importanza di definire un protocollo chiaro e dettagliato prima di realizzare uno studio di ML/DL in Medicina. Troppo spesso, infatti, il lavoro è organizzato a partire da dati disomogenei (come spesso capita nel caso di studi retrospettivi) e senza un'adeguata coscienza di quali siano i quesiti clinici per i quali si cerca risposta. L'articolo riporta quindi il protocollo stabilito a monte per la realizzazione di un sistema di IA a scopo prognostico per l'ernia diaframmatica congenita (CDH). Si tratta di una patologia rara del feto, consistente nel passaggio parziale dei visceri addominali nel torace, attraverso un difetto del diaframma (erniazione), con conseguente compressione e iposviluppo degli organi toracici (in particolare i polmoni), insufficienza respiratoria e ipertensione polmonare persistente, con elevata mortalità alla nascita quando il bambino dovrà utilizzare i propri polmoni per la respirazione.

Le previsioni sull'esito dei pazienti con ernia diaframmatica congenita (CDH) sono ancora limitate nella stima prenatale dell'ipertensione polmonare postnatale (PH). L'articolo propone di applicare approcci di Machine Learning (ML)

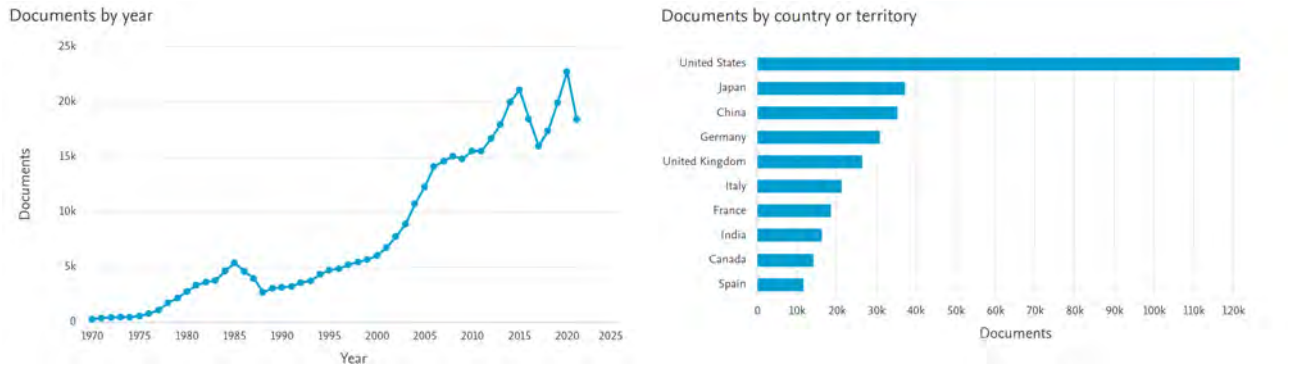

Figura 22: *Evoluzione nel tempo (sinistra) e distribuzione geografica (destra) del numero di pubblicazioni aventi, come parole chiave o all'interno del titolo o dell'abstract, i termini Machine/Deep Learning, radiomics, o CADx/CADe (fonte: Scopus).*

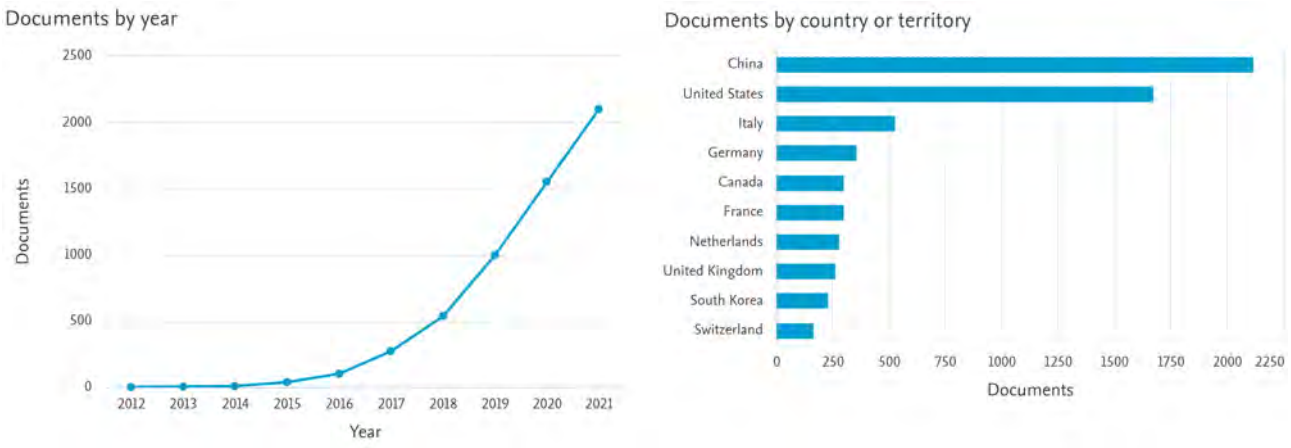

Figura 23: *Evoluzione nel tempo (sinistra) e distribuzione geografica (destra) del numero di pubblicazioni aventi, come parole chiave o all'interno del titolo o dell'abstract, il termine radiomics (fonte: Scopus).*

e Deep Learning (DL) a feti e neonati con CDH per sviluppare modelli di previsione in epoca prenatale, basati sull'analisi integrata di dati clinici, per fornire risultati prognostici quali il PH neonatale *in primis* e, possibilmente:

- risposta favorevole alla *Fetoscopic Endoluminal Tracheal Occlusion*, o FETO: si tratta di un intervento chirurgico avente la finalità di creare un'occlusione nella trachea del feto, per impedire ai fluidi prodotti dai polmoni di sfuggire attraverso di essa; è stato infatti dimostrato che il fluido, se trattenuto nelle vie aeree, aumenta la pressione nella trachea che stimola a sua volta lo sviluppo dei polmoni, altrimenti destinati a restare piccoli e insufficientemente sviluppati;
- necessità di *ExtraCorporeal Membrane Oxygenation*, o ECMO: è un trattamento postnatale che, mediante una macchina che assume le funzioni dei polmoni e del cuore, stabilizza il neonato;
- sopravvivenza all'ECMO;
- morte post-natale.

Inoltre, il protocollo prevede di produrre un sistema di segmentazione polmonare (semi)automatica del feto in risonanza magnetica (MRI), utile durante l'implementazione del progetto (per il calcolo di *feature* radiomiche e per limitare l'ambito di applicazione degli algoritmi di ML e DL) ma importante anche per standardizzare le misure di volume polmonare dei feti. Il sistema sarà sviluppato sulla base dell'approccio DL denominato 3D U-NET. Essendo il fegato l'organo principalmente coinvolto nella erniazione, è prevista eventualmente anche la realizzazione di un sistema di segmentazione automatica del fegato.

Il progetto prevede di arruolare pazienti con CDH isolato da gravidanze singole, i cui controlli prenatali siano stati effettuati presso l'Unità

di Chirurgia Fetale della Fondazione IRCCS Ca' Granda Ospedale Maggiore Policlinico (Milano) dalla 30ma settimana di gestazione. La ricerca sarà in retrospettiva, con la raccolta di variabili cliniche e radiologiche (MRI) di madri e neonati nati su un arco di 9 anni. I dati saranno integrati e analizzati con algoritmi previsionali in ML e DL. Essendo la patologia rara e quindi il *dataset* relativamente limitato, si prevede di adottare metodi di *data augmentation* e di riduzione della dimensionalità (selezione ed estrazione delle *feature*) allo scopo rispettivamente di aumentare la dimensione del campione ed evitare *overfitting*.

Tra le variabili considerate come *feature*, citiamo alcuni parametri radiologici (volumi polmonari, volume epatico, angolo di spostamento mediastinico o MSA e coefficiente medio di diffusione apparente, ADC, dei polmoni e del fegato del feto; altre grandezze derivano da queste, come per esempio la percentuale di fegato erniato %LH) e variabili cliniche, come quelle derivanti dalle ecografie prenatali (parametri Doppler relativi alla circolazione) e dall'ecocardiogramma.

Le figure 24 e 25, tratte dall'articolo, mostrano rispettivamente l'aspetto della segmentazione polmonare (manuale) nelle immagini di Risonanza Magnetica T_2 del feto con evidente iposviluppo polmonare, e il fegato, erniato attraverso il diaframma verso il torace. La figura 26 rappresenta lo schema di principio della rete neurale convoluzionale che verrà implementata.[2] In particolare si nota come nella CNN gli strati di uscita siano adattati al caso della classificazione binaria, ossia con due sole classi.

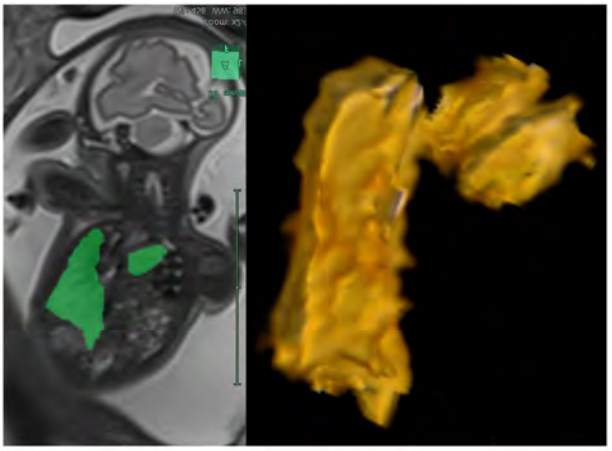

Figura 24: *Segmentazione dei polmoni in immagini MRI fetali e ricostruzione 3D, da [25].*

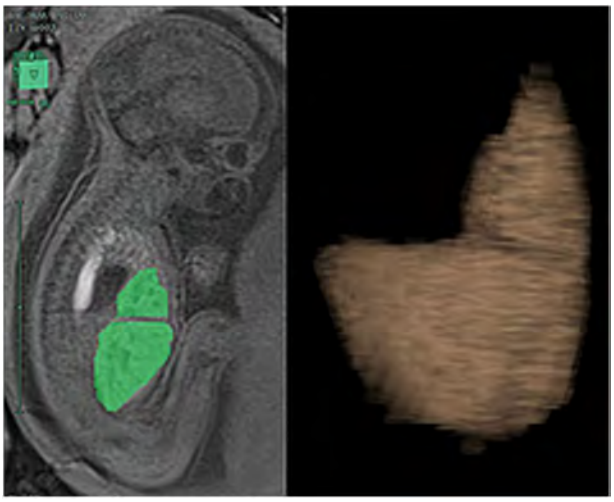

Figura 25: *Segmentazione del fegato in immagini MRI fetali e ricostruzione 3D, da [25].*

Discriminazione tra tipi diversi di tumore alla mammella

Il lavoro descritto in [26] si inserisce nel filone dell'IA come supporto alla diagnosi e riguarda il tumore della mammella. Il cancro al seno è in tutto il mondo la principale causa di morte per cancro nelle donne. Questo tumore aggressivo può essere classificato in due gruppi principali, *in situ* o infiltrante, a loro volta suddivisi in sottotipi. Il tumore infiltrante è tra le lesioni maligne più comuni (il carcinoma duttale invasivo costituisce il 70% di tutti i casi maligni). L'*imaging* per risonanza magnetica ha dimostrato grande sensibilità nel rilevamento e nella discriminazione tra lesioni benigne e maligne, se interpretato da radiologi esperti. Conoscere precocemente il tipo di tumore è fondamentale per orientare la terapia e anche allo scopo di dare alla paziente l'opportuno supporto psicologico nei casi in cui la gravità del tumore lo suggerisca. Lo scopo di questo studio è stato lo sviluppo di un sistema software in grado di differenziare automaticamente tra tumori *in situ* e infiltranti in immagini di risonanza magnetica con mezzo di contrasto (DCE-MRI), sulla base della firma radiomica della lesione. L'importanza di un sistema siffatto

[2] Figura modificata da **https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53**, sito visitato in novembre 2021

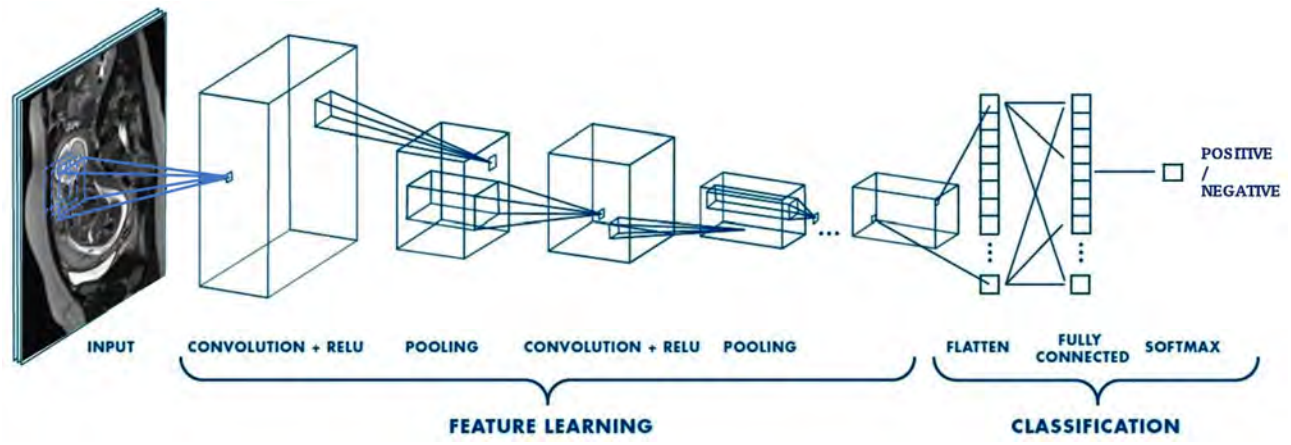

Figura 26: *Schema della rete convoluzionale, da [25]. Gli strati di uscita sono adattati al caso della classificazione binaria, ossia con due sole classi, mentre le reti adoperate per il riconoscimento di oggetti, naturali o artificiali, hanno usualmente circa un migliaio di classi d'uscita.*

consiste nella possibilità di riconoscere il tipo di tumore senza dover effettuare prelievi bioptici e quindi in maniera totalmente non invasiva e precoce.

L'algoritmo è costituito da due livelli di elaborazione principali: (1) localizzazione di possibili regioni di interesse tumorale (ROI) attraverso una procedura iterativa basata su valori di intensità dei livelli di grigio (il cosiddetto ROI *Hunter*), seguita da un metodo di estrazione e classificazione di *feature* avente lo scopo di scartare i falsi positivi, e (2) caratterizzazione delle ROI selezionate e discriminazione tra tumore *in situ* e invasivo, consistente nell'estrazione di *feature* radiomiche e loro classificazione attraverso un algoritmo di ML.

Il sistema CAD è stato sviluppato e valutato utilizzando un *database* di dimensioni limitate (55 pazienti) di immagini DCE-MRI provenienti dal Presidio Ospedaliero "Di Summa - Perrino" di Brindisi e contenenti almeno una massa confermata per immagine, come diagnosticato da un radiologo esperto. Le immagini avevano `voxel size` diverso l'una dall'altra, per cui è stato necessario inserire un modulo di *preprocessing* per rendere i dati omogenei.

Il primo *step* di elaborazione, ossia l'individuazione e la segmentazione delle masse tumorali, segue lo schema usuale dell'identificazione di *candidati tumore* con la massima sensibilità possibile, seguita dalla *rejection* dei falsi positivi per aumentare la specificità. Nel caso particolare, le *feature* da associare ai candidati sono state ricavate da una CNN preallenata, secondo la procedura seguente. I candidati sono stati esplorati con finestre 30×30 *pixel*, ridimensionate (per compatibilità con la CNN) a 224×224 *pixel* utilizzando interpolazione bilineare, convertite in immagini RGB attraverso la replica del bitplane dell'immagine, e infine date come *input* a una rete `GoogleNet` preallenata sul *dataset* `ImageNet`. La rete produce 1000 *feature* che sono state ridotte mediante *recursive feature selection* a 200 e utilizzate per l'allenamento di una rete neurale artificiale multistrato *feed-forward* allenata con *backpropagation*. La procedura di *training/validation* è stata eseguita con lo schema LOPO.

Una volta individuati i tumori, è stato realizzato il secondo *step* dell'elaborazione, ossia la discriminazione tra i due tipi di lesione: infiltrante e *in situ*. Le regioni di interesse sono state modellizzate da 1820 *feature* tessiturali calcolate da un *software* scritto in linguaggio *python* e basato sulla già citata libreria *pyradiomics*. Le *feature* comprendevano variabili basate sulla forma e *feature* tessiturali del primo ordine e di ordine superiore (da matrici GLCM, GLRLM, GLDM, GLSZM, per le quali si rimanda alla documentazione di *pyradiomics*). Dette variabili sono state calcolate direttamente e previa applicazione di diversi filtri di *pre-processing* quali: *Laplacian of Gaussian* (che evidenzia i bordi), *Wavelet*, e altri, compreso il filtro *Local Binary Pattern*. È stata applicata una fase di *feature selection* per eliminare variabili ridondanti e irrilevanti.

La figura 27 riporta, sulla sinistra, lo schema

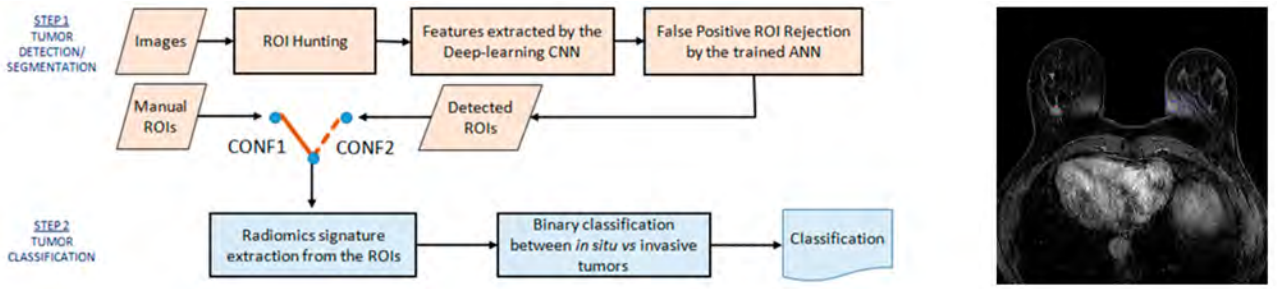

Figura 27: *Sinistra: schema del software di rilevamento e classificazione del tumore della mammella; destra: esempio di detection di una massa tumorale e del riconoscimento di un falso positivo, rispettivamente in rosso e blu (da [26]).*

del *software*.

La sensibilità dell'algoritmo al rilevamento delle masse è risultata essere del 75%, quindi migliorabile (vedere, in figura 27, sulla destra, un esempio di detection di una massa tumorale e del riconoscimento di un falso positivo, rispettivamente in rosso e blu). L'AUC della curva ROC per la discriminazione tra tumori *in situ* e infiltrativi è stata pari a $0,70$: lontana da valori ottimali ma paragonabile a quanto ottenuto da altri ricercatori.

Individuazione di noduli polmonari

In [27], e in alcuni degli articoli ivi citati, è descritto un sistema CAD per l'individuazione automatica dei noduli polmonari in immagini TC a bassa dose. L'esempio è stato scelto perché, oltre a presentare molti elementi descritti nella prima parte di questo lavoro (segmentazione, calcolo di *feature* per la discriminazione, classificazione), pone anche l'accento sull'interesse dell'uso del *cloud* per rendere i sistemi facili da gestire e manutenere. Infatti, l'approccio più comune nello sviluppo dei *software* è l'implementazione di stazioni di lavoro *standalone*, dotate di interfacce grafiche utente (GUI) sviluppate autonomamente (con un costo non indifferente dovuto alle licenze per i *software* di sviluppo e alle necessità di aggiornamento). Un'alternativa è il *cloud computing*, accessibile tramite protocolli Web sicuri, e coniugato con l'approccio SaaS o *Software as a Service*.

Come motivazione del lavoro, occorre porre l'attenzione sul fatto che il rilevamento visivo, da parte del medico radiologo, dei noduli polmonari nelle immagini TC a bassa dose è particolarmente difficile a causa del tasso di rumore nelle immagini. L'utilizzo di sezioni TC sottili per migliorare la risoluzione produce un numero di *slice* 2D compreso tra 300 e 500, il che rende il lavoro di analisi visiva lungo e soggetto a errori. Inoltre, i noduli possono essere di dimensioni molto ridotte e collegati alla superficie della pleura o addossati a un vaso, il che rende l'identificazione ancora più difficile. La disponibilità di *software* che assistano durante la diagnosi può quindi essere un utile supporto a vantaggio dell'accuratezza.

Il sistema presentato, realizzato nell'ambito di un progetto finanziato dalla Commissione Scientifica Nazionale 5 dell'INFN (Istituto Nazionale di Fisica Nucleare), è composto da tre blocchi principali: WIDEN *(Web-based Image and Diagnosis Exchange Network)* gestisce il flusso di lavoro, il caricamento dell'immagine e la notifica del risultato CAD; la *batch farm* IaaS *(Infrastructure as a Service)* basata su `OpenNebula`, che alloca risorse di elaborazione e archiviazione virtuali; il CAD M5L che fornisce la funzionalità di rilevamento dei noduli. L'implementazione proposta gestisce in modo sicuro i dati sensibili dei pazienti, poiché le immagini vengono trasferite con il protocollo HTTPS e la *batch factory* sottostante è isolata. Inoltre è efficiente poiché scala dinamicamente in base alle richieste degli utenti grazie al *backend* fornito dal *cloud*.

Entrando nel dettaglio, WIDEN consente ai medici di scambiare studi di *imaging* e confrontare le diagnosi. Il servizio è disponibile come applicazione *Web* accessibile via *browser* con credenziali adeguate. I file in formato DICOM[3] pos-

[3] *Digital imaging and COmmunication in Medicine*, standard

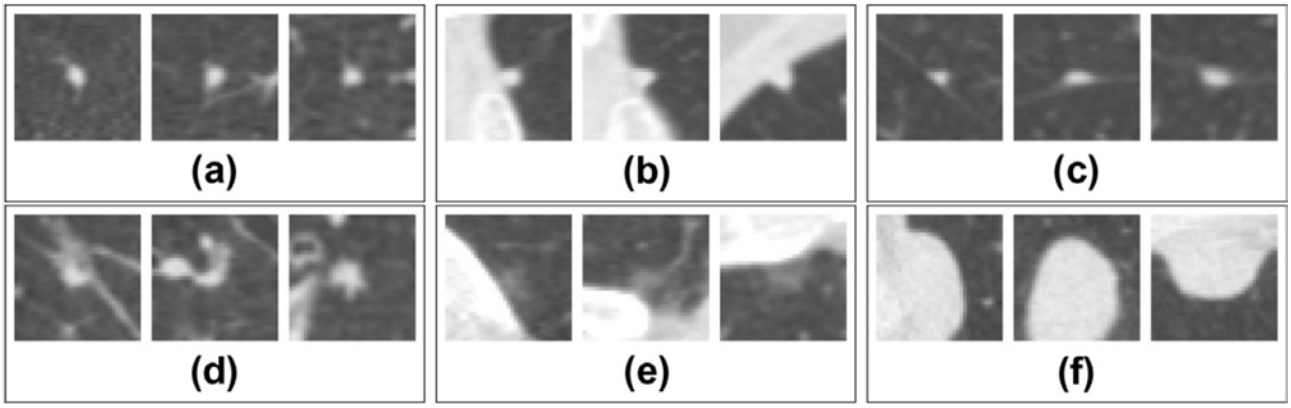

Figura 28: *Esempi di noduli polmonari. In ogni riquadro è visualizzato un nodulo in vista sagittale, coronale e assiale, 35* `voxel` *(circa 25 mm) attorno al punto centrale. La riga superiore mostra tre piccoli noduli, (a) un nodulo isolato di 4,4 mm; (b) un nodulo pleurico di 4,2 mm e (c) un nodulo peri-fissurale di 4,8 mm (la linea sottile visibile su ciascuna vista è la fissura). La seconda fila mostra tre grandi noduli, (d) un nodulo di 5,9 mm con adesione vascolare; (e) un nodulo ground-glass di 5,4 mm (relativamente raro) e (f) un grande nodulo pleurico (18,4 mm). (da [28], immagine parziale).*

sono essere caricati nell'archivio remoto: i medici possono caricare la propria diagnosi e vedere quelle di altri medici. WIDEN dispone di un sistema di notifica basato su *e-mail* e SMS che informano i medici iscritti quando sono aggiunte o aggiornate informazioni al caso di studio.

Le immagini caricate sono elaborate dagli algoritmi di calcolo, a loro volta gestiti come *plug in* dal sistema di *cloud computing* `OpenNebula` che consente il cosiddetto *elastic computing*, ovvero l'ottimizzazione delle risorse a seconda del carico computazionale.

Il progetto MAGIC-5 dell'INFN ha dimostrato che la combinazione di analisi ottenute da algoritmi diversi e complementari può massimizzare il risultato: il sistema implementato combina quindi tre algoritmi con diversi approcci: il Channeler Ant Model (CAM), l'analisi neurale basata su `voxel` (Voxel-Based Neural Approach, VBNA)e il Region Growing Volume Plateau o RGVP (per i dettagli, si rimanda naturalmente agli articoli specifici). Gli algoritmi hanno uno stadio comune, la segmentazione 3D del volume parenchimale, che separa trachea, bronchi e polmoni. Il risultato della segmentazione è salvato come maschera che, sovrapposta al parenchima polmonare, limita spazialmente la ricerca dei candidati noduli.

I tre algoritmi forniscono elenchi di candidati noduli con un certo tasso di falsi positivi, da ridurre mediante un classificatore (una rete neurale multistrato *feed-forward*). Le *feature* calcolate per i candidati sono piuttosto semplici e comprendono variabili basate sia sulla forma della ROI, sia sulle intensità dei grigi al suo interno, sia sui livelli di grigio dei *pixel* vicini; tra queste: volume della ROI, intensità massima, intensità media, deviazione standard dell'intensità, sfericità, intensità dei vicini, autovalori della matrice Hessiana, matrice dei gradienti. In particolare, gli autovalori della matrice Hessiana hanno la proprietà di descrivere bene la geometria locale e quindi di essere legati alla forma degli oggetti visibili nelle immagini.

In figura 29 sono riportate le curve FROC *(Free-Response ROC Curve)* relative ai risultati dei singoli sistemi di *detection*, e della loro combinazione. La FROC[30] è un analogo della ROC in cui l'ascissa riporta il numero di FP per immagine (ovvero per scansione TC) mentre l'ordinata misura la sensibilità. Al contrario della ROC, la FROC non è allocata in uno spazio finito. Essa è talora più utile perché serve a identificare la soglia di uscita del classificatore in base al compromesso scelto tra il contenimento del numero di FP per paziente, e la sensibilità.

Come si vede dal grafico, la combinazione dei sistemi di *detection* risulta migliore dei singoli algoritmi.

per la comunicazione, la visualizzazione, l'archiviazione e la stampa di informazioni di tipo biomedico come le immagini diagnostiche.

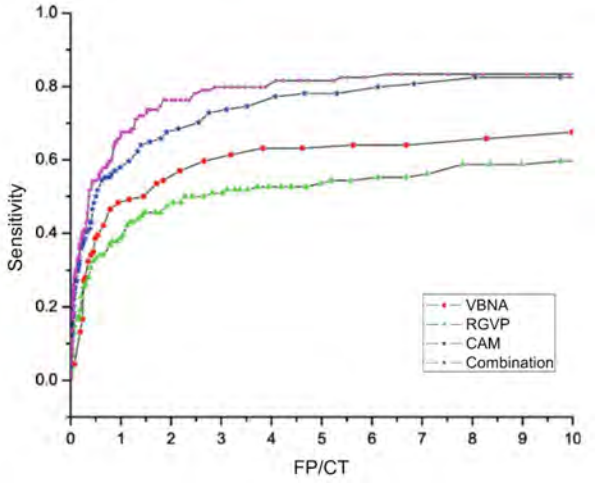

Figura 29: *Le curve FROC ottenute con la combinazione di CAD confrontate con quelle ottenute con i singoli sistemi nel range di FP di interesse radiologico (0–10 FP/CT): in rosso sono riportate le curve VBNA CAD, RGVP CAD e CAM CAD, verde e blu, rispettivamente; la combinazione è mostrata in viola. Da [29].*

Segmentazione di gliomi cerebrali

I gliomi sono i tumori cerebrali primari più comuni. Una diagnosi precoce e una valutazione completa dell'estensione del tumore e della relazione con le strutture anatomiche circostanti sono cruciali nel determinare la prognosi e la pianificazione del trattamento. La crescita diffusa e infiltrativa dei gliomi cerebrali è un importante determinante di prognosi infausta: le cellule tumorali invadono i tessuti circostanti preferenzialmente lungo i tratti di sostanza bianca, diffondendosi oltre l'area anormale individuabile nelle immagini di risonanza magnetica convenzionale. Pertanto, quest'ultima non sempre consente una precisa delineazione dei margini del tumore o la differenziazione del tumore dall'edema o dagli effetti della terapia. Tuttavia l'individuazione e la caratterizzazione delle infiltrazioni microscopiche in maniera non invasiva è di fondamentale importanza per il trattamento chirurgico e la pianificazione della radioterapia o per valutare la risposta alla chemioterapia. L'*imaging* in tensore di diffusione (*Diffusion Tensor imaging*, o DTI, o DT-MRI), al contrario della MRI convenzionale, può identificare anomalie peritumorali della sostanza bianca nei gliomi cerebrali rilevando la presenza di piccole aree con infiltrazione di cellule tumorali attorno al bordo del tumore.

Il DTI può dare informazioni sulle microstrutture cerebrali poiché fornisce contrasto all'immagine in base alle differenze nell'ampiezza della diffusione delle molecole d'acqua all'interno del cervello nelle varie direzioni. Il DTI stima la diffusione delle molecole d'acqua in ciascun `voxel` e produce quindi mappe scalari,[31] tra cui anisotropia frazionaria (FA), la diffusività media (MD), la diffusività assiale (AD), la diffusività radiale (RD), e le mappe p e q. Questi parametri descrivono la microstruttura e l'integrità della materia bianca e grigia nel cervello in un modo molto specifico che consente di monitorare la progressione di una malattia o di studiare lo sviluppo del cervello. In particolare, è stato dimostrato che il DTI è un *marker* sensibile di danno alla sostanza bianca.

Obiettivo del lavoro descritto in [32] è stato caratterizzare il tessuto patologico e sano in DTI mediante analisi tessiturale statistica in 3D (*feature* di Haralick e altre variabili), con lo scopo di sviluppare un CAD per l'individuazione e il contornamento automatico (segmentazione) dei tumori cerebrali. L'utilità del sistema è ad esempio la possibilità di misurare automaticamente (e perciò senza incorrere nell'altrimenti inevitabile variabilità inter- e intra-operatore) un tumore durante la terapia.

Sono stati selezionati quindici pazienti con glioma (9 di basso grado, 6 di alto grado) e sei pazienti sani di controllo dell'Ospedale Vita-Salute San Raffaele (Milano). Sono state acquisite immagini in DTI a 3T. Sono state calcolate svariate mappe di diffusione utilizzando un *software* implementato in `Matlab`: sia mappe di isotropia, come MD e la mappa p, sia di anisotropia, come FA e la mappa q. In seguito i tumori sono stati segmentati manualmente da radiologi esperti in tutte le mappe calcolate. *Feature* tessiturali in 3D sono state calcolate nelle regioni di interesse tumorali segmentate e (al fine di identificare le variabili discriminanti) nelle regioni controlaterali, in particolare le *feature* del primo ordine (dall'istogramma dell'intensità dei livelli di grigio) e del gradiente dei grigi, delle matrici di co-occorrenza (di Haralick) e delle matrici di percorrenza (RLM, *Run Length Matrix*)[33]. Il calcolo è stato effettuato con il *software* `MaZda`.[34] Si è utilizzato un approccio a finestra mobile (le ROI ipsilaterali e controlaterali sono state suddivise in piccoli sotto-volumi sovrapposti in cui è stato fat-

to il calcolo, associando poi il vettore di variabili al *pixel* centrale, da classificare). La scelta delle variabili più discriminanti per ogni mappa è stata poi effettuata in base al coefficiente di Fisher, già citato nella prima parte di quest'articolo.

Per eliminare la ridondanza delle informazioni, la dimensionalità dello spazio delle *feature* è stata ulteriormente ridotta utilizzando la PCA e conservando l'insieme minimo di componenti principali alle quali competesse almeno il 97% della varianza (due o tre componenti principali). Questa procedura non è in generale necessariamente fruttuosa perché la PCA non tiene conto del potere discriminante delle variabili ma, in questo caso, si è rivelata utile.

Il classificatore, allenato mediante una procedura di *hold out*, è stato una rete *feed-forward* con *back-propagation*, implementata in `Matlab` tramite `PRTools`, una libreria di *pattern recognition* liberamente disponibile. L'architettura di rete consisteva in un livello nascosto con tre neuroni, e un neurone di uscita. La qualità della classificazione dei *pixel* dell'insieme di validazione, misurata mediante l'area sotto la curva ROC, è stata > 0.96 per tutte le mappe considerate. In figura 30 è visibile il confronto tra una segmentazione automatica e la corrispondente manuale. La figura 1 già vista nella prima parte dell'articolo mostra altre segmentazioni ricavate mediante il sistema descritto (non riportate in [32]).

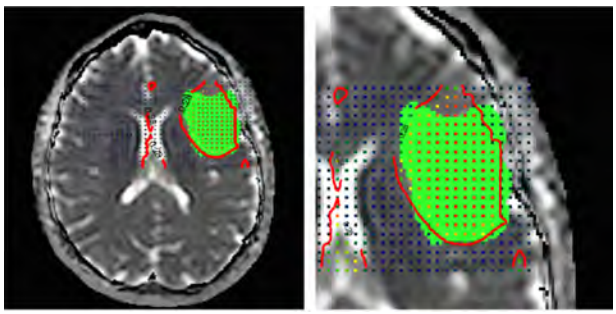

Figura 30: *Segmentazione automatica e manuale di una lesione: i punti segnano le posizioni dei centri della finestra mobile durante l'esplorazione delle immagini, la linea rossa mostra la segmentazione prodotta dal sistema CAD e la regione verde è la corrispondente ROI disegnata manualmente.*

Conclusioni

Le applicazioni dell'Intelligenza Artificiale alla Medicina sono ormai tantissime, alcune già in uso nella pratica clinica presso presìdi ospedalieri d'avanguardia. Tutto fa presagire che il progredire delle tecniche di *Machine/Deep Learning* e la disponibilità di adeguata potenza di calcolo rendano inevitabile la transizione verso una Medicina sempre più supportata da sistemi automatici con finalità diagnostiche e previsionali. Una diagnosi precoce, non invasiva e di qualità, è di beneficio per il paziente ma anche per il Sistema Sanitario Nazionale, per la potenziale riduzione delle spese non necessarie (si pensi ad esempio al risparmio derivante dalla diminuzione del numero di biopsie e dalla mancata prescrizione di esami clinici superflui). Le applicazioni prognostiche dell'IA sono anch'esse estremamente importanti, perché permettono ai medici di orientare meglio le terapie e di essere preparati alla probabile evoluzione di una patologia, garantendo una risposta più pronta quando necessario.

Si tratta delle Medicina del futuro, sempre più personalizzata, sempre più di precisione, in cui figure diverse e una volta scarsamente comunicanti (il medico, il fisico, il matematico, l'ingegnere) stabiliscono un linguaggio comune e collaborano per il bene del paziente.

Appendice 1: le *feature* di Haralick

Supponiamo che la ROI di cui vogliamo caratterizzare la *texture* sia costituita da (o strettamente contenuta in) un rettangolo I di dimensioni $N_r \times N_c$. Facciamo riferimento alla figura 31, in particolare all'immagine sulla sinistra, di dimensioni 5×5, in cui i *pixel* sono rappresentati da caselle con fondo grigio più o meno scuro e dal valore corrispondente. I livelli di grigio che compongono l'immagine siano quantizzati in un numero pari a N (ovvero, siano contenuti nell'insieme $G = \{0, \ldots, N-1\}$; nel caso particolare, $N = 4$). Ove l'immagine originale abbia un numero maggiore di livelli di grigio, è previsto che questi siano riducibili a un valore inferiore

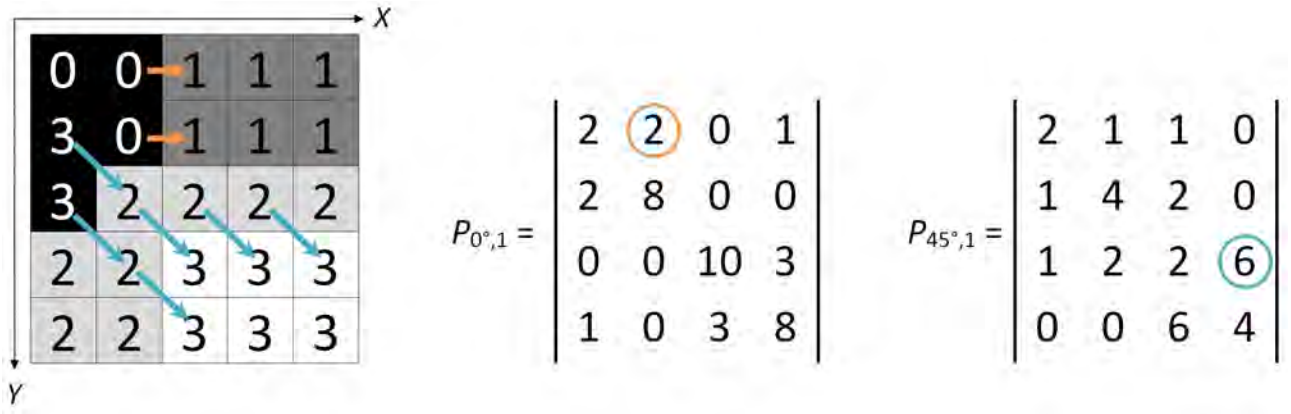

Figura 31: *Costruzione delle matrici di co-occorrenza dei livelli di grigio (GLCM).*

arbitrario, allo scopo di rendere le matrici meno sparse.

Ogni elemento della GLCM $P_{\alpha,d}$, ovvero $P^{ij}_{\alpha,d}$ per $(i,j) \in \{0,\dots,N-1\} \times \{0,\dots,N-1\}$, è calcolato come il numero di occorrenze di coppie di *pixel* distanti d nella direzione individuata dall'angolo α, aventi livelli di grigio (i,j) o (j,i) (quindi, indipendentemente dall'ordine). Ciò equivale (a meno di un fattore di normalizzazione) a valutare la probabilità congiunta che una coppia di punti, che soddisfano le condizioni di distanza d e fissata una direzione α, abbia livelli di grigio (i,j) o (j,i).

Le GLCM $P_{\alpha,d}$ avranno quindi dimensione $N \times N$ e dipenderanno da:

- la distanza d (generalmente posta pari a 1) tra i *pixel* delle coppie utilizzate per il calcolo; la metrica usata è la distanza di Chebyshev: $d(p_1,p_2) = \max_k(|p_{1,k} - p_{2,k}|)$, dove p_1 e p_2 sono i *pixel* considerati e $p_{1,k}$ è la k–ma coordinata del *pixel* p_1;
- l'orientamento α del segmento che congiunge i due *pixel*, espresso come angolo rispetto a un asse di riferimento e usualmente pari a: $0°$, $45°$, $90°$ e $135°$.

Si noti che la formulazione delle GLCM adoperate per il calcolo delle feature di Haralick porta a matrici simmetriche, perché al computo dell'elemento $P^{ij}_{\alpha,d}$ contribuiscono sia le coppie (i,j) che le coppie (j,i).

Ad esempio, nella figura 31 si può notare come l'immagine sia caratterizzata da quattro livelli di grigio, indicati come 0, 1, 2, 3. Nella matrice di co-occorrenza $P_{0°,1}$, ossia con angolo $0°$ e $d = 1$, l'elemento $P^{01}_{0°,1}$ è calcolato dal numero di occorrenze (ossia compresenze) tra *pixel* con valori di grigio 0 e 1, a distanza 1 in direzione x (angolo pari a $0°$). Osservando i valori di grigio dell'immagine, si possono contare due co-occorrenze tra *pixel* 0 e 1 in questa direzione e distanza 1 *pixel* (indicati in colore arancio). Per questo motivo $P^{01}_{0°,1}$ (ma anche $P^{10}_{0°,1}$) è pari a 2.

Stesso ragionamento per $P^{23}_{45°,1}$ (indicato in celeste) e per tutti gli altri elementi delle due matrici. Il *set* delle GLCM è completato da $P_{90°,1}$ e $P_{135°,1}$. Nel caso in cui l'immagine di *input* sia tridimensionale, è sufficiente dichiarare un opportuno *set* di angoli rispetto agli assi, che corrispondano alle direzioni nello spazio.

Dalle matrici di co-occorrenza si calcolano le 19 *feature* tessiturali di Haralick, per le quali si rimanda agli articoli originali [8, 9]. Solo a titolo esemplificativo, eccone due. L'energia, o misura del grado di omogeneità della *texture*, o momento angolare di secondo ordine, è definita come:

$$ENE_{\alpha,d} = \sum_i \sum_j P^{ij}_{\alpha,d}.$$

Valori alti di E corrispondono a tessiture omogenee, ovvero in cui la maggior parte delle coppie di *pixel* (fissati α e d) hanno livelli di grigio simili. Valori bassi competono a situazioni meno omogenee.

L'entropia, definita secondo la formula seguente:

$$ENT_{\alpha,d} = -\sum_i \sum_j P^{ij}_{\alpha,d} \cdot \log P^{ij}_{\alpha,d}$$

è alta quando i valori $P^{ij}_{\alpha,d}$ sono equidistribuiti. L'entropia ha valori bassi se, per esempio, la matrice di co-occorrenza è diagonale, ossia esistono

coppie di livelli di grigio uguali dominanti per una certa direzione e distanza.

Lista degli acronimi

%LH	*Percent of Liver Herniation*
AD	*Axial Diffusivity*
AUC	*Area Under the Curve*
AUROC	*Area Under the ROC Curve*
CAD	*Computer-Assisted Detection/Diagnosis*
CADe	*Computer-Assisted Detection*
CADx	*Computer-Assisted Diagnosis*
CDH	*Congenital Diaphragmatic Hernia*
CNN	*Convolutional Neural Network*
COM	*CoOccurrence Matrices*
CT	*Computed Tomography*
CV	*Cross Validation*
DCE-MRI	*Dynamic Contrast-Enhanced MRI*
DICOM	*Digital Imaging and COmmunications in Medicine*
DL	*Deep Learning*
DT-MRI	*Diffusion Tensor MRI*
DTI	*Diffusion Tensor Imaging*
ECG	Elettrocardiogramma
ECMO	*ExtraCorporeal Membrane Oxygenation*
EEG	Elettroencefalogramma
FA	*Fractional Anisotropy*
FETO	*Fetoscopic Endoluminal Tracheal Occlusion*
FLDA	*Fisher Linear Discriminant Analysis*
FN	*False Negative*
FP	*False Positive*
FROC	*Free-Response Operating Characteristic*
GLCM	*Gray-Level Cooccurrence Matrix*
GLDM	*Gray Level Dependence Matrix*
GLRLM	*Gray-Level Run-Length Matrix*
GLSZM	*Gray Level Size Zone Matrix*
GUI	*Graphical User Interface*
IA	Intelligenza Artificiale
IBSI	*Image Biomarker Standardisation Initiative*
ICA	*Independent Component Analysis*
LDA	*Linear Discriminant Analysis*
LOO	*Leave One Out*
LOO-CV	*Leave One Out Cross Validation*
LOPO	*Leave One Patient Out Cross Validation*
LOSO	*Leave One Subject Out Cross Validation*
MD	*Mean Diffusivity*
ML	*Machine Learning*
MRI	*Magnetic Resonance Imaging*
PCA	*Principal Component Analysis*
PET	*Positron-Emission Tomography*
PH	*Pulmonary Hypertension*
P-R	*Precision-Recall*
PR	*Pattern Recognition*
RD	*Radial Diffusivity*
ROC	*Receiver Operating Characteristic*
ROI	*Region of Interest*
SMOTE	*Synthetic Minority Oversampling TEchnique*
SVM	*Support Vector Machine*
TAC	Tomografia Assiale Computerizzata
TC	Tomografia Computerizzata
TN	*True Negative*
TP	*True Positive*
WIDEN	*Web-based Image and Diagnosis Exchange Network*

[1] G. Buttazzo, *Reti Neurali in grado di apprendere*, Ithaca, XVI (2020) 109.

[2] P. Lambin et al., *Radiomics: Extracting more information from medical images using advanced feature analysis*, European Journal of Cancer, 48 (2012) 441.

[3] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification, 2nd Edition*, Wiley, New York, (2000).

[4] H. Liu, H. Motoda (Editors), *Feature Extraction, Construction and Selection - A Data Mining Perspective*, Springer, Berlino, (1998).

[5] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, Englewood Cliffs NJ, (2007).

[6] A. Zwanenburg et al., *The Image Biomarker Standardization Initiative: Standardized Quantitative Radiomics for High-Throughput Image-based Phenotyping*, Radiology 295 (2020) 328.

[7] J. J. Griethuysen et al., *Computational Radiomics System to Decode the Radiographic Phenotype*, Cancer Research, 77 (2017) e104.

[8] R. M. Haralick, K. Shanmugan, I. Dinstein, *Textural Features for Image Classification*, IEEE Transactions on Systems, Man, and Cybernetics, SMC-3 (1973) 610.

[9] R. M. Haralick, *Statistical and structural approaches to texture*, Proc. IEEE, 67 (1979) 786.

[10] T. Ojala, M. Pietikäinen, D. Harwood, *Performance evaluation of texture measures with classification based on Kullback discrimination of distributions*, Proc. of the 12th IAPR International Conference on *pattern* Recognition (ICPR 1994), (1994) 582.

[11] M. Schmida, M. Vetterlia, K. Wegener, *Polymer powders for laser-sintering: Powder production and performance qualification*, AIP Conference Proceedings, 2065, 020008 (2019).

[12] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton NJ (1957).

[13] T. Hastie, R. Tibshirani, G. James, D. Witten, *An Introduction to Statistical Learning, with applications in R.*, Springer, Berlin (2013).

[14] M. Stone *Cross-Validatory Choice and Assessment of Statistical Predictions*, J. R. Stat. Soc. Ser. B, 36 (1974) 111.

[15] A. Bruce, K. Hopkin, A. Johnson, *L'Essenziale di Biologia Molecolare della cellula*, Zanichelli, Bologna, (2003).

[16] C. Bishop, *Neural networks and their applications*, Review of Scientific Instruments, 65 (1994) 1803.

[17] F. Rosenblatt, *The Perceptron—a perceiving and recognizing automaton*, Cornell Aeronautical Laboratory Report 85-460-1 (1957).

[18] W.S. McCulloch, W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, The bulletin of mathematical biophysics, 5 (1943) 115.

[19] D. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, Cambridge, (2005).

[20] R. Lippmann, *An introduction to computing with neural nets*, IEEE ASSP Magazine, 4 (1987) 4.

[21] G. W. Lindsay, *Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future*, J. Cogn. Neurosci., 33 (2021) 2017.

[22] D. H. Hubel, T. N. Wiesel, *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex*, The Journal of physiology, 160 (1962) 106.

[23] K. Fukushima *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position*, Biological cybernetics, 36 (1980) 193.

[24] J. A. Swets, *Measuring the accuracy of diagnostic systems*, Science, 240 (1998) 1285.

[25] I. Amodeo et al., *A Machine and deep Learning Approach to predict pulmoNary hyperteNsIon in newbornS with congenital diaphragmatic Hernia (CLANNISH): Protocol for a retrospective study*, PLoS ONE 16(11): e0259724 (2021).

[26] L. Conte et al., *Breast Cancer Mass Detection in DCE–MRI Using Deep-Learning Features Followed by Discrimination of Infiltrative vs. In Situ Carcinoma through a Machine-Learning Approach*, Appl. Sci., 10 (2020) 6109.

[27] D. Berzano et al., *On-demand lung CT analysis with the M5L-CAD via the WIDEN front-end web interface and an OpenNebula-based cloud back-end*, Proc. IEEE Nuclear Science Symposium and Medical *imaging* Conference Record (NSS/MIC) (2012) 978.

[28] B. van Ginneken et al., *Comparing and combining algorithms for computer-aided detection of pulmonary nodules in computed tomography scans: The ANODE09 study*, Medical Image Analysis, 14 (2010) 707.

[29] N. Camarlinghi et al., *Combination of computer-aided detection algorithms for automatic lung nodule identification*, Int J Comput Assist Radiol Surg., 7 (2012) 455.

[30] A. I. Bandos et al., *Area under the free-response ROC curve (FROC) and a related summary index*, Biometrics, 65 (2009) 247.

[31] A. Peña et al., *Enhanced visualization and quantification of magnetic resonance diffusion tensor imaging using the p:q tensor decomposition*, The British journal of radiology 79 (2006) 101.

[32] G. De Nunzio et al., *A CAD system for cerebral glioma based on texture features in DT-MR images*, Nucl. Instrum. Methods Phys. Res. A, 648 Suppl. 1 (2011) S100.

[33] G. Castellano, L. Bonilha, L. M. Li, F. Cendes, *Texture analysis of medical images*, Clin. Radiol., 59 (2004) 1061.

[34] M. Strzelecki, et al., *A software tool for automatic classification and segmentation of 2D/3D medical images*, Nucl. Instrum. Methods Phys. Res. A, 702 (2013) 137.

Giorgio De Nunzio: laureato in Fisica presso l'Università di Lecce, ha conseguito il dottorato all'Université de Montpellier II. Professore Aggregato in Fisica Applicata (FIS/07) del Dipartimento di Matematica e Fisica "Ennio De Giorgi" dell'Università del Salento, si occupa principalmente di Fisica e Informatica per la Medicina (in particolare, elaborazione di immagini e segnali di interesse biomedicale, e intelligenza artificiale per la realizzazione di sistemi automatizzati di supporto decisionale). Interessi secondari sono la Fisica e l'Informatica applicate ai Beni Culturali.