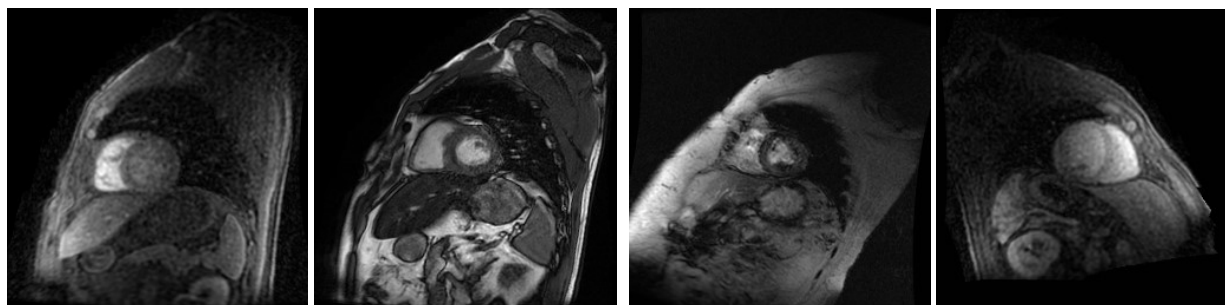


Scopo dell'esercitazione è utilizzare una convolutional neural network (CNN) per la classificazione di immagini biomediche. In particolare, il set di dati comprende immagini di risonanza magnetica cardiaca, acquisite in asse corto con diverse sequenze di acquisizione.



Perfusione

Funzione

T2*

LGE

Le sequenze utilizzate mirano alla valutazione della perfusione cardiaca con primo passaggio del mezzo di contrasto, alla valutazione della funzione cardiaca (sequenza fast-cine), alla misura del valore di T2* con sequenze multi-echo, ed alla valutazione dell'infarto miocardico attraverso sequenze con contrasto ritardato (LGE).

Nell'elaborazione delle immagini cardiache è necessario identificare la posizione della fetta in asse corto rispetto all'asse ventricolare, in particolare ad ogni fetta deve essere assegnata una label (basale, media o apicale) in modo da poter "riempire" il modello di riferimento AHA mediando le informazioni delle varie fette appartenenti alla stessa macro-regione. Questa operazione viene di solito fatta per via manuale. Quello che vogliamo è realizzare una CNN che riconosca in modo automatico la classe (basale, media, apicale) a cui appartengono le fette.

La base di conoscenza è quindi formata da tre gruppi di immagini contenute in tre cartelle che ne definiscono la posizione. Le immagini sono classificate secondo la loro posizione lungo il ventricolo, come basali, medie e apicali, seguendo la classificazione standard AHA.

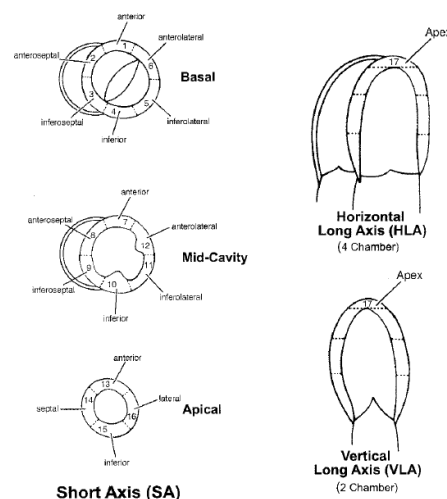


Figure 3. Diagram of vertical long-axis (VLA, approximating the 2-chamber view), horizontal long-axis (HLA, approximating the 4-chamber view), and short-axis (SA) planes showing the name, location, and anatomic landmarks for selection of the basal (tips of the mitral valve leaflets), mid-cavity (papillary muscles), and apical (beyond papillary muscles but before cavity ends) short-axis slices for the recommended 17-segment system. All imaging modalities should use these same landmarks, when available, for slice selection.

Le immagini sono contenute nella cartella TEST_MR che contiene tre cartelle Base, Middle, Apical con le immagini DICOM corrispondenti. Quindi il ground truth è definito dalla posizione delle immagini nelle cartelle.

Attraverso la funzione **imageDatastore** possiamo accedere ai dati ottenendo un dataset di 251 immagini per ogni classe. Essendo le immagini in formato DICOM che non è compreso nei formati standard della funzione *imageDatastore* (forse nelle ultime versioni Matlab si) dobbiamo definire una funzione che importi le immagini come:

```
function I = readFcn_DICOM(filename)
    I = ....;
end
```

e una volta definito il datastore definire la funzione di input delle immagini:

```
datastore.ReadFcn=@readFcn_DICOM;
```

Come notiamo le immagini MR hanno un FOV molto più grande del cuore che è l'oggetto di interesse, quindi è opportuno che la funzione readFcn_DICOM comprenda una fase di pre-processing, in cui converrà eseguire un cropping su tutte le immagini in modo da isolare l'area centrale in cui è presente il cuore stesso. Inoltre le immagini date in input alla rete devono avere tutte le stesse dimensioni quindi dovremo eseguire una interpolazione. Quindi in fase di pre-processing dobbiamo implementare le seguenti operazioni:

- Lettura delle immagini in formato DICOM
- Cropping delle immagini DICOM in modo da includere solo il cuore in tutte le immagini
- Interpolazione in modo da ricondurre tutte le immagini alla stessa dimensione.

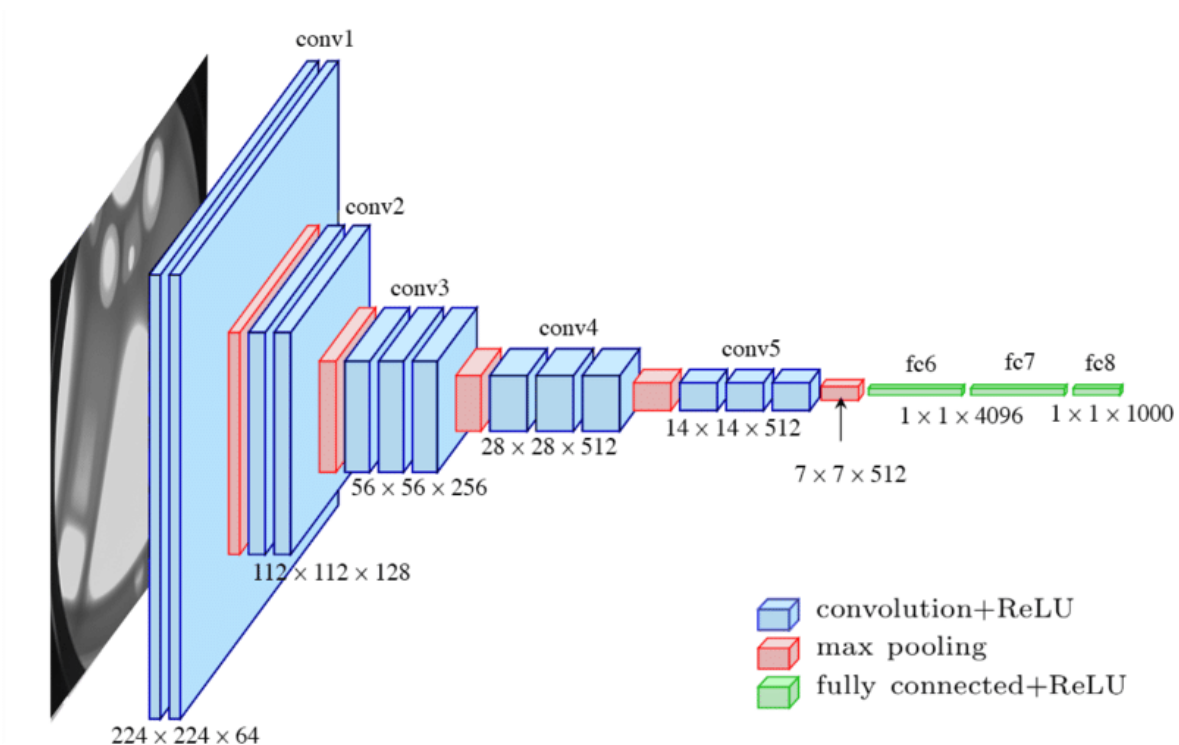
Il dataset andrà diviso in una parte di training (la maggioranza delle immagini) ed una parte di test che verrà usato alla fine per verificare le prestazioni della rete.

Inoltre è opportuno crearsi un validation set che verrà usato nella fase di training (cross-validation) per validare iterativamente la rete, attraverso l'opzione 'ValidationData' della funzione trainingOptions. Quindi divideremo il dataset in tre parti con la funzione splitEachLabel, due uguali (validation e test set, ad esempio il 10-15%, ed una più grande che è il training set).

La rete avrà la classica struttura:

INPUT \rightarrow [[CONV \rightarrow RELU] *N \rightarrow POOL?] *M \rightarrow [FC \rightarrow RELU] *K \rightarrow FC

Dove l'input dipende dalla dimensione a cui sono state interpolate le immagini mentre il layer FC deve avere dimensione 3 come le classi che vogliamo identificare. La struttura degli strati profondi della rete va configurata tentando di ottimizzare il processo di apprendimento.



Il modello base per la classificazione delle immagini è il modello VGG mostrato in figura, composto da una serie di strati convolutivi che riducono la dimensione dell'immagine fino ad arrivare allo stadio di classificazione. In generale è opportuno che la riduzione di risoluzione dall'input all'output avvenga in modo continuo attraverso blocchi POOL con lo stesso stride. La profondità degli strati CONV di solito è dell'ordine del numero di classi, è inutile avere profondità molto più grandi del numero di classi. Altezza e larghezza degli strati CONV corrispondono alla regione di immagine "vista" dalla rete, quindi dipendono dalla grandezza delle immagini e dall'ordine dello strato. Comunque conviene usare più strati CONV in cascata ($N > 1$) piuttosto che ingrandire troppo il singolo strato CONV.

Una volta addestrata la rete vanno misurate le prestazioni sia sul training set che sul test set. In particolare valuteremo l'accuratezza della classificazione dalla confusion matrix (***plotconfusion***) che misura la capacità della rete di riconoscere correttamente le classi. Dalla confusion matrix è possibile valutare la sensibilità, specificità ed accuratezza della classificazione per le tre classi. Ad esempio per le immagini apicali avremo:

VP = Fette apicali riconosciute come tali

VN = Fette non apicali riconosciute come basali o medie

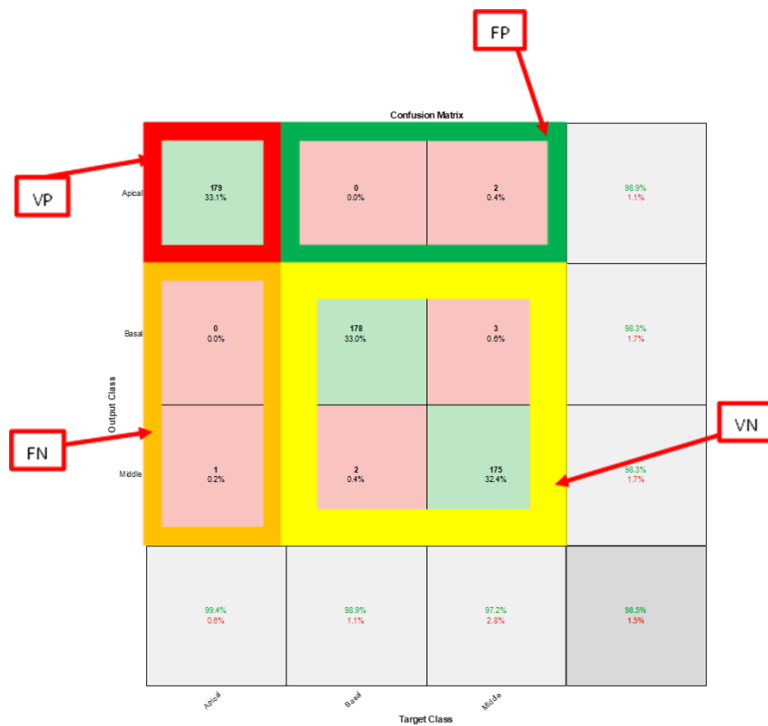
FP = Fette non apicali riconosciute come apicali

FN = Fette apicali non riconosciute come apicali

Sensibilità (Apice) = $VP/(VP+FN)$

Specificità (Apice) = $VN/(VN+FP)$

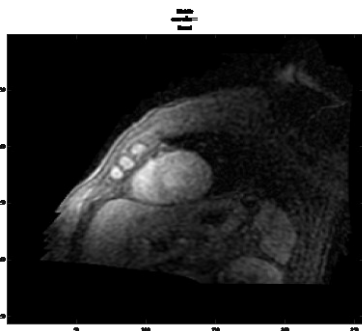
Accuratezza (Apice) = $(VN+VP)/(VN+VP+FN+FP)$



I valori di sensitività, specificità ed accuratezza sul training set se la rete è configurata correttamente dovrebbero essere molto alti (> 95 %).

La stessa analisi deve essere ripetuta sul test set. Anche qui i valori attesi per le prestazioni della rete sono elevati (> 90%).

Infine è utile visualizzare le immagini classificate in modo errato in modo da verificare il motivo di una incorretta classificazione, ad esempio in un run dell'algoritmo potremmo avere:



In questo caso l'immagine è stata classificata come middle ma il GT la classifica come basal. In realtà appaiono i muscoli papillari e quindi probabilmente la classificazione corretta è middle. D'altra parte il GT

è comunque affetto da un possibile errore dovuto alla variabilità inter- ed intra- osservatore della classificazione.