

POLITECNICO DI TORINO

Dipartimento di Ingegneria Meccanica e Aerospaziale

**Corso di Laurea Magistrale
in Ingegneria Biomedica**

Tesi di Laurea Magistrale

Segmentazione automatica di metastasi epatiche su immagini TC con convolutional neural networks



Relatori

prof. Gabriella Balestra

ing. Valentina Giannini

firma dei relatori

.....

.....

Candidato

Giuseppe Passanisi

firma del candidato

A.A. 2018/2019

Indice

Sommario	1
Introduzione	2
Materiali e metodi	6
Pazienti	6
Segmentazione fegato	7
Pre-processing e ricerca delle soglie	8
Ricerca del fegato	12
Criticità e miglioramenti	14
Unet	17
MultiUnet	21
Risultati	23
Unet	23
Numero di layer	23
Learning rate	24
Numero di features	24
Numero di epoche	25
Validazione e miglioramenti	27
MultiUnet	40
Discussioni	46
Bibliografia	47

Indice delle figure

Figura 1. Rappresentazione grafica del tumore del colon retto con metastasi al fegato	2
Figura 2. Curva Kaplan-Meier che mostra la percentuale di sopravvivenza in funzione dei mesi di pazienti con cancro colon rettale in confronto a vari casi metastatici (15)	2
Figura 3. Workflow dell' algoritmo ideato da Smeets et al.(12)	3
Figura 4. Figura che mostra i vari passi dell' algoritmo di ricerca di metastasi con cerchi iperdensi. (a) Individuazione della ROI scelta dall' utente. (b) Istogramma dei valori di grigio dei pixel all' interno della ROI. (c) Risultato della segmentazione della parte interna della metastasi. (d) Riempimento della maschera ottenuta nel passo precedente con valori tipici del bordo esterno (iperdenso). (e) Risultato finale.	4
Figura 5. Worflow dell' algoritmo di Vorontsov et al.	5
Figura 6. Confronto tra immagini standardizzate (tre superiori) e quelle normalizzate tramite min-max scaling (le tre inferiori)	7
Figura 7. (a) Maschera della slice centrale. (b) Eliminazione del lettino e relativo bordo da cui estrapolare il rene di lavoro della maschera scelta.	9
Figura 8. Slice centrale lungo il piano frontale.	9
Figura 9. Maschera dopo applicazione del filtro passabanda.....	10
Figura 10. Maschera dopo l' applicazione dell' operatore morfologico	10
Figura 11. Maschera dopo aver eliminato le regioni che hanno il baricentro che non rientra nell' area di interesse.	10
Figura 12. Immagine mascherizzata della slice relativa alla maschera con l' area più grande.	11
Figura 13. (a) Immagine dopo primo filtraggio. (b) Risultato dopo il primo operatore morfologico. (c) (d) Risultati dopo aver applicato l' operatore morfologico prima sul piano frontale (c), poi su quello sagittale (d). (e) Risultato finale dopo aver applicato l' ultimo operatore morfologico lungo il piano trasversale.....	13
Figura 14. Risultato finale dopo aver scelto la regione che contiene più pixel che rientrano nelle soglie trovate precedentemente.	13
Figura 15. Sovrapposizione contorno di partenza con la maschera ottenuta dai gradienti.	14
Figura 16. Esempio segmentazione fegato nel paziente 8.....	15
Figura 17. Esempio segmentazione fegato nel paziente 6.....	15
Figura 18. (a) Maschera finale e immagine mascherizzata finale (b).	15
Figura 19. Esempio segmentazione fegato nel paziente 17.....	16
Figura 20. Esempio segmentazione fegato nel paziente 17.....	16
Figura 21. Esempio segmentazione fegato nel paziente 26.....	17
Figura 22. Esempio di struttura della Unet a 4 layer, con 64 feature estrapolate nel primo layer di convoluzione.	18
Figura 23. Grafico della funzione ReLU.....	19
Figura 24. Dendrogramma ottenuto dai percentili. In rosso è segnata il livello di taglio.	20
Figura 25. Dendrogramma ottenuto dal rapporto tra l' area della metastasi e l' area del fegato. In rosso è segnato il livello di taglio.....	20
Figura 26. Dendrogramma ottenuto dalle aree della metastasi. In rosso è segnato il livello di taglio.....	21
Figura 27. Schema di funzionamento della MultiUnet.	21
Figura 28. Raggruppamenti dei vari cluster.	22
Figura 29. Andamento di media e mediana della precision al variare del numero di epoche..	26
Figura 30. Andamento di media e mediana del dice al variare del numero di epoche.....	26

Figura 31. Andamento di media e mediana della recall al variare del numero di epoche.	27
Figura 32. Andamento di media e mediana della precision al variare del numero di epoche..	29
Figura 33. Andamento di media e mediana della recall al variare del numero di epoche.	29
Figura 34. Andamento di media e mediana del dice al variare del numero di epoche.....	29
Figura 35. Andamento del dice medio al variare dell prove effettuate per ognuno dei metodi di costruzione di train e validation set.....	35
Figura 36. Andamento del numero di falsi negativi al variare dell prove effettuate per ognuno dei metodi di costruzione di train e validation set.....	35
Figura 37. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set tramite i cluster ottenuti dalle aree delle metastasi. In blu sono riportate le maschere manuali.	36
Figura 38. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set tramite i cluster ottenuti dalle aree delle metastasi. In blu sono riportate le maschere manuali	36
Figura 39. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set tramite i cluster ottenuti dalle aree delle metastasi. In blu sono riportate le maschere manuali	37
Figura 40. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set tramite i cluster ottenuti dai percentili. In blu sono riportate le maschere manuali	37
Figura 41. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set tramite i cluster ottenuti dai percentili. In blu sono riportate le maschere manuali.	38
Figura 42. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set tramite i cluster ottenuti dai percentili. In blu sono riportate le maschere manuali.	38
Figura 43. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set in maniera random. In blu sono riportate le maschere manuali.	39
Figura 44. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set in maniera random. In blu sono riportate le maschere manuali.	39
Figura 45. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set in maniera random. In blu sono riportate le maschere manuali.	40
Figura 46. Andamento del dice medio nel train e nel validation, al variare del numero di epoche, nel cluster 1	42
Figura 47. Andamento del dice medio nel train e nel validation, al variare del numero di epoche, nel cluster 2	42
Figura 48. Andamento del dice medio nel train e nel validation, al variare del numero di epoche, nel cluster 5.	43
Figura 49. Andamento del dice medio nel train e nel validation, al variare del numero di epoche, nel cluster 4	43
Figura 50. Esempio di funzionamento dell'uscita della MultiUnet. (a) è la maschera di uscita del cluster 1, (b) del cluster 2, (c) del cluster 4 e (d) del cluster 5. (f) è il risultato finale.	44
Figura 51. Confronto tra output della MultiUnet (in rosso) e maschera manuale (in blu).....	45
Figura 52. Confronto tra output della MultiUnet (in rosso) e maschera manuale (in blu).....	45
Figura 53. Confronto tra output della MultiUnet (in rosso) e maschera manuale (in blu).....	45

Indice delle tabelle

Tabella 1. Tabella che mostra il numero di metastasi epatiche totali, divise per paziente, con relativo diametro medio.....	6
Tabella 2. Valore dei massimi e minimi percentili per ogni paziente, con relativa soglia scelta.	8
Tabella 3. Valori di grigio minimo e massimo per ogni cluster e relativa numerosità, per ogni paziente.....	11
Tabella 4. Valori di soglia scelta con relativo valore medio che identifica il fegato in ogni paziente.....	12
Tabella 5. Valori, per ogni layer, di media, deviazione standard, mediana, 25° e 75° percentile per dice, precision e recall.	23
Tabella 6. Valori, per ogni valore di learning rate, di media, deviazione standard, mediana, 25° e 75° percentile per dice, precision e recall.	24
Tabella 7. Valori, per ogni numero di feature da estrapolare nel primo layer di convoluzione, di media, deviazione standard, mediana, 25° e 75° percentile per dice, precision e recall.	25
Tabella 8. Valori, per ogni numero di epoche, di media, deviazione standard, mediana, 25° e 75° percentile per dice, precision e recall.	25
Tabella 9. Valori ottenuti dalla validazione di media, deviazione standard, mediana, 25° e 75° percentili dei valori di dice, precision e recall. In basso a destra sono mostrati il numero di falsi negativi totali.....	27
Tabella 10. Valori ottenuti dopo aver cambiato il dataset. Sono riportati la media, deviazione standard, mediana, 25° e 75° percentile di dice, precision e recall al variare del numero di epoche.....	28
Tabella 11. Valori ottenuti dalla validazione di media, deviazione standard, mediana, 25° e 75° percentili dei valori di dice, precision e recall. In basso a destra sono mostrati il numero di falsi negativi totali.....	30
Tabella 12. Confronto delle validazioni tra i vari metodi di costruzione di train e test set utilizzati.....	30
Tabella 13. Valori ottenuti dalla validazione, senza la costruzione dei cluster, di media, deviazione standard, mediana, 25° e 75° percentili dei valori di dice, precision e recall per ogni prova.....	31
Tabella 14. Valori ottenuti dalla validazione, con la costruzione dei cluster ottenuti tramite i percentili, di media, deviazione standard, mediana, 25° e 75° percentili dei valori di dice, precision e recall per ogni prova.	32
Tabella 15. Valori ottenuti dalla validazione, con la costruzione dei cluster ottenuti tramite il rapporto tra le aree delle metastasi con quelle del fegato, di media, deviazione standard, mediana, 25° e 75° percentili dei valori di dice, precision e recall per ogni prova.....	33
Tabella 16. Valori ottenuti dalla validazione, con la costruzione dei cluster ottenuti tramite le aree delle metastasi, di media, deviazione standard, mediana, 25° e 75° percentili dei valori di dice, precision e recall per ogni prova.....	34
Tabella 17. Valori del numero di falsi negativi ottenuti da ogni prova in ognuno dei metodi di costruzione del data set.....	34
Tabella 18. Andamento nel cluster 1 di media e mediana del train e del validation.....	40
Tabella 19. Andamento nel cluster 2 di media e mediana del train e del validation.....	41
Tabella 20. Andamento nel cluster 4 di media e mediana del train e del validation.....	41
Tabella 21. Andamento nel cluster 5 di media e mediana del train e del validation.....	41

Tabella 23. Valori di media, deviazione standard, mediana, 25° e 75° percentile di dice, precision e recall della validazione. In basso a destra sono riportati il numero di falsi negativi.
..... 44

Sommario

Il cancro del colon-retto è il terzo cancro più comune al mondo in termini di incidenza e il quarto nella mortalità. Il fegato è l'organo più comune in cui si presentano delle metastasi nei pazienti con tumore coloretale. Circa il 14-18% dei pazienti con carcinoma del colon-retto presentano metastasi alla prima visita medica. Il problema di quest'ultime è che sono differenti tra loro, quindi rispondono in maniera diversa alla terapia, quindi risulta necessario caratterizzarle, per trovare la migliore strategia di cura. Per poter fare ciò bisogna segmentare le metastasi epatiche dalle immagini acquisiti. La segmentazione può avvenire sia in maniera manuale che in maniera automatica. La prima richiede troppo tempo, quindi si cercano dei metodi per automatizzare la segmentazione di metastasi. Lo stato dell'arte ci fornisce metodi che sono per la maggioranza semiautomatici o funzionano male su lesioni scarsamente contrastate e disomogenee.

L'obiettivo di questa è riuscire a segmentare in maniera totalmente automatica le metastasi del fegato, tramite l'utilizzo di una rete neurale, nello specifico la Unet.

Per fare ciò è stato prima attuato un pre-processing dell'immagine ideando un algoritmo di segmentazione automatica del fegato, in modo da fornire alla rete neurale solo le informazioni più rilevanti.

L'algoritmo di segmentazione si basa sull'individuazione del fegato tramite delle soglie. Questi valori vengono trovati in maniera automatica, andando inizialmente a ricercare l'organo di interesse sull'asse frontale, sfruttando la posizione nota, e applicando un k-means.

Segmentato il fegato viene costruito il data set utilizzato nella Unet. Vengono considerate solo le fette che hanno metastasi, si effettua un crop nell'intorno del fegato e vengono usate due normalizzazioni, in modo da avere due data set. Uno verrà normalizzato tramite standardizzazione, l'altro viene prima uniformato, vengono cambiati i valori di grigio in modo tale che tutti i pazienti rappresentino il fegato alla stessa intensità, e poi viene normalizzato tramite *min-max scaling*.

Creata i data set si passa all'allenamento della Unet. Quest'ultima ha un'architettura basata su una *fully convolutional network*, divisa in un due parti: down-sampling e up-sampling.

Vengono effettuate varie prove per andare a trovare il miglior valore ad ogni parametro che caratterizza la rete. La validazione viene effettuata dividendo il data set e usando il 70% come training set e il restante 30% come test set.

Per migliorare i risultati sono stati testati altri due metodi di costruzione del test e train set: in uno vengono calcolati il 10°, 25°, 50°, 75° e 90° percentile e costruito il dendrogramma, nell'altro viene calcolato il rapporto tra l'area della metastasi (l'area della maschera manuale fornita) e la relativa area del fegato, anche in questo caso viene costruito il dendrogramma. Vengono applicati tagli differenti ai due dendrogrammi e si divide il data nei corrispondenti cluster. Il training set verrà costruito usando il 70% di ogni cluster, il test set dal 30% restante. Vengono validati anche questi due sistemi e si effettuano ulteriori prove dai cluster ottenuti dai percentili. Sempre partendo da quest'ultimi verranno allenate un numero di reti pari al numero di cluster, ognuno con gli elementi di quest'ultimi.

Introduzione

Il cancro del colon-retto è il terzo cancro più comune al mondo in termini di incidenza e il quarto nella mortalità(1), immediatamente dietro al polmone, al fegato e al cancro allo stomaco(2). Occupa il secondo tipo più comune di cancro nelle donne e il terzo negli uomini, rappresenta il 9,7% in tutto il mondo di tutti i tipi di cancro, escluso il cancro della pelle non melanomatoso. Il fegato è l'organo più comune in cui si presentano delle metastasi nei pazienti con tumore coloretale, a causa della sua situazione anatomica per quanto riguarda la circolazione(3). Circa il 14-18% dei pazienti con carcinoma del colon-retto presentano metastasi alla prima visita medica(4) e dal 10 al 25% sono diagnosticati al momento della resezione del tumore primario. In generale, circa il 70% dei pazienti con tumore del colon-retto svilupperà metastasi nel fegato(5).

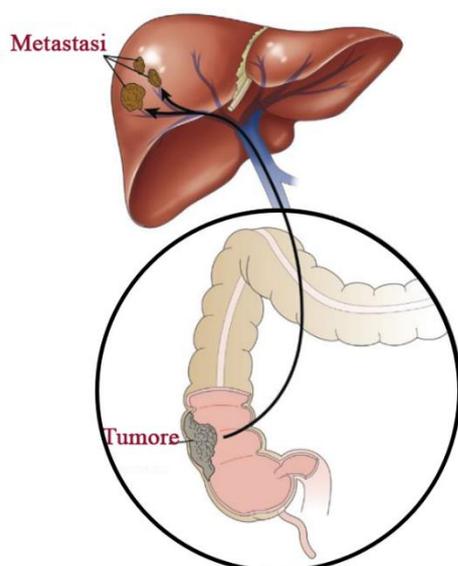


Figura 1. Rappresentazione grafica del tumore del colon retto con metastasi al fegato

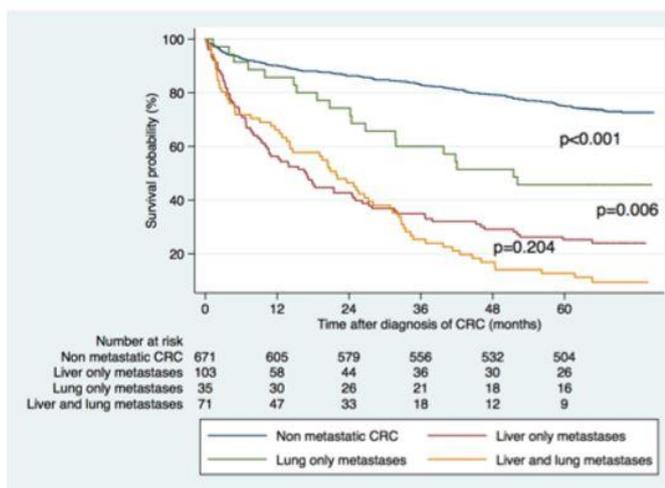


Figura 2. Curva Kaplan-Meier che mostra la percentuale di sopravvivenza in funzione dei mesi di pazienti con cancro colon retto in confronto a vari casi metastatici (15)

Il problema di quest'ultime è che sono differenti tra loro, quindi rispondono in maniera diversa alla terapia. Una soluzione, per individuare il miglior trattamento da applicare caso per caso, sarebbe quella di caratterizzarle. Per fare ciò bisogna prima segmentare le metastasi dalle CT ottenute. È possibile farlo in due modi:

- Segmentazione manuale;
- Segmentazione automatica.

La prima richiede troppo tempo, quindi sarebbe interessante trovare degli algoritmi che riescano a segmentare in maniera automatica le metastasi del fegato.

Lu et al (6) utilizzano i contorni attivi 2D che vengono inizializzati manualmente. Questo è relativamente lento e non molto accurato. Altri autori si concentrano sull'individuazione di metastasi in un fegato segmentato e applicano metodi piuttosto semplici per la segmentazione della lesione. Park et al.(7) e Ciecholewski e Ogiela (8) tentano entrambi di trovare soglie ottimali basate sull'analisi statistica dell'istogramma del fegato, ma questo metodo fallisce su lesioni scarsamente contrastate o disomogenee. Li et al. (9) applica una tecnica di machine learning per classificare le possibili posizioni dei contorni della metastasi su profili di intensità 1-D. Anche in questo caso i bassi contrasti e soprattutto le lesioni in contatto con una struttura della stessa densità porranno problemi.

Un altro approccio è stato presentato da Jolly e Grady (10). Il loro algoritmo è adatto a diversi tipi di lesioni, incluse le metastasi epatiche. Apprende la distribuzione dei valori grigi di un tumore dall'interazione dell'utente e diverse segmentazioni 2D su piani ortogonali e applica infine il *random walker algorithm*. A causa del suo carattere generale, il metodo non è in grado di gestire casi tipici difficili, come le metastasi del fegato adiacenti a una struttura isodensa.

Una tecnica sviluppata da Smeets et al.(11) si basa sull'utilizzo del level set. Inizialmente, l'algoritmo richiede all'utente di posizionare un punto approssimativamente nel mezzo del tumore e di specificare un raggio massimo posizionando un punto nel tessuto epatico circostante. Successivamente, questo input viene utilizzato per generare l'inizializzazione del level set, utilizzando una tecnica di scansione a spirale. Il level set si evolve in base a un'immagine di velocità, che viene calcolata indipendentemente utilizzando la classificazione fuzzy. Dopo la segmentazione, il post-processing viene utilizzato per trasformare la mappa delle distanze in un'immagine binaria. Il workflow del metodo è riportato in fig. 3.

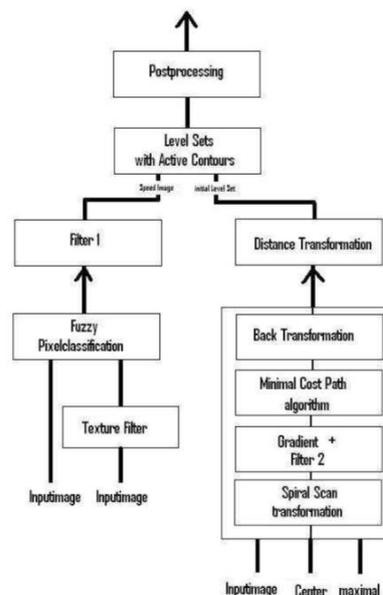


Figura 3. Workflow dell'algoritmo ideato da Smeets et al.(12)

L'algoritmo "Smart opening", sviluppato da Kuhnigk et al.(12) per la segmentazione di noduli polmonari solidi, può essere applicato anche nella segmentazione delle metastasi del fegato,

soprattutto nel caso in cui si hanno dei contorni iperdensi. L'idea del smart opening è quella di scegliere la resistenza all'erosione in modo adattivo. Per facilitare i calcoli, l'erosione viene implementata tramite la soglia su una mappa delle distanze che contiene la distanza di ogni voxel della maschera al voxel di sfondo più vicino. La seconda fase dell'operazione di apertura è una dilatazione che ricostruisce la lesione nella sua dimensione originale. La dilatazione viene nuovamente implementata mediante la soglia su una mappa delle distanze, ma questa volta mostra le distanze di tutti i voxel di sfondo dalla maschera erosa. La soglia viene scelta leggermente più alta della soglia di erosione, in modo che in una fase di affinamento tutti i dettagli del contorno possano essere riprodotti per intersezione con il risultato di crescita della regione. Questa procedura da sola non è adatta se sono presenti connessioni più estese a strutture di densità simile. Al fine di rendere l' algoritmo di apertura intelligente applicabile alle metastasi epatiche, è necessario apportare alcune modifiche. A causa dell'elevata diversità del loro aspetto, le soglie per la crescita della regione iniziale devono essere determinate in modo adattivo. Questo è basato su un'analisi della distribuzione della densità nella ROI. Le informazioni fornite dall'utente vengono riflesse nelle dimensioni e nel centro della ROI. Le metastasi non omogenee possono essere rilevate usando il tratto che l'utente deve disegnare attraverso la lesione.

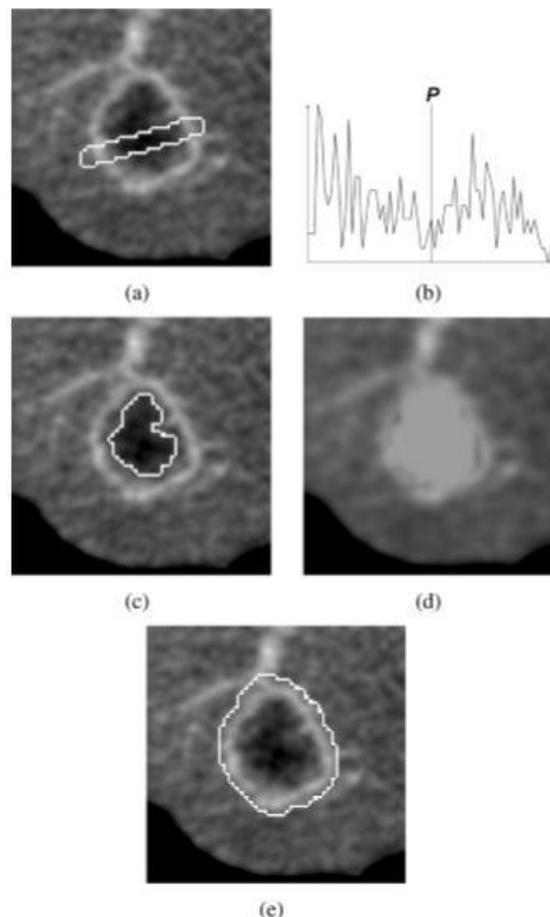


Figura 4. Figura che mostra i vari passi dell'algoritmo di ricerca di metastasi con cerchi iperdensi. (a) Individuazione della ROI scelta dall'utente. (b) Istogramma dei valori di grigio dei pixel all'interno della ROI. (c) Risultato della segmentazione della parte interna della metastasi. (d) Riempimento della maschera ottenuta nel passo precedente con valori tipici del bordo esterno (iperdenso). (e) Risultato finale.

Vorontsov et al. (13) propongono un metodo di segmentazione semiautomatica che utilizza l'apprendimento automatico all'interno di un modello di superficie deformabile. Nello specifico, si utilizza un modello deformabile che utilizza un classificatore voxel basato su un multilayer perceptron per interpretare l'immagine CT. Le classificazioni del voxel a 2 classi (tumore, non tumore) vengono utilizzate come input per un modello di superficie deformabile inizializzato dall'utente, selezionando due punti nell'immagine per posizionare una mesh sferica su una metastasi di interesse. Il nuovo modello di superficie deformabile include un termine adattativo che consente di adattare le forme irregolari ai contorni del tumore. Invece di utilizzare un classificatore addestrato su funzioni di immagine predefinite, il metodo utilizza un modello di rete neurale per l'apprendimento e la classificazione delle caratteristiche.

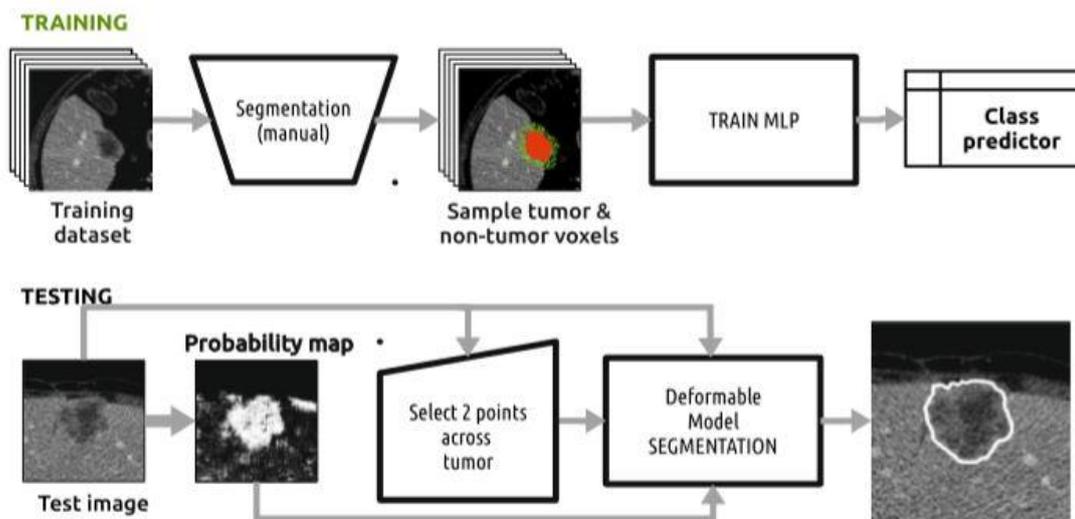


Figura 5. Workflow dell' algoritmo di Vorontsov et al.

Il limite principale degli ultimi tre metodi illustrati è che sono semi-automatici.

L'obiettivo di questa tesi consiste nel segmentare, in maniera totalmente automatica, le metastasi epatiche con l'utilizzo di reti neurali, in particolare con l'utilizzo della Unet. Per poter migliorar l'efficienza di quest'ultima, è stato implementato un algoritmo di segmentazione automatica del fegato, in modo tale da fornire meno informazioni inutili alla rete neurale.

Materiali e metodi

Pazienti

Il dataset utilizzato è composto da CT addominali prelevate da 23 pazienti nei seguenti ospedali: IRCCS di Candiolo, Ospedale Niguarda di Milano, l'Istituto Oncologico Veneto di Padova. Sono state fornite anche le relative maschere manuali delle metastasi. Il numero totale di quest'ultime è di 124, la suddivisione di tali lesioni nei vari pazienti e il proprio diametro medio è illustrato nella tabella seguente

Tabella 1. Tabella che mostra il numero di metastasi epatiche totali, divise per paziente, con relativo diametro medio.

Paziente	N° lesione	Diametro medio [mm]	Paziente	N° lesione	Diametro medio [mm]	Paziente	N° lesione	Diametro medio [mm]
1	1	24.30	10	43	11.60	26	84	19
	2	78.40		44	22.40		85	40
2	3	58		45	16.10		86	43
	4	19		46	10.90		87	121
4	5	82.2		47	11.80		88	28
5	6	90	12	48	23	27	89	30
	7	13		49	25		90	41
	8	13	18	50	23		91	38
	9	11		51	23		92	7
6	10	69.7		52	20		93	18
	11	77	53	18	94	34		
	12	73.6	54	34	95	17		
	13	16.1	55	18	96	18		
	14	22.9	56	28	97	22		
8	15	18.1	57	20	28	98	69.9	
	16	19.80	58	24		99	140	
	17	30.30	59	24		100	46.8	
	18	92.50	60	22		101	45.3	
	19	17.60	61	21		102	32.5	
	20	41.10	62	18	29	103	81	
	21	29.40	17	63	74	30	104	82
	22	60.00		64	8	31	105	32.4
	23	21.30		65	9		106	44.1
	24	25.40	19	66	31.90		107	28
	25	13.80		67	33.70		108	14.7
	26	16.00		68	21.60		109	21.2
	27	16.00		69	45.40	110	27	
	28	9.50	20	70	28.40	32	111	15.2
29	27.10	71		31.30	112		17	
30	17.10	72		9.10	113		24	
31	36.60	22	73	18.8	114		43	
32	33.90		74	68.2	115		17	
33	12.80	23	75	48.7	116	21		
34	31.50		76	11	117	17		
35	48.50		77	24	118	20		
36	23.20		78	45	119	26		
37	33.20	25	79	24	120	50		
38	21.20		80	34	33	121	65	
39	17.60	81	79	122		20		
40	15.10	82	68.2	123		9		
41	51.3	83	19.1	124		42		
42	42.4							

Per la parte relativa alla segmentazione del fegato sono state usate le immagini in formato NIfTI, in int16, di dimensione 512x512. Invece per la Unet sono state utilizzate le 570 slice che mostravano le 124 metastasi totali, in uint8, formato tif. In ogni immagine è stata effettuato un *crop* nell'intorno del fegato, anche nella relativa maschera manuale della metastasi, e riportate alla dimensione di 256x256. Sono stati creati due dataset in questo caso: uno dove le immagini, paziente per paziente, venivano normalizzate sottraendo la media totale dell'intensità dei pixel e dividendo per la relativa deviazione standard; l'altro effettuando inizialmente un "riallineamento" delle immagini, ovvero portando i pixel che rappresentano il fegato ad un valore unico e standard per i vari pazienti (+60 HU), e successivamente normalizzate con *min-max scaling*.

$$img_{standardizzato} = \frac{img - media(img)}{std(img)} \quad img_{normalizzato} = \frac{img - min(img)}{max(img)}$$

In entrambi i dataset le immagini sono state filtrate tramite un filtro passabanda con soglie pari a -100 e +250 di intensità.

Inoltre, per la Unet, inizialmente il test set era uguale al training set, successivamente il training set era formato dal 70% del dataset e il test dal 30%, presi in maniera random.

Tutti gli algoritmi spiegati successivamente sono stati sviluppati in ambiente Matlab®.

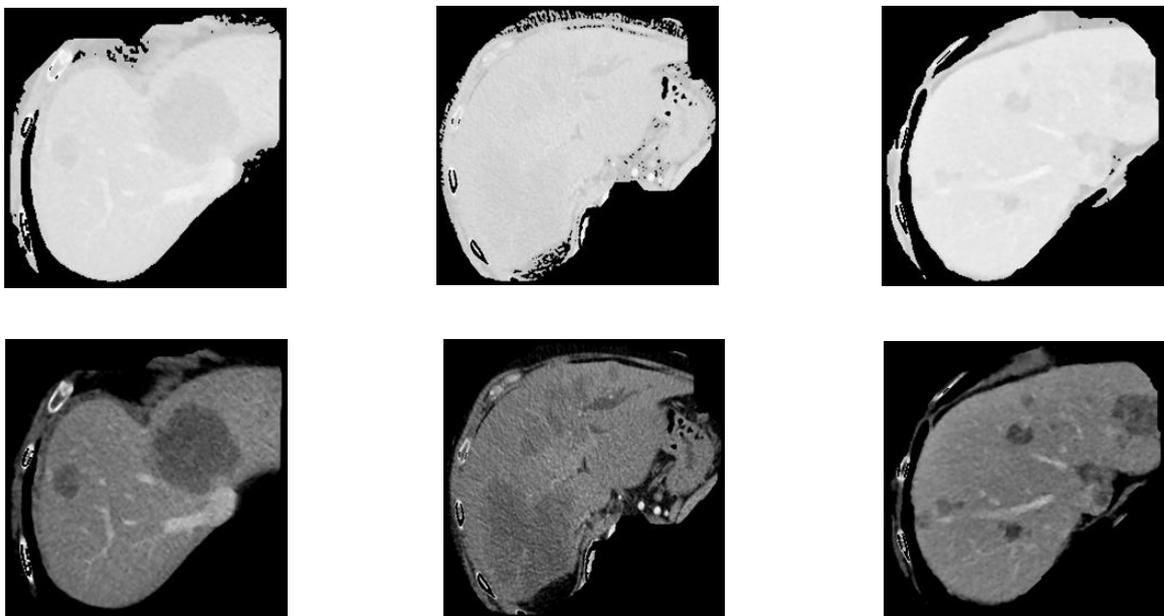


Figura 6. Confronto tra immagini standardizzate (tre superiori) e quelle normalizzate tramite min-max scaling (le tre inferiori)

Segmentazione fegato

Con lo scopo di migliorare le prestazioni della Unet è stato pensato di ridurre le informazioni da fornirgli, effettuando una segmentazione automatica del fegato partendo dalle immagini originali.

Si è visto che l'intensità dei pixel che rappresentano il fegato hanno un range abbastanza limitato, quindi l'algoritmo ricerca la regione all'interno dell'addome che ha il maggior numero di pixel che appartengono a questo range.

Pre-processing e ricerca delle soglie

La prima criticità riscontrata sta nel fatto che in ogni paziente il fegato viene rappresentato con intensità diverse rispetto agli altri, quindi la prima parte dell'algoritmo va ad individuare proprio questo range, per poter successivamente individuare il fegato.

Per fare ciò si ricerca quest'ultimo lungo il piano frontale, sfruttando la propria posizione nota e la dimensione dell'organo stesso. Come primo passo si individuano due valori di intensità tali da poter binarizzare le slice sul piano frontale. Pertanto, inizialmente ogni slice, sul piano trasverso, viene filtrata tramite un filtro mediano con kernel 3x3, in modo tale da uniformare i pixel che rappresentano i vari organi e successivamente viene calcolato il 65° percentile. Si utilizza tale valore perché individua dei valori plausibili di intensità che più si avvicinano a quelli del fegato. Una volta calcolato ogni percentile per slice si utilizzano il minimo e il massimo come valori di soglia per la binarizzazione successiva, entrambi rispettivamente sottratti e sommati di un valore pari a 30, solo per allargare l'intervallo e avere la sicurezza di prendere anche eventuali metastasi e parti più chiare del fegato.

Tabella 2. Valore dei massimi e minimi percentili per ogni paziente, con relativa soglia scelta.

Paziente	Minimi percentili	Massimi percentili	Soglia inferiore	Soglia superiore
1	46	137	16	167
10	82	250	52	280
12	52	119	22	149
17	55	158	25	188
18	61	162	31	192
19	72	181	42	211
2	95	223	65	253
20	53	172	23	202
22	65	149	35	179
23	92	190	62	220
25	66	146	36	176
26	61	179	31	209
27	72	186	42	216
28	46	181	16	211
29	61	148	31	178
30	63	170	33	200
31	65	191	35	221
32	62	156	32	186
33	106	135	76	165
4	53	104	23	134
5	50	133	20	163
6	61	170	31	200
8	68	205	38	235

Una criticità riscontrata nell'individuare il fegato lungo il piano frontale sta nel fatto che l'algoritmo potrebbe segmentare come l'organo di interesse la schiena o il lettino, visto che hanno intensità simili. Per questo si individua un intervallo di slice su cui effettuare la ricerca, tale intervallo viene individuato prendendo come riferimento la slice centrale sul piano trasverso, viene binarizzata con un threshold pari a -600, si trova il contorno della maschera con aerea più grande, in modo da eliminare già il lettino.

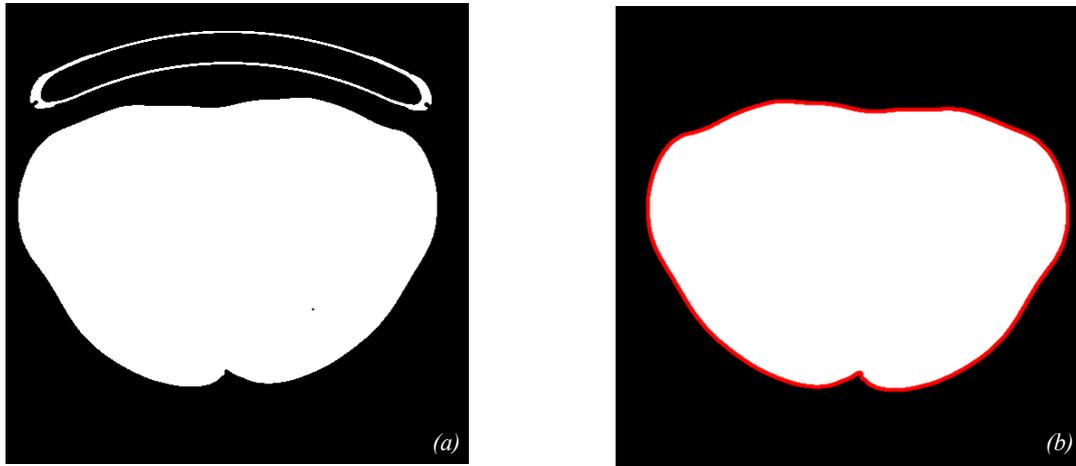


Figura 7. (a) Maschera della slice centrale. (b) Eliminazione del lettino e relativo bordo da cui estrapolare il range di lavoro della maschera scelta.

Da tale contorno viene determinato il l'intervallo di slice su cui viene rappresentato l'addome sul piano frontale, che va dal valore minimo al valore massimo delle ordinate del contorno. Il range delle slice su cui verrà effettuata la ricerca sarà il seguente:

$$(y_{max}-y_{min})*0,45+y_{min} : (y_{max}-y_{min})*0,9+y_{min}$$

Ottenuto l'intervallo di lavoro e le soglie per effettuare la binarizzazione, si passa sul piano frontale, effettuando una rotazione 3D di 90° lungo l'asse coronale.

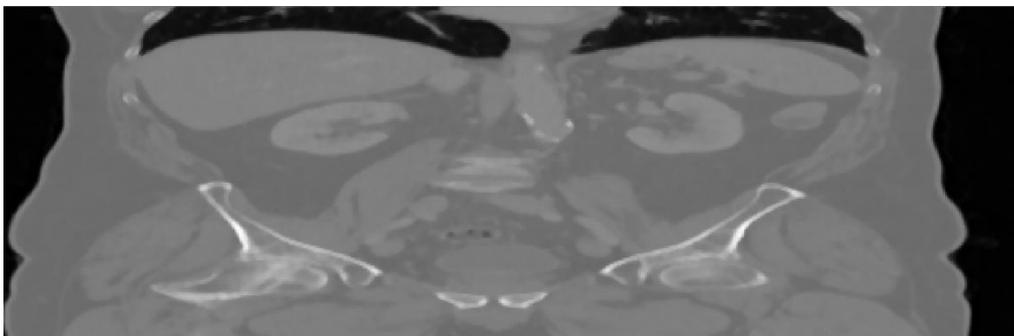


Figura 8. Slice centrale lungo il piano frontale.

L'immagine viene filtrata tramite un filtro mediano, questa volta con kernel 3D di dimensioni pari a 3x3x3. Si utilizza un filtro 3D per uniformare i pixel che caratterizzano i vari organi non solo all'interno di ogni slice, ma anche fetta per fetta. Vengono binarizzate le immagini con le soglie trovate precedentemente e successivamente viene applicato un operatore morfologico, ovvero una chiusura (dilatazione seguita da erosione) con elemento strutturale pari ad un disco con un diametro di 8 pixel. Si effettua questo passaggio in modo tale da eliminare varie regioni non di interesse di piccole dimensioni e chiudere quelle di grandi dimensioni.



Figura 9. Maschera dopo applicazione del filtro passabanda



Figura 10. Maschera dopo l'applicazione dell'operatore morfologico

In ogni slice quindi vengono trovati i bordi delle varie maschere trovate e vengono calcolati i baricentri. Le regioni che hanno quest'ultimi che non rientrano ad un intervallo che va da 134 mm a 250 mm verranno non considerate ed eliminate. Di quelle salvate vengono calcolate le proprie aree.



Figura 11. Maschera dopo aver eliminato le regioni che hanno il baricentro che non rientra nell'area di interessa.

Una volta effettuato questo procedimento su tutte le slice di interesse viene scelta come maschera di riferimento quella che ha l'area maggiore di tutte. Questa rappresenta, approssimativamente, la fetta centrale delle slice che rappresentano il fegato. La maschera così ottenuta viene moltiplicata per la rispettiva immagine, in modo tale da avere il fegato rappresentato dai propri pixel



Figura 12. Immagine mascherizzata della slice relativa alla maschera con l'area più grande.

Da questa immagine viene individuato l'intervallo di intensità di pixel che rappresenta il fegato. Per fare ciò viene applicato un k-means dei pixel contenuti all'interno della maschera. Vengono inizializzati i tre centroidi: uno sarà pari al 10° percentile, uno il 90° percentile e l'ultimo sarà pari alla media tra il 10° percentile e il 90°. Applicato il k-means si trovano 3 cluster, quello che ha il maggior numero di elementi sarà quello che identificherà il fegato. L'intervallo di intensità andrà dal minimo valore al massimo degli elementi del cluster considerato.

Tabella 3. Valori di grigio minimo e massimo per ogni cluster e relativa numerosità, per ogni paziente.

Paziente	Cluster 1			Cluster 2			Cluster 3		
	Minimo	Massimo	N° elementi	Minimo	Massimo	N° elementi	Minimo	Massimo	N° elementi
1	17	65	1726	66	110	5502	111	165	1101
10	53	136	1936	137	185	4249	186	279	3626
12	23	71	3033	72	104	11157	105	145	3140
17	26	81	1905	82	128	11242	129	184	5971
18	32	94	1162	95	127	4269	128	190	2409
19	43	110	964	111	158	6991	159	210	1394
2	66	136	648	137	182	5043	183	252	1671
20	24	80	7053	81	138	944	139	200	979
22	36	89	2581	90	131	8477	132	178	2896
23	63	123	1727	124	163	3604	164	219	3896
25	37	88	1859	89	127	5836	128	172	3198
26	32	103	3229	104	144	4375	145	206	2696
27	43	108	3991	109	145	1638	146	211	1228
28	17	57	11483	58	117	8109	118	211	1626
29	32	94	4279	95	133	5517	134	176	5312
30	34	84	1564	85	133	7900	134	197	1855
31	36	96	3646	97	146	1300	147	218	4817
32	33	95	2087	96	142	7384	143	186	3091
33	77	102	1355	103	122	2188	123	162	744
4	24	76	1175	77	100	3314	101	129	2519
5	21	86	2064	87	126	7344	127	163	2808
6	32	93	2305	94	147	6490	148	198	1260
8	39	104	3293	105	171	7930	172	234	1120

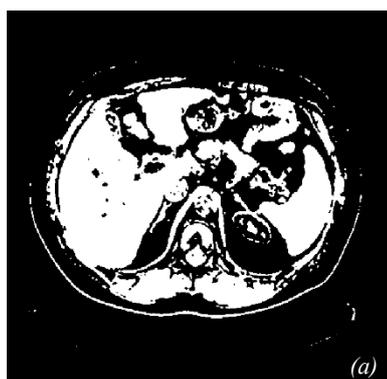
Tabella 4. Valori di soglia scelta con relativo valore medio che identifica il fegato in ogni paziente

Paziente	Valore medio del fegato	Soglia inferiore	Soglia superiore
1	85	66	110
10	184	137	185
12	90	72	104
17	100	82	128
18	124	95	127
19	139	111	158
2	166	137	182
20	56	24	80
22	105	90	131
23	176	164	219
25	110	89	127
26	135	104	144
27	93	43	108
28	54	17	57
29	121	95	133
30	101	85	133
31	161	147	218
32	121	96	142
33	107	103	122
4	95	77	100
5	105	87	126
6	119	94	147
8	138	105	171

Ricerca del fegato

Ottenute le intensità caratteristiche del fegato, l'algoritmo ritorna sul piano trasverso ed effettua un pre-processing sempre con un filtro mediano con kernel pari a 3x3, seguito da una binarizzazione usando dei valori di soglia pari a quelli individuati precedentemente, sottraendo 15 al valore minimo e sommando 30 a quello massimo.

Effettuato quest'ultimo filtraggio, si effettuano una chiusura con elemento strutturale pari al disco con diametro rispettivamente di 4. Per migliorare la continuità sulle slice delle maschere si effettuano di nuovo una apertura e una chiusura, con disco pari a 10 e 7 pixel di diametro, prima sul piano frontale e poi su quello sagittale. Ritornando sul piano trasverso si effettua un'ultima chiusura con disco di diametro pari nuovamente a 10 pixel.



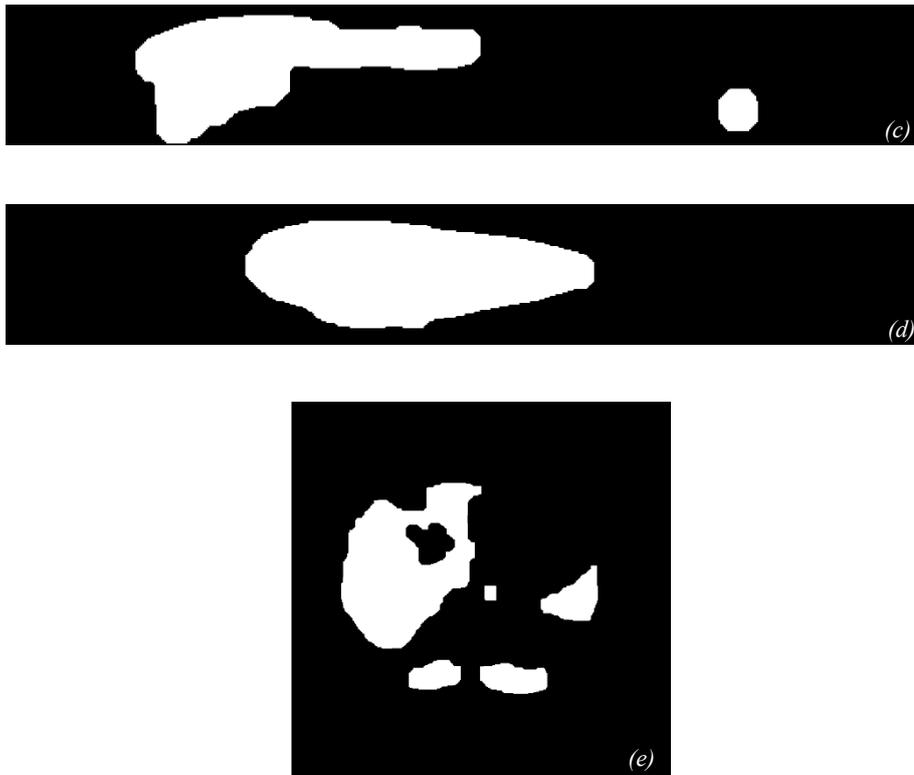


Figura 13. (a) Immagine dopo primo filtraggio. (b) Risultato dopo il primo operatore morfologico. (c) (d) Risultati dopo aver applicato l'operatore morfologico prima sul piano frontale (c), poi su quello sagittale (d). (e) Risultato finale dopo aver applicato l'ultimo operatore morfologico lungo il piano trasversale.

Quindi, in ogni slice l'algoritmo riconoscerà varie regioni, per ognuna di queste si effettuerà una ricerca, prima lungo le ordinate poi lungo le ascisse, dei pixel che hanno un'intensità che rientra nell'intervallo individuato dal k-means. La regione che risulta averne di più verrà considerata possibile fegato, e se ne calcola l'area, le altre vengono scartate.



Figura 14. Risultato finale dopo aver scelto la regione che contiene più pixel che rientrano nelle soglie trovate precedentemente.

Criticità e miglioramenti

Questo approccio, però, porta con sé due criticità:

1. Si riesce ad individuare molto bene il fegato quando ha un'area relativamente grande (slice centrale), mentre non individua molto bene le fette con area piccola (slice iniziali e finali);
2. L'area di segmentazione in media è sottostimata.

Per ovviare al primo problema si parte individuando la fetta che, verosimilmente, individuerrebbe la fetta centrale delle slice che rappresentano il fegato. Lo si fa scegliendo la maschera con l'area più grande. Partendo da quella, si effettuano due cicli, uno che va fino alla prima slice e l'altro fino all'ultima. In ognuno di questi due cicli viene calcolata la distanza di Hausdorff tra i contorni di tutte le geometrie, riconosciute dalla binarizzazione, presenti nella slice e il contorno della segmentazione del fegato della slice precedente/successiva; viene mantenuta solo la maschera che ha la distanza di Hausdorff più piccola. Per eliminare anche geometrie non continue, in questi due cicli viene calcolato l'overlap tra le due maschere e viene esclusa la maschera che risulta avere un overlap pari a zero.

Per quanto riguarda la seconda criticità, verranno calcolati il gradiente orizzontale e verticale per ogni slice e viene sovrapposto il contorno della segmentazione corrispondente. Ogni punto del bordo si sposterà fin quando non incontra un punto del gradiente.

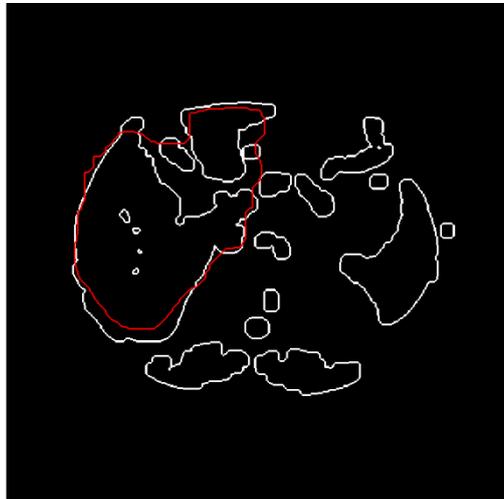


Figura 15. Sovrapposizione contorno di partenza con la maschera ottenuta dai gradienti.

Dal contorno ottenuto viene ricavata la maschera e la si migliora facendo una dilatazione, con un disco di diametro pari a 10 pixel.

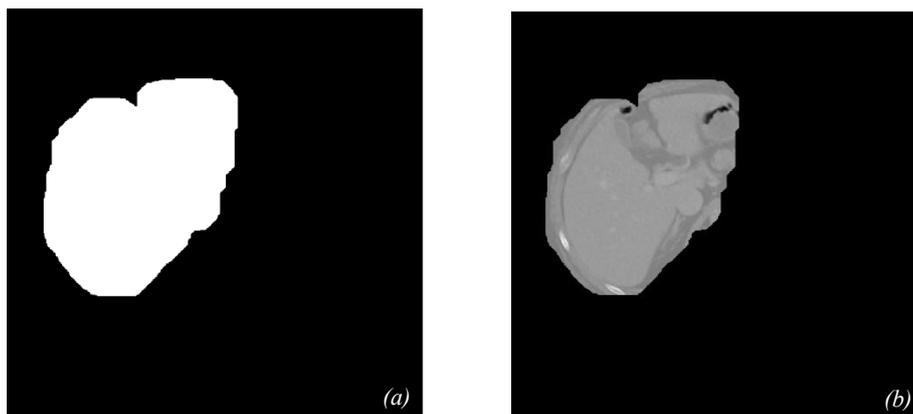


Figura 18. (a) Maschera finale e immagine mascherizzata finale (b).

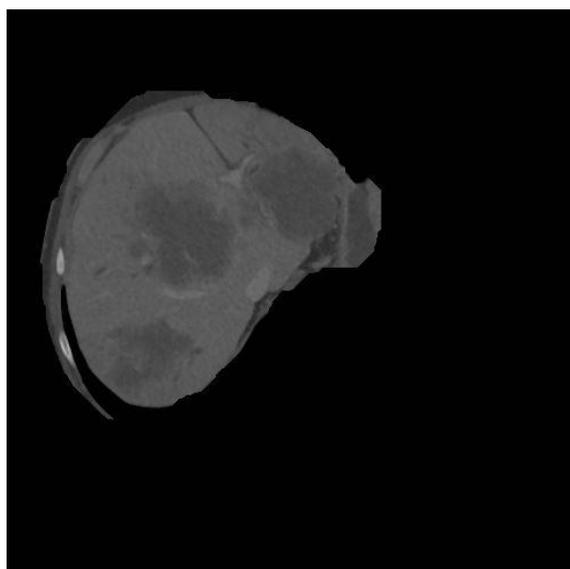


Figura 17. Esempio segmentazione fegato nel paziente 6.

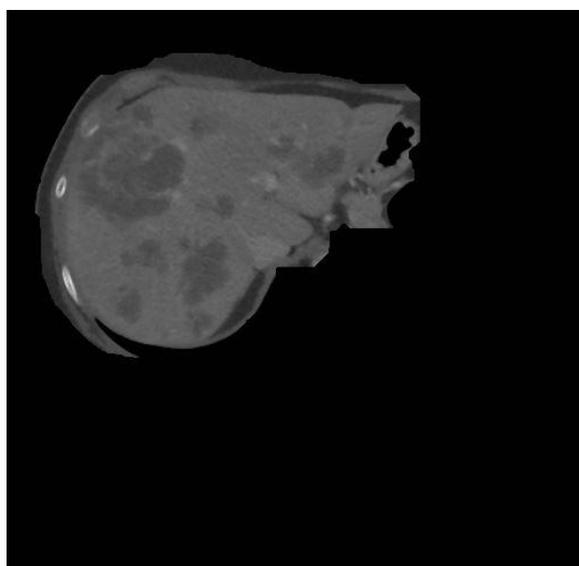


Figura 16. Esempio segmentazione fegato nel paziente 8.

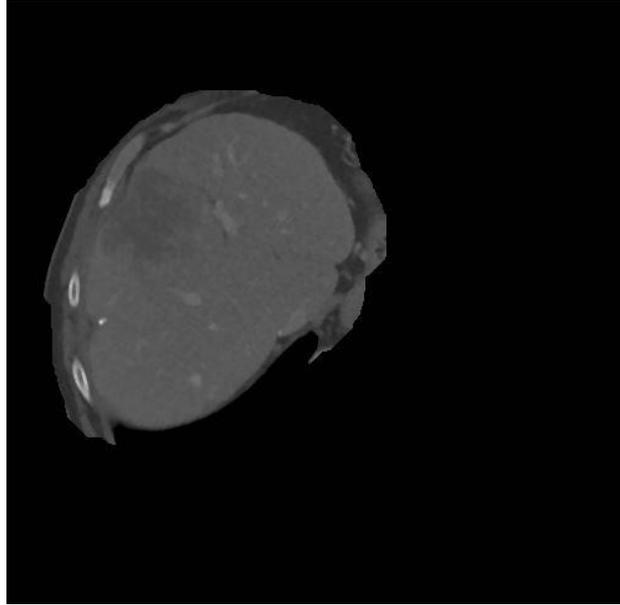


Figura 20. Esempio segmentazione fegato nel paziente 17.

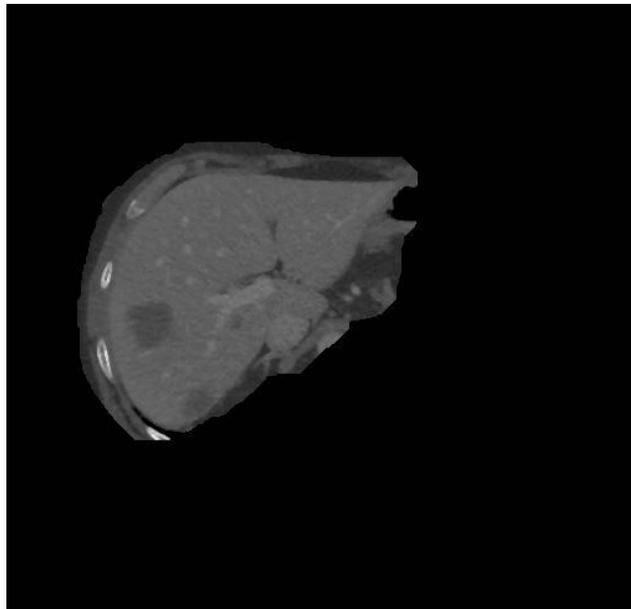


Figura 19. Esempio segmentazione fegato nel paziente 17.

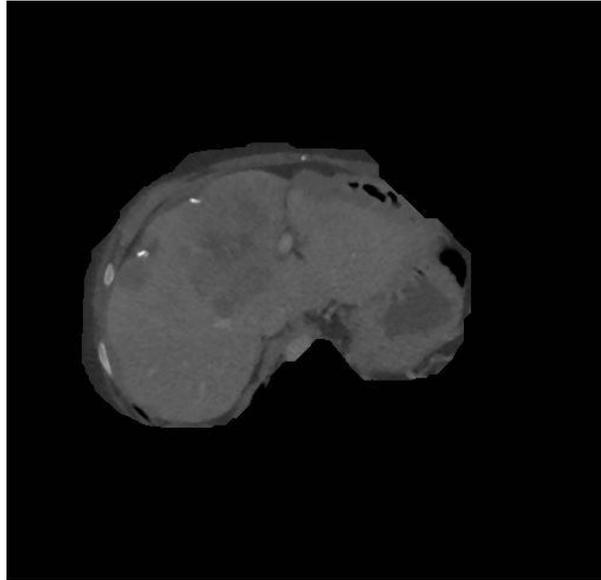


Figura 21. Esempio segmentazione fegato nel paziente 26.

Unet

Per la segmentazione delle metastasi è stata implementata una Unet. Quest'ultima è una rete neurale creata nel 2015 da Olaf Ronneberger, Philipp Fischer e Thomas Brox (14).

L'architettura di questa rete neurale si basa su una *fully convolutional network*, divisa in un due parti: down-sampling e up-sampling.

Dal layer di input l'immagine attraversa un neurone, il quale applica un kernel nell'immagine estrapolando n feature per ogni pixel, quindi costruendo n feature map. Successivamente ripassano in un altro neurone in cui applica la stessa convoluzione. In questo modo si hanno delle feature più stabili. Dopo questa doppia convoluzione, le feature map passano attraverso un layer chiamato max-pool: vengono confrontati 2x2 pixel contigui e viene considerato solo quello che presenta il valore più alto. In questa maniera le dimensioni delle feature map si ridurranno della metà. Dopodiché seguiranno a cascata i passaggi precedentemente descritti fino ad arrivare all'ultimo layer di downsampling. Nella fase di risalita viene effettuato un up-sampling delle feature map che verranno concatenate con quelle corrispondenti del layer di discesa. Questa concatenazione è fondamentale per non perdere informazioni di luogo delle feature. La risultante della concatenazione passa attraverso un neurone in cui viene applicata una convoluzione e l'uscita ripassa in un'altra convoluzione. Questo procedimento viene applicato fino al ritorno delle dimensioni originali.

Un esempio di architettura a 4 layer, in cui vengono estrapolate 64 feature è mostrato nella figura sottostante.

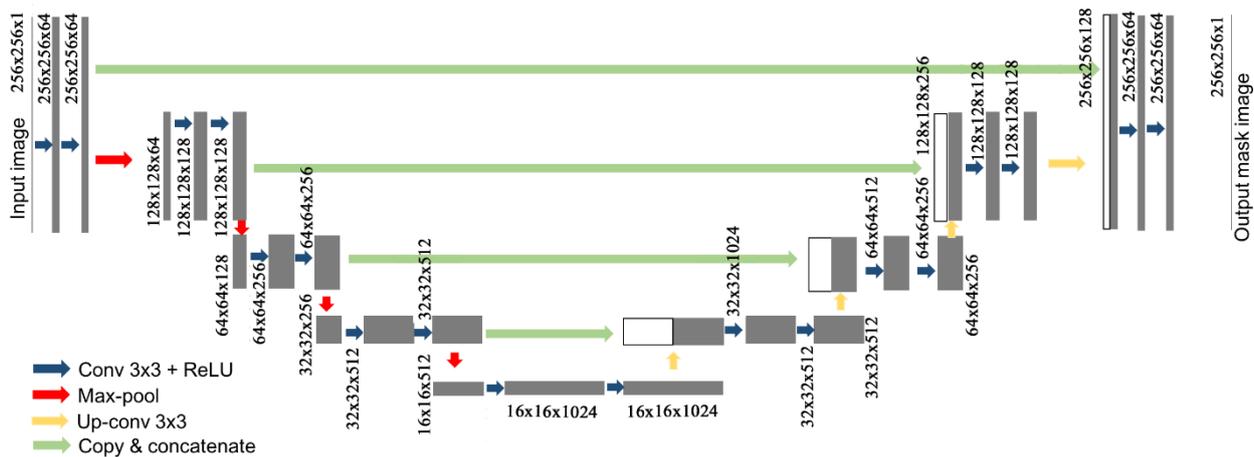


Figura 22. Esempio di struttura della Unet a 4 layer, con 64 feature estrapolate nel primo layer di convoluzione.

Sfruttando il backpropagation, i pesi dei neuroni vengono modificati iniziando un batch, che identifica il numero di immagini dal quale viene calcolato il gradiente di errore.

In sintesi, i parametri da modificare per migliorare le performance della rete sono:

- Numero di layer;
- Numero di feature da estrapolare con la prima convoluzione;
- Funzione di attivazione dei neuroni di convoluzione;
- Funzione di attivazione del neurone di output;
- Funzione di ottimizzazione, cioè come viene calcolato il gradiente dell'errore;
- Funzione di loss (calcola l'errore);
- Learning rate: di quanto deve essere modificato il peso ad ogni iterazione;
- Numero di epoche;

Nelle varie prove effettuate i parametri che sono stati cambiati sono il numero di layer, il numero di epoche, il learning rate e il numero di feature.

La funzione di attivazione dei neuroni di convoluzione, quella del neurone di output, la funzione di ottimizzazione e la funzione di loss non sono state modificate durante tutte le prove. Nello specifico sono:

- Funzione di attivazione dei neuroni di convoluzione: Rectified Linear Unit (ReLU);
- Funzione di attivazione del neurone di output: softmax, che è definita come $p_k(x) = \frac{e^{a_k(x)}}{\sum_{k=1}^K (e^{a_{kr}(x)})}$, dove $a_k(x)$ indica l'attivazione della feature k al pixel in posizione x ; K identifica il numero di classi;
- Funzione di ottimizzazione: Adam (*Adaptive moment estimation*), un metodo per l'ottimizzazione stocastica che richiede minima memoria. Il metodo calcola il learning rate basandosi su stime dei momenti di primo e secondo ordine dei gradienti.

- Funzione di loss: *binary cross entropy*, che è definita come $-(y \log(p) + (1-y) \log(1-p))$, dove y identifica la classe e p la probabilità che un determinato pixel appartenga alla classe y .

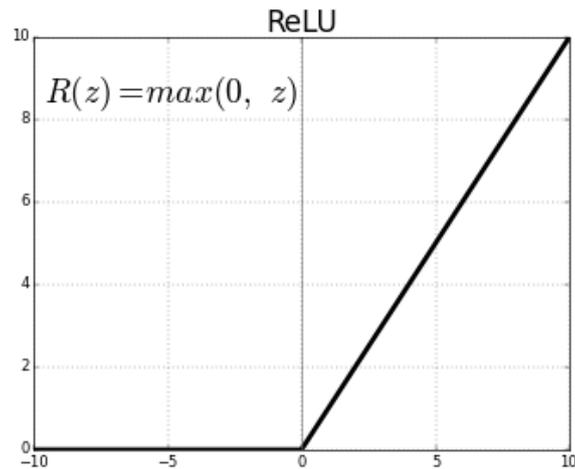


Figura 23. Grafico della funzione ReLU.

Le prime prove effettuate sono state fatte utilizzando le immagini normalizzate con la media e deviazione standard, utilizzando le stesse sia per training che per test.

Inizialmente è stato fatto variare il numero di layer da 3 a 5, lasciando invariato il learning rate a 10^{-4} , il numero di epoche a 10 e il numero di feature a 64. Maggiore è il numero di layer, più la rete si allena a trovare oggetti di grande dimensione, minore è il numero viceversa. Dai risultati ottenuti è stato scelto come numero ottimale di layer 4. Questo perché la dimensione delle lesioni è molto variabile e con 4 layer si riesce a trovare un giusto compromesso.

Il passo successivo è stato far variare prima il learning rate, a 10^{-3} , 10^{-4} , 10^{-5} , e il numero di feature. In questo caso non è stata rilevata una sostanziale differenza di performance, ma solo dei tempi computazionali che aumentavano con il valore del learning rate e con il numero di feature. Sono stati scelti valori pari a 10^{-4} per il learning rate e 64 per le feature.

Delle variazioni di performance rilevanti sono avvenute quando è stato aumentato il numero di epoche, fino a 60. In questo caso sono stati ottenuti i migliori risultati ed è stata validata la rete con tutte le caratteristiche ottenute fino ad ora. La validazione è avvenuta usando il 70% del dataset come training e il restante come test, dividendolo in maniera random. Le performance sono peggiorate più del previsto, pertanto è stato cambiato il dataset, usando quello con le immagini "riallineate" e normalizzate con il *min-max scaling*. È stata utilizzata la stessa struttura trovata precedentemente ed è avvenuto un fenomeno di *overfitting* a 60 epoche, che sono state diminuite a 40. Validando la rete come fatto precedentemente, sono stati ottenuti risultati migliori.

Per cercare di migliorare le performance è stata cambiata la strategia per la costruzione del training e del test set. Si costruiscono due matrici dalle quali vengono calcolati i dendrogrammi in modo tale da clusterizzare le immagini del dataset, per poter costruire un training set maggiormente bilanciato. Le due matrici vengono costruite nel seguente modo:

1. Vengono estrapolati il 10°, 25°, 50°, 75° e 90° percentile di ogni immagine del dataset;
2. Viene calcolato per ogni elemento il rapporto tra l'area delle metastasi e del corrispondente fegato;

3. Le aree delle metastasi;

Nel primo caso verrà effettuato un livello di taglio pari a 5, nel secondo pari a 3 e nell'ultimo pari a 5.

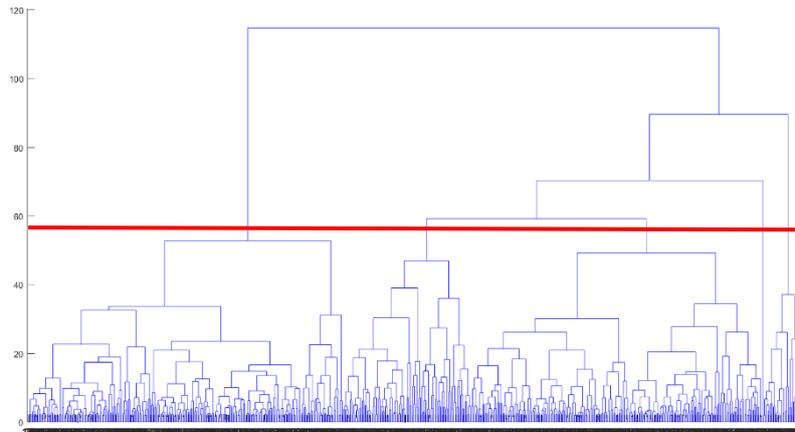


Figura 24. Dendrogramma ottenuto dai percentili. In rosso è segnata il livello di taglio.

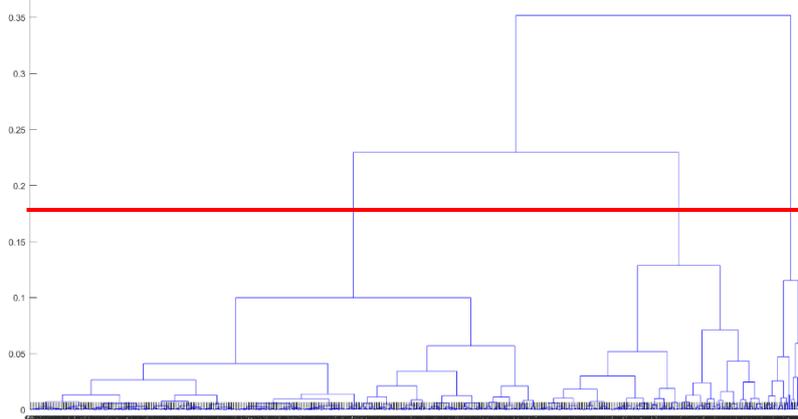


Figura 25. Dendrogramma ottenuto dal rapporto tra l'area della metastasi e l'area del fegato. In rosso è segnato il livello di taglio.

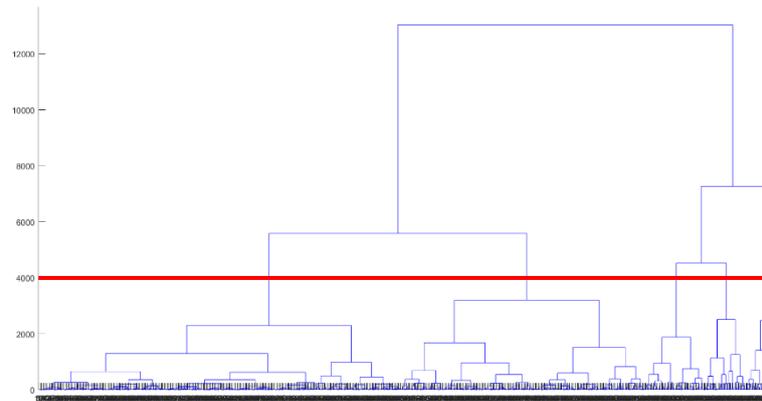


Figura 26. Dendrogramma ottenuto dalle aree della metastasi. In rosso è segnato il livello di taglio.

Così facendo sono stati costruiti tre training e tre test set e sono state rifatte le prove fatte precedentemente.

La divisione tra test e training set avviene in maniera random, quindi sono state effettuate 5 prove per ognuno dei metodi indicati precedentemente più quello che non utilizza i cluster. Sono stati confrontati gli andamenti delle performance per valutare quale dei metodi è il più robusto e offre le prestazioni più stabili.

MultiUnet

Per cercare di migliorare ancora di più le performance, è stato costruito un sistema con 4 Unet parallele, dove ognuna viene allenata dal 70% di ogni cluster ottenuto tramite il dendrogramma costruito con i valori dei percentili. Per la segmentazione finale si effettua l'operazione di *or* tra le maschere: se un pixel viene classificato come metastasi da almeno una rete allora verrà considerato come tale nel risultato finale. Inizialmente le varie reti sono testate usando il training set e successivamente validate usando le immagini escluse da ogni train. Infine, il sistema viene testando provandolo sull'intero dataset.

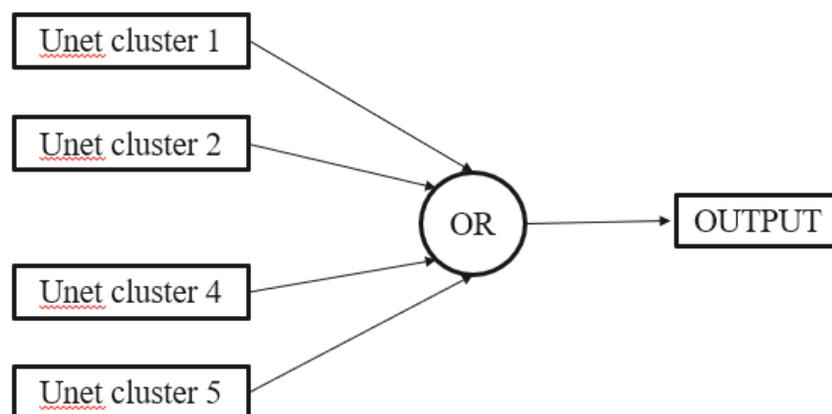


Figura 27. Schema di funzionamento della MultiUnet.

Poiché i cluster non sono bilanciati tra loro, quindi ogni rete aveva un numero diverso di elementi con cui allenarsi, le reti sono state allenate con caratteristiche diverse. Nello specifico si hanno:

- Cluster 1= 85 elementi;
- Cluster 2= 195 elementi;
- Cluster 3= 1 elemento;
- Cluster 4= 23 elementi;
- Cluster 5= 212 elementi.

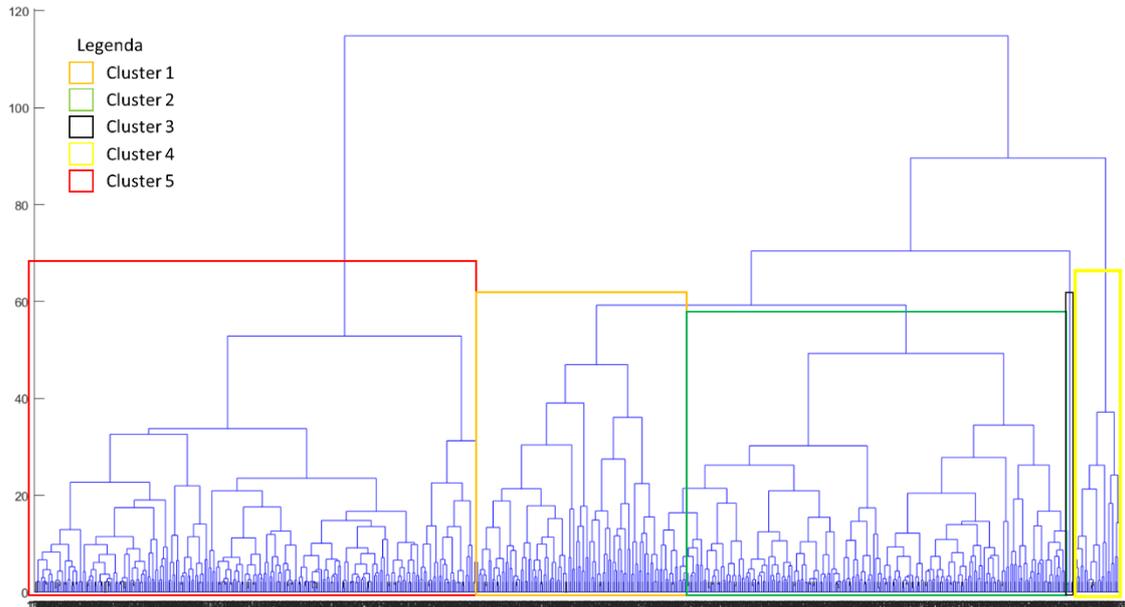


Figura 28. Raggruppamenti dei vari cluster.

Il cluster 3 viene escluso dall'algoritmo visto che ha un solo elemento al proprio interno.

L'architettura è rimasta invariata in tutti e quattro casi, ma il numero di epoche è differente. Infatti, vengono provate variando il numero di epoche tra 5 e 150, le prime due prove a passo di 5, le restanti a passo di 10. In ogni prova viene confrontata la performance della rete con training set uguale a test set con quella che ha il test set pari al 30% degli elementi del relativo cluster. Questo confronto viene effettuato per essere sicuri di non andare in contro al fenomeno dell'*overfitting*.

Una volta trovata la combinazione che offre la migliore prestazione ad ogni rete, il sistema viene validato utilizzando tutte le immagini del dataset e il risultato finale viene costruito applicando l'operazione di *or* tra le maschere ottenute.

Risultati

Per valutare le prestazioni di ogni rete sono stati calcolati i seguenti parametri per ogni immagine del test:

- Coefficiente Sørensen–Dice: $\frac{2|A \cap B|}{|A| + |B|}$;
- Precision: $\frac{|A \cap B|}{|B|}$;
- Recall: $\frac{|A \cap B|}{|A|}$.

A indica la maschera manuale, B la maschera ottenuta dall’algoritmo. Il dice fornisce informazioni su quanto le due maschere siano sovrapposte, la precision indica quanti pixel segmentati dall’algoritmo siano rilevanti (rappresentano metastasi), mentre la recall indica quanti pixel rilevanti ha segmentato l’algoritmo.

Per ognuno dei precedenti parametri vengono calcolati:

- Media;
- Deviazione standard (indicata con *std*);
- Mediana;
- 25° percentile;
- 75° percentile.

In generale, è stato considerato come discriminante il dice. Nello specifico, nelle prove in cui il train era uguale al test vengono confrontati le medie dei dice. Nelle validazioni vengono anche calcolati il numero di falsi negativi (indicato successivamente come FN) e il confronto, invece, avviene tra i percentili, dal momento che i falsi negativi portano ad una diminuzione della media.

Unet

Numero di layer

Le prime prove effettuate sono state quelle per trovare il numero ottimale di layer. Quest’ultimi sono stati fatti variare da 3 a 5, lasciando invariato gli altri parametri e impostando il numero di epoche pari a 10.

Tabella 5. Valori, per ogni layer, di media, deviazione standard, mediana, 25° e 75° percentile per dice, precision e recall.

	DICE			PRECISION			
	3 layer	4 layer	5 layer	3 layer	4 layer	5 layer	
<i>media</i>	0.38	0.50	0.40	<i>media</i>	0.45	0.73	0.51
<i>std</i>	0.26	0.30	0.27	<i>std</i>	0.29	0.28	0.29
<i>mediana</i>	0.42	0.57	0.40	<i>mediana</i>	0.49	0.85	0.56
<i>25pct</i>	0.14	0.22	0.13	<i>25pct</i>	0.16	0.65	0.29
<i>75pct</i>	0.61	0.79	0.67	<i>75pct</i>	0.71	0.91	0.78

RECALL			
	3 layer	4 layer	5 layer
<i>media</i>	0.38	0.42	0.36
<i>std</i>	0.27	0.29	0.27
<i>mediana</i>	0.42	0.44	0.35
<i>25pcrt</i>	0.11	0.14	0.09
<i>75pcrt</i>	0.61	0.71	0.61

Dalle tabelle si evince che i valori di dice, precision e recall più alti si hanno a 4 layer, che verrà impostato come numero ottimale per le successive prove.

Learning rate

Il passo successivo è valutare il valore ottimale di learning rate, facendolo variare da 10^{-3} a 10^{-5} . Anche in questo caso l'architettura è uguale in tutte e tre le reti, vengono utilizzati i 4 layer trovati precedentemente e sempre 10 epoche.

Tabella 6. Valori, per ogni valore di learning rate, di media, deviazione standard, mediana, 25° e 75° percentile per dice, precision e recall.

DICE				PRECISION			
	10^{-3}	10^{-4}	10^{-5}		10^{-3}	10^{-4}	10^{-5}
<i>media</i>	0.39	0.46	0.45	<i>media</i>	0.40	0.62	0.39
<i>std</i>	0.30	0.32	0.25	<i>std</i>	0.32	0.34	0.25
<i>mediana</i>	0.47	0.54	0.49	<i>mediana</i>	0.47	0.77	0.40
<i>25pcrt</i>	0.00	0.09	0.22	<i>25pcrt</i>	0.01	0.36	0.16
<i>75pcrt</i>	0.68	0.77	0.69	<i>75pcrt</i>	0.69	0.89	0.62

RECALL			
	10^{-3}	10^{-4}	10^{-5}
<i>media</i>	0.38	0.42	0.36
<i>std</i>	0.27	0.29	0.27
<i>mediana</i>	0.42	0.44	0.35
<i>25pcrt</i>	0.11	0.14	0.09
<i>75pcrt</i>	0.61	0.71	0.61

In questo caso il valore ottimale risulta essere 10^{-4} , anche se il proprio valor medio del dice è poco più grande di quello con learning rate pari a 10^{-5} , si hanno dei valori di precision e recall più grandi.

Numero di features

Il numero di features da estrapolare nel primo layer di convoluzione è stato confronto tra il valore 64 (di default) e 100. Come nei casi precedenti, il confronto avviene con la stessa architettura e i parametri ottimali trovati fino a questo momento.

Tabella 7. Valori, per ogni numero di feature da estrapolare nel primo layer di convoluzione, di media, deviazione standard, mediana, 25° e 75° percentile per dice, precision e recall.

	DICE		PRECISION		RECALL			
	64	100	64	100	64	100		
<i>media</i>	0.50	0.41	<i>media</i>	0.73	0.47	<i>media</i>	0.42	0.41
<i>std</i>	0.30	0.30	<i>std</i>	0.28	0.32	<i>std</i>	0.29	0.31
<i>mediana</i>	0.57	0.45	<i>mediana</i>	0.85	0.52	<i>mediana</i>	0.44	0.44
<i>25pct</i>	0.22	0.04	<i>25pct</i>	0.65	0.15	<i>25pct</i>	0.14	0.05
<i>75pct</i>	0.79	0.72	<i>75pct</i>	0.91	0.77	<i>75pct</i>	0.71	0.70

L'aumento del numero di features non porta ad un aumento delle performance. Come valore ottimale viene scelto 64.

Numero di epoche

Trovato il numero ottimale di layer, learning rate e numero di feature da estrapolare nel primo layer di convoluzione, si ricerca il numero di epoche. In questo caso la ricerca viene fatta in maniera più approfondita, facendole variare da 5 a 60 epoche.

Tabella 8. Valori, per ogni numero di epoche, di media, deviazione standard, mediana, 25° e 75° percentile per dice, precision e recall.

	DICE				
	5	10	20	40	60
<i>media</i>	0.35	0.53	0.79	0.89	0.92
<i>std</i>	0.27	0.28	0.20	0.12	0.07
<i>mediana</i>	0.34	0.59	0.86	0.92	0.94
<i>25pct</i>	0.05	0.34	0.77	0.88	0.91
<i>75pct</i>	0.62	0.77	0.90	0.94	0.96

	PRECISION				
	5	10	20	40	60
<i>media</i>	0.39	0.59	0.86	0.90	0.95
<i>std</i>	0.28	0.29	0.17	0.12	0.06
<i>mediana</i>	0.39	0.69	0.92	0.93	0.96
<i>25pct</i>	0.11	0.40	0.83	0.89	0.94
<i>75pct</i>	0.63	0.82	0.96	0.96	0.98

	RECALL				
	5	10	20	40	60
<i>media</i>	0.39	0.51	0.75	0.89	0.91
<i>std</i>	0.30	0.29	0.22	0.13	0.10
<i>mediana</i>	0.39	0.54	0.84	0.93	0.94
<i>25pct</i>	0.09	0.32	0.71	0.88	0.89
<i>75pct</i>	0.66	0.78	0.89	0.95	0.96

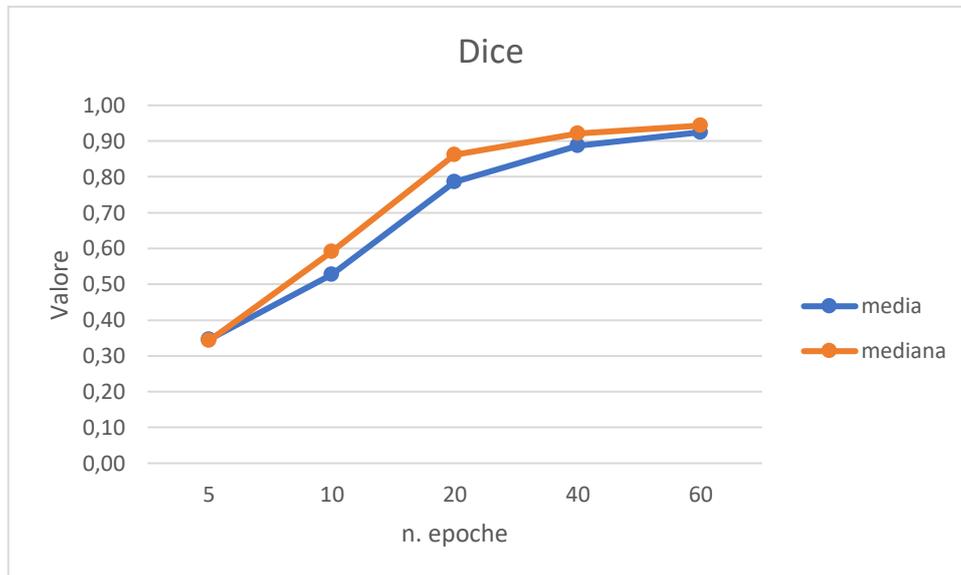


Figura 30. Andamento di media e mediana del dice al variare del numero di epoche.

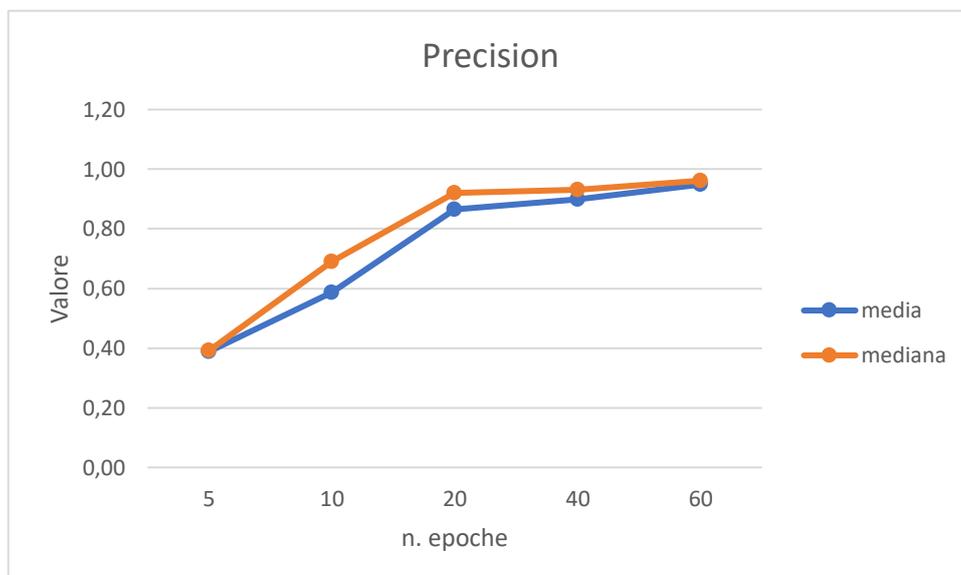


Figura 29. Andamento di media e mediana della precision al variare del numero di epoche.

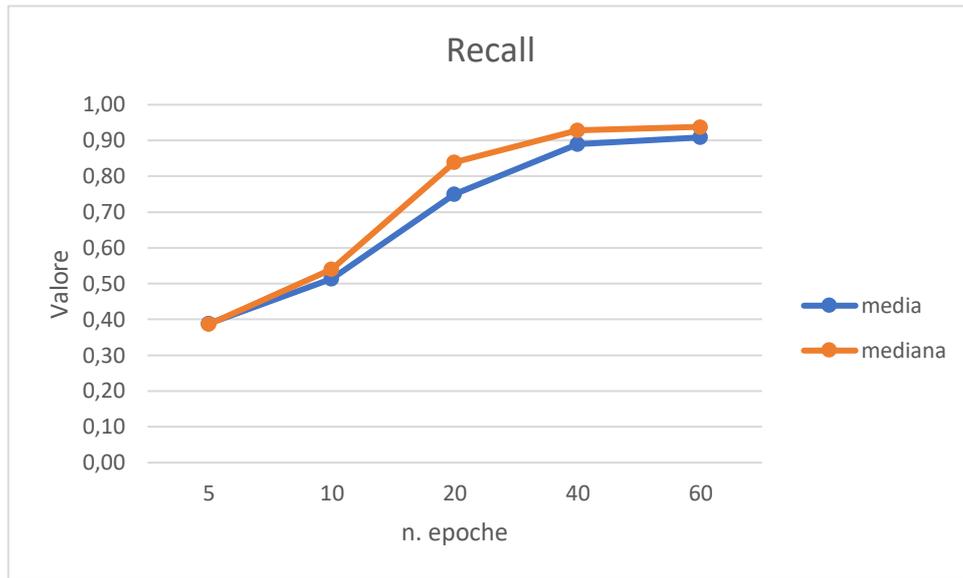


Figura 31. Andamento di media e mediana della recall al variare del numero di epoche.

Il numero di epoche influisce maggiormente sulle prestazioni del sistema. Andando ad osservare l'andamento dei parametri si nota un netto aumento con il variare delle epoche, fino a raggiungere un massimo a 60 epoche, dove si ha un dice medio del 92%.

Dalle prove effettuate si evince che la struttura migliore per la rete neurale è:

- Numero di layer: 4;
- Learning rate: 10^{-4} ;
- Numero di feature: 64;
- Numero di epoche: 60.

Con questa struttura si procederà alla validazione.

Validazione e miglioramenti

La validazione viene effettuata usando il 70% del dataset come training set e il restante 30% come test set, dividendoli in maniera random.

Tabella 9. Valori ottenuti dalla validazione di media, deviazione standard, mediana, 25° e 75° percentili dei valori di dice, precision e recall. In basso a destra sono mostrati il numero di falsi negativi totali.

	Dice	Precision	Recall	
Media	0.67	0.76	0.64	
Dev. Standard	0.28	0.29	0.29	
Mediana	0.78	0.88	0.73	
25° percentile	0.61	0.73	0.54	
75° percentile	0.87	0.93	0.87	FN=15

Validando la rete subisce una diminuzione delle prestazioni superiori delle aspettative, per migliorare ciò in primo luogo è stato modificando il dataset, andando a modificare l'intensità dei pixel dei vari pazienti in modo tale che il fegato venga identificato dallo stesso valore.

In questo caso le prove delle performance avviene solo facendo variare il numero di epoche.

Tabella 10. Valori ottenuti dopo aver cambiato il dataset. Sono riportati la media, deviazione standard, mediana, 25° e 75° percentile di dice, precision e recall al variare del numero di epoche.

DICE						
	5	10	20	40	50	60
<i>media</i>	0.42	0.63	0.78	0.90	0.84	0.48
<i>std</i>	0.26	0.28	0.22	0.08	0.18	0.31
<i>mediana</i>	0.49	0.74	0.87	0.93	0.90	0.55
<i>25pct</i>	0.15	0.49	0.75	0.88	0.84	0.21
<i>75pct</i>	0.66	0.84	0.91	0.94	0.93	0.75

PRECISION						
	5	10	20	40	50	60
<i>media</i>	0.43	0.70	0.78	0.89	0.89	0.77
<i>std</i>	0.28	0.27	0.22	0.08	0.15	0.35
<i>mediana</i>	0.48	0.80	0.87	0.92	0.92	0.95
<i>25pct</i>	0.13	0.58	0.75	0.87	0.87	0.77
<i>75pct</i>	0.69	0.89	0.91	0.94	0.96	0.99

RECALL						
	5	10	20	40	50	60
<i>media</i>	0.45	0.62	0.78	0.92	0.83	0.39
<i>std</i>	0.26	0.30	0.22	0.10	0.21	0.29
<i>mediana</i>	0.48	0.72	0.87	0.94	0.90	0.40
<i>25pct</i>	0.23	0.44	0.75	0.91	0.80	0.12
<i>75pct</i>	0.68	0.85	0.91	0.97	0.95	0.65

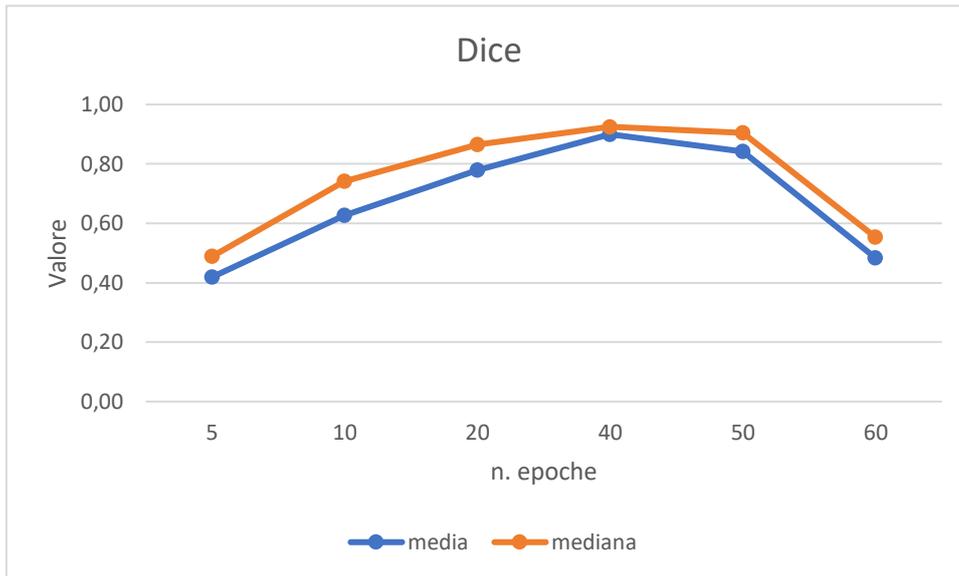


Figura 34. Andamento di media e mediana del dice al variare del numero di epoche.

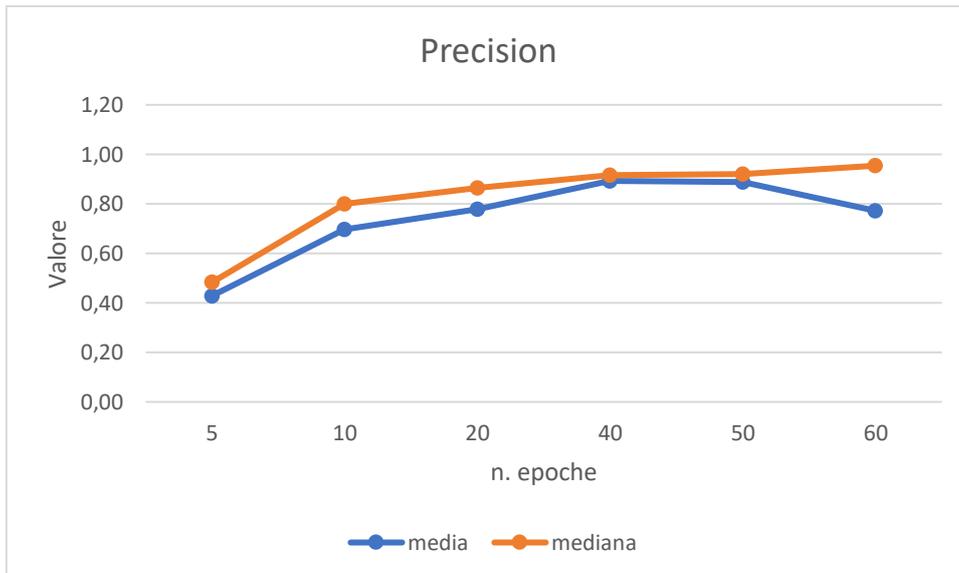


Figura 32. Andamento di media e mediana della precision al variare del numero di epoche.

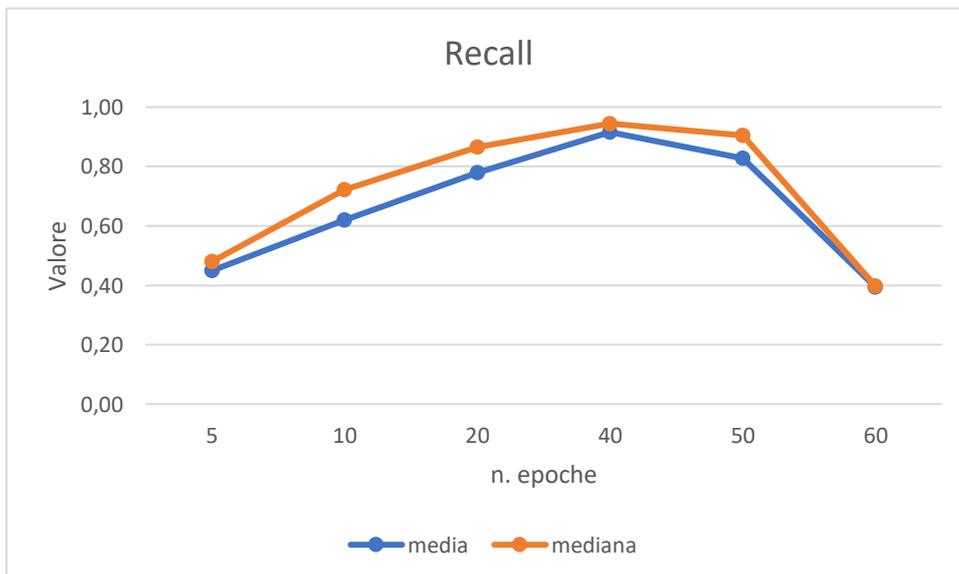


Figura 33. Andamento di media e mediana della recall al variare del numero di epoche.

Con il nuovo dataset dalle 50 epoche in poi si ha *overfitting*, questo perché avendo immagini più simili tra loro la rete si allenerà prima.

Viene validata anche questa rete, con il numero di epoche pari a 40.

Tabella 11. Valori ottenuti dalla validazione di media, deviazione standard, mediana, 25° e 75° percentili dei valori di dice, precision e recall. In basso a destra sono mostrati il numero di falsi negativi totali.

	Dice	Precision	Recall	
Media	0.75	0.83	0.71	
Dev. Standard	0.23	0.23	0.25	
Mediana	0.84	0.91	0.80	
25° percentile	0.70	0.83	0.63	
75° percentile	0.90	0.95	0.90	FN=6

Con questo tipo di dataset si ha un aumento delle prestazioni. Per cercare di migliorarle ulteriormente vengono quindi costruiti il training set e il test clusterizzando il data set come descritto nel capitolo precedente. Vengono messe in confronto le varie validazioni, le due ottenute fino ad ora con quelle ottenute con i test e train set costruiti dai cluster.

Tabella 12. Confronto delle validazioni tra i vari metodi di costruzione di train e test set utilizzati.

IMMAGINI STANDARDIZZATE, FN=15				IMMAGINI NORMALIZZATE, FN=6			
	Dice	Precision	Recall		Dice	Precision	Recall
media	0.67	0.76	0.64	media	0.75	0.83	0.71
std	0.28	0.29	0.29	std	0.23	0.23	0.25
mediana	0.78	0.88	0.73	mediana	0.84	0.91	0.80
25pct	0.61	0.73	0.54	25pct	0.70	0.83	0.63
75pct	0.87	0.93	0.87	75pct	0.90	0.95	0.90
CLUSTER PERCENTILI, FN=6				CLUSTER RAPPORTO AREE, FN=10			
	Dice	Precision	Recall		Dice	Precision	Recall
media	0.75	0.78	0.76	media	0.73	0.76	0.73
std	0.24	0.23	0.26	std	0.25	0.25	0.28
mediana	0.84	0.86	0.87	mediana	0.82	0.85	0.85
25pct	0.71	0.75	0.69	25pct	0.67	0.72	0.58
75pct	0.90	0.91	0.94	75pct	0.89	0.91	0.93
CLUSTER AREE, FN=5							
	Dice	Precision	Recall				
media	0.75	0.77	0.76				
std	0.22	0.24	0.24				
mediana	0.84	0.86	0.85				
25pct	0.71	0.74	0.68				
75pct	0.90	0.91	0.93				

Dai risultati si evince che dividere il data set prendendoli in maniera random o clusterizzando in base ai percentili non cambia più di tanto le performance.

Per vedere quale dei metodi è il più robusto e crea un data set più bilanciato, sono state effettuate 5 prove, creando 5 dataset diversi per ogni metodo, sempre usando la divisione in 70% train e 30% test prendendoli in maniera random.

Tabella 13. Valori ottenuti dalla validazione, senza la costruzione dei cluster, di media, deviazione standard, mediana, 25° e 75° percentili dei valori di dice, precision e recall per ogni prova.

NO CLUSTER					
DICE					
	<i>Prova 1</i>	<i>Prova 2</i>	<i>Prova 3</i>	<i>Prova 4</i>	<i>Prova 5</i>
<i>media</i>	0.75	0.73	0.77	0.75	0.75
<i>std</i>	0.23	0.26	0.23	0.26	0.24
<i>mediana</i>	0.84	0.85	0.85	0.86	0.85
<i>25pcrt</i>	0.70	0.64	0.74	0.74	0.71
<i>75pcrt</i>	0.90	0.89	0.91	0.91	0.90
PRECISION					
	<i>Prova 1</i>	<i>Prova 2</i>	<i>Prova 3</i>	<i>Prova 4</i>	<i>Prova 5</i>
<i>media</i>	0.83	0.79	0.81	0.78	0.83
<i>std</i>	0.23	0.26	0.22	0.26	0.22
<i>mediana</i>	0.91	0.89	0.88	0.87	0.91
<i>25pcrt</i>	0.83	0.77	0.79	0.77	0.80
<i>75pcrt</i>	0.95	0.95	0.93	0.93	0.95
RECALL					
	<i>Prova 1</i>	<i>Prova 2</i>	<i>Prova 3</i>	<i>Prova 4</i>	<i>Prova 5</i>
<i>media</i>	0.71	0.71	0.76	0.76	0.73
<i>std</i>	0.25	0.28	0.25	0.28	0.25
<i>mediana</i>	0.80	0.83	0.85	0.87	0.81
<i>25pcrt</i>	0.63	0.64	0.70	0.71	0.66
<i>75pcrt</i>	0.90	0.90	0.92	0.94	0.91

Tabella 14. Valori ottenuti dalla validazione, con la costruzione dei cluster ottenuti tramite i percentili, di media, deviazione standard, mediana, 25° e 75° percentili dei valori di dice, precision e recall per ogni prova.

PERCENTILI					
DICE					
	<i>Prova 1</i>	<i>Prova 2</i>	<i>Prova 3</i>	<i>Prova 4</i>	<i>Prova 5</i>
<i>media</i>	0.75	0.71	0.71	0.73	0.75
<i>std</i>	0.23	0.27	0.27	0.24	0.23
<i>mediana</i>	0.85	0.83	0.85	0.82	0.84
<i>25pcrt</i>	0.70	0.64	0.64	0.68	0.73
<i>75pcrt</i>	0.89	0.89	0.89	0.90	0.90
PRECISION					
	<i>Prova 1</i>	<i>Prova 2</i>	<i>Prova 3</i>	<i>Prova 4</i>	<i>Prova 5</i>
<i>media</i>	0.81	0.85	0.74	0.80	0.81
<i>std</i>	0.22	0.27	0.29	0.22	0.24
<i>mediana</i>	0.87	0.96	0.86	0.88	0.90
<i>25pcrt</i>	0.79	0.90	0.72	0.78	0.79
<i>75pcrt</i>	0.93	0.98	0.92	0.92	0.95
RECALL					
	<i>Prova 1</i>	<i>Prova 2</i>	<i>Prova 3</i>	<i>Prova 4</i>	<i>Prova 5</i>
<i>media</i>	0.74	0.64	0.74	0.72	0.74
<i>std</i>	0.25	0.27	0.29	0.28	0.25
<i>mediana</i>	0.84	0.74	0.86	0.84	0.83
<i>25pcrt</i>	0.65	0.50	0.69	0.57	0.68
<i>75pcrt</i>	0.92	0.84	0.92	0.93	0.91

Tabella 15. Valori ottenuti dalla validazione, con la costruzione dei cluster ottenuti tramite il rapporto tra le aree delle metastasi con quelle del fegato, di media, deviazione standard, mediana, 25° e 75° percentili dei valori di dice, precision e recall per ogni prova.

RAPPORTO AREA METASTASI SU AREA FEGATO					
DICE					
	<i>Prova 1</i>	<i>Prova 2</i>	<i>Prova 3</i>	<i>Prova 4</i>	<i>Prova 5</i>
<i>media</i>	0.76	0.74	0.76	0.64	0.76
<i>std</i>	0.23	0.23	0.24	0.27	0.24
<i>mediana</i>	0.84	0.82	0.84	0.76	0.85
<i>25pcrt</i>	0.70	0.69	0.73	0.54	0.73
<i>75pcrt</i>	0.91	0.89	0.89	0.84	0.90
PRECISION					
	<i>Prova 1</i>	<i>Prova 2</i>	<i>Prova 3</i>	<i>Prova 4</i>	<i>Prova 5</i>
<i>media</i>	0.77	0.75	0.82	0.70	0.83
<i>std</i>	0.23	0.23	0.21	0.27	0.23
<i>mediana</i>	0.85	0.84	0.88	0.78	0.91
<i>25pcrt</i>	0.73	0.68	0.80	0.64	0.83
<i>75pcrt</i>	0.92	0.90	0.92	0.89	0.95
RECALL					
	<i>Prova 1</i>	<i>Prova 2</i>	<i>Prova 3</i>	<i>Prova 4</i>	<i>Prova 5</i>
<i>media</i>	0.78	0.77	0.75	0.67	0.74
<i>std</i>	0.25	0.25	0.26	0.30	0.25
<i>mediana</i>	0.87	0.87	0.86	0.79	0.83
<i>25pcrt</i>	0.69	0.72	0.69	0.49	0.65
<i>75pcrt</i>	0.95	0.93	0.93	0.91	0.90

Tabella 16. Valori ottenuti dalla validazione, con la costruzione dei cluster ottenuti tramite le aree delle metastasi, di media, deviazione standard, mediana, 25° e 75° percentili dei valori di dice, precision e recall per ogni prova.

AREA METASTASI					
DICE					
	<i>Prova 1</i>	<i>Prova 2</i>	<i>Prova 3</i>	<i>Prova 4</i>	<i>Prova 5</i>
<i>media</i>	0.75	0.75	0.72	0.74	0.78
<i>std</i>	0.22	0.23	0.26	0.29	0.24
<i>mediana</i>	0.84	0.83	0.82	0.87	0.87
<i>25pcrt</i>	0.71	0.70	0.67	0.72	0.76
<i>75pcrt</i>	0.90	0.89	0.89	0.91	0.92
PRECISION					
	<i>Prova 1</i>	<i>Prova 2</i>	<i>Prova 3</i>	<i>Prova 4</i>	<i>Prova 5</i>
<i>media</i>	0.77	0.80	0.82	0.77	0.82
<i>std</i>	0.24	0.22	0.24	0.29	0.21
<i>mediana</i>	0.86	0.89	0.90	0.88	0.89
<i>25pcrt</i>	0.74	0.77	0.80	0.79	0.82
<i>75pcrt</i>	0.91	0.94	0.95	0.92	0.93
RECALL					
	<i>Prova 1</i>	<i>Prova 2</i>	<i>Prova 3</i>	<i>Prova 4</i>	<i>Prova 5</i>
<i>media</i>	0.76	0.73	0.68	0.74	0.77
<i>std</i>	0.24	0.25	0.27	0.31	0.26
<i>mediana</i>	0.85	0.81	0.78	0.86	0.88
<i>25pcrt</i>	0.68	0.65	0.62	0.67	0.71
<i>75pcrt</i>	0.93	0.90	0.88	0.93	0.93

Tabella 17. Valori del numero di falsi negativi ottenuti da ogni prova in ognuno dei metodi di costruzione del data set.

CONFRONTO NUMERO DI FALSI NEGATIVI					
	<i>Prova 1</i>	<i>Prova 2</i>	<i>Prova 3</i>	<i>Prova 4</i>	<i>Prova 5</i>
<i>No Cluster</i>	6	7	6	11	4
<i>Percentili</i>	6	13	11	6	9
<i>Area</i>	5	7	8	16	5
<i>Metastasi/fegato</i>	8	5	7	6	7

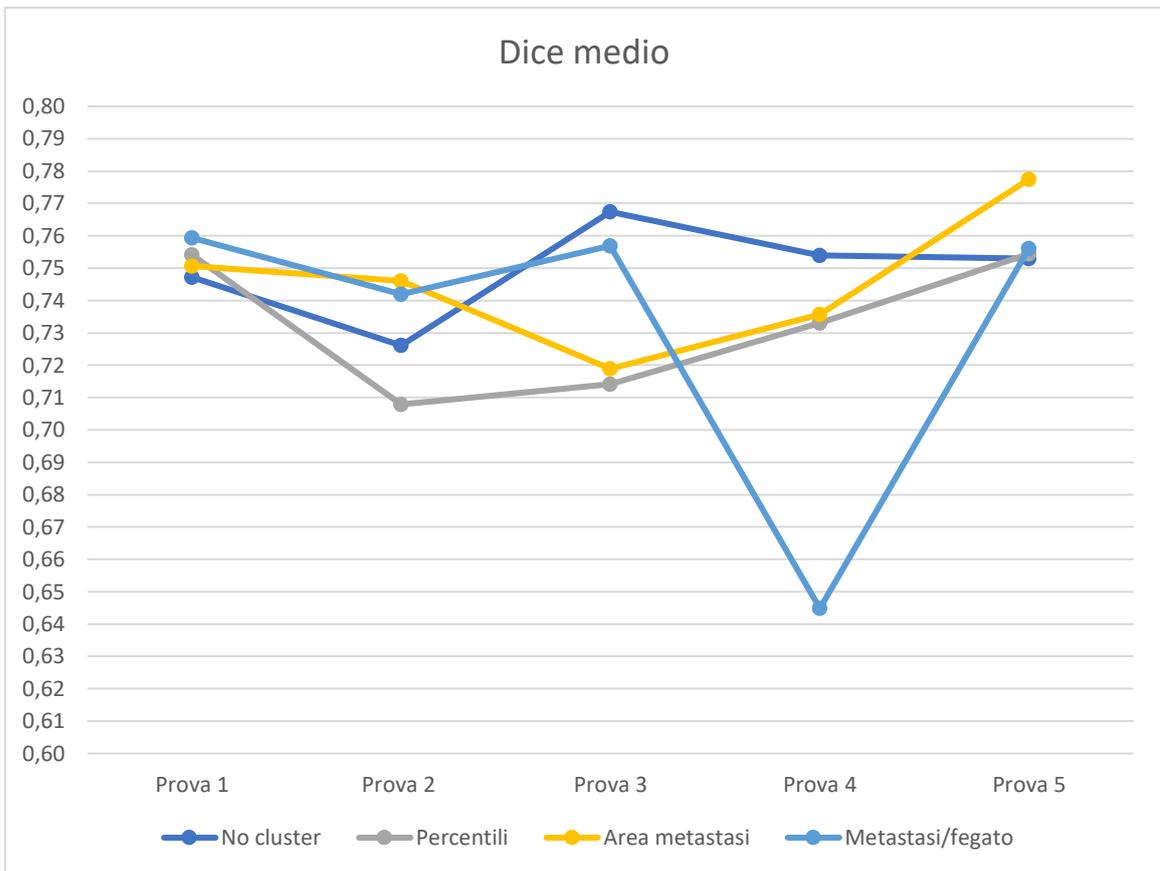


Figura 35. Andamento del dice medio al variare dell prove effettuate per ognuno dei metodi di costruzione di train e validation set.

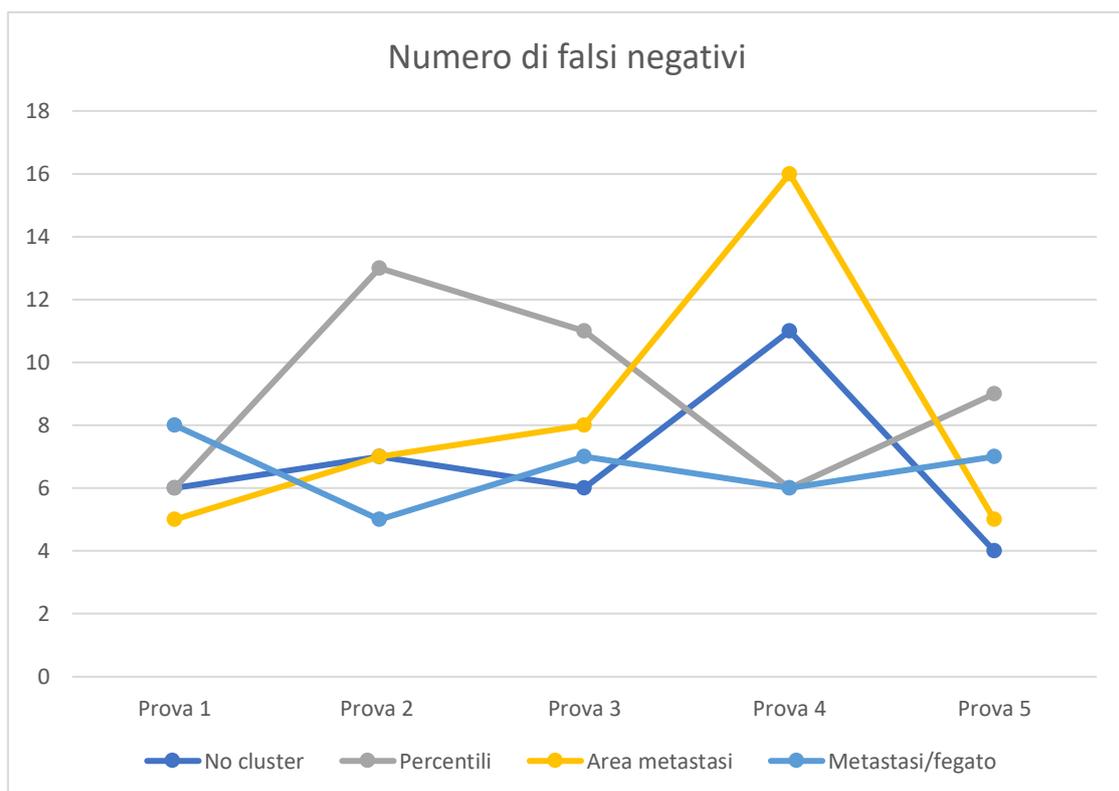


Figura 36. Andamento del numero di falsi negativi al variare dell prove effettuate per ognuno dei metodi di costruzione di train e validation set.

Dalle prove effettuate si evince che i metodi che offrono una maggiore robustezza nella costruzione del data sono tutti tranne quello che utilizza il rapporto tra le aree delle metastasi con quelle del fegato.

In particolare, le prove che offrono le migliori performance sono la prova 3 per il metodo che non utilizza i cluster e la prova 5 sia per i cluster ottenuti dai percentili che per quelli ottenuti dalle aree delle metastasi.

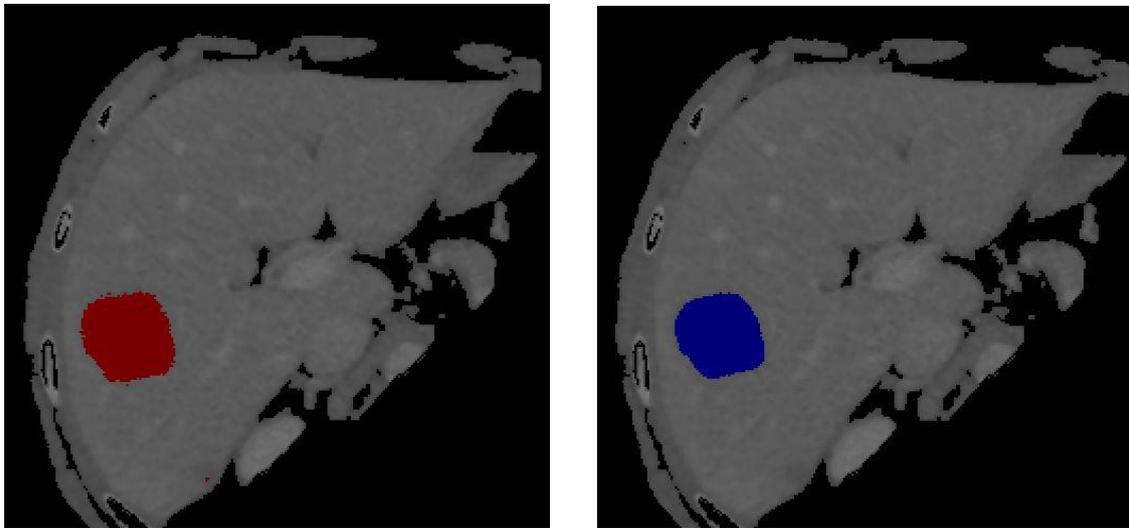


Figura 37. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set tramite i cluster ottenuti dalle aree delle metastasi. In blu sono riportate le maschere manuali.

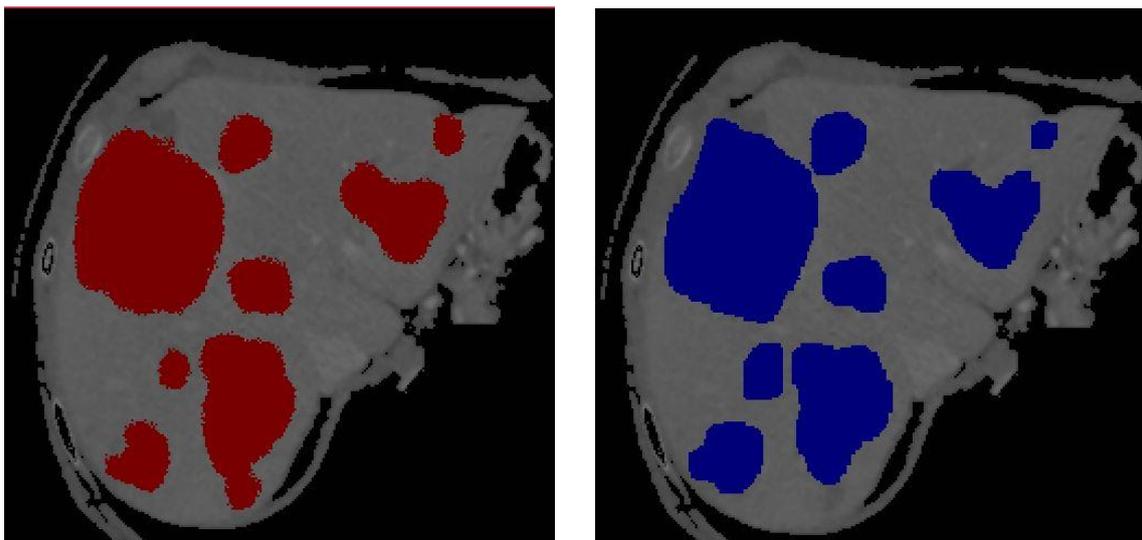


Figura 38. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set tramite i cluster ottenuti dalle aree delle metastasi. In blu sono riportate le maschere manuali

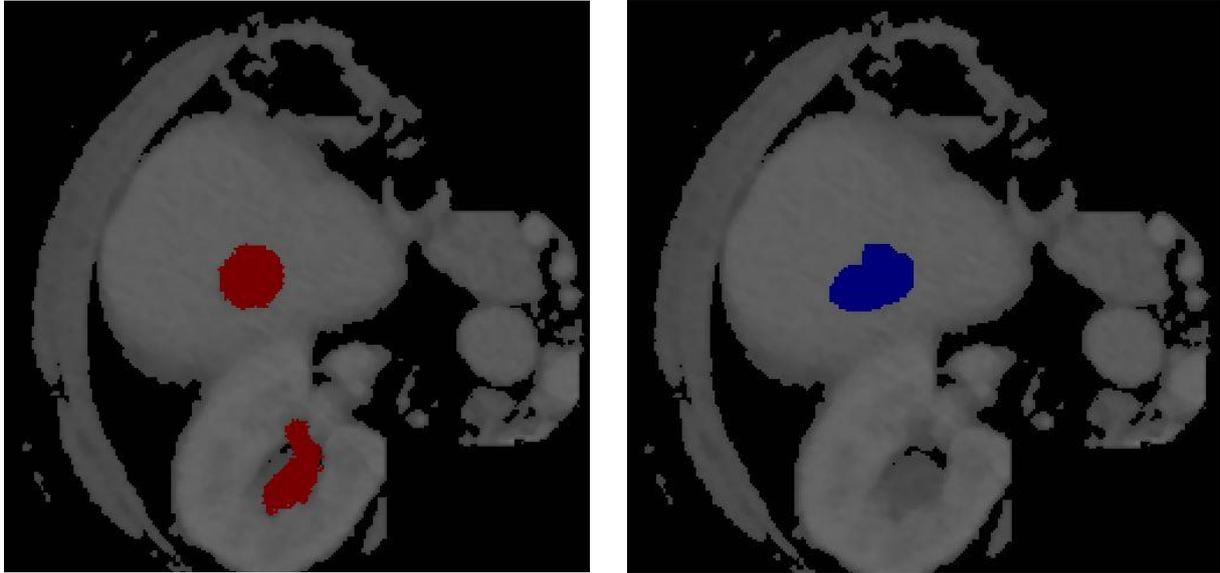


Figura 39. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set tramite i cluster ottenuti dalle aree delle metastasi. In blu sono riportate le maschere manuali

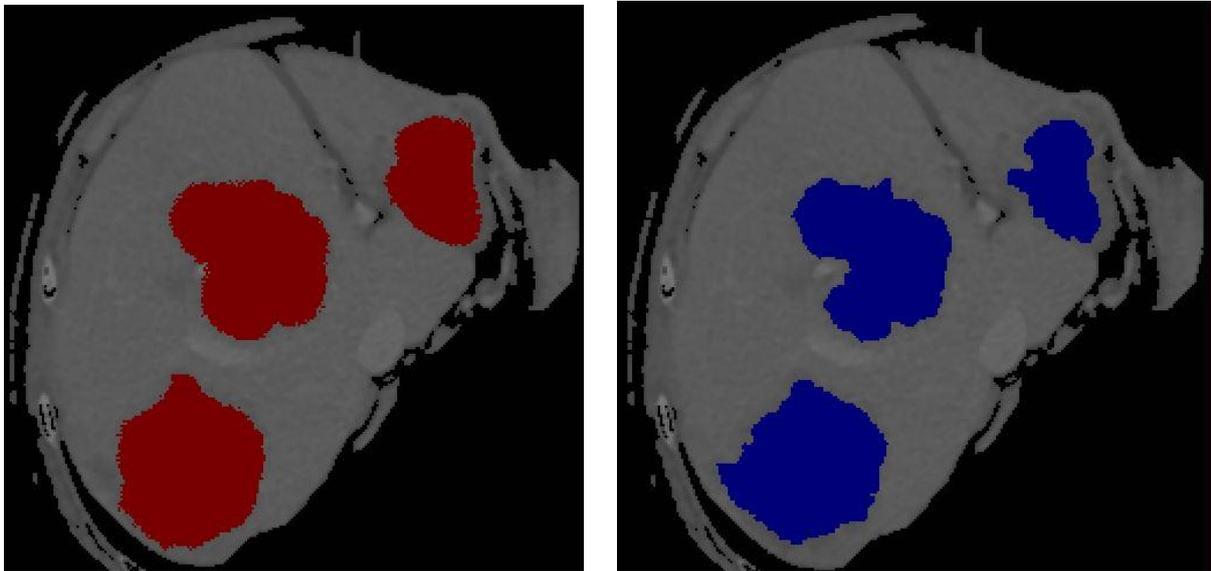


Figura 40. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set tramite i cluster ottenuti dai percentili. In blu sono riportate le maschere manuali

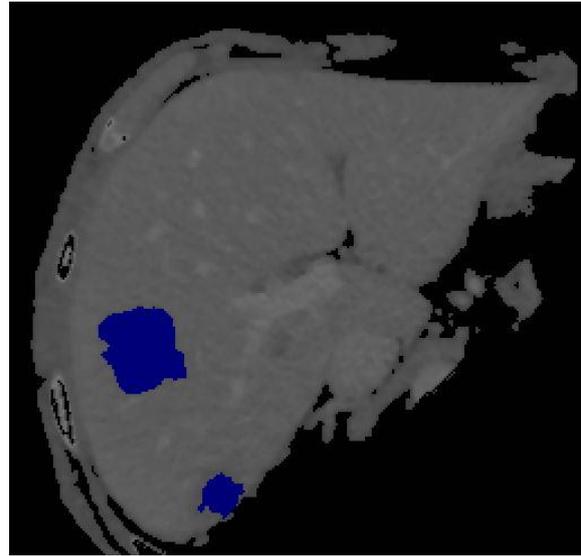
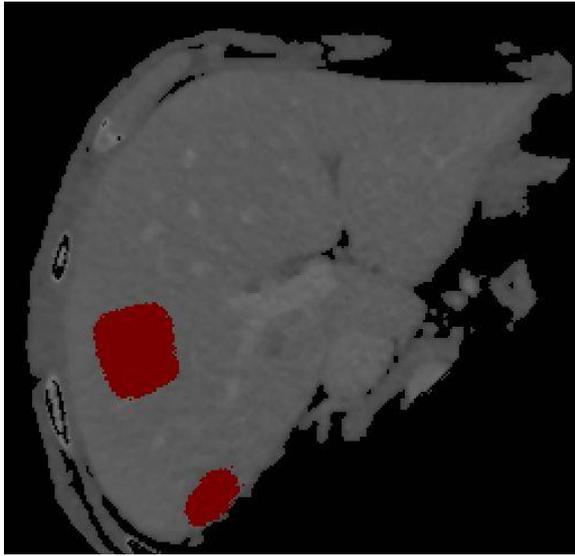


Figura 42. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set tramite i cluster ottenuti dai percentili. In blu sono riportate le maschere manuali.

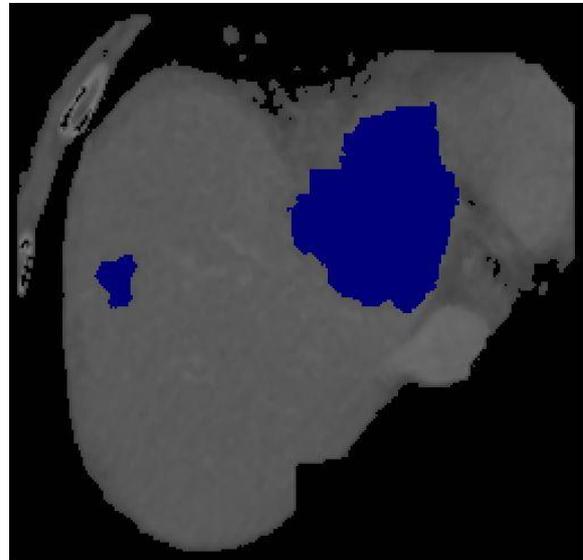
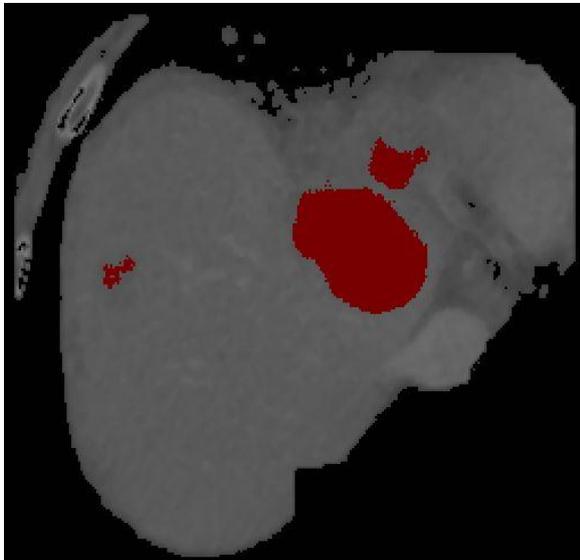


Figura 41. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set tramite i cluster ottenuti dai percentili. In blu sono riportate le maschere manuali.

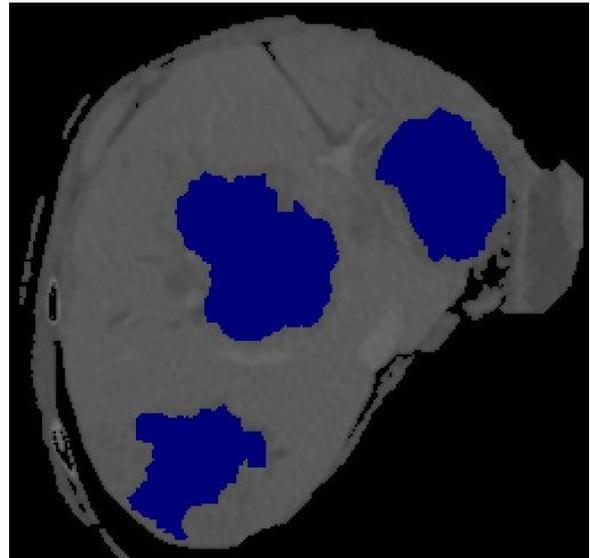
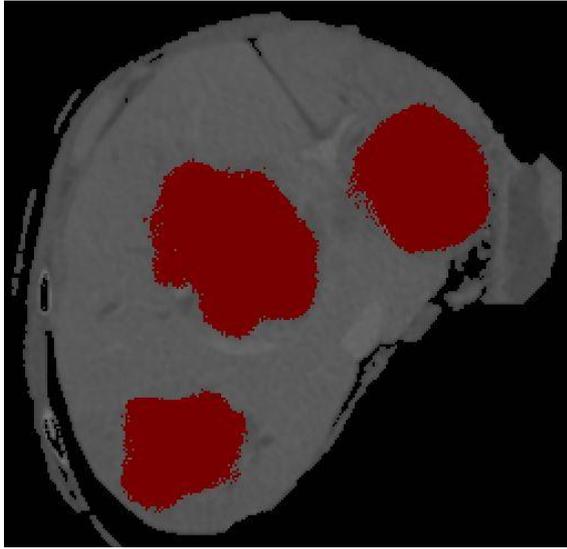


Figura 44. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set in maniera random. In blu sono riportate le maschere manuali.

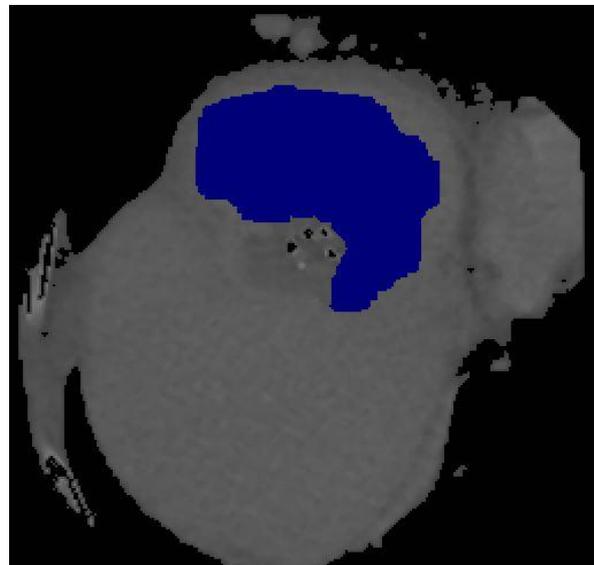
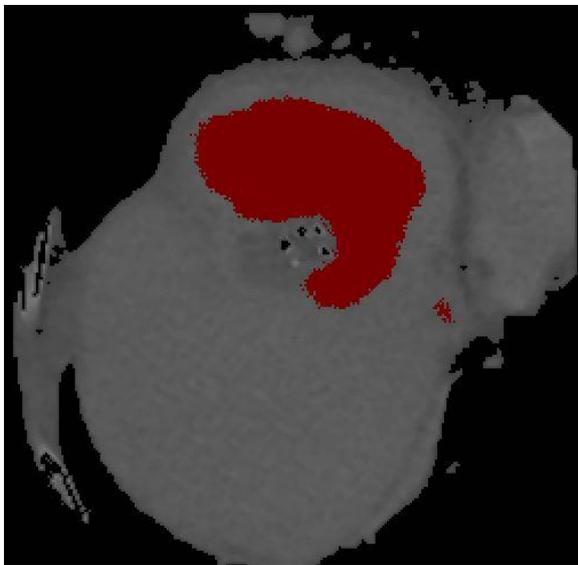


Figura 43. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set in maniera random. In blu sono riportate le maschere manuali.

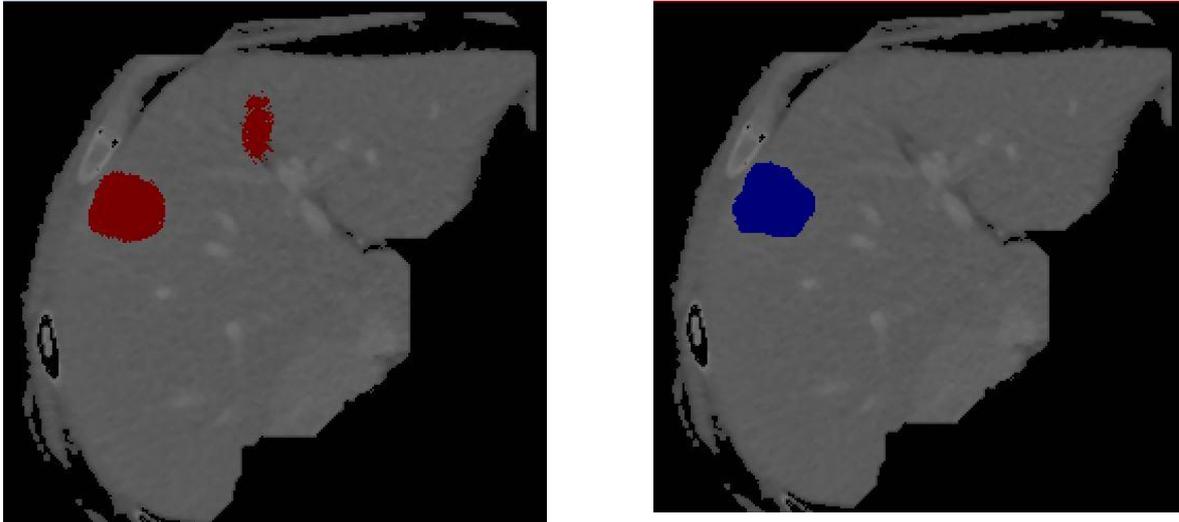


Figura 45. Esempi di maschere ottenute dalla Unet (in rosso) tramite la divisione del data set in maniera random. In blu sono riportate le maschere manuali.

MultiUnet

Le prestazioni della MultiUnet sono state valutate come fatto precedentemente: in ogni rete si è fatto variare il numero epoche fino a trovare le prestazioni migliori. In particolare, quest'ultime sono state fatte variare 5 a 150, le prime due a passo di cinque e poi a passo di 10. In questo caso, però, sono state confrontate le prestazioni quando il test è uguale al train e quando il test è il 30% degli elementi del cluster. Per valutare le performance sono state calcolate media e mediana dei due casi di dice, precision e recall.

Tabella 18. Andamento nel cluster 1 di media e mediana del train e del validation.

		Cluster1															
<i>n° epoche</i>		5	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
<i>media validation</i>	<i>Dice</i>	0.05	0.10	0.18	0.25	0.54	0.55	0.58	0.65	0.56	0.67	0.42	0.62	0.65	0.68	0.66	0.64
	<i>Precision</i>	0.08	0.25	0.19	0.42	0.67	0.68	0.67	0.72	0.73	0.78	0.71	0.77	0.73	0.72	0.68	0.75
	<i>Recall</i>	0.05	0.07	0.22	0.20	0.50	0.49	0.57	0.63	0.52	0.63	0.36	0.55	0.60	0.69	0.71	0.59
<i>media train</i>	<i>Dice</i>	0.06	0.15	0.26	0.43	0.82	0.86	0.83	0.91	0.85	0.92	0.77	0.96	0.90	0.97	0.93	0.97
	<i>Precision</i>	0.09	0.32	0.27	0.64	0.84	0.90	0.83	0.89	0.87	0.93	0.92	0.97	0.93	0.95	0.89	0.97
	<i>Recall</i>	0.06	0.11	0.30	0.36	0.82	0.85	0.84	0.93	0.85	0.91	0.71	0.95	0.88	0.99	0.97	0.98
<i>mediana validation</i>	<i>Dice</i>	0.05	0.07	0.13	0.06	0.58	0.62	0.68	0.75	0.63	0.74	0.37	0.68	0.67	0.78	0.77	0.75
	<i>Precision</i>	0.04	0.12	0.09	0.24	0.75	0.85	0.81	0.75	0.80	0.85	0.91	0.87	0.80	0.85	0.82	0.87
	<i>Recall</i>	0.04	0.05	0.21	0.04	0.55	0.59	0.69	0.75	0.62	0.69	0.24	0.63	0.65	0.77	0.77	0.69
<i>mediana train</i>	<i>Dice</i>	0.06	0.13	0.20	0.39	0.86	0.89	0.87	0.92	0.88	0.92	0.85	0.97	0.91	0.97	0.93	0.98
	<i>Precision</i>	0.06	0.28	0.25	0.76	0.87	0.91	0.89	0.90	0.90	0.95	0.95	0.98	0.95	0.95	0.90	0.97
	<i>Recall</i>	0.06	0.10	0.30	0.28	0.87	0.89	0.91	0.95	0.89	0.92	0.78	0.96	0.89	0.99	0.98	0.98

Tabella 19. Andamento nel cluster 2 di media e mediana del train e del validation.

Cluster2																	
<i>n° epoche</i>		5	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
<i>media validation</i>	<i>Dice</i>	0.41	0.31	0.60	0.77	0.79	0.75	0.77	0.62	0.76	0.79	0.80	0.80	0.66	0.80	0.25	0.81
	<i>Precision</i>	0.39	0.45	0.62	0.81	0.80	0.78	0.81	0.78	0.79	0.80	0.82	0.81	0.73	0.80	0.51	0.82
	<i>Recall</i>	0.51	0.26	0.62	0.77	0.80	0.77	0.78	0.57	0.77	0.80	0.82	0.83	0.63	0.84	0.17	0.84
<i>media train</i>	<i>Dice</i>	0.42	0.31	0.64	0.85	0.92	0.90	0.92	0.66	0.87	0.95	0.95	0.96	0.78	0.95	0.26	0.96
	<i>Precision</i>	0.39	0.45	0.65	0.89	0.91	0.91	0.94	0.79	0.88	0.94	0.95	0.95	0.79	0.92	0.50	0.94
	<i>Recall</i>	0.54	0.26	0.66	0.83	0.94	0.90	0.91	0.61	0.86	0.96	0.96	0.98	0.81	0.98	0.18	0.99
<i>mediana validation</i>	<i>Dice</i>	0.45	0.31	0.71	0.83	0.85	0.85	0.85	0.77	0.85	0.85	0.86	0.86	0.74	0.85	0.27	0.86
	<i>Precision</i>	0.39	0.46	0.75	0.86	0.84	0.87	0.88	0.89	0.87	0.88	0.87	0.85	0.81	0.85	0.60	0.87
	<i>Recall</i>	0.57	0.26	0.71	0.85	0.86	0.84	0.84	0.68	0.86	0.87	0.88	0.89	0.72	0.90	0.18	0.90
<i>mediana train</i>	<i>Dice</i>	0.43	0.31	0.73	0.90	0.94	0.93	0.94	0.81	0.90	0.96	0.97	0.97	0.84	0.96	0.28	0.97
	<i>Precision</i>	0.36	0.49	0.76	0.91	0.93	0.93	0.96	0.90	0.91	0.96	0.96	0.95	0.87	0.94	0.57	0.96
	<i>Recall</i>	0.59	0.25	0.76	0.88	0.95	0.92	0.94	0.72	0.91	0.97	0.98	0.99	0.87	0.98	0.19	0.99

Tabella 20. Andamento nel cluster 4 di media e mediana del train e del validation.

Cluster4																	
<i>n° epoche</i>		5	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
<i>media validation</i>	<i>Dice</i>	0.74	0.74	0.75	0.79	0.80	0.80	0.81	0.81	0.81	0.81	0.82	0.81	0.82	0.81	0.81	0.81
	<i>Precision</i>	0.74	0.74	0.78	0.79	0.80	0.80	0.81	0.81	0.81	0.80	0.81	0.82	0.81	0.81	0.81	0.81
	<i>Recall</i>	0.75	0.76	0.76	0.81	0.81	0.81	0.83	0.82	0.84	0.82	0.83	0.83	0.83	0.84	0.83	0.83
<i>media train</i>	<i>Dice</i>	0.04	0.06	0.17	0.60	0.80	0.80	0.85	0.91	0.86	0.92	0.99	0.98	0.97	0.98	1.00	1.00
	<i>Precision</i>	0.09	0.12	0.43	0.64	0.84	0.87	0.89	0.96	0.87	0.95	0.99	0.98	0.95	0.98	1.00	1.00
	<i>Recall</i>	0.04	0.05	0.11	0.59	0.78	0.75	0.82	0.87	0.86	0.89	0.99	0.98	0.99	0.99	1.00	1.00
<i>mediana validation</i>	<i>Dice</i>	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.87	0.86
	<i>Precision</i>	0.84	0.84	0.84	0.85	0.85	0.85	0.86	0.87	0.85	0.86	0.86	0.87	0.86	0.86	0.86	0.87
	<i>Recall</i>	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89
<i>mediana train</i>	<i>Dice</i>	0.05	0.06	0.21	0.58	0.81	0.80	0.87	0.92	0.87	0.92	0.99	0.98	0.97	0.99	1.00	1.00
	<i>Precision</i>	0.08	0.10	0.48	0.63	0.86	0.89	0.90	0.96	0.89	0.96	1.00	0.98	0.96	0.98	1.00	1.00
	<i>Recall</i>	0.04	0.05	0.14	0.58	0.78	0.78	0.85	0.89	0.86	0.91	0.99	0.98	0.99	0.99	1.00	1.00

Tabella 21. Andamento nel cluster 5 di media e mediana del train e del validation.

Cluster5																	
<i>n° epoche</i>		5	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
<i>media validation</i>	<i>Dice</i>	0.37	0.50	0.62	0.67	0.72	0.71	0.71	0.72	0.73	0.67	0.31	0.68	0.71	0.71	0.68	0.69
	<i>Precision</i>	0.37	0.60	0.75	0.76	0.71	0.76	0.71	0.74	0.78	0.78	0.38	0.77	0.78	0.75	0.75	0.78
	<i>Recall</i>	0.43	0.46	0.57	0.63	0.77	0.69	0.76	0.75	0.72	0.63	0.28	0.66	0.69	0.73	0.66	0.65
<i>media train</i>	<i>Dice</i>	0.42	0.63	0.82	0.91	0.93	0.94	0.94	0.95	0.95	0.93	0.40	0.89	0.97	0.96	0.97	0.96
	<i>Precision</i>	0.43	0.76	0.88	0.95	0.88	0.96	0.91	0.92	0.95	0.98	0.48	0.95	0.97	0.95	0.98	0.98
	<i>Recall</i>	0.50	0.58	0.77	0.87	0.98	0.93	0.97	0.98	0.96	0.90	0.36	0.86	0.96	0.97	0.96	0.93
<i>mediana validation</i>	<i>Dice</i>	0.35	0.62	0.74	0.81	0.85	0.84	0.82	0.83	0.85	0.81	0.24	0.82	0.84	0.84	0.82	0.83
	<i>Precision</i>	0.43	0.72	0.89	0.91	0.84	0.88	0.81	0.85	0.89	0.93	0.30	0.91	0.90	0.86	0.89	0.93
	<i>Recall</i>	0.55	0.48	0.69	0.78	0.89	0.84	0.89	0.88	0.84	0.77	0.24	0.81	0.84	0.85	0.79	0.80
<i>mediana train</i>	<i>Dice</i>	0.45	0.73	0.86	0.93	0.94	0.95	0.95	0.96	0.96	0.95	0.43	0.93	0.98	0.97	0.98	0.96
	<i>Precision</i>	0.50	0.86	0.92	0.96	0.91	0.97	0.93	0.94	0.96	0.99	0.59	0.96	0.98	0.96	0.99	0.99
	<i>Recall</i>	0.53	0.64	0.82	0.89	0.98	0.95	0.98	0.98	0.97	0.92	0.37	0.92	0.97	0.98	0.97	0.94

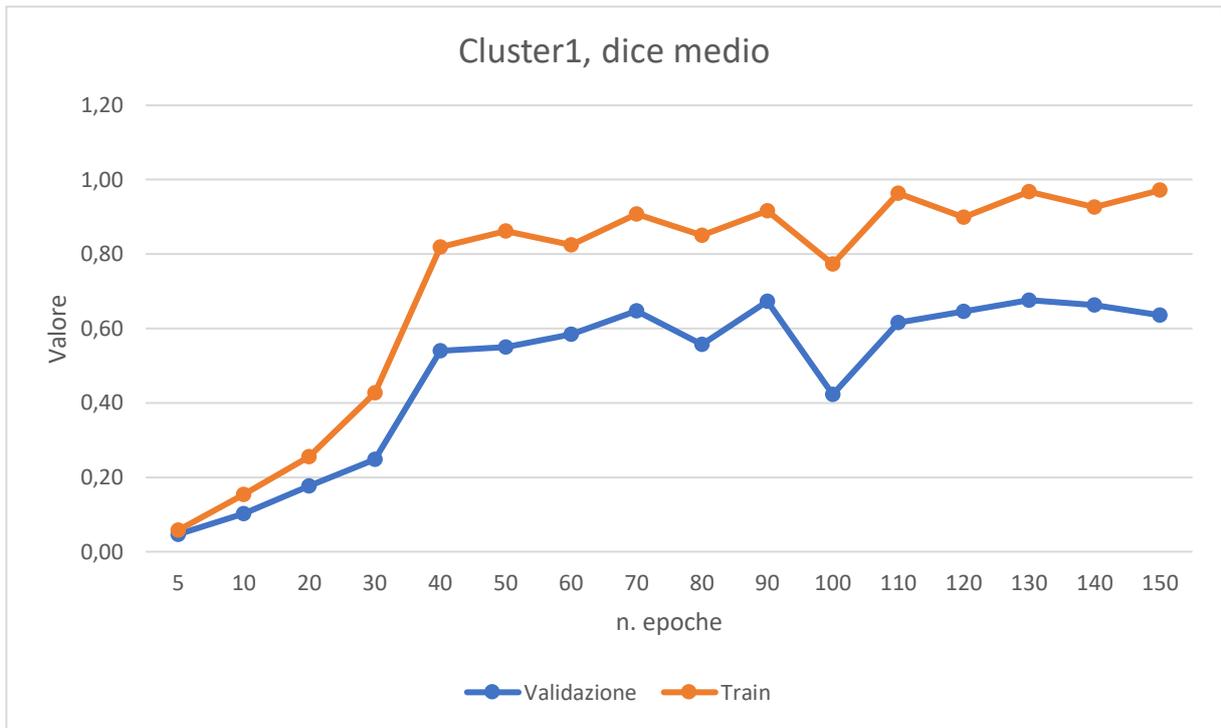


Figura 46. Andamento del dice medio nel train e nel validation, al variare del numero di epoche, nel cluster 1

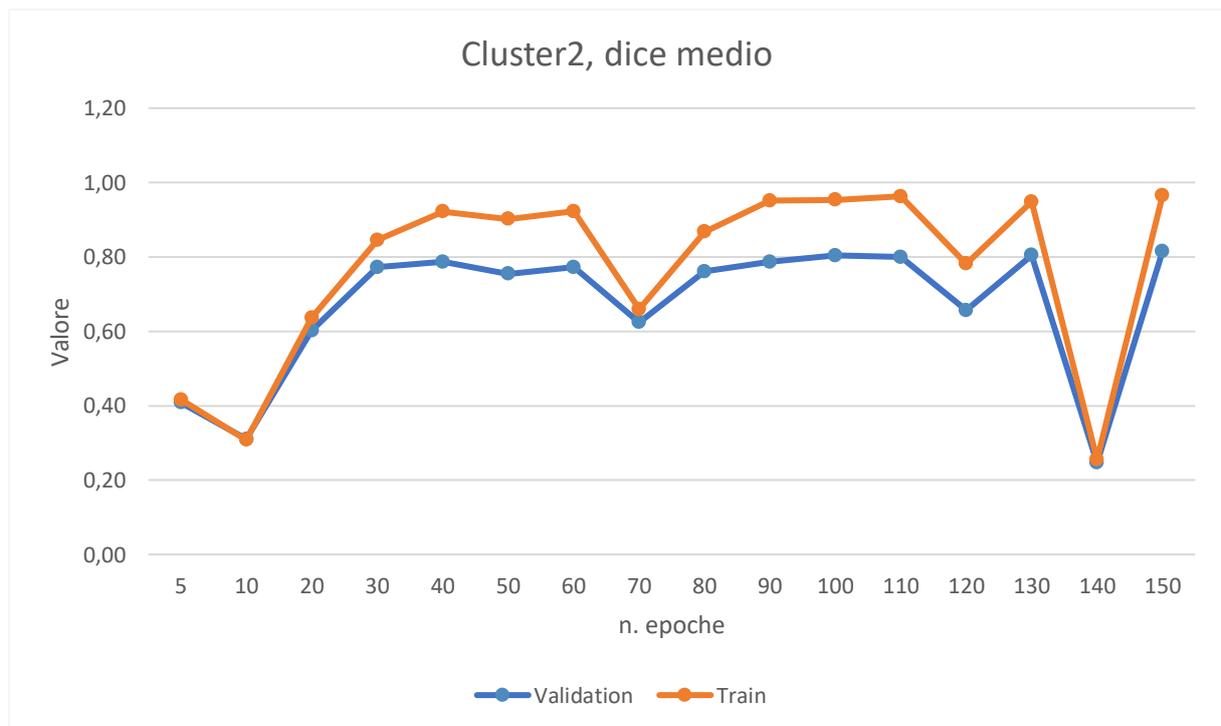


Figura 47. Andamento del dice medio nel train e nel validation, al variare del numero di epoche, nel cluster 2

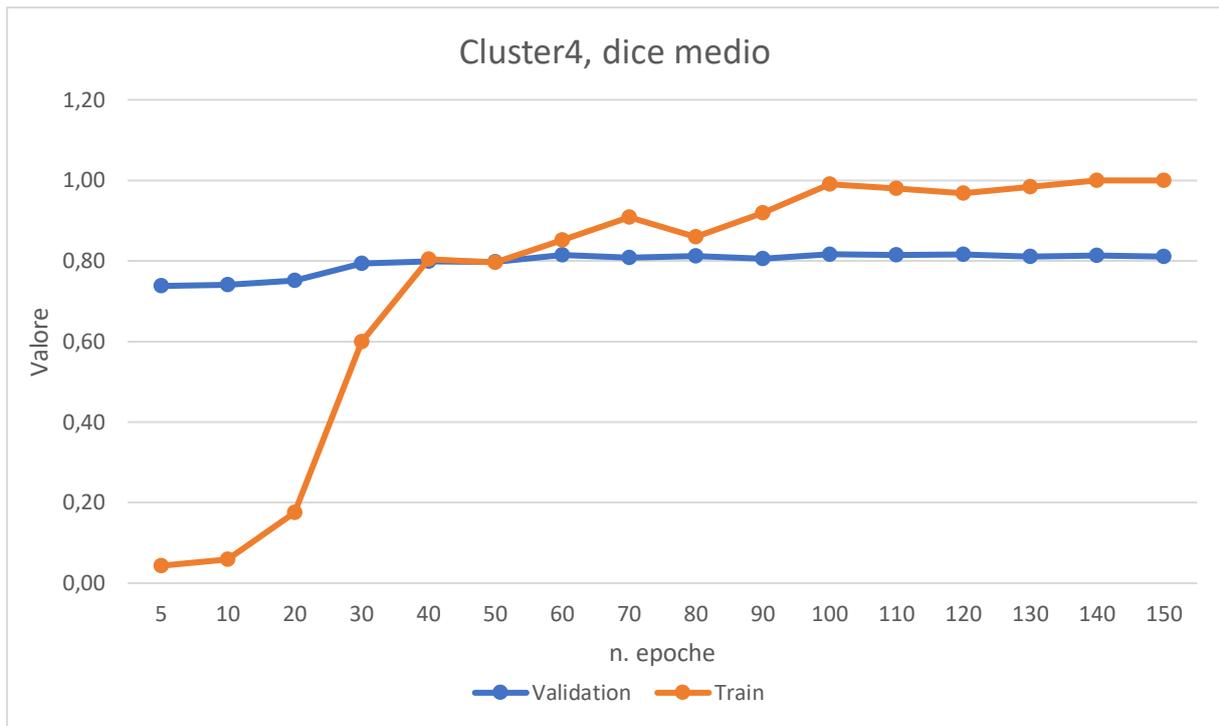


Figura 49. Andamento del dice medio nel train e nel validation, al variare del numero di epoche, nel cluster 4



Figura 48. Andamento del dice medio nel train e nel validation, al variare del numero di epoche, nel cluster 5.

Dalle prove effettuate si ottiene la giusta combinazione di epoche per ogni rete, che saranno 50 per la rete che utilizza il primo cluster, 40 per quella che utilizza il secondo, 70 per il quarto e 40 per il quinto.

Successivamente viene validato il sistema con la combinazione appena ottenuta e la maschera finale si ottiene applicando l'operatore *or* alle maschere ottenute. Il sistema viene validato usando l'intero dataset.

Tabella 22. Valori di media, deviazione standard, mediana, 25° e 75° percentile di dice, precision e recall della validazione. In basso a destra sono riportati il numero di falsi negativi.

	Dice	Precision	Recall
Media	0.70	0.62	0.86
Dev. Standard	0.23	0.25	0.18
Mediana	0.78	0.68	0.93
25° percentile	0.55	0.43	0.85
75° percentile	0.88	0.83	0.97

FN=8

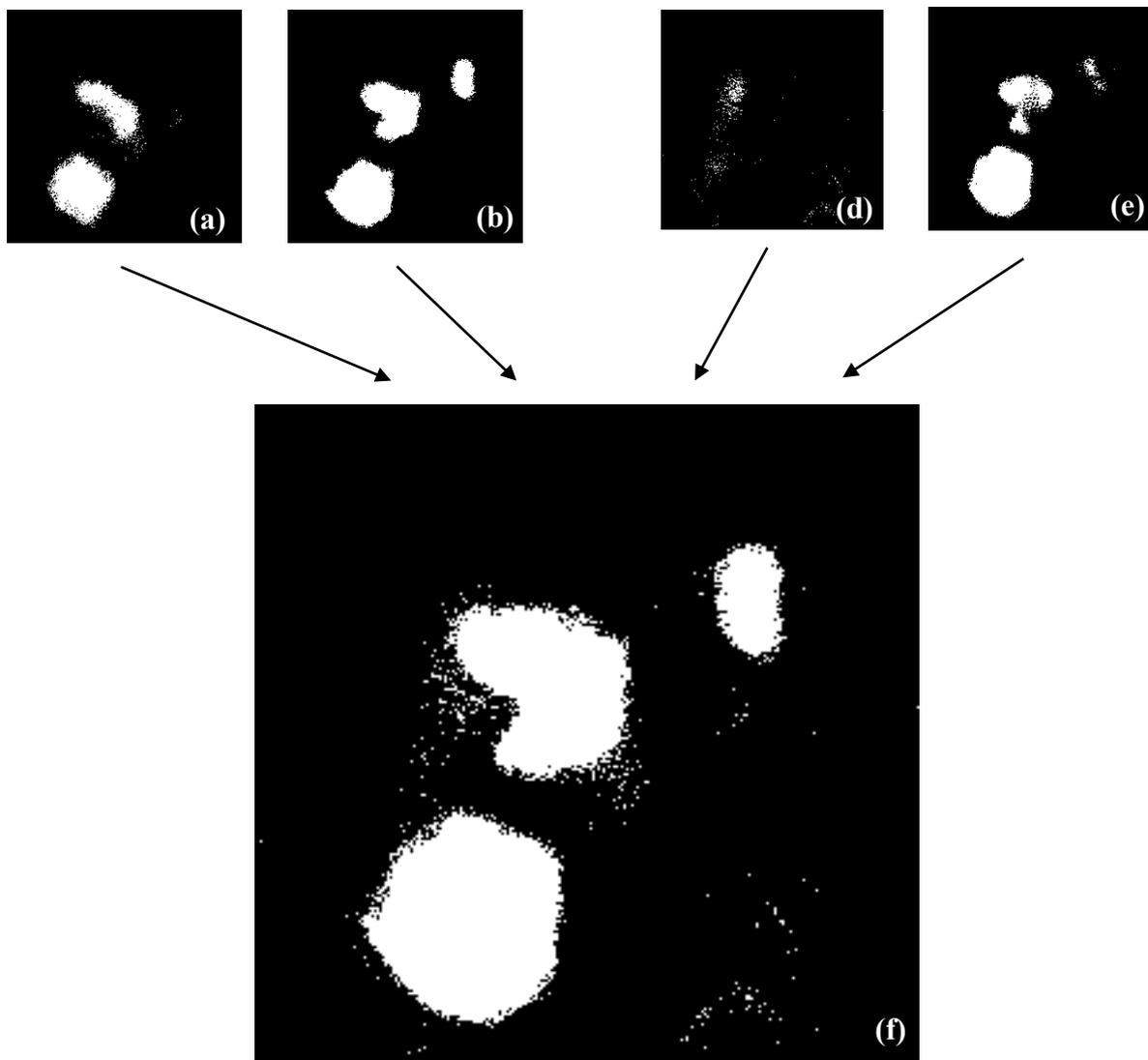


Figura 50. Esempio di funzionamento dell'uscita della MultiUnet. (a) è la maschera di uscita del cluster 1, (b) del cluster 2, (c) del cluster 4 e (d) del cluster 5. (f) è il risultato finale.

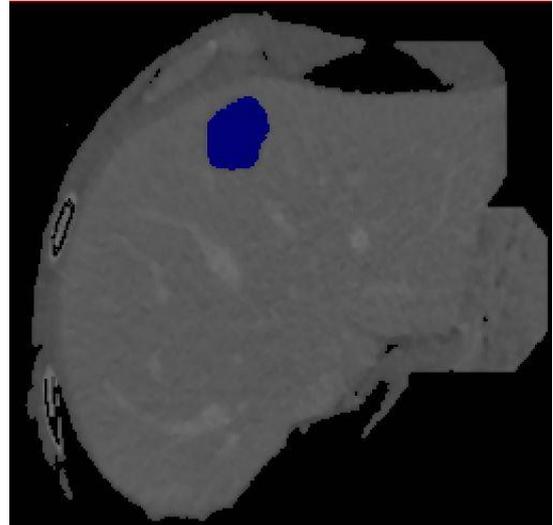
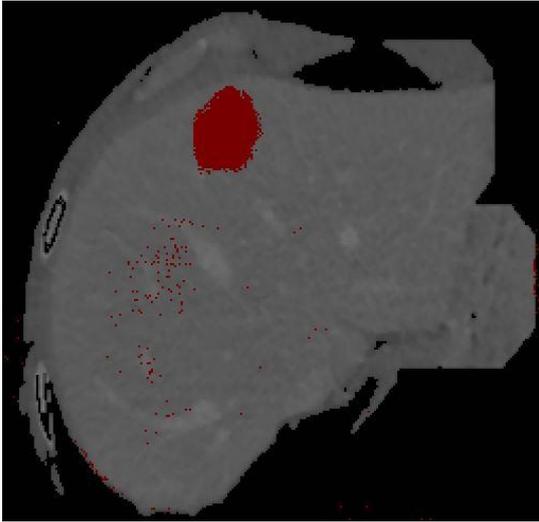


Figura 52. Confronto tra output della MultiUnet (in rosso) e maschera manuale (in blu).

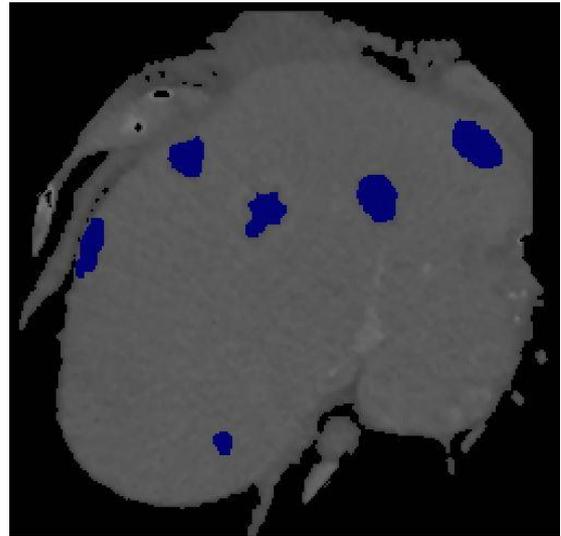
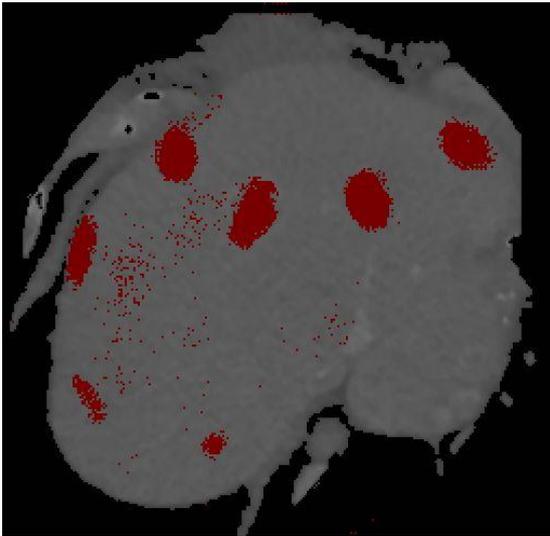


Figura 53. Confronto tra output della MultiUnet (in rosso) e maschera manuale (in blu).

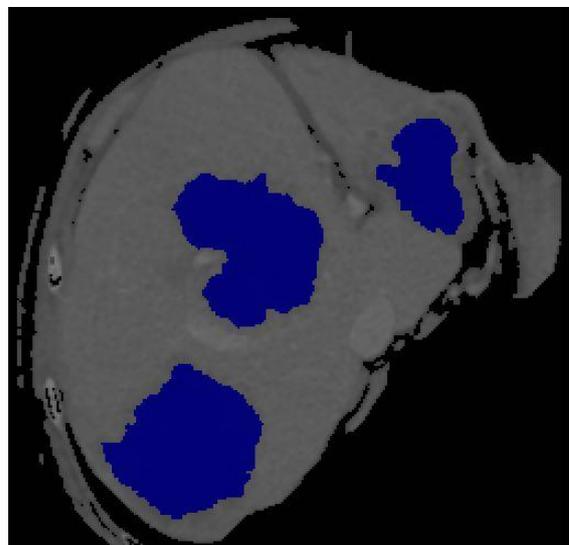
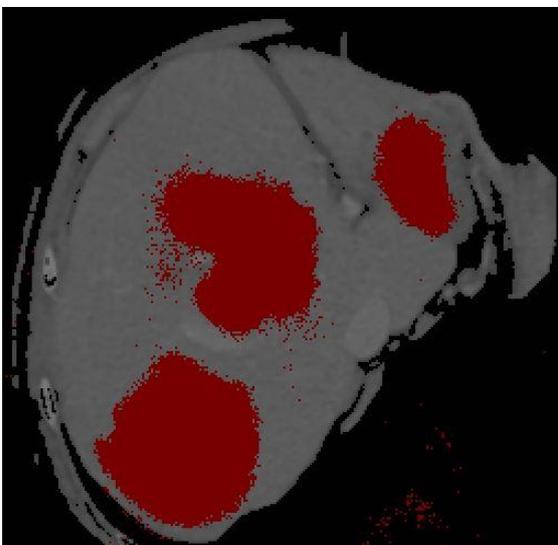


Figura 51. Confronto tra output della MultiUnet (in rosso) e maschera manuale (in blu).

Discussioni

Il mondo delle reti neurali è un mondo vasto e ancora in fase di scoperta. La Unet, in particolar modo, è un sistema nuovo, con non tante applicazioni, e con vaste possibilità di miglioramenti.

Con questa tesi sono state gettate delle basi sullo studio di segmentazioni di metastasi epatiche tramite l'utilizzo della Unet. Prendendo i risultati ottenuti, che sono paragonati i sistemi in uso, si può affermare che questo lavoro risultò un primo esempio della validità della rete Unet per la segmentazione di metastasi epatiche.

In particolare, lo studio di Vorontsov et al. (13) fornisce dei risultati con un dice medio \pm deviazione standard di 0.80 ± 0.11 , non andando a considerare le metastasi con un volume inferiore a 1.5 cm^3 , e 0.77 ± 0.12 considerando anche quest'ultime. In questo lavoro, la Unet, in maniera totalmente automatica, ha raggiunto un dice medio \pm deviazione standard di 0.78 ± 0.23 nel caso migliore (valore abbassato dai 5 falsi negativi, infatti si ha un 25° percentile pari a 0.76 e un 75° pari a 0.92), mentre la MultiUnet di 0.70 ± 0.23 . Sono entrambi sistemi nel loro stadio iniziale (soprattutto il secondo) con margini di miglioramento, quindi saranno delle buone strade da percorrere nello studio e caratterizzazione di metastasi epatiche.

Si potrebbero ottenere dei miglioramenti creando un dataset maggiormente bilanciato, soprattutto nel caso della MultiUnet, in modo tale da allenare tutte le reti in maniera più omogenea tra loro. Un altro miglioramento si potrebbe ottenere allenando una rete parallela solo per la segmentazione del fegato, in modo tale da avere una segmentazione più precisa e dare alla rete informazioni meno irrilevanti ma più informative.

Bibliografia

1. Adam R, de Gramont A, Figueras J, *et al.*: Managing synchronous liver metastases from colorectal cancer: A multidisciplinary international consensus. *Cancer Treat Rev*, 2015.
2. Lozano R, Naghavi M, Foreman K, *et al.*: Global and regional mortality from 235 causes of death for 20 age groups in 1990 and 2010: A systematic analysis for the Global Burden of Disease Study 2010. *Lancet*, 2012.
3. Sheth KR and Clary BM: Management of hepatic metastases from colorectal cancer. *Clin Colon Rectal Surg*, 2005.
4. Siriwardena AK, Mason JM, Mullamitha S, Hancock HC and Jegatheeswaran S: Management of colorectal cancer presenting with synchronous liver metastases. *Nat Rev Clin Oncol*, 2014.
5. van de Velde CJH: Colorectal cancer: Treatment of liver metastases of colorectal cancer. *Ann Oncol*, 2005.
6. Rui Lu, Marziliano P and Choon Hua Thng: Liver Tumor Volume Estimation By Semi-Automatic Segmentation Method., 2006.
7. Park S-J, Seo K-S and Park J-A: Automatic Hepatic Tumor Segmentation Using Statistical Optimal Threshold., 2010.
8. Ciecholewski M and Ogiela MR: Automatic Segmentation of Single and Multiple Neoplastic Hepatic Lesions in CT Images. In: *Nature Inspired Problem-Solving Methods in Knowledge Engineering*, 2007.
9. Yuanzhong L, Hara S and Shimura K: A machine learning approach for locating boundaries of liver tumors in CT images. In: *Proceedings - International Conference on Pattern Recognition*, 2006.
10. Jolly MP and Grady L: 3D general lesion segmentation in CT. In: *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Proceedings, ISBI*, 2008.
11. Smeets D, Loeckx D, Stijnen B, De Dobbelaer B, Vandermeulen D and Suetens P: Semi-automatic level set segmentation of liver tumors combining a spiral-scanning technique with supervised fuzzy pixel classification. *Med Image Anal*, 2010.
12. Kuhnigk JM, Dicken V, Bornemann L, Bakai A, Wormanns D, Krass S and Peitgen HO: Morphological segmentation and partial volume analysis for volumetry of solid pulmonary lesions in thoracic CT scans. *IEEE Trans Med Imaging*, 2006.
13. Vorontsov E, Tang A, Roy D, Pal CJ and Kadoury S: Metastatic liver tumour segmentation with a neural network-guided 3D deformable model. *Med Biol Eng Comput* 55: 127–139, 2017.
14. Ronneberger O, Fischer P and Brox T: U-net: Convolutional networks for biomedical image segmentation. *Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)* 9351: 234–241, 2015.
15. Biasco G, Derenzini E, Fanello S and Brandi G: Liver metastases from colorectal cancer., 2005.