

# REPORT MARZO 2024



WILLIAM BASSOLINO

## OBIETTIVI

1. Studio di **metriche** per valutare il grado di incertezza della rete sulle 20 maschere e softmax derivanti dal metodo MonteCarlo Dropout
2. Presentazione di queste metriche attraverso grafici

## METRICHE

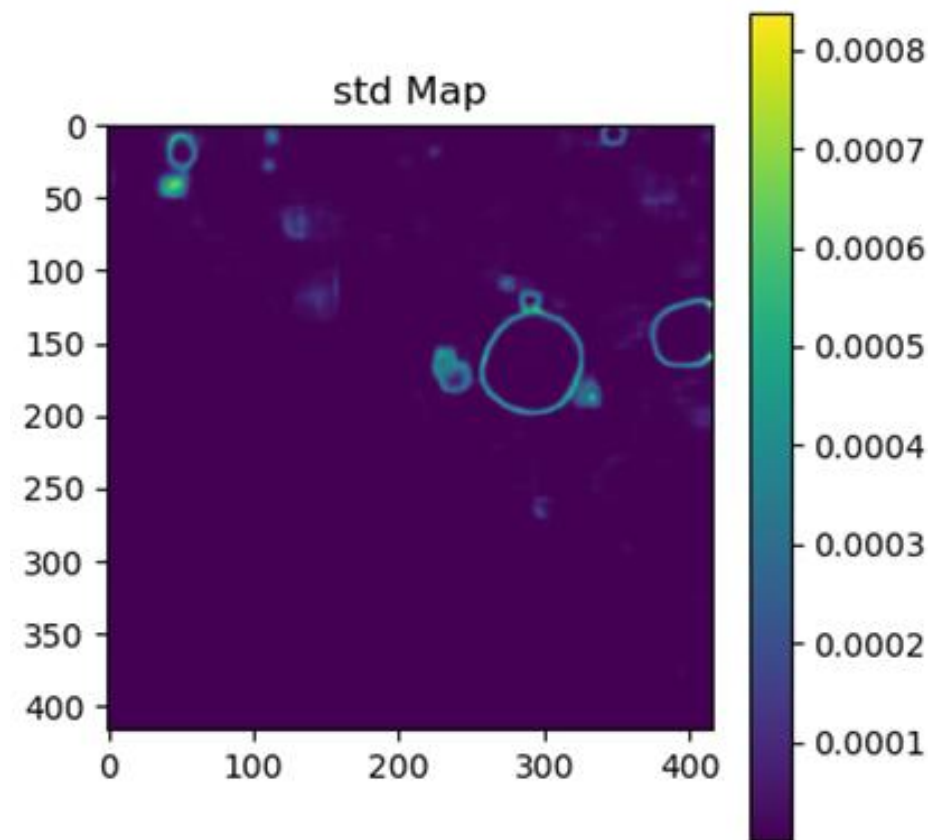
1. Metrica basata sulla **standard deviation** (da qui: **std**)
2. Metrica basata sull'**Entropia di Shannon** (da qui: **Entropy**)
3. Metrica basata sullo studio dei relativi dice tra le 20 maschere derivanti dal MonteCarlo Dropout (da qui: **dice mat**)

## METRICA BASATA SULLA STANDARD DEVIATION - STD

Per calcolare questa metrica, sono stati svolti i seguenti passaggi preliminari:

- Creazione di una matrice quadridimensionale in cui salvare i dati delle 20 softmax generate dalle 20 inference MonteCarlo sulla stessa immagine di partenza (denominata *softmax\_matrix*)
- Alla matrice, per la classe selezionata (in questo caso: 1), viene applicata una funzione (in basso in questa slide) che crea una *std map* bidimensionale, della quale ogni punto  $i,j$  è la standard deviation del valore dei 20 corrispondenti punti  $i,j$  delle 20 softmax
- Il risultato della std map è una mappa di standard deviation, come ad esempio l'immagine a destra
- Il valore associato ad ogni immagine del test set sarà la somma dei valori della std map, moltiplicato per la maschera manuale (ground truth)

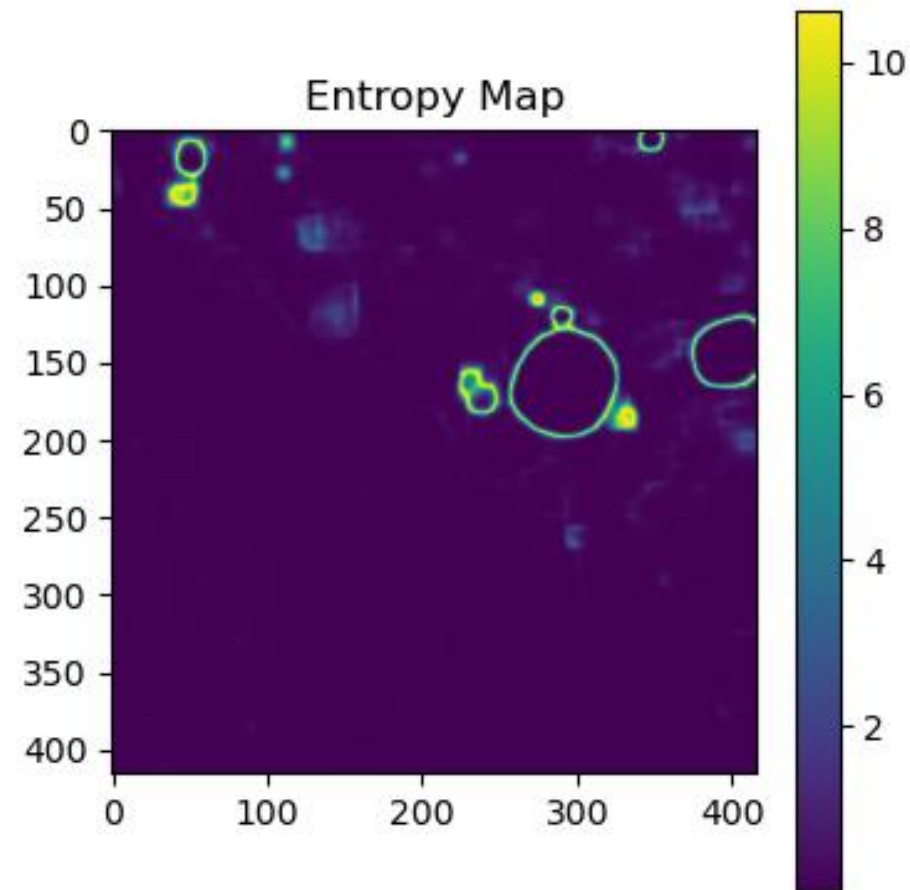
```
def std_map(softmax_matrix):  
    shape = np.shape(softmax_matrix)  
    std_map = np.zeros([shape[0],shape[1]])  
    for i in range(shape[0]):  
        for j in range(shape[1]):  
            ij_vector = softmax_matrix[i,j,1,:]  
            std_ij = np.nanstd(ij_vector)  
            std_map[i,j] = std_ij  
    return std_map
```



## METRICA BASATA SULL'ENTROPIA - ENTROPY

Per calcolare questa metrica, sono stati svolti i seguenti passaggi preliminari:

- Creazione di una matrice quadridimensionale in cui salvare i dati delle 20 softmax generate dalle 20 inference MonteCarlo sulla stessa immagine di partenza (denominata *softmax\_matrix*)
- Alla matrice, per la classe selezionata (in questo caso: 1), viene applicata una funzione (in basso in questa slide) che crea una *entropy map* bidimensionale, della quale ogni punto  $i,j$  è l'entropia del valore dei 20 corrispondenti punti  $i,j$  delle 20 softmax. L'entropia implementata è l'entropia di *Shannon*.
- Il risultato della entropy map è una mappa di entropia, come ad esempio l'immagine a destra
- Il valore associato ad ogni immagine del test set sarà la somma dei valori della entropy map, moltiplicato per la maschera manuale (ground truth)



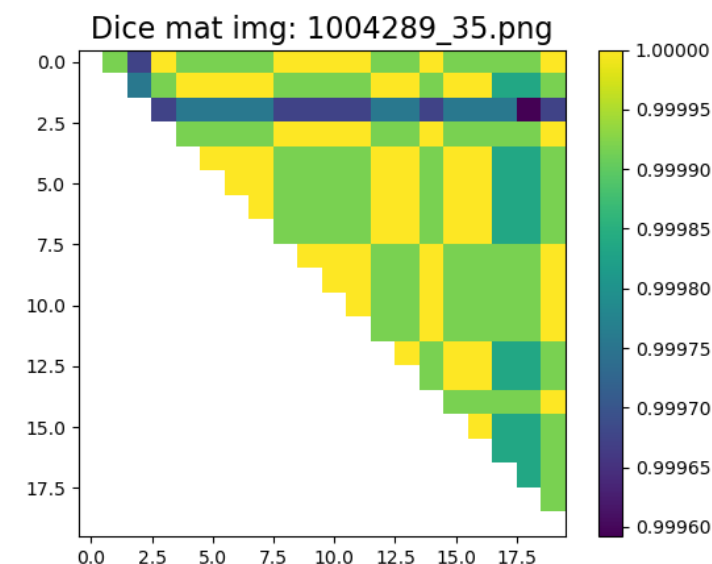
```
def entropy_map(softmax_matrix):  
    shape = np.shape(softmax_matrix)  
    entropy_map = np.zeros([shape[0], shape[1]]).astype(float)  
    for i in range(shape[0]):  
        for j in range(shape[1]):  
            ij_vector = softmax_matrix[i, j, :]  
            entropy_map[i, j] = -(np.multiply(ij_vector[ij_vector!=0], np.log2(ij_vector[ij_vector!=0]))).sum()  
    return entropy_map
```

## METRICA BASATA SUI DICE – DICE MAT

Per calcolare questa metrica, sono stati svolti i seguenti passaggi preliminari:

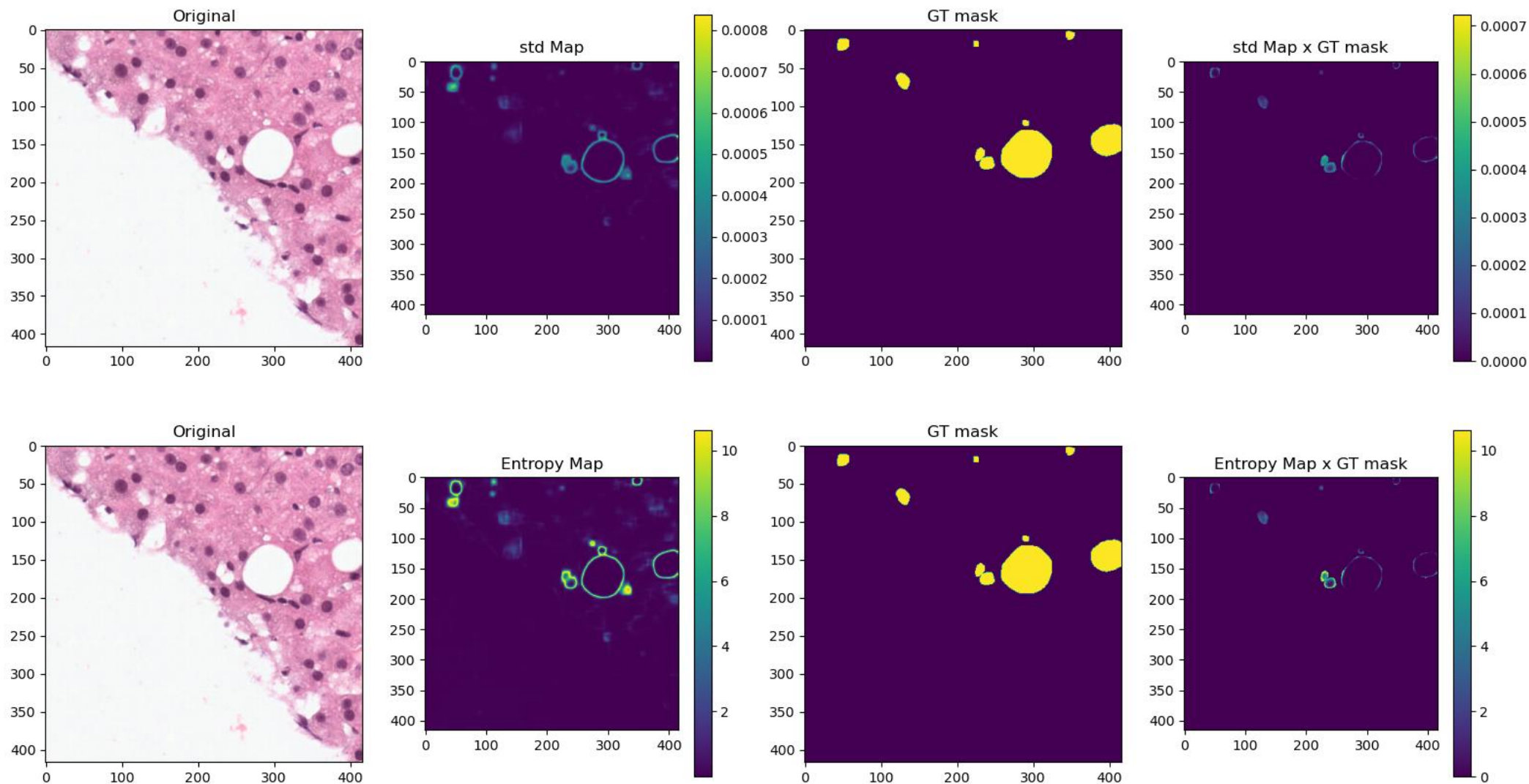
- Creazione di una matrice tridimensionale in cui salvare i dati delle 20 segmentazioni generate dalle 20 inference MonteCarlo sulla stessa immagine di partenza (denominata *seg\_matrix*)
- Alla matrice viene applicata una funzione (in basso in questa slide) che crea una matrice diagonale alta, della quale ogni punto  $i,j$  è il coefficiente di Sorensen-Dice tra la segmentazione  $i$  e la segmentazione  $j$  selezionate tra le 20 maschere di segmentazione derivanti dall'applicazione del MonteCarlo Dropout.
- Il valore associato ad ogni immagine del test set sarà la *standard deviation* dei valori della matrice triangolare alta (dice matrix)
  - È importante notare come questa metrica sia **l'unica delle tre metriche proposte a NON dipendere dal ground truth**, derivando unicamente dalle 20 immagini MonteCarlo

```
dice_mat_temp = -np.ones((20,20))
dice_array_GT = []
for i in range(20):
    for j in range(i+1,20):
        dice_mat_temp[i,j] = dice(seg_matrix[:, :, i], seg_matrix[:, :, j])
        dice_array_GT.append(dice(seg_matrix[:, :, i], GT_mask))
dice_mat_temp[dice_mat_temp<0] = np.nan
```



## ESEMPIO: IMMAGINE 1004289\_35

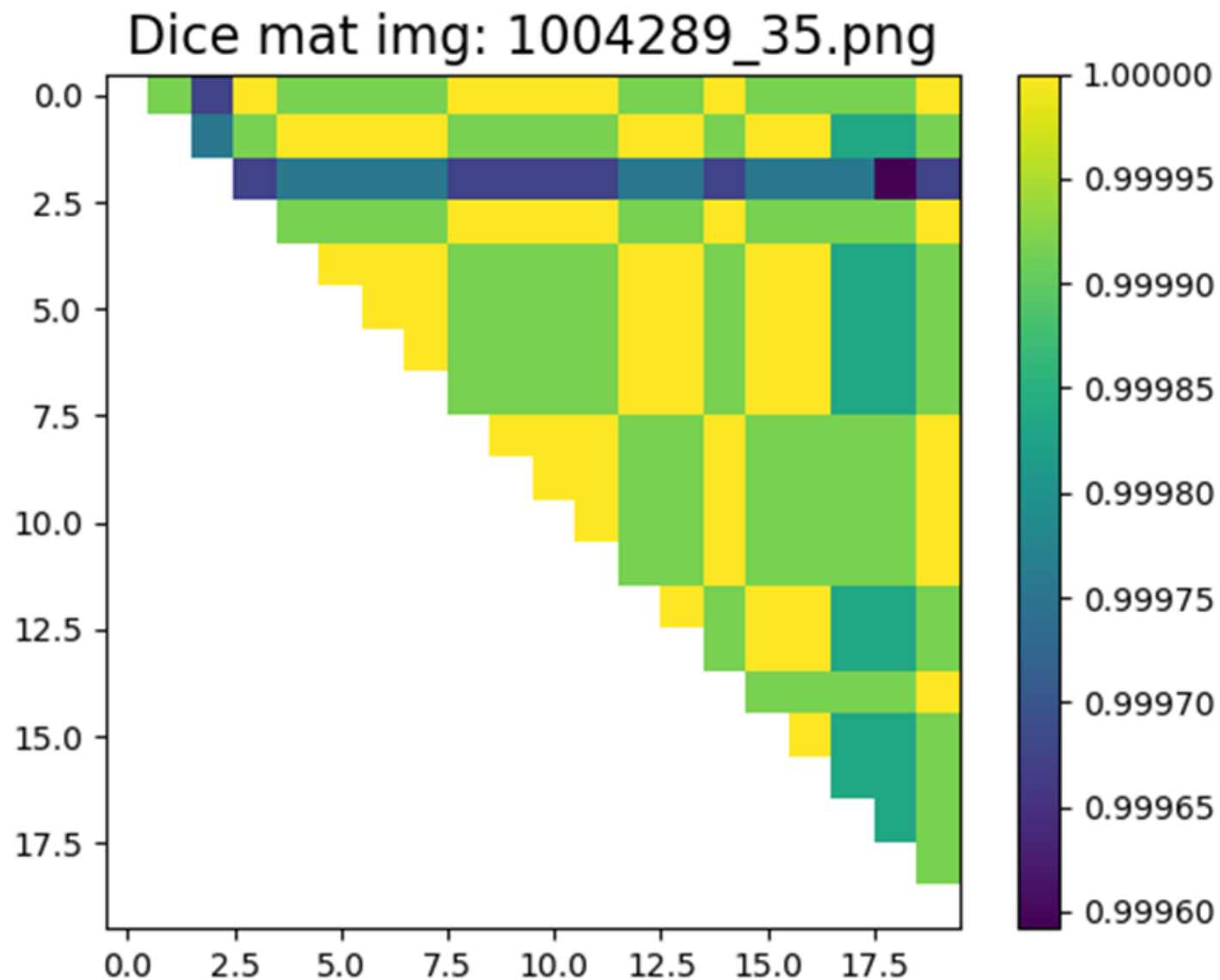
Nell'esempio si possono apprezzare la std map (in alto) e l'entropy map (in basso). Nell'ultima colonna, la maschera manuale e le mappe vengono moltiplicate tra di loro: è da questa matrice che vengono calcolate le metriche, sommando i valori non nulli di ciascuna delle due.





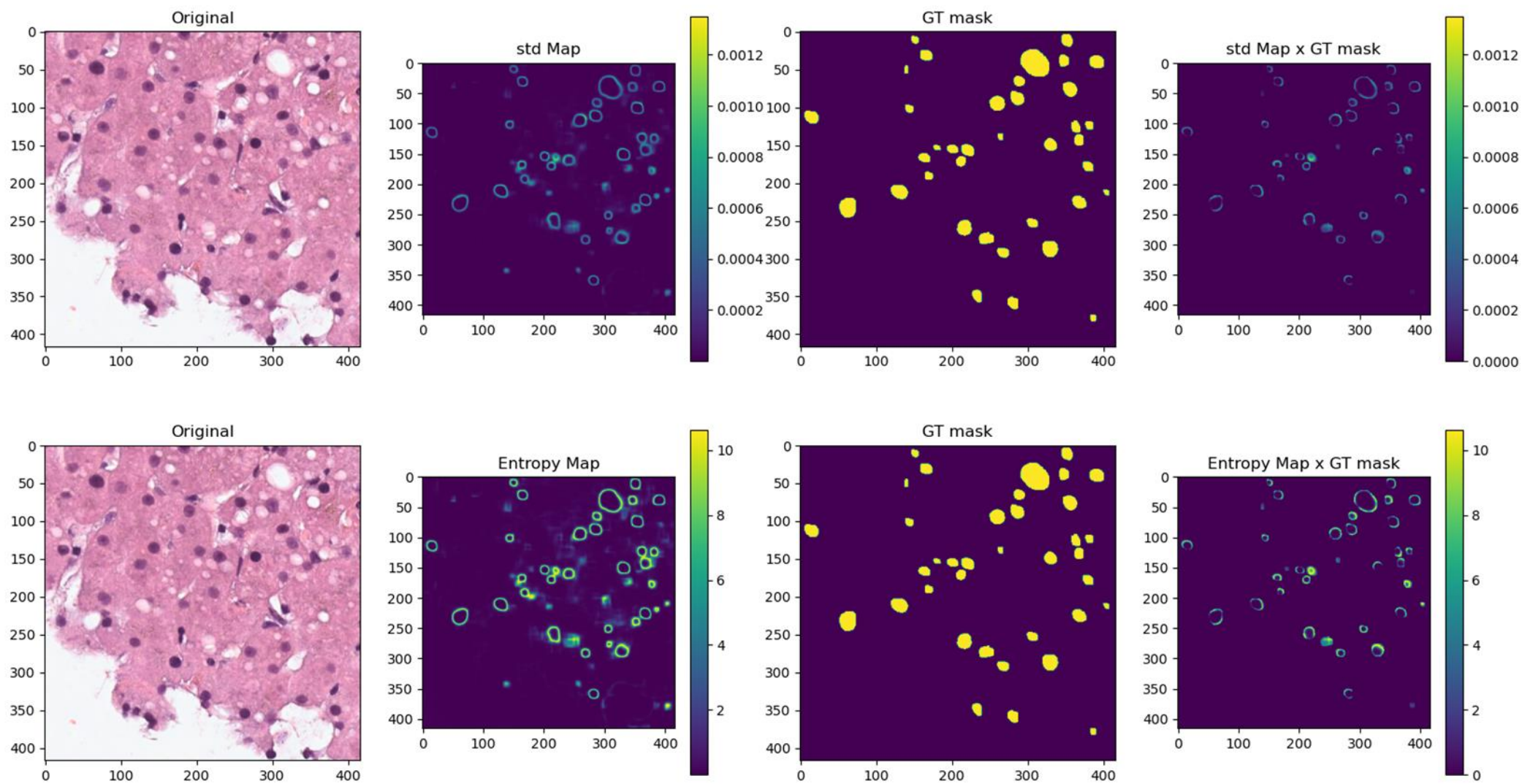
## ESEMPIO: IMMAGINE 1004289\_35

Nell'immagine a destra, si può apprezzare come l'immagine 3 derivante dal MonteCarlo abbia risultato leggermente diverso dalle altre 19 segmentazioni con Dropout.



## ESEMPIO: IMMAGINE 1004935\_97

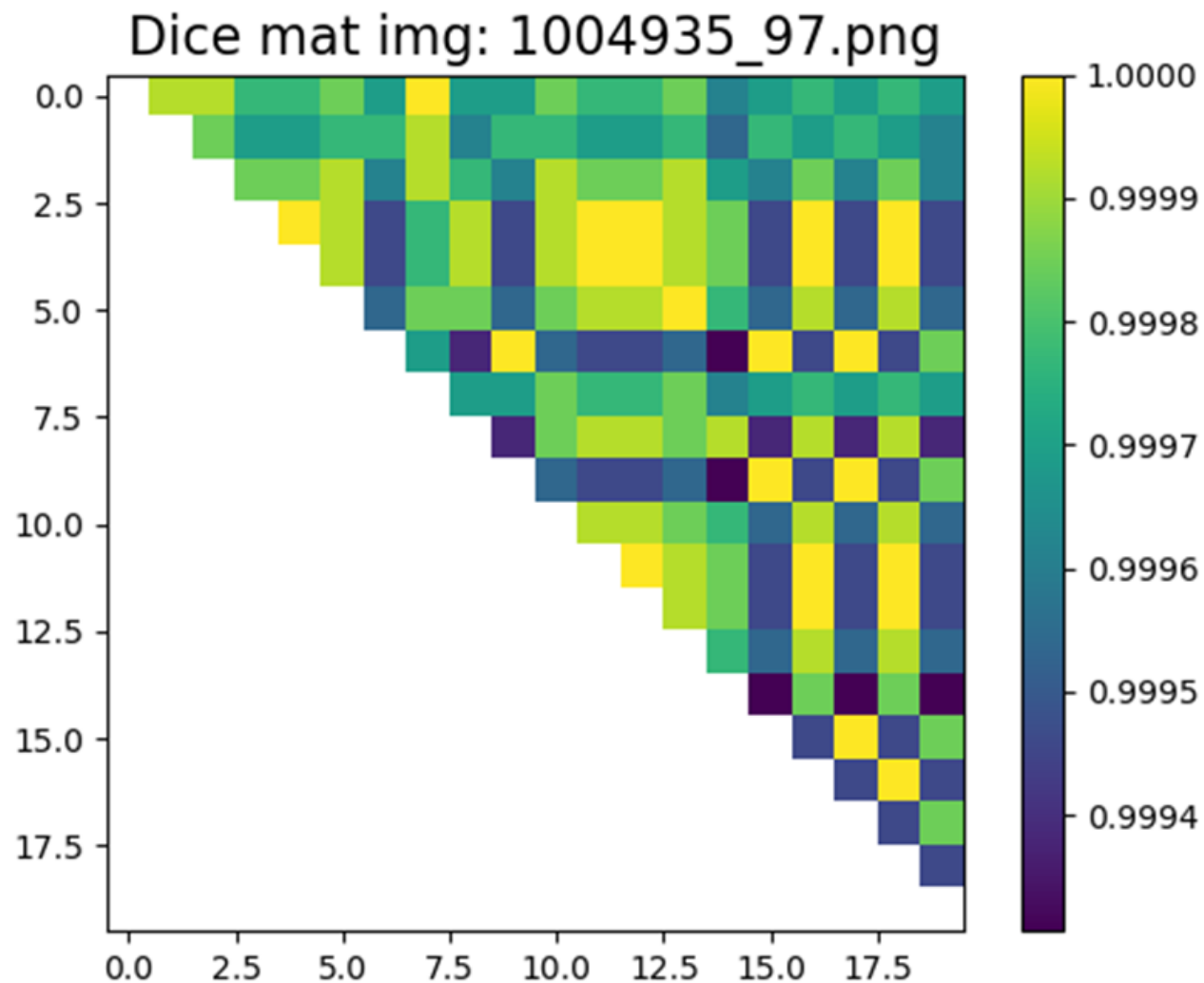
Nell'esempio si possono apprezzare la std map (in alto) e l'entropy map (in basso). Nell'ultima colonna, la maschera manuale e le mappe vengono moltiplicate tra di loro: è da questa matrice che vengono calcolate le metriche, sommando i valori non nulli di ciascuna delle due.





## ESEMPIO: IMMAGINE 1004935\_97

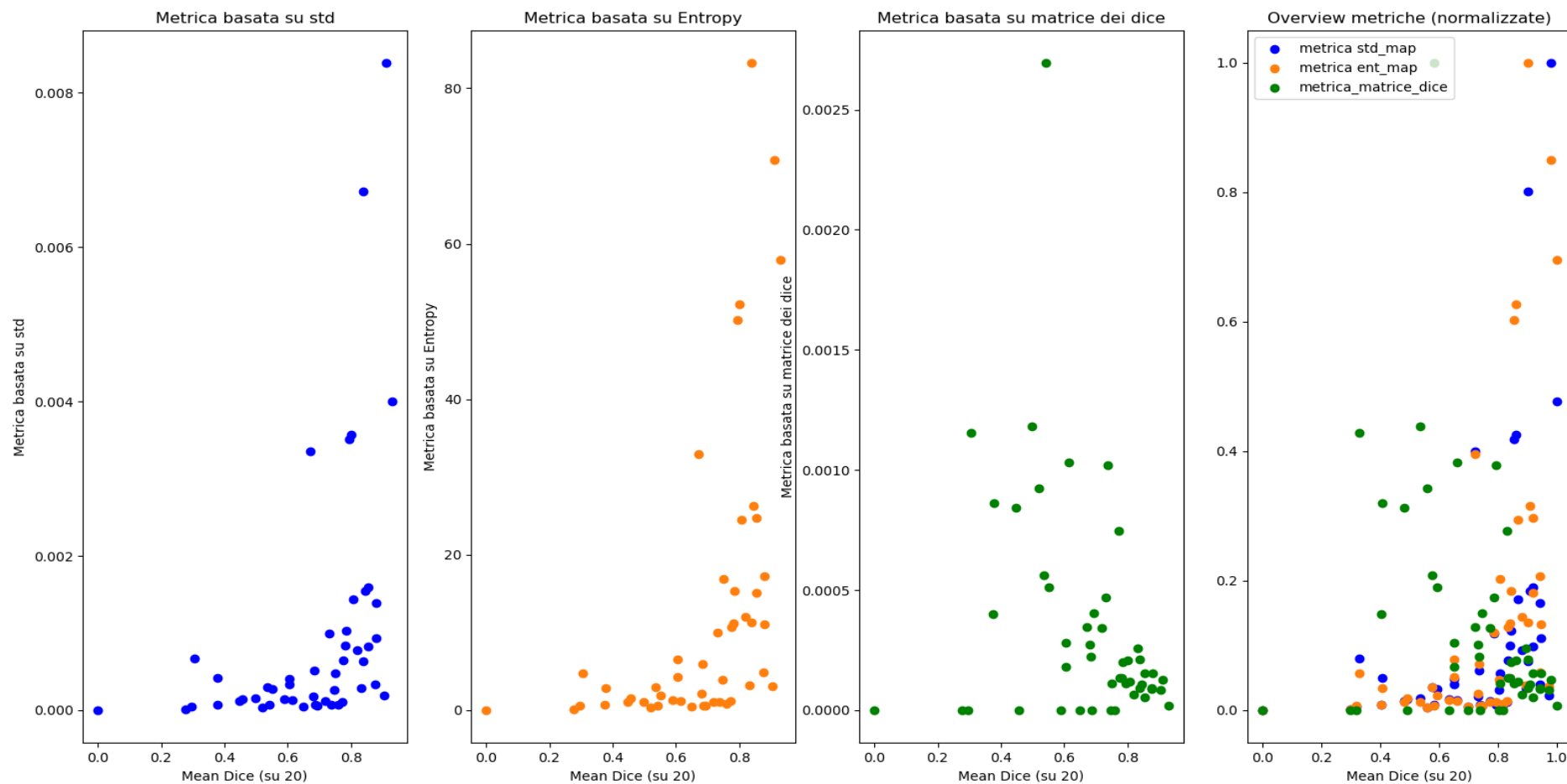
Nell'immagine a destra si può notare come la rete possa essere incerta rispetto alla segmentazione: i valori di dice –che idealmente dovrebbero raggiungere l'unità, non sono stabili, c'è una variazione percettibile.



## ANALISI PRELIMINARE DELLE METRICHE

I quattro scatterplot di cui sotto illustrano le relazioni tra il dice medio (calcolato sulle 20 immagini MonteCarlo in relazione alla maschera Ground Truth) con, rispettivamente: Metrica basata std (in **blu**), Metrica basata sull'entropia (in **arancione**), Metrica basata sulla matrice dei dice (in **verde**)

Scatterplot metriche x dice medio per immagine



## ANALISI PRELIMINARE DELLE METRICHE

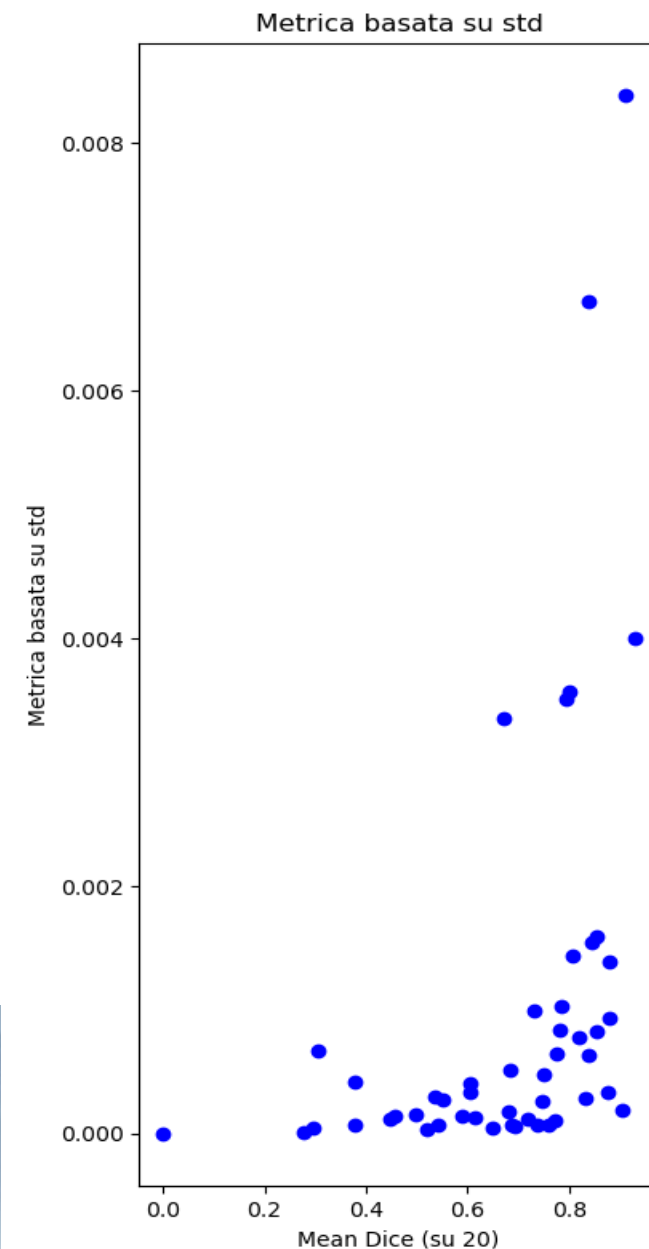
### METRICA BASATA SULLA STD

Idealmente, i punti di questo scatterplot dovrebbero concentrarsi nell'angolo in basso a destra: il dice medio ideale è 1, la std ideale è 0.

A livello di dice, la rete sembrerebbe non avere prestazioni tendenti all'idealità (ce ne si può accorgere da un'analisi visiva analogica: ci sono immagini la cui segmentazione della rete non è per niente corretta), e questo spiega perché c'è diffusione sull'asse delle ascisse.

A livello di std, sembrerebbe poter esserci una tendenza ad avere valori più bassi quando i valori di dice aumentano: è il comportamento che ci si aspetterebbe, anche se ci sono dei picchi a valori di dice più alti; c'è comunque una grande variabilità, non sembra esserci una relazione lineare.

In ogni caso, il campione di 50 immagini potrebbe essere non abbastanza elevato per trarre conclusioni certe.



## ANALISI PRELIMINARE DELLE METRICHE

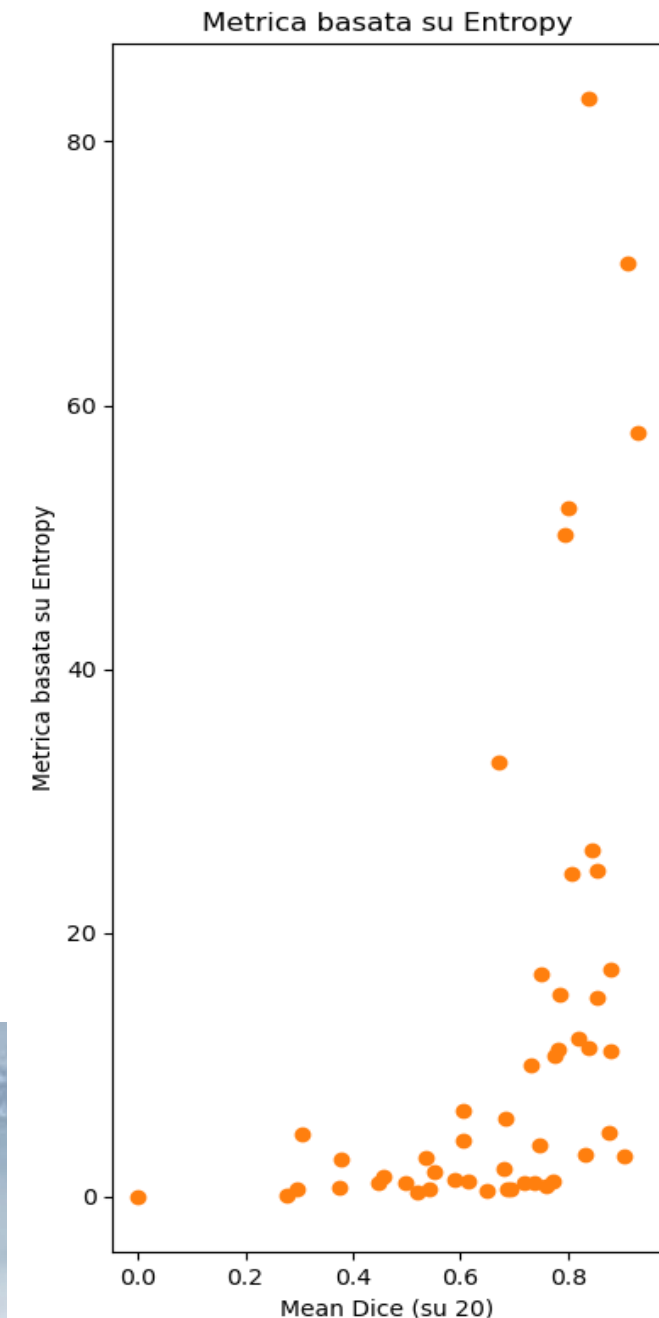
### METRICA BASATA SULL'ENTROPIA

Idealmente, i punti di questo scatterplot dovrebbero concentrarsi nell'angolo in basso a destra: il dice medio ideale è 1, l'entropia ideale è 0.

A livello di dice il discorso è lo stesso della slide precedente, e questo spiega perché c'è diffusione sull'asse delle ascisse.

A livello di entropia, sembrerebbe poter esserci una tendenza ad avere valori più bassi quando i valori di dice aumentano, con però dei picchi di entropia proprio dove il dice aumenta; c'è comunque una grande variabilità, non sembra esserci una relazione lineare.

In ogni caso, il campione di 50 immagini potrebbe essere non abbastanza elevato per trarre conclusioni certe.



## ANALISI PRELIMINARE DELLE METRICHE

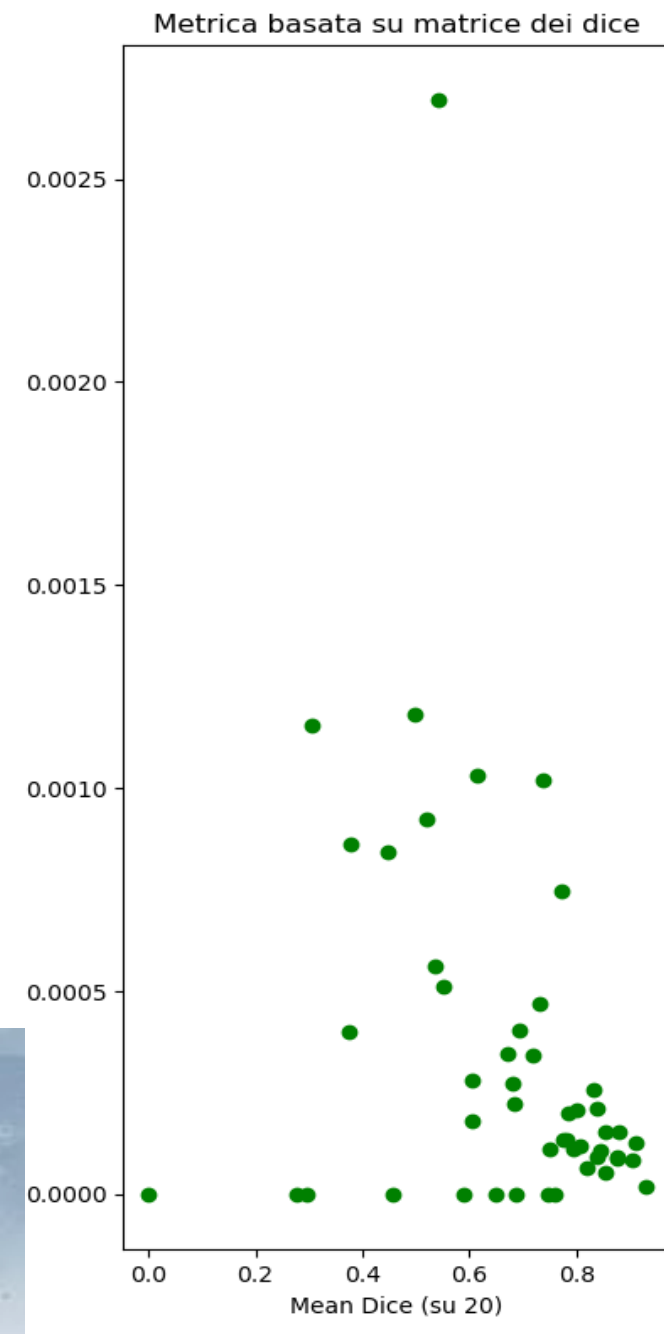
### METRICA BASATA SULLA MATRICE DEI DICE

Idealmente, i punti di questo scatterplot dovrebbero concentrarsi nell'angolo in basso a destra: il dice medio ideale è 1, la standard deviation ideale dei dice della matrice dei dice è 0.

A livello di dice medio il discorso è lo stesso delle slide precedenti, e questo spiega perché c'è diffusione sull'asse delle ascisse.

A livello di standard deviation dei dice, sembrerebbe poter esserci una tendenza ad avere valori più bassi quando i valori di dice aumentano: è il comportamento che ci si aspetterebbe, e sembrerebbe che questo comportamento sia più accentuato del comportamento delle metriche precedenti; in ogni caso, il campione di 50 immagini potrebbe essere non abbastanza elevato per trarre conclusioni certe.

È importante ricordare che la standard deviation della matrice dei dice è l'unica delle tre metriche a **NON dipendere dal ground truth**.

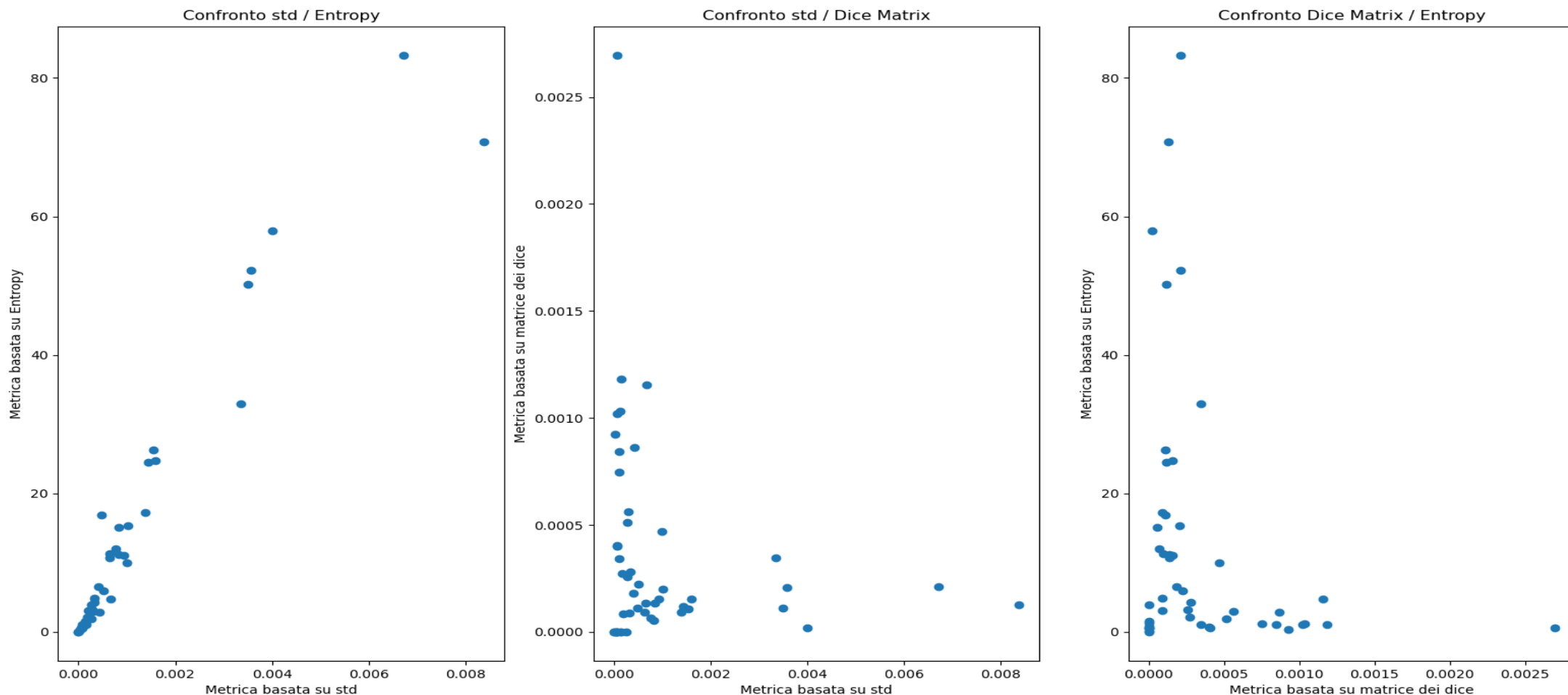




## ANALISI PRELIMINARE DELLE METRICHE

I tre scatterplot di cui sotto illustrano le relazioni tra le metriche fin qui proposte.

Sembrerebbe esserci una possibile correlazione tra std e entropia, mentre la metrica basata sulla dice matrix sembrerebbe essere meno correlata alle altre due.

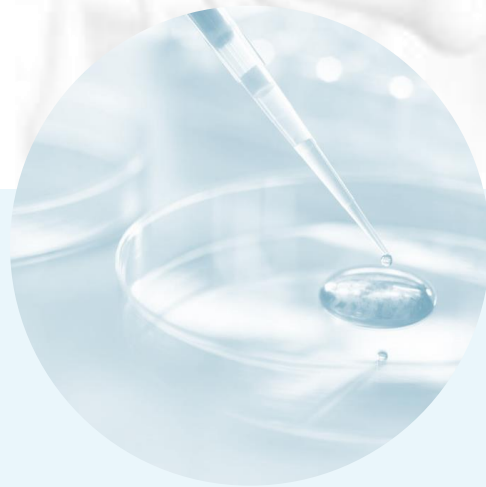


## CONCLUSIONE

Le tre metriche proposte sembrerebbero essere promettenti nell'esprimere l'incertezza con cui la rete segmenta le 50 immagini del test set di steatosi del fegato.

È da sottolineare, nuovamente, come le metriche basate su std e entropy dipendano dal ground truth (poiché le metriche sono ricavate dalla moltiplicazione delle rispettive mappe con la maschera di segmentazione manuale fornita) mentre il metodo basato sulla matrice dei dice NON dipende dal ground truth.

Una limitazione di queste tre metriche, però, è che in caso la rete sia molto poco incerta rispetto ad una segmentazione errata, i risultati potrebbero risultare falsati: se tutte e 20 le segmentazioni e le softmax derivanti dal metodo MonteCarlo fossero esattamente identiche, ma nessuna di queste segmentasse correttamente, il dice sarebbe 0 mentre le tre metriche darebbero il loro risultato idealmente sperato.



# GRAZIE PER L'ATTENZIONE

---

WILLIAM BASSOLINO