

Short report

L'esperimento consiste nel tradurre una 3D brain image registrata in una certa modalità (per esempio PD pesata) nella corrispondente immagine secondo una diversa modalità (per esempio T2 pesata) grazie ad un generator che è stato allenato appositamente tramite una conditional Deep Convolutional GAN. La traduzione tra image modalities diverse permette di acquisire una differente scansione del cervello del paziente senza doverlo sottoporre ad una ulteriore risonanza magnetica.

L'esperimento punterà in primo luogo a verificare l'efficacia della Image Gradient Difference Loss in aggiunta all'immane Adversarial Loss (1); in seconda battuta a migliorare le performance della traduzione modificando la baseline architecture della GAN (2). MAE, MSE e PSNR sono le metriche generali scelte per valutare la qualità della ricostruzione.

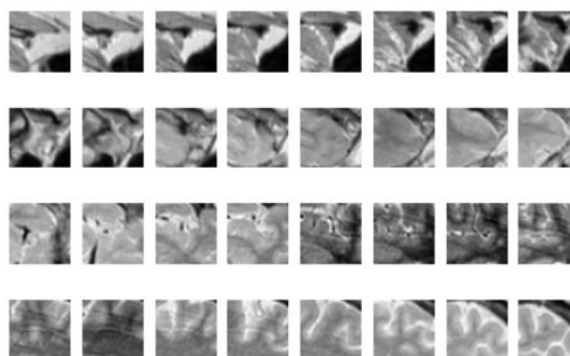
Preparazione del dataset

Sono state utilizzate le immagini di 60 soggetti nell'IXI brain dataset per il training, 3 nel validation set e 26 nel test set. Ogni singolo volume è stato processato per ottenere 128 grayscale slices trasversali di dimensione 224x224, standardizzate e salvate in una matrice. Grazie a questa rappresentazione sarà possibile estrarre dalla matrice una particolare slice del cervello nel piano orizzontale, sagittale o coronale. Nel test finale saranno prese in considerazione le sezioni di tutti e tre i piani.

GAN architecture

Il generator e il discriminator sono rispettivamente delle tipiche 3D FCN e 3D CNN (i dettagli sono rinviati al report finale). L'aspetto fondamentale è che sono patch-based: il generator per esempio prende in input una 32x32x32 3D patch e genera la 3D patch corrispondente nell'altra modalità. Questo approccio ha l'obiettivo di focalizzare la GAN sui singoli dettagli.

Il volume iniziale 128x224x224 è stato dunque partizionato in 196 cubi, il che vuol dire che per il training sono stati impiegati precisamente $60 \times 196 = 11.760$ cubi. Nel report finale illustrerò la ricostruzione del volume completo a partire dalle numerose patches prodotte nel momento del testing: in questa fase, infatti, il volume in input da tradurre non viene più partizionato ma diviso in 8.125 patches sovrapposte.



Loss Function

La loss function del discriminator è la binary cross entropy.

$$L_D(X, Y) = L_{BCE}(D(Y), 1) + L_{BCE}(D(G(X)), 0).$$

La loss function del generator è $\lambda_1 L_{ADV}(X) + \lambda_2 L_1(X, Y) + \lambda_3 L_{GDL}(Y, G(X))$, dove:

- X è la source image
- Y è la real target image
- $G(X)$ è la synthetic target image prodotta dal generator
- $D(\cdot)$ è la probabilità che l'input sia reale secondo il discriminator
- L_1 loss: $L_1(X, Y) = \|Y - G(X)\|_1$
- Adversarial loss: $L_{ADV} = L_{BCE}(D(G(X)), 1)$
- 3D Image Gradient Difference Loss:
$$L_{GDL}(Y, G(X)) = \|\nabla Y_x - \nabla G(X)_x\|^2 + \|\nabla Y_y - \nabla G(X)_y\|^2 + \|\nabla Y_z - \nabla G(X)_z\|^2$$
- Pesi dei componenti: $\lambda_1 = 0.5$, $\lambda_2 = 1$, $\lambda_3 = 1$

L'Image Gradient Difference Loss - principale contributo degli autori del paper - minimizza la somma degli scarti quadratici punto per punto tra le magnitudini dei gradienti di Y e $G(X)$ lungo gli assi x, y, z . La sua funzione è quella di preservare le regioni dell'immagine con gradienti elevati, nella pratica i bordi del cervello rispetto allo sfondo e le sue aree interne di intensità diversa. In definitiva questo terzo componente della loss function permette di ottenere immagini più nitide.

Esperimento 1: contributo dell'Image Gradient Difference Loss

Model	Best epoch on validation set	MAE on test set	PSNR on test set
Adv.	40/42	0.109 (0.024)	30.861 (2.044)
Adv. + GDL	78/80	0.105 (0.022)	31.074 (2.267)

Il modello con Adversarial Loss e Image Gradient Difference Loss (Adv. + GDL) ha richiesto il doppio delle epoch in fase di training per superare le performance del modello base (Adv.) sul validation set. Ciò è dovuto alla complessità della GDL function poiché l'ottimizzazione avviene lungo le tre dimensioni dell'immagine. Alla fine il modello allenato con loss function completa ottiene prestazioni superiori anche sul test set.

Esperimento 2.a: accelerare il training tramite il residual learning (ResGAN)

Le immagini di input PD-weighted e quelle di output T2-weighted nel dataset sono perfettamente allineate e "fortemente" correlate. Dunque una skip connection che somma la source image iniziale all'output dell'ultimo dense layer del generator potrebbe accelerare il

training. Da un punto di vista pratico questa input-output skip connection serve a ricordare al generator che l'output è "molto simile" all'input, un'informazione che viene persa man mano che si scende in profondità nella neural network. Il training sarà più rapido del modello Adv. + GDL grazie al fatto di fornire subito un'immagine di partenza: il generator imparerà a correggere epoch dopo epoch l'output del dense layer finale affinché, sommato pixel per pixel al PD-weighted input, generi un'immagine sempre più simile al T2-weighted real target.

Model	Best epoch on validation set	MAE on test set	PNSR on test set
ResGAN	40/42	0.101 (0.02)	31.229 (2.154)

Esperimento 2.b: ResGAN + Auto-Context Model

L'Auto-Context Model è una tecnica relativamente recente utilizzata nella semantic segmentation. Gli autori del paper hanno voluto applicarla (con successo) anche al loro lavoro, il quale di fatto non esegue nient'altro che un regression task. Si tratta di allenare iterativamente diverse GAN che prendono in input non solo la source patch, ma anche la synthetic target image generata dalla GAN precedente. Quest'ultima patch viene concatenata alla source patch come un second channel, mantenendo però inalterato il resto della ResGAN architecture.

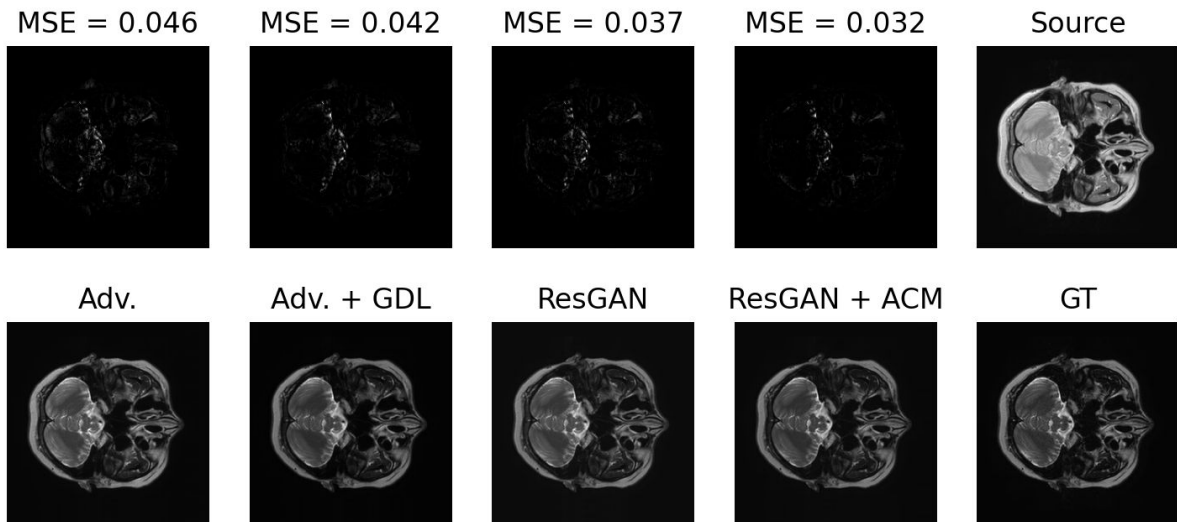
Nel mio progetto ho scelto di usare due sole ResGAN per limiti di tempo e di risorse: l'output della seconda ResGAN sarà considerato come l'output dell'intero modello.

Model	Best epoch on validation set	MAE on test set	PSNR on test set
ResGAN + ACM	40/42	0.102 (0.025)	31.36 (2.232)

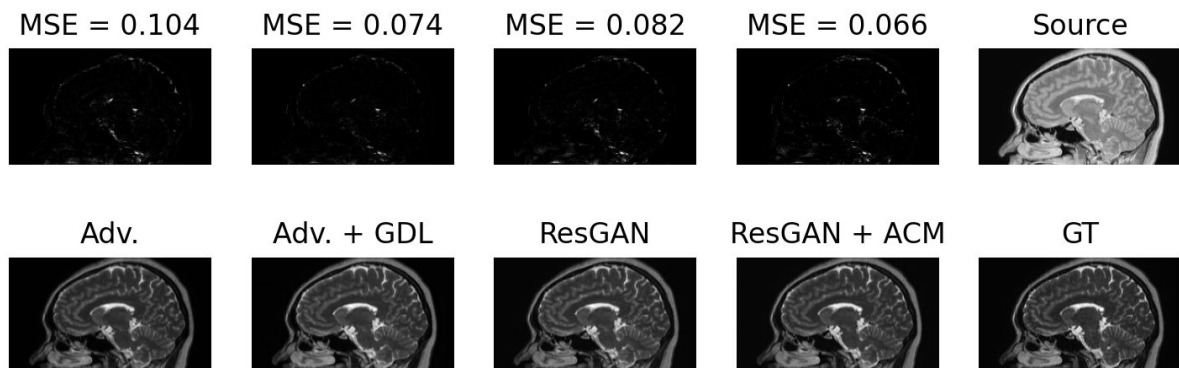
Confronto per immagini

Il MSE si riferisce alla singola slice in figura.

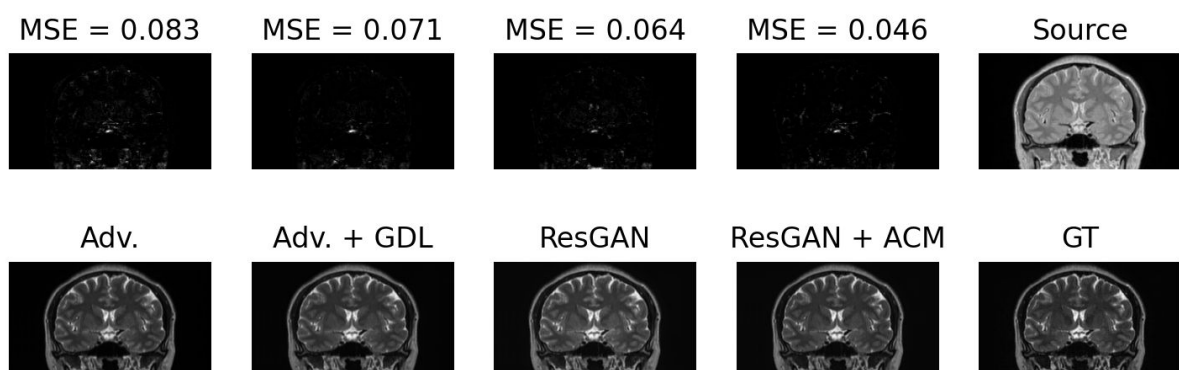
Sezione trasversale



Sezione sagittale



Sezione coronale



A questo indirizzo è possibile visionare altre immagini:

https://colab.research.google.com/drive/18id0T3ocpOfD2m5g1DVA3_sw0uODzIC8?usp=sharing

Il codice sorgente dell'intero progetto è disponibile su GitHub:

<https://github.com/brakes312t4/3D-DCGAN-medical-image-synthesis>