Contribution Title

No Author Given

No Institute Given

1 Performance analysis of context based cross entropy, Seg-GLGAN using Box plots

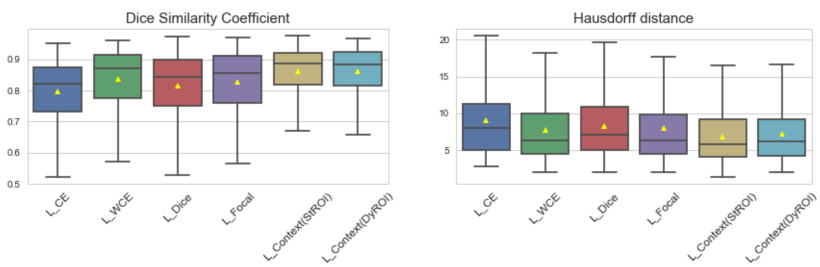

Fig. 1. Comparison of context based cross entropy loss with baseline methods for PROMISE12 dataset. Yellow: (Mean of data points), Left: Dice comparison, Right: Hausdorff comparison

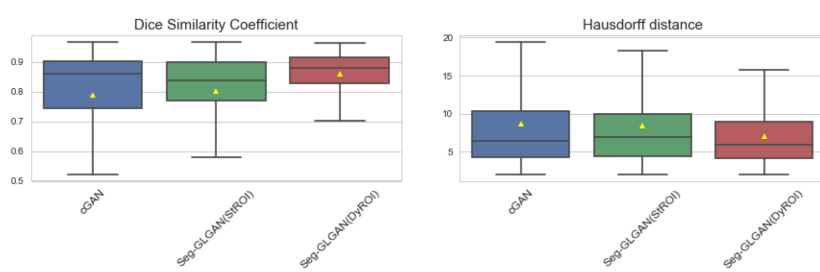

Fig. 2. Comparison of Seg-GLGAN with baseline methods for PROMISE12 dataset. Yellow: (Mean of data points), Left: Dice comparison, Right: Hausdorff comparison

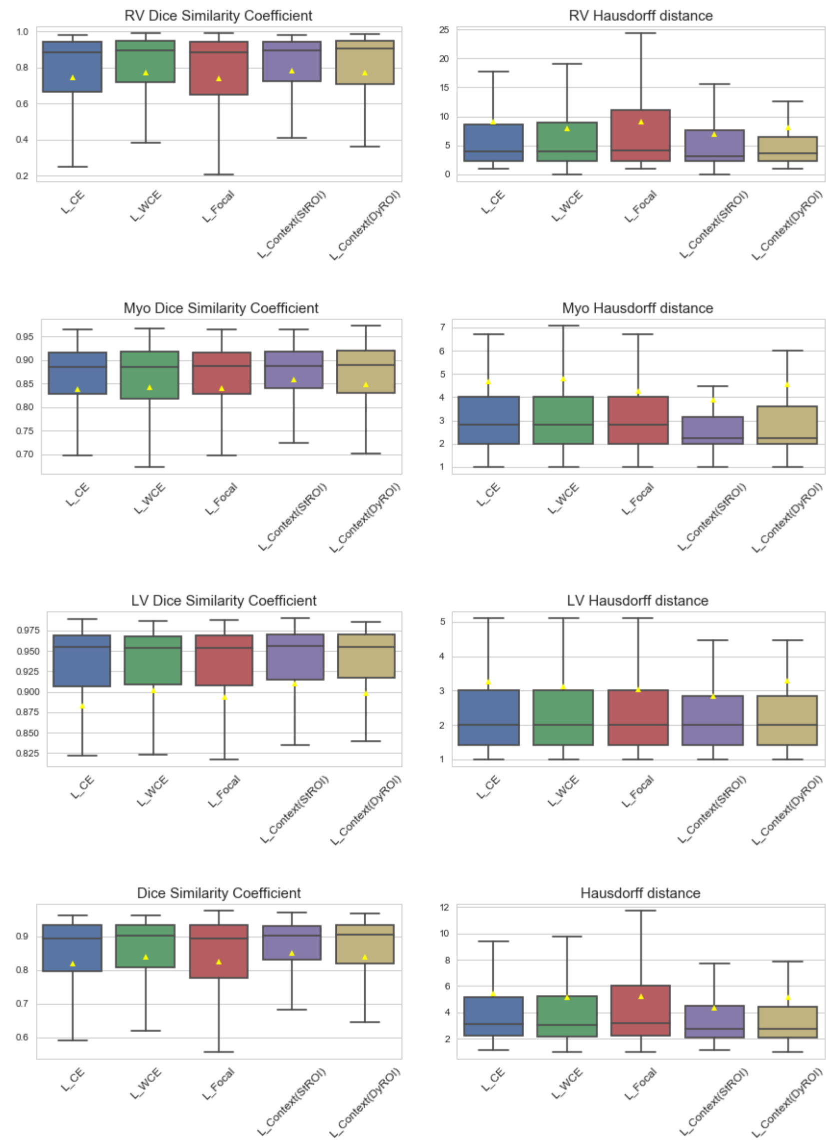

Fig. 3. Comparison of context based cross entropy with baseline methods for ACDC dataset. Yellow: (Mean of data points), Left: Dice comparison, Right: Hausdorff comparison, Top to bottom: RV, Myo, LV, Mean

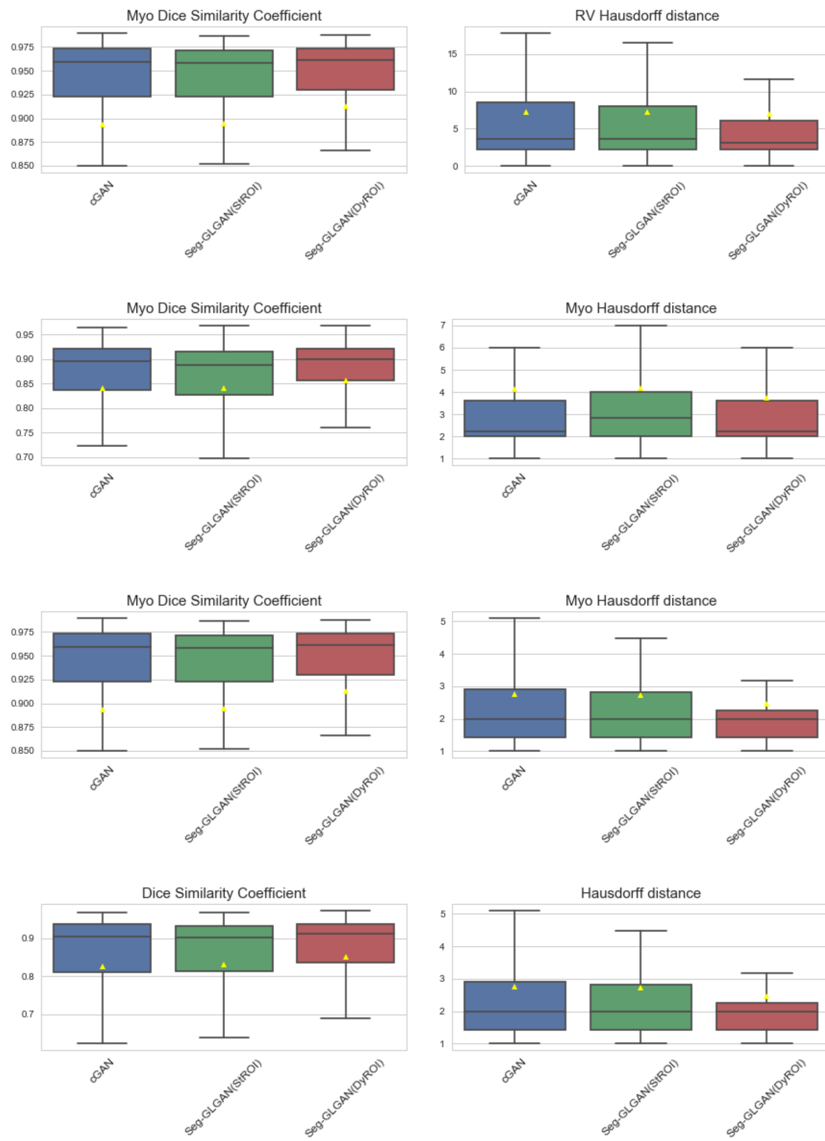

Fig. 4. Comparison of Seg-GLGAN with baseline methods for ACDC dataset. Yellow: (Mean of data points), Left: Dice comparison, Right: Hausdorff comparison, Top to bottom: RV, Myo, LV, Mean

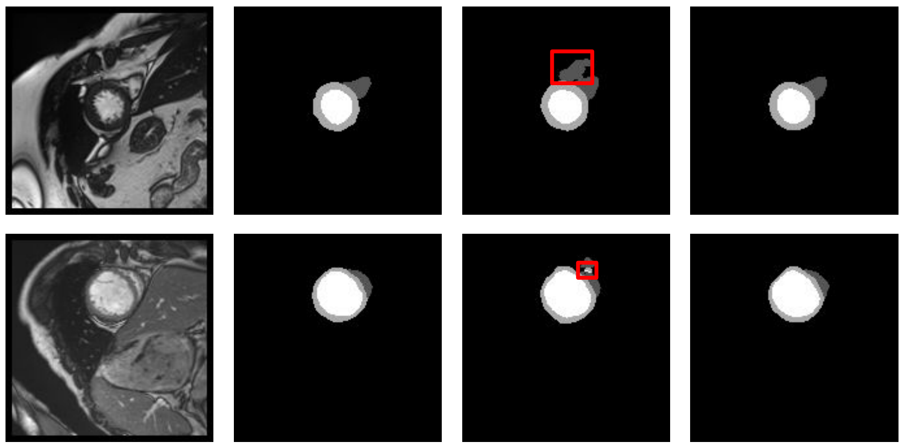

Fig. 5. Left to right: Image, Target, U-Net with CE, U-Net with context CE. (Region in red box indicate outliers)

Table 1. Comparison of GAN, Seg-GLGAN with L_{MCE} as generator loss and Seg-GLGAN with $\mathcal{L}_{context}$ as generator loss for PROMISE12. dataset

		Dice	HD
GAN	GAN	0.7877	8.76
	Seg-GLGAN(StROI)(L_{MCE})	0.8019	8.40
	Seg-GLGAN(StROI)($\mathcal{L}_{context}$)	**0.8158**	**7.93**

2 ROI extraction and training solution

2.1 U-Net: Proposed Context based Cross Entropy Loss

Static ROI A fixed ROI is used. Training can be done with higher batch sizes.

Dynamic ROI The dimensions of ROI vary during training for each image/target, so training with higher batch size will not be possible because of mismatch in dimension. But U-Net can be trained with batch size of 1 and best performance can be obtained.

2.2 Segmentation Global Local Generative Adversarial Network (Seg-GLGAN)

Static ROI A fixed ROI is used. Training can be done with higher batch sizes.

Dynamic ROI Like Dynamic ROI case in U-Net this can't be trained because training the GAN with batch size of 1 is not an ideal option as GAN will have difficulty in becoming stable affecting the convergence. To overcome this problem in GAN, a higher batch size should be used. We approached this problem in the following way: We decided the bounding box dimensions beforehand and created n bins, each bin is assigned with a fixed bounding box dimension. Say in our case, 8 bins $(20 \times 20, 30 \times 30, ...90 \times 90)$. The bounding box of object in target mask is obtained and the image/target pair is assigned to bin with closest largest bounding box dimension. Now, if we fix the batch size as 8, it is not necessary that the bins will have image/target pair count to be a multiple of 8. So, we augment the image/target pair in that bin to make the count to nearest multiple of 8. Finally, all the bins will have a a multiple of 8 image/target pair. Now we have everything set to train the GAN properly.