Unsupervised Domain Adaptation for Semantic Segmentation of NIR Images through Generative Latent Search −Supplementary−

Prashant Pandey*[0000−0002−6594−9685], Aayush Kumar Tyagi*[0000−0002−3615−7283], Sameer Ambekar[0000−0002−8650−3180], and Prathosh AP[0000−0002−8699−5760]

Indian Institute of Technology Delhi
getprashant57@gmail.com, aayush16081@iiitd.ac.in, ambekarsameer@gmail.com, prathoshap@iitd.ac.in

1 Datasets

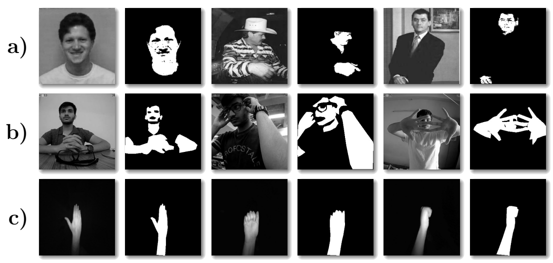


Fig. 1: a) shows samples of COMPAQ dataset [3] images with only Red-channel present b) contains samples from SNV dataset c) contains samples from Hand Gesture dataset.

Each row of Fig. 1 shows few images with the corresponding skin-mask pairs from COMPAQ, SNV and Hand Gesture datasets respectively.

2 Implementation details

2.1 GLSS on NIR images

$\mathcal{S}_\psi$ is the segmentation model (as shown in Fig. 2 in the paper) implemented using DeepLabv3+ (XceptionNet) and UNet (EfficientNet). $\mathcal{S}_\psi$ is trained for

* equal contribution

Algorithm 1 **Generative Latent Search for Segmentation (GLSS)**

```
    Training VAE on source samples
    Input: Source dataset $\mathcal{S}_n = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, Number of source samples $n$, Encoder $g_\phi$, Decoder $h_\theta$, Trained Perceptual Model $P_\psi$, Learning rate $\eta$, Batch-size $B$. Output: Optimal parameters $\phi^*$, $\theta^*$.
 1: Initialize parameters $\phi$, $\theta$
 2: repeat
 3:     sample batch $\{\mathbf{x}_i\}$ from dataset $\mathcal{S}_n$, for $i = 1, ..., B$
 4:     $\mu_{\mathbf{z}}^{(i)}, \sigma_{\mathbf{z}}^{(i)} \leftarrow g_\phi(\mathbf{x}_i)$
 5:     sample $\mathbf{z}_i \sim \mathcal{N}(\mu_{\mathbf{z}}^{(i)}, {\sigma_{\mathbf{z}}^{(i)}}^2)$
 6:     $\mathcal{L}_r \leftarrow \sum_{i=1}^{B} \|\mathbf{x}_i - h_\theta(\mathbf{z}_i)\|_2^2$
 7:     $\mathcal{L}_p \leftarrow \sum_{i=1}^{B} \|P_\psi(\mathbf{x}_i) - P_\psi(h_\theta(\mathbf{z}_i))\|_2^2$
 8:     $\mathcal{L}_g \leftarrow \mathcal{L}_r + \mathcal{L}_p + \sum_{i=1}^{B} \mathbb{D}_{KL}\left[\mathcal{N}(\mu_{\mathbf{z}}^{(i)}, {\sigma_{\mathbf{z}}^{(i)}}^2) \,||\, \mathcal{N}(0,1)\right]$
 9:     $\mathcal{L}_h \leftarrow \mathcal{L}_r + \mathcal{L}_p$
10:     $\phi \leftarrow \phi + \eta \nabla_\phi \mathcal{L}_g$
11:     $\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{L}_h$
12: until convergence of $\phi$, $\theta$

    Inference - Latent Search during testing with Target
    Input: Target sample $\tilde{\mathbf{x}}_{\mathcal{T}}$, Trained decoder $h_{\theta^*}$, Learning rate $\eta$. Output: 'nearest-clone' $\tilde{\mathbf{x}}_{\mathcal{S}}$ for the target sample $\tilde{\mathbf{x}}_{\mathcal{T}}$.
13: sample $\mathbf{z}$ from $\mathcal{N}(0,1)$
14: repeat
15:     $\mathcal{L}_{ssim} \leftarrow 1 - \text{SSIM}(\tilde{\mathrm{x}}_{\mathcal{T}}, h_{\theta^*}(\mathbf{z}))$
16:     $\mathbf{z} \leftarrow \mathbf{z} + \eta \nabla_{\mathbf{z}} \mathcal{L}_{ssim}$
17: until convergence of $\mathcal{L}_{ssim}$
18: $\tilde{\mathbf{z}}_{\mathcal{S}} \leftarrow \mathbf{z}$
19: $\tilde{\mathbf{x}}_{\mathcal{S}} \leftarrow h_{\theta^*}(\tilde{\mathbf{z}}_{\mathcal{S}})$
```

100-150 epochs with losses ($\mathcal{L}_s$) as shown in Eq. 1 and Eq. 2 for UNet and DeepLabv3+ respectively.

$$\mathcal{L}_s = \mathcal{L}_{dice} + \mathcal{L}_{bce} \tag{1}$$

$$\mathcal{L}_s = \mathcal{L}_{focal} \tag{2}$$

$\mathcal{L}_{dice}$ is the dice coefficient loss which calculates the overlap between the predicted and the ground truth mask whereas $\mathcal{L}_{bce}$ is the binary cross-entropy loss. Binary focal loss ($\mathcal{L}_{focal}$) tries to down-weight the contribution of examples that can be easily segmented so that the segmentation model focuses more on learning hard examples.

P_ψ is a perceptual model (as shown in Fig. 1 in the paper) that uses perceptual loss $\mathcal{L}_p$. The perceptual features are taken from the 6th layer of UNet and the last concatenation layer of DeepLabv3+. VAE along with perceptual loss $\mathcal{L}_p$ is trained for 150-200 epochs. $\mathcal{L}_p$ is weighted with a factor β (a hyper-parameter) as shown:

$$\mathcal{L}_{total} = \mathcal{L}_{vae} + \beta\mathcal{L}_p \tag{3}$$

In order to improve the quality of VAE reconstructed images, we weighted the perceptual loss ($\mathcal{L}_p$) with different values of β. For UNet, we have used $\beta = 2$ whereas $\beta = 3$ is used for DeepLabv3+. The first part of Algorithm 1 shows the steps involved in training VAE and second part shows the steps involved in inference procedure.

Using an Intel Xeon processor (6 Cores) with a base frequency of 2.0 GHz, 32GB RAM and NVIDIA® Tesla® K40 (12 GB Memory) GPU, Latent Search for one sample on SNV dataset takes 450 ms and 120 ms on Hand Gesture dataset. The time required is in the order of milliseconds on a basic GPU like K40 which is not very significant. However, this is the cost that is to be paid for being target independent which is a very significant advantage.

2.2 Implementation details of UDA baseline methods for skin segmentation

DeepLabv3+ was used as the segmentation model for all the baselines with images and corresponding masks of size 128×128. AdaptsegNet [6] uses discriminative approach to predict the domain of the images. For discriminator, we used a model with 5 convolutional layers (default implementation). We performed a grid search over $\lambda_{advtarget1}$ and $\lambda_{advtarget2}$ and reported the best IoU score for AdaptsegNet. DISE [2] uses image-to-image translation approach to translate one domain to another. It employs label transfer loss to optimize the segmentation model. Image-to-image translation based methods work well in cases where the structural similarity is more. We used 0.1, 0.25 and 0.5 for λ_{seg} and reported the best IoU using $\lambda_{seg} = 0.1$ while the learning rate was set to 2.5e-4. Advent [7] proposes to leverage an entropy loss to directly penalize low-confident predictions on target domain. If λ_{ent} is large then the entropy drops too quickly and the model is strongly biased towards a few classes. We used 0.001 for λ_{ent} as suggested by the authors regardless of the network and dataset. Also, for adversarial training, 0.001 was used for λ_{adv}. We trained with AdvEnt as it performed better that minEnt as stated in the paper. SGD and Adam were used as optimizers for segmentation and discriminator networks respectively. In DADA [8], authors make use of an additional depth information in the source domain. We performed a grid search over λ_{seg} using values 0.25, 0.5, 1. The learning rate was varied with values 2.5e-4, 1e-4 and 3e-4 and finally best IoU was reported with $\lambda_{seg} = 0.5$ and learning rate = 2.5e-4. CLAN [5] makes use of a category-level joint distribution and align each class with an adaptive adversarial loss, thus ensuring correct mapping of source and target. Compared to traditional adversarial training, CLAN introduces the discrepancy loss and the category-level adversarial loss. Hyperparameters like learning rate, weight decay, λ_{weight} and λ_{adv} were used with values 2.5e-4, 5e-4, 0.01 and 0.001 respectively during training. For training BDL [4], we set the learning rate to 2.5e-4 for the segmentation network and 1e-4 for the discriminator. Grid search was performed for $\lambda_{advtarget}$ with values 1e-3, 2e-3, 5e-3 and best IoU was reported with $\lambda_{advtarget} =$ 1e-3.

2.3 SSIM Loss

SSIM loss compares pixels and their corresponding neighbourhoods between two images, preserving the luminance, contrast and structural information. To perform Latent Search, we used distance metric as SSIM loss, that helps to sample the 'nearest-clone' in the source distribution for the target image from the generative latent space of VAE. Unlike norm-based losses, SSIM loss helps in the preservation of structural information as compared to discrete pixel-level information. We used 11x11 Gaussian filter in our experiments.

SSIM is defined using the three aspects of similarities, luminance $\big(l(\mathbf{x}, \hat{\mathbf{x}})\big)$, contrast $\big(c(\mathbf{x}, \hat{\mathbf{x}})\big)$ and structure $\big(s(\mathbf{x}, \hat{\mathbf{x}})\big)$ that are measured for a pair of images $\{\mathbf{x}, \hat{\mathbf{x}}\}$ as follows:

$$l(\mathbf{x}, \hat{\mathbf{x}}) = \frac{2\mu_{\mathbf{x}}\mu_{\hat{\mathbf{x}}} + C_1}{\mu_{\mathbf{x}}^2 + \mu_{\hat{\mathbf{x}}}^2 + C_1} \tag{4}$$

$$c(\mathbf{x}, \hat{\mathbf{x}}) = \frac{2\sigma_{\mathbf{x}}\sigma_{\hat{\mathbf{x}}} + C_2}{{\sigma_{\mathbf{x}}}^2 + {\sigma_{\hat{\mathbf{x}}}}^2 + C_2} \tag{5}$$

$$s(\mathbf{x}, \hat{\mathbf{x}}) = \frac{\sigma_{\mathbf{x}\hat{\mathbf{x}}} + C_3}{\sigma_{\mathbf{x}}\sigma_{\hat{\mathbf{x}}} + C_3} \tag{6}$$

where μ's denote sample means and σ's denote variances. C_1, C_2 and C_3 are constants. With these, SSIM and the corresponding loss function $\mathcal{L}_{ssim}$, for a pair of images $\{\mathbf{x}, \hat{\mathbf{x}}\}$ are defined as:

$$\text{SSIM}(\mathbf{x}, \hat{\mathbf{x}}) = l(\mathbf{x}, \hat{\mathbf{x}})^{\alpha} \cdot c(\mathbf{x}, \hat{\mathbf{x}})^{\beta} \cdot s(\mathbf{x}, \hat{\mathbf{x}})^{\gamma} \tag{7}$$

where $\alpha > 0$, $\beta > 0$ and $\gamma > 0$ are parameters used to adjust the relative importance of the three components.

$$\mathcal{L}_{ssim}(\mathbf{x}, \hat{\mathbf{x}}) = 1 - \text{SSIM}(\mathbf{x}, \hat{\mathbf{x}}) \tag{8}$$

2.4 Target-Independence of GLSS

GLSS is a general-purpose target-independent UDA method. For UDA, target independence is a merit since a SINGLE source model can be used across multiple targets. However, even with target data (for VAE training) GLSS doesn't degrade while SOTA methods do, for skin segmentation on NIR images (Table below compares IoU).

Table 1: IoU comparison for Target-Independence of GLSS with change in the amount of target data. GLSS performance is not affected by change in the amount of target data during training while other SOTA methods degrade.

% of Target data	Adaptsegnet	BDL	CLAN	Advent	DADA	GLSS
60	0.23	0.30	0.22	0.33	0.31	0.37
40	0.22	0.26	0.22	0.29	0.28	0.37
20	0.21	0.22	0.21	0.24	0.23	0.38

2.5 GAN vs. VAE

GLSS demands a generative model that has both generation and inference capabilities (mapping from latent to data space and vice versa), which is not the case with GANs. This leads to non-convergence of latent search. To validate this, we trained a SOTA BigGAN [1] on COMPAQ Dataset [21] and performed GLSS. Although GAN had better generation quality (FID of 29.7 with BigGAN vs. 44 with VAE), the final IoU was worse as shown in Table 2.

Table 2: IoU score comparison between BigGAN and VAE when trained on SNV and Hand Gesture datasets. VAE scores better in terms of IoU.

SNV/BigGAN	Hand Gesture/BigGAN	SNV/VAE	Hand Gesture/VAE
0.09	0.21	0.38	0.69

3 Additional Results

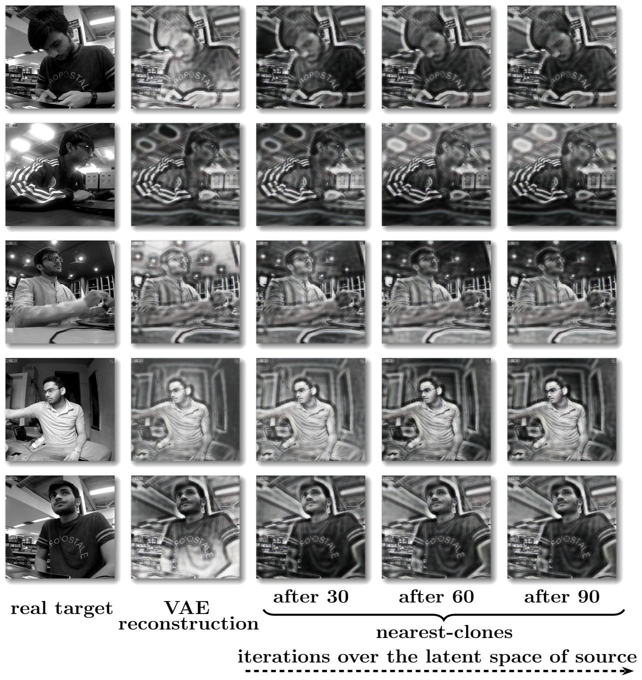


Fig. 2: Illustration of Latent Search (LS) in GLSS for SNV dataset. Prior to the LS, VAE reconstructed target samples are obtained. It is evident that the 'nearest-clones' (images generated using LS) improve as the LS progresses. Also the quality (empirically) of 'nearest-clones' are better as compared to the VAE reconstructed images. The 'nearest-clones' are shown after every 30 iterations.

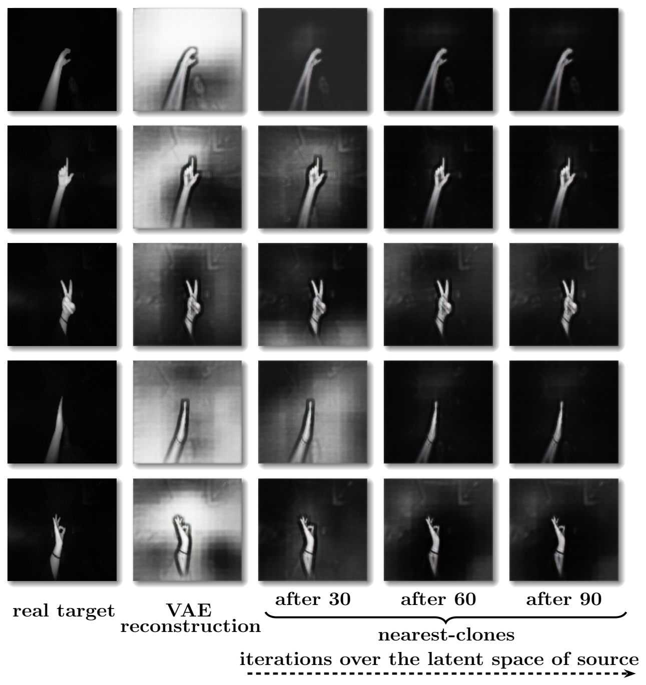


Fig. 3: Illustration of Latent Search (LS) in GLSS for Hand Gesture dataset. Prior to the LS, VAE reconstructed target samples are obtained. It is evident that the 'nearest-clones' (images generated using LS) improve as the LS progresses. Also the quality (empirically) of 'nearest-clones' are better as compared to the VAE reconstructed images. The 'nearest-clones' are shown after every 30 iterations.

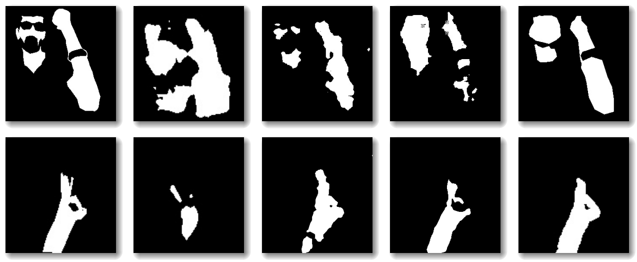
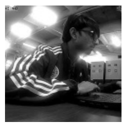
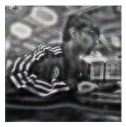


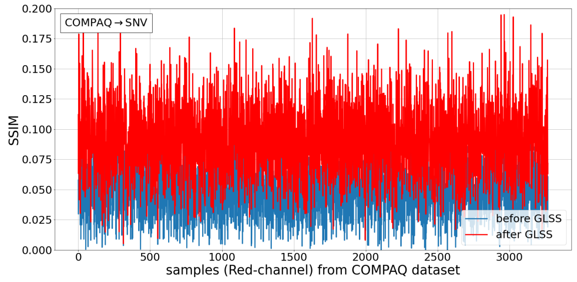
Fig. 4: (a) the ground truth mask for SNV and Hand Gesture datasets, (b) the predicted mask of VAE reconstructed image without edge concatenation, (c) the predicted mask of VAE reconstructed image without $\mathcal{L}_p$, (d) the predicted mask of VAE reconstructed with edge concatenation and perceptual loss when no Latent Search (LS) was performed, (e) the predicted mask with GLSS. It is evident from the predicted masks that with edge concatenation, perceptual loss and Latent Search (LS), quality of predicted masks improve. Each component plays a significant role in improving the IoU. Hence, when all the components are employed (as in GLSS) we get the best IoU.



(a)

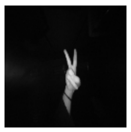


(b)

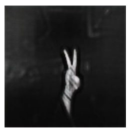


(c)

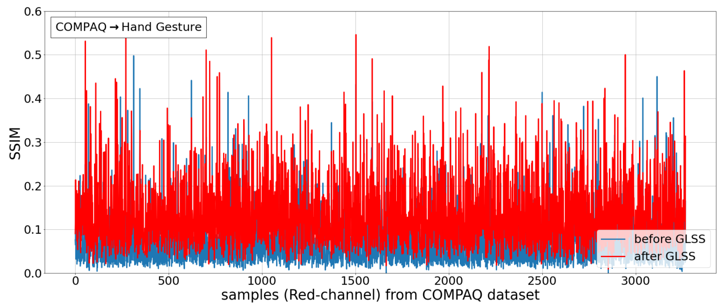
Fig. 5: (a) an NIR image $\tilde{\mathbf{x}}_{\mathcal{T}}$ from SNV dataset (target), (b) 'nearest-clone' $\tilde{\mathbf{x}}_{\mathcal{S}}$ generated from GLSS, (c) Structural Similarity Index (SSIM) scores calculated between $\tilde{\mathbf{x}}_{\mathcal{T}}$ and all the samples (having only Red-channel) of COMPAQ dataset (source) are shown with blue color in the plot. Similary, SSIM scores calculated between $\tilde{\mathbf{x}}_{\mathcal{S}}$ and all the samples (having only Red-channel) of COMPAQ dataset are shown with red color. It is evident from the figure that the SSIM scores are higher for the 'nearest-clone' $\tilde{\mathbf{x}}_{\mathcal{S}}$ as compared to the scores with $\tilde{\mathbf{x}}_{\mathcal{T}}$. It indicates that $\tilde{\mathbf{x}}_{\mathcal{S}}$ is more closer to the source domain (COMPAQ) as compared to $\tilde{\mathbf{x}}_{\mathcal{T}}$. Hence, the 'nearest-clone' $\tilde{\mathbf{x}}_{\mathcal{S}}$ generated by GLSS for target $\tilde{\mathbf{x}}_{\mathcal{T}}$ is used as a proxy in the segmentation network S_{ψ} which is trained only on COMPAQ dataset, thereby increasing the IoU for $\tilde{\mathbf{x}}_{\mathcal{T}}$.



(a)



(b)



(c)

Fig. 6: (a) an NIR image $\tilde{\mathbf{x}}_{\mathcal{T}}$ from Hand Gesture dataset (target), (b) 'nearest-clone' $\tilde{\mathbf{x}}_{\mathcal{S}}$ generated from GLSS, (c) Structural Similarity Index (SSIM) scores calculated between $\tilde{\mathbf{x}}_{\mathcal{T}}$ and all the samples (having only Red-channel) of COMPAQ dataset (source) are shown with blue color in the plot. Similary, SSIM scores calculated between $\tilde{\mathbf{x}}_{\mathcal{S}}$ and all the samples (having only Red-channel) of COMPAQ dataset are shown with red color. It is evident from the figure that the SSIM scores are higher for the 'nearest-clone' $\tilde{\mathbf{x}}_{\mathcal{S}}$ as compared to the scores with $\tilde{\mathbf{x}}_{\mathcal{T}}$. It indicates that $\tilde{\mathbf{x}}_{\mathcal{S}}$ is more closer to the source domain (COMPAQ) as compared to $\tilde{\mathbf{x}}_{\mathcal{T}}$. Hence, the 'nearest-clone' $\tilde{\mathbf{x}}_{\mathcal{S}}$ generated by GLSS for target $\tilde{\mathbf{x}}_{\mathcal{T}}$ is used as a proxy in the segmentation network S_{ψ} which is trained only on COMPAQ dataset, thereby increasing the IoU for $\tilde{\mathbf{x}}_{\mathcal{T}}$.

References

1. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018)
2. Chang, W.L., Wang, H.P., Peng, W.H., Chiu, W.C.: All about structure: Adapting structural information across domains for boosting semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1900–1909 (2019)
3. Jones, M.J., Rehg, J.M.: Statistical color models with application to skin detection. International Journal of Computer Vision **46**(1), 81–96 (2002)
4. Li, Y., Yuan, L., Vasconcelos, N.: Bidirectional learning for domain adaptation of semantic segmentation. arXiv preprint arXiv:1904.10620 (2019)
5. Luo, Y., Zheng, L., Guan, T., Yu, J., Yang, Y.: Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2507–2516 (2019)
6. Tsai, Y.H., Hung, W.C., Schulter, S., Sohn, K., Yang, M.H., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7472–7481 (2018)
7. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: CVPR (2019)
8. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Dada: Depth-aware domain adaptation in semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7364–7373 (2019)