Skin Segmentation from NIR Images using Unsupervised Domain Adaptation through Generative Latent Search

Prashant Pandey*, Aayush Kumar Tyagi*, Sameer Ambekar, and Prathosh AP

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

Abstract. Segmentation of the pixels corresponding to human skin is an essential first step in multiple applications ranging from surveillance to heart-rate estimation from remote-photoplethysmography. However, the existing literature considers the problem only in the visible-range of the EM-spectrum which limits their utility in low or no light settings where the criticality of the application is higher. To alleviate this problem, we consider the problem of skin segmentation from the Near-infrared images. However, Deep learning based state-of-the-art segmentation techniques demands large amounts of labelled data that is unavailable for the current problem. Therefore we cast the skin segmentation problem as that of target-independent unsupervised domain adaptation (UDA) where we use the data from the Red-channel of the visible-range to develop skin segmentation algorithm on NIR images. We propose a method for target-independent segmentation where the ‘nearest-clone’ of a target image in the source domain is searched and used as a proxy in the segmentation network trained only on the source domain. We prove the existence of ‘nearest-clone’ and propose a method to find it through an optimization algorithm over the latent space of a Deep generative model based on variational inference. We demonstrate the efficacy of the proposed method for NIR skin segmentation over the state-of-the-art UDA segmentation methods on the two newly created skin segmentation datasets in NIR domain despite not having access to the target NIR data.

Keywords: Unsupervised Domain Adaptation, Skin segmentation, Near IR Dataset, VAE

1 Introduction

1.1 Background

Human skin segmentation is the task of finding pixels corresponding to skin from images or videos. It serves as a necessary pre-processing step for multiple applications like video surveillance, people tracking, human computer interaction,
face detection and recognition, facial gesture detection and monitoring heart rate and respiratory rate\textsuperscript{[7,31,32,35]} using remote photoplethysmography. Most of the research efforts on skin detection have focused on visible spectrum images because of the challenges that it poses including, illumination change, ethnicity change and presence of background/clothes similar to skin colour. These factors adversely affect the applications where skin is used as conjugate information. Further, the algorithms that rely on visible spectrum images cannot be employed in the low/no light conditions especially during night times where the criticality of the application like human detection is higher. These problems which are encountered in visible spectrum domain can be overcome by considering the images taken in the Near-infrared (NIR) domain\textsuperscript{23} or hyper spectral imaging\textsuperscript{34}. The information about the skin pixels is invariant of factors such as illumination conditions, ethnicity etc., in these domains. Moreover, most of the surveillance cameras that are used world-wide are NIR imaging devices. Thus, it is meaningful to pursue the endeavour of detecting the skin pixels from the NIR images.

1.2 Problem setting and contributions

The task of detection of skin pixels from an image is typically cast as a segmentation problem. Most of the classical approaches relied on the fact that the skin-pixels have a distinctive color pattern\textsuperscript{[12,18]} compared to other objects. In recent years, harnessing the power of Deep learning, skin segmentation problem has been dealt with using Deep-Neural networks that show significant performance enhancement over the traditional methods\textsuperscript{[20,29,37]}, albeit generalization across different illuminations still remain a challenge. While there exists sufficient literature on skin segmentation in the visible-spectrum, there is very little work done on segmenting the skin pixels in the NIR domain. Further, all the state-of-the-art Deep learning based segmentation algorithms demand large-scale annotated datasets to achieve good performance which is available in the case of visible-spectrum images but not the NIR images. Thus, building a fully-supervised skin segmentation network from scratch is not feasible for the NIR images because of the unavailability of the large-scale annotated data. However, the underlying concept of ‘skin-pixels’ is the same across the images irrespective of the band in which they were captured. Additionally, the NIR and the Red-channel of the visible-spectrum are close in terms of their wavelengths. Owing to these observations, we pose the following question in this paper - Can the labelled data in the visible-spectrum (Red-channel) be used to perform skin segmentation in the NIR domain?

We cast the problem of skin segmentation from NIR images as a target-independent unsupervised domain adaptation (UDA) task where we consider the the Red-channel of the visible-spectrum images as the source domain and NIR images as the target domain. The state-of-the-art UDA techniques demand access to the target data, albeit unlabelled, to adapt the source domain features to the target domain. In the present case, we do not assume existence of any
data from the target domain, even unlabelled. This is an important desired attribute which ensures that a model trained on the Red-channel does not need any retraining with the data from NIR domain. The core idea is to sample the ‘nearest-clone’ in the source domain to a given test image from the target domain. This is accomplished through a simultaneous sampling-cum-optimization procedure using a latent-variable Deep-neural generative network learned on the source distribution. Thus, given a target sample, its ‘nearest-clone’ from the source domain is sampled and used as a proxy in the segmentation network trained only on the samples of the source domain. Since the segmentation network performs well on the source domain, it is expected to give the correct segmentation mask on the ‘nearest-clone’ which is then assigned to the target image. Specifically, the core contributions of this work is listed as follows:

1. We cast the problem of skin segmentation from NIR images as a UDA segmentation task where we use the data from the Red-channel of the visible-range of the EM-spectrum to develop skin segmentation algorithm on NIR images.
2. We propose a method for target-independent segmentation where the ‘nearest-clone’ of a target image in the source domain is searched and used as a proxy in the segmentation network trained only on the source domain.
3. We theoretically prove the existence of the ‘nearest-clone’ given that it can be sampled from the source domain with infinite data points.
4. We develop a joint-sampling and optimization algorithm using variational inference generative model to search for the ‘nearest-clone’ through implicit sampling in the source domain.
5. We demonstrate the efficacy of the proposed method for NIR skin segmentation over the state-of-the-art UDA segmentation methods on the two newly created skin segmentation datasets in NIR domain.

2 Related Work

In this section, we first review the existing methods for skin segmentation in the visible-range followed by a review of UDA methods for segmentation.

2.1 Skin Segmentation in Visible-range

Methods for skin segmentation can be grouped into three categories, i.e. (i) Thresholding based methods [12, 24, 36], (ii) Traditional machine learning techniques to learn a skin color model [28, 47], (iii) Deep learning based method to learn an end-to-end model for skin segmentation [2, 7, 14, 38, 46]. The thresholding methods focus on defining a specified range in different color representation spaces (HSV) [33], Orthogonal color space (YCbCr) [3, 17] that to be differentiate skin pixel from others. Traditional machine learning can be further divided into pixel based and region based methods. In pixel based methods, each pixel is classified as skin or non-skin without considering the neighbours [41] whereas
region based approaches use spatial information to identify similar regions [8].
In recent years, Fully convolutional neural networks (FCN) are used to solve
the problem [29], [37] proposed a U-Net architecture, consisting of an encoder-
decoder structure with backbones like InceptionNet [39] and ResNet [13]. Holistic
skin segmentation [11] combine inductive transfer learning and UDA. They term
this technique as cross domain pseudo-labelling and use it in an iterative manner
to train and fine tune the model on the target domain. [14] propose mutual guid-
ance to improve skin detection with the usage of body masks as guidance. They
use dual task neural network for joint detection with shared encoder and two
decoders for detecting skin and body simultaneously. While all these methods
offer different advantages, they do not generalize to low-light settings with NIR
images, which we aim to solve through UDA.

2.2 Domain Adaptation for semantic segmentation

Unsupervised domain adaptation aims to improve the performance of deep neu-
ral networks on a target domain, using labels only from a source domain. UDA
for segmentation task can be grouped into following categories.

Adversarial training based methods: These methods use the principles of adver-
sarial learning [15], which generally consist of two networks. One predict-
ing the segmentation mask of input image coming from either source or target
distribution while other network acts as discriminator which tries to predict the
domain of the images. AdaptSegNet [42] exploits structural similarity between
the source and target domains in a multi-level adversarial network framework.
ADVENT [43] introduce entropy-based loss to directly penalize low-confident
predictions on target domain. Adversarial training is used for structure adapta-
tion of the target domain to the source domain. CLAN [30] considers category-
level joint distribution and align each class with an adaptive adversarial loss.
They reduce the weight of the adversarial loss for category-level aligned features
while increasing the adversarial force for those that are poorly aligned. DADA
[44] uses the geometry of the scene by simultaneously aligning the segmentation
and depth-based information of source and target domains using adversarial
training.

Feature-transformation based methods: These methods are based on the
idea of learning image-level or feature-level transformations between the source
and the target domains. CyCADA [1] adapts between domains using both gen-
erative image space alignment and latent representation space alignment. Image
level adaptation is achieved with cycle loss, semantic consistency loss and pixel-
level GAN loss while feature level adaptation employs feature-level GAN loss
and task loss between true and predicted labels. DISE [4] aims to discover a
domain-invariant structure feature by learning to disentangle domain-invariant
structure information of an image from its domain-specific texture information.
BDL [25] involves two separated modules a) image-to-image translation model
b) segmentation adaptation model, in two directions namely “translation-to-segmentation” and “segmentation-to-translation”.

3 Proposed method

Most of the UDA methods assume access to the unlabelled target data which may not be available at all times. In this work, we propose a UDA segmentation technique by learning to find a data point from the source that is arbitrarily close (called the ‘nearest-clone’) to a given target point so that it can used as a proxy in the segmentation network trained only on the source data. In the subsequent sections, we describe the methodology used to find the ‘nearest-clone’ from the source distribution to a given target point.

3.1 Existence of nearest source point

To start with, we show that for a given target data point, there exists a corresponding source data point, that is arbitrarily close to, provided that infinite data points can be sampled from the source distribution. Mathematically, let $P_s(x)$ denote the source distribution and $P_t(x)$ denote any target distribution that is similar but not exactly same as $P_s$ (in the current case, Red-channel and NIR images). Let the underlying random variable on which $P_s$ and $P_t$ are defined form a separable metric space $\{\mathcal{X}, \mathcal{D}\}$ with $\mathcal{D}$ being some distance metric. Let $S_n = \{x_1, x_2, x_3, \ldots, x_n\}$ be i.i.d points drawn from $P_s(x)$ and $\tilde{x}$ be a point from $P_t(x)$. With this, the following lemma shows the existence of the ‘nearest-clone’.

**Lemma 1.** If $\tilde{x}_S \in S_n$ is the point such that $\mathcal{D}\{\tilde{x}, \tilde{x}_S\} < \mathcal{D}\{\tilde{x}, x\}$ $\forall x \in S_n$, then $\lim_{n\to\infty} \tilde{x}_S$ converges to $\tilde{x}$ with probability 1.

**Proof.** Let $B_r(\tilde{x}) = \{x : \mathcal{D}\{\tilde{x}, x\} \leq r\}$ be a closed ball of radius $r$ around $\tilde{x}$ under the metric $\mathcal{D}$. Since $\mathcal{X}$ is a separable metric space [10],

$$\text{Prob}(B_r(\tilde{x})) \triangleq \int_{B_r(\tilde{x})} P_s(x) \, dx > 0, \forall r > 0, \quad (1)$$

With this, for any $\delta > 0$, the probability that none of the points in $S_n$ are within $B_\delta(\tilde{x})$ of radius $\delta$ is:

$$\text{Prob}\left[\min_{i=1,2,\ldots,n} \mathcal{D}\{x_i, \tilde{x}\} \geq \delta\right] = \left[1 - \text{Prob}(B_\delta(\tilde{x}))\right]^n \quad (2)$$

Therefore, the probability of $\tilde{x}_S$ (the closest point to $\tilde{x}$) lying within $B_\delta(\tilde{x})$ is:

$$\text{Prob}\left[\tilde{x}_S \in B_\delta(\tilde{x})\right] = 1 - \left[1 - \text{Prob}(B_\delta(\tilde{x}))\right]^n \quad (3)$$

$$= 1 \; \text{as} \; n \to \infty \quad (4)$$

Thus, given any infinitesimal $\delta > 0$, with probability 1, $\exists \tilde{x}_S \in S_n$ (‘nearest-clone’) that is within $\delta$ distance from $\tilde{x}$ as $n \to \infty$. \qed
Fig. 1: Variational Auto-Encoder training. Edges of an input image are concatenated with the features from the decoder \( h_\theta \). Encoder and decoder parameters \( \phi, \theta \) are optimized with reconstruction loss \( \mathcal{L}_r \), KL-divergence loss \( D_{KL} \) and perceptual loss \( \mathcal{L}_p \). Perceptual model \( P_\psi \) is trained on source samples. A zero mean and unit variance isotropic Gaussian prior is imposed on the latent space \( z \).

While Lemma 1 guarantees the existence of a ‘nearest-clone’, it demands the following two conditions:

- It can be sampled from the source distribution \( P_s \) with infinite data points.
- It is possible to search for the ‘nearest-clone’ in the \( P_s \), for a target sample \( \hat{x} \) under the distance metric \( \mathcal{D} \).

We propose to employ Variational Auto-encoding based sampling models on the source distribution to simultaneously sample and find the ‘nearest-clone’ through an optimization over the latent space.

### 3.2 Variational Auto-Encoder for source sampling

Variational Auto-Encoders (VAEs) \(^{22}\) are a class of latent-variable generative models that are based on the principles of variational inference where the variational distribution, \( Q_\phi(z|x) \) is used to approximate the intractable true posterior \( P_\theta(z|x) \). The log-likelihood of the observed data is decomposed into two terms, an irreducible non-negative KL-divergence between \( P_\theta(z|x) \) and \( Q_\phi(z|x) \) and the Evidence Lower Bound (ELBO) term which is given by Eq. 5:

\[
\ln P_\theta(x) = \mathcal{L}(\theta, \phi) + D_{KL}[Q_\phi(z|x)||P_\theta(z|x)]
\]

where,

\[
\mathcal{L}(\theta, \phi) = \mathbb{E}_{Q_\phi(z|x)}[\ln (P_\theta(x|z))] - D_{KL}[Q_\phi(z|x)||P_\theta(z)]
\]

The non-negative KL-term in Eq. 5 is irreducible and thus, \( \mathcal{L}(\theta, \phi) \) serves as a lower bound on the data log-likelihood which is maximized in a VAE by parameterizing \( Q_\phi(z|x) \) and \( P_\phi(x|z) \) using probabilistic encoder \( g_\phi \) (that outputs the parameters \( \mu_z \) and \( \sigma_z \) of a distribution) and decoder \( h_\theta \) neural networks.
Fig. 2: Latent Search procedure during inference with GLSS. The latent vector \( z \) is initialized with a random sample drawn from \( \mathcal{N}(0,1) \). Iterations over the latent space \( z \) are performed to minimize the Structural Similarity loss \( L_{ssim} \) between the input target image \( \tilde{x}_T \) and the predicted target image \( \hat{x} \), which are the output of the trained decoder (blue dotted lines). After convergence of \( L_{ssim} \) loss, the optimal latent vector \( \tilde{z}_S \), generates the closest clone \( \tilde{x}_S \) which is used to predict the mask of \( \tilde{x}_T \) using the segmentation network \( S_{\psi} \) trained on source samples.

latent prior \( \mathcal{P}(z) \) is taken to be arbitrary prior on \( z \) which is usually a 0 mean and unit variance Gaussian distribution. After training, the decoder network is used as a sampler for \( \mathcal{P}_s(x) \) in a two-step process: (i) Sample \( z \sim \mathcal{N}(0, I) \), (ii) Sample \( x \) from \( \mathcal{P}_\theta(x|z) \).

The likelihood term in Eq. 5 is approximated using norm-based losses and it is known to result in blurry images. Therefore, we use the perceptual loss [19] along with the standard norm-based losses. Further, since the Edge-information is generally invariant across the source and target domains, we extract edge of the input image and use it in the decoder of the VAE via a skip connection, as shown in Fig. 1. This is shown to reduce the blur in the generated images. Fig. 1 depicts the entire VAE architecture used for training on the source data.

3.3 VAE Latent Search for finding the ‘nearest-clone’

As described, the objective of the current work is to search for the nearest point in the source distribution, given a sample from target distribution. The decoder \( h_\theta \) of a VAE trained on the source distribution \( \mathcal{P}_s(x) \), outputs a new sample from it from the Normally distributed latent sample as input. That is,

\[
\forall z \sim \mathcal{N}(0, I), \hat{x} = h_\theta(z) \sim \mathcal{P}_s(\hat{x})
\]

(7)

With this, our goal is to find the ‘nearest-clone’ to a given target sample. That is, given a \( \bar{x} \sim \mathcal{P}_t(x) \), find \( \bar{x}_S \) as follows:

\[
\bar{x}_S = h_\theta(\bar{z}_S) : \{ D(\bar{x}, \bar{x}_S) < D(x, \bar{x}) \ \forall x = h_\theta(z) \sim \mathcal{P}_s(x) \}
\]

(8)
Since $D$ is pre-defined and $h_\theta(z)$ is a Deep neural network, finding $\tilde{x}_S$ (Eq. 9) can be cast as an optimization problem with over $z$ with minimization of $D$ as the objective. Mathematically,

$$\tilde{z}_S = \arg\min_z D(\tilde{x}, h_\theta(z))$$

(9)

$$\tilde{x}_S = h_\theta(\tilde{z}_S)$$

(10)

The optimization problem is Eq. 9 can be solved using gradient-descent based techniques on the decoder network $h_{\theta^*}$ ($\theta^*$ are the parameters of the decoder network trained only on the source samples $S_n$) with respect to $z$. This implies that given any input target image, the optimization problem in Eq. 9 will be solved to find its ‘nearest-clone’ in the source distribution which is used as a proxy in the segmentation network trained only on $S_n$. We call the iterative procedure of finding $\tilde{x}_S$ through optimization using $h_{\theta^*}$ as the Latent Search (LS). Finally, inspired by the observations made in [16], we propose to use structural similarity index (SSIM) based loss $L_{ssim}$ for $D$ to conduct the Latent Search. Unlike norm-based losses, SSIM loss helps in preservation of structural information which is needed for segmentation. Fig. 5 depicts the complete inference procedure employed in the proposed method named as the Generative Latent Search for Segmentation (GLSS).

4 Implementation Details

4.1 Training

Architectural details of the VAE used are shown in Fig. 1. Sobel operator is used to extract the edge information of the input image which is concatenated with one of the layers of the Decoder via a $\text{tanh}$ non linearity as shown in Fig. 1. The VAE is trained using (i) the Mean squared error reconstruction loss $L_r$ and KL divergence $D_{KL}$ and (ii) the perceptual loss $L_p$ for which the features are taken from the $l$th layer (a hyper-parameter) of the Deeplabv3+ [6] and the Unet [37] with EfficientNet backbone [40] segmentation network. The segmentation network ($S_\psi$ Fig. 5) is either DeepLabv3+ with Xception network [9] or Unet with EfficientNet network and is trained on source dataset. For training $S_\psi$ we use combination of binary cross-entropy ($L_{bce}$) and dice coefficient loss ($L_{dice}$) for Unet with EfficientNet with RMSProp ($lr = 0.001$) and binary focal loss ($L_{focal}$) [27] with $\gamma = 2.0$, $\alpha = 0.75$ and RMSProp ($lr=0.01$) for Deeplabv3+ (XceptionNet). For the VAE , the hidden layers of Encoder and Decoder networks use Leaky ReLU and $\text{tanh}$ as activation functions with the dimensionality of the latent space being 64. VAE is trained using standard gradient descent procedure with RMSprop ($\alpha=0.0001$) as optimizer. We train VAE for 150-200k iterations with batchsize 64. The first part of Algorithm 1 shows the steps involved in training VAE.
### Algorithm 1 Generative Latent Search for Segmentation (GLSS)

**Training VAE on source samples**

**Input:** Source dataset $\mathcal{S}_n = \{x_1, \ldots, x_n\}$, 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 $\phi^\ast$, $\theta^\ast$.

1. Initialize parameters $\phi$, $\theta$
2. repeat
3. sample batch $\{x_i\}$ from dataset $\mathcal{S}_n$, for $i = 1, \ldots, B$
4. $\mu^{(i)}_z, \sigma^{(i)}_z \leftarrow g_\phi(x_i)$
5. sample $z_i \sim \mathcal{N}(\mu^{(i)}_z, \sigma^{(i)}_z)^2$
6. $\mathcal{L}_r \leftarrow \sum_{i=1}^B \|x_i - h_\theta(z_i)\|^2_2$
7. $\mathcal{L}_p \leftarrow \sum_{i=1}^B \|P_\psi(x_i) - P_\psi(h_\theta(z_i))\|^2_2$
8. $\mathcal{L}_g \leftarrow \mathcal{L}_r + \mathcal{L}_p + \sum_{i=1}^B D_{KL}\left[\mathcal{N}(\mu^{(i)}_z, \sigma^{(i)}_z)^2 \| \mathcal{N}(0, 1)\right]$
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 $\tilde{x}_T$, Trained decoder $h_\theta^\ast$, Learning rate $\eta$. **Output:** ‘nearest-clone’ $\tilde{x}_S$ for the target sample $\tilde{x}_T$.

13. sample $z$ from $\mathcal{N}(0, 1)$
14. repeat
15. $\mathcal{L}_{ssim} \leftarrow 1 - \text{SSIM}(\tilde{x}_T, h_\theta^\ast(z))$
16. $z \leftarrow z + \eta \nabla_z \mathcal{L}_{ssim}$
17. until convergence of $\mathcal{L}_{ssim}$
18. $\tilde{z}_S \leftarrow z$
19. $\tilde{x}_S \leftarrow h_\theta^\ast(\tilde{z}_S)$

#### 4.2 Inference

Once the VAE is trained on the source dataset, given an image $\tilde{x}_T$ from the target distribution, the Latent Search algorithm searches for an optimal latent vector $\tilde{z}_S$ that generates its ‘nearest-clone’ $\tilde{x}_S$ from $\mathcal{P}_S$. The search is performed by minimizing the SSIM loss $\mathcal{L}_{ssim}$ between the input target image $\tilde{x}_T$ and the VAE-reconstructed target image, using a gradient-descent based optimization procedure such as ADAM [21] with $\alpha = 0.1$, $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The Latent Search is performed for $K$ (hyper-parameter) iterations over the latent space of the source for a given target image. Finally, the segmentation mask for the input target sample is assigned same as the one given by the segmentation network which is trained on source data $\mathcal{S}_\psi$ on the ‘nearest-clone’ $\tilde{x}_S$. Second part of Algorithm 1 shows the steps involved in inference procedure.
5 Experiment and Results

5.1 Datasets

We consider the Red-channel of the COMPAQ dataset as our source data. It consists of 4675 RGB images with the corresponding annotations of the skin, a few samples of which are shown in the First row of Fig. 3. Since there is no publicly available dataset with NIR images and corresponding skin segmentation labels we create and use two NIR datasets (which we make publicly available) as targets. The first one named as the Skin NIR Vision (SNV) consists of 800 images of multiple human subjects taken in different scenes, captured using a WANSVIEW 720P camera in the night-vision mode. The captured images cover wide range of scenarios which encumbers skin detection task like presence of multiple humans, backgrounds similar to skin color, different illuminations, saturation levels and different postures of subjects to ensure diversity. Additionally, we made use of the publicly available multi-modal Hand Gesture dataset as another target dataset which we call as Hand Gesture dataset. This dataset covers 16 different hand-poses of multiple subjects. We randomly sampled 500 images in-order to cover illumination change and diversity in hand poses. Both SNV and Hand Gesture datasets are manually annotated with precision. Fig. 3 shows few images with the corresponding skin-mask pairs from both the datasets.

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

5.2 Benchmarking on SNV and Hand Gestures data

To begin, we performed supervised segmentation experiments on both SNV and Hand Gestures datasets with 80-20 train-test splits using SOTA segmentation algorithms, to create performance upper-bound. Table 1 shows the standard performance metrics such as IoU and DICE-coefficient measured using FPN

1 https://www.gti.ssr.upm.es/data/MultiModalHandGesture_dataset
Table 1: Benchmarking Skin NIR Vision dataset (SNV) and Hand Gesture dataset on standard architectures with 80-20 train-test split.

| Method         | SNV          | Hand Gesture |
|----------------|--------------|--------------|
|                | IoU  | Dice | IoU  | Dice |
| FPN [26]       | 0.792 | 0.895 | 0.902 | 0.950 |
| Unet [37]      | 0.798 | 0.890 | 0.903 | 0.950 |
| DeepLabv3+ [6] | 0.750 | 0.850 | 0.860 | 0.924 |
| Linknet [5]    | 0.768 | 0.872 | 0.907 | 0.952 |
| PSPNet [48]    | 0.757 | 0.850 | 0.905 | 0.949 |

Table 2: Empirical analysis of GLSS along with standard UDA methods. IoU and DICE-coefficient are computed for both SNV and Hand Gesture datasets using Unet (EfficientNet) and Deeplabv3+ (XceptionNet) as segmentation networks.

| Models      | SNV         | Hand Gesture |
|-------------|-------------|--------------|
|              | Unet | DeepLabv3+ | Unet | DeepLabv3+ |
|              | IoU  | Dice | IoU  | Dice | IoU  | Dice |
| Source Only | 0.295 | 0.426 | 0.215 | 0.426 | 0.601 | 0.711 |
| AdaptSegnet | 0.315 | 0.435 | 0.320 | 0.435 | 0.614 | 0.720 |
| Advent      | 0.341 | 0.517 | 0.508 | 0.540 | 0.622 | 0.729 |
| CLAN        | 0.248 | 0.442 | 0.255 | 0.426 | 0.622 | 0.729 |
| BDL         | 0.320 | 0.518 | 0.301 | 0.509 | 0.647 | 0.720 |
| DICE        | 0.341 | 0.557 | 0.339 | 0.532 | 0.672 | 0.789 |
| DADA        | 0.332 | 0.534 | 0.314 | 0.521 | 0.643 | 0.743 |
| ours (GLSS) | **0.406** | **0.597** | **0.385** | **0.597** | **0.736** | **0.844** |

5.3 Baseline UDA Experiments

We have performed the UDA experiments with the SOTA UDA algorithms using Red-channel of the COMPAQ as the source and SNV and Hand Gesture as the target. Table 2 compares the performance of proposed GLSS algorithm with six SOTA baselines along with the Source Only case (without any UDA). We have used entire target dataset for IoU and DICE-coefficient evaluation.

Two architectures Deeplabv3+ (XceptionNet) and Unet (EfficientNet) were used for the segmentation network \( S_\psi \). It can be seen that although all the UDA SOTA methods improve upon the Source Only performance, GLSS offers significantly better performance despite not using any data from the target distribution. Hence, it may be empirically inferred that GLSS is successful in...
Fig. 4: Qualitative comparison of predicted segmentation skin masks on SNV and Hand Gesture datasets with standard UDA methods. Top *four* rows show skin masks for SNV dataset and the last *four* are the masks for Hand Gesture dataset. It is evident that GLSS predicted masks are very close to the GT masks as compared to other UDA methods. (SO=Source Only, ASN=AdaptSegNet, GT=Ground Truth).

producing the ‘nearest-clone’ through implicit sampling from the source distribution, reducing the domain shift. It is also observed that the performance of the segmentation network \( S_\psi \) does not degrade on the source data with GLSS. The output predicted masks with Deeplabv3+ (XceptionNet) on a few images are shown in Fig. 4 for SNV and Hand Gesture datasets, respectively. It can been that GLSS is able capture fine facial details like eyes, lips etc., and body parts like hands better as compared to SOTA methods. It is also seen that the predicted masks for Hand Gesture dataset are sharper in comparison to other methods.

5.4 Ablation Study

We have conducted several ablation experiments on GLSS using both SNV and Hand Gesture datasets using DeepLabv3+ (XceptionNet) as segmentation networks \( (S_\psi) \) to ascertain the utility of different design choices we have made in our method.

**Effect of number of iterations on LS:** The inference of GLSS involves a gradient-based optimization through the decoder network \( h_\theta \) to generate the
Skin Seg. from NIR images using UDA through GLS

Fig. 5: Illustration of Latent Search in GLSS. Real target is a ground truth mask. Source Only masks are obtained from target samples by training segmentation network $S_\psi$ on source dataset. Prior to the LS, skin masks are obtained from VAE reconstructed target samples. It is evident that predicted skin masks improve as the LS progresses. The predicted masks for the ‘nearest-clones’ are shown after every 30 iterations.

Fig. 6: Performance of gradient-based Latent Search during inference on target SNV and Hand Gesture images using different objective functions; MSE, MAE, SSIM loss. Deeplabv3+ (XceptionNet) is employed as segmentation network. It is evident that the losses saturate at around 90-100 iterations.

‘nearest-clone’ for a given target image. In Fig. 5 we show the skin masks of the transformed target images after every 30 iterations. It is seen that with the increasing number of iterations, the predicted skin masks improves using GLSS as the ‘nearest-clones’ are optimized during the Latent Search procedure. We plot the IoU as a function of the number of iterations during Latent Search as shown in Fig. 6 where it is seen that it saturates around 90-100 iterations that are used for all the UDA experiments described in the previous section.

**Effect of Edge concatenation:** As discussed earlier, edges computed using Sobel filter on input images are concatenated with one of the layers of decoder for both training and inference. It is seen from Table 3 that IoU improves for both the target datasets with concatenation of edges. It is observed that without
Table 3: Ablation of different components of GLSS during training and inference; Edge, perceptual loss $L_p$ and Latent Search (LS).

| Edge | $L_p$ | LS | SNV IoU | Hand Gesture IoU |
|------|-------|----|---------|------------------|
| ✓    | ✓     | ✓  | 0.112   | 0.227            |
| ✓    | ✓     | ✓  | 0.178   | 0.560            |
| ✓    | ✓     | ✓  | 0.120   | 0.250            |
| ✓    | ✓     | ✓  | 0.128   | 0.238            |
| ✓    | ✓     | ✓  | 0.330   | 0.615            |
| ✓    | ✓     | ✓  | 0.182   | 0.300            |
| ✓    | ✓     | ✓  | 0.223   | 0.58             |
| ✓    | ✓     | ✓  | 0.385   | 0.698            |

the edge concatenation, the generated images (‘nearest-clones’) are blurry thus the segmentation network fails to predict sharper skin masks.

**Effect of Perceptual loss $L_p$:** We have introduced a perceptual model $P_\psi$ while training on source samples. It ensures that the VAE reconstructed image is semantically similar to the input unlike the norm-based losses. Table 3 clearly demonstrates the improvement offered by the use of perceptual loss while training the VAE.

**Effect of SSIM for Latent Search:** Finally to validate the effect of SSIM loss for Latent Search, in Fig. 6 we plot the IOU metrics using two norm based losses MSE (Mean squared error) and MAE (Mean absolute error) for the Latent Search procedure. On both the datasets, it is consistently seen that SSIM is better than the norm-based losses at all iterations affirming the superiority of the SSIM loss in preserving the structures while finding the ‘nearest-clone’.

6 Conclusion

In this paper, we addressed the problem of skin segmentation from NIR images. Owing to the non-existence of large-scale labelled NIR datasets with skin segmentation, the problem is casted as unsupervised domain adaptation where we use the segmentation network trained on the Red-channel of a large-scaled labelled Visible-spectrum dataset is adapted to NIR data. We propose a novel method for the UDA without the need for the access to the target data (even unlabelled). Given a target image, we sample an image from the source distribution that is ‘closest’ to it under a distance metric. We show that such a ‘closest’ sample exists and describe a procedure using an optimization algorithm over the latent space of a VAE trained on the source data. We demonstrated utility of the proposed method along with the comparisons with the SOTA UDA segmentation methods on the skin segmentation task on two NIR datasets that were created. Our future work aims at exploring the proposed method for a generalized UDA segmentation task.
References

1. Cycada: Cycle consistent adversarial domain adaptation. In: International Conference on Machine Learning (ICML) (2018)
2. Al-Mohair, H.K., Saleh, J., Saundi, S.: Impact of color space on human skin color detection using an intelligent system. In: 1st WSEAS international conference on image processing and pattern recognition (IPPR’13). vol. 2 (2013)
3. Brancati, N., De Pietro, G., Frucci, M., Gallo, L.: Human skin detection through correlation rules between the ycb and ycr subspaces based on dynamic color clustering. Computer Vision and Image Understanding 155, 33–42 (2017)
4. 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)
5. Chaurasia, A., Culurciello, E.: Linknet: Exploiting encoder representations for efficient semantic segmentation. In: 2017 IEEE Visual Communications and Image Processing (VCIP). pp. 1–4. IEEE (2017)
6. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
7. Chen, W., Wang, K., Jiang, H., Li, M.: Skin color modeling for face detection and segmentation: a review and a new approach. Multimedia Tools and Applications 75(2), 839–862 (2016)
8. Chen, W.C., Wang, M.S.: Region-based and content adaptive skin detection in color images. International journal of pattern recognition and artificial intelligence 21(05), 831–853 (2007)
9. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1251–1258 (2017)
10. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE transactions on information theory 13(1), 21–27 (1967)
11. Dourado, A., Guth, F., de Campos, T.E., Weigang, L.: Domain adaptation for holistic skin detection. arXiv preprint arXiv:1903.06969 (2019)
12. Erdem, C., Ulukaya, S., Karaali, A., Erdem, A.T.: Combining haar feature and skin color based classifiers for face detection. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1497–1500. IEEE (2011)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
14. He, Y., Shi, J., Wang, C., Huang, H., Liu, J., Li, G., Liu, R., Wang, J.: Semi-supervised skin detection by network with mutual guidance. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2111–2120 (2019)
15. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649 (2016)
16. Hore, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: 2010 20th International Conference on Pattern Recognition. pp. 2366–2369. IEEE (2010)
17. Hsu, R.L., Abdel-Mottaleb, M., Jain, A.K.: Face detection in color images. IEEE transactions on pattern analysis and machine intelligence 24(5), 696–706 (2002)
18. Huynh-Thu, Q., Meguro, M., Kaneko, M.: Skin-color-based image segmentation and its application in face detection. In: MVA. pp. 48–51 (2002)
19. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
20. Jones, M.J., Rehg, J.M.: Statistical color models with application to skin detection. International Journal of Computer Vision 46(1), 81–96 (2002)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
22. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
23. Kong, S.G., Heo, J., Abidi, B.R., Paik, J., Abidi, M.A.: Recent advances in visual and infrared face recognition—a review. Computer Vision and Image Understanding 97(1), 103–135 (2005)
24. Kovac, J., Peer, P., Solina, F.: Human skin color clustering for face detection, vol. 2. IEEE (2003)
25. Li, Y., Yuan, L., Vasconcelos, N.: Bidirectional learning for domain adaptation of semantic segmentation. arXiv preprint arXiv:1904.10620 (2019)
26. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
27. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
28. Liu, Q., Peng, G.z.: A robust skin color based face detection algorithm. In: 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010). vol. 2, pp. 525–528. IEEE (2010)
29. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
30. 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)
31. Mahmoodi, M.R.: High performance novel skin segmentation algorithm for images with complex background. arXiv preprint arXiv:1701.05588 (2017)
32. Mahmoodi, M.R., Sayedi, S.M.: A comprehensive survey on human skin detection. International Journal of Image, Graphics & Signal Processing 8(5) (2016)
33. Moallem, P., Mousavi, B.S., Monadjemi, S.A.: A novel fuzzy rule base system for pose independent faces detection. Applied Soft Computing 11(2), 1801–1810 (2011)
34. Pan, Z., Healey, G., Prasad, M., Tromberg, B.: Face recognition in hyperspectral images. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(12), 1552–1560 (2003)
35. Prathosh, A., Praveena, P., Mestha, L.K., Bharadwaj, S.: Estimation of respiratory pattern from video using selective ensemble aggregation. IEEE Transactions on Signal Processing 65(11), 2902–2916 (2017)
36. Qiang-rong, J., Hua-lan, L.: Robust human face detection in complicated color images. In: 2010 2nd IEEE International Conference on Information Management and Engineering. pp. 218–221. IEEE (2010)
37. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
38. Seow, M.J., Valaparla, D., Asari, V.K.: Neural network based skin color model for face detection. In: 32nd Applied Imagery Pattern Recognition Workshop, 2003. Proceedings. pp. 141–145. IEEE (2003)
39. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
40. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 (2019)
41. Taqa, A.Y., Jalab, H.A.: Increasing the reliability of skin detectors. Scientific Research and Essays 5(17), 2480–2490 (2010)
42. 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)
43. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: CVPR (2019)
44. 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)
45. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13(4), 600–612 (2004)
46. Wu, Q., Cai, R., Fan, L., Ruan, C., Leng, G.: Skin detection using color processing mechanism inspired by the visual system (2012)
47. Zaidan, A., Ahmad, N.N., Karim, H.A., Larbani, M., Zaidan, B., Sali, A.: On the multi-agent learning neural and bayesian methods in skin detector and pornography classifier: An automated anti-pornography system. Neurocomputing 131, 397–418 (2014)
48. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890 (2017)
Skin Segmentation from NIR Images using Unsupervised Domain Adaptation through Generative Latent Search
—Supplementary—

Prashant Pandey*, Aayush Kumar Tyagi*, Sameer Ambekar, and Prathosh AP

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

1 Implementation details

$S_\psi$ is the segmentation model (as shown in Fig. 2 in the paper) implemented using DeepLabv3+ (XceptionNet) and Unet (EfficientNet). $S_\psi$ is trained for 75-100 epochs with losses ($L_s$) as shown in Eq. 1 and Eq. 2 for Unet (EfficientNet) and DeepLabv3+ (XceptionNet) respectively.

$$L_s = L_{dice} + L_{bce} \quad (1)$$

$$L_s = L_{focal} \quad (2)$$

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

$P_\psi$ is a perceptual model (as shown in Fig. 1 in the paper) that uses perceptual loss $L_p$. The perceptual features are taken from the 6th layer of Unet (EfficientNet) and the last concatenation layer of DeepLabv3+ (XceptionNet). VAE along with perceptual loss $L_p$ is trained for 150-200 epochs. $L_p$ is weighted with a factor $\beta$ (a hyper-parameter) as shown:

$$L_{total} = L_{vae} + \beta L_p \quad (3)$$

In order to improve the quality of VAE reconstructed images, we weighted the perceptual loss ($L_p$) with different values of $\beta$. For Unet (EfficientNet), we have used $\beta = 2$ whereas $\beta = 3$ is used for DeepLabv3+ (XceptionNet).

2 Additional Results

* equal contribution
Skin Seg. from NIR images using UDA through GLS

real target
VAE reconstruction

Fig. 1: 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. 2: 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. 3: (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.
Fig. 4: (a) an NIR image \( \tilde{x}_T \) from SNV dataset (target), (b) ‘nearest-clone’ \( \tilde{x}_S \) generated from GLSS, (c) Structural Similarity Index (SSIM) scores calculated between \( \tilde{x}_T \) and all the samples (having only Red-channel) of COMPAQ dataset (source) are shown with blue color in the plot. Similarly, SSIM scores calculated between \( \tilde{x}_S \) 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’ \( \tilde{x}_S \) as compared to the scores with \( \tilde{x}_T \). It indicates that \( \tilde{x}_S \) is more closer to the source domain (COMPAQ) as compared to \( \tilde{x}_T \). Hence, the ‘nearest-clone’ \( \tilde{x}_S \) generated by GLSS for target \( \tilde{x}_T \) is used as a proxy in the segmentation network \( S_\psi \) which is trained only on COMPAQ dataset, thereby increasing the IoU for \( \tilde{x}_T \).
Fig. 5: (a) an NIR image $\tilde{x}_T$ from Hand Gesture dataset (target), (b) ‘nearest-clone’ $\tilde{x}_S$ generated from GLSS, (c) Structural Similarity Index (SSIM) scores calculated between $\tilde{x}_T$ and all the samples (having only Red-channel) of COMPAQ dataset (source) are shown with blue color in the plot. Similarly, SSIM scores calculated between $\tilde{x}_S$ 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’ $\tilde{x}_S$ as compared to the scores with $\tilde{x}_T$. It indicates that $\tilde{x}_S$ is more closer to the source domain (COMPAQ) as compared to $\tilde{x}_T$. Hence, the ‘nearest-clone’ $\tilde{x}_S$ generated by GLSS for target $\tilde{x}_T$ is used as a proxy in the segmentation network $S_\psi$ which is trained only on COMPAQ dataset, thereby increasing the IoU for $\tilde{x}_T$. 