Target-Independent Domain Adaptation for WBC Classification using Generative Latent Search

Prashant Pandey, Prathosh AP, Vinay Kyatham, Deepak Mishra and Tathagato Rai Dastidar

***Abstract*—Automating the classification of camera-obtained microscopic images of White Blood Cells (WBCs) and related cell subtypes has assumed importance since it aids the laborious manual process of review and diagnosis. Several State-Of-The-Art (SOTA) methods developed using Deep Convolutional Neural Networks suffer from the problem of domain shift - severe performance degradation when they are tested on data (target) obtained in a setting different from that of the training (source). The change in the target data might be caused by factors such as differences in camera/microscope types, lenses, lighting-conditions etc. This problem can potentially be solved using Unsupervised Domain Adaptation (UDA) techniques albeit standard algorithms presuppose the existence of a sufficient amount of unlabelled target data which is not always the case with medical images. In this paper, we propose a method for UDA that is devoid of the need for target data. Given a test image from the target data, we obtain its 'closest-clone' from the source data that is used as a proxy in the classifier. We prove the existence of such a clone given that infinite number of data points can be sampled from the source distribution. We propose a method in which a latent-variable generative model based on variational inference is used to simultaneously sample and find the 'closest-clone' from the source distribution through an optimization procedure in the latent space. We demonstrate the efficacy of the proposed method over several SOTA UDA methods for WBC classification on datasets captured using different imaging modalities under multiple settings.**

***Index Terms*—WBC, Microscopic imaging, Unsupervised domain adaptation, Generative models, VAE.**

I. Introduction

A. Background

MICROSCOPIC review of Peripheral Blood Smear (PBS) slides by clinical pathologists is considered as the gold standard for detection of various disorders [1]. This requires manual counting and classification of various types of cells, including White Blood Cells (WBCs or leukocytes) and analysing their morphological characteristics in PBS slides. The presence, absence, or relative counts of these cells help in the diagnosis of several types of diseases, including different forms of blood cancer, anaemia, and presence of parasites like in malaria. This process of manual review is both laborious and error prone. In addition, due to variations in stain, smearing process, the differentiation between various subclasses of cells is often blurry. It takes significant expertise and experience to correctly classify all types of cells. Lack of qualified medical professionals, especially in non-urban areas of developing countries, accentuates the problem. Furthermore, the misdiagnosis, often caused by lack of adequate time to examine a slide thoroughly, can even lead to fatalities. Thus, automating and standardising this process is a pressing need.

Several attempts have been made to automate some of these manual processes using methods ranging from classical computer vision [2–4] to image cytometry [5, 6]. While classical vision techniques suffer from issues like poor-generalization, image cytometry is limited by its operational speed and inability to engineer complex features [7]. An alternative is to harness the power of Deep Convolutional Neural Networks (CNNs) in addressing some of these issues [8]. In SC-CNN [9], a weighted sum of multiple classifiers is used to predict the class label of cell nuclei detected with a Spatially Constrained CNN. In [10], a Conditional Generative Adversarial Network (cGAN) [11] is used for nuclei segmentation, a fundamental task for cell classification. MGCNN [12] is a White Blood Cells classification framework that combines modulated Gabor wavelet [13] and deep CNN kernels. A few commercial products too have been built utilizing some of these techniques. CellaVision [14], Shonit [15], etc., automate the counting and classification of leukocytes and other blood cells. These systems consist of an automated microscope equipped with a digital camera, which captures the images of a biological sample on a glass slide. A software based analysis system, built using CNN models, is then used to localise and classify different types of cells in the sample.

B. Motivation and Problem setting

Even though the aforementioned models and systems are effective in their own ways, they suffer from certain issues that may limit their utility. For instance, Deep CNN models used for microscopic image classification are typically trained using proprietary datasets. These datasets tend to be homogeneous in terms of the capture device – microscopes, lens and cameras used. This homogeneity and limited number of images in the training dataset cause the models trained on them to over-fit on specific characteristics of the image capturing device. As a result, when images captured with a different device or camera

Prashant Pandey and Prathosh AP are with Department of Electrical Engineering, Indian Institute of Technology Delhi, New Delhi 110016, India. Deepak Mishra is with Department of Computer Science and Engineering, Indian Institute of Technology Jodhpur, Rajasthan 342037, India. Vinay Kyatham and Tathagato Rai Dastidar are with SigTuple Technologies Pvt. Ltd., Bangalore 560102, India. Email: getprashant57@gmail.com, prathoshap@iitd.ac.in, dmishra@iitj.ac.in, trd@sigtuple.com, vinay.k@sigtuple.com. The code for our implementation is available at https://github.com/prinshul/WBC-Classification-UDA.

are presented to these models, they often wrongly classify new images, even though the trained human observers will have no difficulty in classification (images shown in Figure 3). Hence, as the image capturing device changes, these models fail to adapt to the new input data distribution. This is known as the domain shift problem. Domain shift also occurs when the underlying imaging modality itself changes. For instance, a deep learning model trained on Flow Cytometry images [6] will not readily generalize for microscopic PBS images even though both capture WBCs. The problem of domain shift exists not only for medical images, but for any deep learning system trained with single image source [16].

A natural solution to this problem is to (re)-train the model with large amount of data obtained from the new device. However, generating sufficient quantity of annotated medical data is a time consuming and costly process. In addition, bottlenecks such as regulatory clearances, cause a large development cycle and delay in building such systems. We consider one such problem in this paper, where performance of CNNs trained on a dataset from a single source camera for automatic classification of images of WBCs taken from PBS, degrade when tested on unseen target dataset collected from different cameras. This falls within the ambit of a well-known computer vision problem known as Unsupervised Domain Adaptation (UDA). However, almost all the SOTA methods on UDA [16–18] need access to the unlabelled target data during the time of training. While it may be feasible to obtain unlabelled target data, retraining of the UDA model for every newly emerging target domain might be infeasible, post their deployment in the field. Therefore, an unsupervised domain adaptation method that can operate without target data is desirable [19]. Motivated by these observations, in this paper we propose a UDA technique for WBC classification with following core contributions:

1) We propose a UDA technique that does not require access to the target data during the time of training.
2) We cast the problem of UDA as finding the 'closest-clone' in the source domain for a given target image that is used as a proxy for the target image in the classifier trained on the source data.
3) We theoretically prove the existence of the 'closest-clone' given that infinite data points can be sampled from the source distribution.
4) We propose an optimization method over the latent space of variational inference based Deep generative model, to find the aforementioned clone through implicit sampling.
5) We demonstrate through extensive experimentation, the efficacy of the proposed method over several state-of-the-art UDA techniques for WBC classification on several datasets obtained using different imaging modalities with multiple domain shifts. We also validate our algorithm on the standard datasets used for UDA.

II. Related work

Unsupervised Domain Adaptation (UDA) refers to the design of techniques aimed at improving the performance of machine learning tasks such as classification and segmentation when the classifier is trained using labels only from a source domain and tested on data from related but a shifted target domain. In this section, we present a review of the state-of-the-art UDA techniques based on their principle of operation and their use in the medical imaging community.

1) Adversarial-learning: These methods [16–18] learn domain-invariant representations using the principles of adversarial learning. ADDA [16] employs a source network, pre-trained with labeled source data. Adversarial adaptation is performed by learning a target network such that a domain discriminator fails to predict the domain labels of the source and target features. During inference, the target images are mapped to the shared feature space by using the target network which are predicted by the source classifier. Generate To Adapt (GTA) [18] learns domain invariant embeddings using a joint generative-discriminative set-up. During training, a feature extraction network outputs embeddings that are used by label prediction network for classification with a Generative Adversarial Network (GAN) framework to generate realistic source images. DIRT-T [20] employs a Virtual Adversarial Domain Adaptation (VADA) model that pushes the decision boundaries away from regions of high data density by penalizing violation of the cluster assumption in the target domain. Transferable Adversarial Training (TAT) [21] generates transferable examples to fill in the gap between the source and target domains without distorting feature distributions. Domain Agnostic Learning (DAL) [22] uses Deep Adversarial Disentangled Auto-Encoders (DADA) to disentangle domain-invariant features in the latent space by minimizing the mutual information between domain-invariant and domain-specific features. The principles of adversarial feature learning has been used in [23, 24] to transform real images to a synthetic-like representation using unlabeled synthetic endoscopy images and achieve stain independence. In [25], a siamese architecture with adversarial training is used to improve the classification performance of target prostate histopathology whole-slide images. Zhang et al. [26] used adversarial learning for a noise adaptation task that allows a trained model to work effectively for medical images with different noise patterns.

2) Target Reconstruction: These approaches for UDA reconstructs source or target samples as an auxiliary task that simultaneously focuses on creating a shared representation between the two domains while keeping the individual characteristics of each domain intact. CyCADA [27] adapts between domains by aligning both generative and latent space representations, with cycle and semantic consistency loss. PixelDA [28] learns transformation in the pixel space from one domain to the other using task-specific and content–similarity losses. SBADA-GAN [29] maps source samples into the target domain and vice versa by imposing a class consistency loss to improve the quality of reconstructed images. I2I Adapt [30] is a framework that learns from the source domain and adapt to the target domain by extraction of domain agnostic features, domain specific reconstruction with cycle consistency losses. Tulder et al. [31] proposed a representation learning method that transforms data from different sources to a shared feature representation using per-feature normalization, a cross-modality based objective function. Goetz et al. [32] used

domain adaptation to correct the sampling bias introduced with sparsely labeled MR images for tissue classification.

3) Divergence Minimization: In these methods, source and target distributions are aligned by minimizing a divergence measure between the two distributions. Joint Adaptation Networks (JAN) [33] learns a transfer network by aligning the joint distributions of multiple domain-specific layers across domains based on a Joint Maximum Mean Discrepancy (JMMD) criterion. Maximum Classifier Discrepancy (MCD) [34] aligns distributions of source and target by utilizing the task-specific decision boundaries. Task-specific classifiers are trained to detect the target samples that are far from the support of the source. Contrastive Adaptation Network (CAN) [35] estimates the underlying label hypothesis of target samples through clustering and adapts the feature representations according to the Contrastive Domain Discrepancy (CDD) metric. Pacheco et al. [36] addressed the discrepancies related to the stem cell differentiation process by minimizing a Maximum Mean Discrepancy (MMD) based loss function in a Recurrent Neural Network (RNN) classifier.

4) Domain Randomization: Domain Randomization [37] (DR) is another class of methods related to UDA that are used to improve the generalization of classifiers. The idea is to reduce the domain shift by randomizing properties in the training environment (like source domain). Every data point in the source domain is perturbed randomly during training while assigning the same ground truth to the perturbed samples. In methods such as [38], cinematically rendered source domain images are varied in color and texture. For RGB images, such transformations can be obtained by varying hue, saturation, contrast and brightness. In [39], source images intensity is divided into multiple non-overlapping ranges. A random perturbation is added to the start/end pixel values by sampling from a Gaussian distribution. Finally, one of the following randomisation is applied to each range, a) shift the intensity values by adding a random value from a uniform distribution or b) transform the intensity values using cumulative distribution function of beta distribution or c) simply invert the intensity range. [40] varies source images background color, add uniform noise, change the illumination and distort source images with different scaling factors.

III. Proposed Method

A. Motivation

All the UDA methods mentioned in the previous section assumes that one has access to images from the target distribution. These images are either used to retrain the original classifier in a domain-invariant way [16–18] or to align the target distribution to the source distribution [27, 28, 33, 35]. Also, in most of the methods [16–18, 35], the original classifier trained on the source data is altered, so that a new decision boundary is learned using the images from the target data in an unsupervised manner. However, in many practical situations, such as the current one, there would neither be access to the target data nor the scope to retrain the classifier. Further, a new unseen target domain may arise in the field which was not used during adaptation.

We propose to address these issues in this paper by first assuming that the classifier learned on the source data (Oracle classifier) will perform well as long as the data comes from the source distribution. Subsequently, (i) we learn to sample from the source distribution and (ii) given an image from the target distribution, we find an image from the source distribution that is arbitrarily close ('closest-clone') to the given target image, under some distance metric. Finally, the target image is replaced with its 'closest-clone' from the source distribution before its class is inferred by the Oracle classifier.

B. Existence of closest source 'clone'

To begin with, we prove that given an image from the target distribution, there exists an arbitrarily close image in the source distribution (named as 'closest-clone'), provided infinite data can be sampled from the source distribution [41].

Let $\mathcal{P}_s(\mathbf{x})$ and $\mathcal{P}_t(\mathbf{x})$ denote the source and the target distributions, respectively. We assume that the the underlying random variable on which $\mathcal{P}_s$ and $\mathcal{P}_t$ are defined, forms a separable metric space $\{\mathfrak{X}, \mathfrak{D}\}$ where $\mathfrak{D}$ is some distance metric. Let $\mathcal{S}_n = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_n\}$ be i.i.d. points drawn from $\mathcal{P}_s(\mathbf{x})$ and $\tilde{\mathbf{x}}_\mathcal{T}$ be any point drawn from $\mathcal{P}_t(\mathbf{x})$. The following lemma asserts that as $n \to \infty$, there exists a point in $\mathcal{S}_n$ that it arbitrarily close to $\tilde{\mathbf{x}}_\mathcal{T}$, with probability one.

Lemma 1. *If $\tilde{\mathbf{x}}_\mathcal{S} \in \mathcal{S}_n$ is the point such that $\mathfrak{D}\{\tilde{\mathbf{x}}_\mathcal{T}, \tilde{\mathbf{x}}_\mathcal{S}\} < \mathfrak{D}\{\tilde{\mathbf{x}}_\mathcal{T}, \mathbf{x}\}\ \forall\ \mathbf{x} \in \mathcal{S}_n$, then as $n \to \infty$, $\tilde{\mathbf{x}}_\mathcal{S}$ converges to $\tilde{\mathbf{x}}_\mathcal{T}$ with probability 1 (Refer supplementary material for proof).*

Lemma 1 guarantees that given an image from the target distribution, an image from the source distribution, that is arbitrarily close to the given target image can be found out given the following requirements are met:

- Given a few images from the source distribution $\mathcal{P}_s$, one can sample infinite images from it.
- Given infinite samples from $\mathcal{P}_s$, it is possible to find the 'closest-clone' (under $\mathfrak{D}$) in $\mathcal{P}_s$, to the target image $\tilde{\mathbf{x}}_\mathcal{T}$.

To satisfy the above requirements, in subsequent sections, we employ variational inference based sampling methods on the source distribution with which one can implicitly sample and find the 'closest-clone' simultaneously.

C. Variational inference for source sampling

In variational inference based generative models [42], it is assumed that the data or the observed variable (in this case images from $\mathcal{P}_s$) is generated via a two step process: (i) sample from the distribution $\mathcal{P}_\theta(\mathbf{z})$ of an unobserved or latent variable $\mathbf{z}$, (ii) given a data point from the latent variable, sample from the conditional distribution $\mathcal{P}_\theta(\mathbf{x}|\mathbf{z})$ to obtain the data. Owing to the fact that the parameters of the true latent prior $\mathcal{P}_\theta(\mathbf{z})$ and data conditional $\mathcal{P}_\theta(\mathbf{x}|\mathbf{z})$ are unknown, and the posterior $\mathcal{P}_\theta(\mathbf{z}|\mathbf{x})$ is intractable, a variational distribution, $\mathcal{Q}_\phi(\mathbf{z}|\mathbf{x})$ is used to approximate the true posterior. With this, it can be shown that the log-likelihood of the observed data will decompose into two terms (Eq. 1), an irreducible non-negative KL-divergence between $\mathcal{P}_\theta(\mathbf{z}|\mathbf{x})$ and $\mathcal{Q}_\phi(\mathbf{z}|\mathbf{x})$ and the Evidence Lower Bound (ELBO) given by Eq. 2.

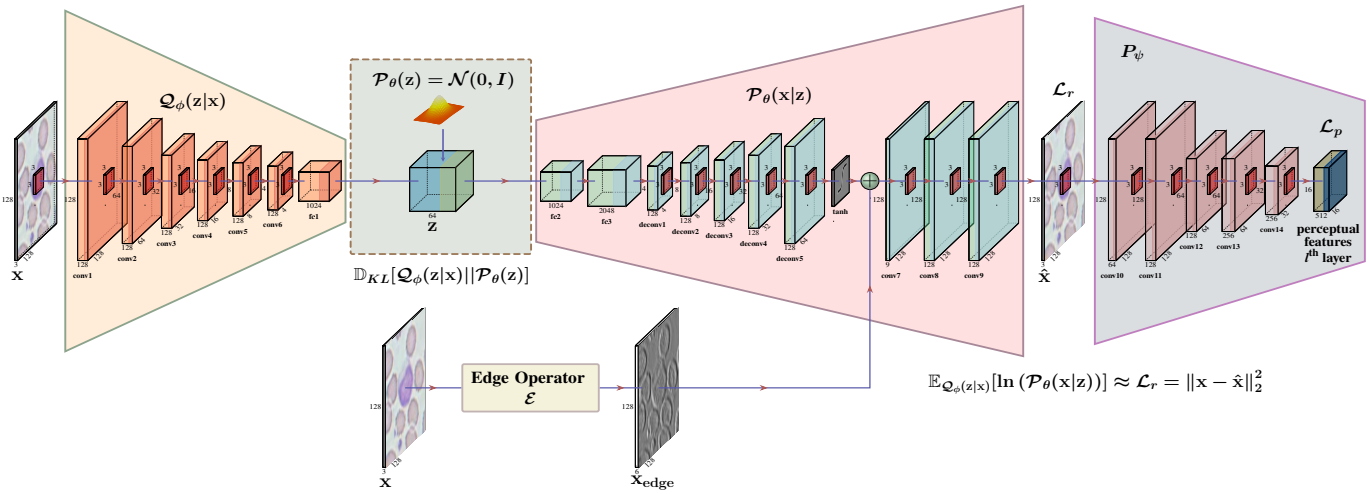


Figure 1. The architecture for the Variational Auto-Encoder in the proposed method (TIGDA). Edges of the input microscopic image is concatenated with the features from the decoder h_θ. The encoder and decoder parameters ϕ, θ are optimized with reconstruction loss $\mathcal{L}_r$, KL-divergence loss $\mathbb{D}_{KL}$ and the perceptual loss $\mathcal{L}_p$. The perceptual model P_ψ outputs l^{th} layer features of VGG-16 (or ResNet-50) classifier trained on source data. A zero mean and unit variance isotropic Gaussian prior is imposed over the latent space $\mathbf{z}$.

$$\ln \mathcal{P}_\theta(\mathbf{x}) = \mathcal{L}(\theta, \phi) + \mathbb{D}_{KL}[\mathcal{Q}_\phi(\mathbf{z}|\mathbf{x})||\mathcal{P}_\theta(\mathbf{z}|\mathbf{x})] \tag{1}$$

Here, $\mathcal{L}(\theta, \phi)$ represents ELBO which is given by,

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{\mathcal{Q}_\phi(\mathbf{z}|\mathbf{x})}[\ln(\mathcal{P}_\theta(\mathbf{x}|\mathbf{z}))] - \mathbb{D}_{KL}[\mathcal{Q}_\phi(\mathbf{z}|\mathbf{x})||\mathcal{P}_\theta(\mathbf{z})] \tag{2}$$

In Eq. 1, the KL-term is irreducible and non-negative and thus, $\mathcal{L}(\theta, \phi)$ serves as a lower bound on the data log-likelihood which is optimized. In deep generative model frameworks, $\mathcal{Q}_\phi(\mathbf{z}|\mathbf{x})$ and $\mathcal{P}_\theta(\mathbf{x}|\mathbf{z})$ are parameterized using probabilistic encoder g_ϕ (that outputs the parameters $\mu_\mathbf{z}$ and $\sigma_\mathbf{z}$ of a distribution) and decoder h_θ neural networks with parameters ϕ and θ respectively, that maps the data space into latent space and vice-versa. Additionally, $\mathcal{P}_\theta(\mathbf{z})$ is taken to be an arbitrary prior on $\mathbf{z}$ which is usually a 0 mean and unit variance Gaussian distribution. The first term in Eq. 2 is approximated using a norm-based divergence metric between the input and the output of the decoder as below:

$$\mathbb{E}_{\mathcal{Q}_\phi(\mathbf{z}|\mathbf{x})}[\ln(\mathcal{P}_\theta(\mathbf{x}|\mathbf{z}))] \approx \mathcal{L}_r = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 \tag{3}$$

Note that Eq. 3 can be seen as 'reconstruction' or 'Auto-Encoding' of the data. Further, the second term in ELBO employs a variational approximation to the true posterior $\mathcal{P}_\theta(\mathbf{z}|\mathbf{x})$. Thus, the aforementioned method is famously referred to as the Variational Auto-Encoder (VAE) [42]. For the current problem of interest, a VAE is trained using the images from the source distribution $\mathcal{S}_n$ and once trained, the decoder network serves as a sampler for the source distribution using a two step process: (i) sample $\mathbf{z} \sim \mathcal{N}(0, I)$, (ii) sample $\mathbf{x}$ as the output of the decoder h_θ.

VAEs are know to produce blurred images in their conventional formulation with norm-based losses. To address this, we use the edge information (extracted using standard edge detectors) of the input image by passing it to the decoder via a skip connection, as shown in Figure 1. Rationale behind this is that unlike features such as colour and contrast, edges are in general invariant to the changes in camera characteristics. Edge information reduces the blurring due to the decoder as shown in Figure 9 and ablation studies in Table VI.

Further, we also incorporate the perceptual loss, which is known to enhance the generation quality of VAEs, along with the standard norm-based losses. Perceptual loss $\mathcal{L}_p$ between two images $\mathbf{x}$ and $\hat{\mathbf{x}}$ is defined as the Euclidean distance between the representations or the features obtained under a pre-trained classifier model (P_ψ). Mathematically,

$$\mathcal{L}_p = \|P_\psi(\mathbf{x}) - P_\psi(\hat{\mathbf{x}})\|_2^2 \tag{4}$$

The idea behind $\mathcal{L}_p$ is that the distance metrics in a representational space learned by a classifier model trained on large scale data are better than on raw image space. This is shown to enhance image quality in several applications [43]. Figure 1 depicts the network diagram of the VAE on the source data with the proposed edge concatenation.

D. Finding 'closest-clone' through Latent Search

As mentioned in the previous sections, the objective is to simultaneously sample and search for the 'closest-clone' in the source distribution, given a sample from target distribution. Suppose a VAE has been trained on the source distribution $\mathcal{P}_s(\mathbf{x})$, the decoder h_θ of which outputs a 'de-novo' image from $\mathcal{P}_s(\mathbf{x})$ by taking a normally distributed latent variable as input. That is,

$$\forall\ \mathbf{z} \sim \mathcal{N}(0, I), \hat{\mathbf{x}} = h_\theta(\mathbf{z}) \sim \mathcal{P}_s(\hat{\mathbf{x}}) \tag{5}$$

Our goal is to find the 'closest-clone' under some distance metric $\mathfrak{D}$, for any given image from the target distribution. Mathematically, given a $\tilde{\mathbf{x}}_\mathcal{T} \sim \mathcal{P}_t(\mathbf{x})$, find $\tilde{\mathbf{x}}_\mathcal{S}$ as follows:

$$\begin{aligned} \tilde{\mathbf{x}}_\mathcal{S} = \ & h_\theta(\tilde{\mathbf{z}}_\mathcal{S}) : \Big\{ \mathfrak{D}\{\tilde{\mathbf{x}}_\mathcal{T}, \tilde{\mathbf{x}}_\mathcal{S}\} < \mathfrak{D}\{\mathbf{x}, \tilde{\mathbf{x}}_\mathcal{T}\} \\ & \forall\ \mathbf{x} \ = h_\theta(\mathbf{z}) \sim \mathcal{P}_s(\mathbf{x}) \end{aligned} \tag{6}$$

Since $\mathfrak{D}$ is computable and h_θ is a neural network that outputs a sample from $\mathcal{P}_s(\mathbf{x})$ as a function of the latent variable $\mathbf{z}$, finding $\tilde{\mathbf{x}}_\mathcal{S}$ (Eq. 7) can be cast an optimization problem over $\mathbf{z}$ with minimization of $\mathfrak{D}$ as the objective:

$$\tilde{\mathbf{z}}_\mathcal{S} = \underset{\mathbf{z}}{\operatorname{argmin}} \ \mathfrak{D}\big(\tilde{\mathbf{x}}_\mathcal{T}, h_\theta(\mathbf{z})\big) \tag{7}$$

$$\tilde{\mathbf{x}}_\mathcal{S} = h_\theta(\tilde{\mathbf{z}}_\mathcal{S}) \tag{8}$$

The optimization problem is Eq. 7 can be solved using gradient descent based techniques on the decoder network h_{θ^*} $\big(\theta^*$ are the parameters of the decoder network trained only on the source images $\mathcal{S}_n\big)$ with respect to $\mathbf{z}$. This implies that given any input image, the optimization problem in Eq. 7 will be solved to find its 'closest-clone' in the source distribution which is used as a proxy in the original classifier trained only on $\mathcal{S}_n$. We call the iterative procedure of finding $\tilde{\mathbf{x}}_\mathcal{S}$ through optimization using h_{θ^*} as the Latent Search (LS).

Finally, inspired by the observations made in [44, 45], we propose to use Structural Similarity Index (SSIM) loss for $\mathfrak{D}$ to conduct the Latent Search. Unlike norm-based losses, SSIM loss helps in preservation of structural information as compared to discrete pixel level information. SSIM is defined in [46] 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:

$$l(\mathbf{x}, \hat{\mathbf{x}}) = \frac{2\mu_\mathbf{x}\mu_{\hat{\mathbf{x}}} + C_1}{\mu_\mathbf{x}^2 + \mu_{\hat{\mathbf{x}}}^2 + C_1} \tag{9}$$

$$c(\mathbf{x}, \hat{\mathbf{x}}) = \frac{2\sigma_\mathbf{x}\sigma_{\hat{\mathbf{x}}} + C_2}{{\sigma_\mathbf{x}}^2 + {\sigma_{\hat{\mathbf{x}}}}^2 + C_2} \tag{10}$$

$$s(\mathbf{x}, \hat{\mathbf{x}}) = \frac{\sigma_{\mathbf{x}\hat{\mathbf{x}}} + C_3}{\sigma_\mathbf{x}\sigma_{\hat{\mathbf{x}}} + C_3} \tag{11}$$

where μ's denote sample means and σ's denote variances. C_1, C_2 and C_3 are constants as defined in [46]. With these, SSIM and the corresponding loss function $\mathcal{L}_{ssim}$, for a pair of images $\{\mathbf{x}, \hat{\mathbf{x}}\}$ are defined as:

$$\text{SSIM}(\mathbf{x}, \hat{\mathbf{x}}) = l(\mathbf{x}, \hat{\mathbf{x}})^\alpha \cdot c(\mathbf{x}, \hat{\mathbf{x}})^\beta \cdot s(\mathbf{x}, \hat{\mathbf{x}})^\gamma \tag{12}$$

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

$$\mathcal{L}_{ssim}(\mathbf{x}, \hat{\mathbf{x}}) = 1 - \text{SSIM}(\mathbf{x}, \hat{\mathbf{x}}) \tag{13}$$

Since our method does not utilize target images and employs generative Latent Search, we call our method Target-Independent Generative Domain Adaptation (TIGDA). The target independence of our method refers to the fact that we do not use target data during training, unlike SOTA UDA methods. The inference for TIGDA is shown in Figure 2.

IV. Implementation Details

A. Training of the VAE

The Encoder g_ϕ and Decoder h_θ network architectures for the VAE are shown in Figure 1. We use Sobel Edge operator for Edge concatenation. Edges of the input image are concatenated with the output of *tanh* nonlinearity as shown in Figure 1. The VAE is trained using (i) the Mean squared error reconstruction loss $\mathcal{L}_r$ between the real and VAE reconstructed images and (ii) the perceptual loss $\mathcal{L}_p$ for which the features are taken from the l^{th} layer of the VGG-16 (10^{th} layer) or RestNet-50 (38^{th} layer) classifier trained on source images for WBC classification task. The hidden layers of Encoder and Decoder networks use Leaky ReLU and *tanh* as activation functions with the dimensionality of the latent space being 64. VAE is trained using a standard gradient descent procedure with RMSprop optimizer.

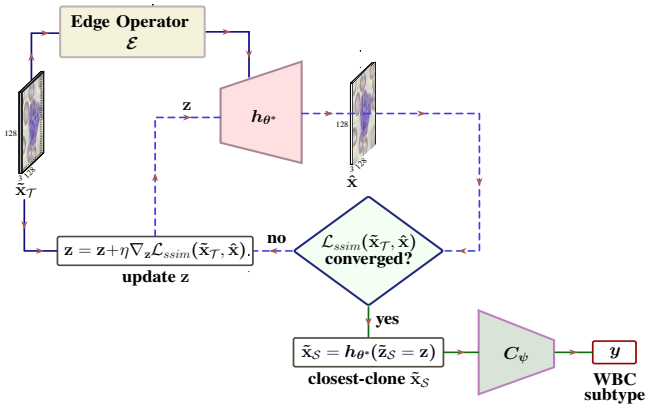


Figure 2. Latent Search procedure during inference with TIGDA. The latent vector $\mathbf{z}$ is initialized with a random sample drawn from $\mathcal{N}(0, 1)$. Iterations over the latent space $\mathbf{z}$ are performed to minimize the Structural Similarity loss $\mathcal{L}_{ssim}$ between the input target image $\tilde{\mathbf{x}}_\mathcal{T}$ and the predicted target image $\hat{\mathbf{x}}$, which is the output of the trained decoder (blue dotted lines). After convergence of $\mathcal{L}_{ssim}$ loss, the optimal latent vector $\tilde{\mathbf{z}}_\mathcal{S}$, generates the 'closest-clone' $\tilde{\mathbf{x}}_\mathcal{S}$ which is used to predict the class of $\tilde{\mathbf{x}}_\mathcal{T}$ using the classifier C_ψ trained on source samples.

B. Inference through Latent Search

Once the VAE is trained, given an image $\tilde{\mathbf{x}}_\mathcal{T}$ from the target distribution, the Latent Search algorithm searches for an optimal latent vector $\tilde{\mathbf{z}}_\mathcal{S}$ that generates its 'closest-clone' $\tilde{\mathbf{x}}_\mathcal{S}$ from $\mathcal{P}_S$. The search is performed by minimizing the SSIM loss $\mathcal{L}_{ssim}$ between the input target image $\tilde{\mathbf{x}}_\mathcal{T}$ and VAE reconstructed target image. The latent vector is optimized using a gradient-based optimization procedure, performed for K (a hyper-parameter) iterations over the latent space of the VAE for every target image. The gradient based optimization is implemented with Nesterov Accelerated Gradient method with a momentum of 0.5. Finally, the class for the input target image is assigned the same as the one given by the source classifier C_ψ on $\tilde{\mathbf{x}}_\mathcal{S}$. C_ψ is a VGG-16 or RestNet-50 classifier trained on source images. Note that our algorithm solves an optimization problem before predicting class for every input target image. However, since it involves only a forward-pass through a trained neural network (decoder h_{θ^*}), the time taken is only of the order of few milliseconds on standard CPUs. The complete algorithmic steps and the architectural details for TIGDA are given in the supplementary material.

V. Dataset Details

The datasets used in this study will be described in this section. Peripheral blood smear (PBS) consists primarily of three cell types – RBC (Red Blood Cell or erythrocyte), WBC

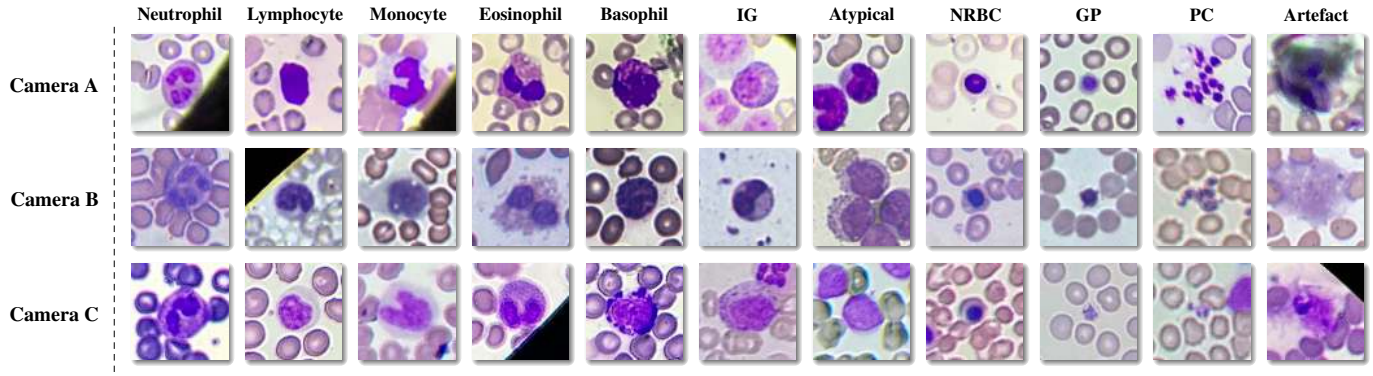


Figure 3. Samples of White Blood Cells and related microscopic images (categorized into 11 classes) taken from three different cameras A, B and C. (IG=Immature granulocytes, NRBC=Nucleated red blood cells, GP=Giant platelets, PC=Platelet clumps). It is to be noted that there are no visually distinctive features across cameras but it is easy for a human-pathologist to correctly classify despite camera changes. On the other hand, deep learning models fail to generalize across cameras.

Table I
NUMBER OF WHITE BLOOD CELLS AND RELATED MICROSCOPIC IMAGES FOR EACH SUBTYPE (CLASS) CAPTURED WITH THREE DIFFERENT CAMERAS A, B AND C. (NE=NEUTROPHIL, LY=LYMPHOCYTE, MO=MONOCYTE, EO=EOSINOPHIL, BA=BASOPHIL, IG=IMMATURE GRANULOCYTES, NRBC=NUCLEATED RED BLOOD CELLS, GP=GIANT PLATELETS, PC=PLATELET CLUMPS).

Camera	NE	LY	MO	EO	BA	IG	Atypical	NRBC	GP	PC	Artefact	Train/Test
A	3,885	1,507	2,224	2,076	65	863	984	651	486	138	2,550	10,849/4,580
B	2,045	1,840	612	373	67	1,073	2,257	97	918	796	1,437	7,997/3,518
C	85	43	144	85	12	323	861	321	303	11	16	1,548/656

(White Blood Cell or leukocyte) and platelet (or thrombocyte). Each of these primary classes have subclasses. The subclasses of WBCs are: neutrophil, lymphocyte, monocyte, eosinophil, basophil, immature granulocytes and atypical/blast cells. Apart from these, there are other types of cells and artefacts which can have appearance similar to leukocytes. These are – nucleated red blood cell (NRBC), large platelets, platelet clumps, and stain artefacts [1]. In the current study, we consider classification of 11 categories of which seven are subtypes of WBCs and rest four are NRBC, large platelets, platelet clumps, and stain artefacts (images shown in Figure 3). Data used in our experiments comprises images from the PBS slides processed after complete de-identification to remove all the patient information, including age and gender. These were collected from two large clinical laboratories in Bangalore, India. The internal ethics committee of the respective laboratories approved the study. The samples were collected retrospectively without prospective patient recruitment.

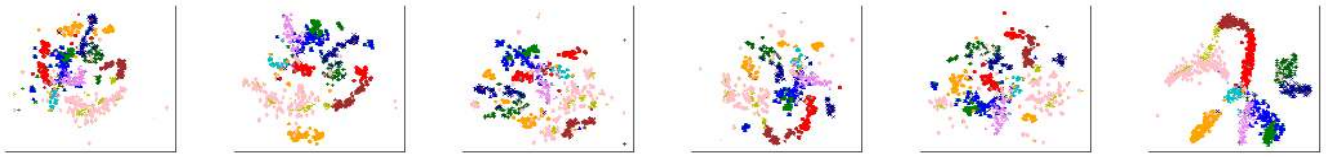
The hardware consists of the following components, (a) Optical system: Consists of an optical tube (40X or 100X Plan Achromat objective and 10X eyepiece) and Abbe Condenser with white LED source, (b) Camera: The system is built such that either a mobile phone or a USB camera can be fitted on top of the eyepiece with a 3D printed attachment, aligning the optical axis of the tube/eyepiece with the camera, (c) Hardware control: A small PCB designed to receive USB commands and drive motors and LED, (d) XYZ slide stage: The XYZ platform is built using commercially available low-cost ball screws and stepper motors, along with some machined parts [47]. The images used in this work are captured through 3 different cameras – One cell phone make (iPhone 6s) and two brands of USB camera (from e-con systems [48] and das-Cam [49]). All cameras had resolution of at least 13MP with varying hardware and optical designs that induce the domain shift. For example, econ camera has an AR1335 CMOS image sensor and lens with 1/3.2" optical format while das-Cam contains an OV13850 CMOS sensor with a lens of 1/3.06" form factor.

Images are collected only from the 'monolayer' region of the slides – where the red blood cells are just touching each other. This is the area of the slide which is typically used for manual analysis [1]. Slides prepared using varied staining types were used. The images are of size approximately 13MP, with a spatial resolution of around 5.5 pixels per micron. WBC and other similar looking cells (as described above) are localised in these images using a U-Net [50] based technique described in [15]. Each sample slides can potentially yield hundreds of unique WBC candidates. For annotation, we cropped 128×128 area around the WBCs identified by the extraction model. These cells are then presented to three different certified medical professionals for annotating into different subtypes, using an in-house web based annotation tool. There is usually a high degree (as high as 20%) of inter-observer variability in the data annotation process. Therefore, we use only those images where at least 2 out of 3 clinical pathologists agree on the class while the rest of the images are rejected. Table I describes the summary of the datasets named as A, B and C corresponding to three cameras used.

Table II

ACCURACY (MEAN ± STD%) VALUES FOR UDA TASKS ON WBC AND RELATED MICROSCOPIC IMAGES CAPTURED WITH THREE DIFFERENT CAMERAS A, B AND C. X→Y INDICATES MODEL TRAINED ON IMAGES FROM SOURCE CAMERA X AND TESTED ON IMAGES FROM TARGET CAMERA Y. RESULTS ARE REPORTED AS AN AVERAGE OVER FIVE INDEPENDENT RUNS USING VARIOUS STATE-OF-THE-ART UDA AND DOMAIN RANDOMIZATION METHODS. NOTE THAT WHILE ALL UDA METHODS PERFORM BETTER THAN THE SOURCE ONLY MODEL, TIGDA OFFERS THE BEST PERFORMANCE DESPITE NOT USING THE TARGET IMAGES.

	ResNet-50						VGG-16					
Models	A→B	A→C	B→A	B→C	C→A	C→B	A→B	A→C	B→A	B→C	C→A	C→B
Source Only	42.7±0.5	51.3±0.4	35.8±0.6	46.2±0.2	22.8±0.6	26.9±0.4	37.4±0.5	47.6±0.4	31.2±0.3	40.1±0.5	17.6±0.6	22.7±0.2
DR1 [38]	52.5±0.3	57.7±0.1	43.6±0.2	51.7±0.4	34.5±0.3	36.2±0.2	44.6±0.1	50.9±0.2	38.2±0.4	46.5±0.3	27.3±0.3	30.8±0.2
DR2 [39]	60.3±0.2	65.4±0.3	55.9±0.2	64.2±0.4	44.6±0.3	49.8±0.4	54.1±0.1	59.6±0.2	48.7±0.1	60.5±0.4	41.3±0.3	45.2±0.1
DR3 [40]	50.4±0.2	53.4±0.4	40.5±0.2	49.8±0.3	29.5±0.3	32.7±0.4	41.8±0.3	47.5±0.2	35.9±0.1	42.1±0.2	23.6±0.2	28.3±0.3
ADDA [16]	43.5±0.1	52.7±0.2	37.3±0.1	48.1±0.5	24.9±0.4	29.1±0.5	39.3±0.2	50.1±0.3	33.6±0.4	43.3±0.2	19.8±0.4	25.2±0.5
GTA [18]	56.2±0.4	66.3±0.5	48.1±0.2	56.7±0.6	35.5±0.4	37.8±0.1	52.6±0.7	62.1±0.3	41.9±0.6	50.7±0.3	30.1±0.1	33.7±0.6
TAT [21]	65.8±0.5	70.5±0.4	54.8±0.3	63.1±0.7	44.7±0.2	48.2±0.3	61.7±0.5	67.3±0.4	50.6±0.4	58.3±0.6	40.3±0.1	42.5±0.1
DIRT-T [20]	55.7±0.5	65.1±0.6	49.2±0.2	55.4±0.3	34.2±0.3	37.5±0.4	53.1±0.8	61.9±0.7	40.7±0.5	50.3±0.5	31.3±0.4	32.9±0.7
DAL [22]	64.7±0.2	69.4±0.3	56.3±0.2	62.7±0.4	43.5±0.1	47.5±0.5	60.8±0.2	66.5±0.5	51.8±0.4	59.1±0.3	39.7±0.1	41.1±0.2
CyCADA [27]	67.2±0.5	73.7±0.1	58.2±0.2	64.5±0.6	48.4±0.4	50.2±0.3	62.3±0.3	70.2±0.2	53.4±0.4	59.7±0.2	42.6±0.6	43.9±0.7
PixelDA [28]	65.9±0.2	71.8±0.7	59.1±0.8	66.2±0.5	47.8±0.4	50.6±0.5	61.5±0.3	68.4±0.4	54.6±0.7	58.8±0.6	41.3±0.6	42.5±0.4
SBADA-GAN [29]	66.3±0.2	70.5±0.2	60.3±0.3	65.6±0.4	46.4±0.7	51.1±0.1	62.7±0.6	67.9±0.8	53.8±0.7	58.7±0.2	42.7±0.4	44.6±0.7
I2IAdapt [30]	64.4±0.6	68.7±0.5	61.2±0.3	65.4±0.4	45.2±0.1	49.7±0.6	63.9±0.8	65.1±0.1	52.5±0.7	55.6±0.4	43.8±0.8	45.3±0.3
JAN [33]	49.6±0.2	58.2±0.5	43.3±0.2	54.7±0.4	30.2±0.7	35.4±0.8	43.5±0.6	54.2±0.4	39.1±0.3	47.5±0.3	26.3±0.4	31.4±0.6
MCD [34]	55.4±0.4	67.1±0.8	49.2±0.7	55.8±0.6	36.1±0.2	39.2±0.5	50.9±0.7	63.2±0.4	42.3±0.3	50.4±0.5	31.9±0.8	34.8±0.5
CAN [35]	67.8±0.4	71.3±0.5	63.4±0.5	65.4±0.3	47.3±0.2	51.2±0.4	61.9±0.8	68.1±0.3	54.6±0.6	59.3±0.4	40.9±0.2	45.7±0.8
TIGDA (Ours)	**76.2±0.3**	**80.1±0.4**	**72.3±0.5**	**74.8±0.6**	**53.5±0.4**	**56.2±0.3**	**71.8±0.5**	**76.7±0.2**	**63.2±0.5**	**68.6±0.7**	**50.8±0.2**	**55.1±0.4**



(a) ADDA (b) GTA (c) DIRT-T (d) TAT (e) DAL (f) TIGDA

Figure 4. t-SNE plots of features generated by ADDA [16], GTA [18], DIRT-T [20], TAT [21], DAL [22] and TIGDA on domain adaptation task A→C. We used different markers and different colors to denote 11 categories. It is seen that TIGDA offers better clustering as compared to the rest.

VI. EXPERIMENTS AND RESULTS

A. Benchmarking Experiments

In the first set of experiments, we benchmark performance of the baseline classifier with the following experiments: (a) Train and test on the same dataset type (A/B/C), (b) Train and test by combining images from all dataset types (A+B+C), (c) Train on one dataset and test on the other (all six combinations) with and without class balancing. The notation X→Y symbolizes training on a dataset X and testing on Y.

Table III lists the results of experiment (a) which establishes an upper bound on the performance and (b) where it is seen that the performance degrades when all images from all three datasets are combined. This is due to the existence of domain shift between the datasets that makes learning difficult even with supervision. Moreover, combining datasets is not possible in the UDA setting where the labels are not known for the target data. Results of experiment (c) are shown in Table IV where it is seen that the accuracy severely degrades when train and test sets are from different domains despite inducing an artificial class balance. The goal of UDA techniques is to improve the accuracies reported in Table IV.

B. Baseline Experiments

The first set of task is of classification across 11 classes with classifiers trained on one (source) dataset and tested on another (target) dataset. We report average classification accuracies with standard-deviation (averaged over five independent runs) with two backbone architectures for the source classifier: ResNet-50 and VGG-16. For all the UDA tasks, the VAE is trained with the entire source data and tested on the entire target data. Table II compares the performance of TIGDA with 12 SOTA UDA baselines, along with the accuracy without any UDA (called Source Only). It is seen that although all the UDA methods improve upon the Source Only performance, TIGDA offers the best performance despite not using any

Table III

BENCHMARKING A,B AND C DATASETS USING RESNET-50 CLASSIFIER WITH DIFFERENT TRAIN AND TEST SETS. IT IS SEEN THAT COMBINING ALL DATASETS MAKES LEARNING DIFFICULT BECAUSE OF DOMAIN SHIFT.

Measure	A→A	B→B	C→C	(A+B+C)→(A+B+C)
Train Acc.	98.6±0.1	99.3±0.2	100.0±0.0	98.7±0.2
Test Acc.	95.2±0.2	94.0±0.3	92.5±0.1	84.4±0.3

Table IV

ACCURACY ON RESNET-50 CLASSIFIERS FOR DIFFERENT ADAPTATION TASKS. IN THE SECOND ROW, ALL THE THREE DATASETS ARE MADE TO HAVE SAME SIZE BY RANDOMLY SUBSAMPLING THE DATASETS.

Measure	A→B	A→C	B→A	B→C	C→A	C→B
W/o Balance	42.7±0.5	51.3±0.4	35.8±0.6	46.2±0.2	22.8±0.6	26.9±0.4
With Balance	40.4±0.1	36.2±0.4	38.9±0.2	30.5±0.2	24.5±0.4	28.2±0.3

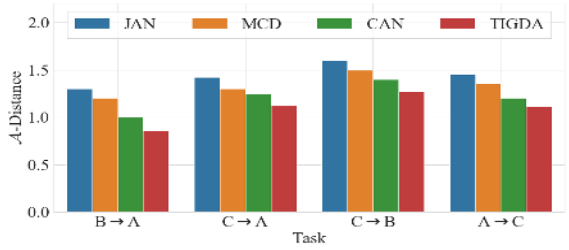


Figure 5. $\mathcal{A}$-Distance (lower is better) of JAN [33], MCD [34], CAN [35] and TIGDA.

data from the target distribution. The confusion matrix for a few methods is given in the Figure 2 of the Supplementary material. We also compare with three Domain Randomization (DR) techniques, DR1 [38], DR2 [39] and DR3 [40]. While DR provides performance boost, they have poorer performance as compared to TIGDA. This is because DR methods typically work well when the unseen target is within the scope of the class of random perturbations that are made on the source which is not the case always. In TIGDA on the other hand, every target image is made to resemble the source image through implicit sampling. Since VAE learns to sample from the entire source domain, the domain shift is implicitly reduced during inference without explicitly assuming any form for the shift. It is also observed that the performance of the classifier when trained and tested on single source domain (around 92-95% for all the datasets) do not degrade with TIGDA.

1) t-SNE: To further examine our hypothesis, in Figure 4 we depict the t-SNE [51] plots of features generated by adversarial based UDA methods (ADDA [16], GTA [18], DIRT-T [20], TAT [21] and DAL [22]) for the domain adaptation task A→C. For TIGDA, we plot the embeddings of the latent variable $\tilde{\mathbf{z}}_{\mathcal{S}}$ obtained through the LS on the target images. It is seen that the representation generated by the LS of TIGDA is more separated compared to those generated by adversarial training based UDA methods. A similar observation is made on the first two principal component plots of the latent representations (Please refer to Figure 1 in supplementary material).

2) $\mathcal{A}$-Distance: To ascertain the closeness of the 'closest-clones' obtained through the LS, to the source distribution, we compute the $\mathcal{A}$-distance [52], which is a measure of similarity between two probability distributions. Similar feature distributions will have lower $\mathcal{A}$-distance between them as compared to dissimilar feature distributions. $\mathcal{A}$-distance is given by $\hat{d}_A = 2(1 - 2\epsilon)$ where ϵ is the generalization error of a linear SVM classifier trained to discriminate between the source and target domains. Figure 5 displays $\hat{d}_A$ for the four domain adaptation tasks with JAN [33] features, MCD [34] features and CAN [35] features, respectively. In our case, $\hat{d}_A$ is measured between the latent vectors (produced by the Encoder of the VAE) of the source images and the latent vectors of the 'closest-clones' for target images obtained from Latent Search. We observe that $\hat{d}_A$ is smallest in our case as compared to other methods for all the tasks. This implies that the features obtained using TIGDA are transferable between the source and target domains, aiding better adaptation.

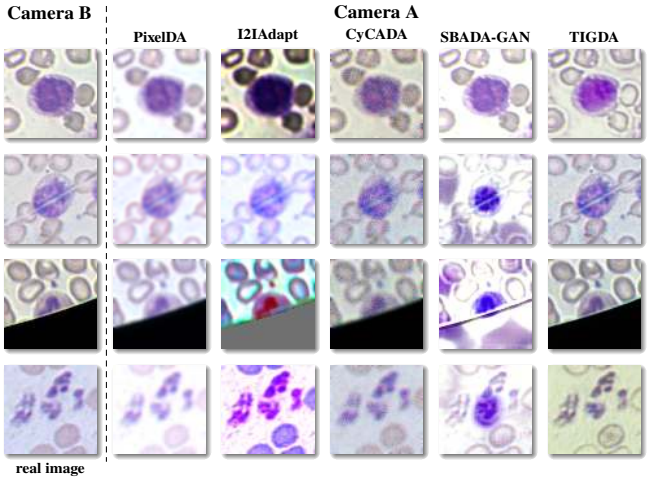


Figure 6. Translation of images from one domain (Camera B) to other (Camera A) using reconstruction based domain adaptation methods: PixelDA [28], I2IAdapt [30], CyCADA [27], SBADA-GAN [29]. In TIGDA, we depict the 'closest-clones' of Camera B (target) images in the Camera A (source) domain. It is seen that TIGDA preserves the edges, perceptual quality and structural details in the generated clones.

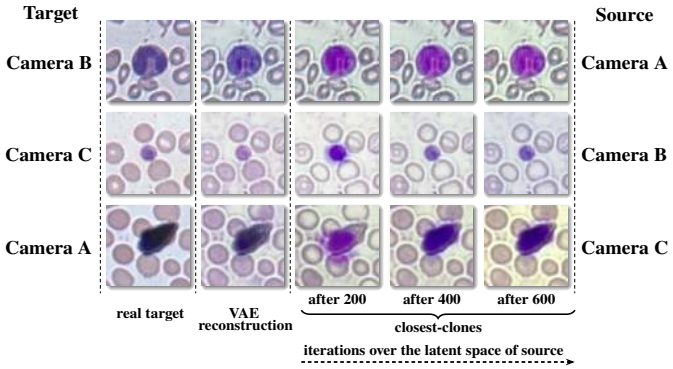


Figure 7. Illustration of Latent Search in TIDGA. VAE reconstructs images prior to LS. The closest-clones obtained after every 200 iterations are shown. A transformation is observed from the target to the source domain as the LS progresses.

3) Qualitative examination: To qualitatively examine the performance of the reconstruction-based methods, we plot the transformed target samples from (source) Camera B to (target) Camera A for different methods as shown in Figure 6. It is seen that I2IAdapt [30] and SBADA-GAN [29] are not able to capture fine subtleties of partially visible White Blood Cells in microscopic images that results in poor performance. PixelDA [28] and CyCADA [27] result in blurry images while TIGDA generated images are better where it is seen that the subtleties like edge information are well-preserved. In summary, we have demonstrated that TIGDA achieves better performance over the SOTA adversarial, divergence and reconstruction based UDA methods without any requirement for target images.

4) One-shot learning: Even though TIGDA does not utilize the target data during training, target image is used for LS during inference. Therefore, we also compare TIGDA with SOTA one-shot learning techniques in Table V. In one-shot learning methods, a single target image is used during training for adaptation. It is seen that TIGDA outperforms such

Table V
COMPARISON OF TIGDA WITH ONE-SHOT LEARNING METHODS.

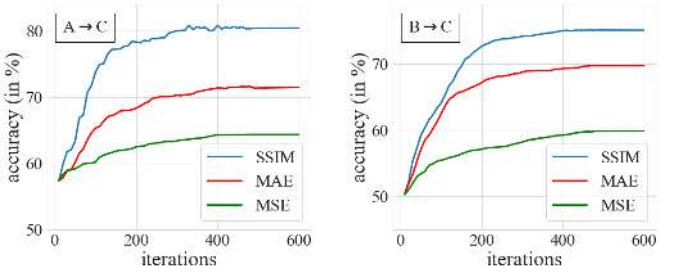
Method	A→B	C→B
ProtoNet [53]	61.9±0.1	49.6±0.3
MatchingNet [53]	57.6±0.2	43.7±0.1
DAPN [54]	68.9±0.2	51.9±0.2
DN4 [55]	55.4±0.1	44.6±0.2
FADA [56]	60.6±0.3	45.9±0.3
TIGDA (Ours)	76.2±0.3	56.2±0.3

techniques. This is because, in one-shot learning methods, the target image that is used for training is fixed which restricts the learnability. However in TIGDA, no target image is used during training but a fresh latent search is conducted on each input target image during inference.

C. Ablation studies

To examine the contributions made by each of the proposed components, we conduct several ablation experiments on TIGDA in this section.

1) Effect of number of iterations on LS: The inference of TIGDA involves a gradient-based optimization through the decoder network h_{θ^*} to generate the 'closest-clone' for a given target image. In Figure 7, we show the transformation of a few target images after every 200 iterations. It can be seen that as the number of iterations increase, the target images change their characteristics to move towards the source distribution. Quantitatively, we plot the accuracy as a function of number



(a) Inference on camera C microscopic images when the model is trained on camera A images.

(b) Inference on camera C microscopic images when the model is trained on camera B images.

Figure 8. Performance of gradient-based Latent Search during inference on target microscopic images for two domain adaptation tasks using different objective functions; MSE=Mean Squared Error, MAE=Mean Absolute Error, SSIM=Structural Similarity Index. It is seen that the loss saturates around 500-600 iterations.

of iterations in Figure 8 where it is seen that it saturates around 500-600 iterations. We thus used 600 iterations in all the previous experiments in Table II.

2) Effect of the Edge concatenation: As described earlier, the edge-map of the input image is concatenated with one of the layers of decoder both while training and inference. Figure 9b shows the quality of image generated after Latent Search when the model was trained without edge concatenation (wEc). It can be observed that edge information of the nucleus and surrounding cells is lost resulting in a blurry image.

Table VI
ABLATION OF DIFFERENT COMPONENTS OF TIGDA DURING TRAINING AND INFERENCE; EDGE, PERCEPTUAL LOSS $\mathcal{L}_p$ AND LATENT SEARCH (LS). ACCURACY (MEAN ± STD%) VALUES ARE REPORTED AS AN AVERAGE OVER FIVE INDEPENDENT RUNS FOR TWO TASKS.

Edge	$\mathcal{L}_p$	LS	A→B	B→C
			35.8±0.2	39.5±0.1
✓			39.7±0.4	42.2±0.3
	✓		38.9±0.5	43.4±0.3
		✓	50.2±0.3	52.8±0.2
✓	✓		43.7±0.2	46.9±0.5
	✓	✓	57.6±0.4	60.3±0.2
✓		✓	53.4±0.3	57.1±0.4
✓	✓	✓	76.2±0.3	74.8±0.6

Further, the accuracy drops to 57.6% if edge concatenation is removed from VAE for the task A→B as evident from Table VI, whereas the accuracy for TIGDA is 76.2% for the same task. Similarly, the accuracy drops to 60.3% for the task B→C without edge concatenation while it is 74.8% for TIGDA.

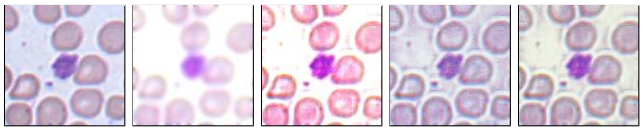


Figure 9. Ablation of TIGDA for task C→A. (wEc=without Edge concatenation, wPl=without Perceptual loss, wLS=without Latent Search). The best source-like features are observed in the image with all the components of TIGDA.

3) Effect of Perceptual loss $\mathcal{L}_p$: We have used a perceptual model P_ψ trained on source samples while training the VAE. Perceptual loss minimizes the Euclidean distance between the (perceptual) feature vectors of input and reconstructed source images. It measures image similarities more robustly than per-pixel losses (e.g., Mean squared error). It ensures that the VAE reconstructed image is semantically similar to the input. We can observe from Figure 9c that VAE reconstructed image without perceptual loss (wPl) during training, has different color and texture patterns from the real target image shown in Figure 9a. The finer background details are missing in Figure 9c. Such images will result in a poor latent space and the performance on target images will drop during inference. Table VI shows that the accuracy drops to 53.4% for the task A→B without perceptual loss while it is 76.2% for TIGDA that uses perceptual loss during training. Similarly the accuracy drops to 57.1% for the task B→C when perceptual loss was not employed during training but the accuracy on the same task is 74.8% with perceptual loss.

4) Effect of Latent Search and other Loss functions: To validate the importance of the Latent Search procedure, in Figure 9d we show the VAE reconstructed images without Latent Search for the target image shown in Figure 9a. Figure 9e shows the generated image after Latent Search for the task C→A. It is observed (empirically) that the 'closest-clone' obtained through TIGDA shown in Figure 9e is visually more closer to the source domain as compared to VAE reconstructed

image shown in Figure 9d. When no Latent Search is employed, the accuracy for the tasks A→B and B→C drops to 43.7% and 46.9% respectively as shown in Table VI. To affirm the usefulness of the choice of SSIM as loss for the Latent Search, we implemented Latent Search with three different losses, Mean Squared Error (MSE), Mean Absolute Error (MAE) and Structural Similarity Index (SSIM) loss and found that SSIM loss is the best performing among the three. SSIM loss compares pixels and their corresponding neighborhoods in two images, preserving the luminance, contrast and structure information. On the other hand, MSE or MAE measures only the absolute pixel differences rather than the structural differences. Figure 8a and 8b depict the outcome of these ablation studies where the superiority of the SSIM loss is seen over MSE and MAE for the tasks A→C and B→C respectively. Table VI summarizes all the ablation studies conducted on two domain adaptation tasks with different combinations of the components. It can be noted that the best performance is observed by utilizing all the three components: Edge concatenation, perceptual loss and Latent Search procedure. Thus, with all the aforementioned studies, we have demonstrated the utility of all the individual components used in TIGDA for UDA task on WBC classification.

5) Effect of other hyperparameters: In this section, we study the effect of four hyperparameters: (a) the window size for the SSIM loss used for Latent Search, (b) the position of the Edge-operator in the decoder network, (c) use of Skip connection as in [50] instead of edge concatenation, (d) number of source samples required to generate high-fidelity images using VAE. Figure 10(a) depicts the change in the performance for A→B with varying window sizes of SSIM. While the performance varies with different window sizes, the best accuracy is observed with the default choice of 11 that is used in all our experiments.

Next, in Figure 10(b), we vary the layer of the decoder to concatenate edges. It is seen that the performance is best at the penultimate layers since the edges are used only to reduce the blurriness of the generated image that occurs near the last few layers of the decoder. Providing the edge information at initial layers of the decoder, regularizes more than required, thus degrading the quality of the generated image.

To further quantify the effect of edge concatenation as a regularizer, we replace it with another type of spatial contiguity in the form of skip connections as in a segmentation network such as UNet [50]. We have used five different types of skip connections. Type-1 refers to no skip connection. Type-2 connects FC1 layer (Refer to Supplementary material for the names of the layers in the architecture) of the encoder with FC2 layer of the decoder network. Type-3 connects all the layers in the encoder with layers of corresponding dimensions in the decoder (like a U-Net). Type-4 connects Conv1 layer in the encoder with Conv9 layer of the decoder. Type-5 is combination of Type-2 and Type-4 skip connections. We observe in Figure 10(c) that having skip connection is better than not having it since it regularizes the network. Further, Type-4, that connects the initial layers of the encoder with final layers of the decoder, has the best performance. This can be explained by the fact that initial layers of the CNNs are known to extract edge-like features which is shown to enhance the performance in the given task. Connecting more layers as in Type-3 and Type-5 leads to over regularization and degrades the performance. However, explicit edge concatenation still provides the best performance.

In the final plot Figure 10(d), we report the Fréchet Inception Distance (FID) [57], that quantifies the quality of the generated data (lower the better) for any generative model, as a function of the number of source samples used to train the VAE. It is seen that with the increase in number of images for training VAE, the quality of generated images improve as shown by the FID values. Therefore, with about 10K samples, one can expect the VAE to sample high-fidelity source images.

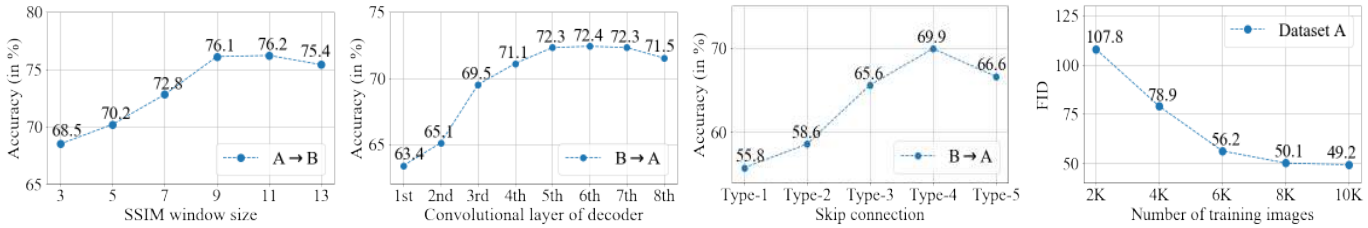
VII. TIGDA BEYOND PBS

In this section, we examine the effectiveness of the proposed method TIGDA on two datasets, Imaging Flow Cytometry [6] and Office-31 [62], apart from PBS. In Cytometry dataset, WBCs from whole blood samples were stained using a ImageStream-X MK II imaging flow cytometer. A three channel image is extracted with two bright-field (at wavelengths of 420 nm - 480 nm and 570 nm - 595 nm) and a dark-field channel. Four classes of WBCs are employed in this study: Eosinophil (1470 images), Neutrophil (4809 images), Lymphocyte (4570 images) and Monocyte (1239 images). The objective of this experiment is to examine if TIGDA can perform domain adaptation when the source is Cytometry data and the target is PBS and vice versa. Since Cytometry data doesn't have the notion of color, we take the grayscale version of the PBS dataset with a 60×60 central crop (in all the images) representing the nucleus. Figure 11 depicts a sample image from each class of the Cytometry dataset and the PBS dataset which apparently shows a significant domain shift. Office-31 [62], a publicly available standard dataset for UDA tasks (some sample images are given in the supplementary material), contains images from 31 common object types taken with three different imaging sources namely Dslr (D), Webcam (W) and Amazon (A). The objective of UDA is to adapt between these three domains.

Table VII lists the results of TIGDA along with some SOTA UDA methods for domain adaptation tasks on both the Office-31 and Cytometry datasets. It is seen that on Cytometry and gray-PBS datasets, TIGDA performs the best by significantly improving upon the Source Only model for gray-PBS→Cyto. and Cyto.→gray-PBS tasks. Whereas, on the Office-31 dataset, TIGDA's average performance is comparable (less than a percent) to the best SOTA method. All these experiments firmly demonstrate the effectiveness of TIGDA in UDA despite not using the target data during training.

VIII. CONCLUSION

In this work, we have considered the problem of domain shift occurring with the CNN-based classifiers for WBC classification. The performance of the existing deep learning based techniques is known to degrade with the change in camera characteristics. We cast the problem of performance degradation of WBC classifiers with the change in camera as



(a) Accu. vs. SSIM window sizes. (b) Accu. vs. Edge concat. position. (c) Accuracy vs Skip connections. (d) FID vs. no. of training images.

Figure 10. (a) Accuracy of TIGDA on task A→B by selecting different window sizes in SSIM during Latent Search (b) Performance of TIGDA when the edges of input images are concatenated with different convolutional layers in decoder h_θ (c) Performance of TIGDA when edge concatenation is replaced with different types of skip connections between encoder g_ϕ and decoder h_θ layers. Window size of 11 gives the best performance. For the same task, edge concatenation is better than skip connections. (d) FID of VAE generated images when TIGDA is trained on dataset A with different number of images ranging from 2,000 (2K) to 10,000 (10K).

Table VII
ACCURACY (MEAN ± STD%) VALUES FOR UDA TASKS ON OFFICE-31 AND IMAGING FLOW CYTOMETRY (CYTO.) AND GRAYSCALE PERIPHERAL BLOOD SMEAR (GRAY-PBS) WHITE BLOOD CELL DATASETS. RESULTS ARE REPORTED AS AN AVERAGE OVER FIVE INDEPENDENT RUNS USING VARIOUS SOTA UDA METHODS USING RESNET-50 CLASSIFIER. NOTE THAT WHILE ALL UDA METHODS PERFORM BETTER THAN THE SOURCE ONLY MODEL, TIGDA OFFERS SIGNIFICANT PERFORMANCE ENHANCEMENT DESPITE NOT USING THE TARGET IMAGES DURING TRAINING.

	Office-31							WBC		
Models	A→W	D→W	W→D	A→D	D→A	W→A	Avg	gray-PBS→Cyto.	Cyto.→gray-PBS	Avg
Source Only	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1	42.6±0.1	22.2±0.2	32.4
JAN [33]	85.4±0.3	97.4±0.2	99.8±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3	67.5±0.2	57.2±0.3	62.3
MADA [58]	90.0±0.2	97.4±0.1	99.6±0.1	87.8±0.2	70.3±0.4	66.3±0.1	85.2	73.3±0.2	61.8±0.3	67.5
SimNet [59]	88.6±0.5	98.2±0.2	99.7±0.2	85.3±0.3	73.4±0.8	71.8±0.6	86.2	76.4±0.2	66.8±0.2	71.6
GTA [18]	89.5±0.5	97.9±0.3	99.8±0.4	87.7±0.5	72.8±0.3	71.4±0.4	86.5	75.2±0.4	66.5±0.3	70.8
DAAA [60]	86.8±0.2	99.3±0.1	100.0±0.0	88.8±0.4	74.3±0.2	73.9±0.2	87.2	75.8±0.3	68.2±0.1	72.0
CDAN [61]	94.1±0.1	98.6±0.1	100.0±0.0	92.9±0.2	71.0±0.3	69.3±0.3	87.7	78.6±0.2	67.1±0.1	72.8
CAN [35]	94.5±0.3	99.1±0.2	99.8±0.2	95.0±0.3	78.0±0.3	77.0±0.3	90.6	79.4±0.3	68.9±0.2	74.1
TIGDA (Ours)	93.2±0.2	99.4±0.4	99.8±0.1	93.6±0.3	76.7±0.2	75.7±0.3	89.7	80.3±0.4	71.4±0.3	75.8

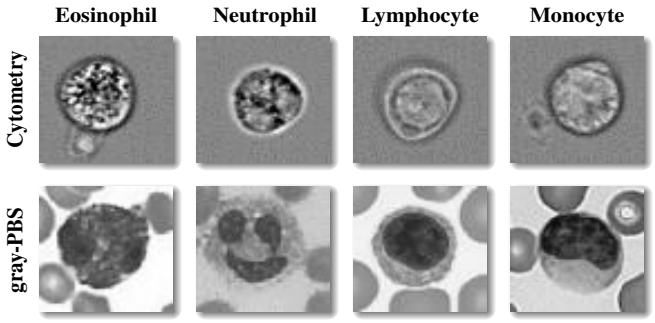


Figure 11. Imaging Flow Cytometry [6] and grayscale Peripheral Blood Smear (gray-PBS) White Blood Cell datasets.

that of Unsupervised Domain Adaptation (UDA) and propose a method that is devoid of need for access to the target data during training. We have demonstrated the efficacy of the proposed method for UDA with experiments on multiple datasets acquired under different settings. A few possible future directions can be: (i) extension of TIGDA for medical data beyond WBC, (ii) combining multiple sources for UDA.

IX. ACKNOWLEDGEMENTS

We sincerely thank the Associate Editor and the Anonymous Reviewers for their thoughtful comments that helped to significantly improve our paper. We also thank Maxim Lippeveld, Ghent University for his generous help in providing and navigating through the Cytometry dataset. We thank Sameer Ambekar and Aayush Tyagi for their help in experiments.

REFERENCES

[1] B. J. Bain, "A beginner's guide to blood cells," 2004.
[2] L. H. Lee, A. Mansoor, B. Wood, H. Nelson, D. Higa, and C. Naugler, "Performance of cellavision dm96 in leukocyte classification," *Journal of pathology informatics*, vol. 4, 2013.
[3] I. T. Young, "The classification of white blood cells," *IEEE Transactions on Biomedical Engineering*, no. 4, pp. 291–298, 1972.
[4] S. F. Bikhet, A. M. Darwish, H. A. Tolba, and S. I. Shaheen, "Segmentation and classification of white blood cells," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, vol. 4, pp. 2259–2261, IEEE, 2000.
[5] C. Hagwood, J. Bernal, M. Halter, and J. Elliott, "Evaluation of segmentation algorithms on cell populations using cdf curves," *IEEE transactions on medical imaging*, vol. 31, no. 2, pp. 380–390, 2011.
[6] M. Lippeveld, C. Knill, E. Ladlow, A. Fuller, L. J. Michaelis, Y. Saeys, A. Filby, and D. Peralta, "Classification of human white blood cells using machine learning for stain-free imaging flow cytometry," *Cytometry Part A*, 2019.
[7] C. L. Chen, A. Mahjoubfar, L.-C. Tai, I. K. Blaby, A. Huang, K. R. Niazi, and B. Jalali, "Deep learning in label-free cell classification," *Scientific reports*, vol. 6, p. 21471, 2016.
[8] F. Qin, N. Gao, Y. Peng, Z. Wu, S. Shen, and A. Grudtsin, "Fine-grained leukocyte classification with deep residual learning for microscopic images," *Computer methods and programs in biomedicine*, vol. 162, pp. 243–252, 2018.
[9] K. Sirinukunwattana, S. e Ahmed Raza, Y.-W. Tsang, D. R. Snead, I. A. Cree, and N. M. Rajpoot, "Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images.," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1196–1206, 2016.
[10] F. Mahmood, D. Borders, R. Chen, G. N. McKay, K. J. Salimian, A. Baras, and N. J. Durr, "Deep adversarial training for multi-organ nuclei segmentation in histopathology images," *IEEE transactions on medical imaging*, 2019.
[11] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[12] Q. Huang, W. Li, B. Zhang, Q. Li, R. Tao, and N. H. Lovell, "Blood cell classification based on hyperspectral imaging with modulated gabor and cnn," *IEEE journal of biomedical and health informatics*, 2019.

[13] T. S. Lee, "Image representation using 2d gabor wavelets," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[14] A. Kratz, H.-I. Bengtsson, J. E. Casey, J. M. Keefe, G. H. Beatrice, D. Y. Grzybek, K. B. Lewandrowski, and E. M. Van Cott, "Performance evaluation of the cellavision dm96 system: Wbc differentials by automated digital image analysis supported by an ann," *American journal of clinical pathology*, vol. 124, no. 5, pp. 770–781, 2005.

[15] D. Mundhra, B. Cheluvaraju, J. Rampure, and T. R. Dastidar, "Analyzing microscopic images of peripheral blood smear using deep learning," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pp. 178–185, Springer, 2017.

[16] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7167–7176, 2017.

[17] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," in *Domain Adaptation in Computer Vision Applications*, pp. 189–209, Springer, 2017.

[18] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8503–8512, 2018.

[19] P. Pandey, A. K. Tyagi, S. Ambekar, and P. AP, "Skin segmentation from nir images using unsupervised domain adaptation through generative latent search," *arXiv preprint arXiv:2006.08696*, 2020.

[20] R. Shu, H. H. Bui, H. Narui, and S. Ermon, "A dirt-t approach to unsupervised domain adaptation," *ArXiv:1802.08735*, 2018.

[21] H. Liu, M. Long, J. Wang, and M. Jordan, "Transferable adversarial training: A general approach to adapting deep classifiers," in *International Conference on Machine Learning*, pp. 4013–4022, 2019.

[22] X. Peng, Z. Huang, X. Sun, and K. Saenko, "Domain agnostic learning with disentangled representations," *ArXiv:1904.12347*, 2019.

[23] F. Mahmood, R. Chen, and N. J. Durr, "Unsupervised reverse domain adaptation for synthetic medical images via adversarial training," *IEEE transactions on medical imaging*, vol. 37, no. 12, pp. 2572–2581, 2018.

[24] M. Gadermayr, L. Gupta, V. Appel, P. Boor, B. M. Klinkhammer, and D. Merhof, "Generative adversarial networks for facilitating stain-independent supervised & unsupervised segmentation: A study on kidney histology," *IEEE transactions on medical imaging*, 2019.

[25] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*, pp. 850–865, Springer, 2016.

[26] T. Zhang, J. Cheng, H. Fu, Z. Gu, Y. Xiao, K. Zhou, S. Gao, R. Zheng, and J. Liu, "Noise adaptation generative adversarial network for medical image analysis," *IEEE Transactions on Medical Imaging*, 2019.

[27] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," *arXiv preprint arXiv:1711.03213*, 2017.

[28] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3722–3731, 2017.

[29] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo, "From source to target and back: symmetric bi-directional adaptive gan," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8099–8108, 2018.

[30] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4500–4509, 2018.

[31] G. van Tulder and M. de Bruijne, "Learning cross-modality representations from multi-modal images," *IEEE transactions on medical imaging*, vol. 38, no. 2, pp. 638–648, 2018.

[32] M. Goetz, C. Weber, F. Binczyk, J. Polanska, R. Tarnawski, B. Bobek-Billewicz, U. Koethe, J. Kleesiek, B. Stieltjes, and K. H. Maier-Hein, "Dalsa: domain adaptation for supervised learning from sparsely annotated mr images," *IEEE transactions on medical imaging*, vol. 35, no. 1, pp. 184–196, 2015.

[33] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2208–2217, JMLR. org, 2017.

[34] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3723–3732, 2018.

[35] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4893–4902, 2019.

[36] C. Pacheco and R. Vidal, "An unsupervised domain adaptation approach to classification of stem cell-derived cardiomyocytes," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 806–814, Springer, 2019.

[37] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30, IEEE, 2017.

[38] F. Mahmood, R. Chen, S. Sudarsky, D. Yu, and N. J. Durr, "Deep learning with cinematic rendering: fine-tuning deep neural networks using photorealistic medical images," *Physics in Medicine & Biology*, vol. 63, no. 18, p. 185012, 2018.

[39] D. Toth, S. Cimen, P. Ceccaldi, T. Kurzendorfer, K. Rhode, and P. Mountney, "Training deep networks on domain randomized synthetic x-ray data for cardiac interventions," 2018.

[40] S. Zakharov, W. Kehl, and S. Ilic, "Deceptionnet: Network-driven domain randomization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 532–541, 2019.

[41] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[42] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[43] J. Yang, N. C. Dvornek, F. Zhang, J. Chapiro, M. Lin, and J. S. Duncan, "Unsupervised domain adaptation via disentangled representations: Application to cross-modality liver segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 255–263, Springer, 2019.

[44] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2016.

[45] D. Mishra, S. Chaudhury, M. Sarkar, and A. S. Soin, "Ultrasound image enhancement using structure oriented adversarial network," *IEEE Signal Processing Letters*, vol. 25, no. 9, pp. 1349–1353, 2018.

[46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[47] T. R. Dastidar and R. Ethirajan, "Whole slide imaging system using deep learning-based automated focusing," *Biomedical Optics Express*, vol. 11, no. 1, pp. 480–491, 2020.

[48] https://www.e-consystems.com/, 2008.

[49] https://www.das-nano.com/das-cam/, 2008.

[50] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[51] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[52] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Advances in neural information processing systems*, pp. 137–144, 2007.

[53] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," *arXiv preprint arXiv:1904.04232*, 2019.

[54] A. Zhao, M. Ding, Z. Lu, T. Xiang, Y. Niu, J. Guan, J.-R. Wen, and P. Luo, "Domain-adaptive few-shot learning," *arXiv preprint arXiv:2003.08626*, 2020.

[55] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, and J. Luo, "Revisiting local descriptor based image-to-class measure for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7260–7268, 2019.

[56] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, pp. 6670–6680, 2017.

[57] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in neural information processing systems*, pp. 6626–6637, 2017.

[58] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[59] P. O. Pinheiro, "Unsupervised domain adaptation with similarity learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8004–8013, 2018.

[60] G. Kang, L. Zheng, Y. Yan, and Y. Yang, "Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 401–416, 2018.

[61] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, pp. 1640–1650, 2018.

[62] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*, pp. 213–226, Springer, 2010.

Target-Independent Domain Adaptation for WBC Classification using Generative Latent Search –Supplementary–

Prashant Pandey, Prathosh AP, Vinay Kyatham, Deepak Mishra and Tathagato Rai Dastidar

X. Proof for Lemma 1

:

Lemma 2. *If $\tilde{\mathbf{x}}_{\mathcal{S}} \in \mathcal{S}_n$ is the point such that $\mathfrak{D}\{\tilde{\mathbf{x}}_{\mathcal{T}}, \tilde{\mathbf{x}}_{\mathcal{S}}\} < \mathfrak{D}\{\tilde{\mathbf{x}}_{\mathcal{T}}, \mathbf{x}\}\ \forall\ \mathbf{x} \in \mathcal{S}_n$, then as $n \to \infty$, $\tilde{\mathbf{x}}_{\mathcal{S}}$ converges to $\tilde{\mathbf{x}}_{\mathcal{T}}$ with probability* 1.

Proof. Let $\mathbb{B}_r(\tilde{\mathbf{x}}_{\mathcal{T}})$ be a closed ball of radius r around $\tilde{\mathbf{x}}_{\mathcal{T}}$ under the metric $\mathfrak{D}$. That is, $\mathbb{B}_r(\tilde{\mathbf{x}}_{\mathcal{T}}) = \{\mathbf{x} : \mathfrak{D}\{\tilde{\mathbf{x}}_{\mathcal{T}}, \mathbf{x}\} \leq r\}$. Since $\mathfrak{X}$ is a separable metric space, $\forall r > 0$, $\mathbb{B}_r(\tilde{\mathbf{x}}_{\mathcal{T}})$ has non-zero probability measure [40]. That is,

$$\mathbf{Pr}\big(\mathbb{B}_r(\tilde{\mathbf{x}}_{\mathcal{T}})\big) \triangleq \int_{\mathbb{B}_r(\tilde{\mathbf{x}}_{\mathcal{T}})} \mathcal{P}_s(\mathbf{x})\ d\mathbf{x} > 0 \tag{14}$$

For any $\delta > 0$, the probability that none of the points in $\mathcal{S}_n$ are within the ball $\mathbb{B}_\delta(\tilde{\mathbf{x}}_{\mathcal{T}})$ of radius δ is given by:

$$\mathbf{Pr}\left[\min_{i=1,2..,n} \mathfrak{D}\{\mathbf{x}_i, \tilde{\mathbf{x}}_{\mathcal{T}}\} \geq \delta\right] = \big[1 - \mathbf{Pr}\big(\mathbb{B}_\delta(\tilde{\mathbf{x}}_{\mathcal{T}})\big)\big]^n \tag{15}$$

Therefore, the probability of $\tilde{\mathbf{x}}_{\mathcal{S}} \in \mathcal{S}_n$, lying within $\mathbb{B}_\delta(\tilde{\mathbf{x}}_{\mathcal{T}})$ is given by:

$$\mathbf{Pr}\Big[\tilde{\mathbf{x}}_{\mathcal{S}} \in \mathbb{B}_\delta(\tilde{\mathbf{x}}_{\mathcal{T}})\Big] = 1 - \big[1 - \mathbf{Pr}\big(\mathbb{B}_\delta(\tilde{\mathbf{x}}_{\mathcal{T}})\big)\big]^n \tag{16}$$

$$= 1 \quad as\ n \to \infty \tag{17}$$

Thus, given any $\delta > 0$, with probability 1, $\exists\ \tilde{\mathbf{x}}_{\mathcal{S}} \in \mathcal{S}_n$ that is within δ distance from $\tilde{\mathbf{x}}_{\mathcal{T}}$ as $n \to \infty$ □

XI. Algorithm for TIGDA

Algorithm 1 Target-Independent Generative Domain Adaptation (TIGDA)

Training VAE on source data

Input: Source dataset $\mathcal{S}_n = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, Number of source images n, Encoder g_ϕ, Decoder h_θ, Trained Perceptual Model P_ψ, Learning rate η, Batchsize B. **Output**: Optimal parameters ϕ^*, θ^*.

Initialize parameters ϕ, θ
repeat
  sample batch $\{\mathbf{x}_i\}$ from dataset $\mathcal{S}_n$, for $i = 1, ..., B$
  $\mu_{\mathbf{z}}^{(i)}, \sigma_{\mathbf{z}}^{(i)} \leftarrow g_\phi(\mathbf{x}_i)$
  sample $\mathbf{z}_i \sim \mathcal{N}(\mu_{\mathbf{z}}^{(i)}, {\sigma_{\mathbf{z}}^{(i)}}^2)$
  $\hat{\mathbf{x}}_i \leftarrow h_\theta(\mathbf{z}_i)$
  $\mathcal{L}_r \leftarrow \sum_{i=1}^{B} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$
  $\mathcal{L}_p \leftarrow \sum_{i=1}^{B} \|P_\psi(\mathbf{x}_i) - P_\psi(\hat{\mathbf{x}}_i)\|_2^2$
  $\mathcal{L}_g \leftarrow \mathcal{L}_r + \mathcal{L}_p + \sum_{i=1}^{B} \mathbb{D}_{KL}\left[\mathcal{N}(\mu_{\mathbf{z}}^{(i)}, {\sigma_{\mathbf{z}}^{(i)}}^2) \,||\, \mathcal{N}(0,1)\right]$
  $\mathcal{L}_h \leftarrow \mathcal{L}_r + \mathcal{L}_p$
  $\phi \leftarrow \phi + \eta \nabla_\phi \mathcal{L}_g$
  $\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{L}_h$
until convergence of ϕ, θ

Inference - Latent Search for target images

Input: Target image $\tilde{\mathbf{x}}_{\mathcal{T}}$, Trained decoder h_{θ^*}, Learning rate η. **Output**: 'closest-clone' $\tilde{\mathbf{x}}_{\mathcal{S}}$ for the target image $\tilde{\mathbf{x}}_{\mathcal{T}}$.

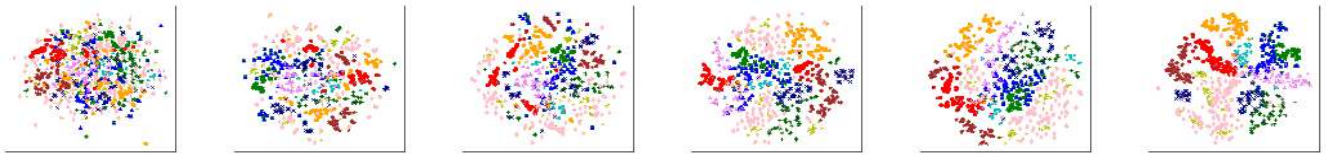
sample $\mathbf{z}$ from $\mathcal{N}(0,1)$
repeat
  $\mathcal{L}_{ssim} \leftarrow 1 - \text{SSIM}(\tilde{\mathbf{x}}_{\mathcal{T}}, h_{\theta^*}(\mathbf{z}))$
  $\mathbf{z} \leftarrow \mathbf{z} + \eta \nabla_{\mathbf{z}} \mathcal{L}_{ssim}$
until convergence of $\mathcal{L}_{ssim}$
$\tilde{\mathbf{z}}_{\mathcal{S}} \leftarrow \mathbf{z}$
$\tilde{\mathbf{x}}_{\mathcal{S}} \leftarrow h_{\theta^*}(\tilde{\mathbf{z}}_{\mathcal{S}})$

Table VIII
ENCODER ARCHITECTURE FOR THE VARIATIONAL AUTO-ENCODER (VAE) IN THE PROPOSED METHOD (TIGDA). CONVOLUTION KERNEL IS 3×3 AND FOR LEAKY RELU $\alpha = 0.2$.

Layer (type)	Output shape
encoder_input (InputLayer)	(128, 128, 3)
Conv1 (Convolution)	(128, 128, 128)
leakyReLU1 (Activation)	(128, 128, 128)
Conv2 (Convolution)	(64, 64, 128)
leakyReLU2 (Activation)	(64, 64, 128)
Conv3 (Convolution)	(32, 32, 128)
leakyReLU3 (Activation)	(32, 32, 128)
Conv4 (Convolution)	(16, 16, 128)
leakyReLU4 (Activation)	(16, 16, 128)
Conv5 (Convolution)	(8, 8, 128)
leakyReLU5 (Activation)	(8, 8, 128)
Conv6 (Convolution)	(4, 4, 128)
leakyReLU6 (Activation)	(4, 4, 128)
FC1 (Dense)	(1024)
Z (Dense)	(64)

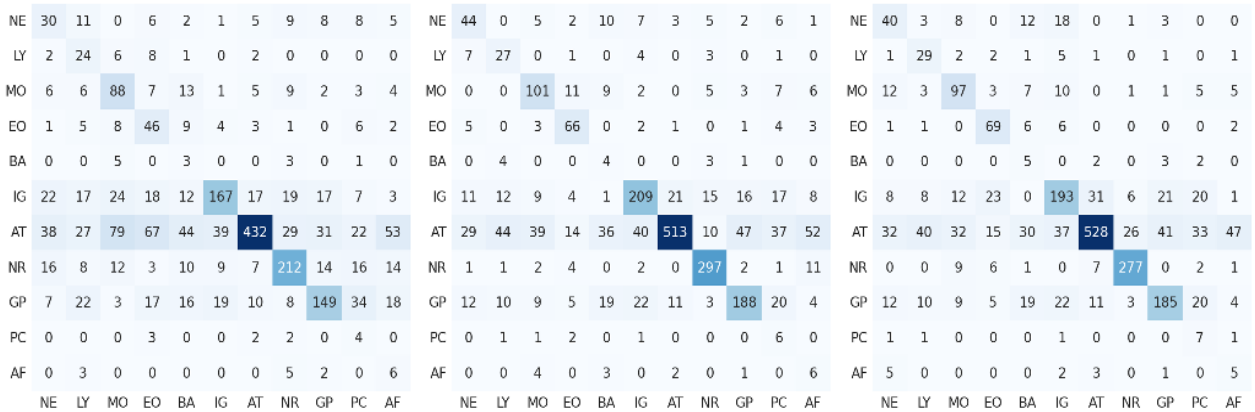
Table IX
DECODER ARCHITECTURE FOR THE VAE IN TIGDA. CONVOLUTION KERNEL IS 3×3 AND FOR LEAKY RELU $\alpha = 0.2$.

Layer (type)	Output shape
decoder_input (InputLayer)	(64)
FC2 (Dense)	(1024)
leakyReLU7 (Activation)	(1024)
FC3 (Dense)	(2048)
leakyReLU8 (Activation)	(2048)
Deconv1 (Deconvolution)	(4, 4, 128)
leakyReLU9 (Activation)	(4, 4, 128)
Deconv2 (Deconvolution)	(8, 8, 128)
leakyReLU10 (Activation)	(8, 8, 128)
Deconv3 (Deconvolution)	(16, 16, 128)
leakyReLU11 (Activation)	(16, 16, 128)
Deconv4 (Deconvolution)	(32, 32, 128)
leakyReLU12 (Activation)	(32, 32, 128)
Deconv5 (Deconvolution)	(64, 64, 128)
leakyReLU13 (Activation)	(64, 64, 128)
Deconv6 (Deconvolution)	(128, 128, 3)
tanh1 (Activation)	(128, 128, 3)
edge_input (InputLayer)	(128, 128, 6)
edge_concat (Concatenate)	(128, 128, 9)
Conv7 (Convolution)	(128, 128, 128)
leakyReLU14 (Activation)	(128, 128, 128)
Conv8 (Convolution)	(128, 128, 128)
leakyReLU15 (Activation)	(128, 128, 128)
Conv9 (Convolution)	(128, 128, 3)
tanh2 (Activation)	(128, 128, 3)

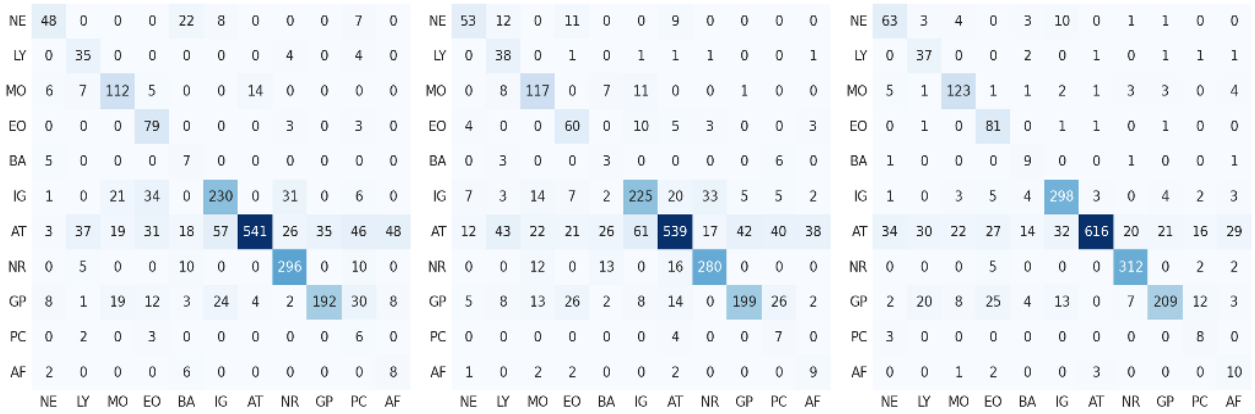


(a) ADDA (b) GTA (c) DIRT-T (d) TAT (e) DAL (f) TIGDA

Figure 12. PCA plots of the first two principal component using features generated by ADDA, GTA, DIRT-T, TAT, DAL and TIGDA on task A→C.



(a) ADDA (b) GTA (c) DIRT-T



(d) TAT (e) DAL (f) TIGDA

Figure 13. Confusion Matrices for ADDA, GTA, DIRT-T, TAT, DAL and TIGDA on task A→C. Classes are Neutrophil (NE), Lymphocyte (LY), Monocyte (MO), Eosinophil (EO), Basophil (BA), Immature granulocytes (IG), Atypical (AT), Nucleated red blood cells (NR), Giant platelets (GP), Platelet clumps (PC), Artefact (AF).

Figure 14. Samples from the Office-31 dataset from the three sources, Amazon, Dslr and Webcam.