# Deep Domain Adaptation: A Sim2Real Neural Approach for Improving Eye-Tracking Systems

VIET DUNG NGUYEN, Rochester Institute of Technology, USA
REYNOLD BAILEY, Rochester Institute of Technology, USA
GABRIEL J. DIAZ, Rochester Institute of Technology, USA
CHENGYI MA, Rochester Institute of Technology, USA
ALEXANDER FIX, Meta Reality Labs, USA
ALEXANDER ORORBIA, Rochester Institute of Technology, USA

Eye image segmentation is a critical step in eye tracking that has great influence over the final gaze estimate. Segmentation models trained using supervised machine learning can excel at this task, their effectiveness is determined by the degree of overlap between the narrow distributions of image properties defined by the target dataset and highly specific training datasets, of which there are few. Attempts to broaden the distribution of existing eye image datasets through the inclusion of synthetic eye images have found that a model trained on synthetic images will often fail to generalize back to real-world eye images. In remedy, we use dimensionality-reduction techniques to measure the overlap between the target eye images and synthetic training data, and to prune the training dataset in a manner that maximizes distribution overlap. We demonstrate that our methods result in robust, improved performance when tackling the discrepancy between simulation and real-world data samples.

CCS Concepts: • **Computing methodologies** → **Image segmentation**; *Neural networks*; *Learning latent representations*.

Additional Key Words and Phrases: Eye-tracking, Domain adaptation, Eye segmentation, Generative modeling, Deep learning

**ACM Reference Format:**
Viet Dung Nguyen, Reynold Bailey, Gabriel J. Diaz, Chengyi Ma, Alexander Fix, and Alexander Ororbia. 2024. Deep Domain Adaptation: A Sim2Real Neural Approach for Improving Eye-Tracking Systems. *Proc. ACM Comput. Graph. Interact. Tech.* 7, 2, Article 25 (June 2024), 17 pages. https://doi.org/10.1145/3654703

## 1 INTRODUCTION

Research in semantic segmentation has a wide range of applications, including autonomous vehicles, medical image analysis, and virtual reality [Minaee et al. 2020]. In the context of eye-tracking, the ability to accurately segment the eye's features provides great utility for the task of gaze estimation. Most modern approaches to eye-tracking rely on different segmented features of the eye, including the iris or pupil centroid or boundary [Ghosh et al. 2021]. For instance, some schemes for estimating and tracking gaze dynamics requires access to iris features uncovered through the use of its texture

Authors' addresses: Viet Dung Nguyen, vn1747@rit.edu, Rochester Institute of Technology, Rochester, New York, USA, 14623; Reynold Bailey, rjbvcs@rit.edu, Rochester Institute of Technology, Rochester, New York, USA, 14623; Gabriel J. Diaz, Gabriel.Diaz@rit.edu, Rochester Institute of Technology, Rochester, New York, USA, 14623; Chengyi Ma, cxm3593@rit.edu, Rochester Institute of Technology, Rochester, New York, USA, 14623; Alexander Fix, alexander.fix@meta.com, Meta Reality Labs, Redmond, Washington, USA, 98052; Alexander Ororbia, ago@cs.rit.edu, Rochester Institute of Technology, Rochester, New York, USA, 14623.


ACM 2577-6193/2024/6-ART25
https://doi.org/10.1145/3654703

and velocity [Pelz and Hansen 2017] while others do so by extracting them according to color thresholding and ellipse fitting [Shah and Ross 2009], multi-grid search via gradient ascent, and 2-D Gabor filters [Daugman 1993]. In addition, segmenting different parts of the eye simultaneously enables us to perform center localization, elliptical contour estimation, and blink detection [Yiu et al. 2019]. Ultimately, eye segmentation serves as an essential component of the general modeling toolbox for the eye-tracking community at large.

In general, approaches to eye segmentation [Chantapakul et al. 2019; Chaudhary et al. 2019; Kothari et al. 2022, 2021; Perry and Fernandez 2019; Stember et al. 2019] leverage deep convolutional neural networks [Krizhevsky et al. 2012; LeCun et al. 1995, 2015] (CNNs) and, consequently, require large eye datasets in order to train these neural models effectively. The requisite datasets can be collected by recording synthetic information from simulations [Nair et al. 2020] or from human subjects directly [Garbin et al. 2019; Zhang et al. 2015]. Although, real-world eye datasets, such as OpenEDS [Garbin et al. 2019] or MPIIGaze [Zhang et al. 2015], provide invaluable samples of data/images to train the CNN models on, constructing such datasets requires a great deal of human annotation effort (high labeling burden) as well as introduces potential human subject image privacy issues. In contrast, synthetic eye datasets circumvent these issues, reducing the data collection effort inherent to working with actual human participants as well as manual labeling work needed to generate ground truth segmentation masks [Nair et al. 2020]. As a result, generating datasets of synthetic data samples offers the potential to train powerful eye-tracking CNN-based systems at greatly reduced overall cost.

Despite the promise that synthetic data brings with it, a major issue emerges – due to the imperfections underlying computer simulation and 3D graphics models, a "reality gap" or mismatch exists between the synthetic data produced by a simulated environment (i.e. synthetic eye images) and the real world [Kaspar et al. 2020; Tobin et al. 2017] (i.e. images of real eyes). Fundamentally, this mismatch (known as the *sim2real* problem) is caused by a domain gap or domain distribution shift which results from a violation of the *independent and identically distributed (iid)* assumption that drives much of modern-day machine learning; the simulator represents the training distribution of the eye-tracking CNN system and the real-world represents a test distribution that the simulator can only at best approximate. This distributional mismatch results in degraded test-time generalization ability when the neural model is trained using data from the simulator but evaluated in the real world. For example, past work has suggested that training eye-tracking CNN models with multiple eye datasets potentially degrades their segmentation performance for a particular within-domain dataset [Kothari et al. 2022]. In this work, we will address this *sim2real* problem in order to improve the accuracy of the segmentation ability of neural eye-tracking models. Specifically, our research will improve the performance of a segmentation neural network on an eye dataset, consisting of real-world image patterns, training it using synthetic eye datasets and only a small number of real images. In this work, we will utilize the OpenEDS eye segmentation dataset [Garbin et al. 2019] as the real-world dataset and RITEyes as the synthetic dataset [Nair et al. 2020].

The key contributions of this work are as follows: **1)** in closing the domain gap between synthetically generated eye images and real eye images, our approach will ensure that the eye-tracking CNN model is trained with a large number of synthetically generated images in proportion to real-world images by leveraging a learned neural distance model, resulting in little to no degradation of segmentation performance on the real-world test dataset, and **2)** empirically, we will demonstrate that our scheme results in overall higher generalization performance, with respect to mean intersection over union (Jaccard Index) [Kosub 2019], compared to baseline models trained on synthetic images only.

## 2 RELATED WORK

### 2.1 Segmentation through Deep Learning

A critical component of an image segmentation system is the segmentation network. Concretely, a segmentation network is either a parameterized multi-layer perceptron (MLP) [Haykin 1998] or convolution neural network (CNN) [LeCun et al. 1995, 2015]. In our context, the segmentation network takes in an eye image and produces a classification of each pixel (as 'pupil', 'iris', 'sclera', or 'background/other'); this is often referred to as the segmentation map [Ronneberger et al. 2015]. Note that a segmentation network's input and output have the same width and height dimension, and this network is normally based on the U-net architecture [Ronneberger et al. 2015]. Research in eye tracking further extends the segmentation network form/design to better fit within the eye tracking context, introducing additional task-specific objective functions, e.g., as in RITnet [Chaudhary et al. 2019] or Ellseg [Kothari et al. 2021].

*U-net Architecture.* As mentioned above, the U-net architecture is a popular method for generating an image segmentation [Ronneberger et al. 2015]. It is designed to predict the probability of multiple segmentation classes that each pixel within the image could fall under. The U-net encoder-decoder neural architecture specifically designs the encoder such that each layer has a synaptic skip connection to the corresponding layer in the decoder. This skip connection involves the concatenation of the encoder layer output with the upsampled feature from its same-level decoder layer. The synaptic skip connections provide contextual information with respect to the current layer's image resolution, allowing them to consider the context from the macro to micro-features of the image itself [Ronneberger et al. 2015]. This makes the U-net architecture quite effective in, and appropriate for tasks related to image segmentation.

*RITnet.* Specific to the context of eye-tracking, Chaudhary et al. developed an efficient real-time eye segmentation model known as RITnet [Chaudhary et al. 2019], which utilizes a U-net architecture in tandem with additional objective functions that focus on segmenting particular eye regions. The first loss function in RITnet is the generalized dice loss (GDL), which combines the weighted dice score (this measures the overlap of the prediction and ground truth coefficient) with the cross-entropy loss; this pairing results in improved stability with respect to the cross-entropy objective. The second loss is the boundary aware loss (BAL), which weights the loss in each pixel by its distance to the nearest edge. To achieve this, a mask is computed by dilating the edge using the Canny edge detector[Canny 1986]; the mask is then used to weight the original cross-entropy loss, thus maximizing the correct pixel near each boundary. Finally, the RITnet framework introduces the surface loss, which scales the loss value at each pixel with respect to the pixel's distance to the boundary of the corresponding segmentation class. Specifically, for each segmentation class, a heat map of distance is computed by assigning to each pixel the relative distance to the nearest boundary of the corresponding class. The final surface loss is achieved by averaging the product of the network segmentation output and the surface heat map for each segmentation class.

### 2.2 Generative Adversarial Learning in Domain Transfer

This work's approach to improving segmentation performance will rely on synthetic data and, in order to ensure the synthetic data is useful, we will craft a scheme that will make the (in-) distribution of the training data closer to the real (out-) data distribution of real eye images. More specifically, our approach could be likened to a refinement process that takes in output from an eye simulator (e.g. synthetic images produced by Blender) and modifies it to produce images that are closer to the distribution of the real images. One possible way that we could implement this refinement is through a generative method, such as histogram matching [Tu and Dong 2013; Yaras

et al. 2021] where the pixel values in the source image are adjusted so that the histogram of the source and target images match one another. However, changing the features directly, such as pixel values, to match the raw image statistics, e.g., feature histograms, results in a difficult and complex image distributional modeling problem. Additionally, histogram matching can potentially introduce noise to the output image, increase the contrast, or distort the structure of the images [Mustafa and Kader 2018]. In contrast to generative methodology based on statistic matching, generative approaches based on unsupervised deep neural networks, specifically the generative adversarial network (GAN) [Goodfellow et al. 2014a] and variants such as the CycleGAN [Zhu et al. 2017], offer a parameterized means of mapping from source in-distribution to a target dataset (out-) distribution. Given the modeling flexibility afforded by neural models, our refinement process will leverage and build on the GAN as a central component.

*Generative Adversarial Network (GAN).* A GAN essentially consists of a "generator" and a "discriminator" neural network that work together to perform unsupervised image generation. The generator takes in as input noise, i.e., a noisy latent vector, and outputs a synthesized image pattern that looks similar to those in the desired image space. The discriminator specifically tries to classify whether an image is the output of the generator (fake) or the real image (real). The objective of the generator is to generate images that are plausible enough so as to reduce the accuracy of (or "fool") the discriminator. The objective of the discriminator $D$ is to maximize the binary cross-entropy loss for data coming from both the real image domain and the generated image domain.

*CycleGAN.* The CycleGAN [Zhu et al. 2017], which is an extension of the basic GAN, is built with the goal of facilitating image translation across different (input) data domains. Given an image from one data domain, the CycleGAN works to output an image that resembles images from another (target) data domain. This model is trained using two loss functions: the standard adversarial objective of the original GAN (in order to generate meaningful images) and a "cycle consistency" loss that maximizes the domain similarity of the model's generated image space. The cycle consistency loss helps to guide the CycleGAN's generator to map the source image to the desired general features in the target domain. Later efforts related to CycleGAN introduced the identity loss [Liu 2022; Taigman et al. 2017], which was further shown to stabilize the image-to-image translation process.

*Domain Adversarial Neural Network (DANN).* Another form of adversarial-based domain adaptation is based on the DANN model. DANN [Ganin et al. 2017] is a classification model that integrates an additional domain classifier trained under an adversarial process. In essence, training a DANN is similar to training a GAN given that it has a class predictor which tries to maximize the accuracy of the prediction of two domains while another domain classifier tries to distinguish between the two domains based on the bottleneck latent representation produced by the system's encoder. However, in the DANN, we do not have access to the gradient for training the encoder in order to maximize the domain classifier loss. Therefore, a "reverse gradient" layer is used for the latent embedding output before going into the domain classifier layers. Mechanically, a reverse gradient layer is simply an identity function that operates within the backward propagation process (for computing parameter gradients) that further negates the resulting estimated gradient values.

### 2.3 Metric Learning

The measurement of the distribution shift among datasets can provide useful information for closing the aforementioned reality gap, e.g., one can measure the domain shift by computing the maximum mean discrepancy (MMD) integral probability metric between two datasets [Gretton et al. 2012], minimizing this metric might potentially reduce the gap between datasets. To achieve

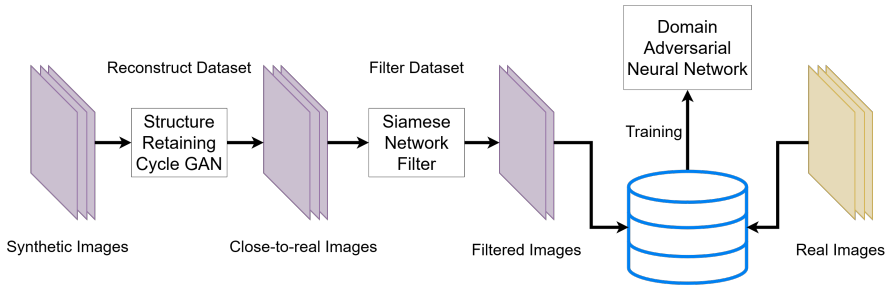


Fig. 1. Overall process diagram of our proposed computational system for image segmentation. The synthetic images are first refined/processed using our novel Structure Retaining CycleGAN, then filtered by our Siamese Network that considers the distance between the latent representations of real and synthetic images, and finally placed into a training set that is used for training our adapted domain adversarial neural network.

this, normally, a function parameterized by a deep neural network is learned in a process known as metric learning. In the context of this work, we will utilize metric learning in our refinement process, e.g., such as with a Siamese Network [Koch et al. 2015], in order to quantitatively measure the degree of distribution shift between the synthetic samples we produce and the samples within real eye datasets.

*Siamese Network.* The Siamese Network [Koch et al. 2015] consists of two identical deep neural networks that specifically share the same set of weights and structure (the structure and parameters are "tied"). The purpose of the overall model is to output the representation of two different objects such that we may compute the distance between these objects in terms of their corresponding projections in a latent space, using a distance function such as the L2 (Euclidean) distance. Overall, learning in a Siamese Network is similar to learning an albeit complex distance function. Generally, a Siamese Network is trained to minimize a contrastive loss [Hadsell et al. 2006; Melekhov et al. 2016] which penalizes the system for any deviation in its predicted distance values from a chosen distance measurement. Other loss functions used for training include the triplet loss and the binary cross entropy loss [Dong and Shen 2018; Koch et al. 2015; Nanni et al. 2021].

## 3 METHODS

To address the *sim2real* problem through domain adaptation, we develop and study the following: **1)** modifying the synthetic dataset such that it is closer to the real one so that the neural system benefits from the familiarity of the data at test time, and **2)** modifying the neural system to be capable of generalizing among different domains. Based on these notions, we propose a multi-step approach to the problem by utilizing image-to-image transfer (Section 3.1), dataset filtering (Section 3.2), and a domain generalization feature-based network (Section 3.3). The overall architecture, as shown in Fig. 1, first involves the implementation of our proposed *Structure Retaining CycleGAN*, which is a generalization of the CycleGAN [Zhu et al. 2017] that focuses on reconstructing the synthetic eye images under the constraint of matching the distribution of the real eye images. Next, we design a Siamese Network [Koch et al. 2015] for filtering out poorly-reconstructed images (i.e., a learned form of dataset pruning). Finally, we employ a model adapted from the domain-adversarial neural network structure [Ganin et al. 2017] which we will demonstrate has the ability to perform well across multiple domains. All project code is available at https://github.com/PerForm-Lab-RIT/domain-adaptation-eye-tracking.

### 3.1 Structure Retaining CycleGAN

In the context of image generation/creation, a CycleGAN can be used to perform domain transfer such that images in the transferred domain are as diverse and as close to the target domain as possible [Heusel et al. 2017; Lucic et al. 2017; Salimans et al. 2016; Zhu et al. 2017]). In the context of domain adaptation, we also need to perform transferring within the label domain so as to ensure that the transferred labels match, i.e., the transferred pupil segmentation matches the exact pupil segmentation in the transferred eye image. The GeomaskGAN [Lu et al. 2022] model uses a double input architecture which takes both the eye image and the segmentation label as input while performing the transfer. As an alternative, we propose another model that tries to preserve the structure of the eyes that avoids the need to transfer the segmentation label. We can achieve this by reconstructing images that have the same segmentation map structure as the original eye image. This inspiration comes from the intuition that there exists perceptually indistinguishable "noise", e.g., noise used to perturb an image in adversarial attack [Goodfellow et al. 2014b], in the real data distribution, such that the model, when trained on the synthetic distribution, has reduced performance when inferred on this real distribution. This problem serves as the basis for a divergence in the pixel distribution between a synthetic image and a real one.

*Problem Formulation.* Formally, we want to learn a function that maps an image from the synthetic eye domain $\mathcal{S} = \{\mathbf{s}^i\}_{i=1}^M$ to the real eye domain $\mathcal{R} = \{\mathbf{r}^i\}_{i=1}^N$ given the segmentation mask sets $\mathcal{M}_\mathcal{S} = \{\mathbf{m}_s^i\}_{i=1}^M$ and $\mathcal{M}_\mathcal{R} = \{\mathbf{m}_r^i\}_{i=1}^N$. We want to build the mapping function from domain $\mathcal{S}$ to domain $\mathcal{R}$ such that the transferred image $\mathbf{t}_{sr}^i \in \mathcal{T}_\mathcal{R}$ has the same original eye segmentation map $\mathbf{m}_s^i$ as the original image $\mathbf{s}^i$ and, furthermore, similar color features for each segmentation class $\mathbf{m}_r^i$ within the target domain image $\mathbf{r}^i$.

Similar to CycleGAN-like architectures [Heusel et al. 2017; Lucic et al. 2017; Salimans et al. 2016; Zhu et al. 2017], our model architecture contains two separate image generators (under an encode-decoder setup) and two discriminators. Each generator maps an image from one domain to the other while each discriminator distinguishes the domain for each generated image. Particularly, we define two generators as $G_{\mathcal{SR}} : \mathcal{S} \rightarrow \mathcal{T}_\mathcal{R} \approx \mathcal{R}$ and $G_{\mathcal{RS}} : \mathcal{R} \rightarrow \mathcal{T}_\mathcal{S} \approx \mathcal{S}$, and we define two discriminators as $G_\mathcal{S} : \mathcal{T}_\mathcal{S} \cup \mathcal{S} \rightarrow \mathbb{R} = \{0, 1\}$ and $G_\mathcal{R} : \mathcal{T}_\mathcal{R} \cup \mathcal{R} \rightarrow \mathbb{R} = \{0, 1\}$ which output 0 if the image sample comes from the generator and 1 if the image sample comes from $\mathcal{S}$ or $\mathcal{R}$.

*Adversarial Loss.* In order to train the above model, we first employ the adversarial loss [Goodfellow et al. 2014a], similar to what has been done in the CycleGAN literature [Zhu et al. 2017]. Particularly, this objective function encourages the generator to produce images that are closer to the target domain while the discriminator must distinguish between images that actually come from the source distribution or those produced by the generator, i.e, $\min_G \max_D \mathcal{L}_{adv}$. The objective function for $G_{\mathcal{SR}}$ and $D_\mathcal{R}$ is defined as:

$$\mathcal{L}_{adv}(G_{\mathcal{SR}}, D_\mathcal{R}) = \mathbb{E}_{\mathbf{r}\sim\mathcal{R}}[\log D_\mathcal{R}(\mathbf{r})] + \mathbb{E}_{\mathbf{s}\sim\mathcal{S}}[1 - \log D_\mathcal{R}(G_{\mathcal{SR}}(\mathbf{s}))]. \tag{1}$$

*Cycle Consistency Loss.* As mentioned before, this loss guides a network to learn a mapping function where recovered images are likely to closely match the original images [Zhu et al. 2017]. Particularly, we encourage the recovery of an image after translating it to another domain and back to the original domain. Similar to the original CycleGAN, we use the mean absolute error in order to compute the loss between the image before and after the domain transfer as follows:

$$\mathcal{L}_{cyc}(G_{\mathcal{SR}}, G_{\mathcal{RS}}) = \mathbb{E}_{\mathbf{s}\sim\mathcal{S}}[\| G_{\mathcal{RS}}(G_{\mathcal{SR}}(\mathbf{s})) - \mathbf{s} \|_1] + \mathbb{E}_{\mathbf{r}\sim\mathcal{R}}[\| G_{\mathcal{SR}}(G_{\mathcal{RS}}(\mathbf{r})) - \mathbf{r} \|_1]. \tag{2}$$

*Identity Loss.* This loss is often used in image-to-image translation problems in order to ensure that the color and tint of the translated image are as close to the original image as possible [Liu 2022; Taigman et al. 2017]. Furthermore, the identity loss states that a generator for the target

domain, when receiving an image from the same domain, must produce the image in the same domain, i.e, $G_{\mathcal{SR}}(\mathbf{r}) \approx \mathbf{r}$. The identity loss, in our context, is formulated below as follows:

$$\mathcal{L}_{id}(G_{\mathcal{SR}}, G_{\mathcal{RS}}) = \mathbb{E}_{\mathbf{s}\sim\mathcal{S}}[\parallel G_{\mathcal{RS}}(\mathbf{s}) - \mathbf{s} \parallel_1] + \mathbb{E}_{\mathbf{r}\sim\mathcal{R}}[\parallel G_{\mathcal{SR}}(\mathbf{r}) - \mathbf{r} \parallel_1]. \tag{3}$$

*Edge Retaining Loss.* Note that the cycle consistency objective function does not guarantee that the transferred image in another (target) domain has the same segmentation structure (one can see an incorrect mapping in the center image of Fig. 2). To overcome this problem, we propose that the structure of the image may be retained by keeping the edges of the image fixed throughout the translation process. In order to achieve this, we propose that the edge features of the original image should be as close as possible to the edge features of the translated image as well as the recovered image. For example, in the translation from domain $\mathcal{S}$ to domain $\mathcal{R}$, the edge features among the original image $\mathbf{s} \sim \mathcal{S}$, the translated image $G_{\mathcal{SR}}(\mathbf{s})$, and the recovered image $G_{\mathcal{RS}}(G_{\mathcal{SR}}(\mathbf{s}))$ should be approximately equal. In support of this, we implemented the Sobel filter [Kanopoulos et al. 1988] (denoted as $\mathcal{F}$) in order to compute the edges of the image by performing a convolution over the designated image (the convolution operator is denoted as $*$). The objective function is then formulated as follows:

$$\begin{aligned}\mathcal{L}_{edge}(G_{\mathcal{SR}}, G_{\mathcal{RS}}) = \mathbb{E}_{\mathbf{s}\sim\mathcal{S}}[\parallel \mathcal{F} * G_{\mathcal{SR}}(\mathbf{s}) - \mathcal{F} * \mathbf{s} \parallel_1 + \parallel \mathcal{F} * G_{\mathcal{SR}}(\mathbf{s}) - \mathcal{F} * G_{\mathcal{RS}}(G_{\mathcal{SR}}(\mathbf{s})) \parallel_1] \\ + \mathbb{E}_{\mathbf{r}\sim\mathcal{R}}[\parallel \mathcal{F} * G_{\mathcal{RS}}(\mathbf{r}) - \mathcal{F} * \mathbf{r} \parallel_1 + \parallel \mathcal{F} * G_{\mathcal{RS}}(\mathbf{r}) - \mathcal{F} * G_{\mathcal{SR}}(G_{\mathcal{RS}}(\mathbf{r})) \parallel_1].\end{aligned} \tag{4}$$

*Color Mean and Variance Retaining Loss.* A generator that outputs the correct edge structure of the eye may not necessarily output the correct segmentation feature corresponding to its edges, e.g., the translated pupil is half-dark. Therefore, we propose a loss function that encourages the minimization of the statistical (mean and variance) color difference between the translated image and the target domain image, i.e., $\mathbf{t}^i_{sr}$ and $\mathbf{r}^i$, respectively. This loss works to increase the unity with respect to the color estimation within each eye part (e.g., pupil) when performing image translation. Particularly, we compute the difference in mean and variance for each corresponding segmentation class $k \in K$ number of classes, i.e., pupil, iris, sclera, and background, between image pairs, i.e., translated image $\mathbf{t}^i_{sr}$ and target domain image $\mathbf{r}^i$. Let the mean of the pixels for class $k$ of image $x$ be $\mu_k(x)$, where each class $k$ has $P_k$ number of pixels. As a result, we obtain the following:

$$\mu_k(x) = \frac{1}{P_k}\sum_{p=1}^{P_k} x_p. \tag{5}$$

Given the above, the color mean retaining loss function is then represented as follows:

$$\mathcal{L}_{mean}(G_{\mathcal{SR}}, G_{\mathcal{RS}}) = \mathbb{E}_{\mathbf{s}\sim\mathcal{S},\mathbf{r}\sim\mathcal{R}} \sum_k |\mu_k(G_{\mathcal{SR}}(\mathbf{s})) - \mu_k(\mathbf{r})| + \mathbb{E}_{\mathbf{r}\sim\mathcal{R},\mathbf{s}\sim\mathcal{S}} \sum_k |\mu_k(G_{\mathcal{RS}}(\mathbf{r})) - \mu_k(\mathbf{s})|. \tag{6}$$

Similarly, let the variance of the pixels for class $k$ of image $x$ be $\sigma_k(x)$, where each class $k$ has $P_k$ number of pixels. We then compute the following:

$$\sigma_k(x) = \frac{1}{P_k}\sum_{p=1}^{P_k} (x_p - \mu_k(x))^2. \tag{7}$$

The color variance retaining loss function is then represented finally in the following manner:

$$\mathcal{L}_{var}(G_{\mathcal{SR}}, G_{\mathcal{RS}}) = \mathbb{E}_{\mathbf{s}\sim\mathcal{S},\mathbf{r}\sim\mathcal{R}} \sum_k |\sigma_k(G_{\mathcal{SR}}(\mathbf{s})) - \sigma_k(\mathbf{r})| + \mathbb{E}_{\mathbf{r}\sim\mathcal{R},\mathbf{s}\sim\mathcal{S}} \sum_k |\sigma_k(G_{\mathcal{RS}}(\mathbf{r})) - \sigma_k(\mathbf{s})|. \tag{8}$$

*Final Structure Retaining CycleGAN Objective Function.* Given the above designed set of loss functions, the full objective function used to train our neural system is the following:

$$\mathcal{L}(G_{\mathcal{SR}}, G_{\mathcal{RS}}, D_{\mathcal{R}}, D_{\mathcal{S}}) = \mathcal{L}_{adv}(G_{\mathcal{SR}}, D_{\mathcal{R}}) + \mathcal{L}_{adv}(G_{\mathcal{RS}}, D_{\mathcal{S}}) + \gamma_{cyc}\mathcal{L}_{cyc}(G_{\mathcal{SR}}, G_{\mathcal{RS}}) + \gamma_{id}\mathcal{L}_{id}(G_{\mathcal{SR}}, G_{\mathcal{RS}}) + \gamma_{edge}\mathcal{L}_{edge}(G_{\mathcal{SR}}, G_{\mathcal{RS}}) + \gamma_{mean}\mathcal{L}_{mean}(G_{\mathcal{SR}}, G_{\mathcal{RS}}) + \gamma_{var}\mathcal{L}_{var}(G_{\mathcal{SR}}, G_{\mathcal{RS}}) \tag{9}$$

where $\gamma_{cyc}$, $\gamma_{id}$, $\gamma_{edge}$, $\gamma_{mean}$, $\gamma_{var}$ are the coefficients that control the effect that each corresponding loss term has on the full system optimization process.

### 3.2 Siamese Network Filtering

After a synthetic image has been reconstructed to be closer to the real eye image distribution, there will still exist parts of the images that are not very close to the real distribution. As can be seen from the PCA plot of intermediate latent vectors (see Fig. 3), the real image distribution does not fully cover the reconstructed image distribution. In order to overcome this, we remove images that are far away from the real image distribution by thresholding their distance to the real image dataset's centroid. In order to measure the distance of one image from the other, we craft a Siamese Network [Chen et al. 2019; Deng et al. 2017; Koch et al. 2015] that infers the latent representation of each image such that the distance, i.e, the L2 distance, between images from two different domains should be far from each other. As a result, we employ a contrastive loss [Hadsell et al. 2006; Melekhov et al. 2016] to achieve this goal.

*Problem Formulation.* We construct a Siamese Network that maps from image/pixel space to a latent vector of size $n$ ($n = 2$ in our case): $f : \mathbb{I} \rightarrow \mathbb{R}^2$. We then filter the reconstructed dataset $\mathcal{T}_{\mathcal{R}}$ by thresholding each synthetic image's distance-to-centroid on the real dataset. In particular, we first compute the centroid vector representation $c_{\mathcal{R}}$ of the real dataset in the following way:

$$c_{\mathcal{R}} = \mathbb{E}_{\mathbf{r}\sim\mathcal{R}}\left[f(\mathbf{r})\right]. \tag{10}$$

We next compute the distance $d^i$ of each reconstructed image $\mathbf{t}^i_{sr} \in \mathcal{T}_{\mathcal{R}}$ to the real domain's centroid:

$$d^i = \| f(\mathbf{t}^i_{sr}) - c_{\mathcal{R}} \|_2^2 . \tag{11}$$

Finally, we may then choose only images in which the distance $d^i$ is smaller than a certain threshold (0.005) to ultimately synthesize a filtered dataset.

### 3.3 A Domain-Adversarial Neural Network (DANN) for Segmentation

Current segmentation methodologies have excelled in working with domain-specific datasets. However, when performing inference over different domains, performance degradation is often observed, i.e., training a segmentation model on a synthetic dataset yields low mean intersection over union on the test dataset of real images. One way to close the domain gap is to learn a feature extractor that can generalize across different domains in its latent space. To achieve this, we took inspiration from the training of domain-adversarial neural network (DANN) [Ganin et al. 2017]. In our problem context, we propose constructing a decoder head (for segmentation) instead of a class predictor as in the original DANN. Our goal is to make the encoder's output the feature/component that generalizes across two domains. As a result, we need to reverse the gradients [Ganin et al. 2017] that come from the domain classifier. In particular, given a set of $n$-dimensional latent vectors $\mathbb{R}^n$, the set of 2D images (height $h$, width $w$, $c$ channels) $\mathbb{I}^{h\times w\times c}$, $K$ number of segmentation classes, the encoder (of image $x$) $e_\theta(x) : \mathbb{I}^{h\times w\times c} \rightarrow \mathbb{R}^n$ (which has 5 down blocks [Chaudhary et al. 2019]), the decoder $d_\theta(e(x)) : \mathbb{R}^n \rightarrow \mathbb{I}^{h\times w\times K}$ (which has 4 up blocks [Chaudhary et al. 2019]), and the domain classifier $f_\lambda(e(x)) : \mathbb{R}^n \rightarrow \mathbb{R}^1$, the optimization objective of our network is formally the

following:

$$\min_{e_\theta, d_\theta, f_\lambda} \max_{e_\theta} \mathcal{L}(e_\theta, d_\theta, f_\lambda) \Leftrightarrow \min_{e_\theta, d_\theta} \mathcal{L}_{\text{ritnet}}(e_\theta, d_\theta) \quad \text{and} \quad \max_{e_\theta} \min_{f_\lambda} \mathcal{L}_{\text{domain}}(e_\theta, f_\lambda)$$
$$\text{with } \mathcal{L}_{\text{domain}}(e_\theta, f_\lambda) = -\mathbb{E}_{x \sim \mathbb{I}^{h \times w \times c}} \left[ l \log \left( f_\lambda(e_\theta(x)) \right) + (1 - l) \log \left( 1 - f_\lambda(e_\theta(x)) \right) \right]. \tag{12}$$

Note that the above objective function can be deconstructed into two key components. First, it involves optimizing the segmentation model (RITnet) loss function, which further decomposes into the minimizing of a generalized dice loss [Chaudhary et al. 2019; Milletari et al. 2016; Sudre et al. 2017], a boundary-aware loss [Chaudhary et al. 2019], and a surface loss [Kervadec et al. 2021] explained in Section 2. Second, our domain classifier loss $\mathcal{L}_{\text{domain}}(e_\theta, f_\lambda)$ can be expressed as the binary cross-entropy loss between the predicted dataset classification of image $x$ and its label $l$ – where $l$ denotes which domain that the image $x$ comes from. Our objective is to minimize this domain classifier loss with respect to the domain classifier $f_\lambda$ such that it maximizes the same loss for the encoder $e_\theta$. This also means that the gradient signal that optimizes the domain classifier $f_\lambda$ should be negated when it flows through the encoder $e_\theta$. To achieve this, we utilize the reverse gradient layer [Ganin et al. 2017] at the end of the encoder so as to make sure that the gradients minimize the domain classifier loss for the domain classifier $f_\lambda$ while still ensuring that the encoder $e_\theta$ is maximizing that same loss value.

## 4 RESULTS AND DISCUSSION

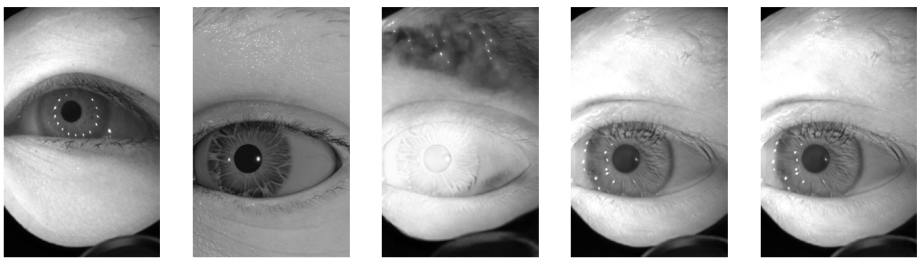

Fig. 2. Sample images from datasets used in our experiments. From left to right: OpenEDS (target domain), and four synthetic/constructed source domains - RITEyes, CGAN, SRCGAN, and SRCGAN-S.

Table 1. Number of images used. Source domains include RITEyes, CGAN, and SRCGAN. Filtered Source domain includes SRCGAN-S. Target domain includes OpenEDS.

| Set/Domain | Source | Filtered Source | Target |
|---|---|---|---|
| Train | 9,216 | 2,915 | 8,916 |
| Validation | 1,024 | 323 | 2,403 |
| Total | 10,240 | 3,238 | 11,319 |

Table 2. Number of training epochs for each combination of $M$ images in the source domain and $N$ images in target domain.

| N/M | 64 | 256 | 1,024 | 2,048 | 4,096 |
|---|---|---|---|---|---|
| 0 | 1,600 | 800 | 200 | 100 | 70 |
| 64 | 400 | 150 | 120 | 100 | 70 |
| 8,192 | 120 | 100 | 80 | 70 | 60 |

*Dataset Details.* Fig. 2 shows sample images from five datasets. The first image is from the real OpenEDS dataset [Garbin et al. 2019], the second image is synthetic and was generated using the RITEyes rendering pipeline [Nair et al. 2020], and we have constructed the remaining three datasets – CGAN (created using the CycleGAN method [Zhu et al. 2017] described in related work, SRCGAN (created using our Structure Retaining CycleGAN method described in Section 3.1, and SRCGAN-S

(created by filtering the SRCGAN dataset through our Siamese Network described in Section 3.2. We denote the OpenEDS dataset [Garbin et al. 2019] as the target dataset/domain that the other datasets must be adapted to (the other four are labeled as the source datasets). Both the source (synthetic) and target (real) dataset have four types of label – pupil, sclera, iris, and background – although they may have different locations within the images. The image resolution of the Open EDS dataset is $400 \times 640$ pixels [Garbin et al. 2019], so we used the same resolution for the synthetic dataset generated by the RITEyes pipeline [Nair et al. 2020] as well as for the three constructed datasets (CGAN, SRCGAN, and SRCGAN-S) in order to reduce variance in our neural networks' input space. We also generated synthetic images with the same number of channels (grayscale images) and the same number of segmentation classes, i.e., pupil, iris, sclera, background. Visual inspection suggests that even the more convincing of artificial images differ from the real images along several dimensions, including the realism in eye lashes, skin texture, and iris texture. The training procedure meant to reduce these differences proceeded under a 3-fold cross-validation scheme. The number of images used in each dataset is shown in Table 1. While generating data for training, the data augmentation methods described in RITnet [Chaudhary et al. 2019] were used. These include vertical axis reflection, Gaussian blur (with a kernel of size $7 \times 7$ with standard deviation $2 \leq \sigma \leq 7$), image translation of 0-20 pixels along both axes, image corruption by drawing 2-9 random vertical and horizontal thin lines, and image corruption using a starburst pattern. Each of the augmentation methods has a probability of 0.2 of being selected when generating training images. The number of training epochs/iterations is shown in Table 2. Note that the number of epochs is manually adjusted to be higher while training datasets with lower number of images. This is done to relatively balance the total training steps across training instances.

*Architecture of CycleGAN-based Models.* We utilize elements of the ResNet architecture [He et al. 2016; Zhu et al. 2017] within our CycleGAN-based models (i.e. CycleGAN and Structure Retaining CycleGAN). The generator consists of a convolution neural network (CNN) block, followed by downsampling by a factor of 4, which is then followed by 8 residual blocks [He et al. 2016], and finally, upsampling is applied with a factor of 4 to obtain the generated image. For the discriminator, we use 4 CNN blocks with a stride of 4 and a leaky ReLU activation function ($\alpha = 0.2$), followed by a linear transformation layer that outputs a single neuron which predicts if an input is real or fake.

In Section 3.1, a number of hyperparameters that control the Structure Retaining CycleGAN objective function are mentioned. We choose to keep the cycle loss $\gamma_{\text{cyc}}$ and identity loss $\gamma_{\text{id}}$ coefficient the same as in prior work [Liu 2022; Taigman et al. 2017; Zhu et al. 2017] (i.e, 10). For the coefficients of the newly proposed objective functions ($\gamma_{\text{edge}}$, $\gamma_{\text{mean}}$, $\gamma_{\text{var}}$), we perform a test over multiple combinations of parameters and choose the best combination based on model performance, i.e., with respect to mean distance to the real distribution's centroid. The best combination of hyperparameters was $\gamma_{\text{edge}} = 0.1$, $\gamma_{\text{mean}} = 0$, $\gamma_{\text{var}} = 60$.

There are multiple ways to measure the performance of CycleGAN-based models such as Inception Score where generated images are evaluated based on predictability and diversity [Lucic et al. 2017; Salimans et al. 2016]. In our context, the CycleGAN-based model-generated images have to be both meaningful, i.e., predictable by the Inception Network classifier, as well as meet the goal of being close to the real domain. We choose to utilize mean intersection over union (mIoU) to compare classification performance across models in the context of segmentation prediction. In addition, we measure the closeness of generated datasets to the real dataset. In order to achieve this, we first employ the Inception Network [Szegedy et al. 2017] used in the Siamese Network (Section 3.2) to infer the latent representation vector corresponding to each image. Then, we compute the statistics for each vector in the source domain distribution compared to the centroid of the real distribution, i.e., using the L2 distance. We next compute the real distribution's centroid by averaging every real

Table 3. Model performance (mIoU and mmIoU$^C$) comparison on the real target dataset (OpenEDS) of the RITnet segmentation network (a) and our DANN segmentation network (b). Both models were trained on different number of images ($N$) from the 4 source domains (see Fig. 2). The final standard deviation of mmIoU$^C$ is computed based on Bessel's correction formula. Best performance in bold and highlighted.

(a) RITnet

| Dataset/N | 64 | 256 | 1,024 | 2,048 | 2,915 | 4,096 | mmIoU$^C$ |
|---|---|---|---|---|---|---|---|
| RITEyes | 0.36±0.11 | 0.47±0.03 | 0.54±0.03 | 0.53±0.04 | - | 0.56±0.04 | 0.49±0.09 |
| CGAN | 0.20±0.01 | 0.21±0.00 | 0.22±0.01 | 0.23±0.02 | - | 0.22±0.01 | 0.21±0.01 |
| SRCGAN (ours) | 0.50±0.02 | 0.55±0.01 | 0.55±0.01 | 0.57±0.03 | - | **0.61±0.01** | 0.55±0.04 |
| SRCGAN-S (ours) | **0.52±0.02** | **0.56±0.04** | **0.57±0.01** | **0.61±0.01** | **0.66±0.02** | - | **0.59±0.05** |

(b) DANN

| Dataset/N | 64 | 256 | 1,024 | 2,048 | 2,915 | 4,096 | mmIoU$^C$ |
|---|---|---|---|---|---|---|---|
| RITEyes | 0.55±0.04 | 0.54±0.05 | 0.52±0.03 | 0.55±0.05 | - | 0.57±0.03 | 0.54±0.04 |
| CGAN | 0.22±0.01 | 0.22±0.01 | 0.22±0.01 | 0.24±0.01 | - | 0.23±0.01 | 0.22±0.01 |
| SRCGAN (ours) | 0.60±0.03 | **0.62±0.02** | **0.64±0.02** | **0.65±0.04** | - | **0.70±0.04** | **0.64±0.04** |
| SRCGAN-S (ours) | **0.62±0.02** | 0.61±0.03 | 0.63±0.04 | **0.65±0.02** | **0.71±0.04** | - | **0.64±0.04** |

image vector. The mean distance-to-real-centroid can be formally stated as:

$$\mu_d^C = \mathbb{E}_{c\sim C}\left[\| f(c) - c_{\mathcal{R}} \|_2^2\right]. \tag{13}$$

where $C$ represents the source dataset being evaluated and $c_{\mathcal{R}}$ is the vector representation of the real dataset's centroid as calculated in Equation 10. We obtain $\mu_d^{\text{CGAN}} \approx 0.023$ and $\mu_d^{\text{SRCGAN}} \approx 0.005$ (the best given the hyperparameter combination). The results notably align with the PCA plots of the DANN module (see Section 3.3) across different datasets as shown in Fig. 3.

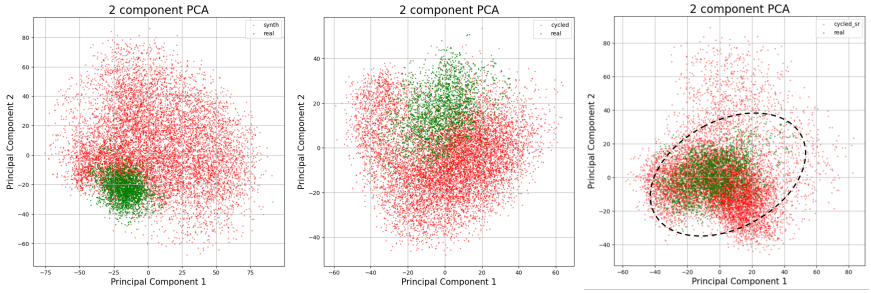


Fig. 3. Comparison of PCA plots of intermediate latent vectors for source and target domains produced by the DANN module (described in Section 3.3). Left: RITEyes (red) vs. OpenEDS (green). Middle: CGAN (red) vs. OpenEDS (green). Right: SRCGAN (red) vs. OpenEDS (green). Note that red dots inside the ellipse make up the SRCGAN-S distribution which represents filtered images that are close to the real distribution.

The mIoU results of our system, trained on each source dataset, also demonstrates that using the Structure Retaining CycleGAN model improves the performance of the model when processing real datasets (Fig. 4 and Table 3). Specifically, let $\{C^N\}$ be the set of all dataset $C$ instances that have $N$ number of training images, e.g., for $\{\text{SRCGAN-S}^N\}$, $N \in \{64, 128, 1024, 2048, 2915\}$. We may then compute the representative mean of the individual mIoUs (mmIoU$^C$) for each dataset $C$ as:

$$\text{mmIoU}^C = \mathbb{E}_{C_i \in \{C^N\}} \text{mIoU}^{C_i} \tag{14}$$

where mIoU$^{C_i}$ is the mIoU of the segmentation model trained on the $i$-th instance in that collection of instances. In other words, we average all segmentation mIoU scores for a model over every

dataset $C$ instance for each dataset $C$ in order to obtain the per-dataset statistics mmIoU$^C$. We observe that the RITnet segmentation model, when trained with ($C$ =) SRCGAN-S images, has 0.04, 0.38, and 0.10 higher mmIoU$^C$ measurements than when trained with SRCGAN, CGAN, and RITEyes images, respectively. Similarly, the DANN model, when trained with ($C$ =) SRCGAN-S images, yields 0.00, 0.42, and 0.10 higher mmIoU$^C$ scores as compared to training with SRCGAN, CGAN, RITEyes images, respectively (see Table 3). Note that, in Fig. 4 and Table 3, there is no data for the number of images greater than 2,915 for SRCGAN-S dataset because there are only 2,915 images in the training set (see Table 1). This result shows that we have developed a dataset production and refinement method that maps from the synthetic domain to the real domain, improving the plausibility of the synthetic images and providing a means of closing the domain/*sim2real* gap.

*Siamese Network for Image Filtering.* We train a Siamese Network based on the Inceptionv4 architecture [Szegedy et al. 2017] as the feature extractor, resulting in a total number of 27,465,826 parameters. Concretely, we train the Siamese Network for 20 epochs with 10,000 pairs of the same source, same target, and different domain images for each epoch. We achieve a (final) 0.0001 contrastive loss when estimating the L2/Euclidean distance between image pairs given that the labeled margin between two different-domain images is one. This means that the images that are close to each other (within the same domain) will have an estimated distance close to zero while images that are further away from one another (from different domains) will have a distance around one (see the sample estimation of L2 distance shown in Fig. 5).

After training the Siamese Network, the resulting model is used to output a latent representation of each image which is then used to filter the SRCGAN dataset. Through experimentation, we set the distance threshold used for filtering to be 0.005 (the best mean distance-to-real-centroid $\mu_d^{\text{SRCGAN}}$ considering datasets generated from various Structure Retaining CycleGAN models). We then train all of the models on this filtered dataset and measure the mIoU. As seen in Fig. 4 and Table 3, the model trained on the filtered dataset desirably results in a higher mIoU compared to every other model. This score is specifically higher than that of the model trained on the non-filtered synthetic dataset (RITEyes) by about 10%. This result shows that, after processing the synthetic dataset with our pipeline's other modules, we can further refine this dataset by filtering out images that are estimated not close to the real image distribution. This, in effect, further boosts the performance of the eye segmentation network.

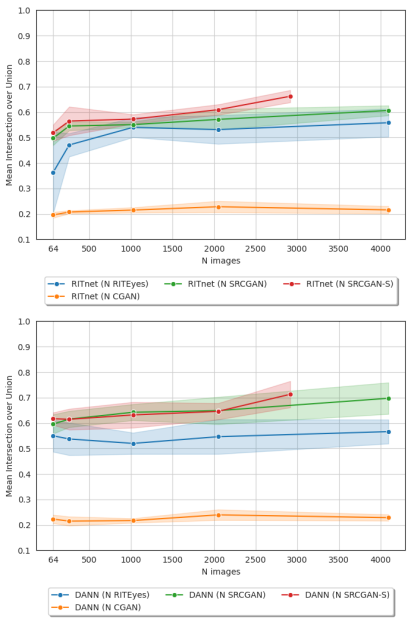


Fig. 4. Performance comparison on the real target dataset (OpenEDS) of the RITnet (Left) and our DANN (Right) segmentation networks. Both models were trained on the 4 source domains (see Figure 2). Shaded regions depict ±1 SD in the 3-fold cross validation scheme.

*Domain Adversarial Neural Network.* Our DANN module consists of a segmentation network based on RITnet with the addition of a fully-connected neural network domain classifier. The bottleneck output of the segmentation network, i.e., encoder output, is then fed into a reverse gradient layer. This reverse gradient layer acts as the identity function in the forward pass and negation function in the backward pass. The domain classifier then takes this in as input and

Table 4. Performance (mIoU) of RITnet vs. DANN when training on 4,096/2,915 source domain images together with 8,192 (a), or 64 (b) images from the OpenEDS dataset. Best performance in bold and highlighted.

(a) 8,192 OpenEDS images used in training

| Model/Dataset | 4,096 RITEyes | 4,096 CGAN | 4,096 SRCGAN (ours) | 2,915 SRCGAN-S (ours) |
|---|---|---|---|---|
| RITnet | 0.94±0.00 | **0.94±0.00** | 0.94±0.00 | **0.94±0.00** |
| DANN (ours) | 0.94±0.00 | 0.93±0.00 | 0.94±0.00 | 0.93±0.01 |

(b) 64 OpenEDS images used in training

| Model/Dataset | 4,096 RITEyes | 4,096 CGAN | 4,096 SRCGAN (ours) | 2,915 SRCGAN-S (ours) |
|---|---|---|---|---|
| RITnet | 0.83±0.04 | 0.80±0.04 | 0.81±0.01 | 0.82±0.01 |
| DANN (ours) | **0.90±0.02** | **0.89±0.01** | **0.90±0.01** | **0.89±0.02** |

outputs a single classification probability (score). The network classifier is made up of five dense layers that use ReLU activation functions with the logistic sigmoid activation in the last layer.

We compare the performance of the RITnet and DANN segmentation networks when trained on datasets consisting of 4,096 or 2,915 synthetic eye images and a varying number of real images. Our results are shown in Table 4 for 8,192 and 64 real images (top and bottom). We observe that the performance of the two models is close to each other when the number of real training images is high, and the DANN module outperforms RITnet when training with a smaller number of real images (see Table 4). Therefore, we conclude that the DANN module improves the generalization between different domains, significantly closing the domain gap while increasing the mIoU performance of the segmentation networks when fewer real images are used in the training process.

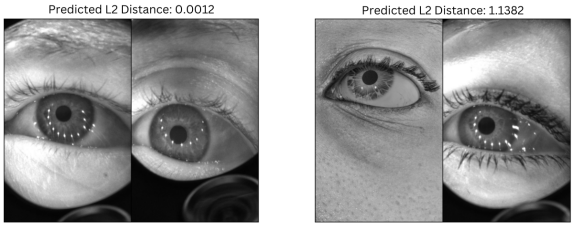


Fig. 5. Sample distance prediction measurements of the Siamese Network for two images from OpenEDS dataset (left) and an image from RITEyes and one from OpenEDS respectively (right).

*Privacy and Ethics.* No new human data was recorded for this study. We instead utilized an existing dataset of real human eyes (OpenEDS) as our target domain. The source datasets were either rendered using computer graphics (RITEyes) or generated by neural networks (CGAN, SRCGAN, SRCGAN-S). Importantly, our work makes substantial contributions toward the objective of minimizing reliance on actual human training data. Concretely, given that we develop a modular neural system that is trained mostly on synthetic data in order to estimate the segmentation for real eyes, we reduce the need to collect eye tracking data on actual human subjects. Each method in our work also contributes to the mitigation of privacy issues/concerns. For example, while our Structure Retaining CycleGAN method needs a real eye dataset to which to map a synthetic dataset, only a small fixed number of real images are ultimately required to establish the centroid and spread of the target distribution (compared to performant systems that require a vast collections of human eye images containing sensitive data). This reduces the risk of human data exposure and violation of the subjects' data privacy. Furthermore, avoiding/reducing the need to record human data further protects humans from exposing other aspects of their identity such as facial biometric information, facial behaviors, gaze behavior, and subject personality.

In Table 4, we observe that although the performance of both the RITnet and DANN segmentation networks is low when there are no real training images and performance for both increases proportionally with respect to the number real training images, the performance of the DANN model is higher and more stable than that of RITnet once we use a small number of real training images. This further reinforces the fact that DANN exhibits the ability to generalize across domains given only a fixed, finite set of real training image samples while RITnet cannot. This, again, circumvents the need to record large amounts of training data from the human subjects. While using a small, finite number of real human eye image samples as the target domain is beneficial for the reasons listed above, we acknowledge that this approach can potentially introduce issues of bias and fairness, particularly when different ages, genders, races, eye-textures, skin color, etc. are not represented in the target real dataset.

## 5 CONCLUSION

In this paper, we presented a multi-step neural pipeline for tackling the problem of *sim2real* in the context of eye-tracking through the use of domain adaptation. Our architecture consists of three main components. First, a novel Structure Retaining CycleGAN is implemented to reconstruct synthetic eye images while ensuring they match the distribution of real eye images. Second, a Siamese Network is designed to filter out poorly-reconstructed images through a learned dataset pruning approach. Lastly, a model adapted from a domain-adversarial neural network structure is employed to semantically segment the real images.

Subjectively, the datasets reconstructed at the different stages as we progress through our pipeline do appear to be more realistic/plausible (see Fig. 2). Our objective results further indicate that the later stage datasets do indeed yield greater performance (see Fig. 4 and Table 3). Specifically, the SRCGAN dataset outperforms CGAN in terms of mean distance to the real centroid and downstream mIoU score. The SRCGAN-S dataset performance is similar to SRCGAN but offers the additional benefit of faster training time since the number of images in the dataset is greatly reduced compared to SRCGAN (2,915 for SRCGAN-S vs. 9,216 for SRCGAN, see Table 1). The fact that we are able to achieve similar performance with fewer images confirms that our Siamese Network successfully filters out problematic synthetic image samples. Finally, our results show that our proposed DANN segmentation network outperforms RITnet in terms of segmentation mIoU score when only a small number of real images is included among the synthetic datasets used for training (see Table 4). Overall, we have provided empirical evidence that our multi-step neural architecture results in improved synthetic datasets for training semantic segmentation models, and we also present an improved segmentation model (DANN). Furthermore, our results have positive implications for reducing the cost and burden associated with capturing and manually labeling large quantities of real human eye data, which in turn also promotes data privacy.

In terms of future research work, the results we presented focus on the overall mIoU score averaged across distinct eye regions. However, delving into region-specific mIoU scores could yield additional insights and improvements. For instance, our observations indicate that the mIoU score for the sclera region tends to lag behind those for the pupil and iris regions. Further exploration of region-specific objective functions may effectively address and enhance performance in these specific areas. Additionally, in our DANN sub-module, the reverse gradient from the domain classifier currently flows through the encoder of only the segmentation network, which makes the domain generalization learning occur in the encoder but not in the decoder. It is worth exploring whether this might be a contributing factor in the DANN model's slight underperformance compared to RITnet when the number of real images gets (much) larger. Another challenge central to the problem of domain adaptation relates to the ability to generalize eye segmentation models with respect to the target domain. Although the OpenEDS dataset consider different combinations of

age, sex, usage of glasses, and corneal topography [Garbin et al. 2019], they may not account for the diversity in other features such as sclera/iris texture. While it is possible to introduce diversity in the rendering pipeline by using different head models representing different genders, ages, races, iris/sclera textures, etc. as in RITEyes [Nair et al. 2020], our current pipeline does not prioritize the retention of these features during domain transfer. Therefore, it is worthwhile to explore domain transfer functions that preserve such features. Additionally, while mapping from the synthetic to the real domain, it may be beneficial to incorporate eye glint as an additional transferable structure in order to further improve the realism of the resulting eye images. While our current research integrates glints within the eye region textures, future studies could explore framing it as a separate structural element and aim to preserve its characteristics from the source domain. Lastly, future works can also explore the impacts of different image resolution on the efficiency of our algorithms.

## ACKNOWLEDGMENTS

We would like to thank Meta Reality Labs for providing invaluable feedback and supporting this work.[1] This material is based upon work supported by the National Science Foundation under Award No. DGE-2125362. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

John Canny. 1986. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8, 6 (1986), 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

Watchanan Chantapakul, Linda Hansapinyo, and Karn Patanukhom. 2019. Eye Semantic Segmentation Using Ensemble of Deep Convolutional Neural Networks. *Proceedings of the 2019 2nd Artificial Intelligence and Cloud Computing Conference* (2019). https://api.semanticscholar.org/CorpusID:211520180

Aayush K. Chaudhary, Rakshit Kothari, Manoj Acharya, Shusil Dangi, Nitinraj Nair, Reynold Bailey, Christopher Kanan, Gabriel Diaz, and Jeff B. Pelz. 2019. RITnet: Real-time Semantic Segmentation of the Eye for Gaze Tracking. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE. https://doi.org/10.1109/iccvw.2019.00568

Chen Chen, Qifeng Chen, Minh Do, and Vladlen Koltun. 2019. Seeing Motion in the Dark. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 3184–3193. https://doi.org/10.1109/ICCV.2019.00328

J.G. Daugman. 1993. High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 11 (1993), 1148–1161. https://doi.org/10.1109/34.244676

Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. 2017. Image-Image Domain Adaptation with Preserved Self-Similarity and Domain-Dissimilarity for Person Re-identification. https://doi.org/10.48550/ARXIV.1711.07027

Xingping Dong and Jianbing Shen. 2018. Triplet Loss in Siamese Network for Object Tracking. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 472–488.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2017. *Domain-Adversarial Training of Neural Networks*. Springer International Publishing, Cham, 189–209. https://doi.org/10.1007/978-3-319-58347-1_10

Stephan J. Garbin, Yiru Shen, Immo Schuetz, Robert Cavin, Gregory Hughes, and Sachin S. Talathi. 2019. OpenEDS: Open Eye Dataset. arXiv:1905.03702 [cs.CV]

Shreya Ghosh, Abhinav Dhall, Munawar Hayat, Jarrod Knibbe, and Qian Ji. 2021. Automatic Gaze Analysis: A Survey of Deep Learning based Approaches. *ArXiv* abs/2108.05479 (2021). https://api.semanticscholar.org/CorpusID:236987204

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014a. Generative Adversarial Networks. https://doi.org/10.48550/ARXIV.1406.2661

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014b. Explaining and Harnessing Adversarial Examples. https://doi.org/10.48550/ARXIV.1412.6572

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A Kernel Two-Sample Test. *Journal of Machine Learning Research* 13, 25 (2012), 723–773. http://jmlr.org/papers/v13/gretton12a.html

[1]Author Alexander Fix at Meta was involved as an advisor on this research, which was conducted at RIT

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2. 1735–1742. https://doi.org/10.1109/CVPR.2006.100

Simon Haykin. 1998. *Neural Networks: A Comprehensive Foundation* (2nd ed.). Prentice Hall PTR, USA.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. https://doi.org/10.1109/CVPR.2016.90

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6629–6640.

N. Kanopoulos, N. Vasanthavada, and R.L. Baker. 1988. Design of an image edge detection filter using the Sobel operator. *IEEE Journal of Solid-State Circuits* 23, 2 (1988), 358–367. https://doi.org/10.1109/4.996

Manuel Kaspar, Juan D. Muñoz Osorio, and Juergen Bock. 2020. Sim2Real Transfer for Reinforcement Learning without Dynamics Randomization. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 4383–4388. https://doi.org/10.1109/IROS45743.2020.9341260

Hoel Kervadec, Jihene Bouchtiba, Christian Desrosiers, Eric Granger, Jose Dolz, and Ismail Ben Ayed. 2021. Boundary loss for highly unbalanced segmentation. *Medical Image Analysis* 67 (jan 2021), 101851. https://doi.org/10.1016/j.media.2020.101851

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese Neural Networks for One-shot Image Recognition.

Sven Kosub. 2019. A note on the triangle inequality for the Jaccard distance. *Pattern Recognition Letters* 120 (2019), 36–38. https://doi.org/10.1016/j.patrec.2018.12.007

Rakshit S. Kothari, Reynold J. Bailey, Christopher Kanan, Jeff B. Pelz, and Gabriel J. Diaz. 2022. EllSeg-Gen, towards Domain Generalization for Head-Mounted Eyetracking. *Proceedings of the ACM on Human-Computer Interaction* 6, ETRA (may 2022), 1–17. https://doi.org/10.1145/3530880

Rakshit S. Kothari, Aayush K. Chaudhary, Reynold J. Bailey, Jeff B. Pelz, and Gabriel J. Diaz. 2021. EllSeg: An Ellipse Segmentation Framework for Robust Gaze Tracking. *IEEE Transactions on Visualization and Computer Graphics* 27, 5 (may 2021), 2757–2767. https://doi.org/10.1109/tvcg.2021.3067765

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems* 25 (01 2012). https://doi.org/10.1145/3065386

Yann LeCun, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361, 10 (1995), 1995.

Yann LeCun, Y. Bengio, and Geoffrey Hinton. 2015. Deep Learning. *Nature* 521 (05 2015), 436–44. https://doi.org/10.1038/nature14539

Sizhe Liu. 2022. Study for Identity Losses in Image-to-Image Domain Translation with Cycle-Consistent Generative Adversarial Network. *Journal of Physics: Conference Series* 2400, 1 (dec 2022), 012030. https://doi.org/10.1088/1742-6596/2400/1/012030

Conny Lu, Qian Zhang, Kapil Krishnakumar, Jixu Chen, Henry Fuchs, Sachin Talathi, and Kun Liu. 2022. Geometry-Aware Eye Image-To-Image Translation. In *2022 Symposium on Eye Tracking Research and Applications* (Seattle, WA, USA) *(ETRA '22)*. Association for Computing Machinery, New York, NY, USA, Article 69, 7 pages. https://doi.org/10.1145/3517031.3532524

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. 2017. Are GANs Created Equal? A Large-Scale Study. https://doi.org/10.48550/ARXIV.1711.10337

Iaroslav Melekhov, Juho Kannala, and Esa Rahtu. 2016. Siamese network features for image matching. *2016 23rd International Conference on Pattern Recognition (ICPR)* (2016), 378–383. https://api.semanticscholar.org/CorpusID:9740232

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. 2016. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*. 565–571. https://doi.org/10.1109/3DV.2016.79

Shervin Minaee, Yuri Boykov, Fatih Murat Porikli, Antonio J. Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. 2020. Image Segmentation Using Deep Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44 (2020), 3523–3542. https://api.semanticscholar.org/CorpusID:210702798

Wan Azani Mustafa and Mohamed Mydin M. Abdul Kader. 2018. A Review of Histogram Equalization Techniques in Image Enhancement Application. *Journal of Physics: Conference Series* 1019, 1 (jun 2018), 012026. https://doi.org/10.1088/1742-6596/1019/1/012026

Nitinraj Nair, Rakshit Kothari, Aayush K. Chaudhary, Zhizhuo Yang, Gabriel J. Diaz, Jeff B. Pelz, and Reynold J. Bailey. 2020. RIT-Eyes: Rendering of near-eye images for eye-tracking applications. In *ACM Symposium on Applied Perception 2020* (Virtual Event, USA) *(SAP '20)*. Association for Computing Machinery, New York, NY, USA, Article 5, 9 pages. https://doi.org/10.1145/3385955.3407935

Loris Nanni, Giovanni Minchio, Sheryl Brahnam, Davide Sarraggiotto, and Alessandra Lumini. 2021. Closing the Performance Gap between Siamese Networks for Dissimilarity Image Classification and Convolutional Neural Networks. *Sensors (Basel, Switzerland)* 21 (08 2021). https://doi.org/10.3390/s21175809

Jeff Pelz and Dan Witzner Hansen. 2017. System and Method for Eye Tracking. Pub. No.: WO/2017/205789 International Application No.: PCT/US2017/034756 Publication Date: 30.11.2017 International Filing Date: 26.05.2017; 2017/034756; G06K 9/00 (2006.01), G06K 9/62 (2006.01), G06K 9/46 (2006.01).

Jonathan Perry and Amanda Fernandez. 2019. MinENet: A Dilated CNN for Semantic Segmentation of Eye Features. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 3671–3676. https://doi.org/10.1109/ICCVW.2019.00453

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (Eds.). Springer International Publishing, Cham, 234–241.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training GANs. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Barcelona, Spain) *(NIPS'16)*. Curran Associates Inc., Red Hook, NY, USA, 2234–2242.

Samir Shah and Arun Ross. 2009. Iris Segmentation Using Geodesic Active Contours. *IEEE Transactions on Information Forensics and Security* 4, 4 (2009), 824–836. https://doi.org/10.1109/TIFS.2009.2033225

Joseph Stember, H Celik, E Krupinski, P Chang, S Mutasa, Bradford Wood, A Lignelli, G Moonis, L Schwartz, and Sachin Jambawalikar. 2019. Eye Tracking for Deep Learning Segmentation Using Convolutional Neural Networks. *Journal of Digital Imaging* 32 (05 2019). https://doi.org/10.1007/s10278-019-00220-4

Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. 2017. *Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations*. Springer International Publishing, 240–248. https://doi.org/10.1007/978-3-319-67558-9_28

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. 2017. Inception-v4, inception-ResNet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (San Francisco, California, USA) *(AAAI'17)*. AAAI Press, 4278–4284.

Yaniv Taigman, Adam Polyak, and Lior Wolf. 2017. Unsupervised Cross-Domain Image Generation. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=Sk2Im59ex

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. 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)*. 23–30. https://doi.org/10.1109/IROS.2017.8202133

Liangping Tu and Changqing Dong. 2013. Histogram equalization and image feature matching. In *2013 6th International Congress on Image and Signal Processing (CISP)*, Vol. 01. 443–447. https://doi.org/10.1109/CISP.2013.6744035

Can Yaras, Bohao Huang, Kyle Bradbury, and Jordan M. Malof. 2021. Randomized Histogram Matching: A Simple Augmentation for Unsupervised Domain Adaptation in Overhead Imagery. https://doi.org/10.48550/ARXIV.2104.14032

Yuk-Hoi Yiu, Moustafa Aboulatta, Theresa Raiser, Leoni Ophey, Virginia L. Flanagin, Peter zu Eulenburg, and Seyed-Ahmad Ahmadi. 2019. DeepVOG: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning. *Journal of Neuroscience Methods* 324 (2019), 108307. https://doi.org/10.1016/j.jneumeth.2019.05.016

Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2015. Appearance-based gaze estimation in the wild. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4511–4520. https://doi.org/10.1109/CVPR.2015.7299081

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 2242–2251. https://doi.org/10.1109/ICCV.2017.244