MultiSDGAN: Translation of OCT Images to Superresolved Segmentation Labels Using Multi-Discriminators in Multi-Stages

Paria Jeihouni, *Member, IEEE*, Omid Dehzangi, *Senior Member, IEEE*, Annahita Amireskandari, Ali Rezai, and Nasser M. Nasrabadi, *Fellow, IEEE*

***Abstract*—Optical coherence tomography (OCT) has been identified as a non-invasive and inexpensive imaging modality to discover potential biomarkers for Alzheimer's diagnosis and progress determination. Current hypotheses presume the thickness of the retinal layers, which are analyzable within OCT scans, as an effective biomarker for the presence of Alzheimer's. As a logical first step, this work concentrates on the accurate segmentation of retinal layers to isolate the layers for further analysis. This paper proposes a generative adversarial network (GAN) that concurrently learns to increase the image resolution for higher clarity and then segment the retinal layers. We propose a multi-stage and multi-discriminatory generative adversarial network (MultiSDGAN) specifically for superresolution and segmentation of OCT scans of the retinal layer. The resulting generator is adversarially trained against multiple discriminator networks at multiple stages. We aim to avoid early saturation of generator model training leading to poor segmentation accuracies and enhance the process of OCT domain translation by satisfying all the discriminators in multiple scales. We also investigated incorporating the Dice loss and Structured Similarity Index Measure (SSIM) as additional loss functions to specifically target and improve our proposed GAN architecture's segmentation and superresolution performance, respectively. The ablation study results conducted on our data set suggest that the proposed MultiSDGAN with ten-fold cross-validation (10-CV) provides a reduced equal error rate with 44.24% and 34.09% relative improvements, respectively (p-values of the improvement level tests$<$.01). Furthermore, our experimental results also demonstrate that the addition of the new terms to the loss function improves the segmentation results significantly by relative improvements of 31.33% (p-value$<$.01).**

***Index Terms*—Optical coherence tomography, generative adversarial networks, superresolution, multi-stage generator, multi-discriminatory, dice loss, SSIM.**

Manuscript received April 12, 2021; revised July 27, 2021 and August 19, 2021; accepted August 22, 2021. Date of publication September 13, 2021; date of current version April 13, 2022. This work was supported by Rockefeller Neuroscience Institute start-up funds. *(Corresponding author: Omid Dehzangi.)*

Paria Jeihouni is with the Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26505 USA (e-mail: pj00001@mix.wvu.edu).

Omid Dehzangi is with the Department of Neuroscience, Rockefeller Neuroscience Institute and Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26505 USA (e-mail: omid.dehzangi@hsc.wvu.edu).

Annahita Amireskandari is with the Department of Ophthalmology and Visual Sciences, School of Medicine, West Virginia University, Morgantown, WV 26505 USA (e-mail: annahita.amireskandari@hsc.wvu.edu).

Ali Rezai is with the Rockefeller Neuroscience Institute, Department of Neuroscience, West Virginia School of Medicine, Morgantown, WV 26505 USA (e-mail: ali.rezai@hsc.wvu.edu).

Nasser M. Nasrabadi is with the Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26505 USA (e-mail: nasser.nasrabadi@mail.wvu.edu).

Digital Object Identifier 10.1109/JBHI.2021.3110265

I. Introduction

OPTICAL Coherence Tomography (OCT) is a noninvasive imaging technology for projecting cross-sectional images of the internal microstructure of human tissue in high-resolution [1]. In 1993 the first OCT scan from the human retina was done [2]; this ushered a new era of rapid development of the OCT technology, enabling cross-sectional visualization of the internal structure of biologic tissues [3], prominently the human retina. Nowadays, OCT is the primary modality for cross-sectional imaging of the human retina in high-resolution.

Alzheimer's Disease (AD) is one of the most common forms of dementia, increasingly prominent among the elderly population. Current standard methods for AD diagnostic imaging such as positron emission tomography and magnetic resonance imaging [4], comes with the added burden of being costly, time-consuming, and somewhat invasive. Preclinical AD, a recently recognized period, is a crucial phase, in which key pathophysiologic changes are underway within the brain, but symptoms have not yet become apparent. Preclinical AD can be diagnosed which is based on the presence of clinically validated biomarkers; however, these processes are invasive and expensive. On the other hand, according to several clinical studies, the neurodegenerative process of Alzheimer's, propelled by the abnormal cerebral accumulation of Amyloid-beta and Tau protein [4], may also affect the retina. In this regard, researchers have shown that the retina also goes through neuronal loss and vascular changes far earlier in disease progression than previously thought, and they suggest that individuals with preclinical AD may already have retinal microvascular abnormalities [5].

The neuronal loss of retinal tissue might be a possible biomarker for the presence of AD. More specifically, studies have shown a decrease in Retinal Nerve Fiber Layer (RNFL)

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

thickness [5] & [6] and also macular volume [7]. It was found that compared to healthy control subjects, those in the preclinical stage of AD showed a significant decrease in macular retinal nerve fiber layer (mRNFL) volume, over a 27- month follow-up interval period, as well as a decrease in outer nuclear layer and Inner Plexiform Layer (IPL) volumes. The decrease in mRNFL was found to be correlated with neocortical A-beta accumulation in the very early stages of AD. The authors suggested that the RNFL layer thickness may also contribute to declining cognitive functions such as audiovisual integration efficiency. The greater volume reduction in the mRNFL was significantly associated with reduced sensitivity to the binding strength of the audiovisual stimulus. Researchers also found relations between the AD progression and the Ganglion Cell Layer (GCL) degeneration [8]. It was found that thinner GCL-IPL complex is associated with dementia prevalence. Further, they support the hypothesis that analysis of retinal degeneration may aid in the diagnosis and progression of AD, citing OCT as a useful tool for monitoring research subjects. Hence, there are many ongoing research studies regarding the viability of OCT for this purpose as this alternative modality offers the benefit of being faster, non-invasive, cost-effective, and may show pathologic changes at an earlier stage of the disease.

To determine the usefulness of OCT as a biomarker, segmentation of the retinal layers is the first significant step. Due to the difficulty associated with manual segmentation of these images, which have a poor signal to noise ratio [9] because of the presence of noise such as micro-saccadic eye movements and the vessel projection shadow, it is imperative to program a method of automatic segmentation. Another hindrance commonly faced is the lack of clarity of the layer boundaries, which compels the research of super resolving the images for improved clarity. In this work, we have identified the goal of jointly super resolving and segmenting the OCT retinal scans. We propose an architecture employing a Generative Adversarial Network (GAN) [10] with a ResNet [11] based generator architecture, as well as analyzing the effect of a Dice loss [12] as an additional constraint, and show how its presence improves the performance.

II. Related Works

Biomedical image segmentation has been a demanding research topic for many years in the domain of computer vision. Since the advent of neural networks as proven methods for effective application in computer vision tasks [13], there have been numerous developments for semantic segmentation of biomedical images. Semantic segmentation algorithms found success following an encoder-decoder-based architecture, popularized by the work in [14] that is called Fully Convolutional Network (FCN). This architecture has two main components, the encoder part which downsizes the image, extracting features. In contrast, the decoder part upsamples the image back to the original size with the output segmentation. One of the issues faced with FCN is that these successive downsampling and upsampling steps result in losing some semantic and spatial information. U-Net [15] solved that issue, as it introduced skip connections in between the encoders and decoders, which would relay the spatial information from the encoder part to the corresponding feature maps of the decoder region. This model has been widely used in the domain of biomedical image segmentation. This architecture has spawned several variations such as the U-Net++, 3D-U-Net,and FC-DenseNet-Tiramisu [16]–[18] with varying degrees of performance. In the OCT segmentation-related domain, a specific model called RelayNet was published in [9], which to the best of our knowledge, provides the state of the art performance. There were other efforts in the segmentation of OCT images, such as the one described in [19], which employs a layer boundary evolution method as well as in [20], which involves the shortest path using the backtracking method. Some of the other efforts which employ deep learning are [21]–[24].

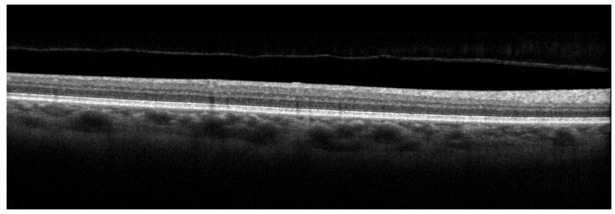


Fig. 1. A sample example of one singular OCT scan of the retina.

Another long, challenging task in the fraternity of computer vision is constructing a high-resolution photo-realistic images from their low-resolution counterparts. This strenuous task, aptly termed superresolution, has been a research topic for many years, even predating the advent of deep learning. Classical methods include various interpolation methods such as the nearest neighbor, bi-cubic or bi-linear, etc. With the success and popularity of FCN, a similar framework was designed as Superresolution Convolutional Neural Networks (SR-CNNs) [25]. In such systems, the image is first upsampled through bi-cubic interpolation, and fed through an FCN, resulting in output with high-resolution. The work in [26] is the continuation of SR-CNN, with residual blocks replacing the conventional convolutional blocks. Using such SR-CNN as the generator architecture, GANs have also been used to reconstruct images with higher resolution, being touted as 'SR-GAN' [26].

GANs have been quite prominent in learning deep representations and modeling high-dimensional data. This type of generative modeling competitively employs two trained networks, one being trained to synthesize new data and the other being trained to classify real and synthesized data. The network which contributes to generating new data based on an embedded input is called the generator. On the other hand, the second network that considers both the generated data and the real ground truth data and distinguishes them is called the discriminator. It works like a two-player game, where two entities are trying to outwit each other. The generator's purpose is to synthesize data that resembles real data in terms of distribution, and with further training, data becomes indistinguishable from the real data. At the same time, the discriminator is being trained to identify real and generated ones. Since its inception, GANs have gradually evolved and have been used for various tasks, including image processing and computer vision. Initially, GANs were trained

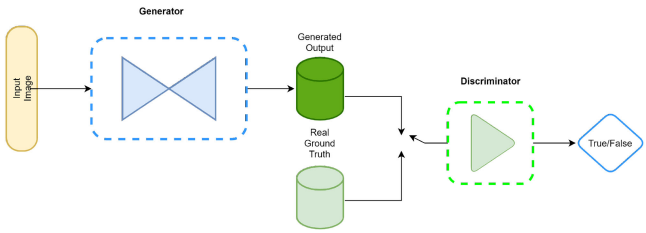


Fig. 2. A generic example of a GAN architecture. Here the generator is a neural network outputting a synthesized image, making it similar to the real ground truth images. Both the generated data and ground truth are fed to the discriminator to train to scrutinize between real and fake data.

with a noise sample from a particular distribution. Later with the advent of conditional GANs [27], [28], it became possible to capture even better representations, by rendering both the generator and the discriminator networks as class conditionals. Conditional GANs showed good performance translating data from one domain to another [27], [28], thus being appropriate for semantic segmentation. Fig. 2 illustrates the generic example of a GAN architecture. There have been subsequent work in using multiple discriminators, especially in [29], where multiple discriminators are used to generalizing the generator training, also in [30], where multiple discriminators are used to scrutinizing different low dimensional projections of the generator output. In [31], multiple discriminators are used to providing additional constructive feedback to the generator to produce an output closer to the original distribution of the ground truth data.

In this paper, we design a novel architecture to achieve joint superresolution and segmentation, simultaneously. By superresolving the OCT scans, we aim to enhance the layer boundaries and improve the layer segmentation. To the best of our knowledge, this is the first time a joint task of superresolving and segmentation is conducted on the OCT images. The goal is to semantically segment and superresolve the segmentation in a joint optimization framework by utilizing our proposed multi-stage multi-discriminatory generative adversarial network (MultiSDGAN). The major contributions of this manuscript are,

1) *Joint optimization of two objective functions:* Our novel architecture is a parameterized and scalable GAN-based domain translation model that learns to increase the medical image resolution from low to high by a factor of four and learn to segment the retinal layers.
2) *Multi-Stage:* We take into account multiple stages of output from different layers of the network, not just the final layer, to increase the granularity of the generator from a partial-view to a full-view of the domain translation task (input/output distributions).
3) *Multi-Discriminator:* Each intermediary output from the multi-stage is subjected to multiple discriminators for greater scrutiny, instead of having to pick one, which generates multiple gradients for training the generator and addresses the trade-off between localization accuracy and the patch size.
4) *Added loss terms:* We explore the suitability of the Dice loss as an additional loss function to improve the overall segmentation performance. Along with the Dice loss, we also add the Structured Similarity Index Measure (SSIM) loss [32] to further evaluate the gain in improving the resolution of the output and also, their combined effect on both tasks.

III. Data Acquisition and Preprocessing

Participants are recruited on the basis of referrals to or current patients at the memory disorders clinic or geriatric clinic at the West Virginia University (WVU). All subjects have a complete eye exam by an ophthalmologist including visual acuity, intraocular pressure, pupillary reaction, and dilated fundus exam. The OCT of the macula and the optic nerve head are obtained using the Heidelberg Spectralis OCT (Heidelberg Engineering Inc., Heidelberg, Germany).

We initiated data collection with normal aging patients (age: 55+) that we had easier access to and also to summarize the normal aging category, which is the dataset employed in this manuscript. The Ophthalmology Department at the WVU medicine provided the OCT images of 55 subjects, each having 19 scans and six subjects had one extra OCT. In total, there are 1,051 images in the dataset. These are 2-D scans, each group of 19 constitute one 3-D scan of the macula. These scans were obtained from the Ophthalmology Department, West Virginia University via the Infinitt software. For the purpose of this task, each image was meticulously labeled for the 7 innermost layers by two experts at the Ophthalmology and the Rockefeller Neuroscience departments. They are Internal Limiting Membrane (ILM), RNFL, GCL, Inner Plexiform Layer (IPL), Inner Nuclear Layer (INL), Outer Plexiform Layer (OPL), and Outer Nuclear Layer (ONL). Finally, all patient data was de-identified prior to analysis. This study was approved by the WVU Institutional Review Board (IRB) and ethics committee (the detailed protocol is approved under the study ID: 1910761036).

A. Data Preparation

One of the first and foremost requirements of any deep learning task is the availability of a dataset of sufficient size. A dataset without enough data could lead to overfitting. Thus the model won't be trained robustly. To alleviate such an issue, various data augmentation techniques were employed. The augmentation techniques were horizontal flip, spatial translation, and rotation [9]. Also, to increase the dataset size synthetically, we used a moving crop window approach of size 224x224, which was moves on the image horizontally with 75% overlap.

One of the persistent problems of OCT images is the presence of speckle noise. This leads to corruption of the boundary edges of the retinal layers, which would make detection and segmentation of the layers difficult for the neural network. To address this problem, we applied a 3x3 median filter to the whole images in our dataset. Afterward, an unsharpening mask was applied to enhance and make the boundary edges more prominent [33]. The above pre-processing operations were applied to the whole dataset before conducting model training and cross validation.

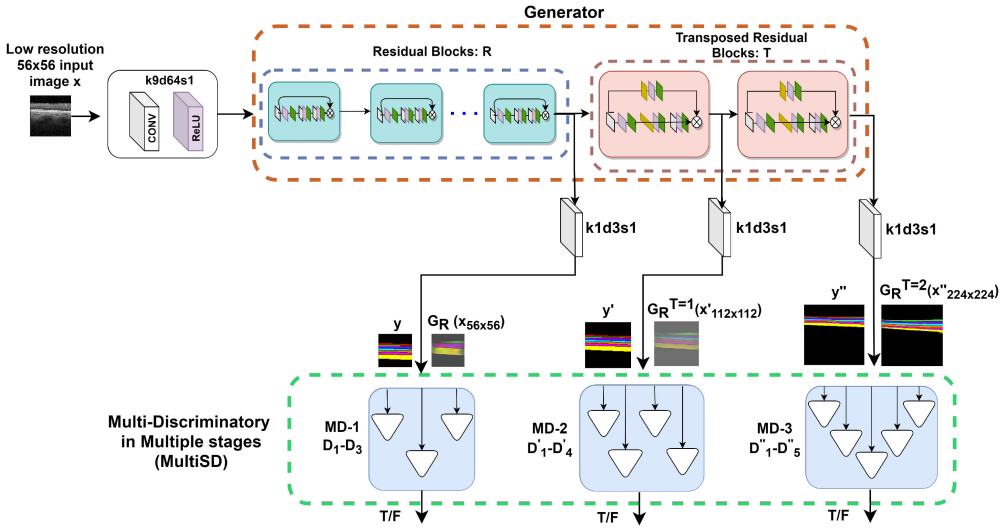


Fig. 3. The Multi-Stage Multi-Discriminatory GAN (MultiSDGAN) architecture is used for superresolution and segmentation. In this network, the input image of size 56×56 is fed to the generator G which consists of several residual blocks and transposed blocks. G outputs generated super-resolved label of size 224×224. The multi-discriminatory modules contain multiple networks inside them that provide scrutiny at different patch levels and are trained on both generated super-resolved labels and the ground truth high-resolution labels (k: kernel size, d: kernel depth, and s: stride).

IV. Methodology

The baseline of this work is a generative adversarial network, it contains two subnetworks aptly named the generator and the discriminator. In this paper, we proposed the use of multiple discriminators instead of a single one, each with different architecture and contribution, at multiple stages of the proposed network forcing the generator to match the full-distribution of the data by generating a series of gradients from individual discriminators, as additional constraints, in each iteration of training. Fig. 3 demonstrates the proposed MultiSDGAN domain translation architecture with a generator, G, including two components: a) R, for feature extraction, and b) T, for superresolution. Fig. 3 also demonstrates the multi-discriminatory modules in multiple stages to guide the generator to achieve higher granularity and scrutiny in the domain translation task. Therefore, we anticipate that the generator will learn a more comprehensive transformation from the input raw OCT domain distribution to the output segmentation labels. In the following sections, we will explain different components of the architecture proposed in this paper.

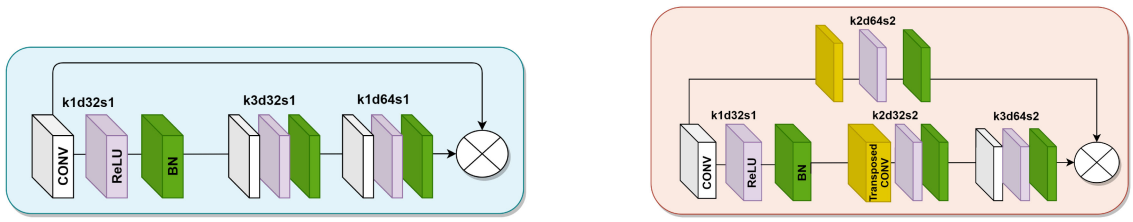
A. Multi-Stage Generator

The purpose of the generator in our framework is twofold, translating the OCT images from the original domain to a segmented domain and upscaling it to a higher resolution. In addition, the generator is also involved in synthesizing segmented images at different intermediary layers, which are in turn passed to the Multi-Discriminatory module.

1) ResNet With Bottleneck Blocks & Transposed Convolution: ResNet is a popular deep learning convolutional neural network-based architecture, which has achieved one of the top performances in the Imagenet competition [11]. This architecture popularized the use of skip connections in convolutional layers between the input and output. The skip connections help avoid vanishing gradients' problems, and it enables the stacking of several layers of convolution. ResNet is the baseline in SR-GAN architecture [26], which is an effective GAN-based superresolution architecture.

In this work, as we are also attempting joint superresolution and segmentation, inspired by [26], we adopt and modify ResNet as our generator architecture. The generator that is employed in this architecture has two major parts. The first part being the feature extractor, and the second part entails superresolving the images to a certain scale. In the feature extraction part, a bottleneck block is used, which is inspired by the original ResNet paper [11]. The original residual block idea was two consecutive 3×3 convolutional layers, with a skip connection between the input and output. However, a computationally more efficient solution, called the bottleneck, was also formulated in [11]. The bottleneck block also has a skip connection from the input to the output. As seen in Fig. 4(a), the bottleneck block consists of three convolutional operations with the kernel sizes of 1×1, 3×3, and 1×1. The first 1×1 convolution is to reduce the feature map depth so that the number of parameters to compute goes down and the overall model acts more efficiently. The last 1×1 convolution is to increase the feature map depth to the original size so that when combining the input to the residual block and the output there is no size mismatch. In between those two, there lies the 3×3 convolution to extract features. A skip connection is added between the input to the overall block and the output. With all the aforementioned modules combined, this bottleneck block is formed.

As the entirety of our network is complex and would require extensive computation, we decided to opt for the bottleneck block, which would lessen the feature map depth resulting in less complexity. We picked the number of the bottleneck blocks in the first part of the generator to be 30



(a) Bottleneck Residual Block (b) Transposed Bottleneck Residual Block

Fig. 4. Two different types of residual blocks. (a) is a typical bottleneck block from [11], with a skip connection from the input to the output. The 1×1 convolutions are used to decrease or increase the depth of the feature maps to lead to more efficient computation. (b) is a variation of the bottleneck block with transposed convolution block in the middle instead of a typical convolution block. The purpose of this bottleneck design is to spatially upscale the feature maps (k: kernel size, d: kernel depth, and s: stride).

via trial & error and based on the trade-off between the gains in the segmentation performance and the computational costs.

To achieve the superresolution and keep the continuity with the residual bottleneck block, a transposed bottleneck block was designed to be added to the generator. The structure of this block is also based on the bottleneck's principle, with a transposed convolutional layer replacing the conventional convolutional layer to increase the spatial resolution of the images. As shown in Fig. 4(b), a 2×2 transposed convolution filter with a stride of 2 replaces the conventional convolutional layer between the two 1×1 convolutional layers. The transposed convolution upscales the feature map resolution, bringing a mismatch between the residual block's input and output spatial resolution. This would hinder the residual skip connection, as the mismatched input and output feature maps cannot be combined with unequal spatial resolution. To solve this issue, a transposed convolution filter is also added to the skip connection to equal the original input resolution to the output one. So to summarise, a feature map entering this transposed bottleneck block would go through the first 1×1 convolutional layer, decreasing the depth.

Afterward, the feature map is put through the transposed convolution layer, which would essentially upscale the resolution of the feature map by a factor of 2. The output is then put through another layer of 1×1, this time to increase the feature map depth back to its original value. To keep up the tradition of residual blocks, skip connections are also applied here. As there is a height and width mismatch between the input and output, there is an additional transposed convolution layer in the skip connection part. This makes the input the same size as the output, thus enabling the combination between them.

2) Multi-Stage (MS): One important feature of our generator is its multi-stage output, which is basically extracting outputs from different intermediatory layers of the network, rather than only the final layer as suggested in [34]. As illustrated in Fig. 3, the final output is $3 \times 224 \times 224$, which is 4 times the size of the input $3 \times 56 \times 56$. Along with the final output, we extract two sets of feature maps from the previous two layers. As those feature maps have an increased depth, they are subjected to a 1×1 convolution to reduce the depth size to 3, to generate an RGB image. Losses are calculated for each of them, and they are back-propagated to train and update generator's weights. In standard convention, the loss is calculated from the final feature map. Even though the final feature map is derived via several convolutional operations from the previous maps, features extracted in the intermediatory layers may not be present in the final feature map. Thus, in our multi-stage approach, feature maps from layers other than the final output layer are also considered. Our multi-stage approach creates the opportunity to incorporate feature granularity information from different parts of the network.

B. Discriminator

The main purpose of the discriminator is to classify between the generated outputs and the ground truths. In this work, multiple discriminators are being used to enhance the discriminatory aspects of the GAN. Each of these discriminators is a PatchGAN [27], in which a convolutional neural network classifies an image as fake or real by focusing on penalizing it at the scale of local image patches of size $N \times N$. PatchGAN is conducted across the image convolutionally and generates a True/False(T/F) decision by averaging all the patch responses, assuming independence between pixels separated by more than the diameter of each $N \times N$ patch. The patch size is a parameter that determines capturing the spectral vs. spatial dimensions of the underlying image.

The PatchGAN classifier is described in Fig. 5. The input image of $3 \times H \times W$ is entered into the network, where 3 is the number of channels. The network has several blocks which consist of a convolution layer followed by ReLu [35] and a batch normalization layer. Each of these blocks has a kernel size of 3×3 with a stride of 2. These blocks are repeated N times, where the value of N denotes the patch size that is utilized. For an image size of $3 \times H \times W$, the patch size will be of $1 \times H/2^N \times W/2^N$.

C. Multi-Discriminatory (MD) Modules

Theoretically, the generator and the discriminator are required to learn and grow together and with a similar pace so that the generator can flawlessly match the distribution of data [10]. Though in many cases, the generator tends to be left behind without useful gradients for further training; Therefore, it stops improving the quality of the synthesized images and has to saturates early to avoid declining [36]. We aim to alleviate

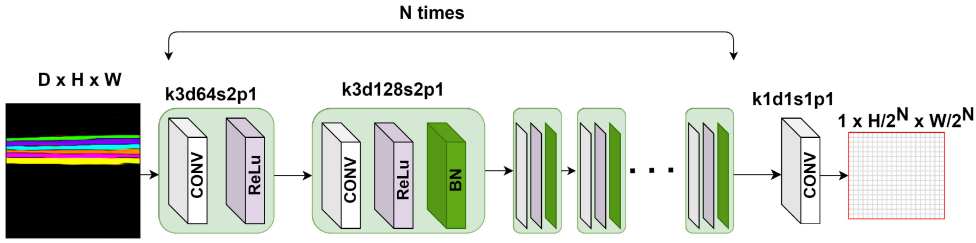


Fig. 5. The basic discriminator architecture, with N repeated blocks of convolutions, ReLu activations, and batch normalizations.

this issue by incorporating MD and also in multiple stages of segmentation label generation.

This module is inspired by [29], [30] and [31]. In these works, multiple discriminators are deployed to allow more inspection over the generator's output, resulting in the generator getting a more constructive feedback and thus being trained in a more generalized manner. Specifically, in [30], different projections of the generator output are fed to different discriminators, so each of the discriminators is scrutinizing different aspects of the output data. In our work, the goal of this module is to implement added scrutiny in various resolutions (stages of the network). Each of the feature maps and its corresponding ground truth will be discriminated using several different patch sizes, rather than just one. This allows the same image, be it ground truth or the generated, to be learned by focusing on localizations of those images with various sizes (rather than learning to differentiate between the ground truth and the generated images by only seeing the entire image as a whole). The intuition is that different parts of the image could have information denoting whether it is artificially generated or is real (ground truth). Thus we design a parameterized MD module with different patch sizes and deem it to be more effective (adding additional layers of inspection and capturing the distribution of the data more comprehensively).

In this work, the MD modules in each stage consist of multiple PatchGANs (shown in Fig. 5), each of which with a different number of N. The value of N determines the patch size by which the discriminators scrutinize the image. N is considered as a parameter to construct multiple discriminators. Each captures specific spectral and spatial dimensions in the input image to strengthen the generator training toward a more balanced GAN training. The output from the generator is fed to the MD modules, where it passes through all of the discriminators, simultaneously. All the discriminators with different patch sizes ($N \times N$) learn to differentiate between the generated image and the ground truth patches.

D. Multi-Stage MD (MultiSD)

In this work, the MDs are further implemented at different superresolution stages (i.e., transposed convolution layers) of the generator providing a multi-scale loss function for both the generator and the MD modules to achieve a more effective and stable segmentation outcome. In this way, instead of a single discriminator looking at the whole generated output, multiple discriminators see through various patchsizes in multiple super-resolution stages.

As shown in Fig. 3, we implemented the MD modules in 3 stages ending in the final layer of our proposed MultiSDGAN architecture to investigate the effectiveness of the framework. The architectures of the MD modules fed by multiple layers of the generator in various stages (i.e., MultiSD) are detailed in Tables I, II and III, respectively. Tables I reports the specifications of the first stage MD module, which is fed by the final output of the residual blocks from the first stage of the generator, $G_R(\mathrm{x}_{56\times56})$, with no up-scaling. As shown in Fig. 5, starting from $N = 1$, it creates a 1-layer patchGAN discriminator with the patch size of $H/2$ and $W/2$ (i.e., 28×28). More discriminators are constructed and added to the MD module by increasing N while H and W are divisible to 2^N. In this stage, since x is of size 56×56, the module includes 3 patchGANs with the resulting patch sizes of $1 \times H/2^N \times W/2^N$, where $N = 1, 2, 3$ (the 3 rd one with the minimum patch size of 7×7). In the same way, Tables II and III report the specifications of the 2nd- and 3rd-stage MD modules fed by the generator output after one and two transposed residual blocks, $G_R^{T=1}(\mathrm{x}'_{112\times112})$ and $G_R^{T=2}(\mathrm{x}''_{224\times224})$, including 4 and 5 patchGANs (ending with the minimum patch size of 7×7), respectively.

In this way, MD modules in multiple stages are combined into our proposed framework of MultiSDGAN. Note that, we selected two layers of upsampling in this work but one can increase those layers depending on their task in hand. Also, while some of the discriminators in different stages have the same resolution, e.g., D_1, D'_2, and D''_3, they process different parts of the input images and use different kernel stacks and therefore, operate complementarily.

E. Loss Function

For this work, besides the two common loss terms used for conditional GANs, namely, the adversarial loss and the reconstruction loss, two loss functions are opted to be used in the training purposes to optimize the MultiCDGAN parameters for the tasks of segmentation and superresolution. They are the Dice loss and the SSIM loss. The combination of these four loss terms contributes to the backpropagation and weight updates during the MultiSDGAN model training. Specifically, we evaluate and compare the effectiveness of incorporating the Dice loss and the SSIM loss individually and together for the task in hand in the discussion section. In the following sections, we will elaborate on each loss term in our proposed MultiSDGAN framework.

TABLE I
SPECIFICATIONS OF THE 1ST STAGE MD MODULE

Layer Name	Kernel	Stride	D_1 Output	D_2 Output	D_3 Output
conv2d_1, Leaky_ReLu	64×3 × 3	2	64×28 × 28	64×28 × 28	64×28 × 28
conv2d, BN, Leaky_Relu	128×3 × 3	2	-	128×14 × 14	128×14 × 14
conv2d, BN, Leaky_Relu	256×3 × 3	2	-	-	256×7 × 7
conv2d, BN, Leaky_Relu	V×3 × 3	1	128×28 × 28	256×14 × 14	512×7 × 7
conv2d	1×3 × 3	1	1×28 × 28	1×14 × 14	1×7 × 7

The list of layers, the operations including the kernel size and stride, and also the final output shape is reported for the discriminators, D_1 to D_3 presented in this module.

TABLE II
SPECIFICATIONS OF THE 2ND STAGE MD MODULE

Layer Name	Kernel	Stride	$D_1^{'}$ Output	$D_2^{'}$ Output	$D_3^{'}$ Output	$D_4^{'}$ Output
conv2d_1, Leaky_ReLu	64×3 × 3	2	64×56 × 56	64×56 × 56	64×56 × 56	64×56 × 56
conv2d, BN, Leaky_Relu	128×3 × 3	2	-	128×28 × 28	128×28 × 28	128×28 × 28
conv2d, BN, Leaky_Relu	256×3 × 3	2	-	-	256×14 × 14	256×14 × 14
conv2d, BN, Leaky_Relu	512×3 × 3	2	-	-	-	512×7 × 7
conv2d, BN, Leaky_Relu	V×3 × 3	1	128×56 × 56	256×28 × 28	512×14 × 14	512×7 × 7
conv2d	1×3 × 3	1	1×56 × 56	1×28 × 28	1×14 × 14	1×7 × 7

The list of layers, the operations including the kernel size and stride, and also the final output shape for the four discriminators in this module, $D_1^{'}$ to $D_4^{'}$, is reported.

TABLE III
SPECIFICATIONS OF THE 3 RD STAGE MD MODULE

Layer Name	Kernel	Stride	$D_1^{''}$ Outp.	$D_2^{''}$ Outp.	$D_3^{''}$ Outp.	$D_4^{''}$ Outp.	$D_5^{''}$ Outp.
conv2d_1, Leaky_ReLu	64x3x3	2	64x112x112	64x112x112	64x112x112	64x112x112	64x112x112
conv2d, BN,Leaky_Relu	128x3x3	2	-	128x56x56	128x56x56	128x56x56	128x56x56
conv2d, BN, Leaky_Relu	256x3x3	2	-	-	256x28x28	256x28x28	256x28x28
conv2d, BN, Leaky_Relu	512x3x3	2	-	-	-	512x14x14	512x14x14
conv2d, BN, Leaky_Relu	512x3x3	2	-	-	-	-	512x7x7
conv2d, BN, Leaky_Relu	Vx3x3	1	128x112x112	256x56x56	512x28x28	512x14x14	512x7x7
conv2d	1x3x3	1	1x112x112	1x56x56	1x28x28	1x14x14	1x7x7

The list of layers, the operations including the kernel size and stride, and also the final output shape is reported for the five discriminators in this module, $D_1^{''}$ to $D_5^{''}$.

1) Adversarial Loss: This is a widely used loss function to train GANs, which is applied for both the generator, G, and the discriminator, D. This loss function by itself puts the two networks in a competitive position, with each trying to outdo the other (i.e., Minimax game).

For training the discriminator, the conventional loss function is:

$$L_{GAN}(G, D) = E_{\mathrm{y}}[logD(\mathrm{y})] + E_{\mathrm{x}}[log(1 - D(G(\mathrm{x}))], \quad (1)$$

and the loss for training the generator is:

$$L_{GAN}(G) = E_{\mathrm{x}}[log(D(G(\mathrm{x})))], \quad (2)$$

where $\mathrm{x} = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}$ is the input image and $\mathrm{y} = \{y_1, y_2, \ldots, y_i, \ldots, y_n\}$ denotes the actual ground truth label. Here the i^{th} values of x and y denotes the individual pixel values of the input image and ground truth, respectively, with n being the total number of pixels.

The adversarial loss for the discriminator consists of two terms, which essentially compete against each other. The first term is to train the discriminator to detect the ground truth labels, which can be termed as real whereas the second term is to train it to detect the generated ones, termed as fake. As the training process goes on, one of the loss terms tends to dominate the other, signifying whether the discriminator is learning to identify the real labels from the generated ones.

As this work uses three MD modules in three stages (i.e., MultiSD), each MD module will have its own adversarial loss given the inputs $\mathrm{x}_{56\times56}$, $\mathrm{x}'_{112\times112}$, and $\mathrm{x}''_{224\times224}$, each of which further breaking down to its individual discriminators operating on different patch sizes (i.e., MultiSD: {D s, D ′ s, D ″ s}). The discriminator adversarial loss in each of the MD modules

at each stage can be defined as:

$$L_{GAN}(G, D_i) = E_{\mathrm{y}}[logD_i(\mathrm{y})] + E_{\mathrm{x}}[log(1 - D_i(G(\mathrm{x}))), i = 1, \ldots, 3, \tag{3}$$

$$L_{GAN}(G, D_i') = E_{\mathrm{y}'}[logD_i'(\mathrm{y}')] + E_{\mathrm{x}'}[log(1 - D_i'(G(\mathrm{x}'))], i = 1, \ldots, 4, \tag{4}$$

$$L_{GAN}(G, D_i'') = E_{\mathrm{y}''}[logD_i''(\mathrm{y}'')] + E_{\mathrm{x}''}[log(1 - D_i''(G(\mathrm{x}''))], i = 1, \ldots, 5. \tag{5}$$

All these losses are combined and are differentiated to be backpropagated. The final discriminator adversarial loss for the proposed MultiSDGAN architecture is defined as:

$$L_{GAN}(G, MultiSD) = \sum_{i=1}^{3} L_{GAN}(G, D_i) + \sum_{j=1}^{4} L_{GAN}(G, D_j') + \sum_{k=1}^{5} L_{GAN}(G, D_k''). \tag{6}$$

In our proposed MultiSDGAN framework, the generator aims to ensure that each segmentation label generated for the input OCT deceives all of the MultiSD modules of various resolutions and scales of that sample. Hence, the final generator adversarial loss is defined as:

$$L_{MultiSDGAN}(G) = \sum_{i=1}^{3} E_{\mathrm{x}}[log(D_i(G(\mathrm{x})))] + \sum_{j=1}^{4} E_{\mathrm{x}}[log(D_j'(G(\mathrm{x}')))] + \sum_{k=1}^{5} E_{\mathrm{x}}[log(D_i''(G(\mathrm{x}'')))]. \tag{7}$$

2) Reconstruction Loss: As seen from the work [27], an additional constraint is used named the reconstruction loss. This loss function is measured between the generator output and the ground truth labels. This constraint aims to reduce the error between the generated outputs and the ground truths, which would lead the generator to be trained better and generate images with improved aesthetics. From [27], an $L1$ loss is selected as the reconstruction loss. This is measured between the generated output $G(\mathrm{x})$ and the ground truth label y. The $L1$ loss is given as:

$$L_{L1}(G) = \|\mathrm{y} - G(\mathrm{x})\|_1. \tag{8}$$

The $L1$ loss measures the distance between the generated label $G(\mathrm{x})$ and the ground truth label y.

3) Dice Loss: Since, the proposed model is used for a segmentation task, we aim to incorporate a loss that specifically targets optimizing the performance of the model for that task. This loss function originates from the semantic segmentation metric called the Dice coefficient. Usually, a Dice coefficient is a metric measure, which entails values within [0,1] range, where the higher value demonstrating better segmentation. Taking the additive inverse of said metric gives us the Dice loss [12]. The Dice loss is given as:

$$L_{Dice}(G) = 1 - \frac{2\sum_{i=1}^{n} G(x_i)y_i}{\sum_{i=1}^{n} G(x_i)^2 + \sum_{i=1}^{n} y_i^2}, \tag{9}$$

where $G(x_i)$ is the generated label and the y_i is the ground truth label of pixel x_i, and n being the total number of pixels. The task of the network is to minimize this function so that the generator can successfully segment the image, which results in minimal Dice loss when calculated against the ground truths. Furthermore, this loss function acts as an additional reconstruction loss to further emphasize and improve the quality of the generator output. This form of the Dice loss can be differentiated yielding the gradient (10):

$$\frac{\partial L_{Dice}(G)}{\partial G(x_i)} = 2\left[\frac{y_i(\sum_{i=1}^{n} G(x_i)^2 + \sum_{i=1}^{n} y_i^2) - 2y_i(\sum_{i=1}^{n} G(x_i)y_i)}{(\sum_{i=1}^{n} G(x_i)^2 + \sum_{i=1}^{n} y_i^2)^2}\right] \tag{10}$$

4) SSIM Loss: The SSIM is an index metric to determine the perceived quality of an image. It is used to compare the quality between two images, where the ground truth is undistorted and noise free. The previous three objective functions are employed to suitably train and improve the segmentation performance of the network. As in this paper we are also doing the task of superresolution, we are opting for the addition of another loss function termed the SSIM loss. This loss function will help the network generate outputs similar to the ground truth superresolved images in terms of perceived quality. The equation for the SSIM is given as:

$$SSIM(G(\mathrm{x}), \mathrm{y}) = \frac{(2\mu_{G(\mathrm{x})}\mu_{\mathrm{y}} + c_1)(2\sigma_{G(\mathrm{x}),\mathrm{y}} + c_2)}{(\mu_{G(\mathrm{x})}^2 + \mu_{\mathrm{y}}^2 + c_1)(\sigma_{G(\mathrm{x})}^2 + \sigma_{\mathrm{y}}^2 + c_2)}. \tag{11}$$

Here, $\mu_{G(\mathrm{x})}$ and μ_{y} are the mean of the pixels for the generated output $G(\mathrm{x})$ and ground truth y, respectively. Similarly, $\sigma_{G(\mathrm{x})}$ and σ_{y} are variances of $G(\mathrm{x})$ and y, respectively. Also, $\sigma_{G(\mathrm{x}),\mathrm{y}}$ is the covariance between $G(\mathrm{x})$ and y. c_1 and c_2 are constant values to compensate for weak denominators. This index can be broken down into two components:

$$l(G(\mathrm{x}), \mathrm{y})) = \frac{(2\mu_{G(\mathrm{x})}\mu_{\mathrm{y}} + c_1)}{(\mu_{G(\mathrm{x})}^2 + \mu_{\mathrm{y}}^2 + c_1)}, \tag{12}$$

and

$$c(G(\mathrm{x}), \mathrm{y}) = \frac{(2\sigma_{G(\mathrm{x}),\mathrm{y}} + c_2)}{(\sigma_{G(\mathrm{x})}^2 + \sigma_{\mathrm{y}}^2 + c_2)}, \tag{13}$$

where the $l(G(\mathrm{x}), \mathrm{y})$ and $c(G(\mathrm{x}), \mathrm{y})$ depicts the luminance and contrast, respectively. This SSIM index value ranges from [0,1], where the value closer to 1 denotes better quality. We can rewrite this metric as a loss function for the network as:

$$L_{SSIM} = 1 - SSIM(G(\mathrm{x}), \mathrm{y}), \tag{14}$$

where we are taking the additive inverse of the SSIM index with a view to use it as a loss function [37] where the derivative of this loss function is:

$$\frac{\partial L_{SSIM}(G(\mathrm{x}),\mathrm{y})}{\partial G(\mathrm{x})} = -\left(\frac{\partial l(G(\mathrm{x}),\mathrm{y})}{\partial G(\mathrm{x})} c(G(\mathrm{x}),\mathrm{y}) + l(G(\mathrm{x}),\mathrm{y}) \frac{\partial c(G(\mathrm{x}),\mathrm{y})}{\partial} G(\mathrm{x}))\right) \tag{15}$$

After introducing all the loss terms in the above, the total loss function for the generator stands as:

$$L_{Generator} = L_{MultiSDGAN}(G) + \lambda L_{L1}(G) + \alpha L_{Dice}(G) + \beta L_{SSIM}(G), \tag{16}$$

where λ, α and β are coefficients of the loss functions respectively, which control the relative importance of each corresponding loss functions. These values are chosen empirically to get the best possible performance, thus we employed grid search [38], where we tested a range of combinations of values on a smaller validation set and empirically chose $\lambda = 100$, $\alpha = 1$ and $\beta = 1$.

V. Experiment

In this paper, the experiments are conducted according to the baseline architecture in Fig. 3. After the preprocessing and augmentation, the dataset size was increased. Then, we applied cross validation (10-CV) to estimate the generalization performance of the compared segmentation models on the unseen data. In this way, the whole set of available images were divided into ten equal parts. In every iteration with permutation, nine parts out of 10 were combined and considered to be the image train-set and the remaining one part is considered to be the test-set (i.e., there was no overlap between the train- and test- sets).

All of the training was done on a system with two GeForce GTX TITAN X GPU. The Adam optimizer [39] was used for training, with a learning rate of 0.0001 for both the generator and the discriminators. The multi-stage and multi-discriminatory model is the primary one used for experimentation, and as ablation, we tried different combinations where the multi-stage aspect or the multi-discriminatory aspect was removed separately or both of them were removed together. Furthermore, we analyzed Dice loss as an additional cost function and the impact of superresolution modules by running the same models with and without them. All of the experiments were run for 100 epochs for each fold.

VI. Results

The metric used to check the quality of the superresolved segmented image is the Dice coefficient within the [0,1] range, where a higher value denotes better quality. The Dice coefficient is chosen over pixel accuracy because the latter does not consider class imbalance. Furthermore, because there is a dominant portion of the background in our images, there is a risk that the pixel accuracy as a metric would lead to an erroneous conclusion. On the other hand, for the evaluation of superresolution, the SSIM index is employed as in [32] and [40]. This is a metric that evaluates the degradation of images from a perceptual point of view. In terms of superresolution, as we are upscaling the image from low-resolution to a high-resolution, we compare the perceived quality of the image for a ground truth high-resolution image.. This index is ranged from [0,1], where a higher value indicates less degradation. We use this metric to further evaluate the quality of the superresolved segmented images. In addition to the Dice Coefficient and SSIM, we also include the $L1$ distances in the result section. In the result comparison tables, we put the mean and standard deviation of the Dice coefficient, the SSIM, and the L1 over 10 fold cross validation (10-CV).

We employed the cross validation error rates per iteration and per each comparative model (10 error values in total per model) as the statistical populations to investigate the statistical significance of the results with t-test [41] and to solidify the effectiveness of the addition of our proposed model components to improve the segmentation results. We conduct the test to compare the error rates associated with adding (vs. not adding) each proposed component within the overall architecture. The null hypothesis is that there is no decrease (increase) in the 10 CV error values per comparative model. The significance levels that their associated p-values are. 05 or better. 01 and below ($p \leq .05$ or $\leq .01$) are reported as significant improvement by adding each component. Beside the 10-CV averaged results along with their standard deviation, we also present the p-value associated with each t-test.

A. The Impact of MultiSDGAN Architecture on the Generator's Performance

This section investigates the impact of various components of the MultiSDGAN architecture on the trained generator's performance (i.e., Segment). In particular, we evaluate and compare the addition of 1) superresolution (SR), that is, the addition of the transposed convolution layers, 2) multi-discriminatory (MD) that is increasing a single patchGAN discriminator at the last layer to multiples ones with various patch sizes, and then, 3) MultiSD, that is including those MD modules in multiple stages against the generator. Finally, as an ablation study, we remove each stage and compare how the addition of intermediate stages affects the model performance. Table IV reports the improvements on the results by adding the above components to the model.

From Table IV, it can be observed that our best proposed model architecture, namely multiSDGAN, increased the Dice coefficient by 44.24% of relative improvement with the $p < .01$, which indicates that extracting feature maps in multiple scales by the MultiSD modules can lead to a substantially improved performance that is also statistically significant. Also, it can be observed that our trained generator with two layers SR and MultiSD improves the one layer SR and MultiSD with 16.18% relative improvement, and this improvement is statistically significant with $p < .05$.

Regarding the SSIM index, by comparing the results in Table IV, it is observed that the inclusion of multiSD provides a major relative improvement of SSIM values by 34.09%, which is also found to be statistically significant ($p < .01$).

TABLE IV
ADDITIVE IMPACT OF SUPERRESOLUTION (SR), MULTI-DISCRIMINATORY (MD), AND MULTI-DISCRIMINATORY IN MULTIPLE STAGES (MUTISD) ON THE PERFORMANCE OF THE GENERATOR (SEGMENTOR)

Model		Dice Coefficient	SSIM	L1
Generator without SR	single discriminator, D_1	0.8235±0.0083	0.8463±0.009	0.344±0.0012
	MD module, D_i, i=1..3	0.8359±0.0092	0.8588±0.0118	0.047±0.0019
Generator with one layer SR	single discriminator $D_1^{'}$	0.8327±0.013	0.8519±0.0083	0.336±0.0015
	single-stage MD module $D_i^{'}$, i=1..4	0.8529±0.0107	0.8582±0.0065	0.042±0.0014
	MultiSD D_i,$D_j^{'}$, i=1..3 , j=1..4	0.8826±0.0059	0.8794±0.0051	0.033±0.001
Generator with two layers SR	single D module $D_1^{''}$	0.8440±0.0039	0.8547±0.0028	0.295±0.0012
	single-stage MD module $D_i^{''}$, i=1..5	0.8658±0.0071	0.8603±0.0036	0.033±0.0013
	MultiSD D_i,$D_j^{'}$,$D_k^{''}$, i=1..3, j=1..4, k=1..5	**0.9016±0.004**	**0.8987±0.0051**	**0.021±0.0011**

TABLE V
EFFECT OF NUMBER OF DISCRIMINATORS IN THE MD MODULES OF THE MULTISDGAN: THE COMPARISON BETWEEN FIXED VS. ENSEMBLE MULTISD

Model		Dice Coefficient	SSIM	L1
Generator: 2-layer SR & MultiSD	D_i,$D_i^{'}$,$D_i^{''}$, i=1	0.8693±0.0063	0.8835±0.0058	0.0295±0.004
	D_i,$D_i^{'}$,$D_i^{''}$, i=1,2	0.8827±0.0064	0.8871±0.0054	0.026±0.004
	D_i,$D_i^{'}$,$D_i^{''}$, i=1..3	0.8945±0.0042	0.8942±0.0054	0.023±0.001
	D_i,$D_j^{'}$,$D_k^{''}$, i,j,k:1..3,..4,..5	0.9016±0.004	0.8987±0.0051	0.021±0.0011

The biggest relative improvement is made by transiting from single-stage MD to multiSD, relative improvement of SSIM values by 28.15%, where the difference between the architecture is basically the additional outputs extracted from intermediate layers. In terms of the L1 distance, by inclusion of MD, several fold improvement is noticed, and including MultiSD further improves the results with a relative improvement of 36.36% on top of MD. This improvement is also found to be statistically significant ($p < .01$). The results reported in Table IV demonstrate that introducing our proposed architecture produces a generator model that achieves a more effective and stable segmentation outcome.

B. The Effect of the Number of Discriminators

A major contribution in this study is the usage of multiple PatchGAN discriminators instead of a single one against the generator in the proposed MultiSDGAN architecture. Therefore, this section investigates the impact of the number of discriminators in the MD modules in the MultiSDGAN framework. Here we evaluate the best performed generator model in Table IV (i.e., ResNet with two layers SR (56x56 to 224x224)), which was trained against the maximum number of PatchGANs in the MD modules at each stage ending with one with the minimum patch size of 7×7 (3, 4, and 5 PatchGANs). Instead, we train that generator against a fixed number of PatchGANs, including using a single discriminator with a fixed patch size, or using 2, or 3 PatchGANs with varying patch sizes per MD module.

As reported in Table V, it is evident that employing the multi-discriminatory module improves the performance in both segmentation and superresolution aspects. Table V also demonstrates consistent improvements by adding more PatchGANs. As reported in Table V, our policy of adding maximum discriminators per stage significantly improves the results over a single discriminator per stage on the Dice coefficient by 23.68% relative improvement ($p < .01$). In terms of the $L1$ distance also, we can see a substantial relative improvement of 26.42% ($p < .01$).

Comparing the SSIM results, as reported in Table V, increasing from all three stages having a single discriminator to multiple discriminators improves the results by 12.65% relative improvement ($p < .05$). While the improvement is significant, there is not as much improvement made compared to the Dice coefficient. It is worth noting in this comparison, every architecture already includes the MultiSD. Therefore, the minor improvements in this current comparison might be because all of the comparative models employ the multi-stage component that has already led to the major improvements in the SSIM as reported in Table IV. Therefore, it can be concluded that the impact of multi-stage component is much more effective on the SSIM index compared to the addition of multi-discriminatory.

C. Evaluation of the Loss Function

In this section, we explore the effectiveness of adding the Dice and SSIM loss terms to the overall loss on the segmentation and superresolution performance, respectively. The main reason for employing the Dice loss was to improve the segmentation outcome. A higher value of the Dice coefficient entails better segmentation, thus we took the additive inverse of the Dice coefficient (9), and used it as an additional loss term. As the network is being optimized, it will try to reduce the Dice loss, thus effectively improving the segmentation. Therefore, different combinations were tried utilizing the Dice loss (9) as an additional constraint as well as without it. As the task in hand is a joint task of segmentation and superresolution, there is an element of image quality involved. As shown in (16), the SSIM loss is mainly used to determine the perceived quality of images. As we are taking low-resolution images and upscaling

TABLE VI
THE IMPACT OF ADDING THE DICE AND SSIM LOSSES TO THE TOTAL LOSS FUNCTION. ABLATION STUDY IS DONE WITH/WITHOUT THOSE ADDITIONAL CONSTRAINTS AND THE RESULTS ARE COMPARED

Model		Dice Coefficient	SSIM	L1
Generator: 2-layer SR & MultiSD	$L_{GAN}\&L_{L1}$	0.8567±0.0104	0.8813±0.0096	0.0304±0.001
	$L_{GAN}\&L_{L1}\&L_{Dice}$	0.8975±0.0062	0.8859±0.0042	0.023±0.0012
	$L_{GAN}\&L_{L1}\&L_{SSIM}$	0.8543±0.0049	0.8943±0.0077	0.0225±0.0009
	$L_G\&L_{L1}\&L_D\&L_S$	0.9016±0.004	0.8987±0.0051	0.021±0.0011

TABLE VII
EVALUATION OF THE GAIN BY THE ADDITION OF SUPERRESOLUTION WITH MULTISD

Model		Dice Coefficient	SSIM	L1
Low-resolution Generator (56x56⇒56x56)	Single disc.,D_1	0.8235±0.0083	0.8463±0.009	0.344±0.0012
	MD , D_i,i : 1..3	0.8359±0.0092	0.8588±0.0118	0.042±0.0019
High-resolution Generator (224x224≻224x224)	Single disc., D_1	0.8809±0.0061	0.8923±0.0045	0.133±0.0013
	MD , D_i,i : 1..5	0.9013±0.005	0.8993±0.0047	0.021±0.0009
Ours: Low-to-high (56x56⇒224x224)	MultiSD	0.9016±0.004	0.8987±0.0051	0.021±0.0011

them to high-resolution images, we want them to be as close to the ground truth high-resolution images in terms of quality. Thus here we test the effectiveness of the SSIM loss as another additive loss term.

Table VI reports the comparative results of the ablation study. We first removed the Dice and SSIM losses as constraints and compared the results. Then, only the Dice loss was added as the additional loss term. In another case, only SSIM loss is considered, and then, both of them are taken in as the additional constraints. The results in Table VI demonstrate that using the Dice loss improves the performance considerably by a relative improvement of 28.47% ($p < .01$) on the segmentation results. Similarly, the $L1$ distance also shows similar relative improvement of 27.63% ($p < .01$). Comparing the SSIM value, a small relative improvement of 4.46% was observed, which was not significant ($p > .05$). As discussed previously, this observation might be due to the fact that all of the comparative models employ the MultiSD components that already improved the SSIM values to those levels seen in Table IV.

Table VI also reports that the addition of only the SSIM loss (comparing the first row with the third row in Table VI), which did not provide any noticeable effect on the Dice coefficient (the segmentation performance of the model). Adding only the SSIM loss provided a small improvement on the SSIM measure of the evaluated model with ($p > .05$). In the case with both the Dice and SSIM loss terms added, the difference in terms of the Dice coefficient performance is substantial and significant ($P < .01$), and for SSIM, it is a smaller improvement but still significant ($P = .05$). Both loss terms separately and together improved the L1 measure significantly ($P < .01$) and adding both losses shown to be the most effective in improving the overall segmentation and superresolution performance.

D. Gains From Superresolution With MultiSD

In this work, the prime tasks of our designed network are to execute segmentation and superresolution in the same forward pass. During the training, the network weights are updated based on the losses encountered while doing both tasks. As mentioned as part of our data preparation (section III.A), the input images of 224×224 are downsized to 56×56 to be used as the lower resolution input. To test the true effectiveness of the superresolution aspect of the network, we remove it to test on simple segmentation. In this ablation study, we first evaluated segmentation of the low-resolution input images with the network without superresolution (i.e, 56×56 raw image to 56×56 segmented image). As shown in Table VII, besides the fact that it does not generate higher resolution segmented images, the performance in all measures is degraded.The transposed bottleneck blocks, are discarded from the network so that the model only serves as a segmentation network only (i.e.,the high resolution generator). From the results in Table VII, it is observed that the generator trained using our proposed low-to-high resolution (56×56 raw image to 224×224 segmented image) architecture with MultiSD improved the performance on all measures when compared with the high resolution generator with a single discriminator. The improvements were significant for the Dice and L1 measures ($p < .01$), which are impressive results but it was not significant for the SSIM measure ($.05 < p < .1$). The high resolution generator with an MD module performed similarly with no noticeable difference to when we apply our proposed & MultiSDGAN architecture on the low resolution data (for all measures, high p-values, $p > 0.5$, were generated). These results demonstrate the effectiveness of the MultiSDGAN model to conduct OCT domain transfer from low-resolution to high-resolution without any degradation in their resulting quality.

E. Comparison With the State-of-the-Art (SOTA) Methods

In this section, we conducted a comparative investigation to demonstrate the usefulness of our proposed model i.e., MultiSDGAN. In this way, we compared MultiSDGAN with a SOTA model for OCT segmentation based on the popular U-net module, which has a very competitive performance i.e., RelayNet [9]. MultiSDGAN and RelayNet are both based on supervised learning paradigm to provide more accurate segmentation outcomes and therefore, it would be a fair comparison. Though, this model do not involve superresolution module. Therefore, for a fair comparison, we used this model on our data for straightforward segmentation of input image sizes of 224×224. Though, we kept the superresolution module intact for MultiSDGAN, thus involving input images of 56×56. MultiSDGAN model superresolves them to 224×224 and simultaneously performs semantic segmentation. We also included a recent and competitive unsupervised segmentation model based

TABLE VIII
COMPARATIVE PERFORMANCE RESULTS AND COMPUTATION COSTS BETWEEN THE PROPOSED MULTISDGAN AND THE SOTA METHODS

dataset	Measure	MultiSDGAN	RelayNet	Unsupervised
WVU-OCT	Dice Coefficient	0.9016±0.004	0.8708±0.0017	0.7897±0.0015
	SSIM	0.8987±0.0051	0.8588±0.0118	0.7831±0.0018
	L1	0.021±0.0011	0.028±0.0006	0.1487±0.0024
	Time/Epoch (Sec)	185.56	117.74	21.56
Duke-OCT	Dice Coefficient	0.9075±0.0083	0.8874±0.009	0.7864±0.0015
	SSIM	0.9011±0.0096	0.8774±0.0010	0.7994±0.0019
	L1	0.0198±0.0092	0.025±0.0006	0.1268±0.0019
	Time/Epoch (Sec)	194.45	126.32	22.12

on a CNN architecture that jointly optimizes feature extraction functions and clustering functions [42]. Our goal was to performance comparison between MultiSDGAN vs. supervised and unsupervised SOTA segmentation models. Furthermore, We conducted an experiment on another publicly available dataset for a more thorough evaluation of our proposed MulltiSDGAN. The Duke SD-OCT publicly available dataset for DME patients was used for this comparison. This dataset consists of 110 annotated SD-OCT B-scan images of size 512×740. We used multiple preprocessing techniques such as: data augmentation and resizing. Then, we fed the labelled dataset to the comparative models for evaluation. Table VIII reports the performance results of the above-mentioned three models (MultiSDGAN, RelayNet [9], and unsupervised [42]) on the two datasets: 1) our dataset (WVU-OCT), and 2) the above-mentioned Duke-OCT dataset.

As reported in Table VIII, we observe that our trained generator with MultiSD outperforms the Relaynet and the unsupervised method with regards to the Dice coefficient metric with relative improvements of 23.84% and 49.52%, respectively. Both improvements found to be statistically significant (the former: $p < .05$ & and the latter: $p < .01$). Similarly in terms of the SSIM index, we can see substantial improvements of 34.42% and 48.93%, respectively ($p < .01$) (similarly for $L1$), which demonstrates the effectiveness of our proposed MultiSDGAN architecture for OCT image segmentation and superresolution task. Also, comparing the results of the comparative models on the WVU-OCT dataset vs. the Duke-OCT dataset demonstrates that MultiSDGAN consistently outperforms the other SOTA methods and the differences in performances were found to be statistically significant.

Table VIII also reports the computational costs in "Time per Epoch (in Sec)" corresponding to the three comparative models (MultiSDGAN, RelayNet [9], and unsupervised [42]) on the two employed datasets in this study. As reported in Table VIII, while MultiSDGAN and RelayNet generate more accurate results in comparison with the unsupervised method, they incur more computational costs in the model training phase. MultiSDGAN have the highest computational expense per epoch. Though, by incorporating a higher cost optimization framework, it consistently generates the most competitive results within the comparison. The Good news is that after training and during the testing phase, it takes several seconds for each of the compared models to perform the segmentation on an input scan.

VII. DISCUSSION

In this paper, our main goal was to achieve high quality superresolution and accurate semantic segmentation of OCT images, simultaneously. In this way, we proposed a GAN-based neural network architecture, MultiSDGAN, which has the additional capability of superresolution with multi-discriminatory in multi-stage for more focused training and added scrutiny. We achieved the multi-stage aspect by providing segmented images at the final and the intermediary layers. the main argument for doing this is to provide more clarity in the training process of the network, as the multi-stage would allow taking into account feature maps from different intermediate layers along with the final one, facilitating more robust training to achieve better quality. Each feature map shows a different abstraction of the network outputs, so a combined training allows the network to take into account features from different abstractions, and not only the final output. Experimentally, it was shown in Table IV that this approach is effective to improve the results.

Additionally, we utilized a multi-discriminatory module, where the output feature map is put through multiple discriminators. Each discriminator is a PatchGAN classifier, where networks determine whether an image is the ground truth or the generated one. With a maximum number of discriminators, each with a different patch size at each stage, the feature maps are being scrutinized more meticulously and cover the distribution of the input data more comprehensively. As part of the ablation study, we progressively add to the multi-discriminatory modules in our proposed segmentation network and tested the segmentation results. Adding to the multi-discriminatory components and in multiple stages proved to be significantly advantageous in improving segmentation and superresolution measures as shown in Table IV and Table V. We further discussed the effect of progressively adding the Dice and the SSIM loss to the overall loss function, shown in Table VI. from Table VI, we can see that the Dice loss alone performs better when compared against the solitary inclusion of SSIM loss, and the inclusion of both the Dice and the SSIM loss together improved over the solitary inclusion of the Dice and the SSIM losses.

We then measured the gains by using our proposed network architecture, namely MultiSD, that takes in low-resolution OCT scans from the patients and translates them into high-resolution and accurate segmentation of the retinal layers (Table VII). This enables us to work with low-resolution images, as the network will do the job of getting a high-resolution segmented output. The results from the low-resolution data would be as good if we used a high-resolution input image in the first place for segmentation purposes. This result is significant since the MultiSDGAN network architecture enables us to work with lower quality images that are common in the domain of medical imaging, including OCTs to generate the segmentation and improve the resolution of the images, simultaneously. In this way, besides lower costs of scans and their storage, we can shorten the duration of the OCT scans and minimize the occurrence of the

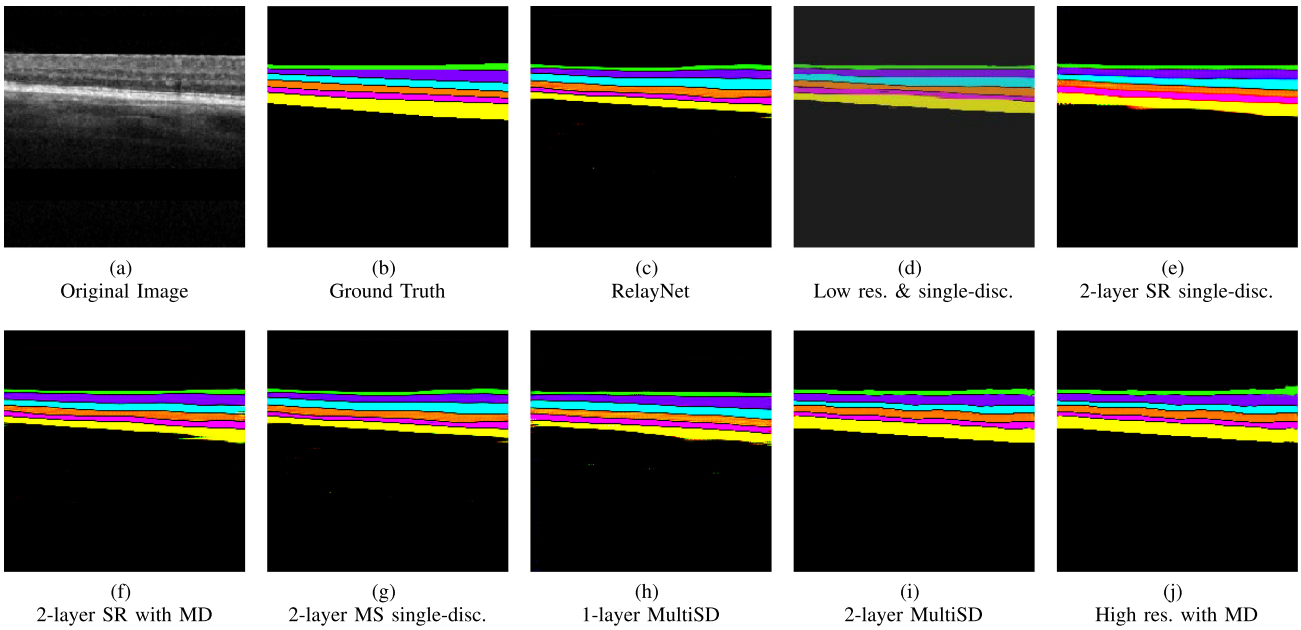


Fig. 6. Visual comparison of the additive impact of superresolution (SR), multi-discriminatory (MD), and multi-discriminatory in multiple stages (MutiSD) on the performance of the Generator (segmentor). Different parts of this figure are as follows: (a) the original input image, (b) ground truth segmented image, (c) The benchmark method, RelayNet (224×224), (d) low-resolution single-discriminator model (56×56), (e) 2-layer SR with single-discriminator model, (f) 2-layer SR model with MD, (g) Multi-stage discriminators with single-discriminator per stage, (h) the model with 1-layer MultiSD, (i) the model with 2-layer MultiSD, and (j) high-resolution MD model (224×224). It illustrates the improved segmentation results progressively as we add SR, MD, and MultiSD modules to the network.

eye movements that can introduce multiple types of artifacts to the resulting scans without any noticeable degradation. Also, our proposed network with superresolution (56×56 to 224×224) was several folds faster to train compared to the counterpart network, without superresolution, (224×224 to 224×224) leading to the comparable results.

Fig. 6 visually illustrates an example of an OCT input image Fig. 6(a), and its corresponding ground truth segmentation, Fig. 6(b) and shows the improved segmentation results progressively as SR, MD, and MultiSD modules are added to the network and in comparison to the benchmarks: Fig. 6(c), the benchmark method, RelayNet (224×224), Fig. 6(d), low-resolution single-discriminator model (56×56), Fig. 6(e), 2-layer SR with single-discriminator model, Fig. 6(f), 2-layer SR model with MD, Fig. 6(g), Multi-stage discriminators with single-discriminator per stage, Fig. 6(h), the model with 1-layer MultiSD, Fig. 6(i), the model with 2-layer MultiSD, and Fig. 6(j), high-resolution MD model (224×224). Comparing the results from our proposed MultiSDGAN framework against the established state of the art model for OCT segmentation, called the Relaynet, it is observed that our proposed model significantly outperforms the RelayNet model in terms of segmentation as well as perceived quality compared to the RelayNet. The results in Fig. 6 support our overall results on the OCT test-set in Tables IV-VII

VIII. Conclusion

In this work, we designed a generator architecture, entitled MultiSDGAN, which performed the dual tasks of semantic segmentation as well as superresolving the segmented images. To do this dual training, we designed the network with consecutive residual bottleneck blocks, which performs as the feature extractor, and then adding a transposed convolution block which performs as a mirror of the aforementioned residual bottlenecks only with the capability of upscaling the images. Another prime feature of the work is the addition of multi-discriminatory modules, which is basically discriminating generator outputs through different patch sizes simultaneously and in multiple stages. We also tested the Dice loss, an objective function originating from the Dice coefficient metric and the SSIM loss to capture the perceived quality of the segmentation outcome, as additional loss functions for the MultiSDGAN model. As evident from the results, this joint training for the dual task of segmentation and superresolution employing the MultiSD module was achieved effectively. These additions play a major role in significantly improving the achieved results. Furthermore, the Dice and SSIM losses as an additional constraints to the original $L1$ loss emphasized the reconstruction performance, and empirically from the results, it can also be seen that they contributed to the considerable improvement in the model performance.

We empirically showed that we can transfer the low-resolution OCT scans to high-resolution segmentation labels with no degradation in the accuracy and the perceived quality. Therefore, we can take faster scans from patients with neurodegenerative disease, especially AD, to minimize eye movements during the scans, which is the most significant source of noise. Our future direction is to collect more OCT scans, particularly from a big population of AD patients with high statistical power progressively, and further analyze the isolated retinal layers to discover patterns and biomarkers in their OCT images toward OCT-based progressive AD diagnosis.

REFERENCES

[1] J. M. Schmitt, "Optical coherence tomography (OCT): A review," *IEEE J. Sel. Topics Quantum Electron.*, vol. 5, no. 4, pp. 1205–1215, Jul./Aug. 1999.

[2] A. F. Fercher, C. K. Hitzenberger, W. Drexler, G. Kamp, and H. Sattmann, "In vivo optical coherence tomography," *Amer. J. Ophthalmol.*, vol. 116, no. 1, pp. 113–114, 1993.

[3] A. F. Fercher, W. Drexler, C. K. Hitzenberger, and T. Lasser, "Optical coherence tomography-principles and applications," *Rep. Prog. Phys.*, vol. 66, no. 2, pp. 239–303, 2003.

[4] L. K. Ferreira and G. F. Busatto, "Neuroimaging in Alzheimer's disease: Current role in clinical practice and potential future applications," *Clinics*, vol. 66, pp. 19–24, 2011.

[5] L. Gao, Y. Liu, X. Li, Q. Bai, and P. Liu, "Abnormal retinal nerve fiber layer thickness and macula lutea in patients with mild cognitive impairment and Alzheimer's disease," *Arch. Gerontol. Geriatrics*, vol. 60, no. 1, pp. 162–167, 2015.

[6] L. P. Cunha *et al.*, "Macular thickness measurements with frequency domain-OCT for quantification of retinal neural loss and its correlation with cognitive impairment in Alzheimer's disease," *PLoS One*, vol. 11, no. 4, 2016, Art. no. e0153830.

[7] M. L. Monteiro, D. B. Fernandes, S. L. Apóstolos-Pereira, and D. Callegaro, "Quantification of retinal neural loss in patients with neuromyelitis optica and multiple sclerosis with or without optic neuritis using fourier-domain optical coherence tomography," *Invest. Ophthalmol. Vis. Sci.*, vol. 53, no. 7, pp. 3959–3966, 2012.

[8] H. A. Bayhan, S. Aslan Bayhan, A. Celikbilek, N. Tanık, and C. Gürdal, "Evaluation of the chorioretinal thickness changes in Alzheimer's disease using spectral-domain optical coherence tomography," *Clin. Exp. Ophthalmol.*, vol. 43, no. 2, pp. 145–151, 2015.

[9] A. G. Roy *et al.*, "RelayNet: Retinal layer and fluid segmentation of macular optical coherence tomography using fully convolutional networks," *Biomed. Opt. Exp.*, vol. 8, no. 8, pp. 3627–3642, 2017.

[10] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[12] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis.*, 2016, pp. 565–571.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2015, pp. 3431–3440.

[15] O. Ronneberger, P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Convolution. Netw. Biomed. Image Segment.*, 2015, vol. 9351, pp. 234–241.

[16] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++ : A nested U-Net architecture for medical image segmentation," in *Deep Learn. Med. Image Anal. Multimodal Learn. Clin. Decis. Support*, 2018, pp. 3–11.

[17] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2016, pp. 424–432.

[18] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional DenseNets for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 11–19.

[19] Y. Liu *et al.*, "Layer boundary evolution method for macular OCT layer segmentation," *Biomed. Opt. Exp.*, vol. 10, no. 3, pp. 1064–1080, 2019.

[20] X. Liu, D. Liu, T. Fu, Z. Pan, W. Hu, and K. Zhang, "Shortest path with backtracking based automatic layer segmentation in pathological retinal optical coherence tomography images," *Multimedia Tools Appl.*, vol. 78, no. 12, pp. 15817–15838, 2019.

[21] S. Masood *et al.* "Automatic choroid layer segmentation from optical coherence tomography images using deep learning," *Sci. Rep.*, vol. 9, no. 1, pp. 1–18, 2019.

[22] Y. He *et al.*, "Fully convolutional boundary regression for retina OCT segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Intervention*, 2019, pp. 120–128.

[23] J. I. Orlando *et al.*, "U2-Net: A Bayesian U-Net model with epistemic uncertainty feedback for photoreceptor layer segmentation in pathological OCT scans," in *Proc. IEEE 16th Int. Symp. Biomed. Imag.*, 2019, pp. 1441–1445.

[24] A. A. Jammal *et al.*, "Detecting retinal nerve fibre layer segmentation errors on spectral domain-optical coherence tomography with a deep learning algorithm," *Sci. Rep.*, vol. 9, no. 1, pp. 1–9, 2019.

[25] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[26] C. Ledig *et al.* "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4681–4690.

[27] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[28] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.

[29] G. Mordido, H. Yang, and C. Meinel, "Dropout-GAN: Learning from a dynamic ensemble of discriminators," 2018, *arXiv:1807.11346*.

[30] B. Neyshabur, S. Bhojanapalli, and A. Chakrabarti, "Stabilizing GAN training with multiple random projections," 2017, *arXiv:1705.07831*.

[31] I. Durugkar, I. Gemp, and S. Mahadevan, "Generative multi-adversarial networks," 2016, *arXiv:1611.01673*.

[32] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.*, 2010, pp. 2366–2369.

[33] L. Levi, "Unsharp masking and related image enhancement techniques," *Comput. Graph. Image Process.*, vol. 3, no. 2, pp. 163–177, 1974.

[34] H. Zhang *et al.*, "StackGAN++ : Realistic image synthesis with stacked generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, Aug. 2019.

[35] A. F. Agarap, "Deep learning using rectified linear units (RELU)," 2018, *arXiv:1803.08375*.

[36] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," vol. abs/1701.04862, 2017.

[37] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 47–57, Mar. 2017.

[38] M. Claesen and B. De Moor, "Hyperparameter search in machine learning," 2015, *arXiv:1502.02127*.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[40] Z. Wang, "The SSIM index for image quality assessment," 2003. [Online]. Available: https://ece. uwaterloo. ca/˜z70wang/research/ssim

[41] T. K. Kim, "T test as a parametric statistic," *Korean J. Anesthesiol.*, vol. 68, no. 6, pp. 540–546, 2015.

[42] W. Kim, A. Kanezaki, and M. Tanaka, "Unsupervised learning of image segmentation based on differentiable feature clustering," *IEEE Trans. Image Process.*, vol. 29, pp. 8055–8068, 2020, doi: 10.1109/TIP.2020.3011269.