Lappeenranta University of Technology
School of Engineering Science
Degree Program in Computational Engineering and Technical Physics
Intelligent Computing Major

Master Thesis

Sinan Kaplan

DEEP GENERATIVE MODELS FOR SYNTHETIC RETINAL IMAGE GENERATION

Examiners: Professor Lasse Lensu
D. Sc. Lauri Laaksonen

Supervisor: Professor Lasse Lensu

ABSTRACT

Lappeenranta University of Technology
School of Engineering Science
Degree Program in Computational Engineering and Technical Physics
Intelligent Computing Major

Sinan Kaplan

Deep Generative Models for Synthetic Retinal Image Generation

Master Thesis

2017

76 pages, 33 figures, 2 tables and 9 appendices.

Examiners: Professor Lasse Lensu
D. Sc. Lauri Laaksonen

Keywords: synthetic retinal image, deep generative models, generative adversarial networks, variational autoencoders, deep learning, retinal imaging, computer vision

The retina is an important part of the eye, which can be used to detect eye-related diseases in advance by applying retinal imaging techniques. However, the main problem of ongoing research in this field is the shortage of synthetic retinal data to be used for further development and validation of retinal data analysis methods. To solve this problem, this thesis studies state-of-the-art deep generative models to generate synthetic retinal data from a noise without conditioning any information regarding to the retina. Synthetic retinal images are generated by Generative Adversarial Networks and Variational Autoencoders. To quantify the quality of generated retinal data, a similarity based quality assessment method is proposed. The utilization of deep generative models reveals that the global structure of the retina can be generated successfully excluding the vessel tree structure.

PREFACE

The idea of this research originated from my passion for studying adversarial learning and the proposal from my supervisor to combine it with retinal imaging. However, I would not be able to reach a certain level of success without the support of my supervisor. Therefore, I would like to thank Lasse Lensu for providing me an excellent guidance during this process. He helped me a lot to finish my thesis before the winter finally came to the Seven Kingdoms.

Also, I would like to thank my friends and those who have contributed practically and academically while conducting the research.

At the end, thanks a lot to you reader for reading even a page. I hope you enjoy your reading.

To my grandmother Dilber Kaplan.

August 3, 2017

Sinan Kaplan

CONTENTS

ABBREVIATIONS AND SYMBOLS

ANN	Artificial Neural Networks.
BatchNorm	Batch Normalization.
cGAN	Conditional Generative Adversarial Networks.
CNN	Convolutional Neural Networks.
CTIS	Computing Tomographic Imaging Spectrum.
D	Discriminator.
DR	Diabetic Retinopathy.
EEG	Electroencephalography.
EHR	Electronic Health Record.
FCM	Fuzzy C-Means.
G	Generator.
GAN	Generative Adversarial Networks.
HRF	High Resolution Fundus.
K-NN	K-Nearest Neighbor.
KL	Kullback-Leibler.
MLE	Maximum Likelihood Estimation.
MRI	Medical Resonance Imaging.
OCT	Optical Coherence Tomography.
PCA	Principal Component Analysis.
RBF	Radial Basis Functions.
ReLU	Rectified Linear Unit.
RGB	Red-Green-Blue.
SGD	Stochastic Gradient Descent.
SVM	Support Vector Machines.
TFD	Toronto Face Database.
VAE	Variational Autoencoders.

β	Beta.
$d(.,.)$	Distance.
$D(.)$	Probability of real data.
E	Expectation.
ϵ	Sample noise.
$f(.)$	Encoder function.
$G(.)$	Probability of generated data.
$g(.)$	Decoder function.

$H(.)$	Entropy.
$h(.)$	Histogram.
I	Identity matrix.
$I(.,.)$	Image.
L	Variational loss.
$\log$	Logarithmic.
μ	Mean.
$\mathcal{N}(.,.)$	Gaussian normal distribution.
$\odot$	Element-wise multiplication.
$p(.,.)$	Joint probability.
$p(.)$	Probability density function.
$p(.\|.)$	Conditional probability.
$q(.)$	Probability density function.
σ	Standard deviation.
σ^2	Variance.
$\sum_{i=n}^{N}$	Sum over i from n to N.
Σ	Covariance.
θ	Parameters.
$\hat{\theta}$	Estimated parameters.
$\bigtriangledown$	Gradient.
x	Data sample.
X	Data set.
$\tilde{x}$	Decoded/Reconstructed data.
z	Latent variable.
Z	Latent space.

1 INTRODUCTION

1.1 Background

The retina is a widely studied part of the eye fundus. It is an important tissue, which is responsible for transforming incoming light into a neural signal. This signal is processed further in the visual cortex of the brain. Thus, it can be considered as a continuation of the brain. The medical examination data acquired from the retina can be used in many ways to detect and track anomalies in the human body [1]. This data can be obtained in several ways, such as Magnetic Resonance Imaging (MRI) and Electroencephalography (EEG) [1]. As the retina is a part of the brain, imaging of the retina makes the brain accessible noninvasively, which enables the examination of the central nervous system for detecting abnormalities. In addition to that, it might also help to design, biomedical identification systems based on its underlying blood vessel structure [2].

Owing to its location and functions, diseases, both eye related and body circulation might be apparent in the retina [3]. The diseases that manifest in the retina include Diabetic Retinopathy [4], Macular Degeneration (particularly age-related) [5], Glaucoma [6], Cardiovascular [7], Tumors [8] and Tuberculosis [9]. For more relevant diseases that can be detected with retinal imaging refer to the study in [10].

The potential of retinal imaging for the detection and diagnosis of the aforementioned diseases as early as possible enables researchers to develop and improve the techniques for analyzing the retinal images. The relevant information can be gathered from both Red-Green-Blue (RGB) and multi-spectral images of the retina. However, the level of information collected in the RGB retinal images is limited and, thus, multi-spectral imaging of the retina is commonly studied for a better understanding of the retina [3, 11].

While the development in the field of retinal image analysis improves with the help of advances in technology, the demand for the retinal data increases. To study and detect possible abnormalities (in particular, the eye-related diseases), the availability of the retinal data is crucial in the field. Although there are publicly available retinal data sets provided by research institutes and hospitals, there is still a considerable need for synthetic retinal data for further development and validation of retinal data analysis methods in the field. For this purpose, one can think of generating synthetic retinal data from the currently available retinal data.

An important approach in terms of generating retinal images is to apply generative models. The generative models enable us to learn the underlying hidden structure of the data that can be further processed to generate new data samples as similar as possible to real ones. Thus, in this thesis state-of-the-art deep generative models are studied and employed to generate synthetic retinal data.

1.2 Objectives and Restrictions

Given the relevant background, the objectives of this thesis are as follows:

(i) To review the related literature of the retinal imaging techniques and the solutions proposed for reconstructing retinal images both from the RGB and spectral images. To apply deep generative models including generative adversarial networks and variational autoencoders for generating synthetic retinal images.

(ii) To propose a similarity-based retinal image quality assessment method to evaluate the generated retinal images quantitatively.

(iii) To reveal the issues and solutions during the training of the deep generative models for avoiding possible model collapses in the case of retinal image generation.

In the scope of the thesis, by considering the time and resources allocated, the restrictions are specified as follows:

(i) Because of the variety of the retinal image resolutions in the data set, the size of the images are downscaled.

(ii) Due to the limited computing power, only a few convolutional neural network design approaches are investigated to generate retinal images.

1.3 Structure of the Report

This thesis is organized as follows: In Section 2, the eye and retinal imaging are studied. The section gives an overview of the related eye structure and the major characteristics of the retina and provides a detailed review of available retinal imaging techniques. Section 2

also introduces studies regarding the reconstruction of retinal images in the literature. The deep generative models for synthetic retinal image generation and the proposed quality assessment method in the context of this thesis are explained in Section 3. Section 4 presents the experimental analysis in details and the evaluation of the generated retinal images. Section 5 discusses the finding from the retinal image generation process with respect to the experimental analysis. Finally, Section 6 concludes the study.

2 THE EYE AND RETINAL IMAGING

This section introduces the anatomy of the eye, currently available retinal imaging techniques and the methods developed for the reconstruction of the retina to provide more data for further method development and validation in the field.

2.1 The Anatomy of the Eye

It is a fact that 80 % of information received by the brain comes from the eye [12] and the retina is an important part of the eye by which it is possible to access the brain tissue in a noninvasive way. As the retina is an active metabolic tissue with blood supply, the body circulations can be observed directly as well. The location and the functionalities of the retina make it attractive for the researchers to study it for detecting and screening of both eye-related problems and the diseases that manifest in the retina such as, widely known diabetic retinopathy. To be able to understand how one can get such information related to diseases, it is crucial to get familiar with the anatomy of the eye [12]. Hence, this subsection covers the details of the eye structure in general.

The eye has a spherical shape and it consists of three major layers: (1) the outer layer constructed by the sclera and the cornea, (2) the center layer composed of the iris, ciliary body, and choroid, (3) the inner layer consisting of the retina. The way how the eye works is that a light ray passes through the cornea and then traveling by way of pupil and lens it reaches the retina. The retina receives a light ray and transforms the light into a neural signal that will be conveyed by the optic nerve to the brain. Figure 1 illustrates both the structure of the eye and the layers of the retina. The significant parts of the eye are described as follows:

- Cornea is the outermost part of the eye by covering the pupil, the iris, and the anterior chamber. The cornea contains oxygen and nutrients and it focuses on an incoming light with the help of its 80% water content. Also, it does not include any blood vessels. The main functionality of the cornea is to refract and transmit the light.
- The aqueous humor is located between the lens and the cornea. It is responsible for providing oxygen and nutrients to the cornea and the lens.
- Iris is a pigmented thin part of the eye that adjusts the amount of incoming light by

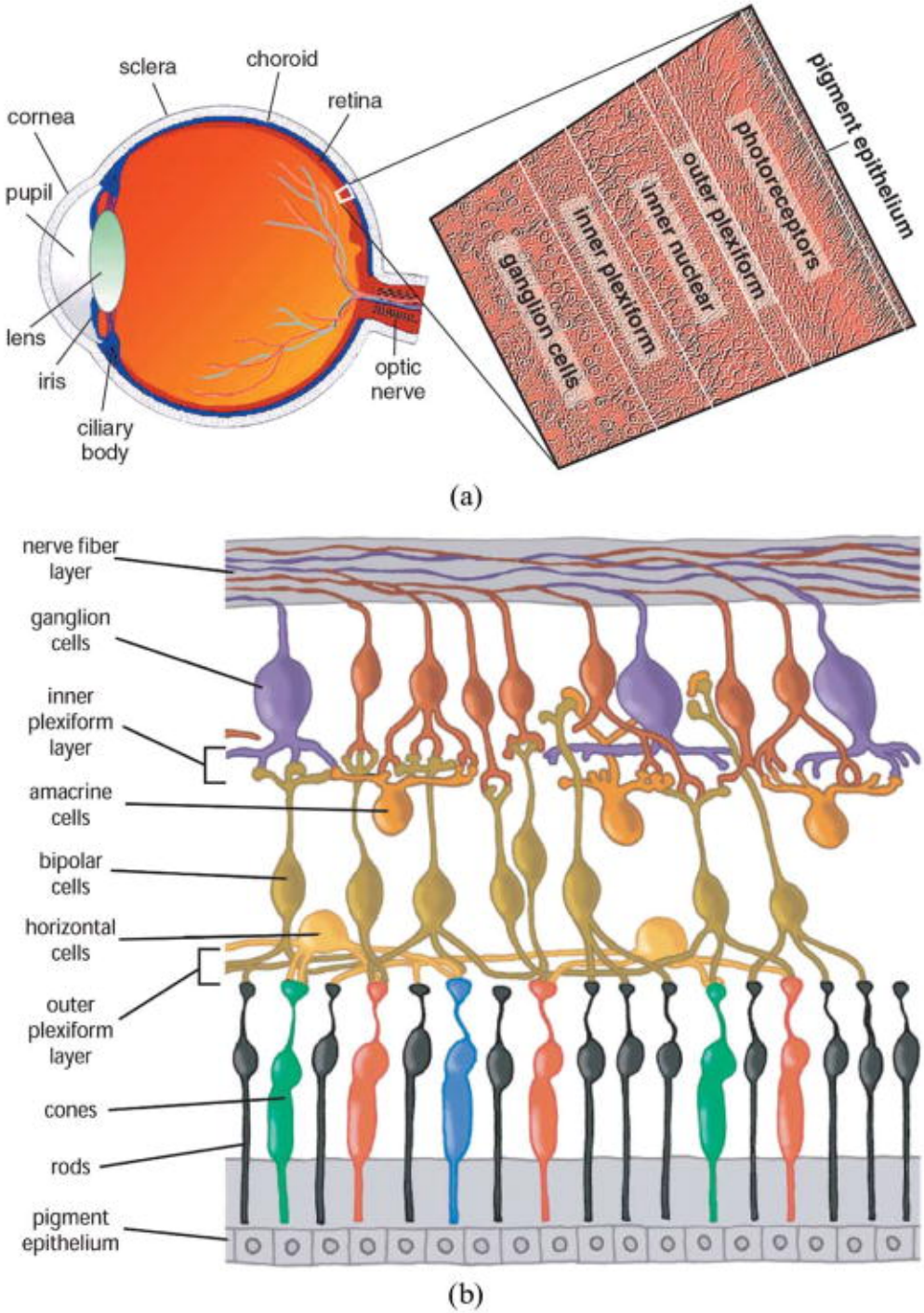


Figure 1. Scheme of the eye structure and the retinal layers: (a) the eye and its main parts [13]; (b) the retina and illustration of its cellular layers [14].

controlling the pupil size. If the size of the pupil is expanded then more light enters the eye. In particular, this helps to focus on distant objects and to see in the dark.

- Pupil is located in the center of the eye and it regulates the amount of light entering the eye.
- Lens is responsible for the refraction of light to be focused on the retina. The shape of the lens determines the focal length of the eye, which enables to have sharp images.
- The vitreous humor mainly covers the space between the lens and the retina with 95% water content. It is the largest part of the eye.
- The sclera is considered as a protective outer shield of the eye. The connected tiny muscles to the sclera control the continuous eye movements. At the very back of the eye, the optic disk and the sclera are attached together.
- The optic disc is the part of the eye where the optic nerve is connected to the eye.
- Retina is located in the inner part of the eye as a layer of neural cells. It is sensitive to light and it receives and transfers the neural signals to the brain. The rods and cones are two light receptor types located in the retina. The rods basically absorb the light intensity, thus, they are responsible for night vision. The cones are color sensitive receptors and absorb strong light, thus, they are responsible for color vision. The retina does contain blood vessels.
- The macula is a highly sensitive part of the retina and it covers the fovea. It is responsible for detailed central vision.
- The fovea is located at the most central part of the macula without any blood vessels. The visual cells in the fovea enable sharp vision.

For more detailed information regarding the eye and its structure refer to the study in [12].

2.2 The Retinal Imaging

Detection of eye-related diseases in advance, particularly diabetic retinopathy (DR), is very crucial in terms of avoiding serious health issues, such as blindness. The different stages of DR can be detected by the ophthalmologists using biomicroscopy. An alternative

way to this approach is to take advantage of retinal imaging. The important amount of information for both the eye-related diseases and the systemic diseases (e.g., diabetes) to be used in diagnostics can be extracted from retinal imaging. By considering the importance of retinal imaging, it is useful to understand the retinal imaging techniques and how it has evolved through the history. Therefore, this section introduces the applied retinal imaging in the field.

Starting from back in the 1800s, there has been an interest to obtain an image of the retina to understand the anatomy of the eye (particularly the image formation in the eye) [15]. Such interest in the image of the retina has led to the invention of the ophthalmoscope early in 1800s [16]. Since the ophthalmologists have begun to apply the knowledge of the retina for the diagnose purpose by using the ophthalmoscope, the retina has discovered with more details and the first image of the retina was released in 1853 [17] as shown in Figure 2.

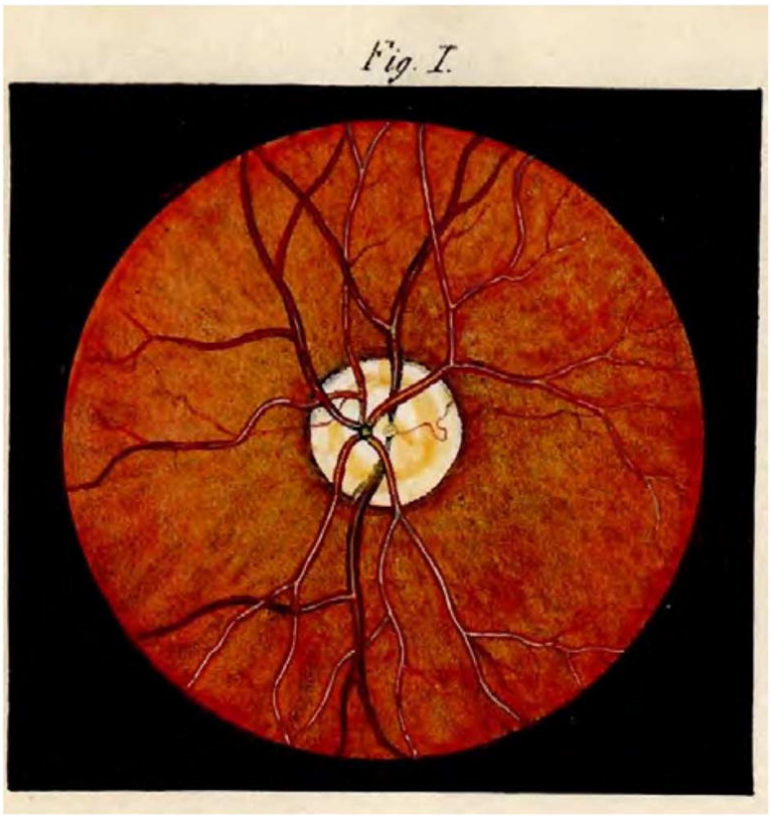


Figure 2. First image of the retina which was drawn back in 1853 [17].

As the ophthalmoscope is an invasive technique for patients, researchers have been interested in to find a non-invasive solution for obtaining the image of the retina. As a result,

the first image of the retina was obtained photographically in 1891 and following that the first fundus camera was developed in 1910 [15]. Since then, the developed fundus camera concept has been applied in retinal imaging field and fundus imaging has been accepted as the main method to acquire an image of the retina. The fundus imaging is a process of obtaining the 2-D projection of 3-D retinal tissue onto an image plane from the reflected light [15].

Afterward, an important development in the field was to use fluorescein angiography for acquiring the image of the retina in which a fluorescent color (or dye) injection is carried through the bloodstream in order to photograph the retina. This technique particularly captures the vessel tree structure of the retina clearly.

By considering the main limitation of the fundus imaging, in which the 2-D projection of the retina obtained, there has been an interest to acquire the 3-D view of the retina. Thus, the first application of the 3-D retinal imaging was developed using stereo [1] fundus photography back in 1964 [18]. In addition to the stereo imaging, with the development of the scanning laser ophthalmoscopy, the 3-D image of the retina can be obtained by tomographic imaging techniques called optical coherence tomography (OCT) [19]. OCT is a retina imaging technique in which imaging is done based on the reflected light. In addition to the dimensions of the image (transverse dimensions), OCT also captures the depth of the retinal image.

In the recent years, there has been rapid development in the field of retinal imaging to be applied in medical care [3, 11]. For instance, the diseases such Diabetic Retinopathy [4], Macular Degeneration (particularly age-related), Glaucoma mostly are detected and diagnosed by fundus imaging. Also, by enabling a specialist to accurately delineate pathological changes, fluorescein angiography is widely applied for detecting and tracking of retinal disorders. However, the limitations of fluorescein angiography, such as failure to image circulation of choroidal correctly, requiring accurate and expensive photography equipment, the need of expert photographers, the lag in treatment and diagnosis due to the process of fluorescein angiography and filming, have made OCT to be preferred over fluorescein angiography [15, 20].

As OCT gives the cross-sectional image (also called tomography) of the retinal tissue, an important advantage of OCT over fluorescein angiography is that it provides better visualization of the region of interest. For instance, Figure 3 demonstrates fluorescein angiography and OCT together on the same region of the eye (Optic Nerve) [21]. From

[1] *In stereo imaging, an image of an object is captured from different angles and the main objective is to find corresponding points in the images to match them.*

the figure, it can be seen that the vessels and their structure are more visible in OCT. Furthermore, since it does not cause any disturbance on the tissue, it is a noninvasive technique [19].

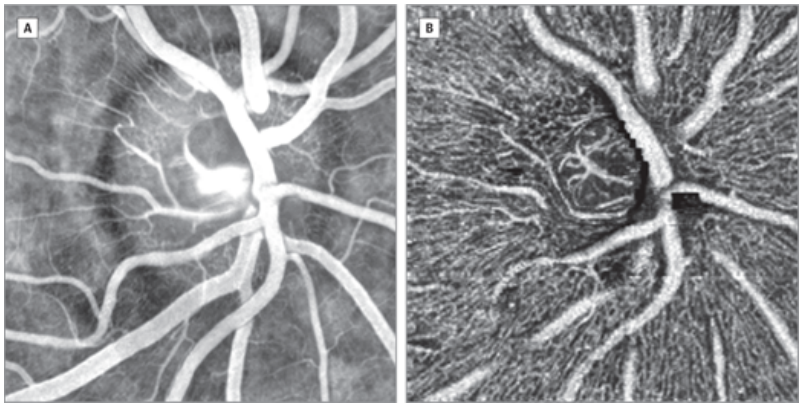


Figure 3. Same region of the eye (Optic Nerve) imaged by: (A) Fluorescein angiography; (b) OCT [21].

In addition to the 3-D image of the retina, the spectral imaging of the retina is also studied for the purpose of having a multi-view of the retinal tissue. For instance, Fält *et al.* [22] have managed to create a spectral retinal data set by using digital fundus imaging. In this study, the created spectral images are within the range of 400 to 700 nm with 10 nm interval. In total, 66 humans were imaged, while 54 of these people were with diabetic retinopathy, 12 of them were imaged with non-diabetic retinopathy. This data set is used in diabetic retinopathy analysis tasks. However, the main drawback of this study is that at the end it suffers from color distortion, particularly at the edges.

The main problem with the above-mentioned retinal imaging techniques is the image acquisition time. To overcome this issue, the snapshot retinal imaging technique was proposed [23, 24, 25]. By taking advantage of snapshot imaging, there are several studies conducted to reveal the potential of this approach in retinal imaging. The first study in this area has used Computed Tomographic Imaging Spectrometer (CTIS) to obtain retinal images by snapshot technique [23]. As a result, a full spatial-spectral image cube from 450 nm to 700 nm in 50 channels is acquired in 3ms from 2 individuals. Except for the improvement of image acquisition time, the main advantage of this method is to obtain spatial information of the retinal image as well. Gao *et al.* studied snapshot imaging for the same purpose by using Image Mapping Spectrometer [24]. They proposed a retinal camera which has acquired a retinal data cube just by a single snapshot. The developed method is able to obtain 48 bands retinal images within the range of 470 nm - 650 nm by taking 5.2 fps. However, this method still needs to be validated through several healthy

individuals and individuals with retinal abnormalities.

To recap, each of the above-mentioned techniques for the retinal imaging has its own advantages and disadvantages. It should be chosen based on the specific need. For instance, if one needs to capture the clear structure of the vessel tree, it can be a good approach to follow eye fundus imaging. However, if the cross-sectional visualization of the retina is more important than the vascular tree, it can be more suitable to choose OCT. Moreover, as the spectral retinal imaging provides multi-view of the retina and the level of the gathered information from the retina is high, one can think of applying spectral retinal imaging to have a better understanding of the retina. Although both the fluorescein angiography and OCT-based retinal imaging can obtain the retina in a robust and accurate way, they mainly suffer from the image acquisition time and constant eye movement. To overcome these issues, one might consider the snapshot retinal imaging as an alternative approach to acquire the retinal images.

2.3 Previous work on Synthetic Retinal Image Generation

Although retinal imaging technology emerges by the developments in the imaging technology, there is still a huge need for retinal images for validation and further development of methods in the field of retinal image processing and analysis. To overcome this shortage of data, researchers have started to conduct studies on reconstruction and synthesis of retinal images. In this section, the detailed review of related recent studies in this area is given.

An earlier study in this field has reconstructed 3-D model of the eye for the purpose of assisting the surgeons in surgery operations [26]. In this way, the surgeons had the 3-D understanding of the eye. As part of the work, the main parts of the eye, including the sclera, cornea, iris, retina, blood vessels, and the eyelashes are modeled successfully.

While reconstructing the retinal images one should bear in mind that the major features of the retina must be preserved for better analysis. These major parts of the retina are optic disc, vessel network, and fovea. To create more realistic fundus images for validation of retinal image analysis algorithms (particularly for segmentation tasks) by preserving those major characteristics of the retina, Fiorini *et al.* [27] reconstructed the retinal images in three steps, which are as follows: (1) Generation of the fovea with a background based on patch based algorithm, (2) Modeling the optic disc and optic disc generation, (3) Generation of the vessel tree by model-based approach. The model is tested on High-

Resolution Fundus (HRF) Image Database [28] and it provides the major features of the retina quite well as shown in Figure 4. One can see from the generated fundus images that the intensity of the vessel network is uniform across the whole retina. This is why there are still adjustments needed for the intensity of the vessel network. Another important issue, which was observed from the obtained results, is that the reconstructed vessel tree does not preserve the vascular structure quite well. Rather, it provides a simple vessel tree, which is supposed to be as complex as in the ground truth.

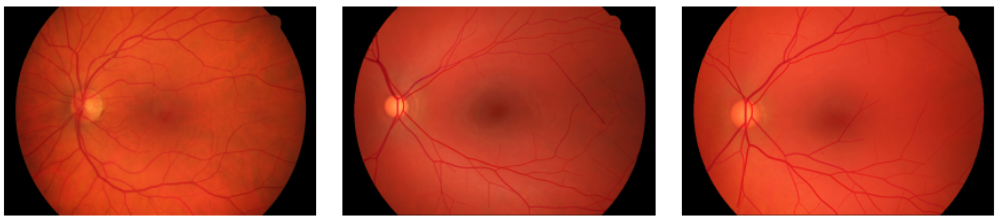


Figure 4. Comparison of generated fundus images: (a) the ground truth from HRF [28]; (b) a reconstructed retina [27]; (c) a reconstructed retina [27].

The study in [27] suffers from unrealistic vascular network structure. Therefore, this study is extended and in the new version, Bonaldi *et al.* [29] succeeded to reconstruct more realistic retinal images while preserving the vascular structure. The developed model is based on Active Shape Model [30]. In the model, PCA is used for dimensionality reduction, Kalman filter [31] is utilized for revealing vessel textures and the Gaussian filter is applied for edge smoothing. By applying this model to HRF data set, RGB channel images are synthesized. The authors evaluated their results by asking medical experts whether reconstructed images seem real or not. The average point from tests is 2.1 out of 4. In addition to that, the reconstructed retinal images are used for segmentation and it provides reliable results for retinal image analysis. However, still few characteristics of the retinal image are reconstructed. Figure 5 shows a reconstructed image from retinal image. One can see that vascular network structure is more realistic than shown in Figure 4.

The reconstruction of the vessel has been a hard task for retinal image reconstruction because of its characteristics like bifurcation points, vessel width, and length, major temporal arcade, and tortuosity. To be able to get a clear structure of the vessel tree in the retina, Guimar *et al.* [32] segmented vascular network of the retina in three-dimensional space for better visualization and better reconstruction of the retina. The proposed algorithm consists of the steps illustrated in Figure 6.

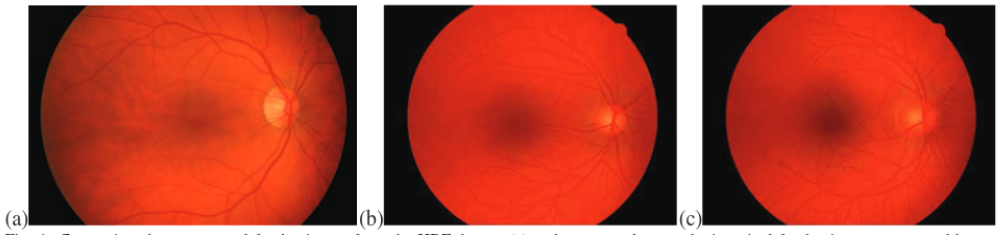


Figure 5. Comparison of generated fundus images: (a) the ground truth from HRF [28]; (b)a reconstructed retina [29]; (c) a reconstructed retina [29].

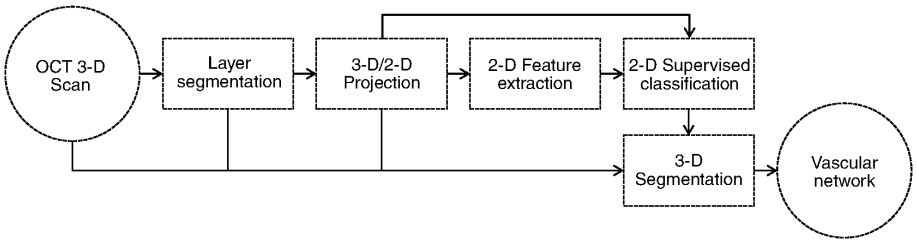
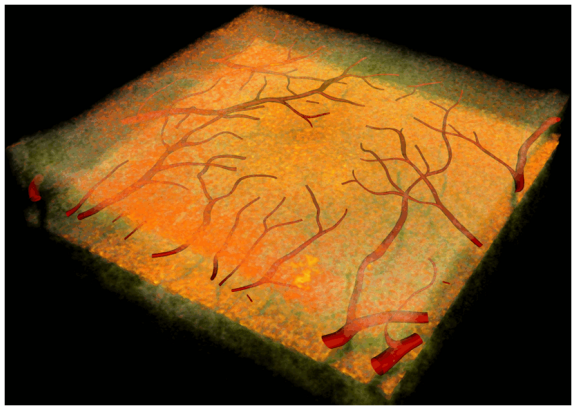


Figure 6. Flowchart of the algorithm for 3-D vessel segmentation and reconstruction [32].

The results obtained from the reconstruction of the Cirrus HD-OCT data set [33] and 17 individuals (2 of them with diabetic retinopathy) show that the vessel structure was segmented accurately when there is few vessel connection at bewildering points of the retina. An example of the 3-D reconstructed vessel structure from OCT data cube can be seen in Figure 7.



.

Figure 7. 3-D reconstructed vascular structure from OCT data [32].

As the vessel trees provide an information about the eye-related diseases beforehand, Fang *et al.* [34] has studied the vessel tree reconstruction in particular. The straightforward way to detect vessel trees is to apply edge detection algorithms, such as Sobel and Laplacian of Gaussian. However, these methods cannot be applied to the vessel tree detection be-

cause of the poor local contrast and these methods usually detect the parallel lines. By considering these facts, the method proposed in the study is based on the dynamic region growing in morphological operations. As the morphological operations are sensitive to the size of the structural element, in a dynamic region growing the window size of the structural element is adjusted based on the local information of the vessel.

Although the studies in the field usually have focused on particularly the reconstruction of the vascular structure of the retina, the study in [35] has focused on the 3-D reconstruction of the optic disc to access more information about possible damages in the optic disc. In order to achieve this goal, the stereo retinal images are used to generate the 3-D view of the optic disc. From stereo images, the estimation of the 3-D shape of an image can be estimated by applying the relative position difference or disparity of one or more correspondence. To get the 3-D view of the optic disc from the stereo retinal images, the disparity map between corresponding retinal image points is constructed and then it is further used to recover the 3-D shape of the optic disc.

The recent work was done by Nguyen *et al.* [36] applies Radial Basis Functions (RBF) to reconstruct the spectral retinal images from the corresponding RGB retinal images. The study is constituted following three steps: (1) the retinal data was quantized by Fuzzy C-Means (FCM) algorithm to cluster both RGB and spectral retinal images; (2) RGB retinal images were mapped to spectral retinal images using RBF; and (3) the reconstruction was performed by using FM based algorithm and the segmentation based algorithm that applies supervised learning.

The performance of the aforementioned study is measured by computing similarity and dissimilarity between the actual retinal image and the recovered/reconstructed retinal image pairs, thus, Spectral Angle Mapper (SAM) and Spectral Information Divergence (SID) are used as dissimilarity metrics while Spectral correlation mapper (SCM) is used as a similarity metric. The experimental results show that these two methods, FCM and segmentation based, are good enough to apply for reconstruction.

By and large, the reconstruction and the generation of the retinal images from the available retinal data is a significant topic in terms of providing more data to the research community in this field for validation and further development of the retinal image analysis algorithms. In particular, the studies commonly focused on reconstructing the spectral retinal images from chromatic colors such as from RGB pairs. This is because of the amount of information provided by spectral retinal data. In addition to the reconstruction of spectral retinal data, chromatic retinal images are also generated from the available

chromatic retinal data. Although the optic disc and fovea within the retina are synthesized quite well, the major issue encountered during these studies is the reconstruction of the vessel tree structure. Most studies suffer from unrealistic reconstructed vessel tree structure. However, there is still demand to the retinal data, especially the distinct retinal images. Therefore, one can focus on generating the diverse retinal data by using the accessible data sets provided by research communities and the hospitals.

3 GENERATION OF SYNTHETIC RETINAL IMAGE

This section covers the detailed explanation of deep generative models for synthetic retinal image generation. First, the section introduces the general framework used to generate retinal images in the context of this thesis. Secondly, it reviews the generative models by giving the difference between the generative and discriminative models and the parameter estimation of the generative models in machine learning (ML) area. For completeness of the mathematical foundations of generative models, relevant examples are presented. Furthermore, the deep generative models used to synthesize the retinal images, which are *Generative Adversarial Networks (GANs)* and *Variational Autoencoders (VAEs)*, are presented by reviewing the relevant literature. Finally, the proposed quality assessment method is given in details.

3.1 Proposed Framework for Generating Synthetic Retinal Images

The proposed solution for generating synthetic retinal images consists of three main steps as illustrated in Figure 8. The first step is a preprocessing step, and it deals with rescaling and cropping tasks. By varying resolutions of the retinal images (taken by different cameras), used in the experimental analysis, have made the preprocessing an essential step. An important point to consider here is the high computational cost of the training procedure that occurs due to the relatively large-scale retinal images (which basically slows down the training procedure). Therefore, the retinal images are downscaled. In addition to that, the cropping is done to get rid of the black region around the retinal images. The second step is a utilization of deep generative models including GANs and VAEs to generate synthetic retinal images. The details of the both methods are given in the next sections. Finally, the retinal image quality assessment is carried for the generated retinal images at the third step by the proposed similarity based retinal image quality assessment method.

3.2 An Introduction to Generative Models

To be able to understand the core of the deep generative models, it is clearly essential to understand main differences between the generative and discriminative models and the parameter estimation of the generative models. Hence, this chapter presents the intuition behind the both generative and discriminative models and how the estimation of generative model parameters is done.

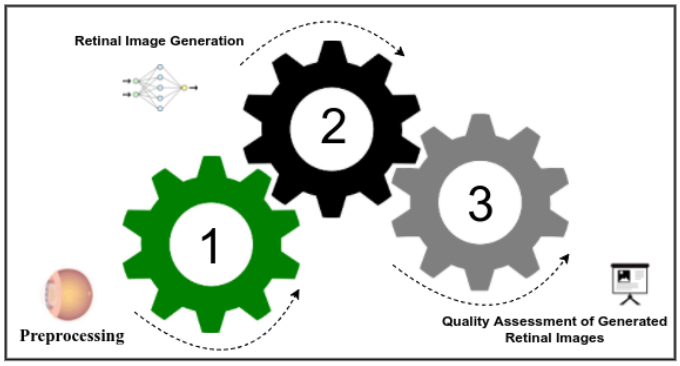


Figure 8. Proposed framework for the generation of the retinal images by deep generative models comprises three major steps: (1) preprocessing; (2) the retinal image generation; and (3) the quality assessment of the generated retinal images.

Machine learning is a way to teach a computer to understand the data and make inferences based on the data. This is achieved by two different learning approaches, which are *supervised* and *unsupervised learning*. In supervised learning, each data instance has its own label/target. The goal is to understand the data based on the given targets and to assign a target value for unseen data inputs. If the target values are defined by categories (or labels) the problem is called *classification* problem. For instance, let us assume we have a spam filtering problem as illustrated in Figure 9 and the objective is to design a machine learning solution to classify the incoming emails into spam and not spam classes (which are the targets). This problem is a classification problem and it can be solved by applying classification algorithms such as logistic regression. Alternatively, if the targets are continuous values, then the problem is a *regression* problem. An obvious example of this type of supervised problem is a house price estimation demonstrated in Figure 9 and the main goal is to compute the price of a house based on the given features like size, the number of rooms and so on. As the price is a continuous variable, this problem can be solved by using regression algorithms such as linear regression.

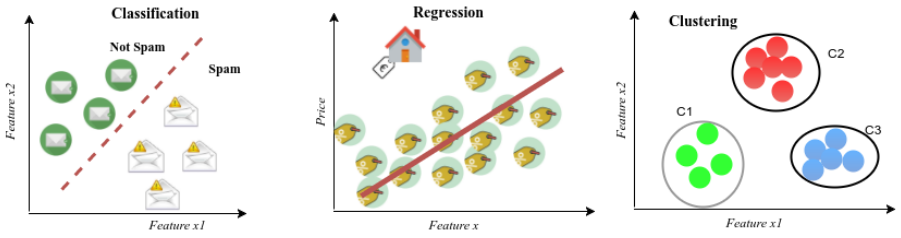


Figure 9. Examples of real-life problems in the context of supervised and unsupervised learning tasks: Spam filtering as a *classification* task and House price estimation as a *regression* task are part of supervised learning; *Clustering* is part of unsupervised learning in which customers are grouped into three different categories based on their purchasing behavior.

Yet, in unsupervised learning, the provided data have no targets. This means there are no certain categories to be assigned to the given input. The objective here is to cluster the data into different groups. For instance, one can think of the clustering of customers based on their purchasing history or group the people based on their movie choices in recommendation systems as an unsupervised learning task shown in Figure 9.

In supervised learning tasks, both generative and discriminative models are widely applied[37, 38]. Each of these models approaches a given problem from a different aspect. Generative models basically try to understand how the data actually are generated to perform classification and it can be used to generate synthetic data samples from the underlying distribution of the given data. It models both the class conditional and prior probability in the context of probabilistic models. On the other hand, discriminative models are not concerned about how the data are generated and basically performs discrimination within the data by modeling the posterior probability.

More specifically, let us assume we have a data x and we want to categorize the data into associated class labels y as a classification task. To achieve this goal, a generative model learns the joint probability $p(x, y)$ whereas a discriminative model learns the posterior probability of class $p(y|x)$ (the probability of class y given x), which is basically a decision boundary between classes. The joint probability is formed as follows:

$$p(x, y) = p(x|y)p(y) \tag{1}$$

or

$$p(x, y) = p(y|x)p(x) \tag{2}$$

where $p(x|y)$ is the class conditional probability, $p(y|x)$ is the posterior probability of class (likelihood), $p(y)$ is the marginal distribution of the class (or the probability of the class) and $p(x)$ is the probability of the data.

In this context, the generative methods learn $p(y)$ and $p(x|y)$ to classify the data. Afterwards, $p(y)$ and $p(x|y)$ are used compute $p(y|x)$ by applying Bayes Rule [37] as follows:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}. \tag{3}$$

The Eq.3 says that the posterior probability can be computed with combination of the likelihood and the prior probability of the data. The evidence here is the normalization

term (by which $\int posteriors = 1$).

To make the difference between the generative and discriminative models clear, let us illustrate an example in which we have two classes of wines, say from Italy $y = 0$ and from France, $y = 1$, and the data set describe the wines with feature x. Our task here is to classify unseen wines into one of these categories using both generative and discriminative models.

Discriminative models, such as logistic regression and support vector machines (SVM) [37], perform the classification using training data as follows:

- Train the model to find a decision boundary, which separates Italian wines from French wines.
- Once the model is trained, it can be used to classify new wines based on the decision boundary. In other words, if $p(x|y)$ is on the left side of the decision boundary shown in Figure 10 the wine is categorized as an Italian wine. Otherwise, it is classified as a French wine.

The procedure described above basically summarizes the way how discriminative models work. The same problem can be solved by generative models like Bayesian Classifier [37] such that:

- First, take the Italian wines from the data set and build a model, which tells *what the Italian wines look like*. This is to learn $p(x|y = 0)$ curve in Figure 10.
- The next step is to take the French wines from the set and build a model, which tells *what the French wines to look like*. This is actually to learn $p(x|y = 1)$ curve in Figure 10.
- The new wine is classified by looking whether it looks more the French or the Italian wine.

Parameter Estimation

After given the intuition and the math behind the generative and discriminative models, as a next stage in the learning process of the generative models, we can explain how the generative models are used to generate a new data sample from the underlying given data distribution.

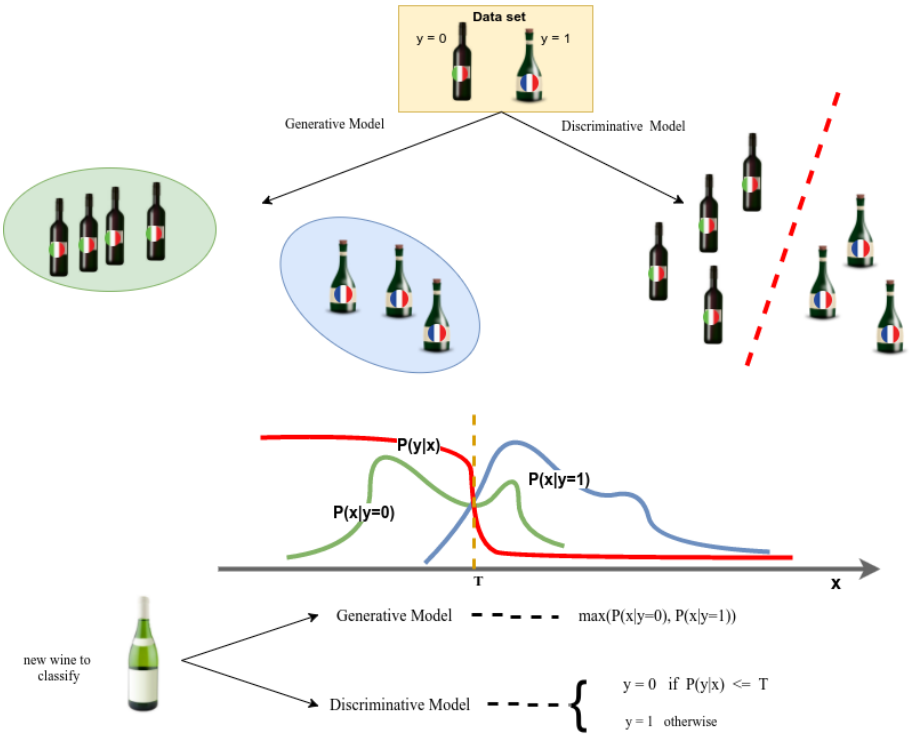


Figure 10. An illustration of the difference between the generative and discriminative models in machine learning based on the wine classification problem. While the generative models learn the characteristics of each wine classes, the discriminative models rather learn the decision boundary between wine classes.

Let us clarify this by using the same wine classification problem (recall that in which the task is to classify the wines whether they are French or Italian wines). Let us assume, we have tasted all the French and Italian wines in the cellar and learn some tricks used to make a wine taste like a French or Italian wine. As an expert now, we are able to distinguish any French wines from Italian ones (Of course, there can a bias, but in this example, it is ignored). After learning about these two types of wines, we would like to make our wines based on the wine samples we have and our goal is to make the wine which should taste as close as possible to the French wine. The task is easier for us as an expert since we have learned the characteristics of each wine and by using these characteristics we can make our own French wines.

The generation of new data samples based on the actual data is nothing more complicated than the above-mentioned wine generation task. In generative models, as wine characteristics, the parameters are learned to generate new data samples. Therefore, the parameter estimation is a crucial task in generative models.

In order to generate new data samples by generative models, there are two stages to follow, which are 1) estimating the model parameters and 2) generating new samples from the

estimated parameters. As has been noted before, the important step is the estimation of the parameters and it can be estimated by using *maximum likelihood estimator (MLE)* in which the goal is to find the parameters that *most likely* explain the given data (or observed data). To simplify that, let us assume we have the data x and the parameter θ associated with the data. The parameters can be mean μ and variance $\sigma^2$of the data by which we can assume that the data have a normal Gaussian distribution with density function of $p(x|\theta)$. MLE tries to find the parameters $\hat{\theta}$ that maximizes $p(x|\theta)$ as in the following equation:

$$\hat{\theta} = \arg\max_{\theta} p(x|\theta). \tag{4}$$

Once the parameters are estimated, new samples can be generated by simply sampling from $\hat{x} \sim p(x|\theta)$.

3.3 Deep Generative Models

The advances in technology have been bringing high computational power together for mining large-scale data sets and enable to run complex models. This change in the research field of machine learning and computer vision particularly has a significant impact on building complex neural networks with deep architectures, so called deep learning [39]. The main obstacle in the previous neural networks has been that they suffer from the overfitting problem due to the large number of parameters to optimize. However, with the availability of high computational power and resources, the issue has been overcome [39]. This development recently has led the scientific community to make significant improvements in deep learning models to solve complex computer vision and machine learning problems [39, 40]. The most recent and interesting development in deep learning field happened to deep generative models. Generative Adversarial Networks (GANs) [41] and variational autoencoders (VAEs) [42] are such generative models that have gained the attention of researchers to apply in synthetic data generation tasks. Thus, in this thesis, synthetic retinal images are generated by applying GANs and VAEs. Following sub-sections cover these models in more details.

3.3.1 Generative Adversarial Networks

The first applied deep generative model for synthetic retinal image generation is Generative Adversarial Networks (GANs) and this section covers the detailed explanation of

GANs with prior to previously conducted studies of GANs.

In 2014, Goodfellow *et al.* [41] proposed an adversarial network framework as an alternative generative model estimation process for deep generative networks, called *Generative Adversarial Networks* (GANs). The main principle of GANs is that there are two neural networks called *Generator (G)* and *Discriminator (D)*. These two networks compete with each other to maximize their gains inspired by the game theory. G draws samples from the random noise distribution and D discriminates whether samples drawn from G or drawn from real data (training data). Since it is explained in [41], one can think of GANs analogous to counterfeiters and police. Here, G as the counterfeiter tries to generate fake money and make use of it without detection. On the other hand, D as the police tries to detect this fake money. The main goal in this context for both parts is to improve their abilities. The building blocks of GANs are demonstrated in Figure 11.

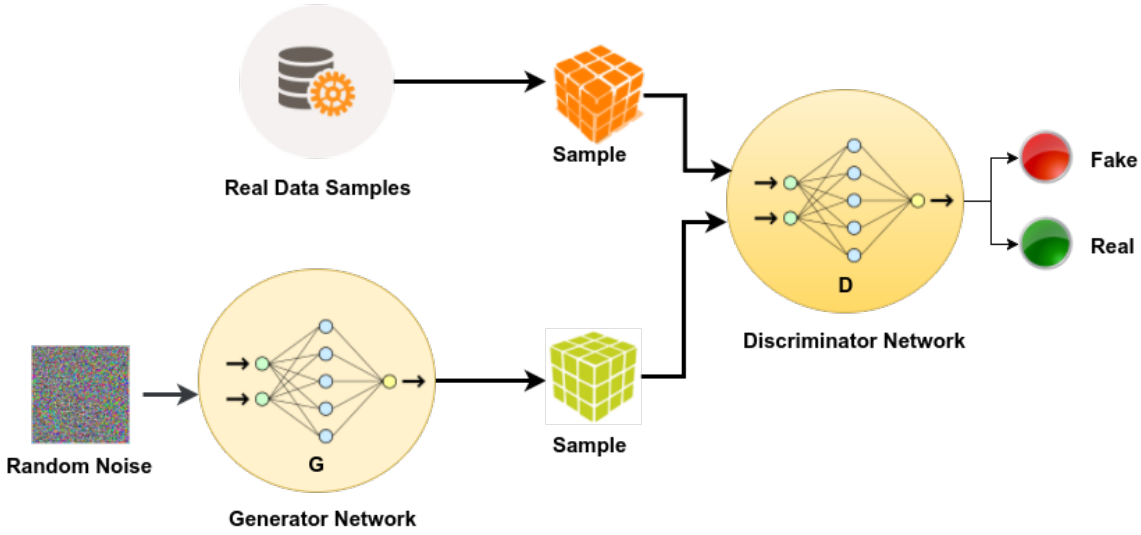


Figure 11. General structure of Generative Adversarial Networks inspired by [41]. The Generator and The Discriminator networks are two different neural networks stacked together to play min-max game for generation of data samples.

In GANs, G learns the distribution p_g over the data x as follows [41]:

1. Define prior input noise variables $p_z(z)$.
2. Construct a mapping to the data space as $G(z;\theta_g)$, where G is defined by a multilayer perceptron with respect to parameters θ_g.
3. Design another multilayer perceptron network $D(x;\theta_d)$. D gives only a scalar value as an output, which indicates whether the sample x is from the real data or the

generated one. The output is higher if the sample is from training data. Otherwise, the output is low for the generated one.

4. Train both D and G at the same time. The goal of D is to maximize the probability of assigning correct labels to an input while G minimizes $\log(1 - D(G(z)))$.

It is clear from the above-mentioned process that D and G actually play a minimax game, which can be formulated with a value function $V(D, G)$ as follows [41]:

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{5}$$

where each part of the equation is defined as follows:

- $E \rightarrow$ expectation,
- $x \sim p_{data}(x) \rightarrow$ x is sampled from the real data,
- $D(x) \rightarrow$ probability of the real data,
- $z \sim p_z(z) \rightarrow$ z is sampled from uniform Gaussian $\mathcal{N}(0, I)$,
- $D(G(z)) \rightarrow$ probability of fake data.

In this context, D is maximizing $D(x)$ and minimizing $D(G(z))$. Hence, the value function associated to D can be formed as follows:

$$\max_D V(D, G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \tag{6}$$

Conversely, G is minimizing the associated value function can be constructed from Eq. 5 as follows:

$$\min_G V(D, G) = E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \simeq \max_G E_{z \sim p_z(z)}[\log(D(G(z)))]. \tag{7}$$

As it was given with the corresponding math, G and D play a min-max game in which both of the players adjust their parameters ($\theta^{(G)}$ and $\theta^{(D)}$) based on the other player's movements [43]. Therefore, the parameters of each player are dependent on the other player's parameters. However, it is important to note that each player does not have a direct control the other player's parameter. Therefore, this scenario is considered a game

rather than an optimization problem and the solution for the game is a Nash Equilibrium [43]. The mathematical explanation of GANs is concisely presented in Figure 12. Also, the training process by Goodfellow is given in Algorithm 1.

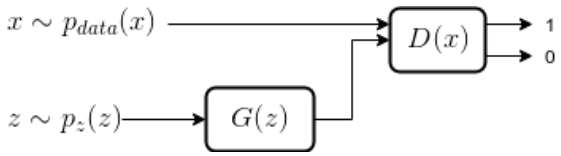


Figure 12. Mathematical representation of GANs based on the intuition described in [43]. $G(z)$ represents the generator network and $D(x)$ is the discriminator network.

The model has been tested on MNIST, the Toronto Face Database (TFD), and CIFAR-10 data sets [41]. It is shown that GANs are able to generate samples quite close to real samples as seen in Figure13.

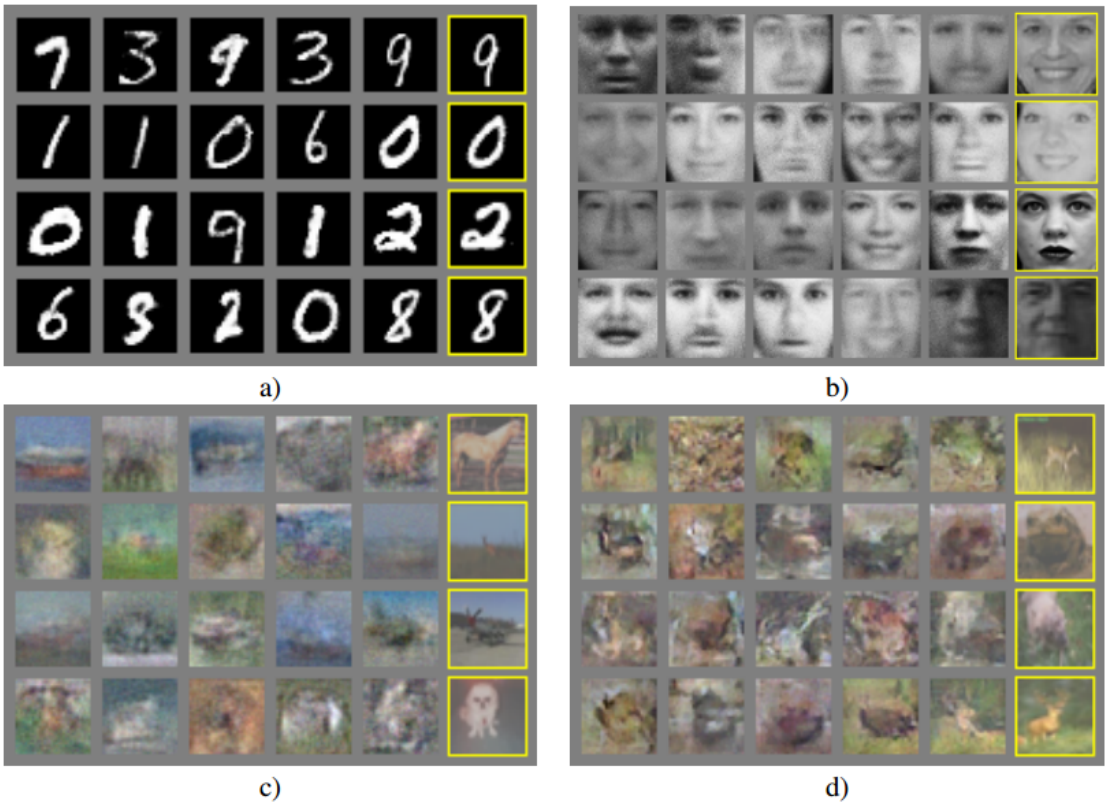


Figure 13. Generated sample images from: a) MNIST; b) TFD; c) CIFAR-10 (fully connected network); d) CIFAR-10 (convolutional network) [41]. The rightmost images are real images from the training sets.

Algorithm 1: Minibatch based training procedure of GAN as defined in [41]. The discriminator is updated k times, which is a hyperparameter and Goodfellow used $k = 1$ by considering computation time.

foreach *number iterations* **do**

  foreach k *steps* **do**

    - Sample minibatch of m noise samples $\{z^{(1)}, ..., z^{(m)}\}$ from noise prior $p_g(z)$.
    - Sample minibatch of m noise examples $\{x^{(1)}, ..., x^{(m)}\}$ from data generating distribution $p_{data}(x)$.
    - Update the discriminator by ascending stochastic gradient descent: $\bigtriangledown_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))]$

  end

    - Sample minibatch of m noise samples $\{z^{(1)}, ..., z^{(m)}\}$ from noise prior $p_g(z)$.
    - Update the generator by descending stochastic gradient descent: $\bigtriangledown_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} [\log(1 - D(G(z^{(i)})))]$

end

Architecture Guideline for Generative Adversarial Networks

An important point to consider while designing a GANs architecture is that there is particular building blocks of CNN which have been proven to accelerate and to avoid model collapses. Therefore, it is essential to review these building blocks (or design patterns).

In the original paper [41], GANs are designed to apply traditional neural networks with hidden layers and neurons. However, the drawbacks of GANs, such as model collapse and its unstable nature during the training have led the researchers to find alternative approaches to design and train the GANs. As the CNN based deep neural networks are proven to be a powerful approach in classification tasks [39, 40], most studies of GANs are built on CNN architectures [44, 46, 50, 69]. As a result, the best practices for stable training of the CNN based models are improved and explored. Particularly, Radford *et al.*[69] have conducted a comprehensive study to reveal best practices for accelerating the training of GANs in a stable way. The important changes demonstrated in this study and in the aforementioned studies [43] are as follows:

1. Rectified Linear Unit (ReLU) as the activation function.

2. Batch Normalization (BatchNorm) as the normalization layer.
3. Leaky ReLU as the activation function.
4. Deconvolution/UpSampling Layer as the upscaling layer.

Because of the nonlinearity introduced by the activation functions, they are essential design patterns while building CNN architectures. It can be said that without the nonlinearity, CNN is only able to perform linear classification. In particular, there are four main activation functions [70] used in the deep neural networks as listed below:

- **Sigmoid** takes a real value as an input and maps it into the range between $[0\ 1]$. In previous GANs designs, it is shown that sigmoid generates blurry and saturated images [44, 46]. Also, due to the non-centered outputs provided by this function, it is not preferred in the state-of-the-art CNN architectures. **Tanh** function maps a real value in the range between $[-1\ 1]$. Unlike the sigmoid, the output is zero-centered and the generated image is sharpened.
- **ReLU** has been introduced recently to accelerate the convergence speed of deep neural networks [71] and applied in most of the previously developed GANs architectures [44, 46, 50, 69, 43]. In contrast to tanh and sigmoid, ReLU is not computationally expensive. However, an important issue of ReLU is that it triggers a dying neuron problem, called Knockout Problem[71] that occurs when the learning rate is large. Since with the large learning rate, the gradients get larger and as a result of that, some neurons might not be activated again.
- **Leaky ReLU** basically solves the dying neuron problem by using a small negative slope [71].

The next pattern widely applied in the previous design GANs is *Batch Normalization* [72], which reduces the training time of the whole network by enabling normalization at particular applied layers. Traditionally, normalization is considered a preprocessing step in machine learning tasks. A common problem encountered in many neural network models is the weight initialization that often causes the overfitting [39]. Although a dropout layer is used earlier in neural networks to prevent overfitting, it has been shown that Batch Normalization actually gives better model accuracy and enables faster learning [43] by preventing the overfitting. One can also think of Batch Normalization as a regularizer that allows the data flow between the intermediate layers in whitened form.

Another key pattern to consider while constructing a CNN is the *Up Sampling* layer [69]. Unlike conventional deconvolution layer, Up Sampling layer takes an image and upscale the image to higher dimensions by computing each pixel value with bilinear interpolation.

Applications of Generative Adversarial Networks

The explained model is basically a utilization of neural networks. Denton *et al.* [44] extended the GANs to convolutional neural networks (CNN) called *LAPGAN*. In this version, images are generated with a framework of Laplacian pyramid [45]. The main contribution of the paper is that the series of GANs are built and each of them operates at a specific scale of a Laplacian pyramid to capture a specific structure of the image.

An important issue of the GANs is that it takes quite a long time for D to minimize the loss. In addition to that, as a black box, it does not provide any ability to control the learning process with additional information, which can help to accelerate the learning process and generate more realistic data samples. To overcome these issues, Gauthier [46] studied the GANs for face generation task on conditioned to prior knowledge. By adding the conditioning ability to the GANs, a conditioned deconvolutional neural network structure, called *cGAN*, is proposed as shown in Figure 14.

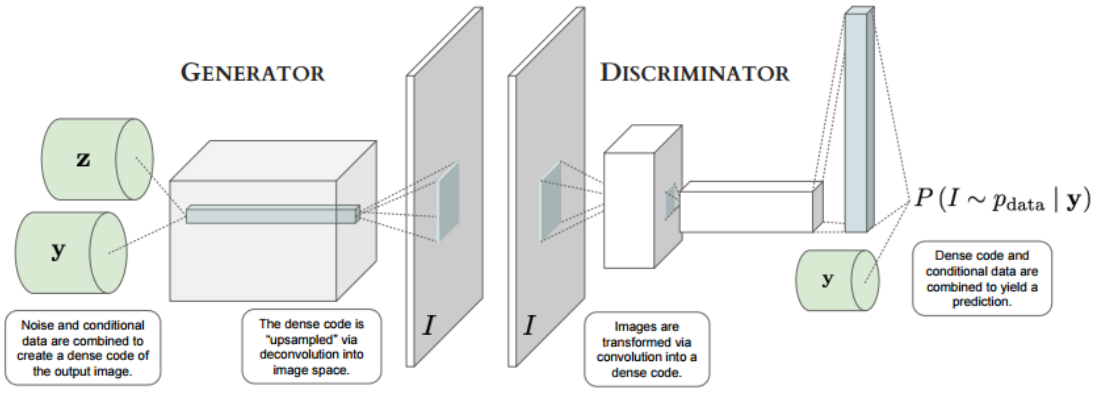


Figure 14. Conditional Generative Adversarial Network structure as defined in [46]. The main difference between the GANs structure described in Figure 11 and cGAN is that the generator takes also a conditional data as an input.

A prominent factor here, which has a huge impact on training of both G and D, is a conditional data vector y. In the study, facial expressions (emotions) and some other face related features such as age (baby, child, senior) and race (black, white, Indian and Asian) are chosen as y. After incorporating y with G and D, they rewrite the defined value func-

tion in Eq. 5 as follows:

$$\min_G \max_D V(D,G) = E_{x,y\sim p_{data}(x,y)}[\log D(x,y)] + E_{y\sim p_y, z\sim p_z(z)}[\log(1 - D(G(z,y),y))]. \tag{8}$$

The proposed models, cGAN and GAN, are applied to The Labeled Faces in the Wild data set [47] and it is shown that cGAN is able to generate more realistic images than GAN. However, this model can be extended to some other applications with a deeper structure, because only a single layered deconvolutional neural network is utilized in the model. In addition to that, one can think of changing the place of the conditional vector y where it is inserted. In this framework shown in Figure 14, it is inserted in both G and D at the last dense layer.

As a deep generative model, the ability of GANs to learn and generate images has attracted researchers to improve and try this approach on different problem domains. For instance, the model called InfoGAN [48] is presented that learns the conditional variable by itself. InfoGAN takes advantage of information theory, which implies the use of mutual information for better model performances. This variable can be a label info or some continuous variables such as rotation angle, and edge structure -depending on the problem domain-. Inspired by cGAN, Yin Weidong *et al.* proposed a new model called Semi-Latent GAN [49]. This model is capable of: (1) generating facial images conditioned on high-level facial features like smiling, male/female, pale skin, etc., (2) modifying facial images conditioned on these facial attributes. Along with these studies, the study in [50] is applied GANs for generation of single super-resolution image from a degraded and downscaled image. The model is called SRGAN and to the best of our knowledge it is the only GANs type that generates high-resolution image from its pair.

In computer vision, image captioning is a well-known problem in which the context of an image can be described textually based on extracted features from the image [39]. To utilize the GANs in this problem domain, Zhang *et al.* decided to reverse the problem and proposed an interesting version of GANs called StackGAN[51] to generate the images from text descriptions. StackGAN has two stacked GANs. First GANs take a text description and generates some basic shapes like edges and corners or color, afterward the second GAN take the output of first GANs together with the text description to generate the high-resolution image. An interesting application of GANs is studied in astrophysical image data [52] for the purpose of feature extraction.

3.3.2 Variational Autoencoders

GANs are considered a transformation learning technique in which randomly drawn noise (in most cases from a uniform distribution) is used to form data samples [43]. Different from GANs, there is a probabilistic graphical deep generative model proposed to generate data samples from a latent space by Kingma and Welling [42] and Rezende *et al.* [53], in which they combined the deep neural networks with Bayesian inference models. This generative directed graphical model is called *variational autoencoders (VAEs)*, a model which is built on the concept of autoencoders. Together with GANs, VAEs are studied in this thesis to generate synthetic retinal images. Thus, this section will explain the details of VAEs.

Latent Space Modeling

In order to understand the core of VAEs comprehensively, it is essential to start from scratch and go through what actually a latent space means and why it is widely used in machine learning.

The latent space primarily provides the information about the underlying hidden structure inside the data [54]. In other words, the unknown characteristics of the data can be accessed via a hidden representation of the data. In real life problems, quite often the informative part of the data are not observed directly, therefore, the features are extracted via its hidden structure. Thus, it is important to make inference out of this abstract property of the data. An obvious example can be given from human psychology. For instance, from the outside, we can only observe the behavior of a person. However, the intention of the person before attempting any action refers to the latent variable, which cannot be observed directly and in most cases, to be able to understand the reasons behind these actions, we try to get to know the person by exploring the hidden intention. Inspired by this scenario, the same concept is also applied in machine learning area, in which the abstract/hidden property of the data is discovered for further analysis on the data. VAEs are such generative models that help to learn the latent variables and then to use these variables for generating new data samples.

In probabilistic models like VAEs, the latent space is used to draw samples from some probability density functions (p.d.f.) that is accepted as a representation of the data in the hidden space. However, for such probabilistic model, it is important to determine that for each data instance, $X = \{x^{(i)}{}_{i=1}^{N}\}$, there is one or more than one latent variable z that can be used to generate a sample similar to $x^{(i)}$ by sampling from p.d.f. over latent space Z. This addresses that each observation $x^{(i)}$ actually is dependent on the latent variable z.

As mentioned earlier, the main goal in generative models is to maximize the likelihood of the observed data. Thus, we can formulate the density of our data by simply taking into account all possible forms of latent variables with respect to prior $p(z)$ and marginalize the density of the data with joint distribution $p(x, z)$ as follows:

$$p(x) = \int p(x, z)dz = \int p(x|z)p(z)dz. \tag{9}$$

Afterwards, the inference of latent variables can be obtained by the Bayes rule as follows:

$$p(z|x) = \frac{p(x|z)p(z)}{\int p(x|z)p(z)dz}. \tag{10}$$

Autoencoders

Traditionally, autoencoders are formed with 3-layer neural networks in which the data are encoded into a latent code by the encoder block and then it is reconstructed from the latent representation of the data by the decoder block [55]. In the context of autoencoders, the data x is mapped into the latent vector z by encoder function $z = f(x)$ and it is reconstructed by the decoder function $\tilde{x} = g(z)$ from z as illustrated in Figure 15. The autoencoders are commonly used to represent the data in lower dimensions. The training process of autoencoders is relatively simple and the goal is to minimize the reconstruction loss in Eq. 11 defined by a mean squared error (MSE) between the actual data x and the reconstructed data $\tilde{x}$.

$$\mathcal{L} = \|x - \tilde{x}\|_2^2 = \|x - g(f(x))\|_2^2 \tag{11}$$

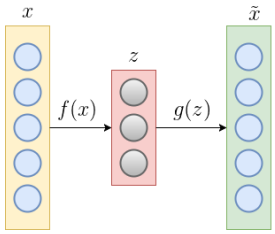


Figure 15. Demonstration of general structure of autoencoders inspired by the study in [55]. The encoder maps the actual data sample into the latent vector $z = f(x)$ and the decoder reconstructs the data from the hidden vector $\tilde{x} = g(z)$.

Variational Autoencoders

Although the autoencoders are able to model the latent space of the data, they cannot be used to generate new data samples. Variational autoencoders (VAEs) are such deep generative probabilistic models in where the characteristics of the autoencoders meet the

Bayesian Inference. Thus, the VAEs can be applied to generate new data samples from a distribution of the latent code. In particular, the power of the VAEs comes from *its capability of controlling the distribution of the latent vector* z . To do so, it adds a constraint on the encoder block by which the encoder block is reinforced to encode the data into a latent space with the unit Gaussian distribution.

More precisely, the VAEs consists of two primary blocks, which are the *Encoder* and the *Decoder*. The data sample (from now on, the data sample refers to an image in this thesis) $x^{(i)}$ is first encoded to a latent vector $z = Encoder(x^{(i)}) \sim q_\phi(z|x^{(i)})$ by the encoder block, and afterwards the second block is used decode this latent vector z into an image as similar as possible to the original image $\tilde{x} = Decoder(z) \sim p_\theta(x^{(i)}|z)$ [42, 53] (see Figure 16 for the graphical illustration). The decoder block forms the reconstruction loss $\mathcal{L}_R$ and the encoder block constitutes the regularizer term $\mathcal{L}_{KL}$ which is the Kullback-Leibler (KL) divergence. By combining $\mathcal{L}_{KL}$ and $\mathcal{L}_R$, the loss function of VAEs $\mathcal{L}_{VAE}$ is defined in Eq. 17 [42].

An important point in Eq. 17 to understand is the KL divergence, which is the only regularizer term in the VAEs. Therefore, it is worthwhile to explain the KL divergence in more details. The KL divergence between two distributions $p(x)$ and $q(x)$ is given as follows:

$$D_{KL}(q(x) \parallel p(x)) = \int q(x) \log \frac{q(x)}{p(x)} dx \tag{12}$$

KL divergence measures how closely $p(x)$ and $q(x)$ match with each other. However, it is crucial to not to consider KL divergence as a distance measure because $D_{KL}(q(x) \parallel p(x)) \neq D_{KL}(p(x) \parallel q(x))$.

In case of VAEs, as noted earlier, we have a constraint on the latent variable to be the unit Gaussian $p(z) = \mathcal{N}(0, I)$. Therefore, we employ KL divergence to measure how different $q_\phi(z|x^{(i)})$ from $p(z)$. This indicates that we also want $q_\phi(z|x^{(i)})$ to be Gaussian as follows:

$$
\begin{aligned}
D_{KL}(\mathcal{N}(\mu, \sigma^{(2)}) \parallel \mathcal{N}(0, I)) &= \int q_\phi(z|x) \log q_\phi(z|x) dz - \int q_\phi(z|x) \log p(z) dz \\
&= \int \mathcal{N}(\mu, \sigma^{(2)}) \log \mathcal{N}(0, I) dz \\
&- \int \mathcal{N}(\mu, \sigma^{(2)}) \log \mathcal{N}(\mu, \sigma^{(2)}) dz \\
&= \frac{1}{2} \sum_{j=1}^{J} (1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2) \qquad (13)
\end{aligned}
$$

As the decoder block in VAEs is responsible for generating a sample image as close as possible to the real image samples in the data, the reconstruction loss $\mathcal{L}_R$ is corresponding to the maximization of the marginal log-likelihood of each image in X. It can be also defined as an expected negative log-likelihood loss function $\mathcal{L}_R$ and it can be approximated by applying Monte Carlo estimation procedures [42] as in Eq. 14:

$$
\mathcal{L}_R = -E_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] \approx \frac{1}{L} \sum_{l=1}^{L} \log p_\theta(x^{(i)}|z^{(i,l)}). \qquad (14)
$$

It has been proven that $\mathcal{L}_{KL}$ can be minimized using gradient descent [53]. The overall goal of the training phase is to optimize Eq. 17 by applying gradient descent.

$$
\mathcal{L}_R = -E_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] \qquad (15)
$$

$$
\mathcal{L}_{KL} = D_{KL}(q_\phi(z|x^{(i)}) \parallel p_\theta(z)) \qquad (16)
$$

$$
\mathcal{L}_{VAE} = \mathcal{L}_R + \mathcal{L}_{KL} \qquad (17)
$$

where $x^{(i)}$ is an input image, z is a latent vector, $p_\theta(z)$ is a prior probability of the latent vector z, $q_\phi(z|x^{(i)})$ is a posterior of the encoder as a recognition model, $p_\theta(x^{(i)}|z)$ is a posterior of the decoder as a generative model, ϕ is a variational parameter and θ is a generative parameter. The objective here is to learn both parameters ϕ and θ jointly. The

proposed algorithm from the original paper [42] is presented in Algorithm 2.

Algorithm 2: Minibatch based training procedure of VAEs as defined in [42]. M is the number of samples in each batch.

$\theta, \phi \leftarrow$ Initialize parameters
repeat
  $X^M \leftarrow$ Get a random minibatch of M data points
  $\epsilon \leftarrow$ Get samples from noise distribution $p(\epsilon)$
  $g \leftarrow \nabla_{\theta,\phi} \mathcal{L}^M_{VAE}(\theta, \phi; X^M, \epsilon)$ (Gradients of minibatch estimator)
  θ, ϕ update parameters
until *convergence of parameters* (θ, ϕ)

In this thesis, VAEs are combined with deep neural networks to generate retinal images. Hence, Eq. 17 is formed based on the studies [42, 53] as shown in Figure 16. For both $z = Encoder(x^{(i)}) \sim q_\phi(z|x^{(i)})$ and $\tilde{x} = Decoder(z) \sim p_\theta(x^{(i)}|z)$, multi-layer perceptrons are used to learn parameters ϕ and θ jointly.

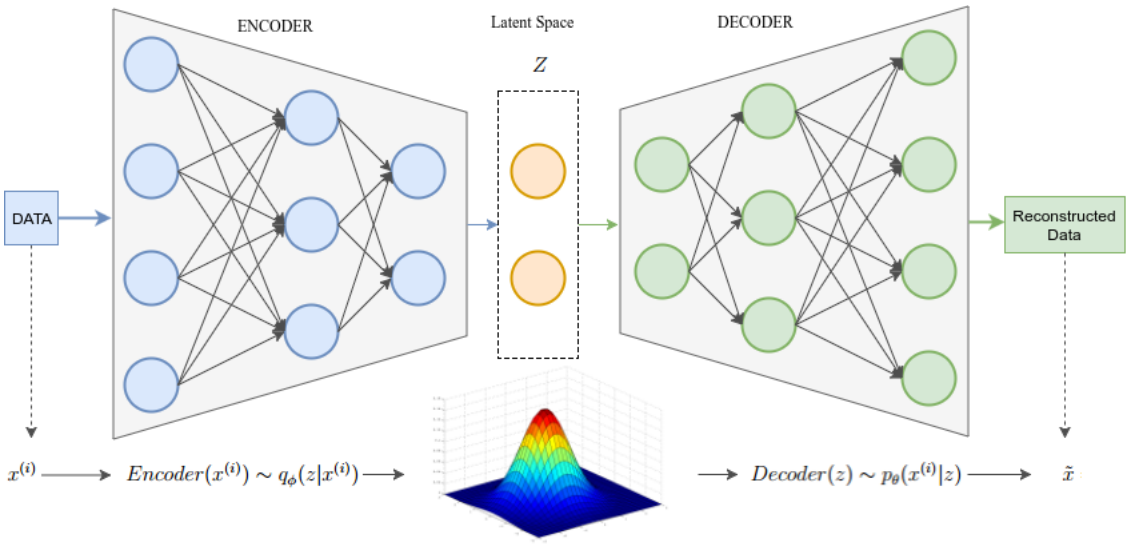


Figure 16. Graphical illustration of VAEs based on the studies [42, 53]. Each data point $x^{(i)}$ is mapped into the latent space Z by the Encoder block and the Decoder block generates new data samples by sampling a latent variable z from the latent space.

To derive Eq. 17 with these assumptions by considering Bayesian inference models, let the prior of latent variable be a multivariate Gaussian $p_\theta(z) = \mathcal{N}(z; 0, I)$ and $p_\theta(x^{(i)}|z)$ be a multivariate Gaussian for which the parameters of distribution are estimated by MLP[2].

[2]*No that, the distribution can be Gaussian or the Bernoulli depending on the data because in the case of binary data Bernoulli p.d.f. is preferred*

Finally, the variational posterior $q_\phi(z|x^{(i)})$ defined as follows:

$$\log q_\phi(z|x^{(i)}) = \log \mathcal{N}(z; \mu^i, \sigma^{2(i)} I) \tag{18}$$

where mean μ^i and variance $\sigma^{2(i)}$ are the outputs of the MLP based encoder. By sampling from the posterior $z^{i,l} \sim q_\phi(z|x^{(i)})$ and considering the way how KL divergence is differentiated in [56], the final equation for VAEs can be formed as follows:

$$\mathcal{L}_{VAE} \simeq \frac{1}{2} \sum_{j=1}^{J} (1 + \log((\sigma_j^{(i)})^2) - (\mu j^{(i)})^2 - (\sigma_j^{(i)})^2) + \sum_{l=1}^{L} \log p_\theta(x^{(i)}|z^{(i,l)}) \tag{19}$$

where $z^{(i,l)} = \mu^{(i)} + \sigma^{(i)} \odot \epsilon^{(l)}$ and $\epsilon^{(l)} \sim \mathcal{N}(0, I)$, L is the number of drawn samples, J is the number of latent variables and $\odot$ is the element-wise multiplication.

An important point in Eq. 19 is that the latent variable $z^{(i,l)}$ is reparametrized in order to perform backpropagation as part of MLP. This is called *reprametrization trick* and explained in the following subsection.

Reparametrization Trick

In order to apply backpropagation in VAEs, the random variable in latent space is transformed from $z \sim \mathcal{N}(\mu^i, \sigma^{2(i)} I)$ into $z^{(i,l)} = \mu^{(i)} + \sigma^{(i)} \odot \epsilon^{(l)}$. This is called reparametrization trick [42] and it enables the VAEs to be optimized by gradient descent since the network does not have any random variables now. With this adjustment, the whole mathematical procedure is demonstrated in Figure 17.

Eq. 19 is optimized by using Gradient Descent for each image $x^{(i)}$. The more details regarding the derivation of the equation with MLE can be found in [42]. In Eq. 19 the first term measures the reconstruction performance of the model on the given input and the second term is KL divergence computed using the prior and the approximated posterior, which is responsible for keeping latent variables near prior by regularizing them.

Applications of Variational Autoencoders

As in the case of GANs, VAEs recently have been used in many application areas to generate images [56] and it provides quite realistic results (refers to Figure 18) by comparing to real images used to train the model. For instance, by using a CNN architecture rather than a fully connected network architecture, Tejas *et al.* [57] succeeded to generate images of the same object with different variations in pose and lighting. In another study proposed by Yunchen *et al.* [58], VAEs are used to generate captions and labels for images. In addi-

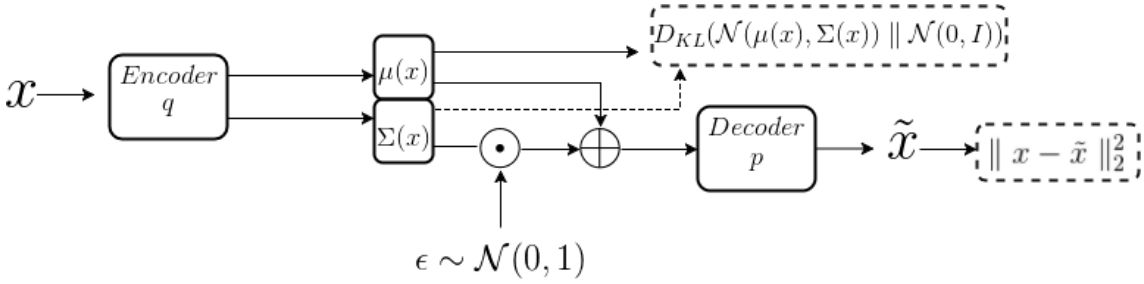


Figure 17. Illustration of math behind VAEs inspired by [42, 53]. Given a real data sample x as an input to the encoder function q, the encoder computes mean $\mu(x)$ and covariance matrix $\Sigma(x)$, which are further used by decoder function p to generate new data samples. ϵ stands for a noise drawn from unit Gaussian distribution. The dashed rectangles are corresponding to the cost functions, which are Kullbeck-Leibler divergence D_{KL} and the reconstruction loss.

tion to these papers, in [59] the authors combined VAEs and Recurrent Neural Networks to generate the text description of an image.

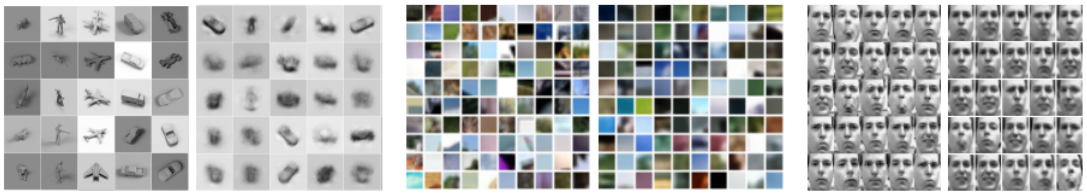


Figure 18. Sample generated images using VAEs from the data sets: a) NORB; b) CIFAR-10; c) Frey [53]. Images on the left hand side the actual images from training set while the images on the right hand side the generated images via VAEs.

The success of both VAEs and GANs has directed researchers to combine both of the models for the generation of high-quality images. For instance, recently Mahesh Gorijala and Ambedkar Dukkipati in [60] applied GANs to generate images based on InfoGAN by conditioning visual descriptors and modify the generated image by learning a latent representation of the data. The combination of GANs and VAEs is also used in the medical field to generate semi-synthetic electronic health record (EHR), the model called medGAN [61]. A general framework introduced by Alireza *et al.* [62] extends the use of probabilistic autoencoders and GANs for several semi-supervised applications, such as changing the style of an image, unsupervised clustering,and dimensionality reduction.

3.4 Retinal Image Quality Assessment

In order to quantify the quality of the generated retinal images, the quality assessment is the crucial part. Thus, this section covers the details of several measurements used to evaluate the generated retinal images.

Earlier studies of GANs and VAEs have suggested that the quality of the generated images can be evaluated by performing classification, segmentation and feature extraction methods based on the problem domain [41, 45, 49, 42, 58]. However, by taking into consideration previously used and developed methods for retinal image quality assessments [63] and the structure of the generated retinal images (which is covered in Section 4), in this thesis the quality assessment of the generated retinal images is performed in a different way such that statistical analysis of the retinal images is conducted for measuring similarity between generated images and a benchmark high-quality retinal image set DiaRetDB1 [64].

Statistics is a commonly used set of tools for data analysis based on some experiments from different disciplines such as medicine, engineering, sociology, etc. There are a variety of statistical measures including mean, standard deviation, variance, mode, median, skewness, kurtosis, and entropy, which are often applied in image processing tasks [65]. In the scope of this thesis, by considering prior research conducted for retinal images [66], *mean, variance, entropy, skewness* and *kurtosis* are used to extract the statistical features from the retinal images for the quality assessment.

The proposed framework for retinal image quality assessment is illustrated in Figure 19. The procedure is relatively simple and starts with extracting the above-mentioned features from the generated retinal images and DiaRetDB1. As the retinal images have three channels (RGB), this leads us to compute the features for each channel separately. Afterward, the comparison between those features is done by applying a similarity-based approach, which uses cosine similarity metric. Given the proposed framework for the quality assessment, it would be better to go through each statistical feature for better understanding and interpretation of the obtained results. The details of the statistical features in the context of image processing is given as follows:

* *Mean* is a measurement of the intensity level of a pixel in an image. Higher intensity

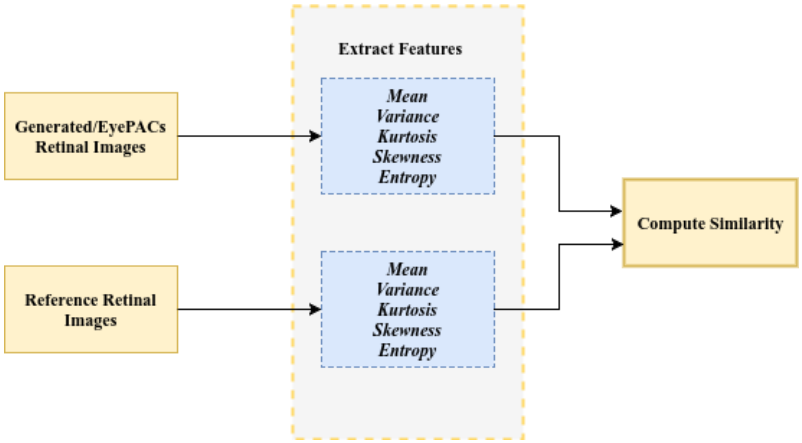


Figure 19. Proposed scheme for the assessment of the generated retinal images quantitatively by using the statistical features extracted from the reference retinal image set and the generated images from GANs and VAEs.

implies a bright image. Mean of an image is defined as follows:

$$\mu_I = \frac{1}{W \times H} \sum_{r,c} I(r,c) \tag{20}$$

where I is an image, W and H are width and height of the image respectively, r and c indicate the row and column indices in the image.

* *Variance* is associated with the contrast level of an image and it captures variability (or diversity) of pixel values in the image. More precisely, it measures the difference between the pixel value and the mean intensity value. Low variance indicates a low contrast, while high variance implies high contrast. The unbiased estimate of the variance is defined as follows:

$$\sigma_I^2 = \frac{1}{W \times H - 1} \sum_{r,c} \Big(I(r,c) - \frac{1}{W \times H - 1} \sum_{r,c} I(r,c) \Big)^2 \tag{21}$$

* *Skewness* is associated with symmetry of the probability distribution of an image. The negative value of skewness expresses the left skewed (or tailed) p.d.f. while the positive skewness values indicate the right skewed p.d.f. For instance, darker regions in an image are positively skewed. Mathematically,

$$s_I = \frac{\sum_{r,c} \Big(I(r,c) - \mu_I \Big)^3}{(W \times H - 1)\sigma_I^3} \tag{22}$$

* *Kurtosis* describes the shape of the p.d.f. with respect to the normal distribution. When the kurtosis value is high, it can be said that the p.d.f. has a peaked histogram.

In contrast to that, a lower value indicates a flat histogram of an image intensity value. Mathematically,

$$k_I = \frac{\sum_{r,c} \Big(I(r,c) - \mu_I \Big)^4}{(W \times H - 1)\sigma_I^4} \tag{23}$$

* *Entropy* is used as a measure of disorder (or randomness) in image processing tasks. For instance, there are certain patterns/textures in an image and those patterns occur at many places in the image. This indicates low variance, low disorder, and low entropy (homogeneity) in an image. Yet, the high entropy implies high contrast in an image. The entropy is defined as follows:

$$H(I) = \sum_i h_I(i) \log N / h_I(i) \tag{24}$$

where I is an image, $h_I(i)$ is the frequency of intensity i and N is the total number of pixels in the image I.

In addition to aforementioned statistical features, the similarity metric is formulated as follows:

* *Cosine similarity* is applied to reveal similarity of two vectors based on the angle between the vectors instead of the distance between the magnitude of the vectors. It computed as a dot product of two normalized vectors as follows:

$$d(x,y) = \frac{xy}{\sqrt{(xx)(yy)}} \tag{25}$$

where x and y are vectors.

4 EXPERIMENTS AND RESULTS

This section presents the details of the experimental analysis as a utilization of the deep generative models explained in the earlier chapter. This chapter will cover: (1) the details of the data sets used for both model training and quality assessment of generated retinal images; (2) the details of the architectural design of the deep generative models; (3) presentation and analysis of the generated retinal images by GANs and VAEs; and (4) the quality assessment of the generated retinal images.

The implementation of the methods is done by using Keras [67], a deep learning framework written in Python and the source code with additional figures will be accessible on Github[3].

4.1 Data Sets

We employed two different data sets to conduct experiments. EyePACS retinal data set [68] are used for model training, whereas DiaRetDB1 [64] is used for quality assessment of the generated retinal images. More details of the both data sets are given in following subsections.

4.1.1 EyePACS

The data set used in this thesis comes from *Diabetic Retinopathy Detection* competition held on Kaggle [68] to detect diabetic retinopathy (DR) and to improve screening of DR. The data set for the competition are provided by EyePACS. The images are obtained from eye hospitals located in the United States and India.

The retinal images were taken by fundus photography using different cameras and camera models. Hence, as one can see in Figure 20, the resolutions and visual appearances of retinal images vary significantly. In some images, it is possible to see almost all details of the retina, including macula, optic disc, and vessel structure. However, in some images, those parts are not clear or they can be seen partially. In the images, there is also noise and some artifacts caused by cameras. This diversity of the images can lead to train a more robust model for segmentation, detection and classification tasks in the field.

[3] *The source code and the materials:* `https://github.com/kaplansinan/MasterThesis`

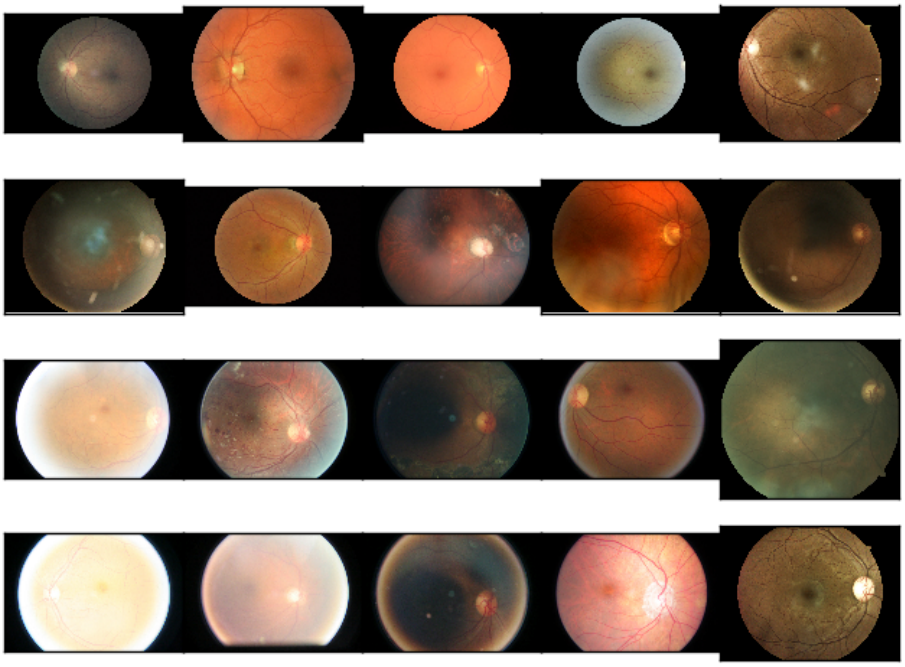


Figure 20. Sample images from the data set provided by EyePACS [68]. In the provided set, the retinal images were taken by different cameras, thus, they have different resolutions.

For the scope of this thesis, by considering computational resources and time to train the whole network, only the training set provided by EyePACS is used. The training set contains 35162 high-resolution retinal images in total.

4.1.2 DiaRetDB1

DiaRetDB1 [64] data set used as a benchmark retinal data set for the quality assessment of the generated retinal images. This set is mainly created for diabetic retinopathy related tasks in the field, and it consists of 89 high-resolution images in total. Some notable retinal images from the set are shown in Figure 21.

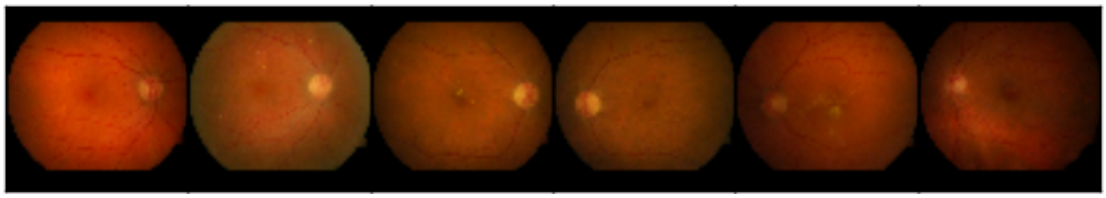


Figure 21. Retinal image samples from the DiaRetDB1 set [64] used to evaluate the quality of generated retinal images.

4.2 Architectural Design of Deep Generative Models

This section covers the details of the proposed architectures for GANs and VAEs. The reviewed studies are taken as a guideline while developing and training the models.

4.2.1 Architecture of Generative Adversarial Networks

Considering the architecture guideline for GANs presented in the previous section, the architecture of the proposed GAN is designed with prior to the previous studies [44, 46, 50, 69], which are taken as a reference. The proposed deep model together with modifications in the layers and the hyperparameters for *Generator (G)* and *Discriminator (D)* (which are the building blocks of GANs) is visualized in Figure 22 and 23 respectively.

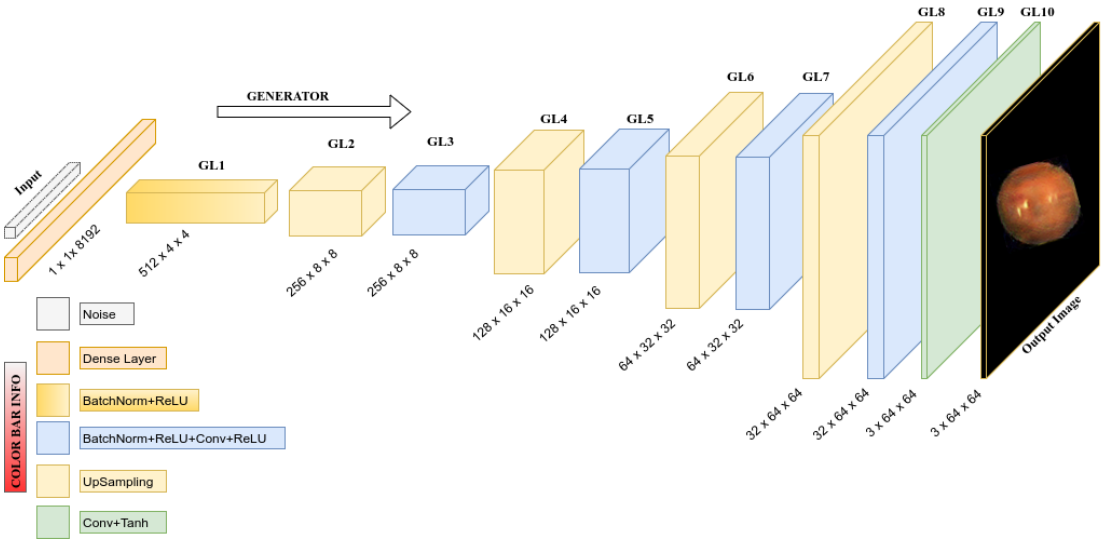


Figure 22. Architecture of proposed generator network, which is part of GAN, based on CNN units with the details, including layers and their dimensions.

Figure 22 demonstrates the layers of G with specific dimensions of each layer *(Depth* $\times$ *Width* $\times$ *Height)*. Color bar in the figure shows the type of each layer. The generator network consists of 10 layers $(GL1 \rightarrow GL2 \rightarrow GL3 \rightarrow GL4 \rightarrow GL5 \rightarrow GL6 \rightarrow GL7 \rightarrow GL8 \rightarrow GL9 \rightarrow GL10)$ - *GL stands for Generator Layer*- and each of them is formed by different building blocks of the CNN as seen in Figure 22. The input is a noise vector of dimension (1×100) dimension drawn from unit Gaussian distribution. The generator basically transforms this input vector into a $(3 \times 64 \times 64)$ dimensional retinal image by passing through each layer in G.

Additionally, the same concept is used to define the layers of D. It is constituted with 6 layers $(DL1 \rightarrow DL2 \rightarrow DL3 \rightarrow DL4 \rightarrow DL5 \rightarrow DL6)$ - *DL stands for Discriminator Layer* - as shown in Figure 23. The discriminator takes an image (an image from an actual set or an image from the generator) and calculates the probability of the image via softmax layer to assign a label to this image. The label has only two different values. The first value (0) is assigned to the images from the generator network while the second value (1) is given to the images from the real set.

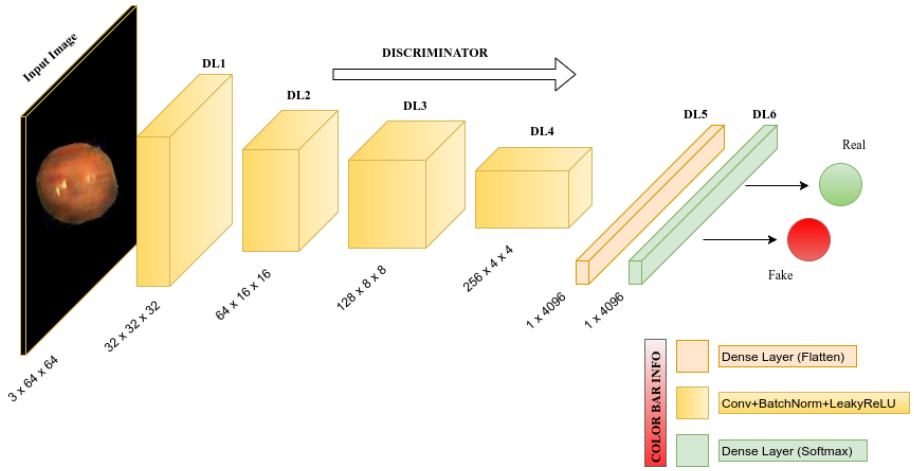


Figure 23. Architecture of proposed discriminator network, which is part of GAN, based on CNN units with the details, including layers and their dimensions.

4.2.2 Architecture of Variational Autoencoders

The variational autoencoder architecture is rather simpler than the architecture of GAN and it is based on multilayer perceptrons as seen in Figure 24. By taking studies in [42, 53] as the reference, both the encoder and the decoder blocks of VAEs are designed to have only one hidden layer with 512 neurons in total. The encoder block takes an image $(3 \times 64 \times 64)$ from the training set and encodes it into latent space by learning mean μ and variance σ^2. Similarly, the decoder samples a noise vector z from the Gaussian distribution with learned mean μ and variance σ^2 and reconstructs it to an image with dimension of $(3 \times 64 \times 64)$.

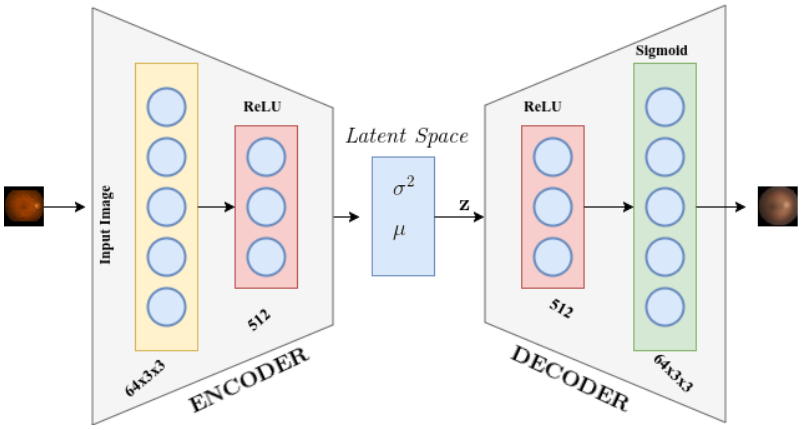


Figure 24. Proposed VAEs architecture based on multi layer perceptrons, which is composed by a single hidden layer both in encoders and decoders.

4.3 Retinal Image Generation via Generative Adversarial Networks

Based on the introduced structure of the GAN (see Figure 22 and 23) in the previous section, the whole network is trained to generate the retinal images by using the EyePACS data set. The potential of the GANs with the issues encountered is explored and stated in this section.

As has been noted in the earlier studies of the GANs, while training the network it is important to choose and define the right hyperparameters. In the proposed GAN design, the hyperparameters consist of filter size of the convolution layers, the parameters associated with the optimizer(s), the number of the epochs to run the whole network and the mini-batch size. Prior to the aforementioned studies of GANs, the discriminator (D) and the generator (G) are optimized by Adam Optimizer [73]. The hyperparameters associated with the Adam Optimizer are learning rate lr, β_1 and β_2, and ϵ as given with the values in Table 1. Also, taking the design proposals from the studies of GANs in Section 3.3.1 into consideration, the stacked network is optimized by applying Stochastic Gradient Descent (SGD) with learning rate lr and momentum m as the hyperparameters shown in Table 1.

Table 1. Hyperparameters associated with the training procedure of GAN: (1) the parameters associated to Adam Optimizer are learning rate lr, coefficients β_1, β_2 and ϵ; (2) the parameters of SGD are learning rate lr and momentum m; and (3) batch size and number of epochs to train the network.

	Adam Optimizer				SGD Optimizer			
Filter size	lr	β_1	β_2	ϵ	lr	m	batch size	epochs
3×3	0.001	0.5	0.9	$0.1 \times 10^{(-7)}$	0.01	0.9	32	400

The samples of generated retinal images from the proposed GAN are demonstrated in Figure 25. In the context of qualitative analysis of the generated images, the following observations can be made:

- The network is able to capture the global structure of the retina, including *shape, optic disk, fovea, and macula*.
- However, the internal structure of the retina, such as vessel trees is not generated clearly.
- Additionally, the network is able to generate the retinal images with varying colors as in the EyePACS set (refer to Figure 20).

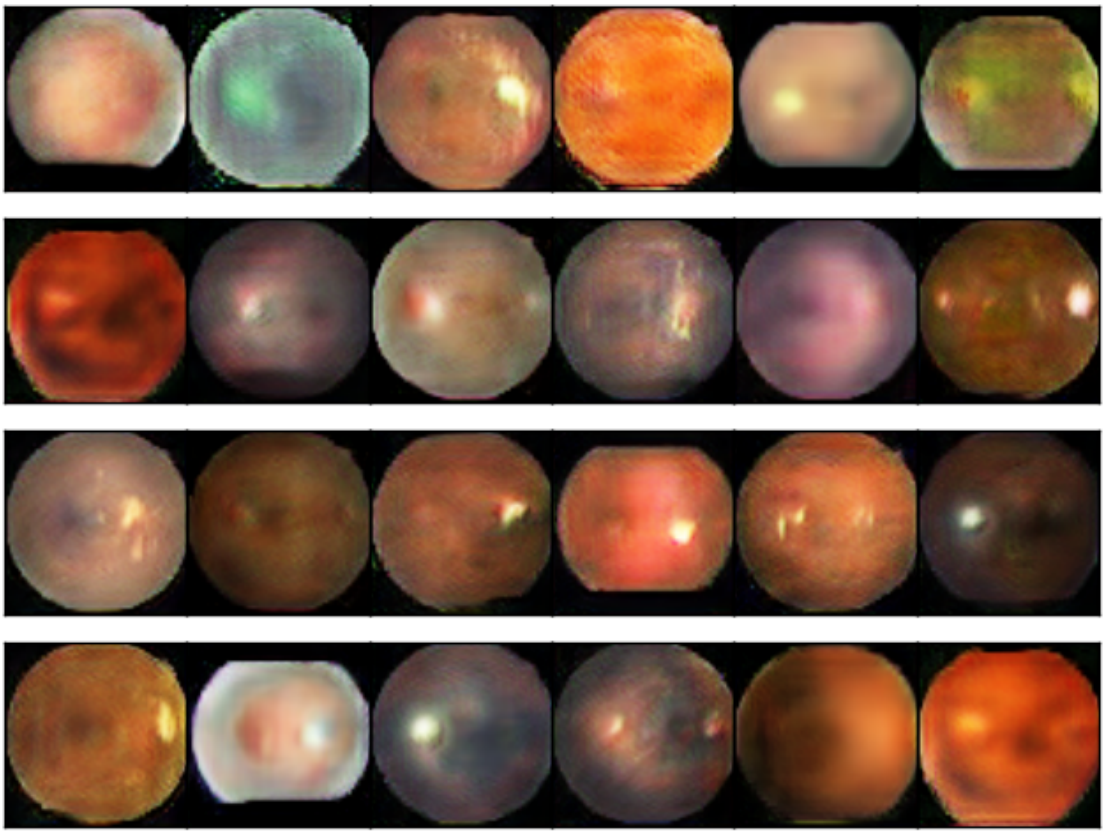
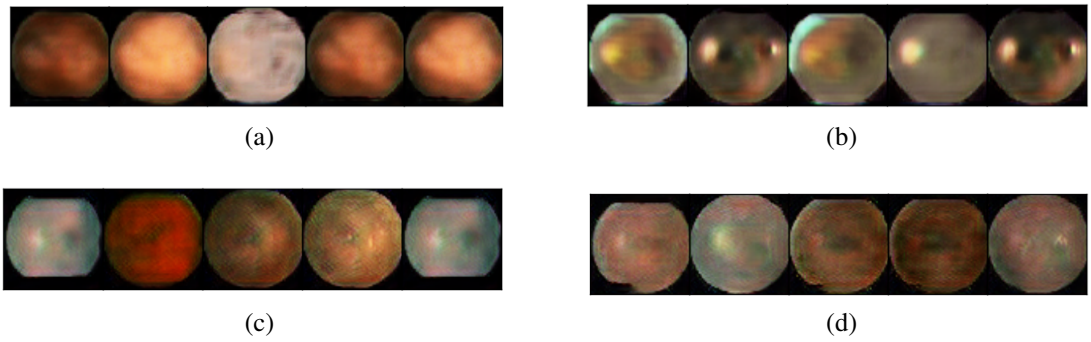


Figure 25. Randomly chosen samples of the generated retinal images by the proposed GAN.

Furthermore, we address following issues and solutions which are quite important to stabilize the training of the proposed GAN model.

- In the literature of GANs studies, it is commonly noted that GANs suffer from the model collapse that causes GANs to generate either unrelated samples with respect

to the data used for training or to generate only a few distinctive data samples. The first condition refers to the complete model collapse and the second condition refers to the partial model collapse [43]. In our case, we faced mainly the partial model collapse in which only a few distinctive retinal images are generated as given in Figure 26. In such cases (in Figure 26 from (a) to (d)), the network has generated the retinal images only a few different colors. To solve this issue, we introduced ReLU both after Batch Normalization and the convolution layers. The previous GANs architectures have been applied ReLU only after the convolution layers.



(a) (b) (c) (d)

Figure 26. Generated retinal images from the collapsed models from (a) to (d).

- An important observation to state here is that in the earlier studies [69, 43] it has been claimed that the small minibatch size can cause the fluctuations during each epoch of the training procedure. However, once we applied the ReLU as mentioned above, the proposed GAN model is able to overcome this issue as well.
- Similar to previous studies in [41, 69, 43], it takes quite a long time for GANs to converge. Although Batch Normalization helps a lot to decrease this time, we also discovered that if the data from the actual retinal set (EyePACS) are normalized before feeding into the discriminator, it contributes the overall convergence speed of the GANs model.

4.4 Retinal Image Generation via Variational Autoencoders

The generated retinal images via VAEs are demonstrated in this section by providing both issues and solutions regarding the training procedure of VAEs. As in the case with GANs, we start by giving details about the hyperparameters of VAEs. Unlike GANs, VAEs have a relatively small number of hyperparameters, which include the number epochs to train

the network, the size of intermediate dimension, batch size and the parameters of the RMSprop optimizer [74] as shown in Table 2. The loss function value during the training procedure for each epoch also is presented in Figure 27.

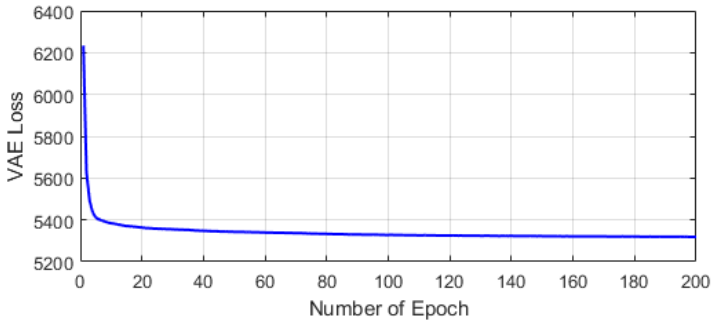


Figure 27. Training loss of VAEs for each epoch during the training.

Table 2. Hyperparameters associated with the training procedure of VAEs: (1) the parameters of RMSprop optimizer, which are learning rate lr, coefficients β_1, β_2 and ϵ; 2) the dimension of the hidden layer *mid-dim-size*, the dimension of the latent space *latent-dim-size*, batch size and number of epochs to train the network.

RMSprop Optimizer						
lr	β	ϵ	mid-dim-size	latent-dim-size	batch size	epochs
0.001	0.9	1	512	2	100	200

The generated retinal images with VAEs can be seen in Figure 28. To examine the results qualitatively, we can make following interpretations about the generated retinal images:

- As in the case of retinal images generated by GANs, the VAEs also generate the retinal images by preserving the global structure of the retina that contains an optic disk, fovea, macula and the overall shape of the retina.
- Yet, the VAEs are also unable to generate the vessel trees as a local structure of the retina.
- Furthermore, one can see from the generated images in Figure 28 that the VAEs mostly generate the retinal images with the dominant red color channel.
- Above all, the generated retinal images look blurry and this is because of the mean squared error which measures the pixel-to-pixel distance basically.

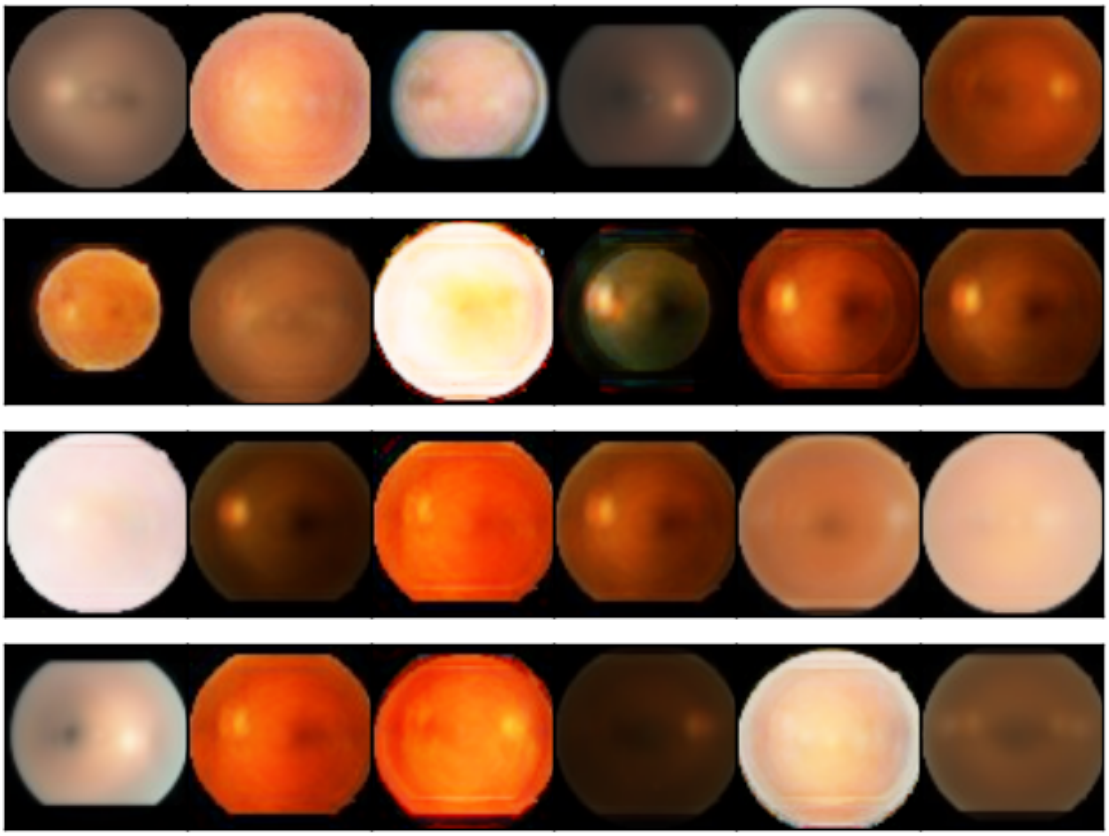


Figure 28. Randomly chosen samples of the generated retinal images by the proposed VAEs.

The VAEs are easy to train and, unlike the GANs, they usually converge without any model collapse. Therefore, we did not encountered any problems during the training process and just modified the hyperparameters of the MLP structure proposed in [42].

4.5 Quantitative Analysis of Generated Retinal Images

When it comes to the quality assessment of the generated images, there is an ambiguity regarding how to quantitatively evaluate the quality of data generated via deep generative models [75]. In order to assess the quality of the generated retinal images, we proposed relatively simple and different approach as explained in Section 3.4.

First, the statistical features (once again mean, variance, skewness, kurtosis, and entropy) are extracted from the benchmark DiaRetDB1, the generated retinal images via VAEs and GANs, and the subset of randomly selected retinal images from the EyePACS set. We have selected five different subsets randomly from EyePACS set. Each set contains 89 retinal images in total. Once the features are extracted, the next step is to compute the

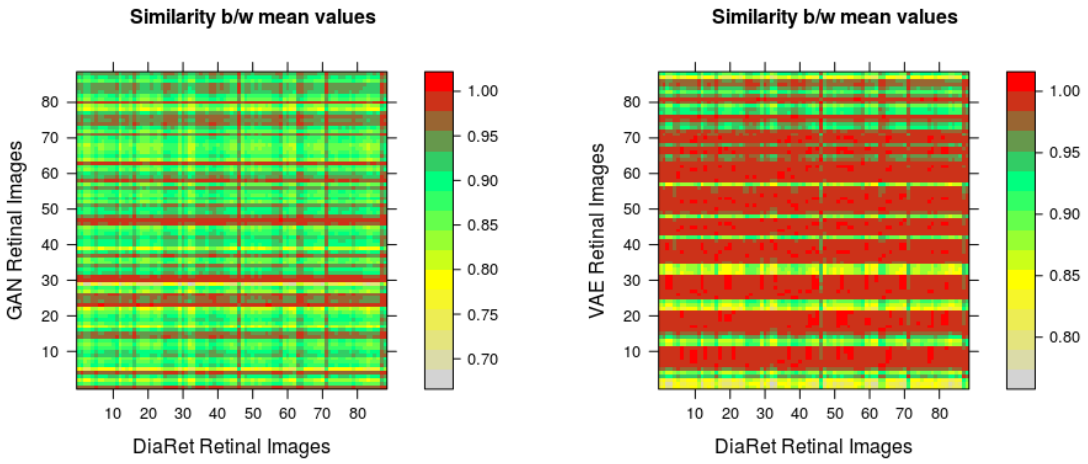
cosine similarity between the features of:

- DiaRetDB1 and the retinal images generated by GANs.
- DiaRetDB1 and the retinal images generated by VAEs.
- DiaRetDB1 and the subset of retinal images from the EyePACS set.

The cosine similarity between EyePACS subsets and DiaRetDB1 is given in Appendix 1. The analysis of each statistical feature of the retinal images generated by the GANs and VAEs is given as follows:

(i) **Mean**

Figure 29 illustrates the mean similarity matrix between (1) DiaRetDB1 and the images generated by the GANs, and (2) DiaRetDB1 and the images generated by the VAEs. As one can see from the figure, the similarity between the retinal images generated by VAEs and DiaRetDB1 is higher than the similarity between the retinal images generated by GANs and DiaRetDB1. It can be explained by the varying colors of the retinal images generated by the GANs, which is opposite of the retinal images in DiaRetDB1 as it contains mostly the retinal images with the dominant red channel.

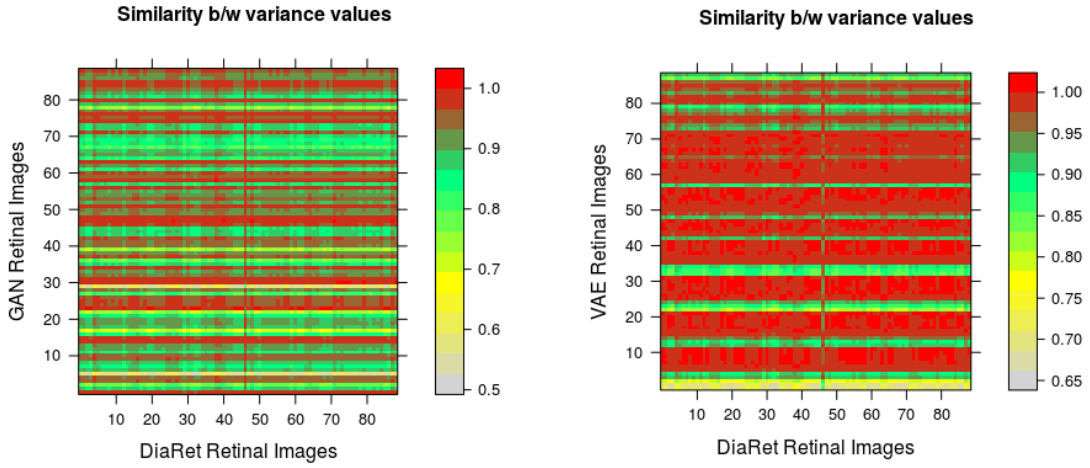


(a) (b)

Figure 29. Comparison of the generated retinal images: (a) Mean similarity between the images generated by the GANs and DiaRetDB1; (b) Mean similarity between the images generated by the VAEs and DiaRetDB1.

(ii) **Variance**

Figure 30 demonstrates the variance similarity matrix between (1) DiaRetDB1 and the images generated by the GANs, and (2) DiaRetDB1 and the images generated by the VAEs. The figure indicates that as the VAEs mainly generate only the retinal images with the dominant red channel, which is close to the retinal images in DiaRetDB1. Therefore, the similarity value of the retinal images generated by the VAEs is higher than the generated retinal images by the GANs.



(a) (b)

Figure 30. Comparison of the generated retinal images: (a) Variance similarity between the images generated by the GANs and DiaRetDB1; (b) Variance similarity between the images generated by the VAEs and DiaRetDB1.

(iii) **Skewness**

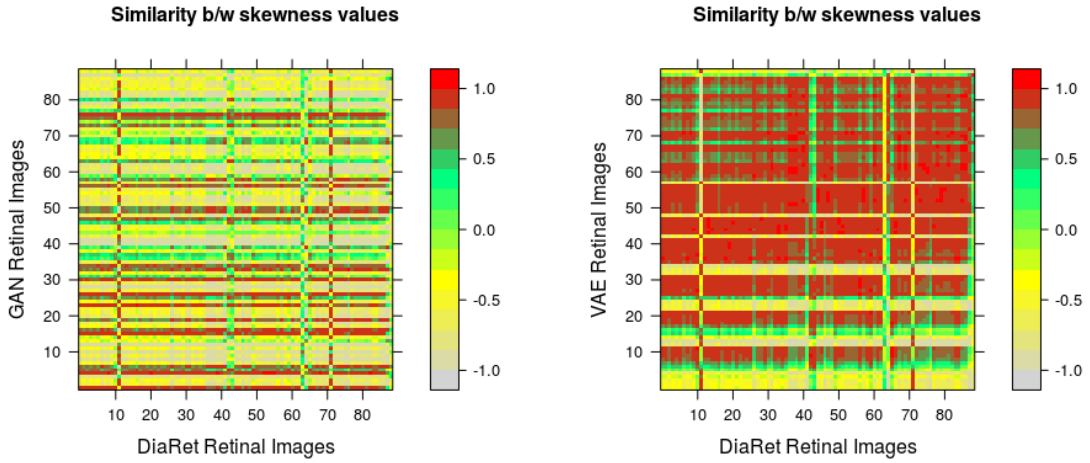
The skewness similarity is shown in Figure 31. The results are similar to the above-mentioned mean and variance. The same reasons stated for the mean and variance also apply here.

(iv) **Kurtosis**

The kurtosis similarity is given in Figure 32. The results are similar to the skewness similarity values and they share the same reason stated above.

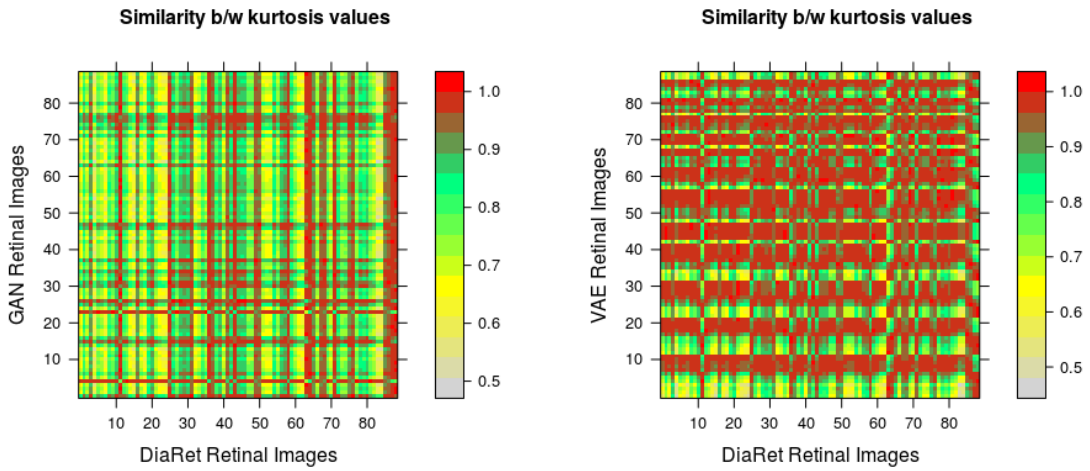
(v) **Entropy**

For the entropy, the results are quite different from the aforementioned cases as seen in Figure 33. It can be explained by the fact that the pixel values in each individual image of each set have low entropy because of the uniformity in the colors. This



(a) (b)

Figure 31. Comparison of the generated retinal images: (a) Skewness similarity between the images generated by the GANs and DiaRetDB1; (b) Skewness similarity between the images generated by the VAEs and DiaRetDB1.

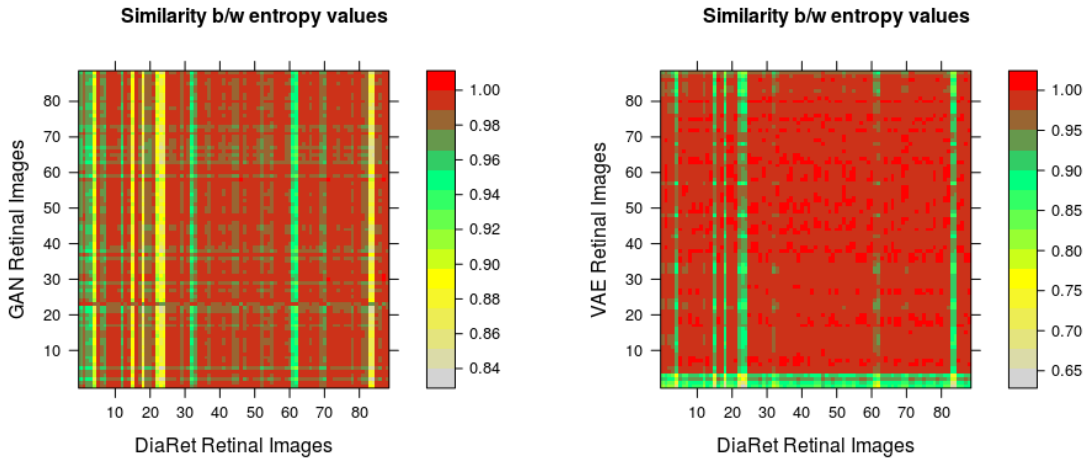


(a) (b)

Figure 32. Comparison of the generated retinal images: (a) Kurtosis similarity between the images generated by the GANs and DiaRetDB1; (b) Kurtosis similarity between the images generated by the VAEs and DiaRetDB1.

leads the entropy values between each data set to have similar values, thereby high similarity between entropy values.

In addition to these figures, the related figures for the other four subsets from the EyePACS are shown in Appendix 1.



(a) (b)

Figure 33. Comparison of the generated retinal images: (a) Entropy similarity between the images generated by GANs and DiaRetDB1; (b) Entropy similarity between the images generated by VAEs and DiaRetDB1.

Histogram Analysis of Statistical Features

The histogram of each statistical feature of each set is also demonstrated in Appendix 2.1 and Appendix 2.2. In image processing tasks, the histogram explains the frequency distribution of the intensity values of an image [76]. In this thesis, the histogram analysis is conducted to reveal how the values of the statistical features are distributed across the data sets including the EyePACS, DiaRetDB1, and the generated retinal images by the GANs and VAEs. To make the interpretations of the results more clear, the set of statistical features are divided into 32 bins and then smoothing is applied. The analysis is performed for each channel (RGB).

The conducted analysis revealed that particularly the statistical features extracted from red channel of the generated retinal images and DiaRetDB1 have similar frequencies. However, the green and the blue channel do not have the same frequency patterns as seen in Appendix 2.1. On the other hand, the statistical features extracted from the subset of retinal images from EyePACS set have and DiaRetDB1 have more similar frequency patterns for each channel as given in Appendix 2.2. This is actually an important implication to distinguish the generated retinal images from the actual ones even by looking at the histogram analysis of the statistical features.

5 DISCUSSION

The retinal imaging techniques can be categorized into two main categories, which are: *digital fundus imaging* and *OCT*. In addition to these techniques, the spectral retinal imaging also is used for gathering the rich amount of information about the retina. Although the fundus imaging is widely applied in the field, OCT has stated a more advanced technique because of the cross-sectional visualization of the retina captured by OCT. A common issue in both of these techniques is that they suffer from the image acquisition time. This problem can be solved by recently proposed snapshot retinal imaging technique.

As the availability of the synthetic retinal images is an important issue in the research area of the retinal imaging based applications for further developments and validations of the algorithms/methods, more attention should be paid to solve this problem. In order to that, one can think of studying deep learning based methods (in particular deep generative models).

As a utilization of deep generative models, *Generative Adversarial Networks (GANs)* and *Variational Autoencoders (VAEs)* were chosen to generate synthetic retinal images in the scope of this thesis. During the training process of both the GANs and VAEs, we explored the following outcomes for retinal image generation:

- The overall structure of the retina is successfully generated by applying the GANs and VAEs. However, both of them are not able to model the vessel tree structure clearly.
- Another key point to state for the GANs and VAEs is that GANs generate the retinal images with distinctive colors as in the training set. On the other hand, the VAEs capture only the dominant red channel. This is because of the constraints on the VAEs in which the data are reinforced to be generated from the Gaussian distribution.
- In the context of the qualitative analysis of the generated retinal images, the GANs generate sharp retinal images while the VAEs generate blurry images because of the mean squared error used to compute the pixel-to-pixel distance between the generated and the actual retinal images.

The quantitative analysis of the generated images by proposed similarity based quality assessment method reveals following outcomes:

- As the GANs and VAEs generate retinal images with a global structure, this led us to choose an evaluation method which is based on the global statistical features including the mean, variance, kurtosis, skewness, and entropy.
- The similarity between the generated retinal images by the VAEs and the benchmark data set is higher than the generated retinal images by the GANs and randomly chosen subsets from EyePACS set. This is because of that the VAEs often generate the retinal images with the dominant red channel.
- The histogram analysis is important to see how the statistical features of the generated retinal images are distributed by comparing to the real retinal data. For the channel-wise comparison, the analysis reveals that the generated retinal images with the red channels are closer to the real samples than the generated retinal images with the green and blue channels. This can be seen as a drawback of the deep generative models which tend to learn the dominant features of the real data.

From the development perspective, we can state the following observations while training the GANs and VAEs:

- It is recommended to use Keras with Theano because of not to have time-consuming configuration issues related to Tensorflow.
- Batch Normalization and ReLU are efficient in accelerating the training process of the GANs and avoiding from possible model collapse.

5.1 Future Work

As the proposed models lack generating the local structure of the retinal image, one can focus on to generate synthetic vascular structure and combine it with our findings. Moreover, the recent studies [60, 61, 62] have shown the potential of combining GANs and VAEs together to synthesize images. Therefore, the retinal image synthesis, in the same way, can be studied to see whether the combination of GANs and VAEs is capable of generating vascular tree structure in an unconditioned way. Also, GANS and VAEs can be applied to retinal image generation tasks by conditioning the models on some specific features of retinal images like intensity field, the label of retinal image that indicates the type of disease manifest in the retina. In this way, the problem based retinal images can be synthesized.

6 CONCLUSION

This thesis investigates the utilization of deep generative models for the retinal image generation in an unconditioned way. In Chapter 2, the anatomy of the eye and the retinal imaging techniques were reviewed. In addition to that, the detailed literature review of existing solutions and proposed methods for generation and reconstruction of the retina were given for both retinal fundus images and spectral retinal images.

In Chapter 3, first the general terms, which are needed to understand the deep generative models, were explained by the examples from real life. Afterward, Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) were presented in details by reviewing the studies of the GANs and VAEs in the literature. Finally, the similarity based method for the quality assessment of the retinal images generated by the GANs and VAEs was introduced.

In Chapter 4, the experimental analysis conducted to generate the synthetic retinal images was demonstrated with the data sets used in the experiments. Also, the details of the architectural design of the GANs and VAEs were given. Additionally, in Chapter 4, the quality assessment of the generated retinal images was carried by applying the similarity based quality assessment method. The findings and the observations during the experimental analysis process were discussed in Chapter 5 with possible research frontiers in the field.

In the experimental analysis, we showed that the overall architecture of the retina can be synthesized by applying deep generative models in an unconditioned way (from only noise). However, the studied deep generative models were unable to generate the vascular tree structure of the retina. The proposed quality assessment method revealed that the overall similarity between the retinal images generated by the VAEs and the benchmark set (DiaRetDB1) were higher than the similarity between the retinal images generated by GANs and the benchmark set (DiaRetDB1).

REFERENCES

[1] Denis Le Bihan. Looking into the functional architecture of the brain with diffusion mri. *Nature Reviews Neuroscience*, 4(6):469–480, 2003.

[2] Richard P Wildes. Iris recognition: an emerging biometric technology. *Proceedings of the IEEE*, 85(9):1348–1363, 1997.

[3] Lauri Laaksonen. *Spectral retinal image processing and analysis for ophthalmology*. PhD thesis, Lappeenranta University of Technology, 2016.

[4] Peter K. Kaiser Sophie J. Bakri. Diabetic retinopathy. In D. Huang, editor, *Retinal Imaging*, chapter 21, pages 233–240. Mosby Elsevier, 2006.

[5] Peter K. Rafael L., Leonid E. Non-neovascular age-related macular degeneration. In D. Huang, editor, *Retinal Imaging*, chapter 11-12, pages 145–163. Mosby Elsevier, 2006.

[6] James M Tielsch, Joanne Katz, Kuldev Singh, Harry A Quigley, John D Gottsch, Jonathan Javitt, and Alfred Sommer. A population-based evaluation of glaucoma screening: the baltimore eye survey. *American Journal of Epidemiology*, 134(10):1102–1110, 1991.

[7] Tien Yin Wong, Anoop Shankar, Ronald Klein, Barbara EK Klein, and Larry D Hubbard. Prospective cohort study of retinal vessel diameters and risk of hypertension. *British Medical Journal*, 329(7457):79, 2004.

[8] Timothy G. Audina M., Elias C. Retinoblastoma. In D. Huang, editor, *Retinal Imaging*, chapter 55, pages 471–479. Mosby Elsevier, 2006.

[9] Haroldo Vieira. Tuberculosis. In D. Huang, editor, *Retinal Imaging*, chapter 39, pages 354–358. Mosby Elsevier, 2006.

[10] D. Huang, editor. *Retinal Imaging*. Mosby Elsevier, 2006.

[11] Pearse A Keane and Srinivas R Sadda. Retinal imaging in the twenty-first century: state of the art and future directions. *Ophthalmology*, 121(12):2489–2500, 2014.

[12] EYK Ng, Jen Hong Tan, U Rajendra Acharya, and Jasjit S Suri. *Human Eye Imaging and Modeling*. CRC Press, 2012.

[13] Helga Kolb. How the retina works. *American Scientist*, 91(1):28–35, 2003.

[14] Helga Kolb, Eduardo Fernandez, Ralph Nelson, and BW Jones. Webvision: Organization of the retina and visual system. 2005. *John Moran Eye Center, University of Utah, USA*.

[15] Michael D Abràmoff, Mona K Garvin, and Milan Sonka. Retinal imaging and image analysis. *IEEE Reviews in Biomedical Engineering*, 3:169–208, 2010.

[16] C Richard Keeler. 150 years since babbage's ophthalmoscope. *Archives of ophthalmology*, 115(11):1456–1457, 1997.

[17] Adrien Christophe van Trigt. *Dissertatio ophthalmologica inauguralis De speculo oculi ejusque usu observation: comprobato*, volume 1. PW van de Weijer, 1853.

[18] Lee Allen. Ocular fundus photography*: Suggestions for achieving consistently good pictures and instructions for stereoscopic photography. *American journal of ophthalmology*, 57(1):13–28, 1964.

[19] Wolfgang Derexler Johannes F. Boer Maciej Wojtkowski David Huang, James G. Fujimoto and Andrzej Kowalczyk. Optical coherence tomography. In D. Huang, editor, *Retinal Imaging*, chapter 3, pages 47–65. Mosby Elsevier, 2006.

[20] Jeffrey L. Olson and Naresh Marandava. Flourescein angiography. In D. Huang, editor, *Retinal Imaging*, chapter 1, pages 3–21. Mosby Elsevier, 2006.

[21] Richard F Spaide, James M Klancnik, and Michael J Cooney. Retinal vascular layers imaged by fluorescein angiography and optical coherence tomography angiography. *JAMA Ophthalmology*, 133(1):45–50, 2015.

[22] Pauli Fält, Jouni Hiltunen, Markku Hauta-Kasari, Iiris Sorri, Valentina Kalesnykiene, and Hannu Uusitalo. Extending diabetic retinopathy imaging from color to spectra. In *Proceedings of Scandinavian Conference on Image Analysis*, pages 149–158. Springer, 2009.

[23] William R Johnson, Daniel W Wilson, Wolfgang Fink, Mark Humayun, and Greg Bearman. Snapshot hyperspectral imaging in ophthalmology. *Journal of Biomedical Optics*, 12(1):014036–014036, 2007.

[24] Liang Gao, R Theodore Smith, and Tomasz S Tkaczyk. Snapshot hyperspectral retinal camera with the image mapping spectrometer (ims). *Biomedical Optics Express*, 3(1):48–54, 2012.

[25] Joel Kaluzny, Hao Li, Wenzhong Liu, Peter Nesper, Justin Park, Hao F Zhang, and Amani A Fawzi. Bayer filter snapshot hyperspectral fundus camera for human retinal imaging. *Current Eye Research*, pages 1–7, 2016.

[26] Mark A Sagar, David Bullivant, Gordon D Mallinson, and Peter J Hunter. A virtual environment and model of the eye for surgical simulation. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 205–212. ACM, 1994.

[27] Samuele Fiorini, Lucia Ballerini, Emanuele Trucco, and Alfredo Ruggeri. Automatic generation of synthetic retinal fundus images. In *Eurographics Italian Chapter Conference*, pages 41–44, 2014.

[28] Jan Odstrcilik, Radim Kolar, Attila Budai, Joachim Hornegger, Jiri Jan, Jiri Gazarek, Tomas Kubena, Pavel Cernosek, Ondrej Svoboda, and Elli Angelopoulou. Retinal vessel segmentation by improved matched filtering: evaluation on a new high-resolution fundus image database. *IET Image Processing*, 7(4):373–383, 2013.

[29] Lorenza Bonaldi, Elisa Menti, Lucia Ballerini, Alfredo Ruggeri, and Emanuele Trucco. Automatic generation of synthetic retinal fundus images: Vascular network. *Procedia Computer Science*, 90:54–60, 2016.

[30] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[31] Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[32] Pedro Guimarães, Pedro Rodrigues, Dirce Celorico, Pedro Serranho, and Rui Bernardes. Three-dimensional segmentation and reconstruction of the retinal vasculature from spectral-domain optical coherence tomography. *Journal of Biomedical Optics*, 20(1):016006–016006, 2015.

[33] Pedro Guimarães, Pedro Rodrigues, Conceição Lobo, Sérgio Leal, João Figueira, Pedro Serranho, and Rui Bernardes. Ocular fundus reference images from optical coherence tomography. *Computerized Medical Imaging and Graphics*, 38(5):381–389, 2014.

[34] Bin Fang, Wynne Hsu, and Mong Li Lee. Reconstruction of vascular structures in retinal images. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II–157. IEEE, 2003.

[35] Li Tang, Young H Kwon, Wallace LM Alward, Emily C Greenlee, Kyungmoo Lee, Mona K Garvin, and Michael D Abràmoff. 3d reconstruction of the optic nerve head using stereo fundus images for computer-aided diagnosis of glaucoma. In

SPIE Medical Imaging, pages 76243D–76243D. International Society for Optics and Photonics, 2010.

[36] Uyen Nguyen, Lauri Laaksonen, Hannu Uusitalo, and Lasse Lensu. Reconstruction of retinal spectra from RGB data using a RBF network. In *Image Processing Theory Tools and Applications (IPTA), 2016 6th International Conference on*, pages 1–6. IEEE, 2016.

[37] Andrew Y Ng and Michael I Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. *Advances in neural information processing systems*, 2:841–848, 2002.

[38] Guillaume Bouchard and Bill Triggs. The tradeoff between generative and discriminative classifiers. In *16th IASC International Symposium on Computational Statistics (COMPSTAT'04)*, pages 721–728, 2004.

[39] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[40] Li Deng, Dong Yu, et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

[41] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[42] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[43] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[44] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems*, pages 1486–1494, 2015.

[45] Peter Burt and Edward Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.

[46] Jon Gauthier. Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester*, 2014:5, 2014.

[47] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[48] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. *ArXiv e-prints*, June 2016.

[49] Weidong Yin, Yanwei Fu, Leonid Sigal, and Xiangyang Xue. Semi-latent GAN: Learning to generate and modify facial images from attributes. *arXiv preprint arXiv:1704.02166*, 2017.

[50] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[51] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1612.03242*, 2016.

[52] Kevin Schawinski, Ce Zhang, Hantian Zhang, Lucas Fowler, and Gokula Krishnan Santhanam. Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit. *arXiv preprint arXiv:1702.00403*, 2017.

[53] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[54] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098, 2002.

[55] Gerben Van Den Broeke. What auto-encoders could learn from brains - generation as feedback in deep unsupervised learning and inference. Master thesis, 2016-01-18.

[56] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[57] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pages 2539–2547, 2015.

[58] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. In *Advances in Neural Information Processing Systems*, pages 2352–2360, 2016.

[59] Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. A hybrid convolutional variational autoencoder for text generation. *arXiv preprint arXiv:1702.02390*, 2017.

[60] Mahesh Gorijala and Ambedkar Dukkipati. Image generation and editing with variational info generative adversarialnetworks. *arXiv preprint arXiv:1701.04568*, 2017.

[61] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete electronic health records using generative adversarial networks. *arXiv preprint arXiv:1703.06490*, 2017.

[62] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[63] Mahnaz Fasih. Retinal image quality assessment using supervised classification. Master thesis, École Polytechnique de Montréal, 2014.

[64] Tomi Kauppi, Valentina Kalesnykiene, Joni-Kristian Kamarainen, L Lensu, I Sorri, J Pietila, H Kalviainen, and H Uusitalo. DiaRetDB1-standard diabetic retinopathy database. *IMAGERET Optimal Detection and Decision-Support Diagnosis of Diabetic Retinopathy*, 2007.

[65] Vijay Kumar and Priyanka Gupta. Importance of statistical measures in digital image processing. *International Journal of Emerging Technology and Advanced Engineering*, 2(8):56–62, 2012.

[66] Herbert Davis, Stephen Russell, Eduardo Barriga, Michael Abramoff, and Peter Soliz. Vision-based, real-time retinal image quality assessment. In *Computer-Based Medical Systems, 2009. CBMS 2009. 22nd IEEE International Symposium on*, pages 1–6. IEEE, 2009.

[67] François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[68] Kaggle. Diabetic retinopathy detection. `https://www.kaggle.com/c/diabetic-retinopathy-detection`, 2015 (accessed May 17, 2017).

[69] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[70] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[71] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*, 2016.

[72] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[73] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

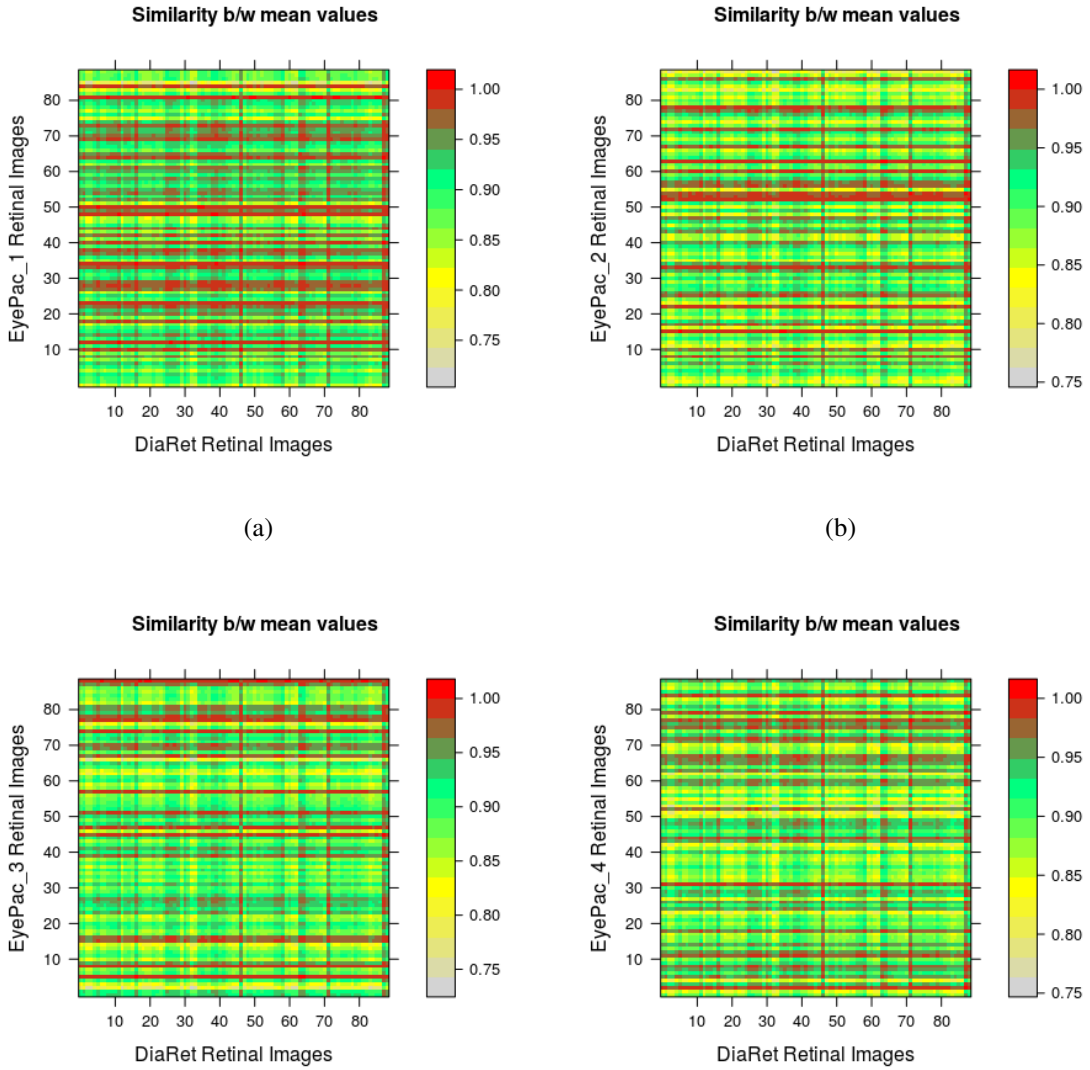
[74] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[75] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

[76] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*, chapter 3, pages 120–144. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

Appendix 1. EyePACS Similarity Evaluation

1.1 Mean Assessment



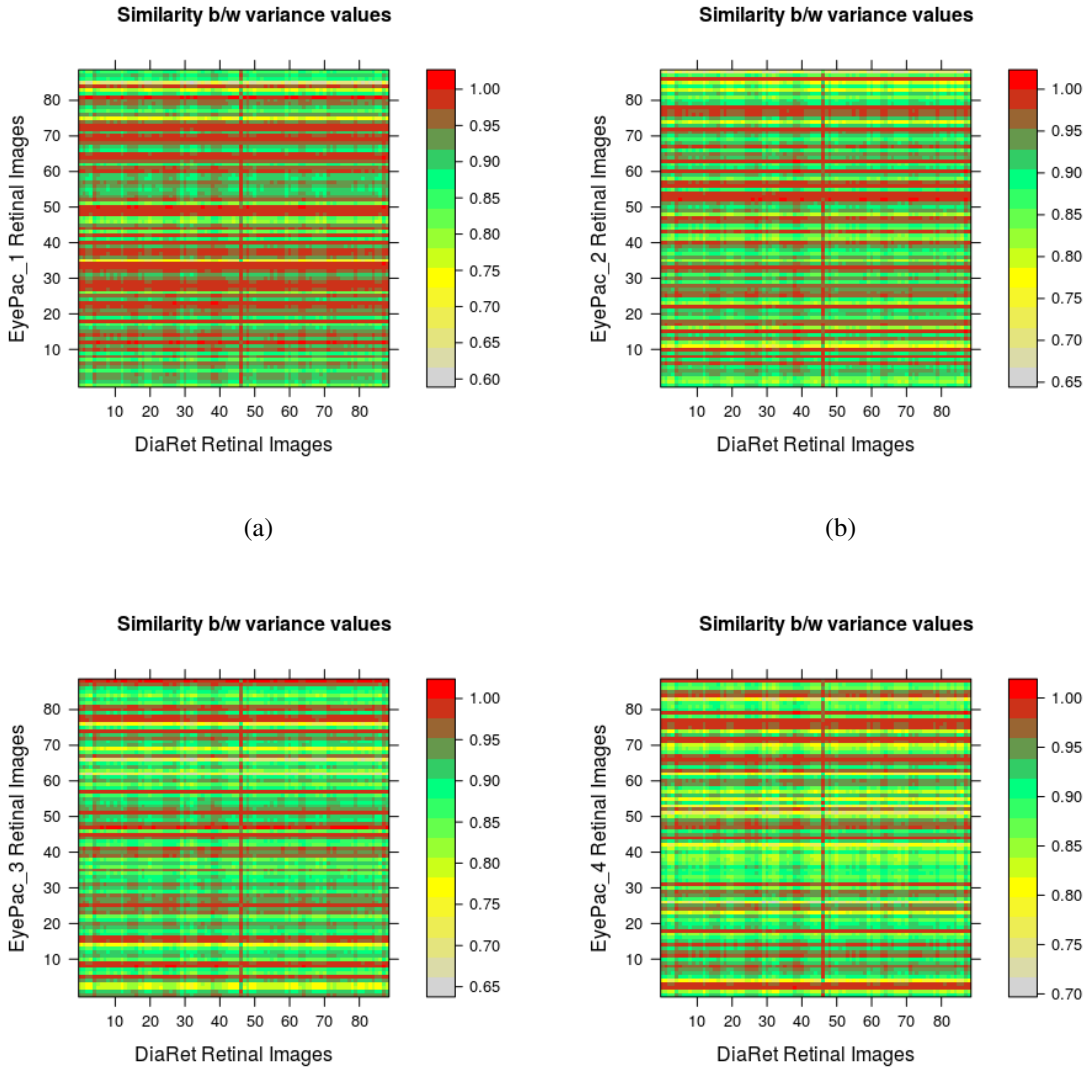
(a) (b)

(c) (d)

Figure A1.1. Comparison of subset of retinal images from EyePACS: (a) Mean similarity between first subset of EyePACS and DiaRetDB1; (b) Mean similarity between second subset of EyePACS and DiaRetDB1; (c) Mean similarity between third subset of EyePACS and DiaRetDB1; (d) Mean similarity between fourth subset of EyePACS and DiaRetDB1.

Appendix 1. EyePACS Similarity Evaluation

1.2 Variance Assessment



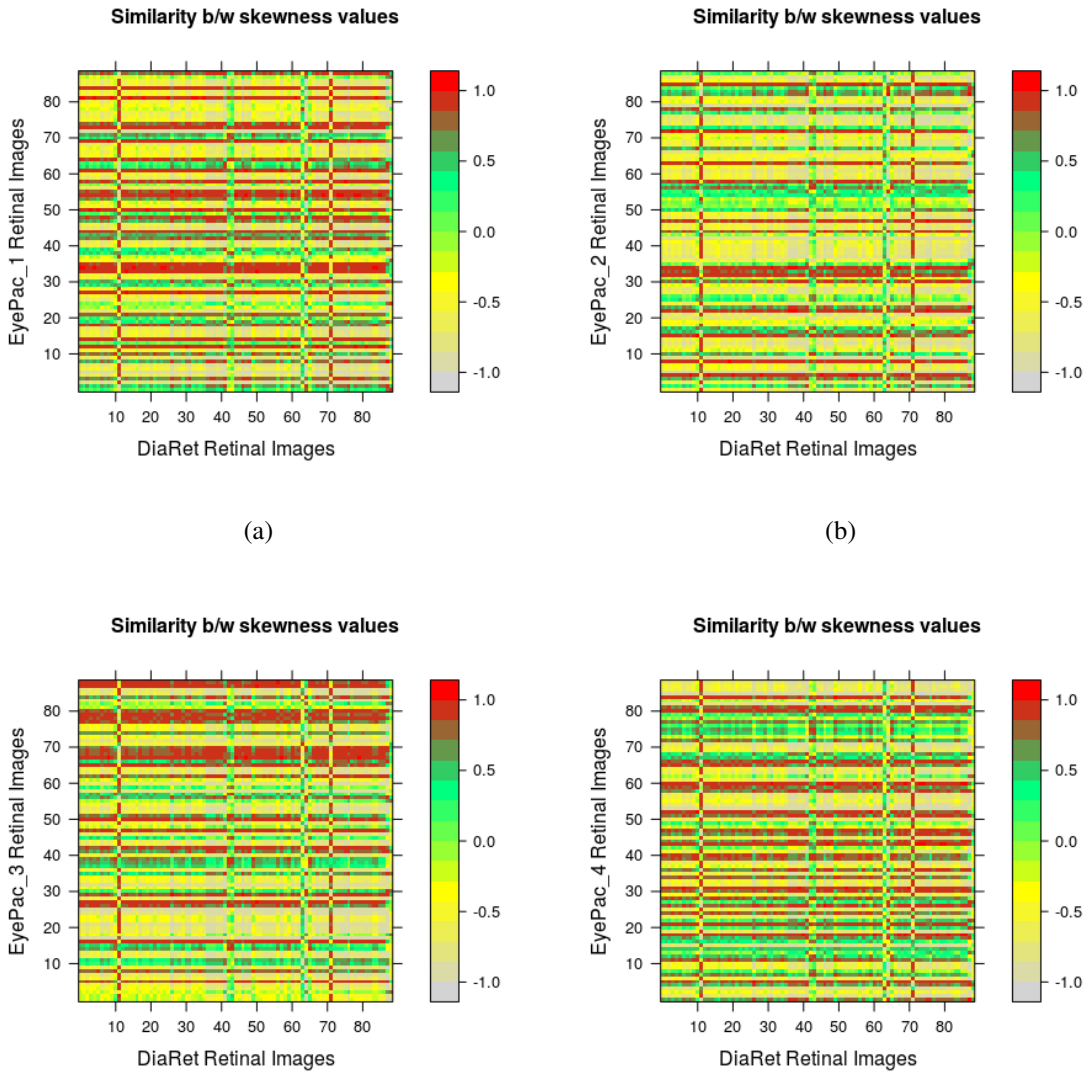
(a) (b)

(c) (d)

Figure A1.2. Comparison of subset of retinal images from EyePACS: (a) Variance similarity between first subset of EyePACS and DiaRetDB1; (b) Variance similarity between second subset of EyePACS and DiaRetDB1; (c) Variance similarity between third subset of EyePACS and DiaRetDB1; (d) Variance similarity between fourth subset of EyePACS and DiaRetDB1.

Appendix 1. EyePACS Similarity Evaluation

1.3 Skewness Assessment



(a) (b)

(c) (d)

Figure A1.3. Comparison of subset of retinal images from EyePACS: (a) Skewness similarity between first subset of EyePACS and DiaRetDB1; (b) Skewness similarity between second subset of EyePACS and DiaRetDB1; (c) Skewness similarity between third subset of EyePACS and DiaRetDB1; (d) Skewness similarity between fourth subset of EyePACS and DiaRetDB1.

Appendix 1. EyePACS Similarity Evaluation

1.4 Kurtosis Assessment

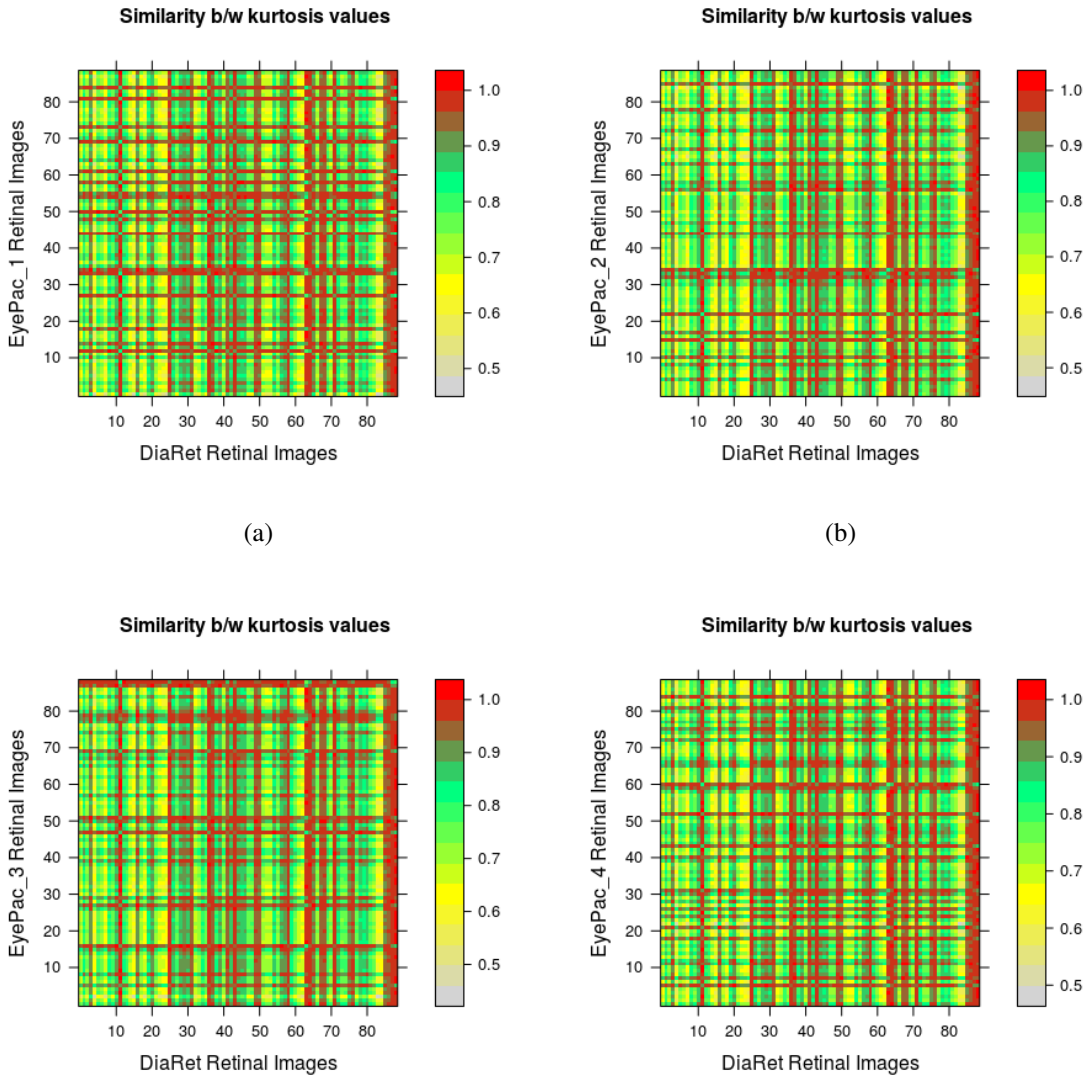
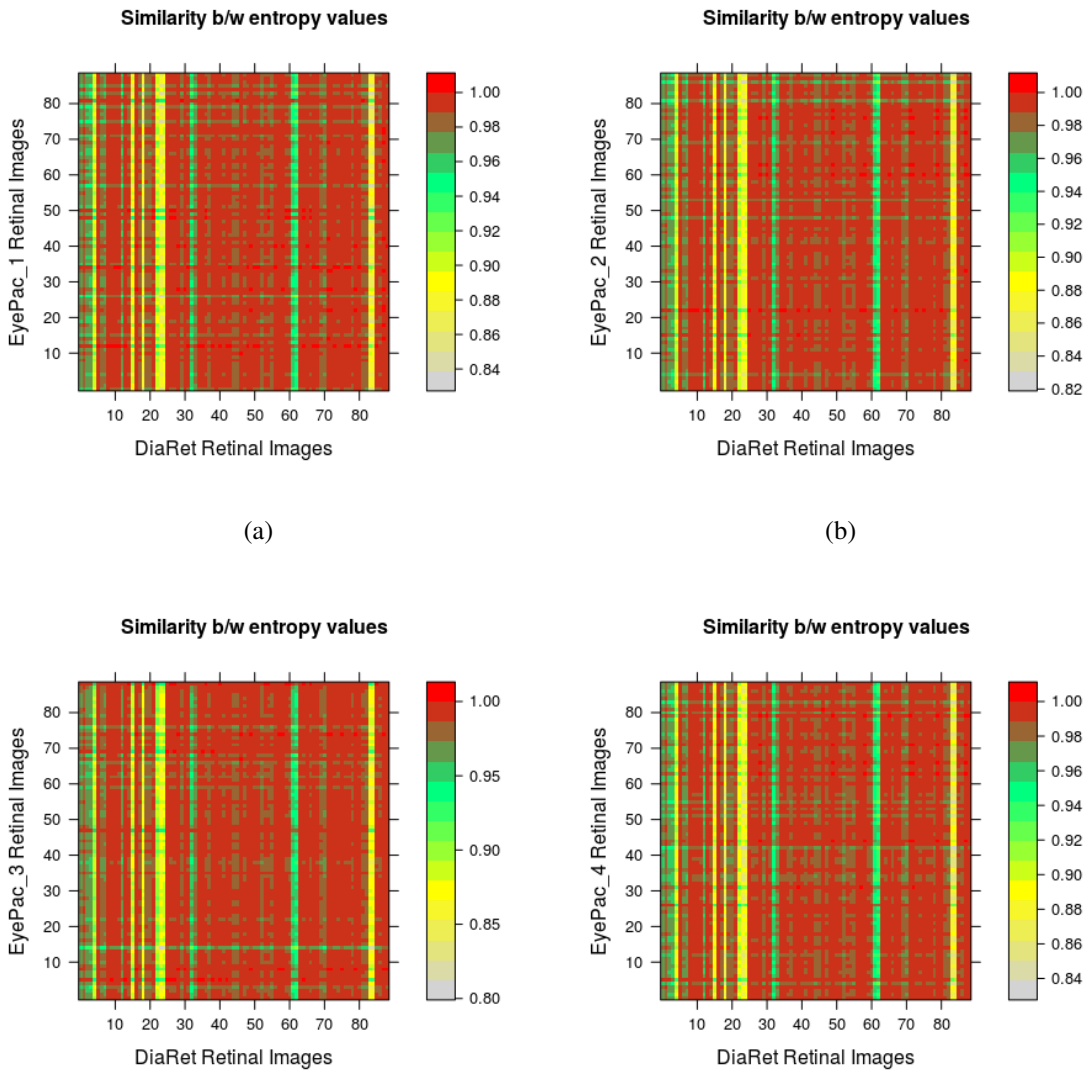


Figure A1.4. Comparison of subset of retinal images from EyePACS: (a) Kurtosis similarity between first subset of EyePACS and DiaRetDB1; (b) Kurtosis similarity between second subset of EyePACS and DiaRetDB1; (c) Kurtosis similarity between third subset of EyePACS and DiaRetDB1; (d) Kurtosis similarity between fourth subset of EyePACS and DiaRetDB1.

Appendix 1. EyePACS Similarity Evaluation

1.5 Entropy Assessment



(a) (b)

(c) (d)

Figure A1.5. Comparison of subset of retinal images from EyePACS: (a) Entropy similarity between first subset of EyePACS and DiaRetDB1; (b) Entropy similarity between second subset of EyePACS and DiaRetDB1; (c) Entropy similarity between third subset of EyePACS and DiaRetDB1; (d) Entropy similarity between fourth subset of EyePACS and DiaRetDB1.

2.1 Histogram Analysis of Generated Retinal Images

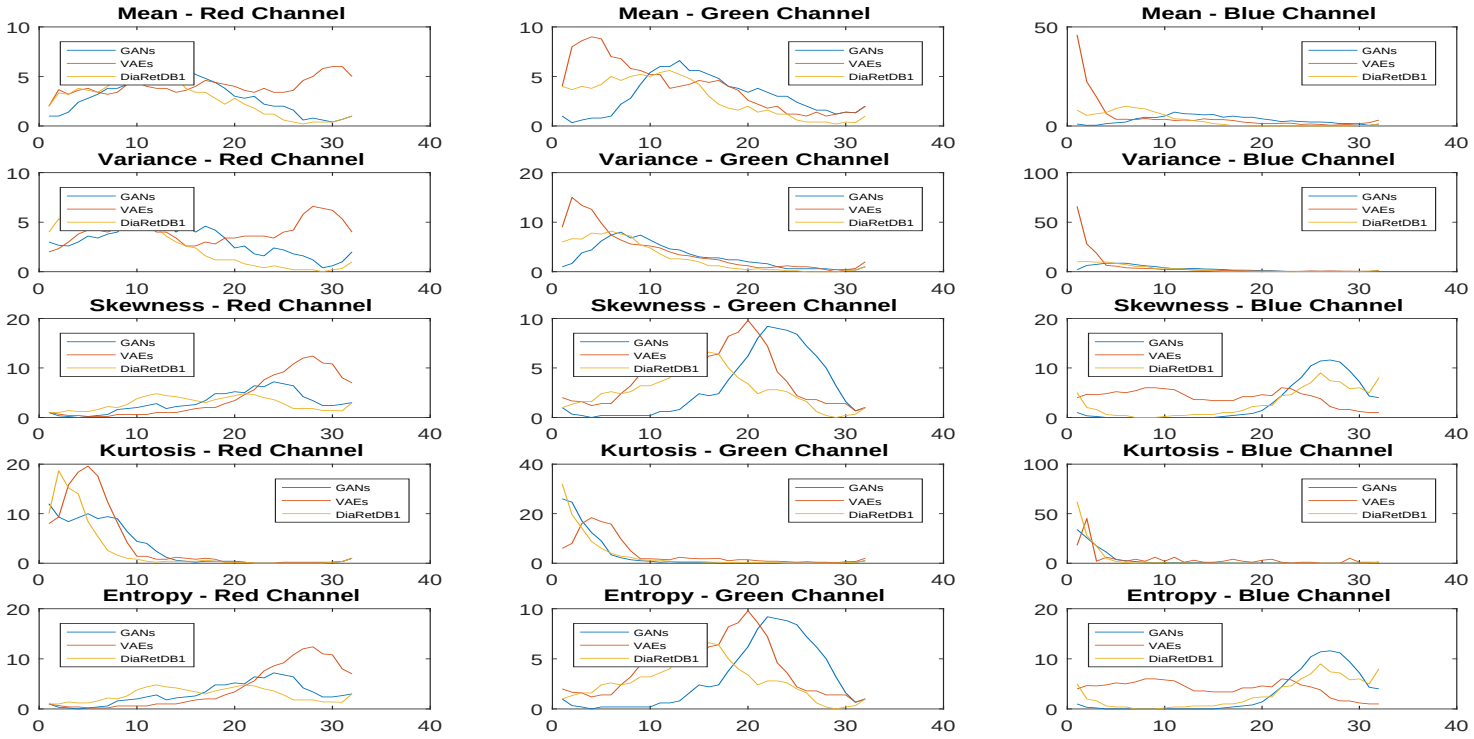


Figure A2.1. Histogram analysis of each statistical feature per color channel. The histogram is computed for each channel separately from the generated retinal images via the GANs, the generated retinal images via the VAEs and the reference set DiaRetDB1.

2.2 Histogram Analysis of EyePACS set

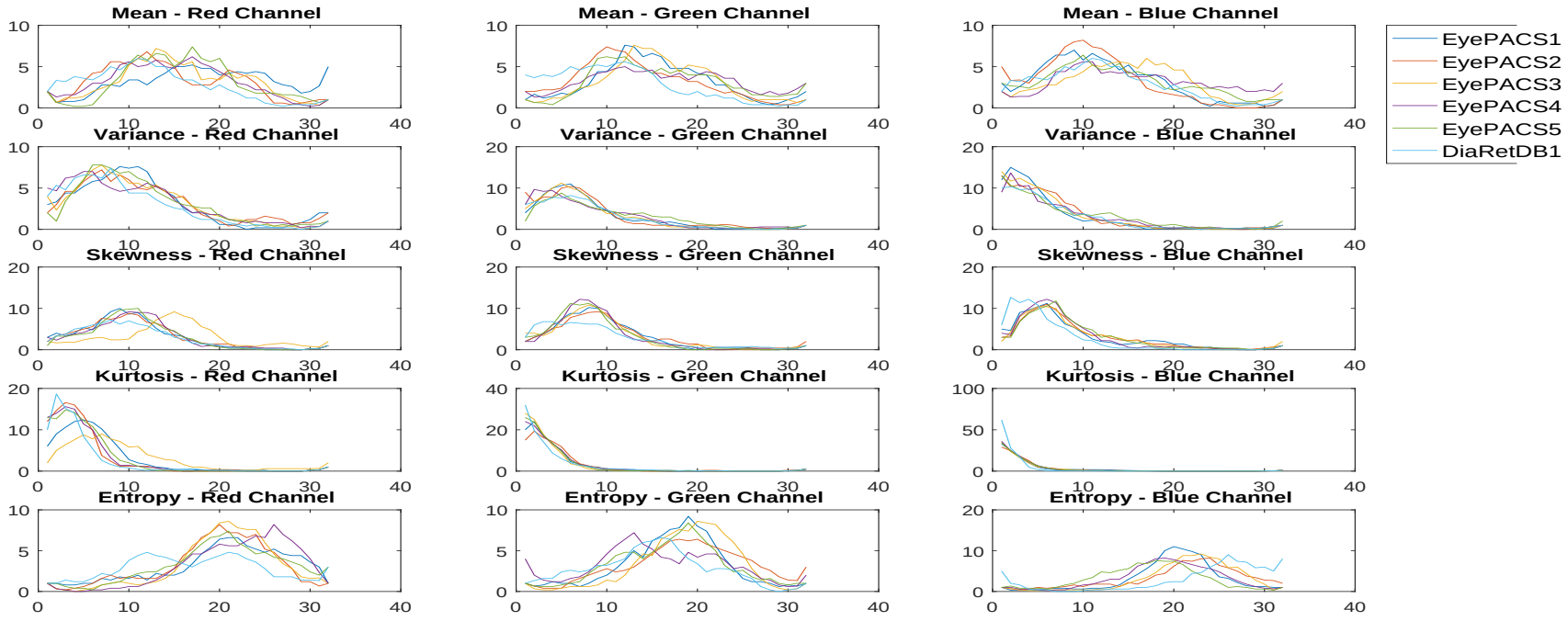


Figure A2.2. Histogram analysis of each statistical feature per color channel. The histogram is computed for each channel separately from subset of retinal images in EyePACS used for the puspose of quality assessment and the reference set DiaRetDB1.

Appendix 3. Generated Retinal Images with Generative Adversarial Networks

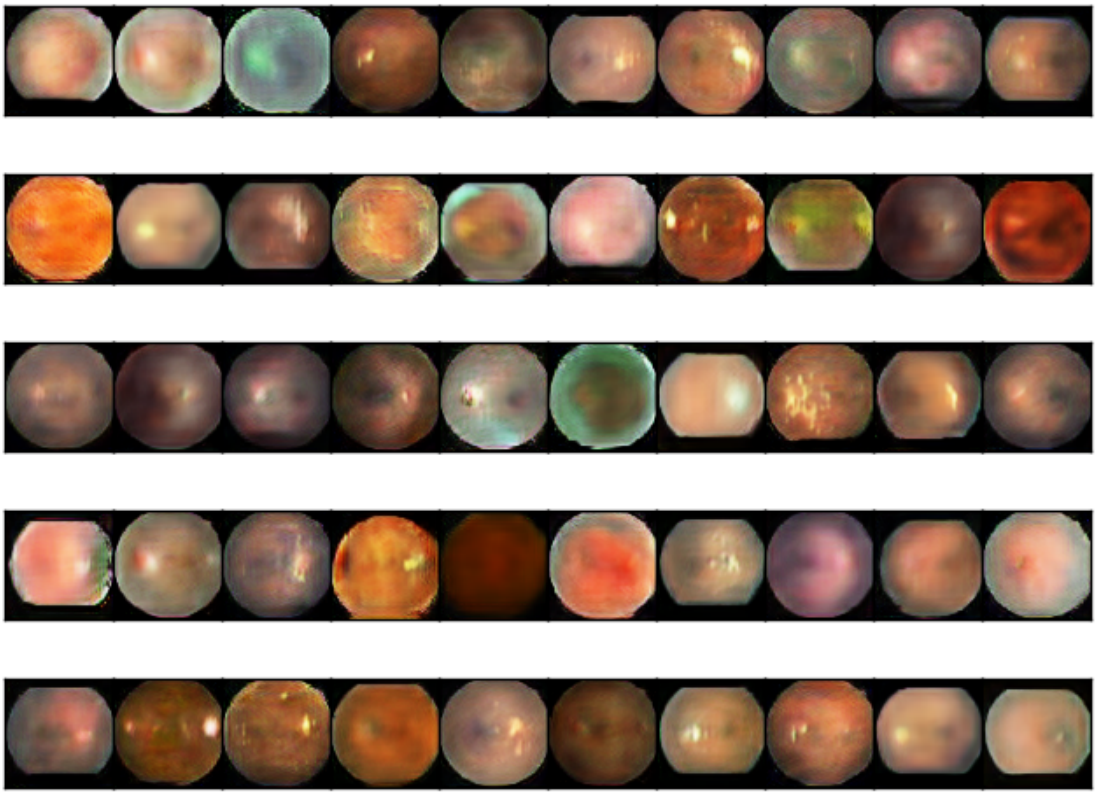


Figure A3.1. Examples of generated retinal images by the proposed GAN.

Appendix 4. Generated Retinal Images with Variational Autoencoders

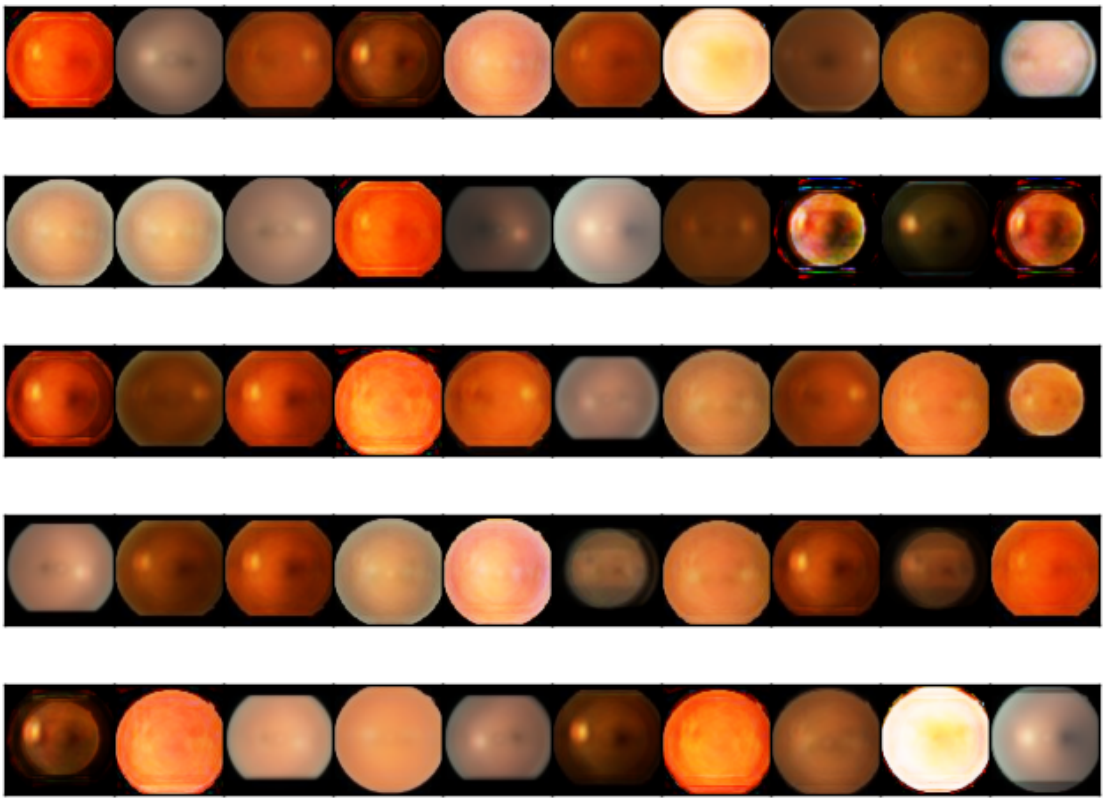


Figure A4.1. Examples of generated retinal images by the proposed VAEs.