In the Name of God

University of Tehran

Faculty of Electrical and Computer Engineering

Neural Networks and Deep Learning

Homework 6

Full Name	Mohammad Taha Majlesi
Student Number	810101504
Submission Deadline	2025/06/21

Contents

List of Figures

List of Tables

1 Question 2: Reconstruction of Endoscopy Polyp Images with EndoVAE

1.1 Introduction

In the last decade, there has been an increase in diseases related to the Gastrointestinal (GI) tract. The mouth, esophagus, stomach, and intestines are all part of the GI tract. Early diagnosis can prevent the progression of these diseases.

Wireless capsule endoscopy is one of the prominent methods for imaging the GI tract and diagnosing diseases. In this method, the patient swallows a capsule containing a camera, which records video from inside the body.

Diagnosing diseases from the captured images is a difficult task for physicians. Therefore, computational tools are used for diagnosis and treatment. With recent advances in deep learning, its use in the field of medical imaging is increasing.

One of the main problems in the analysis of endoscopy images is the lack of data, which, due to patient privacy laws, slows down and limits the creation and distribution of datasets in this field, even for practical research. Also, existing datasets often lack sufficient diversity and are not balanced.

To address these problems, various data augmentation methods have been proposed. Recent methods usually use GAN models, but we know that these models have various training problems such as non-convergence and training instability, and they require a large amount of diverse data for training. This paper has tried to alleviate these limitations to some extent with the help of Variational Autoencoders (VAEs).

1.2 Data Preprocessing

The paper uses the KID dataset, but we use the Kvasir dataset. This dataset was prepared by specialist physicians from 4 hospitals from 470,000 patients for research purposes. The images in this dataset are from different classes of diseases, different parts of the body, and even images after polyp removal, and there are hundreds of images for each class. The dimensions of the images vary from 720x576 pixels to 1920x1072 pixels.

We use three healthy classes that show different parts of a healthy person's body and one polyp class that indicates the disease. Thus, we have a total of 2000 images. After loading the images, we preprocess them. We also create three more images from each image to increase the number of training images and achieve better performance. Thus, we increase the number of training images to 6000. For each image, we create four samples of itself along with its reflection about one of the two axes and both axes. This seems logical to us because we can expect that with symmetry in the images, new images from the same distribution will still be present. Then we crop the middle 90 to 100 percent of the image and resize it to 96x96 pixels and normalize it to the range [0, 1]. Then we save the images in two folders, 'normal' and 'polyps'. By saving the images, we can avoid time-consuming preprocessing operations during training and directly load the preprocessed images.

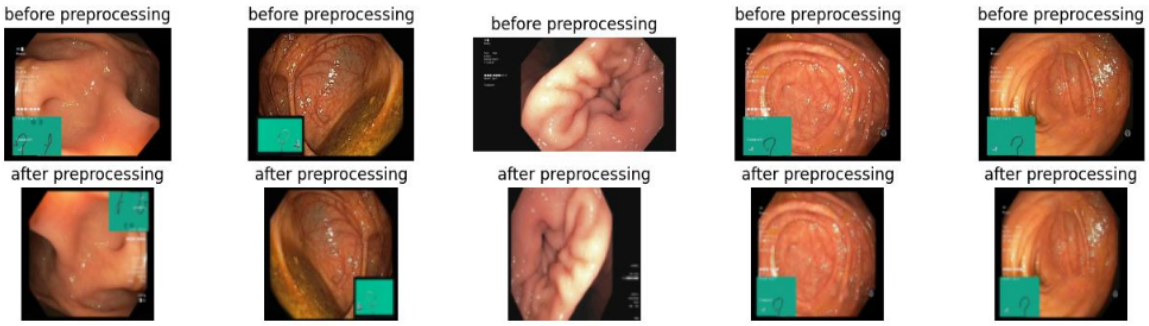

Figure 1: A few sample images before and after preprocessing.

1.3 EndoVAE Architecture Design

VAE models, like autoencoders, consist of two parts: an encoder and a decoder. VAE models are trained by minimizing the reconstruction error between the input x and the decoder's output $\hat{x}$. Also, to force the encoder to map the input to a latent representation space with a Gaussian distribution, the reconstruction error is added to the KL Divergence error, which acts as a regularizer and causes the two distributions to become closer. Thus, the VAE loss function is calculated as follows:

$$L_{VAE} = D_{KL}(\mathcal{N}(\mu, \sigma)||\mathcal{N}(0, 1)) + \frac{1}{L}\sum_{l=1}^{L} \log p(x|z_l; \theta)$$

In the above relation, the first term is the KL Divergence between the Gaussian distribution with the obtained mean and standard deviation and the standard Gaussian distribution, so that the model tries to get as close as possible to a zero mean and a standard deviation of one. The second term represents the reconstruction error.

In the second term, L is the number of samples we choose in the Monte Carlo sampling method to estimate the encoder's distribution. z is also a sample from the latent space obtained from the encoder, and $p(x|z; \theta)$ is the probability distribution of x given z, and θ are the decoder's parameters.

In other words, the first term is for finding the mean and standard deviation as close as possible to the standard normal distribution, and the second term also tries to make the reconstruction of the input from the latent space by the decoder possible.

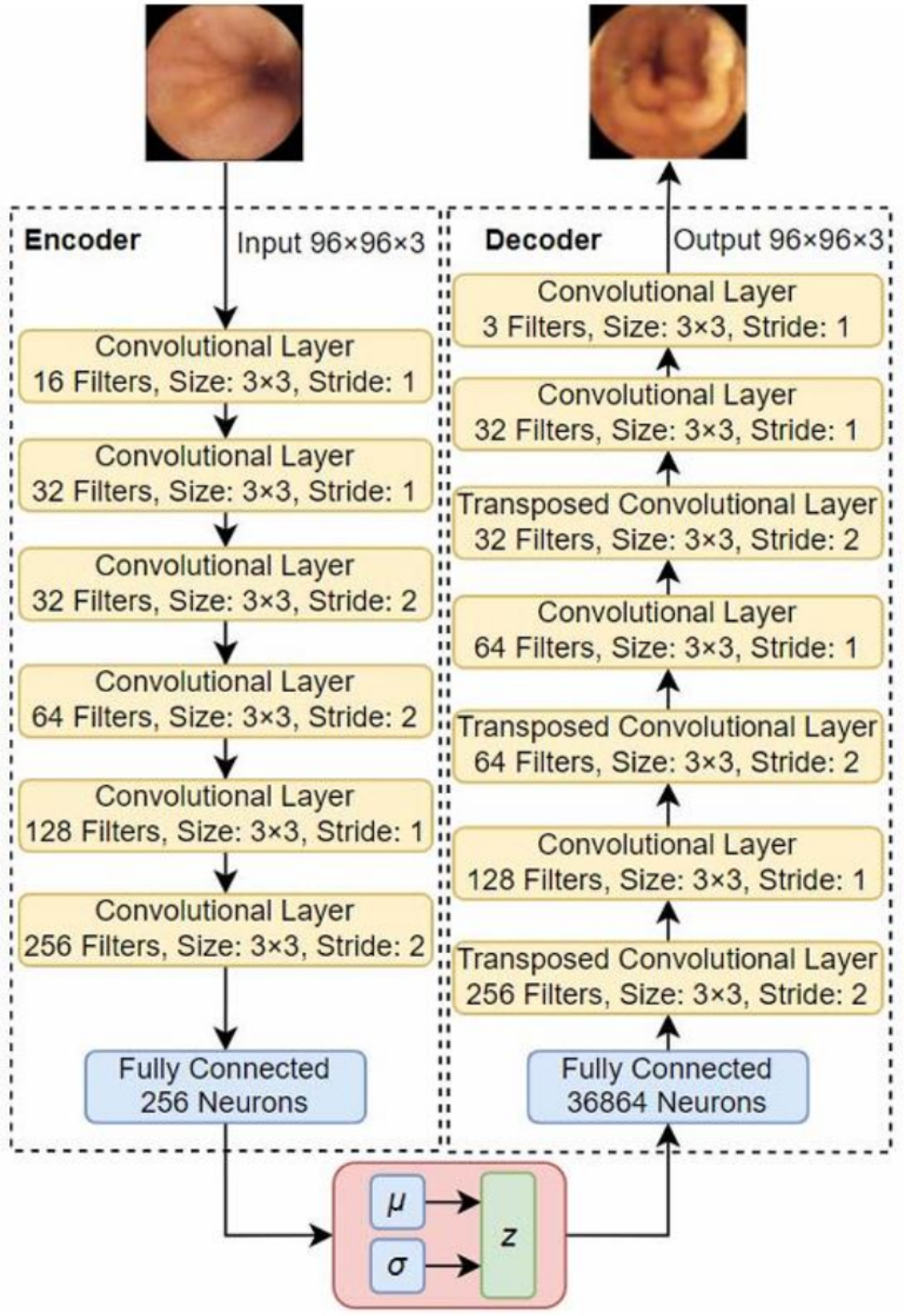

Figure 2: Architecture of the VAE model from Figure 1 of the paper.

1.3.1 Encoder Part

In autoencoders, the encoder part takes the input X to the compressed space Z, but unlike a regular autoencoder, in a VAE, the encoder from the input reaches two separate

vectors, which are the mean μ and the standard deviation σ of the Gaussian distribution $\mathcal{N}$.

The encoder part of the model consists of six consecutive convolutional layers. The first two layers have 16 and 32 filters, respectively. The third layer also has 32 filters, and the number of filters for each of the subsequent layers is twice the previous one. The output of the convolutional layers is converted to a one-dimensional vector and passes through a fully connected layer containing 256 neurons. Then two fully connected layers with 6 neurons are connected in parallel to those 256 neurons, and their outputs are the mean μ and the standard deviation σ of the latent space.

1.3.2 Decoder Part

The decoder part, by receiving the latent space representation, tries to reconstruct the input x. In the model used, the decoder's input initially passes through a layer containing 36864 neurons and then through 7 transposed convolutional layers with 256, 128, 64, 64, 32, 32, and 3 filters, respectively.

The decoder has about 1.3 million parameters, which is much less than the encoder. In this model, convolutional and transposed convolutional layers are placed alternately, and then the input vector is brought to the desired dimensions. A transposed convolutional layer, unlike a regular convolutional layer that reduces the dimensions of its input, increases its input dimensions. That is, for each pixel of the image, it multiplies the kernel values by it and adds them in the output.

1.3.3 Model Assembly

Although one can assume that the encoder's output is the mean and standard deviation and train the model accordingly, in practice, it is usually assumed that the output of the encoder part is the mean and the log variance, i.e., $\log \sigma^2$. We know that the standard deviation and variance are both non-negative, whereas the log variance can have any value. If we want to output the standard deviation, we must force the model with methods like using the ReLU activation function in the output to produce a non-negative output.

Another problem with using the standard deviation is that its values are usually very small numbers between zero and one, and taking the logarithm makes the model's training more stable and prevents problems like exploding and vanishing gradients.

Finally, in the KL Divergence relation present in the loss function, the log variance is present, so by outputting it from the model, we can use it directly and not calculate it again.

We know that the decoder must receive a sample from the Gaussian distribution with the parameters found by the encoder. Sampling from a distribution is a random operation, and its gradient cannot be calculated. For this reason, with **reparameterization** methods, we try to calculate the gradient of random variables. For this, we must write a function that calculates z with the help of the mean, standard deviation, and a new random variable that is not dependent on the model. We assume the random variable epsilon ϵ is a sample from the standard normal distribution, we generate a sample of epsilon and calculate z from the following relation:

$$z = \mu + \sigma \odot \epsilon$$

where $\odot$ in the above relation is element-wise multiplication. Thus, epsilon is an input for the whole model, and we can calculate the gradient from the mean and standard deviation towards the initial parts of the model.

1.4 Defining Loss Functions

As explained in section 2.2, the loss function used in the paper is as follows:

$$L_{VAE} = D_{KL}(\mathcal{N}(\mu, \sigma)||\mathcal{N}(0, 1)) + \frac{1}{L}\sum_{l=1}^{L} \log p(x|z_l; \theta)$$

Usually, the loss functions used in VAEs are similar to the above relation and consist of two parts. The first part, which is the KL Divergence in the above relation, and we also use it, is used with the aim of bringing the distribution closer to the standard normal distribution, i.e., bringing the mean to zero and the standard deviation to one.

The second part is also the reconstruction error, which is used so that the probability of producing the output from the sample z is high, or in other words, the model's output is similar to the input. For this part, we use two common loss functions, Binary Cross-Entropy and Mean Squared Error, and review the results.

Although the BCE loss function is used for classification, its use in VAEs is also common. One of the reasons for this is the use of the sigmoid activation function in the model's output, which is mathematically similar to the Bernoulli distribution, and the use of BCE for the Bernoulli distribution is recommended. Also, the presence of the log in BCE can partially neutralize the effect of the power in the sigmoid and cause the saturation of the sigmoid for very large and small values to be reduced.

Of course, there are some reasons for using MSE instead. Among them is that MSE is the most used error in regression and, unlike BCE, is a symmetric error, meaning, for example, the error between 0.5 and 0.6 is equal to the error between 0.7 and 0.6. As a result, we review both errors.

1.5 Model Training

According to the paper, we use a batch size of 128 and the Adam optimizer. Also, similar to the paper, we use a learning rate of 0.001. The paper states that the model was trained for 5000 epochs, but due to the increase in the number of samples with augmentation and having more samples than the paper, we train the models for 3000 epochs to be economical in terms of training time as well. First, we train the model with the MSE error.

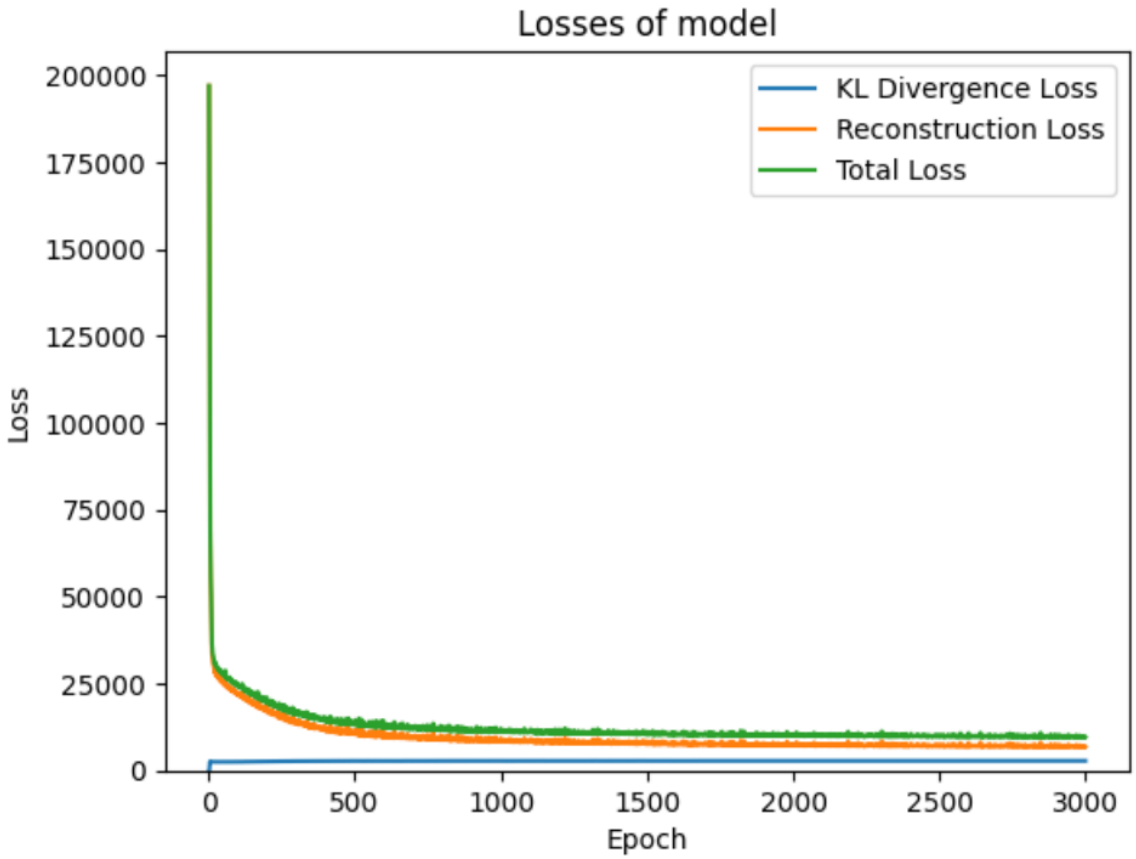

Figure 3: Changes in the loss of the model trained with MSE reconstruction loss.

In the image above, the value of each of the MSE and KLD errors and their sum is shown. Because the values of the two errors are summed on each batch, they have large values. It is also clear that the value of the KLD error is much smaller. At the 50th epoch, the KLD error value is 2477 and the MSE error is 25050, and with the increase of the KLD error in the subsequent epochs, the MSE error and the total error decrease, so that finally the KLD error reaches 2777 and the MSE error reaches 6830.

Now we train another model with the BCE reconstruction loss function for 3000 epochs.

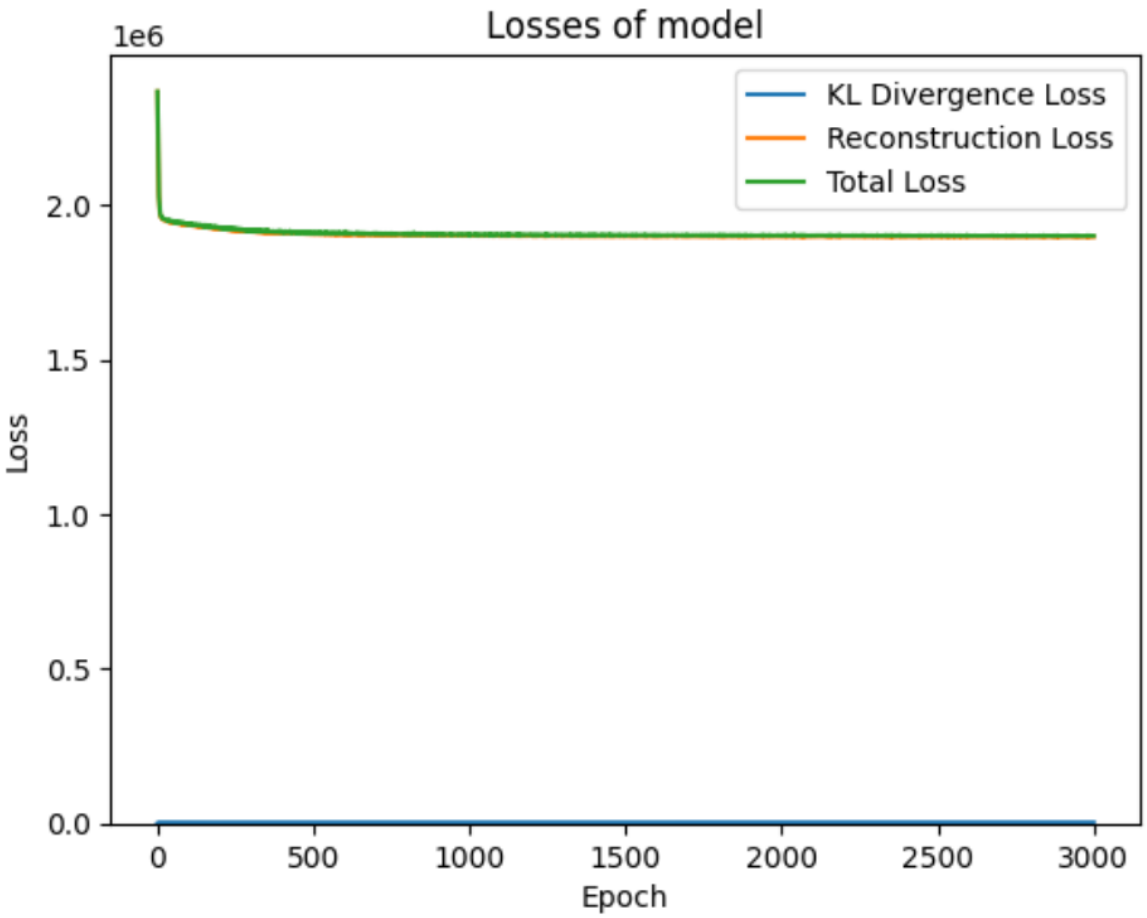

Figure 4: Changes in the loss of the model trained with BCE reconstruction loss.

Similar to before, the KLD error is negligible compared to BCE, and it can be seen that the total error has fallen on the BCE error. At the 50th epoch, the KLD error value is 2990 and the MSE error is 1944474, and with the increase of the KLD error in the subsequent epochs, the MSE error and the total error decrease, so that finally the KLD error reaches 3307 and the MSE error reaches 1896299.

1.6 Qualitative Generation and Reconstruction

1.6.1 Generation

We put both models in ‘eval‘ mode. Then we sample 9 noise vectors of length 6 from the standard Gaussian distribution and display them in a three-by-three grid.

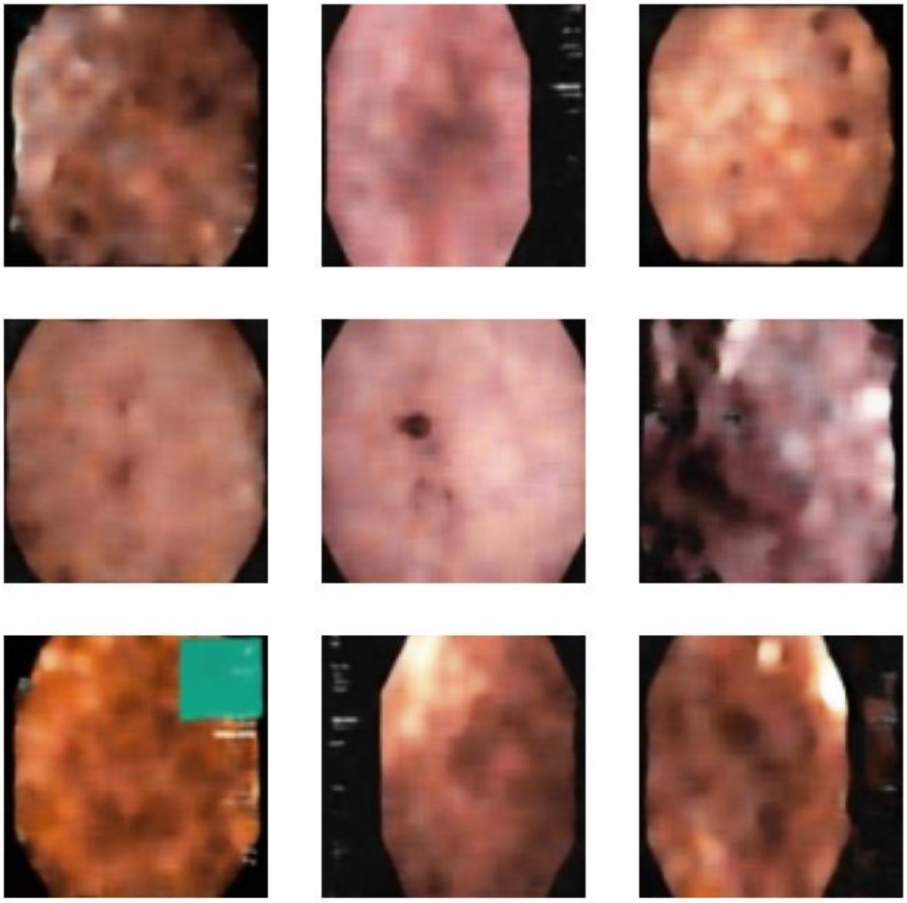

Figure 5: Samples generated from noise by the MSE model.

It can be seen that the MSE model has learned the overall distribution of the images, but similar to what was written in the paper, this model has the problem of blurry images. Also, some unhelpful details like the green square part in some images and the writings next to the images are also visible in the generated images, which, if necessary, can be removed with other preprocessing or more images for training.

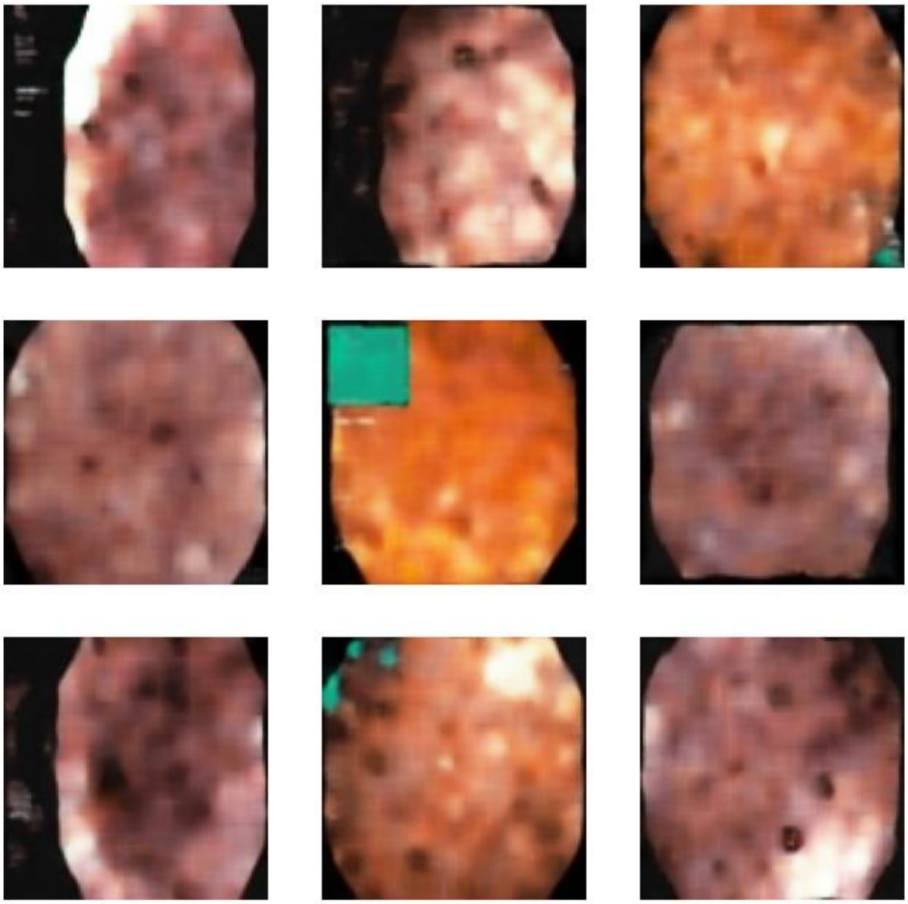

Figure 6: Samples generated from noise by the BCE model.

The samples produced by the model trained with the BCE error are also very similar to the samples of the model trained with the MSE error. One of the things that can be seen in these samples is the corners of images number 3 and 8 from the top left, which can be seen to be somewhat green, but like image number 5, the green part is not in the form of a square and has become part of the main image, which is not desirable.

1.6.2 Reconstruction

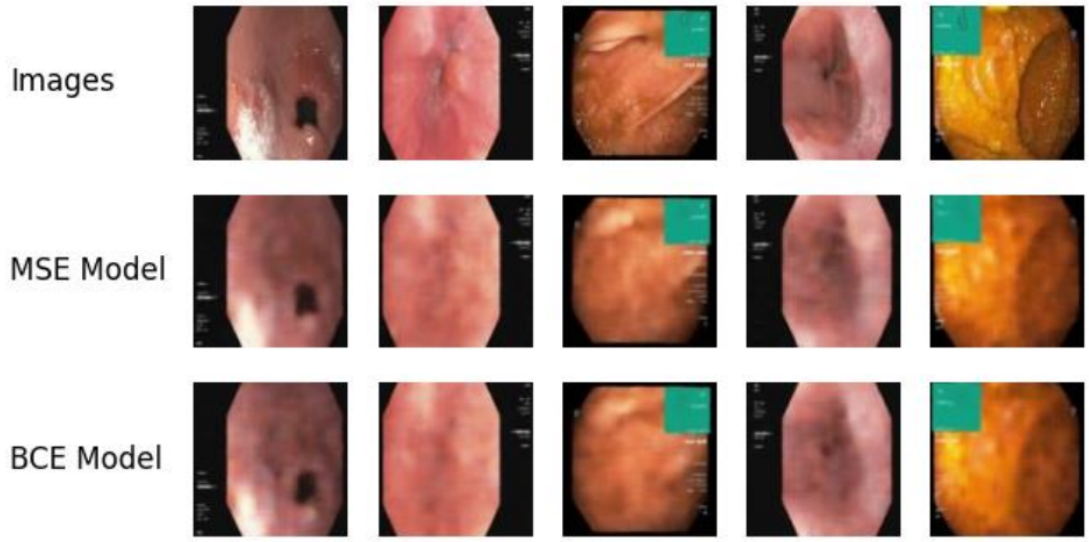

Figure 7: A few samples of healthy images along with their reconstruction by the two models.

From the figure above, it can be seen with what quality the trained models on healthy images can reconstruct the samples seen during training. As is clear from the samples above, the reconstructed images are very similar to the real images, but here the blurriness of the images is more apparent. The blurriness of the images is generally one of the problems of VAEs, and this problem is also mentioned in the paper, and the authors have stated that they will try to solve this problem in their future works. Not much difference is seen between the two models from the image above.

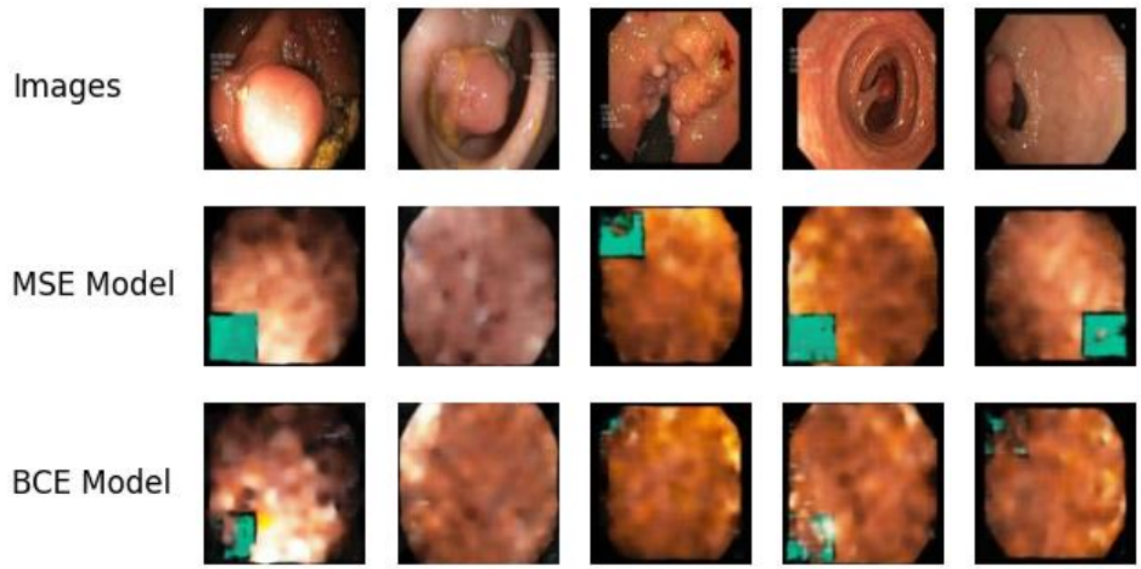

Figure 8: A few samples of polyp images along with their reconstruction by the two models.

In the image above, we give samples of polyp images that the models have not seen during training and their distribution is also slightly different from healthy images to the models to reconstruct. Of course, in the paper, in the training images, along with the healthy samples, a number of patient samples were also present, but as was requested, in this exercise, we trained the model with healthy samples. From the images, it is clear that the reconstructions have a much lower quality, part of which can be due to not seeing these images during training and more importantly, not seeing any images of polyps. Because in none of the images can any details similar to the polyp part be observed. It can be seen that the green square, although not present in the real images above, the model has produced it. This problem is more present in the MSE model. Apart from that and a slight difference in color and light, the reconstructed samples of the two models are similar.

1.7 Quantitative Evaluation

Some metrics are defined for measuring the similarity and correlation between an image and a reference image, and in this exercise, we use two common metrics, PSNR and SSIM, for more accurate numerical measurement. We generate reconstructed samples with the models for 50 polyp samples and then report the mean and standard deviation of the metrics in a table.

PSNR or Peak Signal-to-Noise Ratio is a metric used for evaluating the quality of compression or reconstruction of images, videos, or other data. This metric shows the ratio of the maximum possible value of a signal to the amount of noise. Usually, the higher the value of this metric, the better the quality. To obtain this metric, we first obtain the MSE between two images and then, by taking it to the logarithmic scale, the PSNR value is obtained. More precisely, PSNR is obtained as follows:

$$\text{PSNR} = 20 \times \log_{10}\left(\frac{L}{\sqrt{\text{MSE}}}\right)$$

where L in the above relation represents the maximum possible value, which is 255 in normal images.

The next metric is **SSIM** or Structural Similarity Index Measure, which works based on the natural structure of the image and measures the similarity between two images in terms of three criteria: brightness, contrast, and structure. For this reason, because this metric, unlike the previous metric, compares the overall structure instead of pixels, it can usually be said to be more similar to human vision and expectation.

Table 1: Evaluation metrics of the VAE model trained with MSE loss.

Metric	Mean	Standard Deviation
PSNR	16.952047	2.294434
SSIM	0.470127	0.084576

PSNR values for 8-bit images are suitable between 30 and 50, and it can be seen that the PSNR value is much lower. Also, SSIM, the closer it is to one, the better, and usually values above 0.9 are suitable, but it can be seen that the average SSIM is also low.

Table 2: Evaluation metrics of the VAE model trained with BCE loss.

Metric	Mean	Standard Deviation
PSNR	16.584135	2.276679
SSIM	0.447347	0.093633

The numerical metrics for this model also have similar values to the previous one and are slightly weaker, but not so much that the difference is significant, and it is possible that the differences are due to the difference in the 50 selected images. Again, similar to before, it can be said that the numerical values are far from the suitable values in reconstruction with quality.

1.8 Result Analysis and Final Discussion

The low values of these metrics, in our opinion, can have various reasons. The main reason could be due to the difference in the dataset used with the paper's dataset. Also, as we wrote above, in our opinion, one of the main reasons could be not training on any of the polyp images because, after all, the distribution of these images is slightly different from the distribution of healthy images, and the model has only learned how to reconstruct healthy images.

Of course, as we also saw in the reconstruction of healthy images, the reconstructed samples, although seen during training, were blurry, so other problems also exist. Since in the training images, augmentations must be applied cautiously so that images outside the distribution are not created, and given that the number of our images was even more than the paper before augmentation, and after augmentation, the number of our training images was more than 6 times the training images of the paper, the problem is not from the number of training images. Of course, the training images of the paper seem to be of higher quality and, unlike our dataset's images, do not have writings or logos on them. Of course, by cropping a larger part of the sides of the image, we could have reached images with higher quality but more zoom.

There are many ways that can be explored to achieve better performance. For example, the model's structure can be changed, and by adding skip connections or other methods like various regularizers such as dropout, batch normalization, and other methods, better results may be achieved.

Also, many loss functions have been introduced for VAEs, and the loss function used in this exercise is one of the initial loss functions in VAEs. For example, by using beta-VAE and giving weight to the two components of the error, better performance can be achieved.

One of the very important reasons for the weak performance, in our opinion, is the size of the latent space. The input image size is 96x96x3, and the latent space size is only 6, meaning the model is forced to lose a lot of details in this compression, and thus it leads to very blurry images. Increasing the latent size to larger values like 100 can have a great impact.

1.9 Bonus: Classifier Training

In this section, we intend to define a convolutional classifier to try to distinguish between real and reconstructed images. For this, we select 300 polyp images as real images and,

by giving these images to the models, produce reconstructed images. From among the real images, we select 200 of them and, by selecting the corresponding reconstructed images, form the training images. Similarly, from 100 real images and the corresponding reconstructed images, we form the test images.

We initially used the encoder model's architecture for classification, but the model could not even learn in 100 epochs, the main reason for which is the small number of training images. Of course, the model also has a large number of parameters and complexity for this number of images. For this reason, we chose the pre-trained Resnet18 model, and after loading and freezing its convolutional part, we add 256 neurons with ReLU and then dropout with a rate of 0.3 and then 2 neurons for detecting real or reconstructed to its end. We train the two classifier models for the two VAE models for 30 epochs, and the results are as follows:

Table 3: Evaluation metrics of the classifier for distinguishing real and reconstructed images.

Metric	Classifier for MSE VAE	Classifier for BCE VAE
Accuracy	0.995	1.0
AUC	1.000	1.0

It can be seen that both classifiers correctly identify the images, except for one image for the MSE model's classifier, which again shows the large difference between the real and reconstructed images and their easy distinction. Of course, the Resnet18 model has also been trained on the large ImageNet dataset and can extract features well.

1.10 Summary

In this exercise, we learned how to train VAE models to be able to generate images that we have a small number of with this method in any desired number and use it in the field of medicine. We also became familiar with the loss function used in VAEs and, by reviewing its components and analyzing the results in the use of two components, MSE and BCE, we tried to review the effect of these two loss functions. Our reviews did not observe much difference in the large number of epochs (3000) between these two loss functions.

We also became familiar with two widely used metrics in evaluating the similarity of two images, named PSNR and SSIM, and reviewed the results of the models with the help of these two metrics. By reviewing numerically and qualitatively, we realized that the generated samples have a large difference with the original samples, and we reviewed the possible reasons for this problem.

Finally, by training a convolutional classifier on real and reconstructed images, we were able to distinguish these two classes from each other with good performance, and of course, the main reason for this, in our opinion, is the weakness of the VAE model in producing unrealistic samples, which our qualitative and numerical reviews also indicated this issue.