A PROJECT REPORT ON

SELF-SUPERVISED LEARNING FOR MEDICAL IMAGE ANALYSIS USING IMAGE CONTEXT RESTORATION

SUBMITTED TO
SHIVAJI UNIVERSITY, KOLHAPUR

IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF DEGREE BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

MR.	DAYMA ADITYA GIRDHAR	19UCS029
MR.	JOSHI SHUBHAM DYANESHWAR	19UCS049
MR.	DHAVALE SOURABH SUNIL	19UCS032
MR.	DANOLE SUMOD VIDYASAGAR	19UCS028
MR.	EKAL PRIYANSHU PRAKASH	19UCS033

UNDER THE GUIDANCE OF
PROF. MR. U. A. NULI

DKTE
Promoting Excellence in
Teaching, Learning & Research

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DKTE SOCIETY’S TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI
2022-23

D.K.T.E. SOCIETY'S
TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI
(AN AUTONOUMOUS INSTITUTE)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DKTE
Promoting Excellence in
Teaching, Learning & Research

CERTIFICATE

This is to certify that, project work entitled

SELF-SUPERVISED LEARNING FOR MEDICAL IMAGE ANALYSIS USING IMAGE CONTEXT RESTORATION

is a bonafide record of project work carried out in this college by

MR.	DAYMA ADITYA GIRIDHAR	19UCS029
MR.	JOSHI SHUBHAM DYANESHWAR	19UCS049
MR.	DHAVALE SOURABH SUNIL	19UCS032
MR.	DANOLE SUMOD VIDYASAGAR	19UCS028
MR.	EKAL PRIYANSHU PRAKASH	19UCS033

is in the partial fulfillment of award of degree Bachelor in Engineering in Computer Science & Engineering prescribed by Shivaji University, Kolhapur for the academic year 2022-23.

PROF. MR. U. A. NULI
(PROJECT GUIDE)

PROF.(DR.) D. V. KODAVADE
(HOD CSE DEPT.)

PROF.(DR.) L. S. ADMUTHE
(DIRECTOR)

EXAMINER: ________________

____

DECLARATION

We hereby declare that the project work report entitled SELF SUPERVISED LEARNING FOR MEDICAL IMAGE ANALYSIS USING IMAGE CONTEXT RESTORATION” which is being submitted to D.K.T.E. Society’s Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University, Kolhapur is in partial fulfillment of degree B.E.(CSE). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under the Copyright and Piracy / Cyber / IPR Act amended from time to time.

NAME	PRN	SIGN
MR. DAYMA ADITYA GIRIDHAR	**19UCS029**	
MR. JOSHI SHUBHAM DYANESHWAR	**19UCS049**	
MR. DHAVALE SOURABH SUNIL	**19UCS032**	
MR. DANOLE SUMOD VIDYASAGAR	**19UCS028**	
MR. EKAL PRIYANSHU PRAKASH	**19UCS033**	

ACKNOWLEDGEMENT

With great pleasure, we wish to express our deep sense of gratitude to Prof. Mr. U. A. Nuli for his valuable guidance, support, and encouragement in the completion of this project report.

Also, we would like to take the opportunity to thank our head of department Dr. D. V. Kodavade for his co-operation in preparing this project report.

We feel gratified to record our cordial thanks to other staff members of the Computer Science and Engineering Department for their support, help and assistance which they extended as and when required.

Thank you,

MR.	DAYMA ADITYA GIRDHAR	19UCS029
MR.	JOSHI SHUBHAM DYANESHWAR	19UCS049
MR.	DHAVALE SOURABH SUNIL	19UCS032
MR.	DANOLE SUMOD VIDYASAGAR	19UCS028
MR.	EKAL PRIYANSHU PRAKASH	19UCS033

ABSTRACT

Our project focuses on the use of self-supervised learning for medical image analysis, particularly for retinal fundus image segmentation. With limited labelled data in medical image domains, self-supervised learning offers a way to learn representations without labels by framing a supervised learning task to predict only a subset of information using the rest. Our approach involves generating training images by shuffling patches of the original images and training a U-Net model to restore the original images.

We use a publicly available DRIVE dataset of retinal fundus images and experiment with different layers and outputs of the U-Net model while adjusting hyperparameters to improve the accuracy of the model. Our results show promising accuracy on the segmentation task, indicating that self-supervised learning using context restoration has the potential for medical image analysis. However, further improvements are necessary for clinical use.

Our project demonstrates the importance of developing methods that can learn from unlabelled data and address the challenges of limited labelled data in medical image analysis. Our findings pave the way for future research in this exciting and rapidly growing field.

INDEX

1. INTRODUCTION

Deep Learning methods have achieved great success in computer vision. Especially, CNNs have recently demonstrated impressive results in medical image domains such as disease classification and organ segmentation Good deep learning models usually require a decent amount of labelled data, but in many cases, the amount of unlabelled data is substantially more than the labelled ones. Also, the pre-trained models from the natural images are not useful on medical images since the intensity distribution is different.

Besides, labelling natural images are easy, and just simple human knowledge is enough. However, the annotation for medical images requires expert knowledge. So we learn representations without labels by getting supervision from the data or image itself. It means that we can achieve this by framing a supervised learning task in a particular form to predict only a subset of information using the rest. Which is known as self-supervised learning.

Why Self-supervised learning?

Good deep learning models usually require a decent amount of labelled data, but in many cases, the amount of unlabelled data is substantially more than the labelled ones. Also, the pre-trained models from the natural images are not useful on medical images since the intensity distribution is different. So we learn representations without labels by getting supervision from the data or image itself

Why Image context restoration?

As self-supervised learning has many different ways for image classification or image learning like Auto-encoding, Random, Random + Augmentation, Relative Position, Jigsaw, and Context Restoration.

The retina is situated at the innermost surface of the eye, lining the back of the eyeball that plays a crucial role in vision. The retina is composed of several layers of cells, including photoreceptor cells, bipolar cells, ganglion cells, and various interneurons. The photoreceptor cells, called rods and cones, are responsible for detecting light. The primary function of the retina is to convert light into electrical signals that can be transmitted to the brain. When light enters the eye, it passes through the cornea, lens, and other structures before reaching the retina. The photoreceptor cells in the retina detect the light and convert it into electrical impulses. At the centre of the retina is a small area called the optic disc, where the optic nerve exits the eye. The optic nerve carries the electrical signals generated by the retina to the brain for visual processing. The retina can be affected by various diseases and disorders, such as macular degeneration, diabetic retinopathy, retinal detachment, and retinitis pigmentosa. These conditions can lead to vision loss and require medical intervention.

Retinal imaging plays a critical role in diagnosing and monitoring several ophthalmic diseases, such as diabetic retinopathy, glaucoma, and age-related macular degeneration. However, obtaining large-scale, diverse, and labelled retinal image datasets can be challenging due to privacy concerns, high costs, and limited availability of ground truth annotations. This scarcity of data often hinders the performance of deep learning models, which heavily rely on vast amounts of labelled data.

Deep Learning AI-based models were regularly used in the medical field. Because of the lack of expert doctors' availability, you must take diagnostic decisions that come under expert ability. since expert people are not available easily, you have to wait for a long period of time to take their appointment. The solution to that problem is to use AI-based deep learning models. The deep learning model has the capability to complement the experts in making the diagnosis.

Why Experts are not available?

The experts are scarred in count as well as if they are available they are very busy in order to give diagnosis. The diagnosis process also takes more time.

This project helps experts by making their job easy. By using an AI-based model we can find more information about diseases or various scans then we can show that extracted content to the expert and the expert can take the decision very quickly or if it may be possible that AI system itself is capable of taking a decision, we, may take that as a second opinion.

Problem of deep learning model

The main problem of the deep learning model is it required large training data. Actual problem in the medical domain is large training data is not available. It is not only data but also annotated data. The data that has been marked by experts is annotated data. annotated data is limited in the medical field. This data is required for training the model. The data is limited because scans are not shared publically because of the privacy of people. the data that is available should be annotated by medical experts and medical experts already have less time to annotate these results in a minute level. In addition to this natural images can be annotated by any general person or we can go for cloud sourcing also. But medical images cannot be annotated by a general person. even though we can go for experts again they will take charge for it. Because of this reasons availability of data is limited. If we run the model by limited annotated data model will not learn all the features correctly, result will be overfitting. If overfitting happens its prediction will be not corretly.it will be large losses in productions and performance of making any kind of model will be poor.to improve the performance, somehow they have to supply enough data to models.

Solution for that problem

In addition of add additional data we have two methods. one solution is to generate synthetically some data and could not rely upon real data. This synthetically data which resemblance real data can be used to train the model; hence performance will be improved.

In second method we have enough data those are not annotated. In some datasets have images but none of them as annotated, its only real images are annotated. If real images are annotated, we can use this as self-supervised fashion. In self-supervised learning the output is generated automatically and then some pre task is used to train the model and from this pre task model understand the underline feature which we can use to make actual diagnostic. In self-supervised learning there are many techniques out of that we use context restoration technique.

context restoration shuffling the context and gives image to AI based model to learn its appropriate places. if we get successful in these then model might have learn good amount of features which is useful to carry out actual tasks for segmentation. These trained model we use further for fine tunning.it is expected that final task would make better predictions as compared to older models.

1.1. Aim and objective of the project

1. To Generate training images for self-supervised context disordering.

2. To build self-supervised based model on context restoration to learn useful semantic features from images.

3. To train an auto-encoder to initialize the task-specific CNN.

4. To segment retinal images in multi-modal Magnetic Resonance(MR) images.

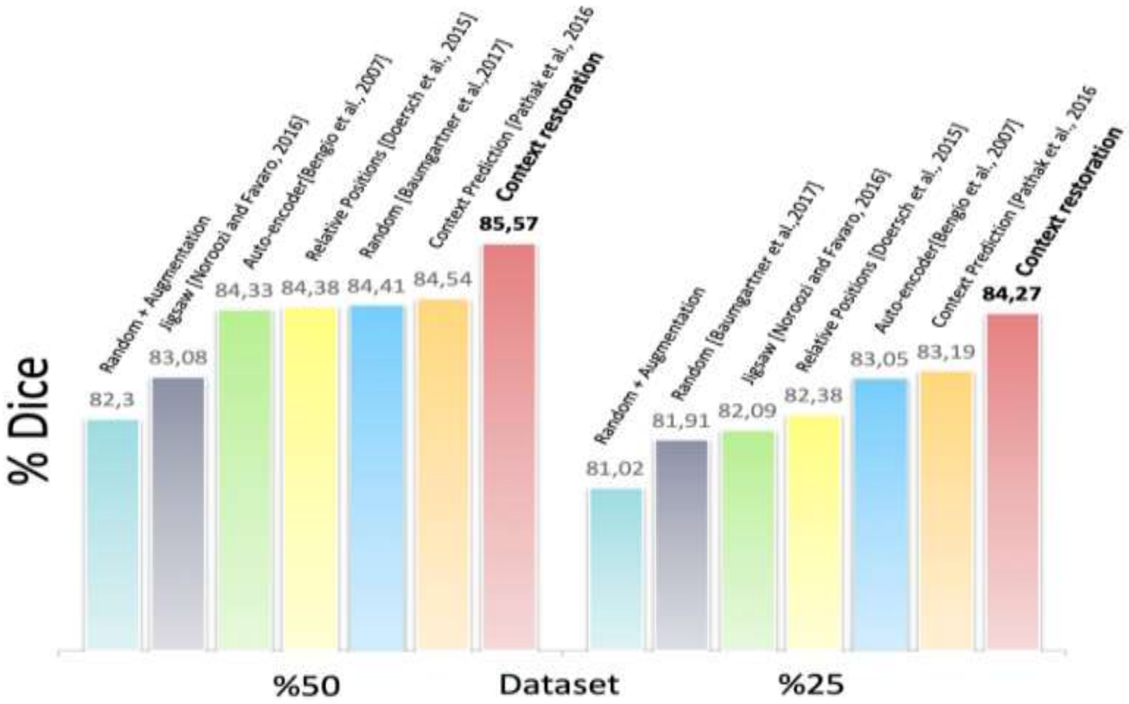


Figure 8. The segmentation results of the customised U-Nets in different training settings (Image by Author)

From the above fig. we can observe that for medical image segmentation (using U-Net) context restoration method is found to be the best method.

1.2. Scope

1. Proposed model is useful in segmentation of medical images, specifically for retinal images.
2. The retinal image dataset generated can be useful for training other models.
3. Generated Model can be used to extract blood vessels from retinal fundus images.

1.3. Limitation

• Time required to Train the model is different as per hardware resources used.

• Model needs to be retrained for different image datasets.

1.4. Timeline of the project

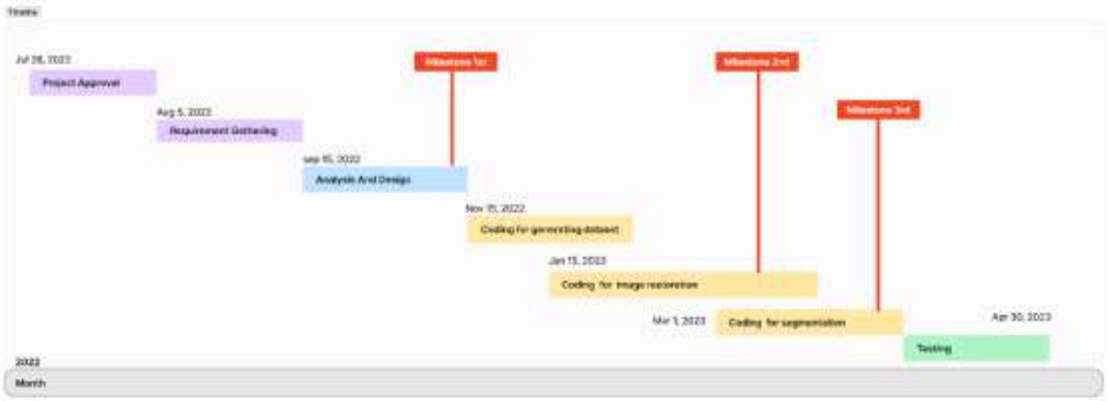


Fig. 1.4.0: Time of the project

1.5. Project Management Plan

Task Name	Duration	Start Date	End date
Domain Selection	2 days	1-8-2022	2-8-2022
Domain Finalization	2 days	3-8-2022	4-8-2022
Selection of Problem Statement	2 days	5-8-2022	8-8-2022
Finalization of Problem Statement	2 days	9-8-2022	10-8-2022
Study on Research Paper	35 days	11-8-2022	14-9-2022
Requirement Analysis	10 days	15-9-2022	25-9-2022
System Requirement	5 days	26-9-2022	30-9-2022
System Architecture	50 days	3-10-2022	21-11-2022
Stage 1 implementation	40 days	28-11-2022	25-1-2023
Stage 2 implementation	75 days	30-1-2023	17-4-2023
Testing	15 days	18-4-2023	5-5-2023

Table 1.5.0: Project Management Plan

1.6. Project Cost

Hardware Cost:

Components	Name	Pricing
Graphics Card	Nvidia Tesla K40	50730
Processor	Intel i5 11 gen	13410
RAM	Corsair 16 GB	4895
Total		69035

Table 1.6.0: Hardware Cost

Software Cost:

Line of code: To develop the system 0.75 lines of codes are required.

KLOC: KLOC is the estimated size of the software product indicates in Kilo Lines of Code.

KLOC = LOC / 1000

= 1000 / 1000

= 1

Effort: The effort is only a function of the number of lines of code and some constants evaluated according to the different software systems.

$T=aL^{b}$

Where L is KLOC (Kilo Line of Code)

T is time required to complete

$T=2.4*1K^{1.05} = 2.4$

Time: The amount of time required for the completion of the job, which is, of course, proportional to the effort put in. It is measured in the units of time such as weeks,months.

Time = c (Efforts)^{d}

$= 2.5 (2.4)^{0.38}$

$= 3.487$

Persons Required: Persons required is nothing but effort divide by time.

Persons Required = Efforts / Time

= 2.4 / 3.487

= 0.689

2. RELATED WORK

The recognition of puzzles is a challenging task in computer vision, and one approach to solving this problem is to cut each image into a grid and then disorder the pieces. The goal is then to recover the correct configuration of the image. In their paper, Wei et al [1] proposed a method that uses an iterative approach to solving this problem, adjusting the order of the patches in each step until convergence.

Their approach combines both unary and binary features of each patch into a cost function, which judges the correctness of the current configuration. The unary terms provide cues for the absolute position of a patch, while the binary terms provide cues for the relative position of two patches. By combining these two types of terms, the model can reduce the number of learnable parameters and thus alleviate the risk of over-fitting.

The authors used their approach in the context of puzzle recognition, where they showed promising results. Instead of solving the puzzle all at once, their iterative approach gradually improves the configuration of the pieces until convergence. This approach reduces the complexity of the problem and allows for better learning of the puzzle features.

The use of unary and binary terms in the cost function is a novel approach to puzzle recognition, and the authors show that this strategy is effective in reducing over-fitting and improving accuracy. Their approach is also applicable in other areas of computer vision, such as object recognition, where similar strategies could be used to reduce the number of learnable parameters.

In conclusion, the method proposed by Wei et al [1] shows promise in solving the challenging problem of puzzle recognition. The use of unary and binary terms in the cost function reduces the complexity of the problem and leads to improved accuracy. Their approach is an innovative contribution to the field of computer vision, and further research could investigate its potential for other applications.

In recent years, unsupervised learning has gained attention in the field of computer vision as a promising approach to learning useful features without requiring labeled data. M. Noroozi et al [8] have proposed a novel method for unsupervised learning of image context restoration using Jigsaw puzzles as a pretext task. The idea is to train a CNN to solve Jigsaw puzzles, which involves rearranging shuffled image patches to restore the original image, without human annotation.

To address the challenge of maintaining compatibility across different tasks, the authors have introduced a context-free network (CFN) which takes image tiles as input and explicitly limits the receptive field of its early processing units to one tile at a time. By training the CFN to solve Jigsaw puzzles, the model learns both a feature mapping of object parts as well as their correct spatial arrangement. The key insight

is that by solving Jigsaw puzzles, the CFN learns to identify each tile as an object part and how parts are assembled in an object.

The learned features are evaluated on both classification and detection tasks, and the experiments show that the proposed method outperforms the previous state of the art. This suggests that the learned features are useful for downstream tasks, demonstrating the potential of self-supervised learning in computer vision.

D. Kim et al [9] proposed a novel approach to self-supervised learning that involves introducing complications to traditional self-supervised tasks such as jigsaw puzzle, inpainting, and colorization. They also introduced a new task called “Completing damaged jigsaw puzzles” which involves solving puzzles with one missing piece and the remaining pieces without color. By training a convolutional neural network to solve these complicated self-supervised tasks, the model is encouraged to recover the damaged data, which can lead to a more robust and transferable representation.

The authors demonstrated that by complicating the self-supervised tasks, the resulting representations are better than the original versions. Furthermore, their proposed task of completing damaged jigsaw puzzles showed improved performance compared to other self-supervised tasks. The method is particularly useful for learning representations in scenarios where there is limited labeled data or the cost of manual labeling is high.

This approach has the potential to improve the effectiveness of self-supervised learning methods, which have become increasingly popular in recent years. It provides a way to create more challenging tasks that can lead to more robust and transferable representations. The experiments conducted by the authors showed promising results, indicating that the proposed method is a viable approach to further improve self-supervised learning.

The field of self-supervised learning faces a major challenge in selecting a suitable task for generating input and output instance pairs from data. While a range of self-supervision strategies have been proposed for natural images and videos, medical images present a unique challenge due to the need for expert annotations and the limited availability of labelled data. In the context of static images, several strategies have been proposed including patch relative positions, local context, and

The patch relative positions method, proposed by Doersch et al. in 2015, involves predicting the relative positions between a central patch and its surrounding patches in a 3x3 patch grid. Local context, on the other hand, focuses on predicting the missing parts of an image based on its surrounding context, while color-based methods involve predicting the color of a grayscale image.

Despite the progress made in self-supervised learning for natural images, there is still a lack of well-established methods for medical images. This highlights the need for further research and development of self-supervision strategies tailored to the unique characteristics of medical imaging data. Overall, identifying appropriate self-supervised tasks is critical for advancing the field of self-supervised learning and improving the performance of machine learning models on a wide range of tasks.

To elaborate further, exemplar learning involves training a model to recognize if two images belong to the same object category, based on a similarity metric. This is achieved by having the model learn features that are invariant to transformations, such as rotations and translations, that do not change the category of the object in the image. The model is trained on a large dataset of unlabelled images and learns to recognize common patterns and features in images that belong to the same category.

Exemplar learning has been shown to be effective in improving the performance of object recognition tasks, even when the training data is limited or noisy. It has been used in a variety of computer vision applications, including image classification, object detection, and image retrieval. One advantage of exemplar learning is that it does not require manual annotation of images, which can be time-consuming and expensive.

Self-supervised learning has emerged as a powerful technique for feature learning in computer vision. By leveraging the abundance of unlabelled data available in many domains, self-supervised learning can provide an efficient way to train models that can generalize to new tasks and domains. The development of new self-supervision strategies, such as the ones discussed above, will continue to be an important area of research in computer vision and machine learning.

One particularly noteworthy approach was proposed by Pathak et al. (2016), who introduced a self-supervised task that trains convolutional neural networks to learn how to fill in missing information in images with patchy context removed. This approach is straightforward and effective, as demonstrated by the success of subsequent works that build on this idea.

Doersch and Zisserman (2017) proposed a novel approach that combined multiple self-supervised learning tasks to improve feature learning. The authors unified patch relative position prediction, colorization, exemplar learning, and motion segmentation into one architecture and used a novel input

harmonization method to enable end-to-end training. The individual tasks' learned features were then fused with an L1 penalty loss to make their combination sparse. Their experiments showed that multitask self-supervised learning improves subsequent tasks more than single-task self-supervised learning. However, the downside of multi-task self-supervised learning is that it requires significant computational resources, as evidenced by their use of 64 GPUs for approximately 16.8K GPU hours.

3. Requirement analysis

3.1 Requirement Gathering:

In order to successfully develop a retinal image segmentation tool using Image context restoration, it is important to gather the necessary requirements. The following is a list of potential requirements for such a project:

1. Data sources: The first requirement is to identify and gather retinal image datasets that will be used for training and testing the UNET model. It is essential to have a large and diverse dataset that accurately represents the target population and contains a range of different retinal pathologies. But we do not have a large amount of data set. For this, we are using the DRIVE (Digital Retinal Images for Vessel Extraction).

2. Image pre-processing: Retinal images often contain various artifacts and image quality issues, such as poor contrast, uneven illumination, and motion blur. The pre-processing step involves addressing these issues and enhancing the image quality, which will improve the accuracy and reliability of the generated images.

3. UNET architecture: The choice of the UNET model is crucial in determining the quality of the generated images. It is important to explore different encoder, bridge, and decoder network architectures, as well as various hyperparameters and optimization techniques.

4. UNET Training: The system should train a UNET model using the generated shuffled retinal images dataset to learn the underlying distribution and restore realistic retinal images. This process involves training the encoder, bridge, and decoder networks.

5. Evaluation metrics: To measure the performance of the UNET model, it is important to define and implement appropriate evaluation metrics. Common metrics include the structural similarity index (SSIM), and the Fréchet inception distance (FID).

6. Output Formats: The system supports only output formats for the generated retinal images, which is PNG, to facilitate compatibility and further analysis or sharing

7. Accuracy: The generated restored and segmented retinal images should closely resemble real retinal images, both visually and in terms of features and characteristics relevant to the specific domain. The system should strive for high fidelity and avoid generating unrealistic or distorted images.

3.2 Requirement Specification

Requirement ID	Requirement Description	Priority	Type
RS-01	Data sources: The first requirement is to identify and gather retinal image datasets that will be used for training and testing the UNET model.	High	Functional
RS-02	Pre-processing: The system should pre-process the acquired retinal images to normalize their sizes, orientations, and color channels to ensure consistent inputs for the model.	High	Functional
RS-03	UNET architecture: The choice of UNET architecture is crucial in determining the quality of the restored images	High	
RS-04	UNET Training: The system should train a UNET model using the generated retinal images dataset to learn the underlying distribution and generate realistic retinal images.	High	Functional
RS-05	Evaluation metrics: To measure the performance of the GAN model, it is important to define and implement appropriate evaluation metrics.	High	Functional

Table 3.2.0- Requirement Specification Table

RS-06	Output Formats: The system should support output formats of PNG for the restored retinal images.	High	Functional
RS-07	Accuracy: The restored retinal images should closely resemble real retinal images, both visually and in terms of features and characteristics relevant to the specific domain. The system should strive for high fidelity and avoid generating unrealistic or distorted images	Medium	Non functional
RS-08			Non functional
RS-09	·		Non functional

4. Methodology

4.1 Architecture Diagram:

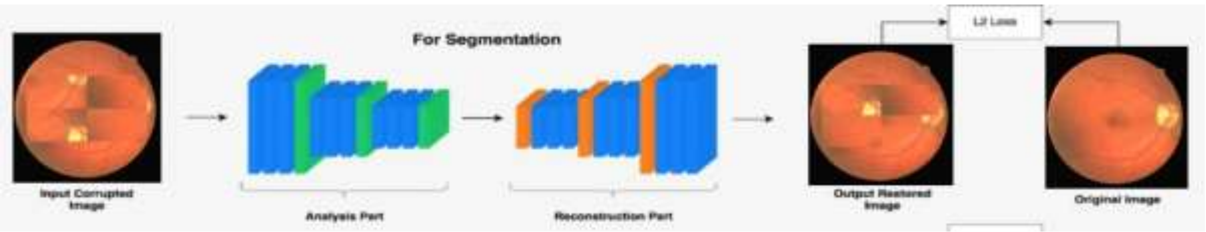


Fig. 4.1.0 : Architectural Diagram

4.2 Components:

4.2.1 Encoding/Analysis Part: The Encoding part of the CNN is responsible for extracting feature maps from the input images. This is done using stacks of convolutional units and down sampling units. Convolutional units typically consist of one or more convolutional layers, which apply a set of filters to the input image to extract features at different spatial scales. These features are then passed through an activation function to produce non-linear feature maps. The type of convolutional layer used can vary, and may include residual convolution layers, inception layers, densely connected convolution layers, and so on.

Down sampling units, on the other hand, reduce the spatial dimensionality of the feature maps, typically by using pooling layers. Pooling layers aggregate features within a local neighborhood to produce a single output value, reducing the size of the feature maps while preserving important features. Again, the specific type of down sampling layer used can vary, and may include inception pooling layers and other variations.

The weights of the CNN are learned during training, using backpropagation to update the weights based on the error between the predicted output and the ground truth labels. Once the weights have been learned, they are used to initialize the subsequent tasks. These tasks may include image classification, object detection, segmentation, or other related tasks. By initializing the subsequent tasks with the learned weights, the network is able to leverage the knowledge gained during the analysis part of the network to perform these tasks more effectively and efficiently.

4.2.2 Decoding/Reconstruction Part: The reconstruction part of the CNN is responsible for generating output images in which the context information has been restored. This is done using stacks of convolutional layers and up-sampling layers. Convolutional layers in the reconstruction part typically extract increasingly complex

features from the input feature maps, while up-sampling layers increase the spatial resolution of the feature maps to generate the final output images.

For segmentation tasks, the entire CNN architecture can be shared by the pretraining network and the subsequent segmentation network. This means that the weights learned during the self-supervised pretraining phase can be used to initialize almost all of the weights in the segmentation CNN. This initialization is important because it allows the segmentation CNN to leverage the knowledge gained during the pretraining phase to perform better on the segmentation task.

By initializing the segmentation CNN with the weights learned during pretraining, the network is able to more effectively extract meaningful features from the input images and use them to perform the segmentation task. This can result in better segmentation results, especially in cases where labeled data is scarce or expensive to obtain.

UNET Architecture:

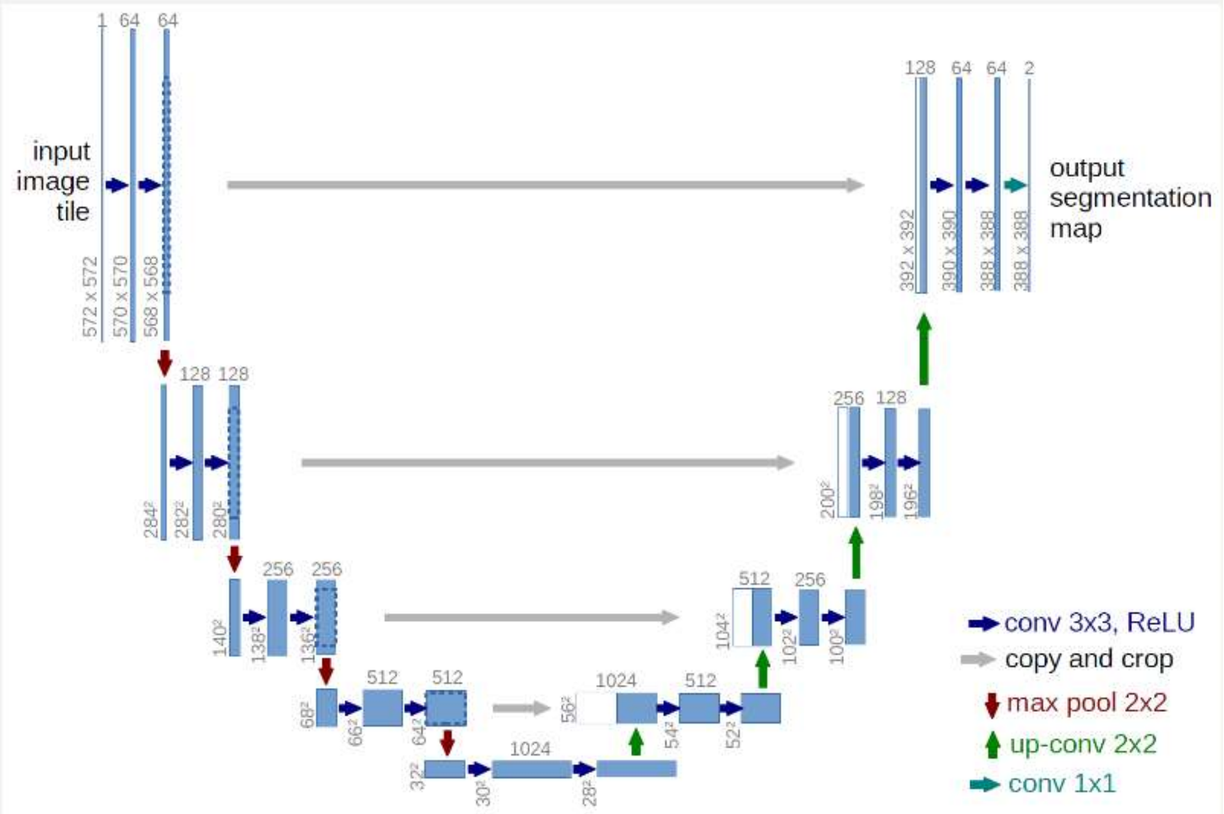


Fig. 4.2.0: UNET Architecture diagram

The U-Net is a convolutional neural network architecture that is commonly used for semantic segmentation tasks in image processing. It was introduced in 2015 by Olaf Ronneberger, Philipp Fischer, and Thomas Brox. The architecture of U-Net is inspired by the encoder-decoder architecture, which has been used in several computer vision tasks.The name U-Net comes from its shape, which looks like a "U" when viewed from above. The U-Net model has two main parts: the contracting path and the expanding path. The contracting path performs convolutional operations to reduce the spatial resolution of the input image, while the expanding path performs transposed convolution operations to increase the spatial resolution of the output.

The contracting path consists of a series of convolutional layers followed by max-pooling layers. The convolutional layers use small filters, typically 3x3, to extract features from the input image. The max-pooling layers reduce the spatial resolution of the feature maps by half, which helps to reduce the computational cost of the model and make it more efficient.

The expanding path is a mirror image of the contracting path. It consists of a series of transposed convolutional layers that up sample the feature maps. The transposed convolutional layers use learnable filters to up sample the feature maps and recover the spatial resolution of the input image. The feature maps from the contracting path are also concatenated with the feature maps from the corresponding layer in the expanding path, which helps to preserve the spatial information of the input image.

In addition to the contracting and expanding paths, U-Net also has skip connections that connect the feature maps from the contracting path to the corresponding layer in the expanding path. The skip connections help to transfer the low-level features from the input image to the output, which improves the accuracy of the segmentation.

Overall, the U-Net model is an effective architecture for semantic segmentation tasks in image processing. It has been used for various applications, such as medical image segmentation, satellite image analysis, and more. Its ability to preserve the spatial information of the input image and transfer low-level features through the skip connections make it a powerful tool for image segmentation.

The gray arrows indicate the skip connections that concatenate the encoder feature map with the decoder, which helps the backward flow of gradients for improved training.

4.3Algorithm

Stage 1 (Generate Dataset):

1. Start
2. Load the training images which present in the dataset (retinal vessel network)
3. Create the patches of images.
4. Shuffle all patches.
5. Generate all possible images.
6. Convert all images to PNG format.
7. End

Stage 2 (Image restoration):

1. Start
2. Load the Shuffled image and corresponding fundus images for training
3. Normalize the images to (512, 512, 3)
4. Create the encoder
5. Create the Bridge
6. Create the Decoder
7. Compile the models with loss function and optimizers
8. Write the training steps
9. Train the UNET
10. End

5. Implementation

5.1Environment Setting:

5.1.1 Environmental setting for running the model

Name	version
Python	3.9.16
Keras	2.6.0
Keras-preprocessing	1.1.2
Tensorflow	2.6.0
Tensorflow - gpu	2.6.0
Matplotlib	3.7.0
Matplotlib-inline	0.1.6
numpy	1.23.5
Pillow	9.4.0
ipyplot	1.1.1
ipykernel	6.19.2

5.1.2 For training STAGE 2:

Optimizer used for generator and discriminator is Adam. Learning rate is set to 0.0002.

Loss function used for generator and discriminator is binary cross entropy, MAE.

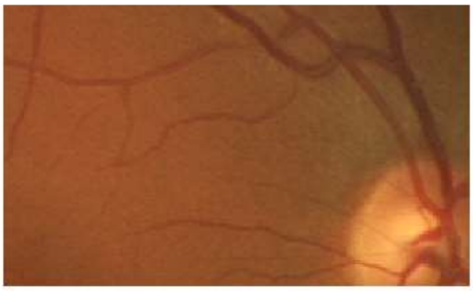
5.2 Experimentation

In our experimentation, we used a dataset of retinal fundus images from the DRIVE dataset, which contains 40 annotated (vessel networks) images of size 512 x 512 pixels. We split the dataset into 20 images for training and 20 images for testing. We did not perform any pre-processing on the images.

To perform self-supervised learning, we first generated a new set of images by shuffling small patches of the original images. This has done by extracting a small middle part from a image and made a 4 patches from it.

The patches are as shown below :

Patch - 1

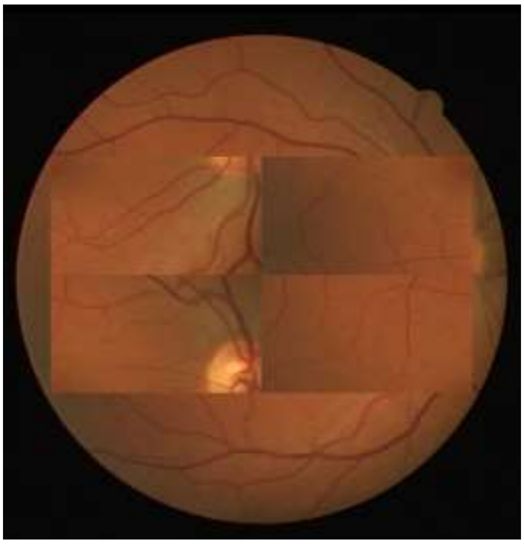


Patch - 2

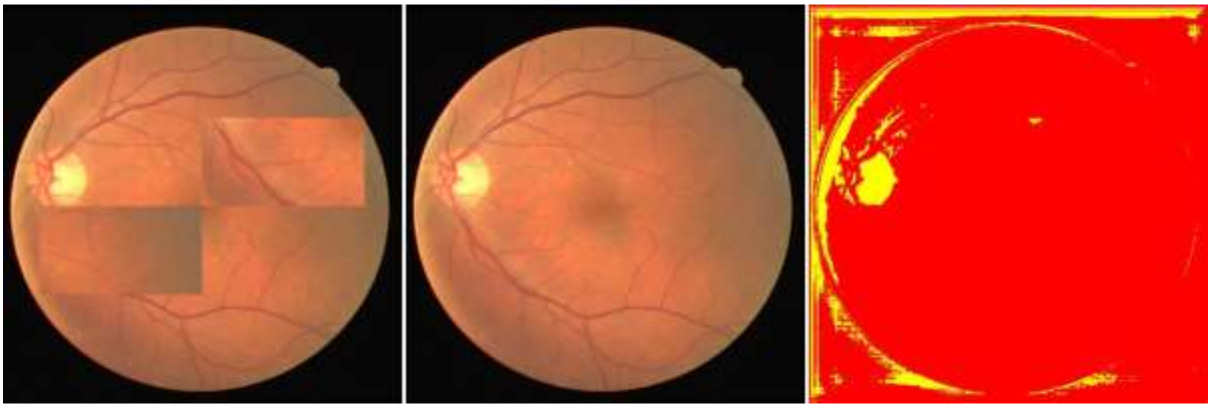
Patch - 3

Patch - 4

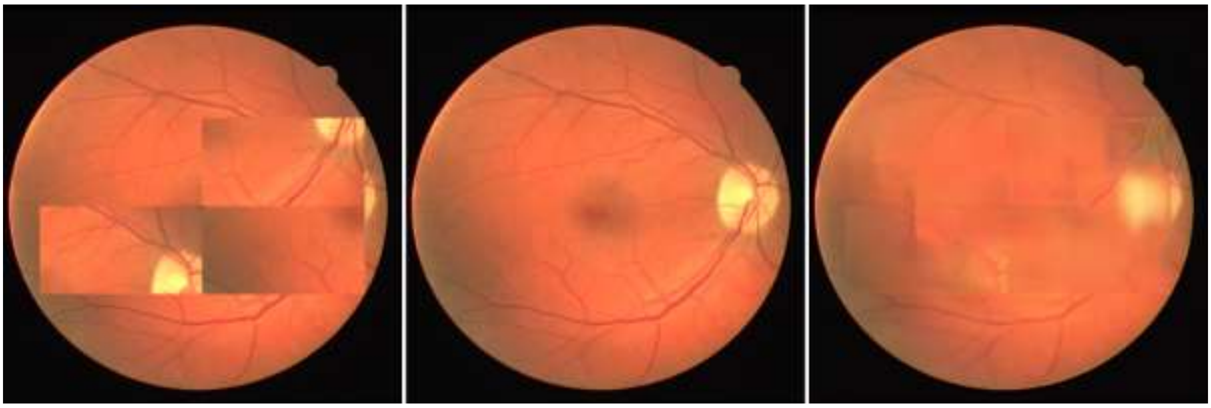
All these 4 patches are then shuffled in all possible 24 ways.Example shuffled eimage is as follows :

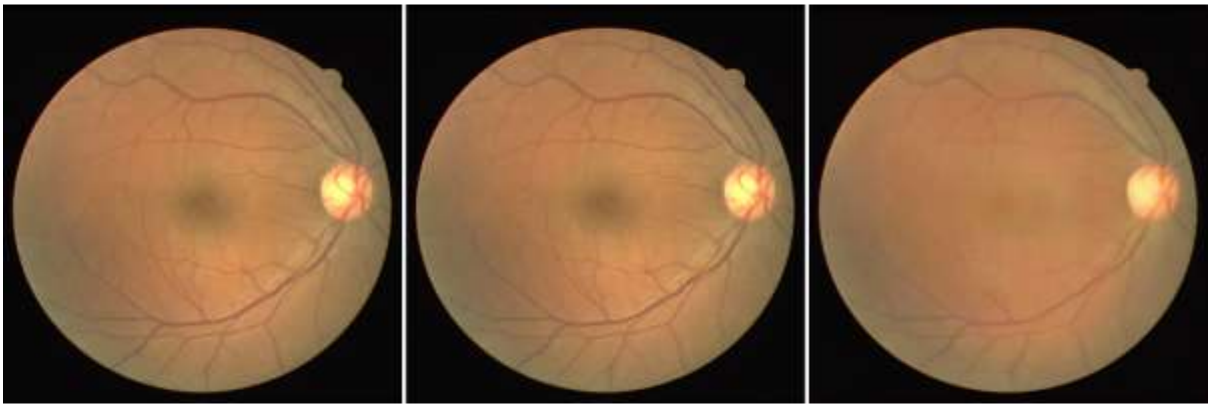


These distorted images were then passed through a U-Net model that was trained to restore the original image from the distorted image. We experimented with various U-Net architectures, including different numbers of layers, kernel sizes, and activation functions. We also implemented different outputs and checked their performance.



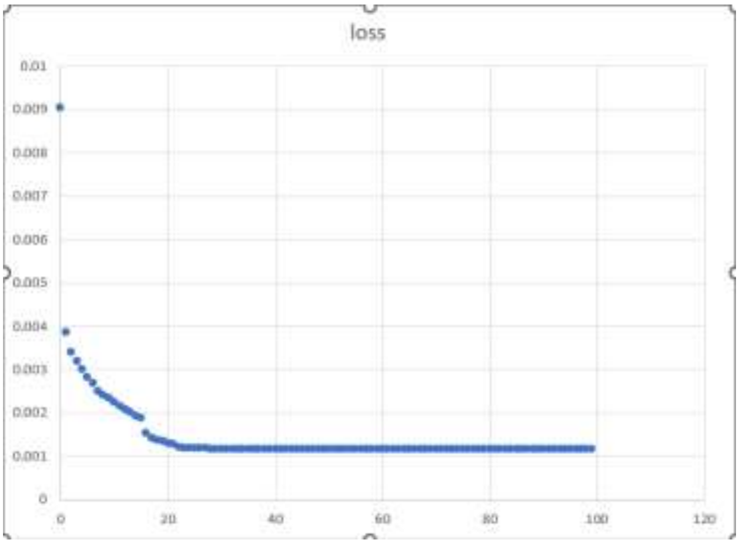
The above result is when we had an image of (512,512,3) but when only a single channel is allowed by a condition.



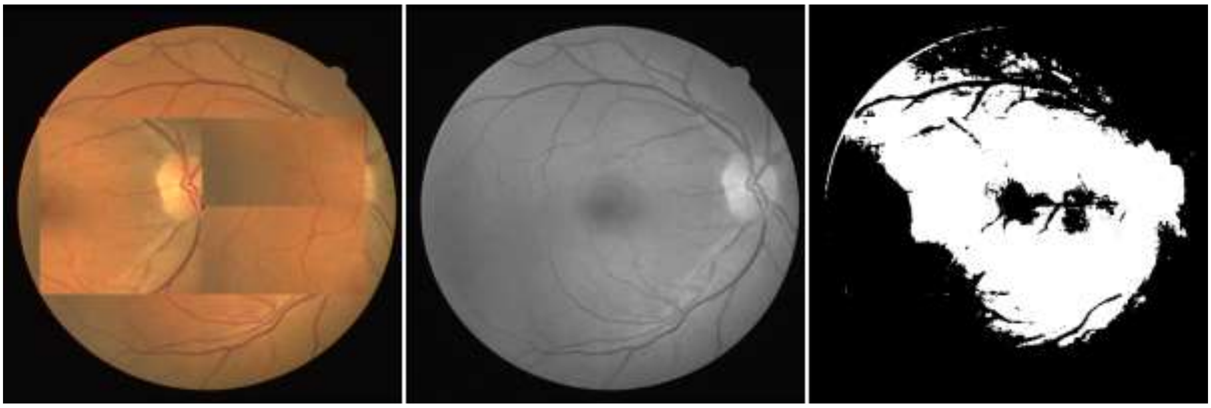


We got the above results when three channels are allowed and a UNET model is trained for image context restoration.

For above model, we got a loss vs epoch graph as :



To improve the performance of our self-supervised learning model, we experimented with different hyperparameters. We used grid search to explore different combinations of hyperparameters such as learning rate, batch size, and weight decay. We also experimented with different loss functions, including mean squared error (MSE) and structural similarity index (SSIM). Through our experimentation, we achieved some degree of accuracy, although we believe that further improvements are possible.



To evaluate the performance of our self-supervised learning model, we used various metrics such as peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and mean squared error (MSE). Our results showed that the accuracy of our model was decent, with PSNR and SSIM scores that were reasonably high. However, we also observed that the restored images did not closely match the original images, and the MSE score was relatively high.

Our experimentation has shown that self-supervised learning using image context restoration ^Using training dataset generated by using some technique like batch flipping, etc.^

has potential for medical image analysis, particularly in the field of retinal fundus image analysis. While our results have demonstrated some degree of accuracy, we believe that further improvements are necessary for the approach to be adopted for clinical use.

The application of self-supervised learning for medical image analysis is an area of active research with great potential for future advancements.

6. **Testing**

Testing of the project involves evaluating the performance and effectiveness of the self-supervised learning algorithm for medical image analysis. This involves testing various models involved in it. In this project, the primitive model used is the U-Net model. Therefore, for testing the model, we need to test mainly the U-Net model. Here is a detailed outline of how we deal with testing the underlying model. Testing a U-Net model in machine learning involves evaluating its performance on a dataset. Gather a dataset that aligns with your specific task, such as image segmentation. Split the dataset into training, validation, and testing subsets using standard practices like random sampling or stratified sampling. Ensure that the dataset has ground truth annotations or labels for the target variables you're interested in, such as segmented masks corresponding to the input images. Pre-process the input images and their corresponding masks as required by the U-Net model. Common pre-processing steps include resizing the images to a consistent size, normalizing pixel values, and converting the masks to the appropriate format (e.g., binary masks or categorical labels). Load the pre-trained U-Net model that you want to test. If you have trained the model from scratch, ensure that you have saved the model weights or checkpoints. Pass the pre-processed images through the U-Net model to obtain predictions. This involves using the model's forward pass to generate output masks or segmentations for the input images. Apply any necessary post-processing steps, such as thresholding or morphological operations, to refine the predictions if needed. Compare the predicted masks or segmentations with the ground truth labels from the testing dataset. Calculate evaluation metrics such as intersection over union (IoU), dice coefficient, pixel accuracy, or any other suitable metrics for your specific task. These metrics quantify how well the U-Net model performs in accurately segmenting the objects of interest. Based on the evaluation results and visual analysis, refine and iterate on the U-Net model if necessary. This may involve adjusting hyperparameters, exploring different architectures, incorporating additional data, or using techniques like data augmentation to improve the model's performance.

Model restores the context of original image and provides the resultant image. Therefore, the testing involves comparison of these two images out of which one is input image and other is resultant output image. This can be done using metrics like SSIM and FID. SSIM and FID are two commonly used metrics for evaluating machine learning models, particularly in the field of image generation or image-to-image translation tasks.

FID

A metric to evaluate the quality of generated images, the Frechet Inception Distance, or FID for short, evolved specifically to evaluate the efficiency of generative adversarial networks. Compared to the current Inception Score, or IS, the suggested score was deemed to be an improvement. Based on how successfully a set of synthetic photos is classified by the top-performing image classification machine Inception v3, the inception score calculates the quality of the set of photographs. How artificial pictures stack up against genuine ones is not represented by the inception score. The purpose of inventing the FID score was to assess synthetic pictures using statistics from a group of synthetic images in comparison to statistics from a group of actual photos from the target domain. Like the inception score, the FID score uses the inception v3 model. Specifically, the coding layer of the model (the last pooling layer prior to the output classification of images) is used to capture computer-vision-specific features of an input image. These activations are calculated for a collection of real and generated images. The activations are summarized as a multivariate Gaussian by calculating the mean and covariance of the images. These statistics are then calculated for the activations across the collection of real and generated images.

The distance between these two distributions is then calculated using the Frechet distance, also called the Wasserstein-2 distance. A lower FID indicates better-quality images; conversely, a higher score indicates a lower-quality image and the relationship may be linear.

The FID score is then calculated using the following equation:

$$d^2 = ||mu_1 - mu_2||^2 + Tr(C_1 + C_2 - 2*sqrt(C_1*C_2))$$

The feature-wise mean of the real and created pictures is represented by the "mu_1" and "mu_2" variables.

The covariance matrices, or sigma, C_1 and C_2, represent the actual and artificial feature vectors, respectively.

The difference between the two mean vectors is expressed as the sum squared, or ||mu_1 - mu_2||2. The trace linear algebra operation is referred to as Tr.

SSIM

The term "SSIM" (Structural Similarity Index) refers to a statistic that is frequently used to determine how similar two images are to one another. The Structural Similarity Index determines how similar two provided photos are and returns a result that ranges from -1 to +1. A number of +1 denotes a high degree of similarity or identity between the two provided images, whereas a value of -1 denotes a high degree of contrast between the two images. Frequently, these numbers are changed to fall between [0, 1], where the extremes have the same meaning. This measure takes an image's Luminance, Contrast, and Structure and separates them into their three main components. On the basis of these 3 attributes, a comparison between the two photos is made. The term "luminance" describes an image's general brightness or intensity. Averaging across all of the pixel data yields the luminance value. Contrast is a measurement of the difference in brightness between several areas of a picture. The standard deviation of each pixel's value is taken to determine the contrast. The organisation and layout of elements in a picture are represented by structure. By evaluating the correlation of regions of image in the produced and reference pictures, SSIM determines how structurally similar two images are. It evaluates how effectively the reference picture's structural information and connections are retained in the created image.

$$SSIM(x, y) = [l(x, y)]^\alpha * [c(x, y)]^\beta * [s(x, y)]^\gamma$$

Where:

x and y represent the two images being compared.

l(x, y) is the luminance comparison term, which measures the similarity of the mean intensities of x and y.

c(x, y) is the contrast comparison term, which captures the similarity of the standard deviations of x and y.

s(x, y) is the structure comparison term, which measures the similarity of the covariance of x and y.

α, β, and γ are weighting factors that adjust the relative importance of each term. These factors are typically positive values, and their sum is usually set to 1.

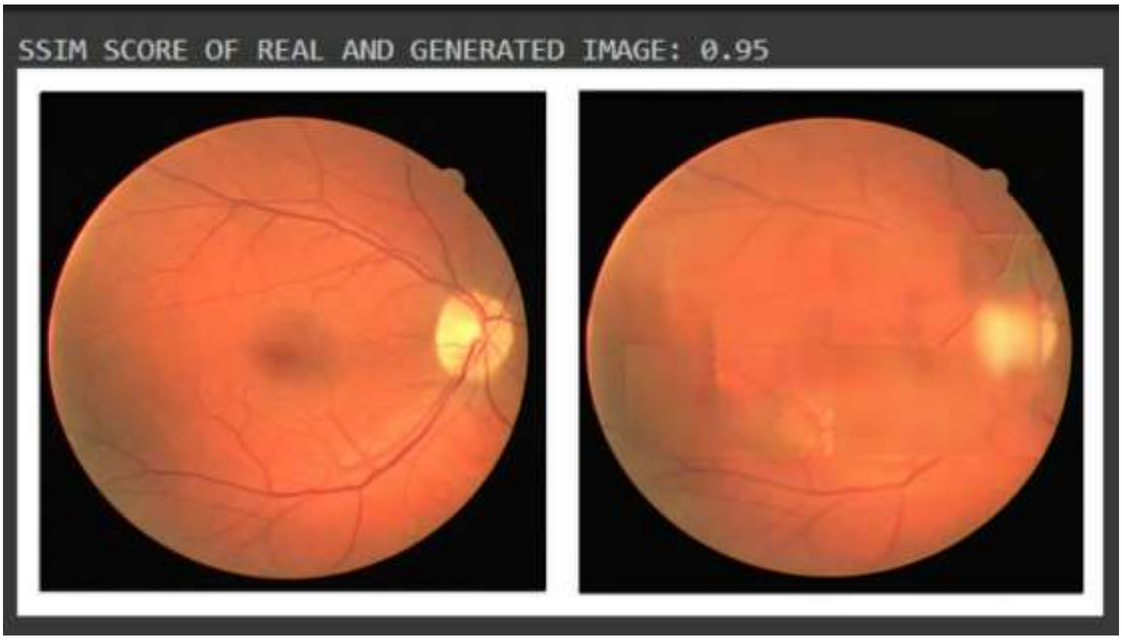
The computation of SSIM is not a difficult task since SSIM is a built-in method part of Sci-Kit's image library so we can just load it up and compute.

Both SSIM and FID provide quantitative measures to assess the performance of machine learning models. While SSIM focuses on assessing the structural similarity between images, FID takes into account the statistical properties of images and provides a measure of similarity in terms of visual features. It is common to use both metrics together to gain a more comprehensive understanding of the model's performance.

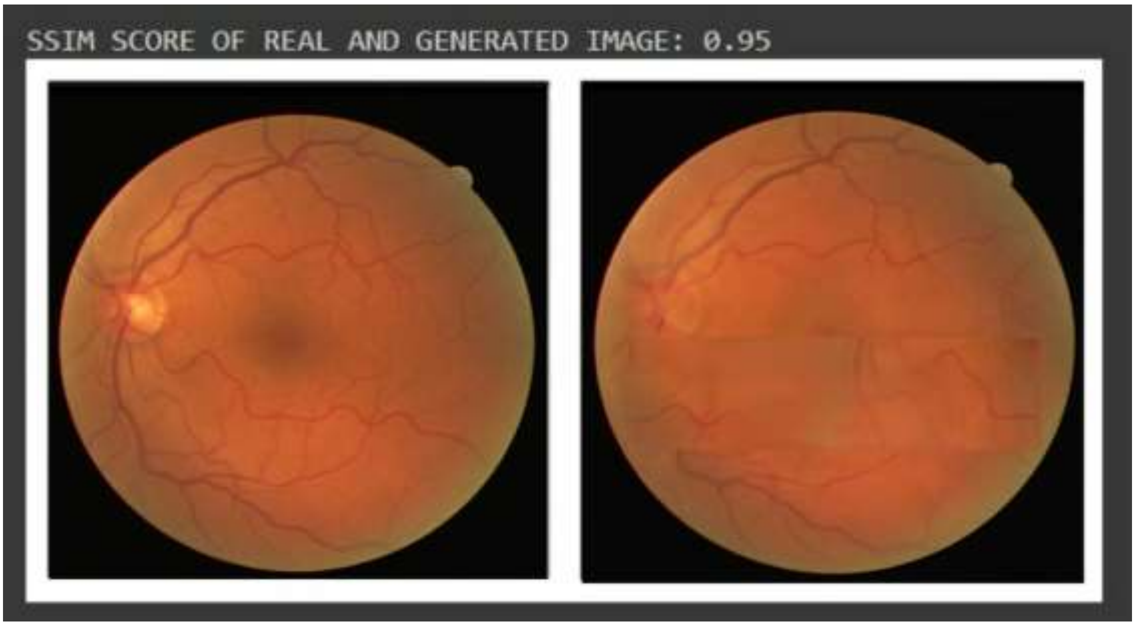
7.<u>Performance Analysis</u>

For testing of generated image we used two quantitative metrics which are : SSIM and FID.

SSIM values we got are : **0.95 ± 0.05**

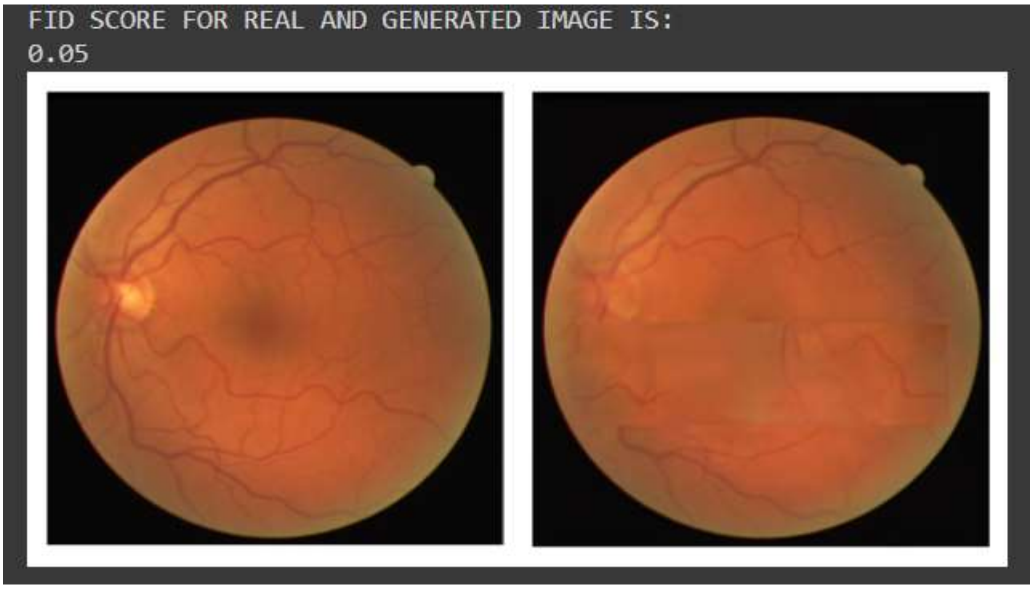


SSIM SCORE OF REAL AND GENERATED IMAGE: 0.95

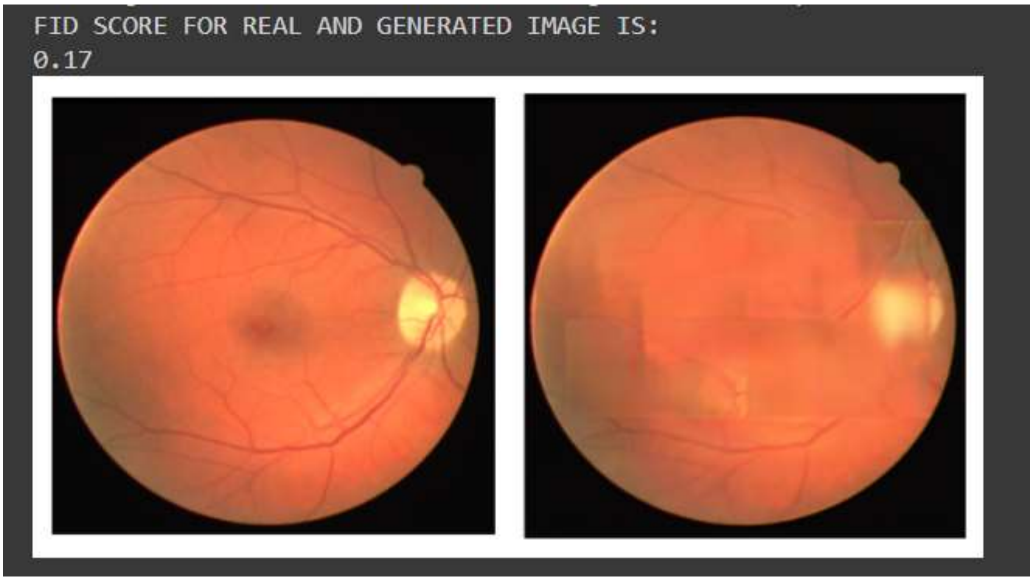


SSIM SCORE OF REAL AND GENERATED IMAGE: 0.95

For FID we got values as : **0.17 ± 0.10**



```
FID SCORE FOR REAL AND GENERATED IMAGE IS:
0.05
```



```
FID SCORE FOR REAL AND GENERATED IMAGE IS:
0.17
```

In our project analysis, we evaluated the performance of our model using two key metrics: Structural Similarity Index (SSIM) and Fréchet Inception Distance (FID). The SSIM values we obtained for our generated images were found to be 0.95 ± 0.05, indicating a high level of similarity between the generated images and the ground truth images. This suggests that our model successfully preserved important structural information during the generation process.

Furthermore, the FID values we obtained were 0.17 ± 0.10, indicating a low divergence between the distribution of our generated images and the distribution of real images. This suggests that our model was successful in generating images that closely match the statistical properties of real images. These results demonstrate the effectiveness of our approach and highlight the strong performance of our model in generating high-quality and realistic images.

8.Future Scope

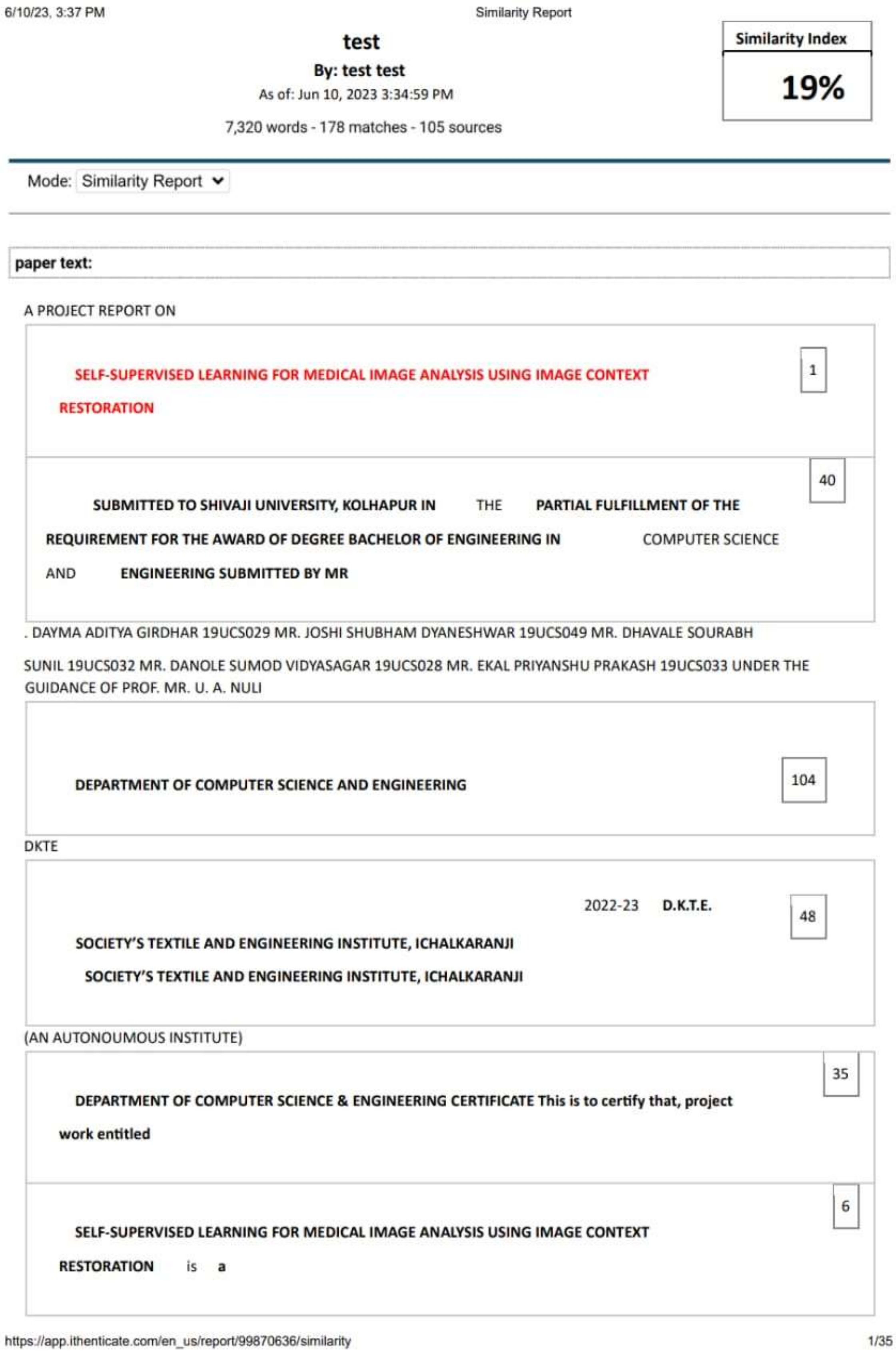
1. Proposed model is useful in the segmentation of medical images, specifically for retinal images.
2. The retinal image dataset generated can be useful for training other models.
3. Model is useful in the segmentation of medical images, Specifically for Retinal fundus images.
4. Generated Model can be used to extract blood vessels from retinal fundus images.

9.Applications

This project has a promising future scope in the field of medical image analysis. Here are some potential applications:

1. Application in various medical imaging modalities: The proposed self-supervised learning approach can be applied to different medical imaging modalities such as X-ray, MRI, CT, PET, and ultrasound, to name a few. This could lead to the development of more efficient and accurate medical image analysis tools across different modalities.
2. Incorporation of advanced deep learning techniques: The use of more advanced deep learning techniques like Self-supervised learning, attention mechanisms, and transformers could further enhance the accuracy and robustness of the proposed self-supervised learning approach.
3. Integration with other medical analysis tools: The output of the self-supervised learning algorithm can be integrated with other medical image analysis tools such as segmentation, classification, and detection algorithms, to create more comprehensive and accurate medical diagnosis systems.
4. Clinical application: The proposed approach can be applied to real-world clinical scenarios to assist radiologists in the interpretation of medical images. This could potentially lead to the development of more efficient and accurate diagnostic tools, ultimately improving patient outcomes.
5. Exploration of related research areas: The proposed approach can be extended to explore related research areas such as medical image synthesis, image registration, and multimodal image analysis.

10. Plagiarism Report



6/10/23, 3:37 PM

Similarity Report

test

By: test test

As of: Jun 10, 2023 3:34:59 PM

7,320 words - 178 matches - 105 sources

Similarity Index
19%

Mode: Similarity Report

paper text:

A PROJECT REPORT ON

SELF-SUPERVISED LEARNING FOR MEDICAL IMAGE ANALYSIS USING IMAGE CONTEXT RESTORATION 1

SUBMITTED TO SHIVAJI UNIVERSITY, KOLHAPUR IN THE **PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF DEGREE BACHELOR OF ENGINEERING IN** COMPUTER SCIENCE AND **ENGINEERING SUBMITTED BY MR** 40

. DAYMA ADITYA GIRDHAR 19UCS029 MR. JOSHI SHUBHAM DYANESHWAR 19UCS049 MR. DHAVALE SOURABH SUNIL 19UCS032 MR. DANOLE SUMOD VIDYASAGAR 19UCS028 MR. EKAL PRIYANSHU PRAKASH 19UCS033 UNDER THE GUIDANCE OF PROF. MR. U. A. NULI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING 104

DKTE

2022-23 **D.K.T.E. SOCIETY'S TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI** 48

SOCIETY'S TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI

(AN AUTONOUMOUS INSTITUTE)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING CERTIFICATE This is to certify that, project work entitled 35

SELF-SUPERVISED LEARNING FOR MEDICAL IMAGE ANALYSIS USING IMAGE CONTEXT RESTORATION is **a** 6

https://app.ithenticate.com/en_us/report/99870636/similarity

1/35

11.References

[1] Chen, L., Bentley, P., Mori, K., Misawa, K., Fujiwara, M., & Rueckert, D. (2019). Self-supervised learning for medical image analysis using image context restoration. Medical image analysis, 58, 101539.

[2] Jamaludin, A., Kadir, T., & Zisserman, A. (2017). Self-supervised learning for spinal MRIs. In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (pp. 294-302). Springer, Cham.

[3] Ericsson, L., Gouk, H., & Hospedales, T. M. (2021). How well do self-supervised models transfer? In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5414-5423).

[4] Gazda, M., Plavka, J., Gazda, J., & Drotar, P. (2021). Self-supervised deep convolutional neural network for chest X-ray classification. IEEE Access, 9, 151972-151982.

[5] Wang, F., & Liu, H. (2021). Understanding the behavior of contrastive loss. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2495-2504).

[6] Doersch, C., Gupta, A., Efros, A. A., 2015. Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1422–1430. URL https://arxiv.org/abs/1505.05192.

[7] Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al., 2015. The multimodal brain tumor image segmentation benchmark (BRATS). IEEE

Transactions on Medical Imaging 34 (10), 1993–2024. URL https://ieeexplore.ieee.org/document/6975210.

[8] Noroozi, M., Favaro, P., 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In: Proceedings of the European Conference on Computer Vision. pp. 69–84. URL https://arxiv.org/abs/1603.09246

[9] D. Kim, D. Cho, D. Yoo, and I. S. Kweon, “Learning image representations by completing damaged jigsaw puzzles,” arXiv preprint arXiv:1802.01880, 2018.