IMPROVING CLASSIFICATION PERFORMANCE OF ENDOSCOPIC IMAGES WITH GENERATIVE DATA AUGMENTATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

ÜMIT MERT ÇAĞLAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
MULTIMEDIA INFORMATICS, MODELLING AND SIMULATION DEPARTMENT

FEBRUARY 2022

Improving classification performance of endoscopic images with generative data augmentation

submitted by **ÜMIT MERT ÇAĞLAR** in partial fulfillment of the requirements for the degree of **Master of Science in Multimedia Informatics, Modelling and Simulation Department, Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin
Dean, **Graduate School of Informatics**

Assoc. Prof. Dr. Elif Sürer
Head of Department, **Multimedia Informatics, Modelling and Simulation**

Prof. Dr. Alptekin Temizel
Supervisor, **Modelling and Simulation, Middle East Technical University**

Examining Committee Members:

Prof. Dr. Alptekin Temizel
Modelling and Simulation, Middle East Technical University

Assoc. Prof. Dr. Erdem Akagündüz
Modelling and Simulation, Middle East Technical University

Prof. Dr. Çiğdem Gündüz Demir
Computer Engineering, Koç University

Date: 08.02.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Ümit Mert Çağlar

Signature :

ABSTRACT

IMPROVING CLASSIFICATION PERFORMANCE OF ENDOSCOPIC IMAGES WITH GENERATIVE DATA AUGMENTATION

Çağlar, Ümit Mert
M.S., Department of Multimedia Informatics, Modelling and Simulation
Supervisor: Prof. Dr. Alptekin Temizel

February 2022, 80 pages

The performance of a supervised deep learning model is highly dependent on the quality and variety of the images in the training dataset. In some applications, it may be impossible to obtain more images. Data augmentation methods have been proven to be successful in increasing the performance of deep learning models with limited data. Recent improvements on Generative Adversarial Networks (GAN) algorithms and structures resulted in improved image quality and diversity and made GAN training possible with limited data. The process of endoscopic imaging is essential for diseases with symptoms occurring inside the body. Medical experts use gastrointestinal endoscopic imaging to assess their patients and treat them. Ulcerative Colitis (UC) is a gastrointestinal disease where the assessment of a patient's health is done by Mayo scoring, where experts evaluate the severity of the disease symptoms. The classification of endoscopic images according to Mayo classes with deep-learning-based approaches has been studied and proven to be feasible. This thesis proposes adopting a GAN-based synthetic image generation process to increase the number of images in the dataset used by deep-learning-based methods. The results show that the classification performance of deep-learning-based approaches can be improved by 2.7% with the help of synthetic images generated by generative adversarial networks.

Keywords: Generative Data Augmentation, Generative Adversarial Networks, Image Classification

ÖZ

ÜRETKEN VERİ ÇOĞALTMA İLE ENDOSKOPİ GÖRÜNTÜLERİNDE SINIFLANDIRMA BAŞARIMININ İYİLEŞTİRİLMESİ

Çağlar, Ümit Mert
Yüksek Lisans, Çokluortam Bilişimi, Modelleme ve Simülasyon Bölümü
Tez Yöneticisi: Prof. Dr. Alptekin Temizel

Şubat 2022, 80 sayfa

Derin öğrenme temelli bilgisayarla görü modellerinin performansı eğitim sırasında kullanılan görüntülerin kalitesi ve çeşitliliğine bağlıdır. Bazı uygulamalarda yeni görüntü toplamak imkansız olabilir. Veri çoğaltma yöntemlerinin kısıtlı veri olması halinde performans iyileştirmesi sağladığı gösterilmiştir. Çekişmeli Üretici Ağlar (ÇÜA) üzerindeki güncel gelişmeler ışığında üretilen sentetik görüntülerin kalitesi ve çeşitliliği artmıştır ve kısıtlı veri ile ÇÜA eğitimi mümkün kılınmıştır. Endoskopik görüntüleme vücut içinde semptomlara neden olan hastalıkların teşhisi için önem arz etmektedir. Ülseratif Kolit (UK) doktorların gastrointestinal endoskopi yöntemi ile hastalarına teşhis ve tedavi uyguladığı bir gastrointestinal hastalıktır. UK hastalığının değerlendirilmesi için hastalık belirtilerinin şiddetine göre Mayo skorlama sistemi kullanılmaktadır. Derin öğrenme temelli yaklaşımlarla endoskopi görüntülerinin Mayo skoruna göre sınıflandırılması denenmiş ve uygulanabilir olduğu gösterilmiştir. Bu tezde ÇÜA eğitimi ile sentetik görüntüler elde edilerek, bu görüntüler ile derin öğrenme modellerinin örnek uzayı genişletilerek nihayetinde sınıflandırma performanslarında iyileştirme sağlanması önerilmektedir. Sonuçlara göre ÇÜA tarafından üretilen sentetik görüntülerle derin öğrenme temelli yaklaşımların sınıflandırma performansında %2.7 artış sağlanmıştır.

Anahtar Kelimeler: Üretici Modeller, Çekişmeli Üretici Ağlar, Görüntü Sınıflandırma

To my family

ACKNOWLEDGMENTS

I express my gratitude to Prof. Dr. Alptekin Temizel for his valuable support and supervision. During this thesis work and my graduate academic studies, he was more than a supervisor to me. His unfaltering help and motivating approach kept me dedicated and focused on my academic life.

I also want to express my gratitude to my thesis jury members for their priceless recommendations and feedback.

I want to thank Dr. Cihan Öngün and Görkem Polat for their support in our weekly meetings of the Deep Learning and Computer Vision research team. Also, for sharing their knowledge in our meetings and discussions on deep learning and generative adversarial networks.

For their never-ending support, unlimited kindness, and ultimate understanding, I would like to express my gratitude to my father Ümit Kenan Çağlar and my mother, Fatma Çağlar. I am always proud to be the son of the noblest father and the most loving mother. The exemplary virtue of my father and the eternal devotion of my mother made it possible for all of my accomplishments.

In the loving memory of my grandparents, who had always valued education and diligence, I want to express my sincere gratitude for envisioning the bright future I want to achieve.

I also want to thank my beloved wife Nilsu for her support and love. Her devotion and considerate approach made this thesis work more productive altogether.

I want to thank Turkish Aerospace and my superiors, Dr. Emre Akyılmaz and Güven Orkun Tanık, for providing me with a healthy and productive professional environment and supporting my graduate studies.

Finally, I want to express my gratitude to TUBITAK ULAKBIM High Performance and Grid Computing Center for providing us researchers with the TRUBA resource servers to run our experiments.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ v

DEDICATION vi

ACKNOWLEDGMENTS vii

TABLE OF CONTENTS viii

LIST OF TABLES xiii

LIST OF FIGURES xv

LIST OF ABBREVIATIONS xviii

CHAPTERS

1 INTRODUCTION 1

  1.1 Research Questions 3

  1.2 Motivation 3

  1.3 Contributions of the Study 3

  1.4 Organization of the Thesis 4

2 LITERATURE REVIEW 5

  2.1 Related Work 5

  2.2 Image Classification 7

2.3 Improving Image Classification Performance 8

2.4 Data Augmentation 9

2.5 Generative Data Augmentation 9

2.6 Generative Adversarial Networks 10

  2.6.1 GAN 11

  2.6.2 DCGAN 12

  2.6.3 WGAN 12

  2.6.4 ProGAN 13

  2.6.5 StyleGAN 13

  2.6.6 StyleGAN2 14

  2.6.7 StyleGAN2-ADA 14

  2.6.8 StyleGAN3 16

2.7 Use Case: Improving Classification Performance of EMS Classification of Endoscopic Images using Generative Adversarial Networks 17

  2.7.1 Endoscopy Imaging 17

  2.7.2 Ulcerative Colitis 17

  2.7.3 Endoscopic Mayo Score 18

  2.7.4 Remission Score 18

  2.7.5 Deep Learning for Image Classification 18

  2.7.6 Generative Data Augmentation to Improve Image Classification Performance . . 19

3 IMPLEMENTATION AND EVALUATION 21

3.1 Implementation 21

3.2 Experiments 22

3.3 Image Classification 22

3.4 Classification Performance Evaluation 22

3.5 Generative Adversarial Networks 24

3.5.1 GAN Performance 24

3.5.1.1 Fréchet Inception Distance (FID) Score 24

3.5.2 Synthetic Image Generation 25

3.5.2.1 Truncation 25

3.5.2.2 Random seeds 28

4 METHOD 29

4.1 Dataset 29

4.1.1 Original Dataset 29

4.1.2 Instruments and Markings in the Original Dataset 30

4.1.3 Dataset Reshaping 32

4.1.4 Subsets 34

4.2 Experimental Design 35

4.2.1 Image Classification Baseline 37

4.2.2 Improving Image Classification Performance 38

4.2.3 Class-Conditional GAN Training 38

4.2.4 Class-Specific GAN training with Weight Initialization through Transfer Learning 38

4.2.5 Saturation Experiments 39

5 GENERATIVE DATA AUGMENTATION WITH CLASS-CONDITIONAL GANS 41

5.1 Class-Conditional GAN Training 41

5.2 Generative Data Augmentation with Class Conditional GANs 41

5.3 Experiments Using the Original Dataset 42

5.4 Generative Data Augmentation Experiment Results on Subsets 43

5.5 Confidence Level Based Performance Analysis 46

5.6 Discussion on Generative Data Augmentation with Class Conditional GANs 48

6 GENERATIVE DATA AUGMENTATION WITH CLASS-SPECIFIC GAN 51

6.1 Class-Specific GAN Training 51

6.2 Generative Data Augmentation with Transfer Learning 51

6.3 Original Dataset Experimentation Results 52

6.4 Experimental Results on Subsets 53

6.4.1 Saturation Experiment 53

6.4.2 50 Images 54

6.4.3 100 Images 55

6.4.4 150 Images 56

6.4.5 200 Images 57

6.4.6 250 Images 58

6.4.7 300 Images 59

6.4.8 350 Images 60

6.4.9 400 Images 61

6.4.10 450 Images 62

6.4.11 500 Images 63

6.5 Discussion on Generative Data Augmentation with Transfer Learning Applied GANs . 64

7 CONCLUSION AND FUTURE WORK 65

7.1 Conclusion 65

7.2 Future Work 67

REFERENCES 69

A DETAILED TABLES FOR SATURATION EXPERIMENTS 73

A.1 Saturation Experiments 73

B FAILURE ANALYSIS 75

B.1 Failure Analysis 75

C ENERGY CONSUMPTION 79

C.1 Energy Consumption of the Experiments 79

LIST OF TABLES

Table 1 Two-class confusion matrix. 22

Table 2 Number of images in train, validation and test sets. 30

Table 3 The number of instruments and markings for each class in train, validation, and test sets. 31

Table 4 Reduced dataset (images with instruments removed), number of images per train validation and test sets. 32

Table 5 Dataset Image shapes and transformations applied. 33

Table 6 Best FID scores obtained with class-specific and class-conditional GAN trainings. Class-conditional GAN utilizes all classes of the training set while class-specific GAN models were trained for each class. Weights of class-specific GAN models were transferred from pre-trained FFHQ model while class-conditional GAN weights were initialized randomly. 36

Table 7 Image classification performance with 10 correct classification out of 10 experiments (100% confidence). 46

Table 8 Image classification performance with 8 correct classification out of 10 experiments (80% confidence). 47

Table 9 Image classification performance with 6 correct classification out of 10 experiments (60% confidence) 47

Table 10 Generative data augmentation performance effects with Conditional GAN. 48

Table 11 Generative data augmentation performance effects with class-specific GAN. 64

Table 12 Highest reported performance increase (%) with generative data augmentation over baseline real images only experiments. 66

Table 13 Average performance increase over baseline and subsequent subsets. GAN training methods of class-specific and class-conditional are compared 66

Table 14 The table shows classification performance details of subset saturation experiments. Each new real image addition improved overall performance metrics, with the worst performance observed at 50 images per class subset. Furthermore, the highest score was obtained by 500 images per class subset. 73

Table 15 Details about F1 and F1r (Remission) scores are shown in the table. Mean and median values, as well as the standard deviation of both scoring methods, are shown. 74

Table 16 Experimentation training costs. GAN training was done with four A100 GPUs, initial training was done on a single NVIDIA GTX3080, and all other classification model training was done on a single A100 GPU. GAN training duration was converted to a single GPU by scaling with the number of GPUs used. 80

LIST OF FIGURES

Figure 1 Example colonoscopy images with EMS ranging from 0 to 3, from left to right. . . 18

Figure 2 Synthetic images generated with the same seed for different target classes 0, 1, 2, 3 respectively, from left to right. 25

Figure 3 Endoscopic Mayo Score 0 truncation 0.0. 26

Figure 4 Endoscopic Mayo Score 0 truncation 0.5. 26

Figure 5 Endoscopic Mayo Score 0 truncation 1.0. 27

Figure 6 Endoscopic Mayo Score 0 truncation 1.5. 27

Figure 7 Endoscopic Mayo Score 0 truncation 2.0. 28

Figure 8 An example endoscopic image with original resolution. 29

Figure 9 An example endoscopic image with an instrument. 30

Figure 10 An example endoscopic image with markings. 31

Figure 11 First proposed naive pixel size reduction in all dimensions. Also the resultant shape is in rectangular form which is harder to transform into a square shape of 256×256 pixels. 33

Figure 12 Second proposed method to remove all metadata and reform circular endoscopic view into a square form. 34

Figure 13 Lost information in original images depicted as red dashed areas on all sides. Crops from each side does not effectively remove metadata, but the final shape is free from all metadata. 34

Figure 14 Best FID scores obtained from class-conditional and class-specific GAN training methods. The scores for class-specific were averaged over four classes. GAN models were trained with subsets $S_N where N \in \{50, 100, ..., 450, 500\}$. Class-conditional GAN training weights were initialized randomly. Class-specific GAN training weights were initialized with transfer learning from pre-trained StyleGAN2-ADA GAN on the FFHQ dataset. 37

Figure 15 Saturation Experiment with real images $N \in \{10, 20, ..., 530, 540\}$. The effect of classical augmentation is marginally present in the graph. 39

Figure 16 Comparison of F1 scores by addition of synthetic images with different truncation options over baseline original dataset. 42

Figure 17 Comparison of F1 score performance increase gained by addition of synthetic images over different baselines of subsets $S_N where N \in \{50, 100, 150, 200, 250\}$ real images per class. Synthetic images were generated with different truncation options of $tr \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.5, 2\}$ 43

Figure 18 Comparison of F1 score performance increase gained by addition of synthetic images over different baselines of $S_N where N \in \{300, 350, 400, 450, 500\}$ real images per class. Synthetic images were generated with different truncation options of $tr \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.5, 2\}$ 44

Figure 19 Baseline performance of F1 score of neural networks trained with only real image data. And average performance of F1 score score with generative data augmentation. . . . 45

Figure 20 Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline original dataset. Synthetic Images were generated with different truncation options. 52

Figure 21 F1 score performance of neural networks trained with only real image data. 53

Figure 22 Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 50 images per class. Synthetic Images were generated with different truncation options. 54

Figure 23 Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 100 images per class. Synthetic Images were generated with different truncation options. 55

Figure 24 Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 150 images per class. Synthetic Images were generated with different truncation options. 56

Figure 25 Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 200 images per class. Synthetic Images were generated with different truncation options. 57

Figure 26 Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 250 images per class. Synthetic Images were generated with different truncation options. 58

Figure 27 Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 300 images per class. Synthetic Images were generated with different truncation options. 59

Figure 28 Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 350 images per class. Synthetic Images were generated with different truncation options. 60

Figure 29 Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 400 images per class. Synthetic Images were generated with different truncation options. 61

Figure 30 Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 450 images per class. Synthetic Images were generated with different truncation options. 62

Figure 31 Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 500 images per class. Synthetic Images were generated with different truncation options. 63

Figure 32 EMS 0 Images incorrectly classified as EMS 1 class. 75

Figure 33 EMS 1 Images incorrectly classified as EMS 0 class. 76

Figure 34 EMS 2 Images incorrectly classified as other EMS classes. 77

Figure 35 The problematic EMS 3 Image incorrectly classified as other EMS 2 class. 77

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Network
DL	Deep Learning
DNN	Deep Neural Network
EMS	Endoscopic Mayo Score
FID	Fréchet inception distance
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Networks
GDA	Generative Data Augmentation
ML	Machine Learning
OOP	Object Oriented Programming
TL	Transfer Learning
TN	True Negative
TP	True Positive
UC	Ulcerative Colitis

CHAPTER 1

INTRODUCTION

Deep learning(DL) based computer vision methods enable various tasks such as classifying and detecting specific patterns or objects in images. Using deep-learning-based methods to classify images is a well-known and widely studied area. Supervised image classification with deep learning requires a dataset and class labels so that a network can be trained according to the supervision of the labels.

Endoscopy (internal imaging) is a widely used medical imaging technique to determine the presence, frequency, and severity of symptoms for particular diseases such as Ulcerative Colitis (UC). A widely used scoring technique for the severity of the symptoms of UC is *Endoscopic Mayo Score* (EMS) where medical doctors score colonoscopy images in a range of 0 to 3, where the severity of the disease increase with the EMS.

Ali et al. [1] concluded that DL methods could be employed to detect diseases in gastrointestinal endoscopy. As DL models are data-driven and probabilistic approaches that perform by training on data, they are particularly useful when a mathematical expression of a phenomenon is extremely hard or impossible to achieve. They are also powerful for image classification, which is impossible to formulate thoroughly. They also pointed out that recent studies have shown that Gastrointestinal diseases can be successfully identified with DL methods. Moreover, DL models for diagnosing UC disease have been actively studied for several reasons. The number of adequately labeled images and datasets is minimal, and there is a lack of experts on the classification task. Because of these reasons, it was imperative to work on this topic as increasing the classification performance of a DL model means that the quality of human lives could be improved.

EMS estimation from colonoscopy images can be handled as a classification task where deep-learning-based classification models are trained using labeled images and then fed with unlabeled images acquired from new patients to classify them with respect to the EMS.

The classification performance of deep networks depends on the quantity and quality of the data. In most cases, data is limited, especially in the medical domain. There may also be quality issues such as data imbalance, where the number of samples in each class category significantly varies. Most of the medical datasets contain more healthy samples than the disease samples. The quality of the data cannot be easily measured as quantity. The quality of a dataset affects the real-world representation power of the dataset; the samples have to follow a distribution similar to the real world.

Data augmentation is a well-known and solid way to improve the performance of DL-based image classification methods. Many studies have shown that the quantity and quality of a dataset could be improved by data augmentation where existing images are manipulated or new images are synthesized.

The manipulations such as rotation, flipping, and cropping result in unique transformations of existing images in the dataset. Generative data augmentation is a sub-branch of data augmentation where generative models are used to generate synthetic data. Adaptive Synthetic (ADASYN) [2] which is a generative approach to generate synthetic data and Synthetic Minority Over-sampling Technique (SMOTE) [3] which is another approach for synthetic oversampling to overcome imbalanced dataset problem were successfully applied in time series and numeric data.

Generative Adversarial Networks (GAN), first introduced by Goodfellow et al. [4] and later described and discussed in comparison to other generative models in detail [5] have been proven to be successful in synthetic image generation and generative data augmentation as discussed by Madhu et al. [6]. The architecture of GAN consists of two neural networks, namely generator, and discriminator. The task of each network is to compete against each other so that the generator network's output images are indistinguishable by the discriminator. This adversarial training enabled the synthesis of images similar to real images.

The performance of GAN has been improved since their first introduction, and different generator and discriminator architectures were proposed in the process. With the improvements in GAN structure and computing power, generative data augmentation in the image domain has become a reality. Yi et al. [7] discuss improvements on GANs and various applications in the medical domain, including image classification. They have also shown that GAN can generate synthetic image data to augment the original data and improve the classification performance by increasing the quantity and quality of the original image dataset.

This thesis aims to improve the classification performance of a deep-learning-based classification method by employing a GAN to generate synthetic images where GAN training is done using limited data. The hypothetical baseline for image classification is accepted to be the network trained using the existing images. The classical data augmentation methods are expected to increase this baseline work's performance, and generative data augmentation is desired to increase the classification performance further.

Several subsets of the original dataset were formed to evaluate the performance contribution of the generative data augmentation method. These subsets were created with a balanced number of images in each class. Every subset contained the previous subset's entire images and included 50 more images per class.

1.1 Research Questions

Data augmentation methods have been proven to increase the performance of deep networks with limited data. Recent advances in GAN have improved image quality and diversity with limited data. GAN can be utilized to generate synthetic images. Can the generated synthetic images be used for generative data augmentation and improve the classification performance of a deep learning model?

1.2 Motivation

The generative data augmentation method discussed in this thesis can be applied to image datasets to improve classification performance by increasing the diversity of the original dataset—the increased diversity results in better deep learning training and better model performance. The diversity of a dataset is also essential when data is limited, which is generally the case in real-world image datasets.

The use case of this thesis aims to improve the classification performance of the deep learning model tasked to classify UC disease severity according to the EMS. Even a slight improvement over the baseline performance can affect many human lives and improve the quality of their lives. As an automated classification model can be helpful for knowledge distribution among experts and help inexperienced medical doctors in classifying endoscopic images, the model's performance is of paramount importance.

1.3 Contributions of the Study

The main contribution of this study is improving the Endoscopic Mayo Score (EMS) classification of UC performance of a deep-learning-based neural network when limited amount of data is present. The contribution can be detailed in three main parts.

Firstly, the generative model training with limited data can provide specific insights about the dataset. The dataset at hand is limited, imbalanced, and contains some artefacts. Cleaning the dataset and training a GAN model is also useful for the training and fair evaluation of the image classification model.

Secondly, the image classification baseline experiments provide insights about the sufficient number of images for the training of image classification model and whether addition of new images improves the classification performance. This phenomenon is called network saturation, and the number of images that are adequate for classification performance can be used as a baseline.

Finally, the image classification performance of the deep-learning-based classification algorithm can be improved with generative data augmentation. The improvements and metrics for evaluating the performance will be analyzed in detail.

1.4 Organization of the Thesis

The thesis work consists of four main parts. The first part is data preparation, where the dataset is cleaned from artifacts, markings, and instruments, and also all images are resized to a square shape. Also, several new subsets of the original dataset were formed to be used in experiments.

The second part is baseline performance evaluation experiments using the whole dataset and partial datasets. This baseline evaluation aims to find the saturation in classification performance associated with the number of authentic images and their contribution to classification performance.

GAN Training is the third part, where partial and whole datasets were used to train conditional GANs and class-specific GANs. Conditional GANs were trained with all four classes with class knowledge, and class-specific GANs were trained over pre-trained Flickr-Faces-HQ Dataset (FFHQ) networks and transfer learning techniques were used.

The fourth part consists of experimentation intending to determine the benefits of synthetic data augmentation over baseline performance for partial datasets. Both conditional and class-specific trained GANs were used, and results were reported using only synthetic images and the performance evaluation in real images.

CHAPTER 2

LITERATURE REVIEW

In this work, Generative Adversarial Networks (GANs) were used to apply generative data augmentation for Endoscopic Mayo Score (EMS) classification of Ulcerative Colitis (UC) disease. This chapter starts with an introduction to data augmentation and generative data augmentation. Then a brief introduction to the main GAN model used throughout this thesis is given. Relative GAN algorithms, i.e., prequel and sequel models to the main GAN model, will be provided. Finally, the details of the use case of endoscopic image EMS classification for UC disease will be shared.

Generative Adversarial Networks (GANs) are used in many different applications. One such application is to synthesize images given some real examples. The synthesized images can be used to enhance the performance of deep learning methods by increasing the number of training images they can use in training.

2.1 Related Work

The performance of deep learning approaches can be improved by data augmentation, as detailed in the works of Shorten et al. [8]. Many different augmentation methods can be applied to increase the performance and make the neural networks more robust against noise. Generative data augmentation is a branch of data augmentation where the aim is to generate synthetic data similar to the original data.

Antoniou et al. [9] experimented with generative data augmentation using GANs. They reported performance improvement in different datasets of OMNIGLOT, EMNIST, and VGG-Face. They have trained conditional GANs, which use class information in the training process, and images can be generated using class conditions.

In their work, Poka et al. [10] used GAN-based data augmentation to improve the face detection performance of deep learning methods of convolutional neural networks (CNN) and Siamese networks. They have approached a different property of GANs in their work than straightforward using synthesized images along with the original data.

They have used the latent representation of face images provided by the GAN structure of StyleGAN2. They have also employed transfer learning from other face image datasets.

Frid-Adar et al. used GAN to enhance classical data augmentation further by injecting synthetic images into network training [11]. They have employed classical augmentation and counted how many augmented images were shown to the classification network.

As they had a limited amount of Computed Tomography (CT), the presence of data augmentation is crucial. They have reported an increase from 78.6% to 85.7 % in sensitivity for using generative data augmentation over classical augmentation. And an increase from 88.4% to 92.4% in specificity.

For image synthesis, they have employed different GANs for each class. The GAN structure they have used in their work was customized DCGAN.

Shin et al. [12] employed GANs to generate 3D images of brain MR images to augment and anonymize. They have used conditional GAN training for the label to image (synthesis) and image to label (classification) tasks.

Rashid et al. [13] have reportedly used GAN based data augmentation to improve the classification performance of deep learning models.

Yorioka et al. employed Auxiliary Classifier Generative Adversarial Network (ACGAN) in their work [14]. The ACGAN structure combines class conditions into generative and discriminative networks. The structure of ACGAN is an extension over conditional GAN and enables discriminators to distinguish class information along with real or fake authentication.

They have created an artificial dataset that is a subset of the original dataset of CIFAR-10 and called it a small dataset. While the original CIFAR-10 dataset contained ten classes and 6000 images per class, resulting in a total of 60000 images, the small CIFAR-10 dataset contained only 500 images per class for ten classes, resulting in a total of 5000 images.

The performance over a small dataset was evaluated as the baseline, and it was also used as the training dataset for their ACGAN. They have also artificially created a classical data augmented dataset, moreover, trained another ACGAN over this data augmented dataset. Furthermore, they applied classical data augmentation over the second trained ACGAN.

They have shown classical augmentation improved their performance metric (accuracy) from 61% to 65%. A similar improvement was absent when generative images were added as only 2% improvement was reported. The further classical data augmentation and retraining ACGAN over classical data augmented dataset decreased the classifier's performance.

The FID score performance of GAN training should not be considered as a metric of classification performance augmentation, as they have reported it as a critical remark as the analysis of their work. The quality of images generated from a GAN does not always translate to improved classification performance when used in classification network training.

Similar work was done by Mudavathu et al. priorly [15], employing Auxiliary Conditional GAN. However, this work was done on the 28 × 28 pixels sized FMNIST dataset. Furthermore, the work lacks a performance baseline of a classification network nor the proposed dataset augmentation improvements. Their work used a discriminator network to label synthetic images, and they have added synthetic images to the original dataset if synthetic images passed their discriminator's test and were labeled as real.

In their work, Yoon et al. [16] employed Generative Data Augmentation aiming to improve lesion detection performance. They have synthesized colonoscopy images and proposed to use generative data augmentation instead of traditional augmentation. They have cropped images to achieve only endoscopic view instead of full-screen view and resized cropped images to 256×256 pixel shape to make them compatible with StyleGAN2 structure. They have achieved an FID score as low as 42 in GAN training.

They have used the YOLOv3 framework as it is fast and a pre-trained Darknet53 network with transfer learning to endoscopic images as the backbone feature extractor of YOLOv3. They have also arranged a visual Turing test to determine the quality of synthetic images generated to augment the original dataset.

Their contribution to the performance of the detection algorithm is comparable with traditional augmentation in the F1 score, as both traditional augmentation and generative augmentation yielded 2-2.5% improvement from 92% to 94%. However, the generative augmentation resulted in 1.5% to 2 % improvement over traditional augmentation in sensitivity performance.

It is evident from the literature review that the Generative Adversarial Networks can be successfully used to generate synthetic images which can be used for data augmentation. It is also noteworthy that prior GAN structures provide limited data augmentation capability while GAN structures that are newer and have better image generation performance provide substantial improvement in deep learning.

Although there are some customized networks for generative networks, the main structure of dual networks of discriminator and generator are the backbone of GAN training. It is also shown in various works that StyleGAN2 architecture is extraordinarily better than previous structures in custom datasets where the amount of data is limited.

StyleGAN2-ADA is relatively and Alias-Free GAN (StyleGAN3) has not yet been released. Their presence in the literature was minimal or was completely absent in this thesis work's literature survey phase. However, experiments on StyleGAN3 were also carried out to compare its performance against StyleGAN2-ADA.

2.2 Image Classification

Image classification is an essential and fundamental task in many artificial intelligence methods such as convolutional neural networks. A classification task can be trivially set up as an algorithm that gets an input data and outputs its class. The research on classification tasks requires specific numerical metrics that outline the performance of an algorithm. There has to be a test set where the ground-truth classes are known to achieve these metrics. The performance of a classification algorithm can be measured with accuracy, precision, recall, F1 score, sensitivity, and specificity. As sensitivity and recall are identical by definition 6, the term recall will be used for consistency throughout the remainder of this thesis.

The performance of a classification algorithm can be improved by many different means. Most importantly, if the concept of the classes and perception is clear, then an algorithm can be created to fit the classification task. An algorithm can be formulated from a mathematical model that simulates the real-world system with known concepts. However, there are only a few wild guess methods and generalized filters to find patterns in the data and classify them into predetermined classes with unknown concepts.

Image classification focuses solely on image data, a two-dimensional signal with spatial information. The media of image data is generally a two-dimensional discrete (digitized) sample in the form of pixels. In image data, the spatial information of each pixel correlates with neighboring pixels. The task of image classification is a well-known task for human beings, and the human-level performance in the task is extraordinarily high compared to numerical data classification. Although the task itself is trivial for humans, there is a considerably limited knowledge on how visual perception works.

Humans can easily classify many different objects, yet how they exactly accomplish this task is a great mystery to this day. With little information on visual perception, a mathematical algorithm that can model image classification can not be formed explicitly. Generalized filters to recognize specific patterns in images have also been studied and researched thoroughly. However, if there is no knowledge about any pattern in images or if these patterns are complex and difficult to formulate, then these generalized filters tend to fail to achieve satisfactory classification performance.

2.3 Improving Image Classification Performance

The neural networks trained for image classification tasks perform better when the number of training samples increase. The advances in computation power, storage, and network capabilities resulted in higher resolution, and larger datasets and neural networks that can train on large datasets have been developed in recent years.

The performance of a classification algorithm depends on many aspects, including dataset split, hyperparameters, loss functions, optimization algorithms and training utilities. Dataset split is essential to measure how good an algorithm will perform in a real-world situation and is equally necessary to prevent a learning algorithm from overfitting on the training set. Training hyperparameters are the parameters that are not learnable and the engineer or scientist sets with intuition. Loss functions are the measurements where training, validation and test sets are assessed according to the ground truth values. Optimization algorithms optimize the deep learning model updates according to the task of minimizing the loss functions associated with the model. Training utilities include early stopping, weight decay, transfer learning, drop-out, and data augmentation.

Early stopping is a method that is employed when further training after a point in deep learning is not beneficial and stopping before reaching the hyperparameter of episode length or total epochs. Weight decay is a regularization method used to prevent overfitting by adding a predefined weight to the loss function calculation so that the exact minima for the training set are never reached; hence, it can be more generalizable. Transfer learning is employed to continue training over a pre-trained model so that previously learned low frequency or low-level features could be preserved, and the newer dataset can be employed to transfer previous networks weights over the new dataset. Drop-out is another regularization technique that is also aimed at preventing overfitting. Finally, data augmentation is a method to prevent overfitting by applying marginal or extreme data manipulations to increase the diversity of the data that a deep learning model trains on.

2.4 Data Augmentation

Data augmentation is a process where data is augmented artificially so that image classification learning improves in stability and, preferably, classification performance. It is shown that data augmentation makes Neural Networks less prone to pixel and shape memorization. Thus, it is an efficient way to overcome overfitting.

An image data can be augmented with the following methods:

- Rotations and flipping
- Mirroring
- Scaling and translations
- Changing color hue or saturation
- Cropping
- Adding noise
- Filtering
- Changing brightness or contrast
- Patch removal

The range of possible data augmentation methods is unlimited. Types of data augmentation range from simple rotation and flipping to more complex pixel shifting, padding, cutting geometrical shapes, and adding noise or filtering. Deep learning models tend to overfit to the available data when possible. The data augmentation processes explicitly attack on overfitting problem to reduce its occurrence and create more stable training runs. Various data augmentation methods are employed generally in many different deep learning models and it became a standard approach when training a deep learning model.

2.5 Generative Data Augmentation

The concept of augmenting datasets for image classification networks fits the image domain as discussed by Aggarwal et al. [17]. It is intuitive to take photographs from different perspectives, angles, and lighting conditions when forming a dataset. Similarly, it is beneficial for CNNs to observe images in a dataset from different perspectives, angles, and lighting conditions.

Data augmentation is an artificial way to augment the original dataset by applying specific manipulations to the training dataset. Even though the augmentation methods include basic image manipulations, their results improve the performance of the classification model while making the model less prone to overfitting. Shorten et al. [8] describes data augmentation as an approach that attacks the cause of overfitting as opposed to other proposed performance improvement methods.

Data imbalance occurs when a dataset contains fewer samples for a class than another class. The severity of imbalance can lead to many different problems in neural network training. The models trained on imbalanced datasets give a poorer performance for underrepresented classes. To alleviate the imbalance of classes oversampling and undersampling techniques can be used. These techniques are naive approaches that re-introduce the exact data for the training (oversampling) or do not introduce some data (undersampling).

There are also adaptive sampling methods (ADASYN) [2] and data augmentation based methods (SMOTE) [3]. Data augmentation is found to be beneficial for overcoming data imbalance. The model is enabled to observe more samples from underrepresented classes, and each sample the model observes is augmented; hence memorization and overfitting are less likely to occur, resulting in better classification performance.

The generative data augmentation method is used to increase the variation of the training dataset by injecting synthetic data into the training dataset. The models trained with the set will observe samples from a wider variety and in different forms [10] and their performance will increase thanks to generative data augmentation.

2.6 Generative Adversarial Networks

Generative models, as described by Goodfellow et al. [18] are probabilistic models where the probability distribution of specific variables can be learned from a dataset. Then the probability distribution can be used explicitly to form graphical models or implicitly to sample from a distribution.

An encoder structure, as described in [19], "encodes" the knowledge behind a set of data. It can be expressed as a funnel that distills critical points of a dataset. The result of this encoding operation can be used in many different applications. One such application is to use a decoder structure to recreate the dataset. The resultant recreated data can either be the same as the input data that has been encoded or similar to it, depending on the architecture of the encoding network.

The structure of a specialised Multi Layer Perceptron called autoencoders was introduced by Cottrell et al. [20]. The system of autoencoders can be expressed as a structure that filters the most or all of the essential features of an input dataset and can reconstruct similar or exact data. This structure stores a smaller amount of information to represent the data, and this compression mechanism can be used in many different applications.

Kingma et al. [21] proposed addition of a noise variant to the latent space of the autoencoder structure hence introducing Variational Autoencoders. The structure of variational autoencoders are same as autoencoder with the difference of enabling the addition of noise as the input to the decoder part of the autoencoder.

Goodfellow et al. introduced Generative Adversarial Networks in their 2014 work [4]. Yann LeCun, AI Scientist at Facebook and Professor at New York University, expressed his thoughts on GAN as the most exciting idea in the last ten years of Machine Learning field in an online question session [22].

From the first GAN structures, the GAN study has come so far from simple and small resolution image generation to fake video generation in high quality that it is tough to distinguish between artificial and natural. Although many improvements have been proposed for the GAN structure, the critical points of GANs are still valid up to this day. Jabbar et al. [23] describes different variants of GANs and similarities between autoencoders and Generative Adversarial Networks in detail.

The performance of how generated data are close to original data can be measured with certain expressions. A specialized network that tries to distinguish if data is generated (fake, forged) or original is called a discriminator. This network aims to learn how to discriminate original data and generated data. The output of this network can be used as feedback to the generator on how well the generator is generating data and how close or far the generated data is away from the original data.

The discriminator-generator duo networks form the basic structure of generative adversarial networks, GAN for short. GANs are trained similarly to any other neural network where initial weights and hyperparameters are set before training. Initial weights are generally randomly generated numbers, while hyperparameters are searched for best performance results. The weights of each network (generator and discriminator) are updated at each learning step. The learning action updates both networks simultaneously.

The generator aims to generate images that resemble the original data. The discriminator aims to distinguish any generated data as fake while original data is original. The performance of the discriminator is only essential to push the generator to achieve better performance. Many different metrics measure the performance of the generator. The first and foremost usage area of GANs is the image domain, where data is in a two-dimensional shape and contains spatial information in each pixel.

One general performance metric for GANs is Fréchet Inception Distance (FID), where a number of randomly generated images is measured in Fréchet Inception Distance to the original dataset. The calculation of the FID score can be done with a certain fixed number of images such as 10 thousand or 50 thousand, and the FID score will be named FID10k and FID50k. The number of original images is generally fixed as the whole dataset.

There are also many metrics to measure how well a generated image falls within the original dataset in terms of mathematical expressions. The list of these metrics and details about the most used one are given in section 3.5.

A brief chronological list starting from the first GAN to the most recent addition of StyleGAN.

1. GAN: Generative Adversarial Network introduced in June 2014 by Goodfellow et al. [4]
2. DCGAN: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks introduced in January 2016 by Radford et al. [24]
3. WGAN: Wasserstein GAN introduced in January 2017 by Arjovsky et al. [25]
4. ProGAN: Progressive Growing of GANs for Improved Quality, Stability, and Variation introduced in October 2017 by Karras et al. [26]
5. StyleGAN: A Style-Based Generator Architecture for Generative Adversarial Networks introduced in December 2018 by Karras et al. [27]
6. StyleGAN2: Analyzing and Improving the Image Quality of StyleGAN introduced in December 2019 by Karras et al. [28]
7. StyleGAN2-ADA: Training Generative Adversarial Networks with Limited Data introduced in June 2021 by Karras et al. [29]
8. StyleGAN3: Alias-Free Generative Adversarial Networks introduced in June 2021 by Karras et al. [30]

2.6.1 GAN

Generative Adversarial Networks, GAN for short, first introduced by Goodfellow et al. in 2014 [4], and had a lot of different discussions on how to improve their output performance and decrease the chronic problems on the training and training efficiency. The discussions and improvements include using new loss functions, augmentation methods, and structural changes. Although there are many different types of GANs, the literature of this thesis work will focus on some of the most used networks and networks related to StyleGAN2-ADA.

The original GAN concept includes a generative and discriminative model where the generator tries to fit the distribution of the original data and synthesize data similar to the original data. The discriminator tries to classify an image as real or synthesized. The generator model aims to minimize the log probability of the discriminator's correct classification for its synthesized data. On the other hand, the discriminator aims to maximize the log probability of correct classification of the generator model's synthetic data.

This adversarial training is named as minimax game with the value function of generative and discriminative losses. The training of GANs depends on the theoretical background of this minimax game, where a unique solution can be found to an arbitrary function expressed as in Eq. 1:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data(x)}}[logD(x)]+E_{z\sim p_{z(z)}}[log(1-D(G(z)))] \tag{1}$$

Where D and G are discriminator and generator networks, respectively. Value function, shown as V, with Discriminator (D) and Generator (G) loss functions as variables, is calculated with the log probability of discriminator identifying fake generator synthesized images and labeling real images correctly. The theoretical solution for this minimax equation can be reached when the probability of the discriminator failing to identify a fake image as fake becomes equal to the probability of the discriminator correctly identifying a real image.

2.6.2 DCGAN

The first iterations on GANs were plagued by many drawbacks such as being prone to mode collapse, requiring large datasets, vanishing gradients, being capable of only generating low-resolution images, and lacking performance metrics for GAN performance. Deep Convolutional Generative Adversarial Networks (DCGAN) has been introduced by Radford et al. [24].

DCGAN changes the native GAN struture into a convolutional structure by the following changes:

- Using convolution operation instead of pooling and omitting fully connected layers in the favor of convolution layers.
- Addition of batch normalization concept to both of the generator and discriminator networks.
- Rectified Linear Unit (ReLU) activation functions instead of sigmoids for the generator
- LeakyReLU instead of maxout activation functions for the discriminator

2.6.3 WGAN

Wasserstein GAN [25], introduces a novel loss function calculated by Wasserstein distance to GAN training. The Wasserstein distance approximates the Earth Mover Distance, while the original GAN used Jensen-Shannon divergence. The authors have shown that GAN training became more stable than the original GAN, and mode-collapsing occurrence became less frequent. With the Wasserstein metric and previously introduced DCGAN, the authors concluded that GAN training could be improved with this method, at the cost of not being able to use momentum-based optimizers.

2.6.4 ProGAN

Progrssively Growing GANs (ProGAN) is a novel approach introduced by Karras et al. [26]. The main difference over the original GAN was to include a growing structure of discriminator and generator networks. The progressive growing enables the model to start from a low resolution image and learn finer details with the addition of new layers. They have shown that this approach increases the both training speed and training stabilization. They have also introduced a feature map layer to replicate the original data set distribution over minibatches and added this new layer towards the end of the discriminator network. They have shown that this approach increased the variation.

ProGAN approached to the problem called as unhealthy competition between the discriminator and the generator networks of GAN with two concepts. In other works, batch normalization was used to elimnate this unhealthy competition. In ProGAN, an equalized learning rate for all weights was ensured that weight initialization does not result in longer convergence times. It was shown in ProGAN work that constraining signal values in the generator network using pixel-wise feature vector normalization and equalized learning rate resulted in a healthier competition between generator and discriminator.

Finally, a novel metric for GAN performance evaluation was introduced using a progressively growing Laplacian pyramid, Sliced Wasserstein Distance. This new metric was proposed instead of multi-scale statistical similarity as the latter lacks image quality assessment.

2.6.5 StyleGAN

The StyleGAN introduced by Karras et al. [27] featured a style-based generator architecture for GAN. It was based on ProGAN with novel changes for the architecture. They have applied style-transfer methodologies detailed in the works of Huang et al. [31] to the generator network.

They have introduced the following additions over the baseline ProGAN architecture:

- Using bilinear downsampling and upsampling
- Including Adaptive Instance Normalization (AdaIN)
- Adding mapping network
- Replacing traditional input layer with a learned constant tensor
- Addition of noise inputs
- Introduction of a novel mixing regularization, removing correlation between neighboring styles.

With the aforementioned additions over ProGAN the StyleGAN architecture successfully applied style-transfer methodology to the generative adversarial networks. The FID performance score of GAN trainings on benchmark datasets with StyleGAN improved over ProGAN.

2.6.6 StyleGAN2

The architecture of StyleGAN was analyzed and improved by Karras et al. [28] with the following changes:

- Using weight demodulation normalization instead of Adaptive Instance Normalization (AdaIN)
- Improvements on progressively growing technique for generator and discriminator networks
- Addition of path length regularization
- Lazy regularization

With these changes the StyleGAN2 framework achieved better performance metrics in terms of FID score, Perceptua Path length, Precision and Recall.

2.6.7 StyleGAN2-ADA

Adaptive Discriminator Augmentation (ADA) is a novel approach [29] that applies classical augmentations in a set order and frequency to the images the discriminator observes during training. The strength of each augmentation was learned during training according to the novel overfitting metric applied to the discriminator. This approach makes the highest performing classical augmentations stronger in discriminator image evaluation. This way, the augmentations in pixel blitting, general geometric transformations, color transformations, filtering, and corruptions were the five categories containing 18 classical augmentation methods. Pixel blitting in the form of:

- Cross flipping
- $90\deg$ rotations
- Integer translation

General geometric transformations in the following ways:

- Isotropic scaling
- Arbitrary rotation
- Anisotropic scaling
- Fractional translation

Color transformations with the following augmentations:

- Brightness
- Contrast
- Luma flip
- Hue rotation
- Saturation

Image-space filtering in the following specific frequency bands of:

- $[0, \frac{\pi}{8}]$
- $[\frac{\pi}{8}, \frac{\pi}{4}]$
- $[\frac{\pi}{4}, \frac{\pi}{2}]$
- $[\frac{\pi}{2}, \pi]$

Image-space corruptions with:

- Additive RGB noise
- Cutout

At the start of training, the discriminator is fed with images synthesized by the generator through classical augmentations. The augmentation probabilities are started to be equal at first. Through training, the StyleGAN2-ADA framework re-arranges the probability of each classical augmentation. This adaptive augmentation improves discriminator performance and enables GAN training with fewer images.

With StyleGAN2-ADA architecture, the convergence of GAN training was made possible with smaller datasets, and the problem of overfitting was resolved. Without ADA, the discriminator focused on particular features while the generator could easily fool the discriminator with those features while generating non-realistic images. However, with ADA, the augmentations enable the discriminator to maintain the focus on the whole image throughout training, removing the discriminator overfitting probability.

ADA improves GAN training performance, decreases the number of required images for training, and increases the synthetic image quality of GANs. They have also included mixed-precision support which directly increases the training speed and reducing memory usage as 16-bit and 32-bit floating point operations could be used together. Authors had also provided better hyperparameter defaults for different types of datasets as a result of their experiments. They had cleaned the codebase over StyleGAN2, included standalone versions of high quality image augmentations in GPU and enabled network import resulting in faster loading times. Because of these improvements, the StyleGAN2-ADA structure was selected to be used in this thesis work.

Furthermore, the official implementation provided by Nvidia in their curated Github repository was ported into the PyTorch framework. The benefits of the PyTorch framework are discussed in section 3.1. Moreover, the official implementation in the PyTorch framework was reportedly 5-30% faster in training speed compared to the TensorFlow implementation. The inference speed was also increased by up to 35%, resulting in faster image generation. The addition of new command-line options also made the operation more flexible. It was also stated that GPU memory usage was similar to the TensorFlow implementation. For the compatibility part, the dataset tool was updated to support PNG and Zip-based datasets, as the dataset used in this thesis was formed in a PNG and later in a zipped format; this update was a positive addition. With many improvements over the previous implementation, the StyleGAN2-ADA-PyTorch was selected to be the main framework to train GANs and generate synthetic images.

2.6.8 StyleGAN3

Alias-Free GAN or StyleGAN3 is the latest edition of the StyleGAN family, developed and maintained by NVIDIA. The proposed improvements over the StyleGAN2-ADA structure are exclusively in signal processing nature. The authors have employed signal processing techniques to overcome aliasing occurring in progressive layers. They have shown that the nature of generating images translates input noise signal into high-frequency noise in final output images by amplifying the input noise in each layer until the final high-resolution image.

They have also proved that applying low pass filters decreases the aliasing problem of the output images. Low-pass filters filter out high-frequency components, i.e., enabling low-frequency data while suppressing high-frequency noise. It was discussed in their work that the aliasing problem is a predominant one that reduces translation and rotation invariance, which corresponds to unnatural effects when images were moved (translation) or rotated. These undesired effects limit the possibility of video generation as images tend to move and frequently rotate in videos.

The theoretical improvements discussed in their work were focused more on transformation and rotation which occurs natively in video domain, image quality on the other hand, did not improve much in terms of FID score compared with previous framework improvements. Moreover, the improvements reduced the training speed of the previous architecture. The expense of increased training time at no marginal benefits in terms of FID score or image quality was the primary reason why this thesis work was not re-done on StyleGAN3 architecture.

2.7 Use Case: Improving Classification Performance of EMS Classification of Endoscopic Images using Generative Adversarial Networks

2.7.1 Endoscopy Imaging

Endoscopy means looking inside the body with a tool called an endoscope. Endoscopy imaging is an essential and necessary way to diagnose and monitor certain diseases when they are present inside the human body. Gastrointestinal diseases such as Ulcerative Colitis show symptoms in the large intestine and only be diagnosed through an expert medical view of the large intestine. The tool used in colonoscopies has an endoscopic imaging instrument and cleaning and sample extraction capabilities.

Medical doctors can extract samples during the endoscopic examination. They can also intervene directly by removing polyps (cancerous tissues), cleaning wounds, and taking reference images to discuss with colleagues. The classification of specific images according to a scoring system called *Endoscopic Mayo Score* (EMS) helps medical doctors to treat their patients.

Polat et al. [32] have shown that polyp detection can be accomplished by using the deep learning method. The performance of a deep learning method can be improved by contemporary means, as discussed in their work. One way of improving the performance of deep learning approaches is using bootstrap aggregating or bagging, where multiple deep learning methods are bagged together, and their predictions are combined. It is shown that with the combined power of multiple deep learning methods, the performance of a classification task can be improved.

Gastrointestinal Endoscopy imaging focuses on two parts of the human intestine to diagnose and treat diseases. The upper and lower large intestine have different characteristics. Diseases occurring in either part are generally diagnosed by gastrointestinal endoscopy imaging.

The datasets in the medical domain have the following shortcomings. First, the data is highly sensitive as it is obtained from patients and medical doctors, and their patients value privacy. Second, the data is precious and yet scarce because of the lack of digitization and large databases and data warehouses as opposed to other image datasets. Thirdly, the datasets in the medical domain are highly imbalanced, i.e., disease-diagnosed imagery is under-represented while normal/healthy images are over-represented.

Endoscopy imaging is no exception for the general rules that medical image datasets follow. The use case dataset is a highly sensitive dataset obtained through many different endoscopic examinations of patients. The medical doctors that obtained and created this dataset have also labeled the imagery for Ulcerative Colitis (UC) symptoms.

2.7.2 Ulcerative Colitis

Ulcerative Colitis (UC) is an Inflammatory Bowel Disease (IBD) occurring in the lower gastrointestinal tract. Its symptoms are certain inflammations occurring depending on the severity of the disease. As exemplified in Fig. 1, it is possible to assess disease symptoms by visual inspection of endoscopic images.

2.7.3 Endoscopic Mayo Score

Medical experts classify Ulcerative Colitis with Endoscopic Mayo Score (EMS) ranging from 0 to 3 with 0 identifying a healthy person where 1,2 and 3 are showing incrementally increasing disease activity. The dataset in the use case scenario has four classes that correspond to the severity of the disease. Deep learning methods used in this thesis focus on EMS estimation by image classification. As shown in Fig. 1, the symptoms are increasing in frequency and severity with increasing EMS.

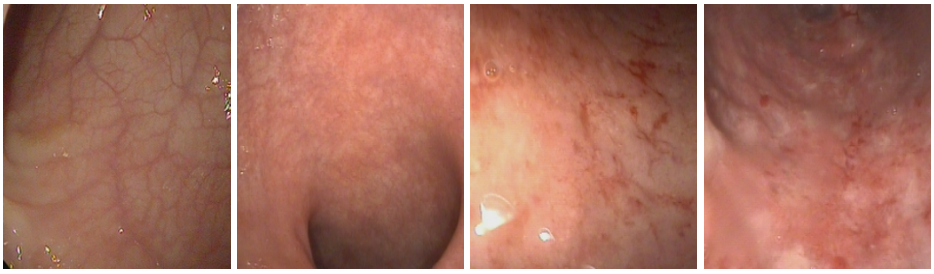

Figure 1: Example colonoscopy images with EMS ranging from 0 to 3, from left to right.

2.7.4 Remission Score

Labeling biases, uncertainty, and subjective scoring result in many discrepancies between disease classification as there is no specific way to distinguish the severity of diseases. Remission classification can be employed to overcome the issues associated with labeling. Remission, by definition, means healthy and without any symptoms. Mayo-0 and Mayo-1 classes, i.e., healthy or only mildly serious cases, are bundled together, and Mayo-2 and Mayo-3 classes that correspond to severe and most severe cases are wrapped together. The performance of image classification algorithms has been obtained for both 4 class classification and binary classification, i.e., remission score.

2.7.5 Deep Learning for Image Classification

Polat et al. [33] discussed that Ulcerative Colitis EMS level classification could be treated as a regression task because the EMS level is ordinal as the EMS level increases with severity. They have treated EMS level as ordinal, introduced a novel loss function, and trained ResNet18, Inception-V3, and MobileNet-V3-Large networks. They have shown that the addition of the novel loss function improved the results for all models. They have also shown that the results for these different networks were in the range of 0.6 percentage points, demonstrating that either of these networks can be used effectively with very similar results.

This thesis utilized deep learning methods to classify endoscopic images, specifically colonoscopy images, to estimate the Endoscopic Mayo Score (EMS) of Ulcerative Colitis (UC). The deep learning method that was used is Residual Network with 18 layers of depth (ResNet18). The performance evaluation of ResNet18 models was done with 4-class classification metrics of accuracy, precision, recall, F1 score, sensitivity, and specificity. The remission score was also used, and binary classification metrics of accuracy, precision, recall, F1 score, sensitivity, and specificity were used.

The classification performance of ResNet18 was thoroughly examined, and comparisons and conclusions were based on the classification performance. The detrimental effects, as well as improvements, were studied, and possible reasons for these effects were included as comments.

2.7.6 Generative Data Augmentation to Improve Image Classification Performance

Generative data augmentation is a subbranch of data augmentation where generative models are used to diversify the original dataset. With the help of StyleGAN2-ADA, synthetic images were generated. The synthetic images were then used in conjunction with the original dataset to determine the effects of generative data augmentation.

CHAPTER 3

IMPLEMENTATION AND EVALUATION

The implementation details and evaluation metrics used in the thesis are detailed in this chapter. The implementation details include the programming language, machine learning framework, version control system, experiment tracking tool, and high-performance computing capabilities.

The evaluation metrics used in the thesis include image classification performance metrics for binary and 4-class classification, generative adversarial network output quality evaluation, and respective calculation details of non-trivial metrics.

With the experiment tool's help, the thesis has been realized in a manner of organized, traceable, and reproducible experiments. The experiments have been realized in parallel with a high-performance computing suite. The programming language and machine learning framework had a high level of cohesion, facilitating an easier debugging and development environment.

3.1 Implementation

This thesis has been realized with Python programming language[34]. Python is a high-level scripting programming language that is easy to read, debug and interpret. Python programmers have agreed on certain conventions to follow and obey when writing in Python. These simple conventions help with the transparency and readability of the code by following Python's unique design. Thanks to these conventions, Python code writing and reading have become more accessible. These conventions define a coding style which is generally referred to as Pythonic. These conventions can be found inside Python Enhancement Proposals (PEPs) [35].

On the other hand, while coding with Python is faster than low-level programming languages, its run-time performance is relatively lower than low-level programming languages like C or C++. Although Python has an interpreter that interprets and runs Python scripts, specific frameworks have been developed to match run-time efficiency of low-level languages. Object-Oriented-Programming (OOP) is an option in Python, and this thesis work followed both Pythonic programming guidelines and OOP methodology.

Implementation of experiments in the thesis was done in PyTorch framework [36]. The pyTorch framework enables Pythonic programming capabilities for the most well-known and established machine learning frameworks. The generative data augmentation component of the thesis was the official

StyleGAN2-ADA Pytorch version [37]. The image classification component of the thesis for EMS estimation of UC was also written in the PyTorch framework following the OOP methodology.

3.2 Experiments

Weights and Biases (WANDB) framework [38] has been used for experiment tracking. 72 GAN models (StyleGAN2-ADA) and 36096 UC EMS classification models (Resnet18) have been trained in total throughout this thesis work. The resource cost of this thesis is detailed in C.1.

3.3 Image Classification

In this work, residual networks [39] trained on image datasets have performed image classification tasks. ResNet18 has been used as the base network as it is expected to converge faster with comparable accuracy with same depth networks.

Classification performance was measured with smaller datasets to find how many images are required to saturate the classification network.

The framework for image classification was based on the works of Kani et al. [40]. The dataset preparation, dataset cleaning, ensemble label evaluation with experts (medical doctors), Python implementation of image classification framework, classification performance evaluation, and baseline WANDB integration were based on Görkem Polat's Ph.D. works which are not yet published.

3.4 Classification Performance Evaluation

The research on classification tasks requires specific numerical metrics that outline the performance of an algorithm using a test set where the ground-truth classes are known. The metrics used in classification tasks are shown in Table 1 as described in detail by Alpaydin [41].

Table 1: Two-class confusion matrix.

		Predicted Values	
		Positive	Negative
Ground Truth	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

- True Positive (TP) is the correct classification as a class that was indeed that class.
- True Negative (TN) measures how well an algorithm does not predict input data as a class that was indeed not that class.
- False Positive (FP), where the algorithm performs an incorrect classification and classifies an input data as a class; however, it was not that class.
- False Negative (FN), where the algorithm classifies an input data incorrectly as not a class; however, the ground-truth states it was that class.

The performance of a machine learning algorithm in a classification task can be measured with accuracy, precision, recall, F1 Score, sensitivity and specificity as detailed by Kubat [42].

1. Accuracy is the total number of correct classifications divided by all predictions—the metric of a classification algorithm's accuracy is calculated in Eq. 2

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

2. Precision, where the total number of correct classifications is divided by both correct classifications and incorrect classifications. The metric of how precise a classification algorithm is. Calculated as in Eq. 3

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

3. Recall where the total number of correct classifications is divided by all elements in a class. It is the metric of how many correct classifications were made in classifying a specific class. Calculated as in Eq. 4

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

4. F1 score is calculated by taking the harmonic mean of precision and recall. The harmonic mean of precision and recall is not as sensitive to data imbalance as accuracy. F1 score is calculated as in Eq. 5

$$Precision = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5}$$

5. The sensitivity, which is identical to recall by definition, is the ratio of correct classifications to all elements in the class. The metric of sensitivity of a classification algorithm is calculated as in Eq. 6

$$Sensitivity = \frac{TP}{TP + FN} \tag{6}$$

6. Specificity is the ratio of true negative to true negative and false positive combined (out of class elements). As sensitivity and recall are identical by definition, the term recall will be used to cover both of the terms. The metric of specificity of a classification algorithm is calculated as in Eq. 7

$$Specificity = \frac{TN}{TN + FP} \tag{7}$$

3.5 Generative Adversarial Networks

Performance evaluation of GANs can be performed with the following metrics, although there were many different metrics, the most used and accepted metric was the Fréchet Inception Distance metric introduced by Heusel et al. [43].

- Average Log-likelihood
- Coverage Metric
- Inception Score (IS)
- Modified Inception Score (m-IS)
- Mode Score
- Activation Maximization (AM) Score
- Frechet Inception Distance (FID)
- Maximum Mean Discrepancy (MMD)
- The Wasserstein Critic
- Birthday Paradox Test
- Classifier Two-sample Tests (C2ST)
- Classification Performance
- Boundary Distortion
- Number of Statistically-Different Bins (NDB)
- Image Retrieval Performance
- Generative Adversarial Metric (GAM)
- Tournament Win Rate and Skill Rating
- Normalized Relative Discriminative Score (NRDS)
- Adversarial Accuracy and Adversarial Divergence
- Geometry Score
- Reconstruction Error
- Image Quality Measures (SSIM, PSNR and Sharpness Difference)
- Low-level Image Statistics
- Precision, Recall and F1 Score

3.5.1 GAN Performance

The output resolution of GANs is desired to match the input resolution of the dataset that GANs are trained with. However, the quality of images differs a lot depending on subjective perception. As manual GAN output evaluation is inefficient and not objective, there has to be a numerical metric that can be used to measure the quality of GAN outputs.

The outputs of GAN models should resemble input dataset features, a GAN trained on images of a specific class is expected to generate synthetic images in that class. On the other hand, the features of the outputs can and will differ from the original images. Fréchet Distance is a distance metric associated between two distributions and can be used to calculate Fréchet Inception Distance (FID) and evaluate the quality of images generated using GANs.

3.5.1.1 Fréchet Inception Distance (FID) Score

Fréchet Inception Distance, which is an advanced metric using Wasserstein Distance introduced by Vaserstein[44], is calculated by the following steps:

- A neural network (generally Inception V3) is trained on the ImageNet dataset.
- The trained neural network is fed the original and synthetic images.
- The mean and variance of neural network activations for original and synthetic images are obtained.
- The distribution of original and synthetic images are assumed to be Gaussian.

- The Wasserstein Distance is then calculated from two distributions.
- The square of this distance is called the FID score.

FID score is calculated by the distance between two distributions that arise from deeper layers of a neural network; hence the distance is more "conceptual" than raw pixel distance. However, as a neural network is used to calculate this score, the FID score calculation is costly.

FID score is used to determine the quality of GAN outputs. FID score is calculated as in the Equation 8:

$$\mathrm{FID} = ||\mu - \mu_w||_2^2 + \mathrm{tr}(\Sigma + \Sigma_w - 2(\Sigma^{1/2}\Sigma_w\Sigma^{1/2})^{1/2}). \tag{8}$$

3.5.2 Synthetic Image Generation

GAN models synthesized images for all four classes, i.e., Endoscopic Mayo Score of 0 to 3. The synthesized images were built from random noise components that turn into colonoscopy images when the random noise is given to the generator network. The generator network was the only part of the GAN used in image generation. Synthetic images generated with the same seed are shown in Fig. 2. The effects of UC is clearly shown with increasing inflammations in the image as expected from EMS.

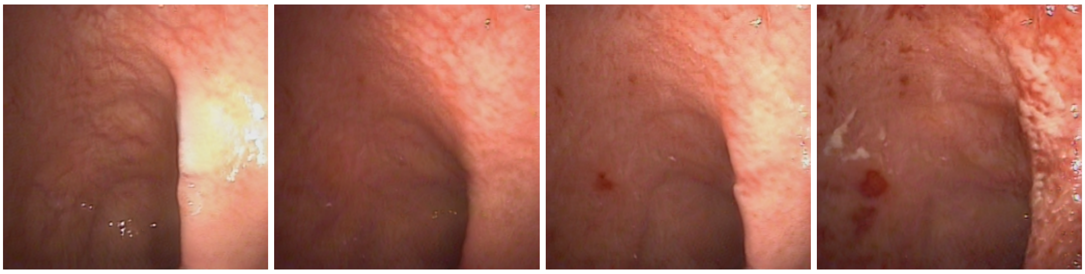

Figure 2: Synthetic images generated with the same seed for different target classes 0, 1, 2, 3 respectively, from left to right.

3.5.2.1 Truncation

When generating synthetic images, the truncation is the main parameter to choose. The truncation parameter determines the range of outputs. The truncation value of 0 means that all outputs will be equal to the dataset's mean; hence, minimal truncation values will not result in different images. The values from 0 to 1 determine the output in the original dataset range. The truncation value of 1 means the outputs will cover the whole dataset diversity. The values below one truncation were used to generate synthetic images that fall into the original dataset distribution.

The values above 1 mean that the output samples will be more extensive than the original dataset distribution. To increase the diversity of the original dataset, values above one were also used to generate synthetic images. Both in-range and out-of-range of the original distribution truncation values were used in the experiments, and their results were analyzed and compared.

As can be observed from the following images in Fig. 3, the truncation=0 parameter results in exact same outputs. The value of zero for the truncation option resulted all synthetic images to be drawn from the exact same noise input that was the mean of the original image data distribution.

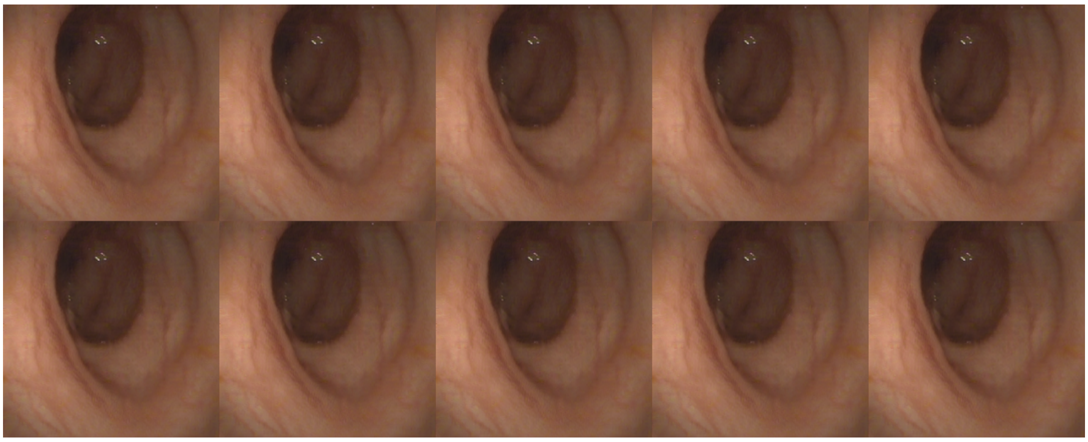

Figure 3: Endoscopic Mayo Score 0 truncation 0.0.

A truncation value of 0.5 can increase the diversity of the synthetic image outputs as shown in Fig. 4. Compared to the truncation value of 0, the images vary for each random noise.

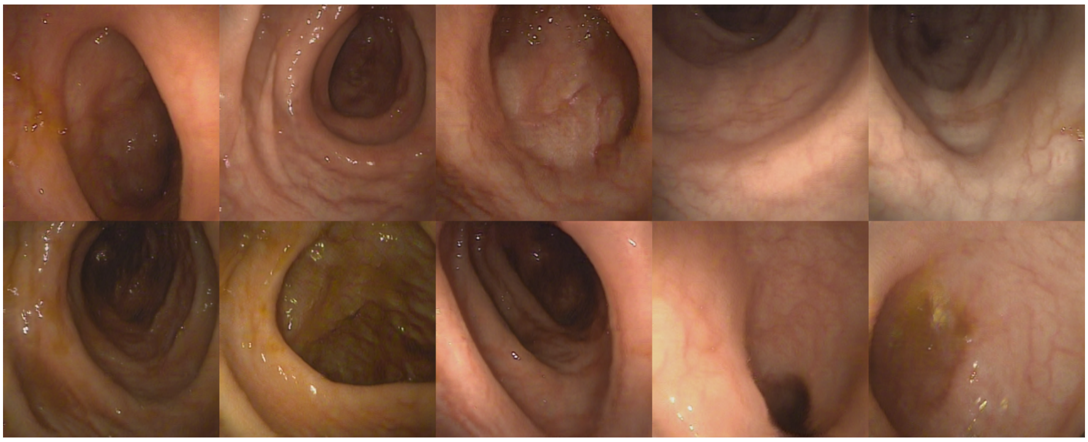

Figure 4: Endoscopic Mayo Score 0 truncation 0.5.

The truncation option of 1.0 is the option where the variance of the original dataset distribution is used exactly. The images are shown in Fig. 5

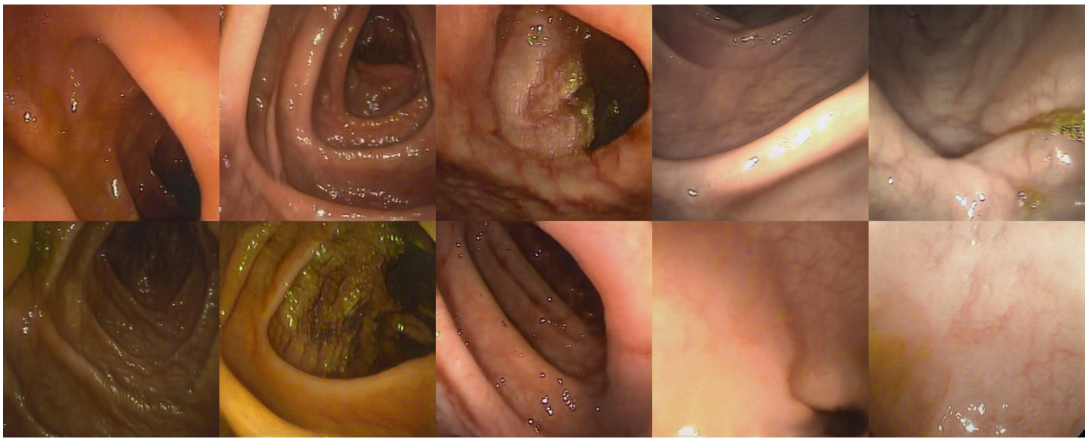

Figure 5: Endoscopic Mayo Score 0 truncation 1.0.

Truncation option of 1.5 produced the images in Fig. 6.

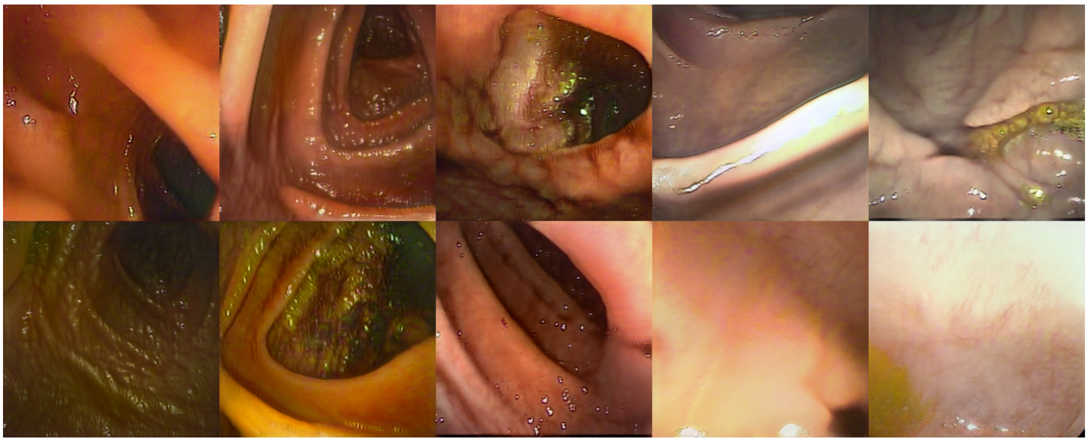

Figure 6: Endoscopic Mayo Score 0 truncation 1.5.

The images in Fig. 7 were produced with the truncation option of 2.0.

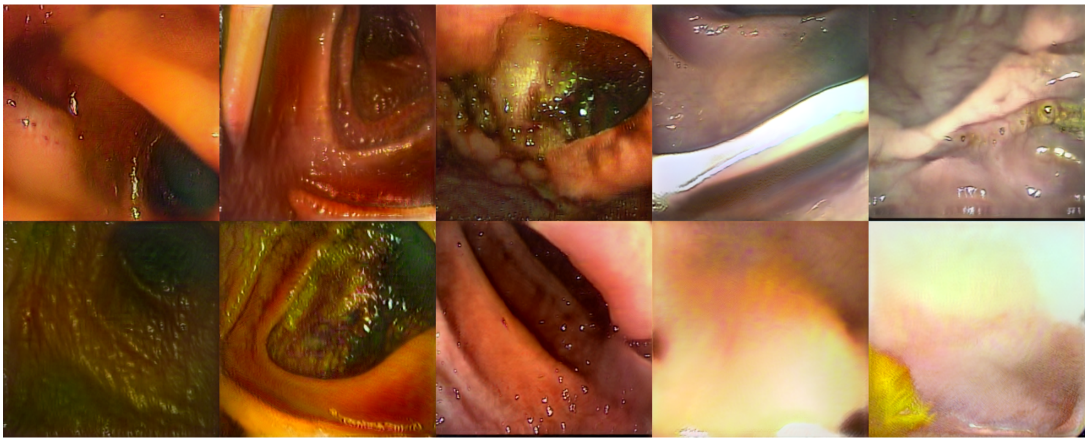

Figure 7: Endoscopic Mayo Score 0 truncation 2.0.

As the truncation value increase, the variance of the distribution of the noisy input used to initiate the process of synthetic image generation increases. The value of 1.0 is the point where the inputs of the generator network has the same variance of the original dataset. Values above 1.0 can be used to generate even more unique in the sense that their initial noise can be outside of the original data distribution.

3.5.2.2 Random seeds

Random seeds are used to allow reproducibility of the same experiments at different times. However, the random number generators in Python language are pseudo-random number generators that use hardware and software-specific components. The random number generators can be given seeds to return the same results at different runs.

Although it is intuitive to seed random number generators and obtain the same results on different experiments, it is impossible to determine whether the numbers generated will result in an outlier in the result distribution. Thus n-fold training approach was selected to evaluate the performance of the neural networks where n is generally 20.

Although this approach increased the required training, the results became statistically more readable. Statistical values such as mean, median, and standard deviation became available and have been considered in performance evaluation and outlier detection.

StyleGAN2-ADA-PyTorch and StyleGAN3 frameworks contain synthetic image generation components that use random seeds to generate input noise. The input noises then turn into synthetic images using the generator network of GANs. The same set of seeds hence the same noise inputs, have been used in all image generation processes.

CHAPTER 4

METHOD

4.1 Dataset

4.1.1 Original Dataset

The images obtained through a number of gastrointestinal endoscopic imaging were formed into a dataset of 19757 images. Three medical doctors jointly provided the EMS for each image, resulting in a labeled dataset of 11276 images.

The original images have the following properties:

1. Images have a rectangular shape with 352 pixels in width and 288 pixels in height.
2. There are metadata printed on the image itself containing date-time and patient ID data.
3. There is also a black zone outside the circular endoscopic imagery area due to fish-eye lenses of the imaging instrument.
4. Specific images have markings such as arrows or circles pointing out extraordinary conditions or symptoms.

An example image sampled from the original dataset can be seen in Fig. 8.

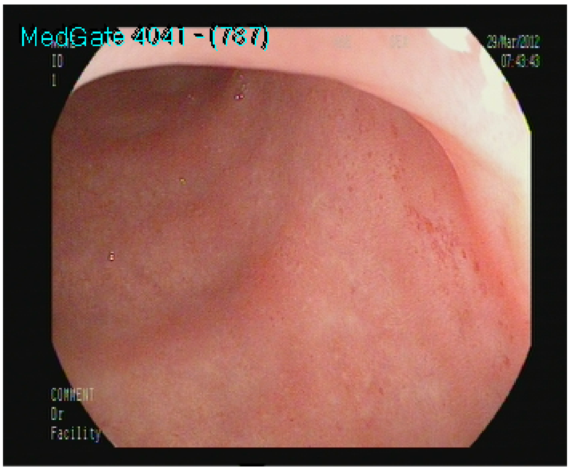

Figure 8: An example endoscopic image with original resolution.

The original dataset class quantities are presented in Table 2. The percentage of the train, validation, and test sets for the original data are 69.7%, 15.9%, and 14.4%, respectively. The intended split of train, validation, and test sets was 70%, 15%, and 15%. However, a patient-level split was also applied so that all images from a patient were grouped in either train, validation or test set.

Table 2: Number of images in train, validation and test sets.

Classes	**Train**	**Validation**	**Test**	**Dataset**
Mayo 0	4247	1002	856	6105
Mayo 1	2097	492	463	3052
Mayo 2	914	168	172	1254
Mayo 3	607	126	132	865
Total:	7865	1788	1623	11276

4.1.2 Instruments and Markings in the Original Dataset

The dataset used to validate theoretical approaches contains images having 352×288 pixels. The original dataset is imbalanced in terms of different classes. The original dataset also contained some images with instruments and markings. An example of the aforementioned image that contains an instrument can be seen in Fig. 9.

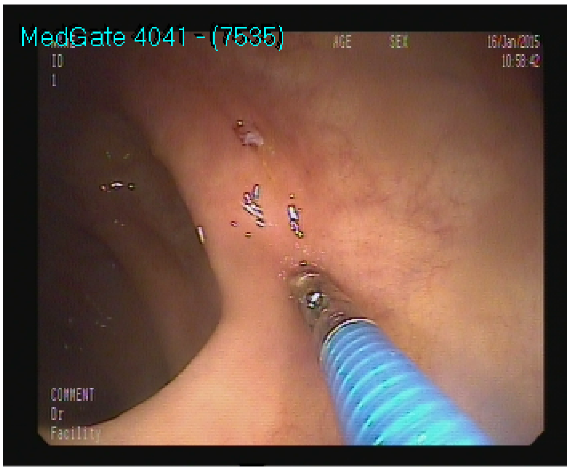

Figure 9: An example endoscopic image with an instrument.

The original dataset also contains markings as exemplified in Fig. 10. Medical doctors use these markings as a reference on a specific part of an image. These markings occur in the shape of arrows or circles pointing out exciting occurrences.

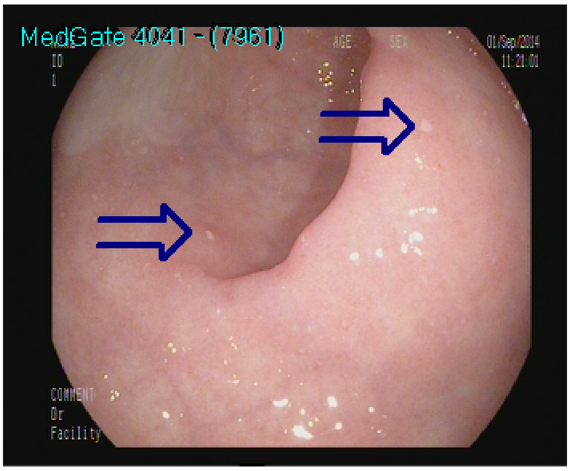

Figure 10: An example endoscopic image with markings.

The presence of these instruments and markings can create undesired biases both in GAN models and classification models. Instruments can create an additional difficulty for generator networks of GAN models to synthesize fake images. This hardship could also result in discriminator networks to focus more on instruments which generator networks fail to imitate. If specific classes contains more images with instruments then a bias would occur towards that class. The classification models can also be affected by the presence of these instruments and markings if there is a bias associated with the presence of the instruments and markings.

Table 3: The number of instruments and markings for each class in train, validation, and test sets.

Classes	Train	Validation	Test	Dataset
Mayo 0	365	61	84	510
Mayo 1	173	44	57	274
Mayo 2	82	17	26	125
Mayo 3	65	5	13	83
Total:	685	127	180	992

The instruments were found to be present in all classes as shown in Table 3. Images with instrumentation and markings do not have a similar distribution as the original dataset. The ratio of the presence of instrumentation and markings in images per class was 8.6% for Mayo-0, 9% for Mayo-1, 10% for Mayo-2, and finally 9.6% for Mayo-3. As the ratio of the presence of instruments and markings increases with increasing EMS, it can be concluded that the presence of them might create a bias towards higher Mayo scores. According to the increased frequency of markings and instrumentation with increased severity of the UC disease, a positive correlation between images with markings and higher UC class images, such as Mayo 3, is valid.

Due to the positive correlation between markings and instrumentation and disease severity, the removal of images containing markings and instrumentation was concluded to be beneficial for both GAN training and classification network training.

Polat et al. [45] concluded that artefacts, e.g., very bright shining spots, can cause a decrease in classification performance. As a correlation between markings and instruments and Endoscopic Mayo Score is present, then the classification network will learn to classify these markings rather than symptoms. In this thesis work, instruments, artificial markings, and imaging artefacts were all treated as undesired properties of endoscopic images; therefore, the removal of endoscopic images containing instruments and markings was performed.

Table 4: Reduced dataset (images with instruments removed), number of images per train validation and test sets.

Classes	Train	Validation	Test	Dataset
Mayo 0	3882	941	772	5595
Mayo 1	1924	448	406	2778
Mayo 2	832	151	146	1129
Mayo 3	542	121	119	782
Total:	7180	1661	1443	10284

After the removal of images containing instruments, the resultant dataset called reduced dataset had the following number of samples per class as shown in Table 4. The original distribution of 69.7%, 15.9%, 14.4% for train validation and test sets respectively changed into 69.8%, 16.2%, 14.0% after images with instruments were excluded from the dataset. It is important to note that the original distribution has been affected by only 0.4% at maximum by this instrumentation and marking removal process. It was concluded that the resultant reduced dataset contained the necessary information for the classification model training.

4.1.3 Dataset Reshaping

The original images have various metadata embedded in the image itself. The patient ID on the top left corner, data time information on the top right corner and comments on the bottom left corner were present in all images. Furthermore, they obscured specific pixels in the original images. Moreover, endoscopy images have an endoscopic view with a circular shape. The original images contained black zones on each corner and side. The presence of black zone and metadata on images could cause problems with deep learning methods in EMS classification in colonoscopy images.

Also, the black zone and metadata information can be misinterpreted as discussed before. GANs trained on the original dataset can either focus too extensively on the synthetic generation of text-based metadata, or the discriminator can cheat by detecting fake text; therefore, the original images of the shape 352×288 have been cropped to 224×224 shape to exclude black zone and metadata, and reshaped to 256×256 as shown in Table 5.

Table 5: Dataset Image shapes and transformations applied.

Property	**Original Image**	**Cropped**	**Resized**
Width	352	224 →68 60←	256
Height	288	224 ↓50 14↑	256
Aspect Ratio	1:0.875	1:1	1:1

Two approaches can be used to remove metadata from original images. The first way is naive cropping from all sides (left, right, top and bottom) to remove writings. The problem of applying this method is that a large area of the original image in the endoscopic view is lost due to this process, as shown in Figure 11.

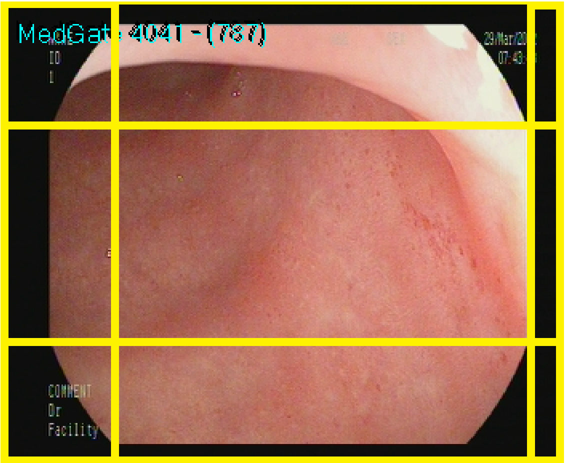

Figure 11: First proposed naive pixel size reduction in all dimensions. Also the resultant shape is in rectangular form which is harder to transform into a square shape of 256 × 256 pixels.

The second proposed method is again a cropping method; however, instead of cropping from the sides that remove all metadata, the calculation was made that the corners of the resultant cropped image would remove all the metadata. Each crop on each side does not effectively remove all metadata; however, the resultant crops from all sides remove all metadata. Moreover, the resultant shape as shown in 12 is square instead of a rectangle.

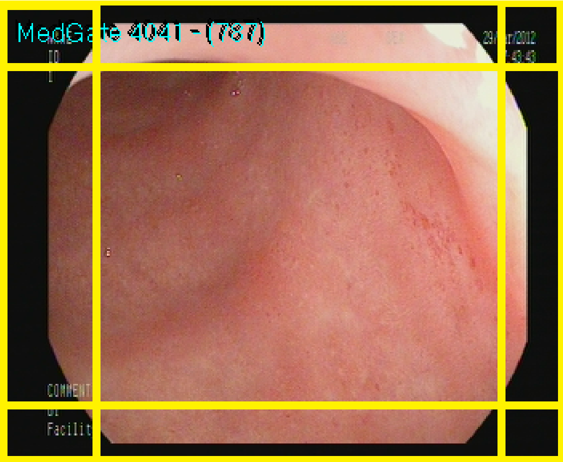

Figure 12: Second proposed method to remove all metadata and reform circular endoscopic view into a square form.

The overall information loss can be observed from Figure 13. The lost information area is minimal with this method, and the resultant shape is 224 × 224, which has a 1:1 aspect ratio. Finally, bicubic interpolation was used, without losing the original aspect ratio, 224 × 224 square has been reshaped into 256 × 256. Thus it is safe to state that there is no distortion due to different aspect ratios.

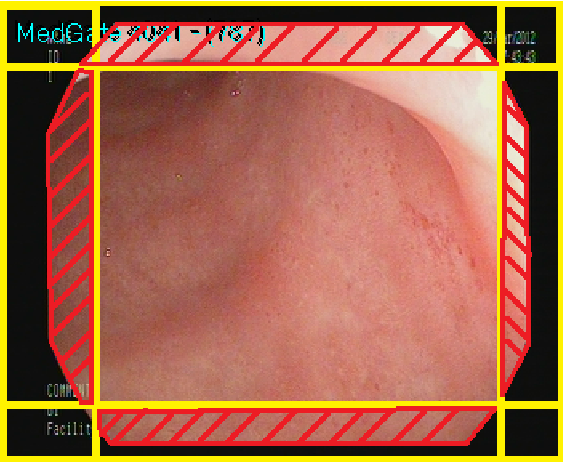

Figure 13: Lost information in original images depicted as red dashed areas on all sides. Crops from each side does not effectively remove metadata, but the final shape is free from all metadata.

4.1.4 Subsets

Datasets with a fixed number of image samples in each class were generated to form subsets. The subsets were always formed using the training dataset. Validation and test datasets remained as original reduced (images with instruments and markings removed) dataset version.

Subsets have been formed out of the original dataset to perform analysis on the benefits of generative data augmentation for the performance of the classification method. For each subset, baselines were determined using only the original images. The contribution of generative data augmentation is evaluated using synthetic images over baseline subsets.

The subsets were generated by the following rules: First, the images in the original dataset was randomly shuffled. Second, the images was used to create subsets, with each subset containing exact images of the previous smaller subset. This way, each subset represents the addition of new real images over the previous subset; hence performance evaluation could be performed both with the addition of real and synthetic images.

4.2 Experimental Design

The focus of this thesis was to find whether addition of synthetic images generated from the original images could be used to improve the performance of the classification network.

Generative data augmentation pipeline was followed for the experiments in chapter 5 and 6 where class-conditional and class-specific GANs were trained on subsets and the original dataset. The pipeline has the following steps:

- Original dataset baseline performance evaluation
- Baseline performance for subsets
- GAN training with the original dataset
- GAN training with each subset
- Evaluation of the classification performance of the original dataset and with the addition of synthetic images generated with GAN trained with the original dataset
- Evaluation of the classification performance of subsets and with the addition of synthetic images generated with GAN trained with each subset

The length of GAN training is shown by the number of images the discriminator had observed in total. A typical number for the training of GAN is around 5 millions to 25 millions. In this thesis, GAN models were trained with 25M images at first, however, the duration of the trainings could not be justified with minimal FID score improvements. All experiments and reported results were taken from the best FID score GAN model checkpoints.

Table 6: Best FID scores obtained with class-specific and class-conditional GAN trainings. Class-conditional GAN utilizes all classes of the training set while class-specific GAN models were trained for each class. Weights of class-specific GAN models were transferred from pre-trained FFHQ model while class-conditional GAN weights were initialized randomly.

Best FID scores obtained with class-conditional and class-specific GAN					
Training Method	**Class-Conditional**	**Class-Specific GAN**			
Training set		Mayo 0	Mayo 1	Mayo 2	Mayo 3
Subset 50	154.8	129.7	110.7	100.6	115.5
Subset 100	128.8	117.0	110.7	102.1	119.0
Subset 150	111.9	98.2	86.6	90.8	104.0
Subset 200	94.6	88.7	77.2	81.6	92.1
Subset 250	96.5	78.0	66.9	74.0	79.9
Subset 300	32.4	70.7	63.4	66.6	74.6
Subset 350	29.5	67.2	58.8	60.8	66.9
Subset 400	23.7	61.8	53.5	58.0	64.8
Subset 450	24.5	57.2	49.5	50.8	59.2
Subset 500	18.9	54.1	50.5	51.1	55.4
Original	8.5	15.1	21.5	35.8	53.3

Class-conditional and class-specific GAN models trained with different subsets and the original dataset resulted in the FID scores shown in Table 6. It is evident that class-conditional GAN training resulted in lower FID scores for subsets larger than 250 images per class and for the original dataset. Class-specific GAN training performed better for subsets containing 250 or less images per class. As expected, the imbalanced original dataset resulted in discrepancy in FID score for class-specific GAN as images with lower EMS were over represented while images with higher EMS were under represented.

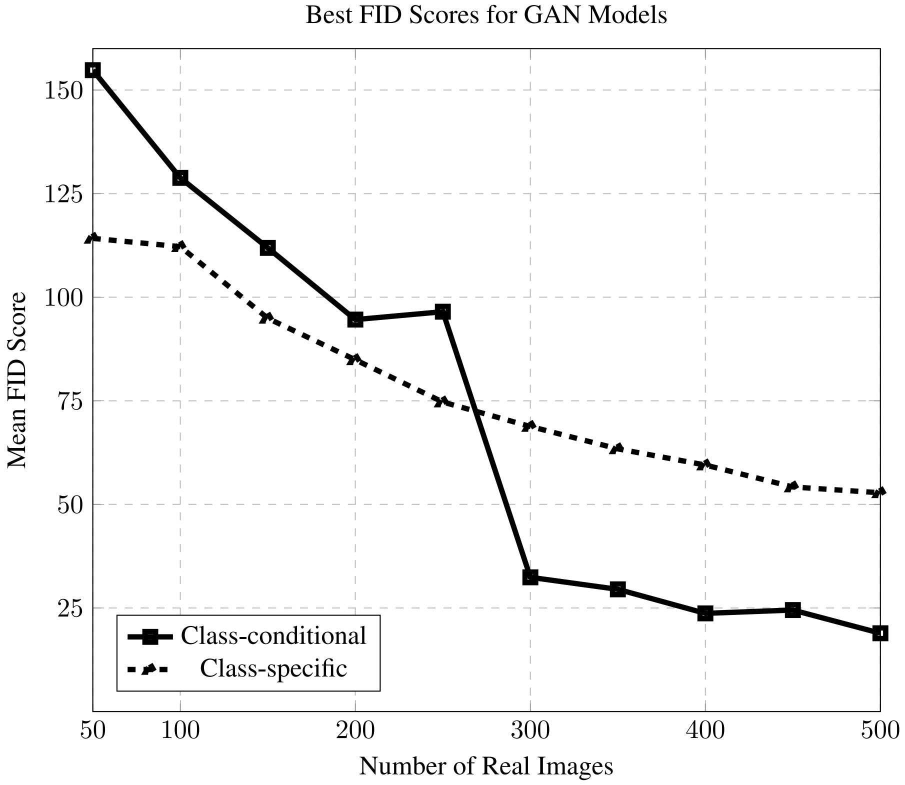

Figure 14: Best FID scores obtained from class-conditional and class-specific GAN training methods. The scores for class-specific were averaged over four classes. GAN models were trained with subsets $S_N where N \in \{50, 100, ..., 450, 500\}$. Class-conditional GAN training weights were initialized randomly. Class-specific GAN training weights were initialized with transfer learning from pre-trained StyleGAN2-ADA GAN on the FFHQ dataset.

4.2.1 Image Classification Baseline

The image classification performance baseline was determined by n times training the same network with random initial weights (no fixed seeds). The number n was determined to be 20 for the work, and all mean, median, and standard deviation are obtained from 20 individual training runs. These runs are grouped in the WANDB experiment tracking tool, and info-graphics can be found online in the relative WANDB project[1] and the appendices.

The image classification baseline performance was obtained with n-experiments and the average of these experiments were reported. The mean and variance of these experiments were also tracked to find outlier results. The experimental trials for image classification baseline are separated into two.

The first part of experiments consists of exclusively classification performance experiments performed over original data. These experiments aim to find a baseline performance using a proven model (not necessarily state-of-the-art). Kani et al. [40] showed and introduced a baseline classification model

---

[1] Experiment tracking with WANDB projects can be reached at https://wandb.ai/metu-vision-lab/projects

and shared its performance evaluation. The same model was trained to obtain similar performance results.

The second part of the experiments was about finding the saturation point where no further addition of real images improved classification performance. These experiments were performed over different subsets. The main difference between subsets was their balanced number of images for each class. There were equal numbers of classes in each dataset's training set. For completeness, the validation and test sets were kept same.

Later parts of experiments contain empirical approaches to find if it is possible to obtain higher classification performance through generative data augmentation.

4.2.2 Improving Image Classification Performance

The image classification performance of neural networks have been evaluated with two distinct types of datasets: subsets and reduced dataset. The reduced dataset is the dataset with instruments removed and with an imbalanced number of samples in each class. The subsets are balanced datasets which are subsets of the reduced dataset. Subsets S containing N number of images in each class, denoted by $S_N where N \in \{50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$, were created to represent different amounts of images.

First, an improvement over the baseline has been aimed with the original dataset. The original dataset was slightly enhanced by excluding the images with instruments and markings as discussed in 3.

Classification performance can be improved by applying classical augmentation methods. The classical augmentation methods of rotation between 0-360 degree and horizontal flipping were used in this thesis. The classification performance metrics of accuracy, precision, specificity, sensitivity, and F1 score are shown in Table 14.

4.2.3 Class-Conditional GAN Training

The class-conditional GAN training were conducted with all subsets and the original dataset. The class information is provided for the GAN training in this method. However, class-conditional training is limited only to random weight initialization as the StyleGAN2-ADA framework does not support transfer learning with conditional GANs. It was expected that class-conditional GAN training would be more beneficial for the imbalanced dataset. The results shown in Table 6 and Fig. 14 shows that class-conditional GAN was more beneficial for higher number of images per class subsets and the original dataset. For lower number of images per class, class-specific GAN training performed better.

4.2.4 Class-Specific GAN training with Weight Initialization through Transfer Learning

Convolutional Neural Networks (CNNs) learn different levels of detail from image datasets in different neural layers. The first layers learn low-level details with a small number of spatial information, while later layers learn high-level details. As low-level features can be learned from any image dataset, the training process for low-level features does not have to be with an imbalanced and highly rare dataset at hand. Although it is recommended that a dataset with similar properties as the original dataset should be used for pre-training, a dataset from different domain can also be used. In this work, a model that has been trained with StyleGAN2-ADA architecture on FFHQ 256×256 dataset was used as the pre-trained model. The pre-trained model has been obtained from [46].

4.2.5 Saturation Experiments

The effects of additional real images have been evaluated in saturation experiments. The trend of performance increase shows that the addition of real images almost always increases the performance of a classification model. However, the rate of increase decreases gradually and it is expected that a classification model converges to an upper performance limit (saturated) when trained with sufficient number of real images.

An important question that arises here is whether or not a dataset is adequate to saturate a classification model. Subsets of the original dataset can be used to evaluate the effects of the data saturation.

It was expected to find a saturation pattern with the number of real images used in classification model training. The saturation can be identified as no performance increase with additional images. The saturation experiment had a limitation set by the least represented class of Mayo 3 which had only 542 images.

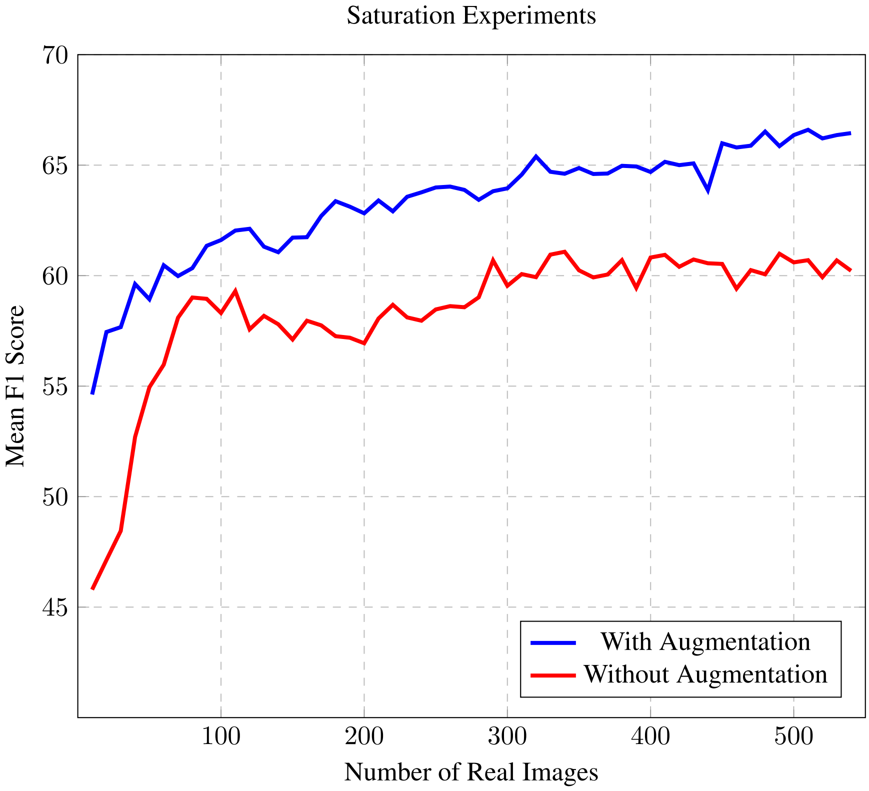

Figure 15: Saturation Experiment with real images $N \in \{10, 20, ..., 530, 540\}$. The effect of classical augmentation is marginally present in the graph.

The saturation experiments of the original dataset have been conducted with subsets $S_N where N \in \{10, 20, ..., 530, 540\}$ presented in Fig. 15. As seen in the figure, using classical augmentation methods increased the classification performance of the deep learning model, highlighting the importance of using classical augmentation. The subsets used in this experiment were not enough to saturate the classification model as expected; hence all subsets in this experiment were indeed data-limited.

CHAPTER 5

GENERATIVE DATA AUGMENTATION WITH CLASS-CONDITIONAL GANS

5.1 Class-Conditional GAN Training

Conditional GANs are trained with the utilization of class information over regular GAN training. The information of class labels is used to create a new dimension with the number of classes as the depth. Conditional GAN is beneficial for training imbalanced datasets where a particular class is underrepresented. In the use case scenario, Mayo 3 class's representation ratio to Mayo 0 is 1:10. In theory, the GAN training can benefit from learning from all classes because all images are from the same object but with different levels of symptom severity. The GAN training can be improved by adding different levels of disease symptoms over the base class Mayo 0.

5.2 Generative Data Augmentation with Class Conditional GANs

The effects of generative data augmentation with class conditionally trained GANs were explored with the following experimental setups.

Firstly, the baseline performance of the original dataset, which had images with instrumentation and artefacts excluded, was evaluated. Secondly, a class conditional GAN was trained using the original dataset. Thirdly, synthetic images with different truncation options were generated using class conditional GAN. Finally, the effects of the generative data augmentation on the performance were evaluated and compared with the baseline performance of the classification network with only real images.

The performance effects of generative data augmentation with limited data: Subsets of the original dataset containing an equal number of samples in each class were used to evaluate the performance of generative data augmentation with limited data. To evaluate generative data augmentation with limited data, the subsets of 50 and its integer multiples per class were used to evaluate the baseline performance of each subset. Each subset was also used to train class conditional GANs. The trained GANs were then generated synthetic images associated with each subset. The effects of generative data augmentation were tested by injecting synthetic images gradually to each subset.

5.3 Experiments Using the Original Dataset

The original imbalanced dataset was used in this experiment. The generative data augmentation was performed to add an equal number of synthetic images to each class. The effects of generative data augmentation on the F1 score can be observed from Fig. 16.

The baseline performance of 68.8% increased to 70.9%. The highest performing truncation option was found to be 0.5. Furthermore, the two highest scores were obtained with 500 and 1000 synthetic images per class.

In this experiment, truncation option of 1.5, which corresponds to samples generated outside the original data distribution, performed worse than other truncation options of 1 or lower.

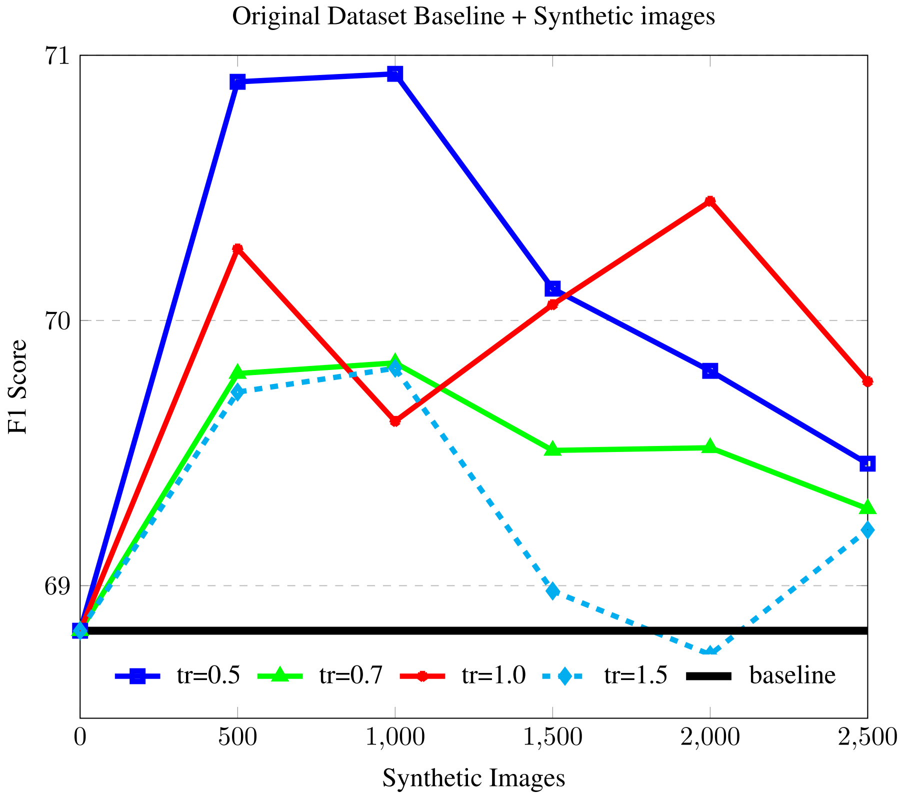

Figure 16: Comparison of F1 scores by addition of synthetic images with different truncation options over baseline original dataset.

It can be concluded that generative data augmentation over the original dataset was beneficial up to a certain amount of synthetic images. In the original dataset the class containing the least number of samples was Mayo 3, with around 500 images. In comparison, the class having the most samples was Mayo 3, with 4000 images. This imbalance in the number of samples in each class was reduced with this approach as the same number of images are added to each class. However, this is not ideal and a data informed approach of synthetic image addition could prove more beneficial for the imbalanced dataset.

5.4 Generative Data Augmentation Experiment Results on Subsets

The experiments in this section were conducted with synthetic images generated with the truncation option of 1.0. The number of synthetic images used for generative data augmentation equals integer multiples of each baseline subset. The subsets are denoted by $S_N where N \in \{50, 100, ..., 450, 500\}$ contained integer multiples of 50.

The performance of real image only subsets was evaluated to determine the advantages and disadvantages of generative data augmentation. The effects of generative data augmentation were shown together with the baseline performance of subsets $S_N where N \in \{50, 100, 150, 200, 250\}$ real images per class in Fig. 17. The subsets of $S_N where N \in \{300, 350, 400, 450, 500\}$ images per class and their relative generative data augmentation effects were shown in Fig. 18.

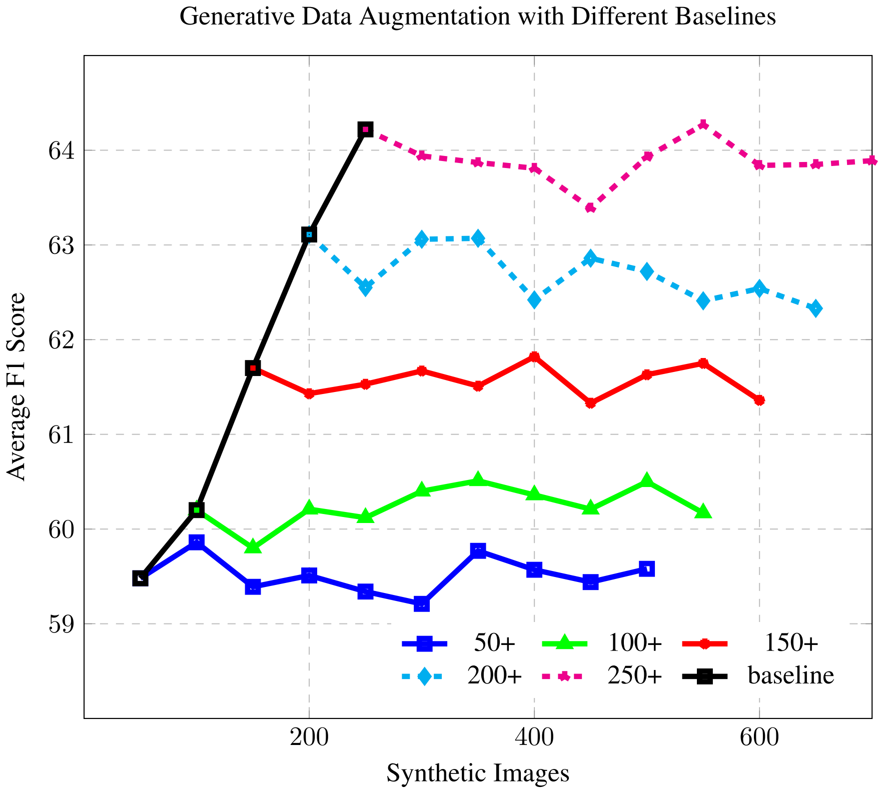

Figure 17: Comparison of F1 score performance increase gained by addition of synthetic images over different baselines of subsets $S_N where N \in \{50, 100, 150, 200, 250\}$ real images per class. Synthetic images were generated with different truncation options of $tr \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.5, 2\}$

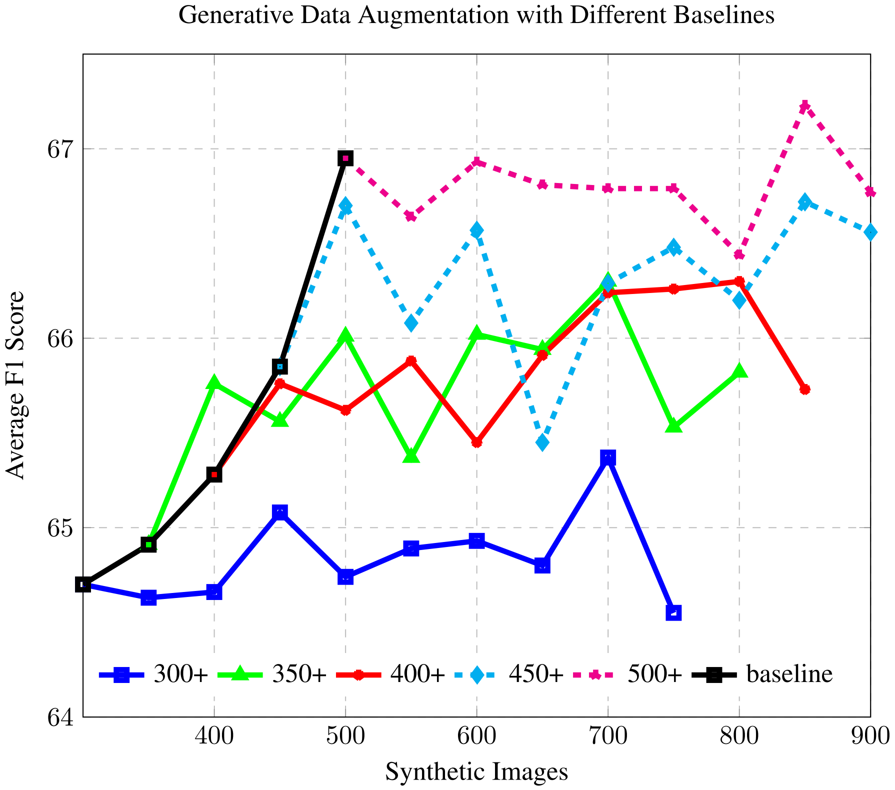

Figure 18: Comparison of F1 score performance increase gained by addition of synthetic images over different baselines of $S_N where N \in \{300, 350, 400, 450, 500\}$ real images per class. Synthetic images were generated with different truncation options of $tr \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.5, 2\}$

According to the Fig. 17 and Fig. 18, most experiments failed to improve the baseline performance. With the exception of 100 and 150 images per class experiments, where there were marginal gains, the majority of others had a steady decline with the increasing number of synthetic images.

The table 19 demonstrates the average performance of generative data augmentation experiments and baseline subset performance.

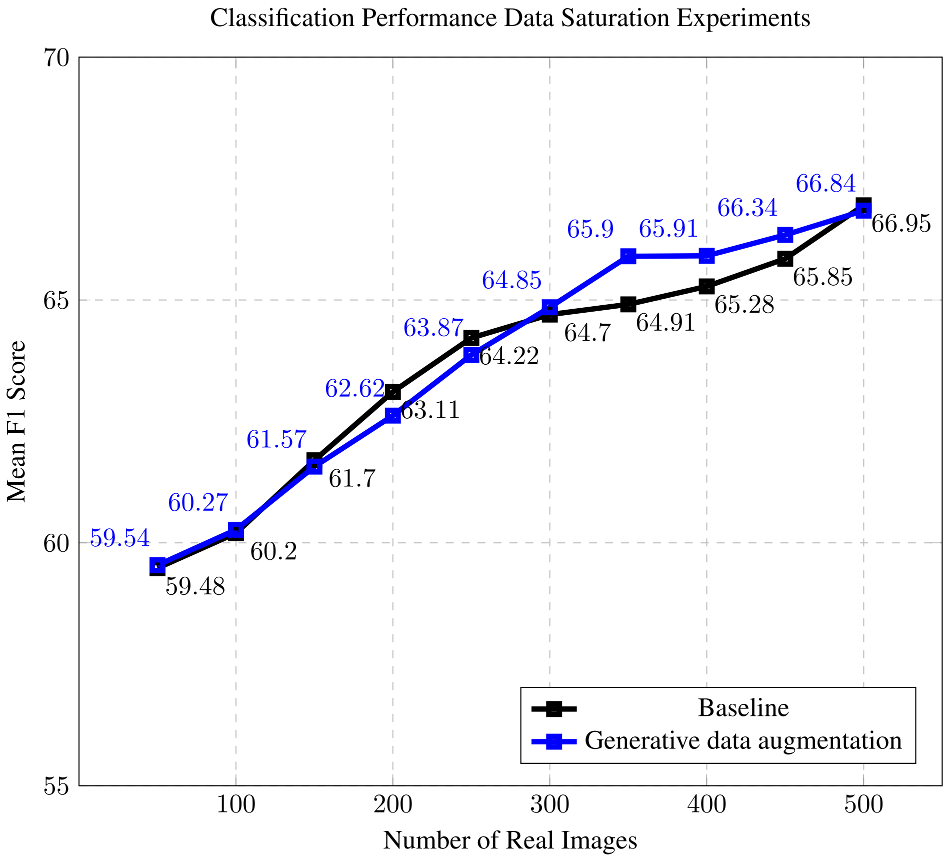

Figure 19: Baseline performance of F1 score of neural networks trained with only real image data. And average performance of F1 score score with generative data augmentation.

5.5 Confidence Level Based Performance Analysis

In this thesis, the results were evaluated by calculating the mean of N number of experiments. Instead of using mean performance to evaluate metrics, a confidence level approach can also be used. Confidence level approaches for four configurations were evaluated. The configurations were:

- No augmentation baseline
- Baseline with classical augmentation
- Generative data augmentation with conditional GAN
- Generative data augmentation with class-specific GAN

Confidence levels of 100%, 80% and 60% were evaluated on a per class basis for 772 images in EMS 0 class, 406 images in EMS 1 class, 146 images in EMS 2 class and 119 images EMS 3 class.

The confidence level of 100% shows a dramatic accuracy decrease in EMS 1 and especially in EMS 2 class compared with EMS 0 and 3 classes in the baseline as shown in table 7. With classical augmentation, EMS 2 and 3 classes increase substantially while an increase is also observed in EMS 1 class and almost no increase could be observed in EMS 0. With the application of generative data augmentation with conditional GAN, performances of EMS 0, 1 and 3 classes increase by 8 percentage points while there is a decrease for EMS 2 class. Class specific GAN powered generative data augmentation showed the best results for EMS 0, 1 and 2 classes with second best results for EMS 3 class.

Table 7: Image classification performance with 10 correct classification out of 10 experiments (100% confidence).

100% Confidence Image Classification Performance				
Configuration	**Mayo 0**	**Mayo 1**	**Mayo 2**	**Mayo 3**
Baseline (No Aug)	65.0%	17.2%	1.4%	29.4%
Baseline (With Aug)	65.2%	24.9%	23.3%	47.9%
GDA (Aug+Cond GAN)	71.6%	31.3%	15.8%	**59.7%**
GDA (Aug+specific GAN)	**72.8%**	**37.2%**	**28.8%**	57.1%

Confidence level of 80% demonstrates a similar baseline performance with EMS 2 having the worst results and EMS 0 the best, as shown in Table 8. Applying classical augmentation increased the performance of EMS 1 2 and 3 classes at the cost of 3 percentage points in EMS 0 class. The overall performance with conditional GAN generative data augmentation did not differ from the baseline with classical augmentation. EMS 0, 1 and 3 scores were slightly improved ad the cost of EMS 2 score. Generative data augmentation with class-specific GAN were again the most successful for EMS 1, 2 and 3 classes.

Table 8: Image classification performance with 8 correct classification out of 10 experiments (80% confidence).

80% Confidence Image Classification Performance				
Configuration	**Mayo 0**	**Mayo 1**	**Mayo 2**	**Mayo 3**
Baseline (No Aug)	81.2%	37.7%	14.4%	58.8%
Baseline (With Aug)	77.6%	50.0%	47.9%	68.1%
GDA (Aug+Cond GAN)	**81.2%**	54.2%	34.9%	68.9%
GDA (Aug+specific GAN)	80.6%	**56.2%**	**51.4%**	**71.4%**

For 60% confidence image classification performance, the baseline with no augmentation demonstrates higher performance for EMS 0 and 3 classes and lower performance for EMS 1 and 2 as shown in table 9. With the application of classical augmentation, the EMS 1, 2 and 3 class performance were increased at the cost of EMS 0 class performance. Application of generative data augmentation with class conditional GAN did not improve the overall performance and even reduced EMS 2 class performance. On the other hand, generative data augmentation with class specific GAN demonstrated the best performance for EMS 1 and 2 classes and second best for EMS 0 and 3 classes.

Table 9: Image classification performance with 6 correct classification out of 10 experiments (60% confidence) .

60% Confidence Image Classification Performance				
Configuration	**Mayo 0**	**Mayo 1**	**Mayo 2**	**Mayo 3**
Baseline (No Aug)	**89.8%**	53.0%	40.4%	68.9%
Baseline (With Aug)	83.8%	63.1%	**63.0%**	**77.3%**
GDA (Aug+Cond GAN)	85.5%	63.3%	50.0%	75.6%
GDA (Aug+specific GAN)	84.3%	**65.8%**	**63.0%**	76.5%

To conclude, the application of classical data augmentation improves the robustness measured by confidence levels. The per-class performances of each confidence level were improved with generative data augmentation. The GAN training method of using transfer learning was superior to the class-conditional GAN training from scratch. Finally, failure analyses were given in detail in Appendix. B

5.6 Discussion on Generative Data Augmentation with Class Conditional GANs

Generative data augmentation with synthetic images generated by Class-Conditional GAN models proved to be beneficial for the original dataset. The baseline classification F1 score performance was increased by 3.1 % with generative data augmentation utilizing class-conditional GAN model. Even though this increase was the highest among many data points, the other truncation options and number of synthetic images were also beneficial. It can be concluded that class-conditional GAN models can be used for generative data augmentation of an imbalanced dataset.

The performance contribution of generative data augmentation with synthetic images was for subsets showed conflicting results. Although some data points were indeed better than the baseline performance, there were no consistent improvement over the baseline performance. For completeness, average F1 score obtained with generative data augmentation were calculated and comparisons with the addition of real images and with the baseline were held. The details about all subsets and the relative percentage performance increase can be observed from Table 10.

Table 10: Generative data augmentation performance effects with Conditional GAN.

Subset	**Baseline Performance**	**Average generative data augmentation**	**Performance Increase**	**Performance Increase Over Real**
50/class	59.48%	59.54%	0.1%	-0.7%
100/class	60.20%	60.27%	0.1%	-1.4%
150/class	61.70%	61.57%	-0.1%	-1.5%
200/class	63.11%	62.62%	-0.5%	-1.6%
250/class	64.22%	63.87%	-0.3%	-0.8%
300/class	64.70%	64.85%	0.1%	-0.1%
350/class	64.91%	65.90%	1.0%	0.6%
400/class	65.28%	65.91%	0.6%	0.1%
450/class	65.85%	66.34%	0.5%	-0.6%
500/class	66.95%	66.84%	-0.1%	N/A

The table 10 shows generative data augmentation performance evaluation averages with different subsets. Generative data augmentation performance for each subset was compared with the baseline performance of that subset and the performance of the subsequent subset baseline. The percentage difference between each data point and the next baseline performance is provided

For every subset, an increase in classification performance could be achieved with generative data augmentation. The class-conditional GAN training with very limited data was not reliably successful for generating synthetic images capable of increasing the performance of a classification model. On the other hand, class-conditional GAN was reliably successful for generative data augmentation with the imbalanced original dataset.

CHAPTER 6

GENERATIVE DATA AUGMENTATION WITH CLASS-SPECIFIC GAN

6.1 Class-Specific GAN Training

Synthetic data augmentation by using synthetic image data generated by GANs trained with transfer learning option are discussed in this chapter. The effects of synthetic data augmentation over the accuracy, precision, recall, and F1 score metrics will be detailed with experiments.

GAN models in this chapter are trained with subsets containing an equal number of image data from each class. These equal number datasets are named subsets, with the number of images per class defining which subset is being used.

There are ten subsets with $S_N where N \in \{50, 100, ..., 450, 500\}$ images per class. Each subsequent subset contains the previous dataset's images and 50 new images per class. This way, each subset can be used to determine the benefits of adding real authentic images over a dataset.

The benefits of the addition of real images can be compared with synthetic image injection, and results for both methods will be discussed. It is imperative to note here that additional real images may not always be possible, whereas the proposed method of synthetic data injection is expected to be available in broader applications.

Each subset will be used as a baseline for synthetic data injection where GANs trained with respective subsets are responsible for generating synthetic images. The synthetic images are generated with different truncation options $tr \in \{0.5, 0.7, 1.0, 1.2, 1.5, 2.0\}$.

6.2 Generative Data Augmentation with Transfer Learning

As the training with 25M images took 10 days to finalize, 5M images trainings of both class-specific and class-conditional GANs were used for the comparison. 25M images training were not feasible for subsets and the original dataset, which would result in 48 times 10 days of training. Instead all subset class specific GAN training were conducted with 2 days trainings of 5M images length.

6.3 Original Dataset Experimentation Results

In this experiment, the baseline classification performance of the original dataset and generative data augmentation approach over it were compared. The generative data augmentation was realized by the GAN training with transfer learning from a pre-trained FFHQ dataset [46]. The FFHQ dataset had the same image dimensions as the original UC dataset. However, due to the lack of transfer learning support for conditional GAN training in StyleGAN2-ADA structure, a different GAN was trained for each Mayo class. StyleGAN3 structure also lacks this feature.

Class-specific GANs were used to generate synthetic images belonging to their own training dataset class. Class-specific GAN training resulted in worse FID scores compared with the class-conditional GAN training. The lowest FID score obtained with class-conditional GAN was 15.1 for 5M images and 8.5 for 25M images. However, class-scpecific GANs trained with the original dataset resulted in FID scores gradually worse for increasing EMS as shown in Table 6, mainly due to having fewer number of samples for these classes.

The contribution of generative data augmentation to the classification performance was limited for all truncation options except 1.2. The synthetic images generated with truncation option 1.2 proved to be highly beneficial for the classification performance. On the other hand, the classification performance almost always degraded with other truncation options.

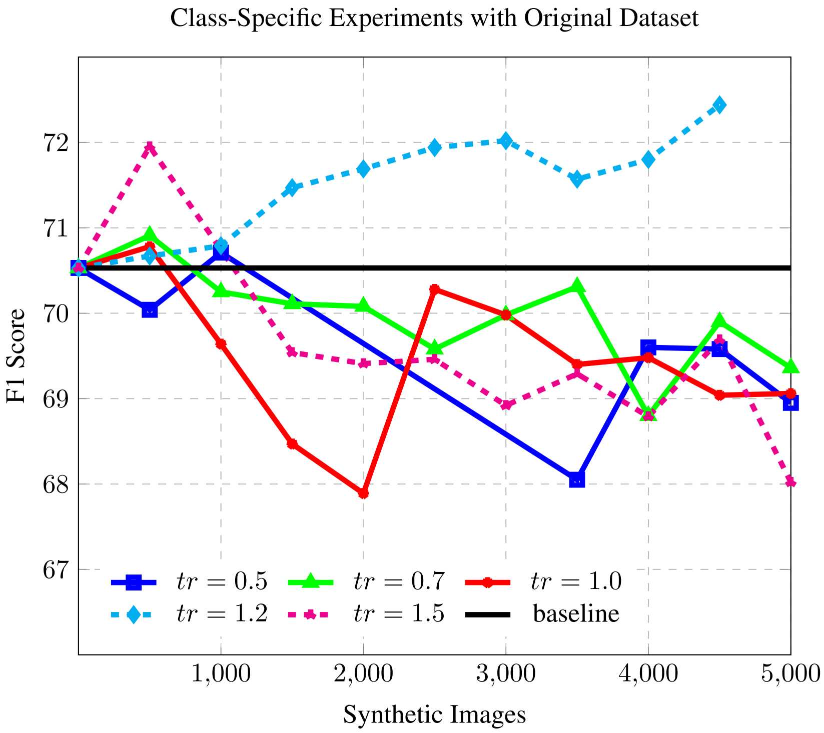

Figure 20: Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline original dataset. Synthetic Images were generated with different truncation options.

6.4 Experimental Results on Subsets

6.4.1 Saturation Experiment

The deep learning model performance increased steadily with the increasing number of real images used in training. As shown in Fig. 21, the mean F1 score performance of the classification network started from 55.8% with 50 real images per class and increased up to 68.1% with 500 real images per class. The addition of extra images over 300 and 350 per class subsets did not improve the F1 performance as their F1 scores were 65.6% and 65.9%, respectively. However, the slight increase in the range of 300 and 450 images per class was followed by the 500 images per class with a 68.1% F1 score. Although a decrease in performance improvement with the additional real images is present in the figure, a saturation, where new images do not contribute to a performance improvement, could not be observed.

The saturation experiment was also valuable to determine the baseline classification performance for all subsets. The classification performance of a deep learning model trained only with real images was later used as a baseline. The later experiments aimed to increase the performance of the classification model with the addition of synthetic images.

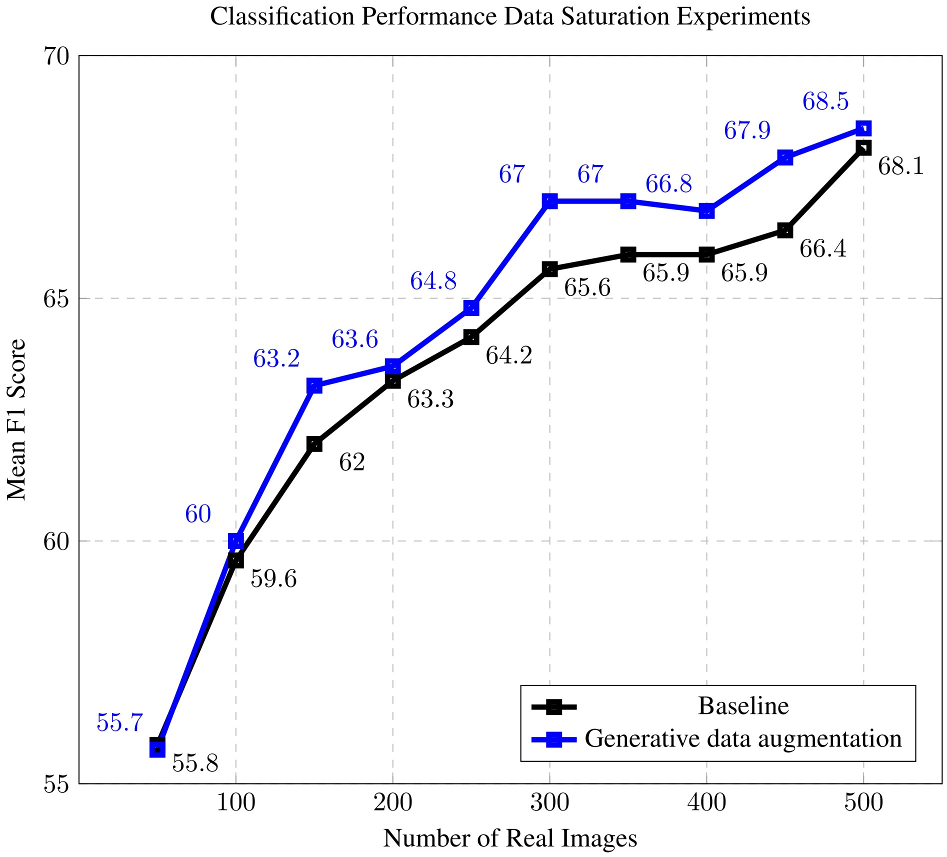

Figure 21: F1 score performance of neural networks trained with only real image data.

6.4.2 50 Images

In this experiment, 50 images per class totaling 200 image data in the training set were present. The aim was to improve the baseline 55.77% F1 score of the classification network. The classification performance with the addition of synthetic images is shown in 22. Synthetic images generated with a truncation option lower than 1.0 resulted in poor generative data augmentation performance compared to those generated with a truncation option higher than 1.0.

The highest performance increase in this experiment was obtained by adding 100 synthetic images per class generated with truncation=1.5 option. The baseline performance of 55.77% for 50 real images per class was increased up to 57.48% for 50 real and 100 synthetic images per class.

It was concluded that further addition of synthetic images was not beneficial for classification performance as the two highest scores were obtained by 50 and 100 synthetic images while a higher number of synthetic images did not surpass these values.

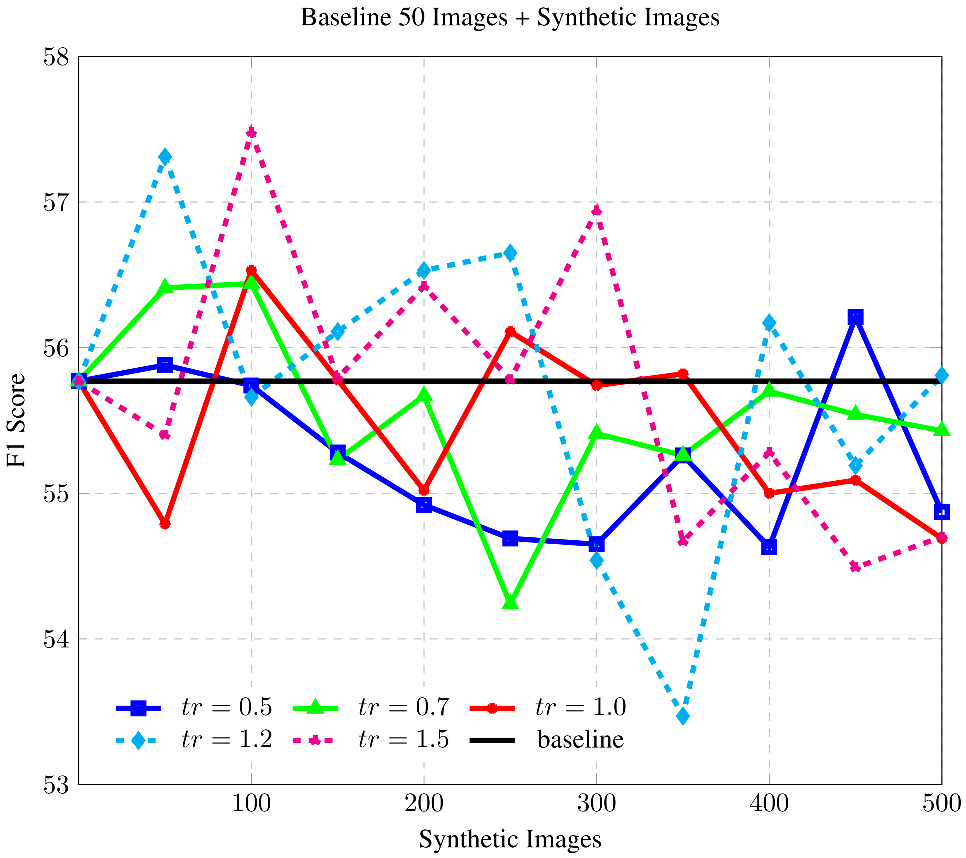

Figure 22: Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 50 images per class. Synthetic Images were generated with different truncation options.

6.4.3 100 Images

The baseline classification performance of the neural network trained with 100 real images per class training set was 59.64%. In this experiment, the effects of the addition of synthetic images were analyzed and shown in Fig. 23. Synthetic images used in this experiment were generated by the same training set used by the classification model.

The addition of only 100 synthetic images caused a decrease in performance or had a negligible effect. However, further injection of synthetic images increased the performance most of the time. The highest increase in performance was obtained by the addition of 600 synthetic images generated with truncation=0.5 option (solid blue line). The baseline performance of 59.64% was increased to 61.49%.

In this experiment, truncation options of 0.5, 1.2, and 2.0 performed better than other options. It is not clear whether the higher or lower truncation options benefit the generative data augmentation in this experiment. However, the best-performing option is observed to be 0.5. The performance increase occurred when 600 or 700 synthetic images per class were introduced to the 100 original images per class. There was no evidence that additional synthetic images would improve the classification performance.

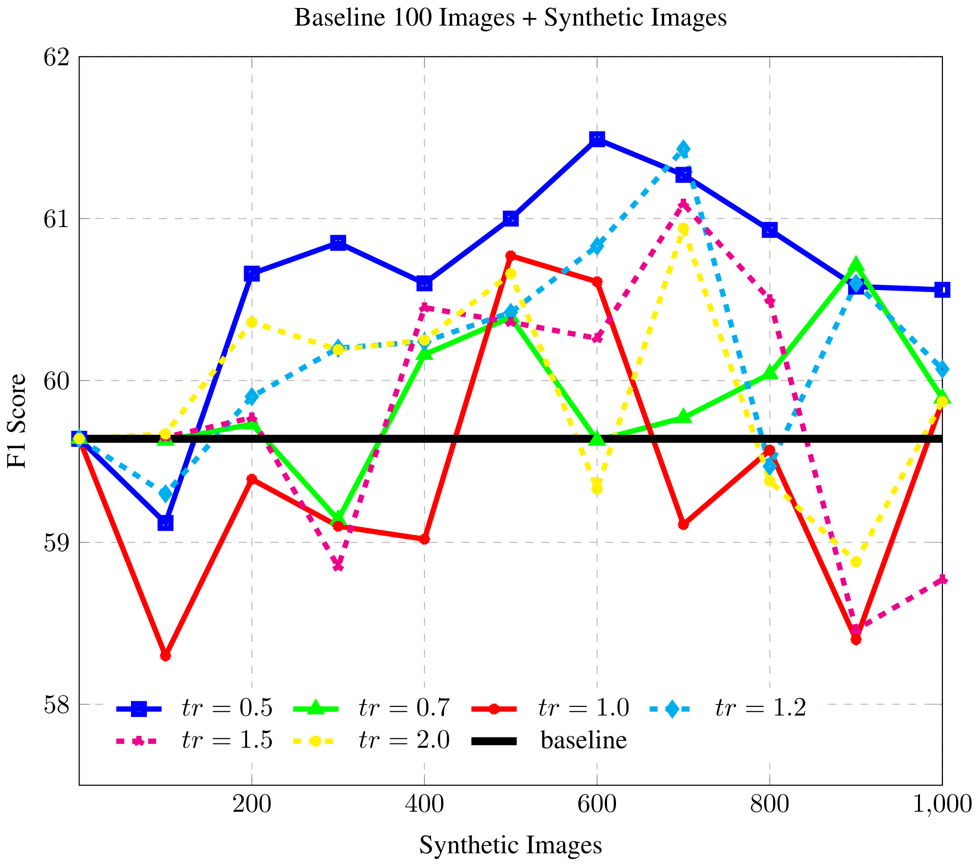

Figure 23: Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 100 images per class. Synthetic Images were generated with different truncation options.

6.4.4 150 Images

The baseline classification of 150 images per class training set was found to be 62%. According to Fig. 24, the best performance improvement of generative data augmentation occurred at 450 per class synthetic images generated with truncation=0.7 option. The baseline performance was increased from 62% to 64.6%.

In this experiment, all truncation options had a positive performance effect above 300 and more images per class data points. The only exception was the truncation 2.0 option and 450 images per class. Just like 50 and 100 real images experiments, the addition of N number of synthetic images deteriorated the classification performance, where N is the original number of samples per class.

Truncation options that were equal or lower than 1.0 performed better in generative data augmentation, while higher truncation options did not deteriorate classification performance at 150 synthetic images per class data point.

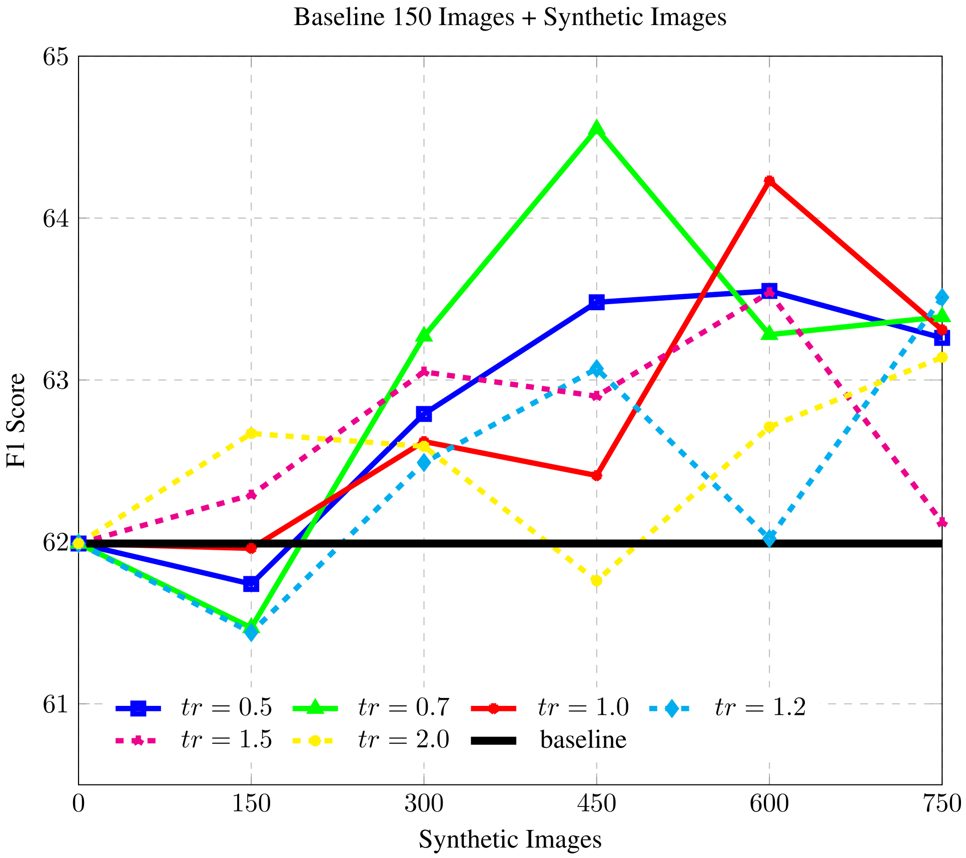

Figure 24: Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 150 images per class. Synthetic Images were generated with different truncation options.

6.4.5 200 Images

In this experiment, the baseline classification performance of 63.31% was increased to 65.84%. The best improvement was obtained by injecting 800 synthetic images per class into the real training set. The best truncation option was 0.5 by far compared with other options. 0.5 truncation option did not deteriorate the baseline performance in any of its data points while other options did so. Also, this option performed better compared to other options.

Truncation option 0.7 performed the second-best data point as well as the worst data point in this experiment. The volatility of the 0.7 truncation option was also evident in the 1.0 option. Both of these options can be categorized as in the range of original data distribution, and they have performed better in some data points compared to out-of-range distribution options.

The truncation options of 1.2, 1.5, and 2.0 performed poorer than the real image baseline in most data points and also performed poorer than other options of 0.5, 0.7, and 1.0. In this experiment, it was evident that a higher performance increase could be obtained by truncation options in the original distribution range. The details and trends of different truncation options can be observed from Fig. 25

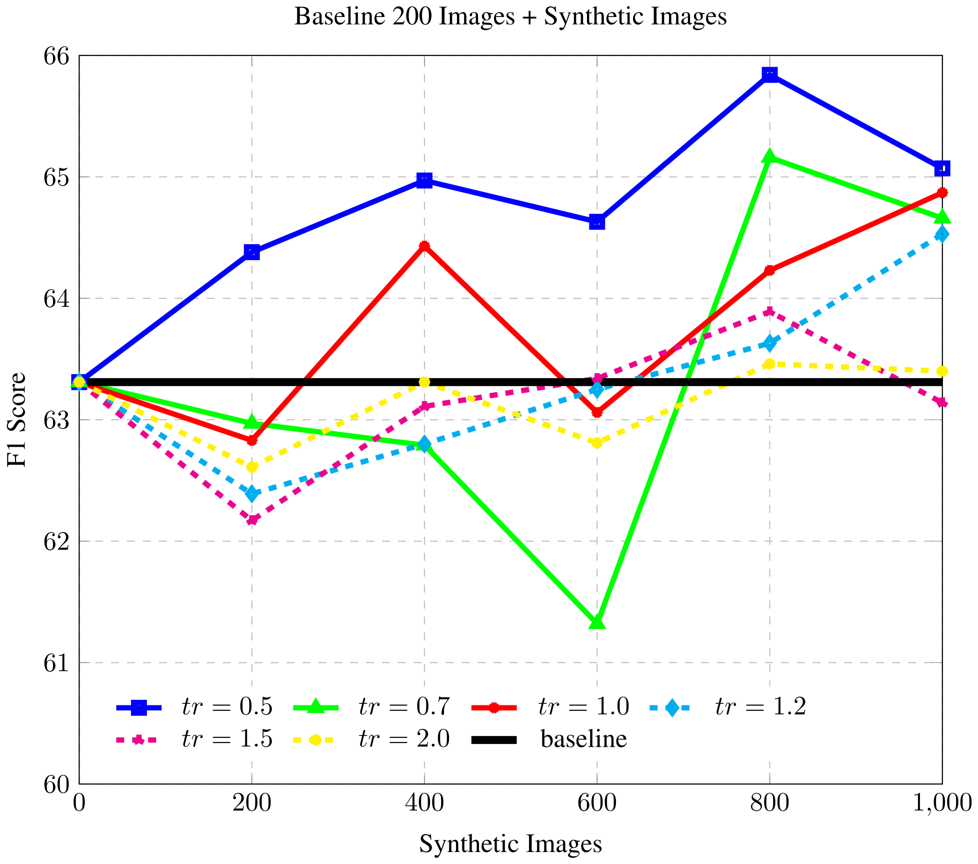

Figure 25: Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 200 images per class. Synthetic Images were generated with different truncation options.

6.4.6 250 Images

The aim of this experiment was to improve the baseline classification performance of the image classification models. The baseline performance of 64.24% was increased to 66.06% at best, as shown in Fig. 26. The highest performance increase was obtained with 750 (N=3) synthetic images, where the synthetic images were trained with the truncation option of 0.7.

With the increasing number of synthetic images, the performance improvement over baseline real images increased. All truncation options performed better than the real baseline; however, some discrepancies were also present. First, the truncation option of 0.5 had never deteriorated the baseline performance. However, it caused mediocre performance and conflicting and varying improvements compared to other options. A similar problem was also present in the option of 1.0 truncation; the worst performance data point belonged to this option.

The truncation options out of the range of original data distribution (1.2, 1.5, and 2.0) had shown better performance with an increasing number of synthetic images. They were also more robust than in-range truncation options.

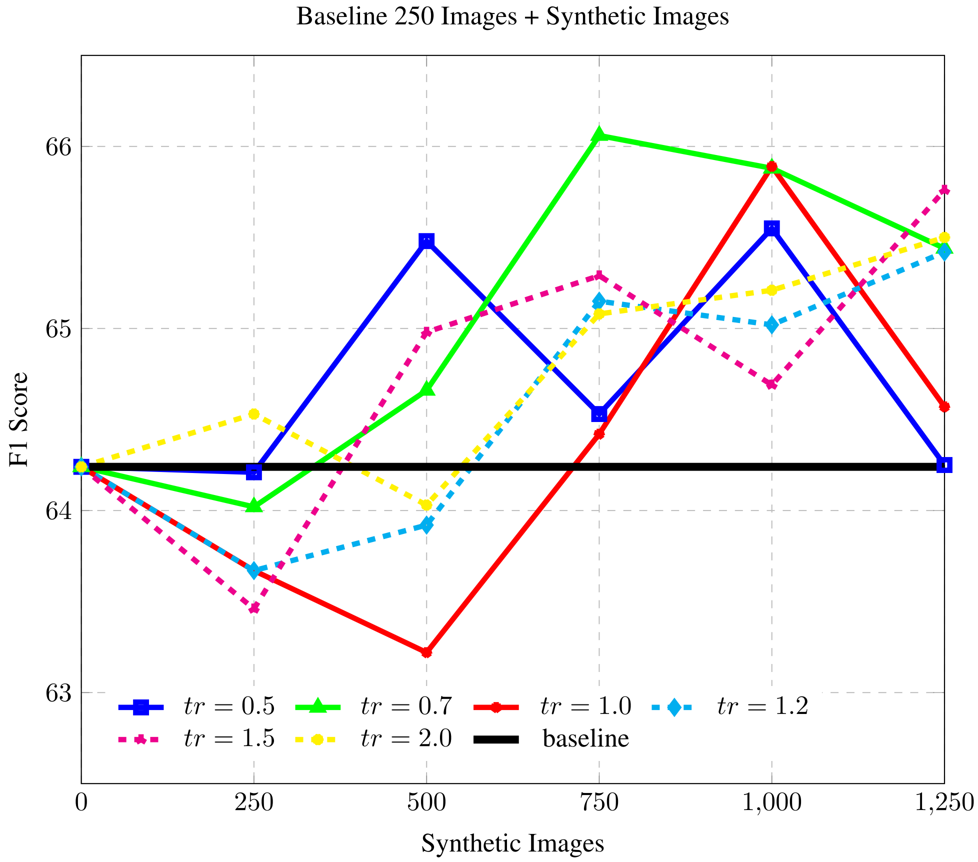

Figure 26: Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 250 images per class. Synthetic Images were generated with different truncation options.

6.4.7 300 Images

This experiment used 300 real images per class to determine the baseline performance. The baseline performance of 65.56% was improved to 68.54%. This best data point was achieved by injecting 750 synthetic images generated with a truncation option of 1.2. However, the truncation option of 1.2 was not robust and eventually deteriorated the baseline performance. The truncation option of 1.0 had also performed poorer than baseline at 150 synthetic images per class data point.

Overall truncation options of 0.5 and 0.7 performed more robustly and achieved 4.5% performance improvement. Out-of-range truncation options also improved baseline performance as they were always better than baseline except for the 1.2 option's 2400 synthetic image data point.

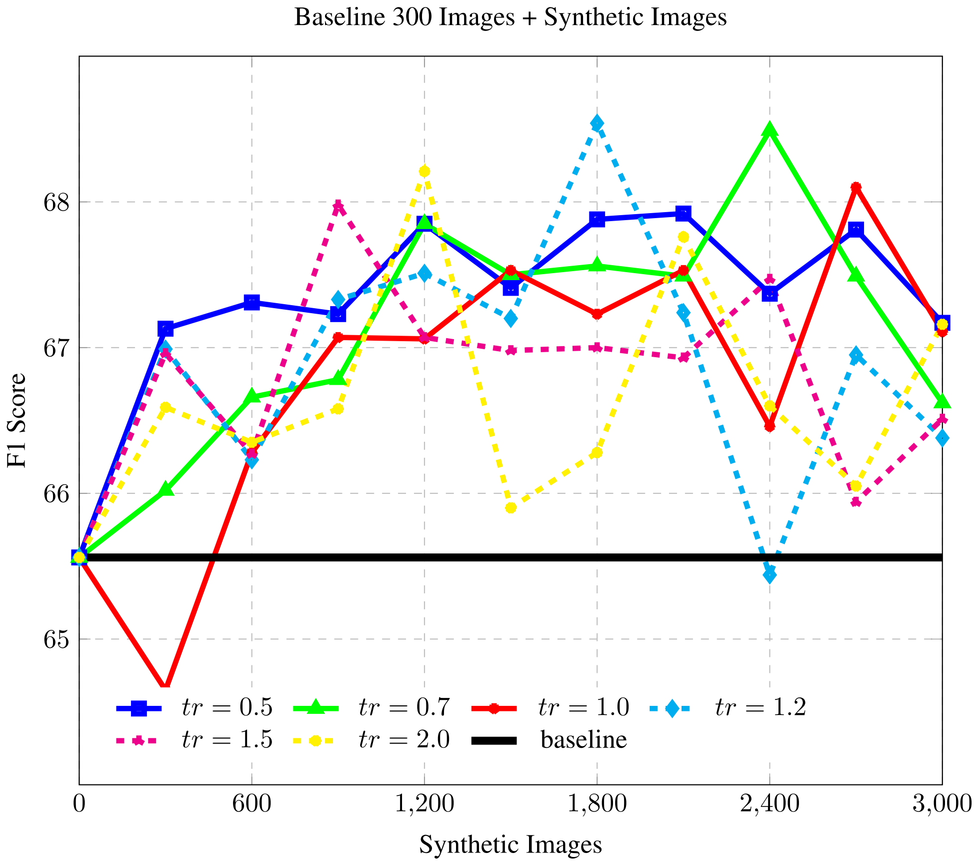

Figure 27: Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 300 images per class. Synthetic Images were generated with different truncation options.

6.4.8 350 Images

The baseline classification performance of 65.88% was improved to 68.24%. The improvement was achieved by injecting 1050 (3N) synthetic images per class into the original dataset. The synthetic images achieving the highest F1 score were generated by the 0.5 truncation option.

The overall performance contribution of all truncation options was similar or better than the baseline performance. The truncation option of 2.0 performed poorly compared with other options.

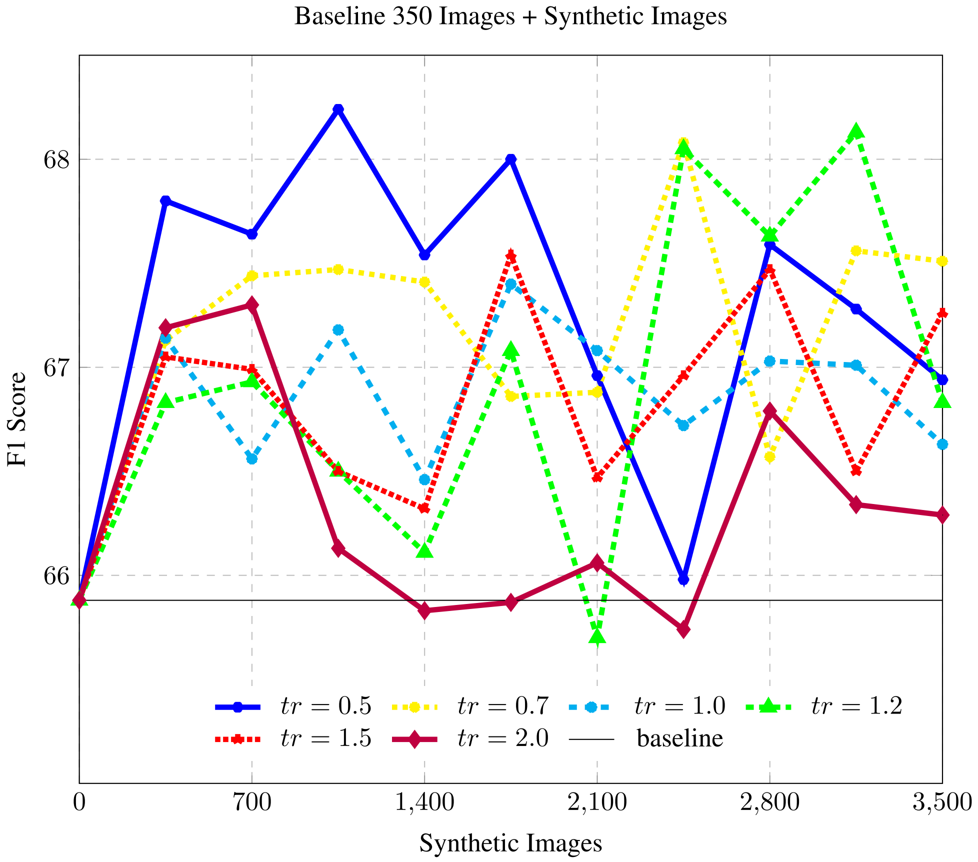

Figure 28: Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 350 images per class. Synthetic Images were generated with different truncation options.

6.4.9 400 Images

With the exception of synthetic images generated with truncation option of 2.0, all generative data augmentation data points shows an increase over the baseline performance. The baseline performance of 65.9% was increased to 69% with 1600 synthetic images generated with the truncation option of 0.5. The overall performance of the truncation option of 0.5 outperforms all other options with a single data point exception where 0.7 option performed better. Overall, a trend can be observed that smaller truncation options performed better at 400 images per class subset.

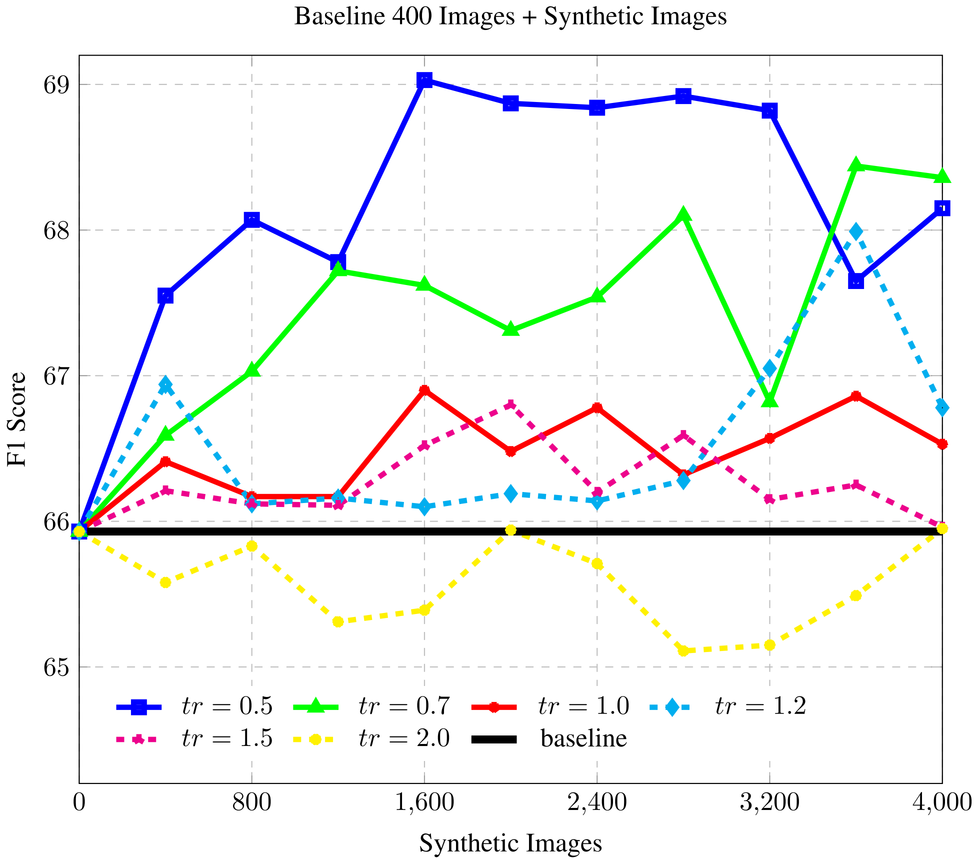

Figure 29: Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 400 images per class. Synthetic Images were generated with different truncation options.

6.4.10 450 Images

The baseline image classification F1 score performance of 66.4% was increased to 70% by the addition of 4500 synthetic images generated with the truncation option of 0.5. The truncation option of 0.5 performed better than other options overall. With only two data points where baseline performance was slightly deteriorated, it can be concluded that generative data augmentation for 450 images per class subset was beneficial.

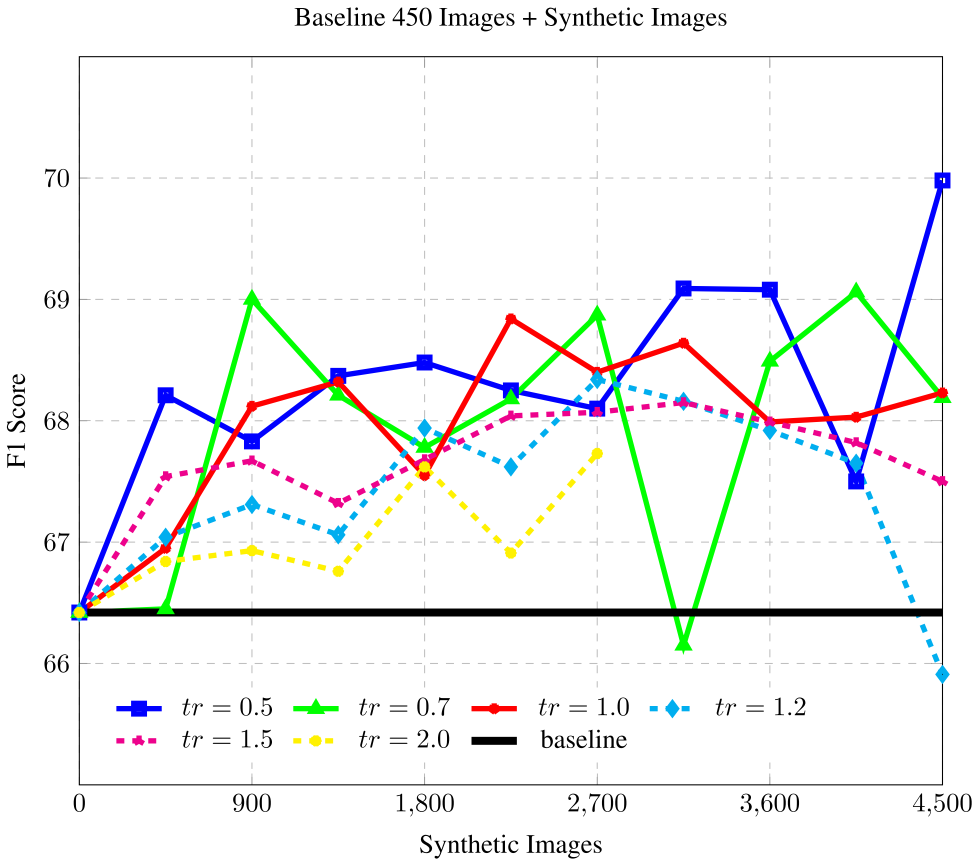

Figure 30: Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 450 images per class. Synthetic Images were generated with different truncation options.

6.4.11 500 Images

The subset of 500 images per class was the final and largest subset. Its baseline classification performance of 68.1% was increased to 70.1%. This increase was obtained by 4500 synthetic images generated with the truncation option of 0.5.

Generative data augmentation with the truncation option of 0.5 demonstrated to be beneficial for the classification performance for all data points. Truncation options of 0.7 and 1.0 also proved to be positive for the classification performance with the exception of single data points for each option. Options higher than 1.0, i.e., 1.2, 1.5 and 2.0 were not as beneficial as other options and each had several data points with worse performance than the baseline.

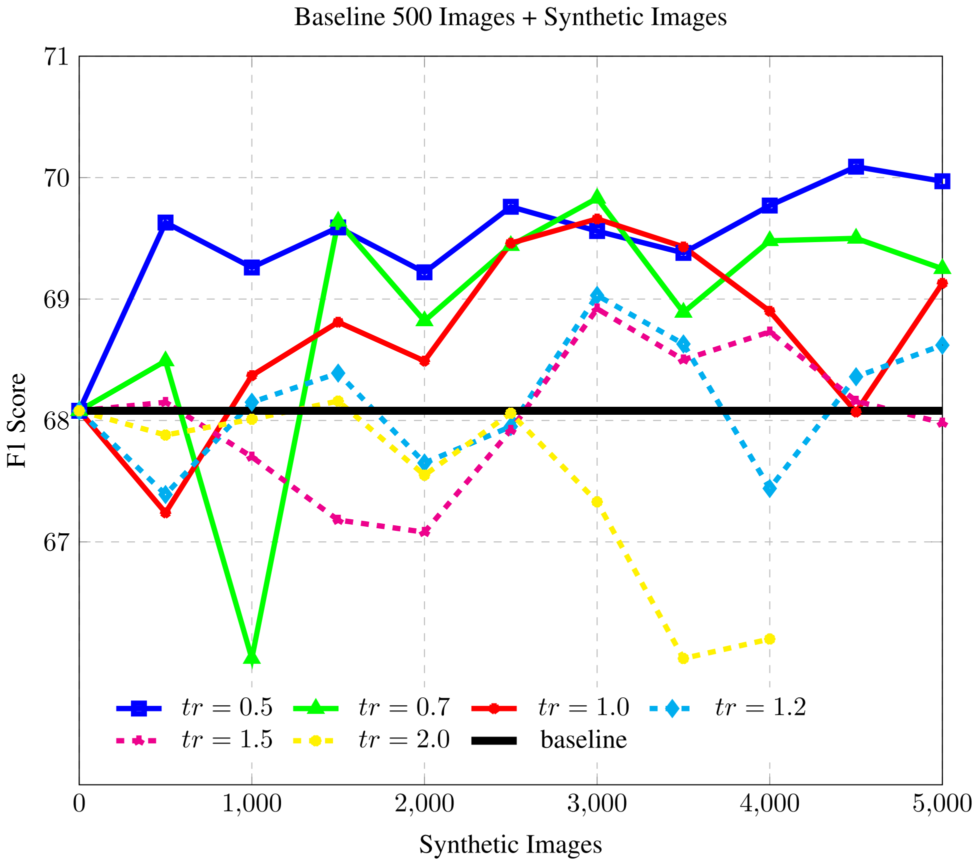

Figure 31: Comparison of F1 score performance increase gained by addition of real images and synthetic injection over baseline 500 images per class. Synthetic Images were generated with different truncation options.

6.5 Discussion on Generative Data Augmentation with Transfer Learning Applied GANs

It can be concluded that the truncation option of 0.5 was the highest performing generative data augmentation option among the options that were tested. The only exception was the subset of 50 images per class case where the truncation option of 1.5 was better than 0.5. However, the experiments with 50 images per class subset were conflicting and inconclusive as performance graphs had several spikes and downfalls. It can be speculated that very limited data, such as 50 images per class subset, can prevent performance improvements with generative data augmentation.

The truncation option of 0.5 had a positive contribution to the classification performance metric of the F1 score. The only exceptions were the first multiples of subsets 100 and 150 and the overall 50 images per class subset experiment.

The baseline performance of the original dataset could only be improved with synthetic images generated with the truncation option of 1.2. All other truncation options failed to improve it consistently. The main reason for this behavior was believed to be the dataset imbalance as there are around 8 Mayo 0 samples for each Mayo 3 sample. The numerous low-level severity images were easier to generate; however, images with higher-level Mayo scores were harder to generate. This resulted in inconsistency for the generative data augmentation. On the other hand, the artificial balance of the subsets proved to be useful for generative data augmentation.

The table 11 shows generative data augmentation performance evaluation with different subsets. Generative data augmentation performance for each subset was compared with the baseline performance of that subset and the performance of the subsequent subset baseline. The percentage difference between each data point and the next baseline performance is provided.

Table 11: Generative data augmentation performance effects with class-specific GAN.

Subset	**Baseline Performance**	**Average generative data augmentation**	**performance increase**	**Performance increase over real**
50/class	55.8%	55.7%	-0.1%	-4.0%
100/class	59.6%	60.0%	0.3%	-2.0%
150/class	62.0%	63.2%	1.2%	-0.1%
200/class	63.3%	63.6%	0.3%	-0.6%
250/class	64.2%	64.8%	0.5%	-0.8%
300/class	65.6%	67.0%	1.5%	1.2%
350/class	65.9%	67.0%	1.1%	1.0%
400/class	65.9%	66.8%	0.9%	0.4%
450/class	66.4%	67.9%	1.5%	-0.2%
500/class	68.1%	68.5%	0.4%	N/A

Generative data augmentation was beneficial for both the original dataset and every subset. The classification performance of the baseline model for all subsets and the original dataset was increased as expected. Moreover, for some specific cases, generative data augmentation was more beneficial than the addition of real images for all subsets except 50 and 100 samples per class. The highest performance improvement for these cases was higher than the subsequent real data subset.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

The deep learning models tasked with image classification were trained to estimate the Endoscopic Mayo Score (EMS) of the inflammatory bowel disease (IBD) Ulcerative Colitis (UC). The training was accomplished with the dataset consisting of colonoscopy images. Several subsets containing an equal number of images per class were also formed.

ResNet18 classification model was trained to perform over this dataset and subsets to classify images according to EMS. This classification performance was improved through classical deep learning approaches as discussed in reference works. Image classification performance of the deep learning model using each subset and the original dataset were evaluated and compared. Each subset performed better than the previous subset, and the original dataset had the highest classification performance.

GAN architecture of StyleGAN2-ADA was employed for generative data augmentation in this thesis work. Synthetic images were supplied to ResNet18 models along with real images. With different amounts of synthetic images injected in classification training, the performance affects of generative data augmentation were evaluated.

The classical data augmentation methods were also proved useful for improving deep learning model performance as expected. Both classical and generative data augmentation methods were used together to obtain higher performance scores compared to the baseline performance.

It was expected to observe a performance improvement with generative data augmentation. The class-conditional GAN, that included class information in training, were more successful for the original dataset. The second type of GAN trained in this thesis, the class-specific GAN with weight initialization through transfer learning from the FFHQ dataset, was more beneficial for the classification score for balanced subsets.

A summary for the highest reported performance increase of all experiments in this thesis can be observed from Table 12. Generative data augmentation with images synthesized by GAN trained with transfer learning had generally improved the baseline performance better than images synthesized by class-conditional GAN. The training method of class-conditional GAN was more successful for generative data augmentation when data is more abundant. This is evident in the experiments with subsets, where class-conditional GANs increased subset baseline performance by 2.4% on the average. In comparison the class-specific GAN increased subset baseline performance by 3.8% on the average.

Generative data augmentation with class conditional GAN improved baseline performance of the original dataset by 3.1% and the experiments were consistent. For the class-specific GAN however, the experiments on the original dataset resulted in conflicting results where only one truncation option proved to be beneficial while others were failed to increase the baseline performance reliably.

Table 12: Highest reported performance increase (%) with generative data augmentation over baseline real images only experiments.

	GAN Type	
Dataset	**Specific**	**Conditional**
Subset:50	3.1	**3.3**
Subset:100	3.1	**3.5**
Subset:150	**4.1**	1.2
Subset:200	**4.0**	1.3
Subset:250	**2.8**	1.9
Subset:300	**4.5**	2.9
Subset:350	3.6	**3.9**
Subset:400	**4.7**	2.5
Subset:450	**5.4**	2.7
Subset:500	**3.0**	0.8
Original	2.7	**3.1**

For average performance improvement with generative data augmentation, the table 13 demonstrates the class-conditional and class-specific GAN contribution over baseline subsets. On the average, the class-specific GAN results were consistently better than class-conditional GAN results.

Table 13: Average performance increase over baseline and subsequent subsets. GAN training methods of class-specific and class-conditional are compared

	Average Performance Increase			
	Class-Conditional GAN		**Class-Specific GAN**	
Subset	**Over Baseline**	**Over Real**	**Over Baseline**	**Over Real**
50/class	**0.1%**	**-0.7%**	-0.1%	-4.0%
100/class	0.1%	**-1.4%**	**0.3%**	-2.0%
150/class	-0.1%	-1.5%	**1.2%**	**-0.1%**
200/class	-0.5%	-1.6%	**0.3%**	**-0.6%**
250/class	-0.3%	-0.8%	**0.5%**	-0.8%
300/class	0.1%	-0.1%	**1.5%**	**1.2%**
350/class	1.0%	0.6%	**1.1%**	**1.0%**
400/class	0.6%	0.1%	**0.9%**	**0.4%**
450/class	0.5%	-0.6%	**1.5%**	**-0.2%**
500/class	-0.1%	N/A	**0.4%**	N/A

Because of different results obtained through experiments with subsets and the original dataset, it can be concluded that GAN training should be tailored to the task at hand. With many training hyperparameters such as r1 regularization, GAN network size and ADA target, it is important to conduct trials to find the best settings. There are also parameters that are important for inference time such as truncation and number of synthetic images. As a result of the numerous experiments in this thesis, it can be concluded that the truncation value for generating synthetic images and the number of injected synthetic images are both decisive for classification model performance.

The main contribution of this thesis was to find empirical data for classification performance improvements with generative data augmentation. The experimental results demonstrated that generative data augmentation was indeed beneficial and in some cases even more useful than real images. The results of this thesis can be used to form a new approach on classification tasks with limited data.

7.2 Future Work

Generative Adversarial Network training requires delicate hyperparameter settings. One such important hyperparameter is the r1 gamma regularization parameter. The regularization parameter is one of the most critical parameters for deep learning models, which is also true for GAN training. A careful search of different r1 gamma regularization parameters can prove useful for better GAN performance.

Another vital factor for GAN training is the network size; too large or too small networks are not desired as their performance can vary drastically from the optimum sized networks. However, the optimum size of a network is not always easy to find, and experiments with different sizes can be conducted for GAN training.

Lower FID does not always correlate to better generative data augmentation. The quality of images generated by a GAN can please human eyes and be indistinguishable for human perception; however, the deep learning models work as a black box with no explicit definition of the hidden layers. The latent space of a deep model will benefit from a synthetic image generated by a very low FID score GAN model as well as by a higher FID score GAN model.

A metric of classification performance improvement can be formulated to determine the benefits of the addition of synthetic images. Generative data augmentation metrics can help with training and employing GANs. The designed metric can also be used with GAN training and even a loss function can be adapted to GAN training.

REFERENCES

[1] S. Ali, M. Dmitrieva, N. Ghatwary, S. Bano, G. Polat, A. Temizel, A. Krenzer, A. Hekalo, Y. B. Guo, B. Matuszewski, M. Gridach, I. Voiculescu, V. Yoganand, A. Chavan, A. Raj, N. T. Nguyen, D. Q. Tran, L. D. Huynh, N. Boutry, S. Rezvy, H. Chen, Y. H. Choi, A. Subramanian, V. Balasubramanian, X. W. Gao, H. Hu, Y. Liao, D. Stoyanov, C. Daul, S. Realdon, R. Cannizzaro, D. Lamarque, T. Tran-Nguyen, A. Bailey, B. Braden, J. E. East, and J. Rittscher, "Deep learning for detection and segmentation of artefact and disease instances in gastrointestinal endoscopy," *Medical Image Analysis*, vol. 70, p. 102002, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841521000487

[2] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pp. 1322–1328, 2008.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002. [Online]. Available: https://arxiv.org/pdf/1106.1813.pdf%0Ahttp://www.snopes.com/horrors/insects/telamonia.asp

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 3, 6 2014.

[5] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *NIPS*, 2016. [Online]. Available: http://arxiv.org/abs/1701.00160

[6] A. Madhu and S. Kumaraswamy, "Data augmentation using generative adversarial network for environmental sound classification," *European Signal Processing Conference*, vol. 2019-Septe, pp. 3–5, 2019.

[7] X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: A review," *Medical Image Analysis*, vol. 58, 2019.

[8] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, 2019.

[9] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," *arXiv preprint arXiv:1711.04340*, 2017.

[10] K. B. Poka and M. Szemenyei, "Data augmentation powered by generative adversarial networks," *2020 23rd IEEE International Symposium on Measurement and Control in Robotics, ISMCR 2020*, pp. 6–10, 2020.

[11] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using gan for improved liver lesion classification," *Proceedings - International Symposium on Biomedical Imaging*, vol. 2018-April, pp. 289–293, 2018.

[12] H.-C. Shin, N. A. Tenenholtz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andriole, and M. Michalski, "Medical image synthesis for data augmentation and anonymization using generative adversarial networks," *International workshop on simulation and synthesis in medical imaging*, pp. 1–11, 2018.

[13] H. Rashid, M. A. Tanveer, and H. A. Khan, "Skin lesion classification using gan based data augmentation," *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, vol. 2019, pp. 916–919, 2019.

[14] D. Yorioka, H. Kang, and K. Iwamura, "Data augmentation for deep learning using generative adversarial networks," *2020 IEEE 9th Global Conference on Consumer Electronics, GCCE 2020*, pp. 516–518, 2020.

[15] K. D. B. Mudavathu, M. V. S. Rao, and K. V. Ramana, "Auxiliary conditional generative adversarial networks for image data set augmentation," *Proceedings of the 3rd International Conference on Inventive Computation Technologies, ICICT 2018*, pp. 263–269, 2018.

[16] D. Yoon, H.-J. Kong, B. S. Kim, W. S. Cho, J. C. Lee, M. Cho, M. H. Lim, S. Y. Yang, S. H. Lim, J. Lee, J. H. Song, G. E. Chung, J. M. Choi, H. Y. Kang, J. H. Bae, and S. Kim, "Colonoscopic image synthesis with generative adversarial network for enhanced detection of sessile serrated lesions using convolutional neural network," *Scientific Reports*, vol. 12, pp. 1–12, 2022. [Online]. Available: https://doi.org/10.1038/s41598-021-04247-y

[17] C. C. Aggarwal, *The Basic Architecture of Neural Networks*. Springer, 2018.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[19] R. Indrakumari, T. Poongodi, and K. Singh, *Introduction to Deep Learning*. Springer, 2021.

[20] G. Cottrell, P. Munro, and D. Zipser, "Learning internal representations from gray-scale images: An example of extensional programming," *In Proceedings of the Ninth Annual Cognitive Science Society Conference*, pp. 461–473, 1 1987.

[21] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[22] Y. LeCun. (2017) Quora session with yann lecun. [Online]. Available: https://quorasessionwithyannlecun.quora.com

[23] A. Jabbar, X. Li, and B. Omar, "A survey on generative adversarial networks: Variants, applications, and training," *ACM Computing Surveys*, vol. 54, pp. 1–38, 2022.

[24] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pp. 1–16, 2016.

[25] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *Proceedings of the 34th International Conference on Machine Learning, PMLR*, 2017. [Online]. Available: http://arxiv.org/abs/1701.07875

[26] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[27] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, 2019.

[28] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 8107–8116, 2020.

[29] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," *arXiv*, 2020.

[30] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[31] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, 2017.

[32] G. Polat, E. Isik-Polat, K. Kayabay, and A. Temizel, "Polyp detection in colonoscopy images using deep learning and bootstrap aggregation," *IEEE International Symposium on Biomedical Imaging*, 2021. [Online]. Available: http://ceur-ws.org

[33] G. Polat, I. Ergenc, H. T. Kani, Y. O. Alahdab, O. Atug, and A. Temizel, "Class distance weighted cross-entropy loss for ulcerative colitis severity estimation," *arXiv preprint arXiv:2202.05167*, 2022.

[34] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[35] Pep 0 – index of python enhancement proposals (peps). Accessed:26 December 2021. [Online]. Available: https://www.python.org/dev/peps/

[36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[37] T. Karras and J. Hellsten. Stylegan2-ada - official pytorch implementation. Accessed:26 December 2021. [Online]. Available: https://github.com/NVlabs/stylegan2-ada-pytorch

[38] L. Biewald. (2020) Experiment tracking with weights and biases. Software available from wandb.com. [Online]. Available: https://www.wandb.com/

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, 2016.

[40] H. T. Kani, I. Ergenc, G. Polat, Y. O. Alahdab, A. Temizel, and O. Atug, "P099 evaluation of endoscopic mayo score with an artificial intelligence algorithm," *Journal of Crohn's and Colitis*, vol. 15, pp. S195–S196, 5 2021. [Online]. Available: https://doi.org/10.1093/ecco-jcc/jjab076.227

[41] E. Alpaydin, *Introduction to Machine Learning*. The MIT Press, 2014.

[42] M. Kubat, *An Introduction to Machine Learning*. Springer, 2017.

[43] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf

[44] L. N. Vaserstein, "Markov processes over denumerable products of spaces, describing large systems of automata," *Problemy Peredachi Informatsii*, vol. 5, no. 3, pp. 64–72, 1969.

[45] G. Polat, D. Sen, A. Inci, and A. Temizel, "Endoscopic artefact detection with ensemble of deep neural networks and false positive elimination," *IEEE International Symposium on Biomedical Imaging*, 4 2020.

[46] NVlabs. Nvidia stylegan2 pretrained models. Accessed:26 December 2021. [Online]. Available: https://catalog.ngc.nvidia.com/orgs/nvidia/teams/research/models/stylegan2

[47] Hesaplama kümeleri - truba kullanıcı dökümanları belgelendirmesi. Accessed:26 December 2021. [Online]. Available: https://docs.truba.gov.tr/TRUBA/kullanici-el-kitabi/hesaplamakumeleri.html

Appendix A

DETAILED TABLES FOR SATURATION EXPERIMENTS

A.1 Saturation Experiments

The details of the subset classification performance saturation experiments are given in Table 14. The F1 score and F1 Remission score details are also provided with Table 15.

Subset	µ Accuracy	µ Precision	µ Specificity	µ Sensitivity	µ F1
50	64.12%	57.84%	86.22%	57.15%	55.77%
100	67.67%	60.52%	87.29%	60.91%	59.64%
150	70.01%	61.89%	88.35%	63.55%	61.99%
200	70.33%	63.71%	88.78%	65.01%	63.31%
250	71.46%	64.56%	88.92%	65.27%	64.24%
300	72.33%	65.38%	89.41%	67.30%	65.56%
350	72.19%	65.73%	89.41%	68.03%	65.88%
400	72.48%	66.04%	89.48%	67.39%	65.93%
450	72.80%	66.75%	89.52%	67.78%	66.42%
500	73.68%	67.13%	90.00%	69.88%	68.08%

Table 14: The table shows classification performance details of subset saturation experiments. Each new real image addition improved overall performance metrics, with the worst performance observed at 50 images per class subset. Furthermore, the highest score was obtained by 500 images per class subset.

Subset	µ F1	µ F1r	M F1	std F1	M F1r	std F1r
50	55.77%	70.15%	56.69%	4.00	70.81%	3.42
100	59.64%	74.94%	59.77%	2.91	75.90%	2.60
150	61.99%	77.93%	62.05%	1.77	78.64%	2.25
200	63.31%	77.98%	64.32%	5.18	79.68%	6.76
250	64.24%	78.61%	64.20%	2.99	79.39%	3.63
300	65.56%	79.96%	66.56%	4.13	81.04%	3.18
350	65.88%	80.13%	66.67%	3.99	81.57%	4.54
400	65.93%	79.37%	66.88%	4.43	81.12%	3.20
450	66.42%	80.60%	66.77%	3.39	81.51%	4.57
500	68.08%	81.43%	68.93%	3.28	81.89%	1.93

Table 15: Details about F1 and F1r (Remission) scores are shown in the table. Mean and median values, as well as the standard deviation of both scoring methods, are shown.

Appendix B

FAILURE ANALYSIS

B.1 Failure Analysis

EMS 0 class had 11 problematic images out of 772 images which were always incorrectly classified as EMS 1. Four of these images are shown in Fig. 32.

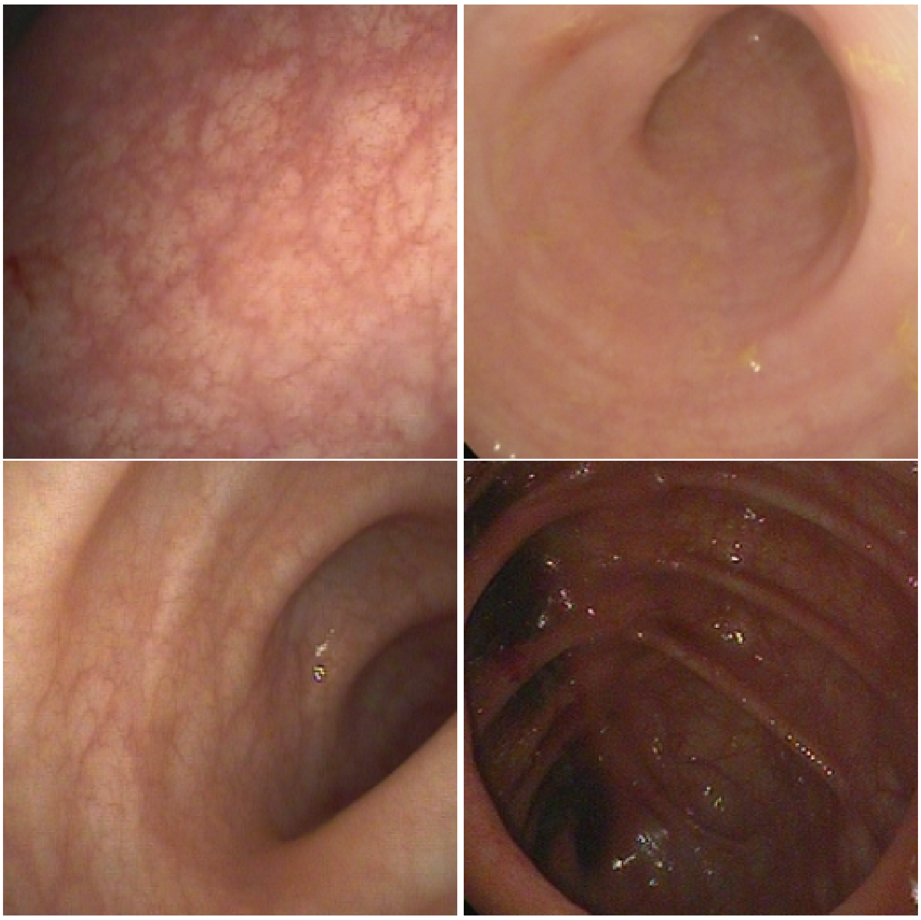

Figure 32: EMS 0 Images incorrectly classified as EMS 1 class.

Similarly EMS 1 class had 43 problematic images out of 406 images which were always incorrectly classified as EMS 0. Four of these images are shown in Fig. 33. It is important to note that the ratio of incorrect classification to the number of images increase dramatically from EMS 0 problematic images.

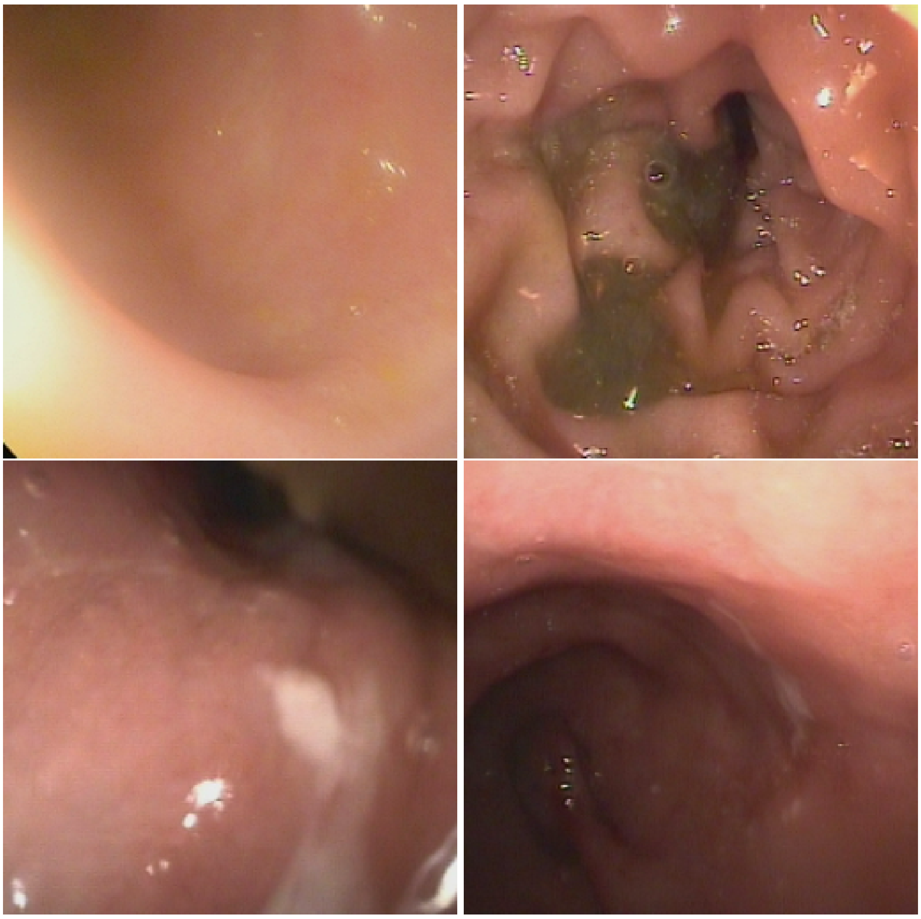

Figure 33: EMS 1 Images incorrectly classified as EMS 0 class.

There were 16 problematic images for the EMS 2 class out of 146 images. Three images out of 16 problematic images were always classified as EMS 0 class, 11 as EMS 1 class, and two as EMS 3 class. The ratio of problematic images and the sample size is similar with EMS 1 class. Examples for these problematic images are shown in Fig. 34

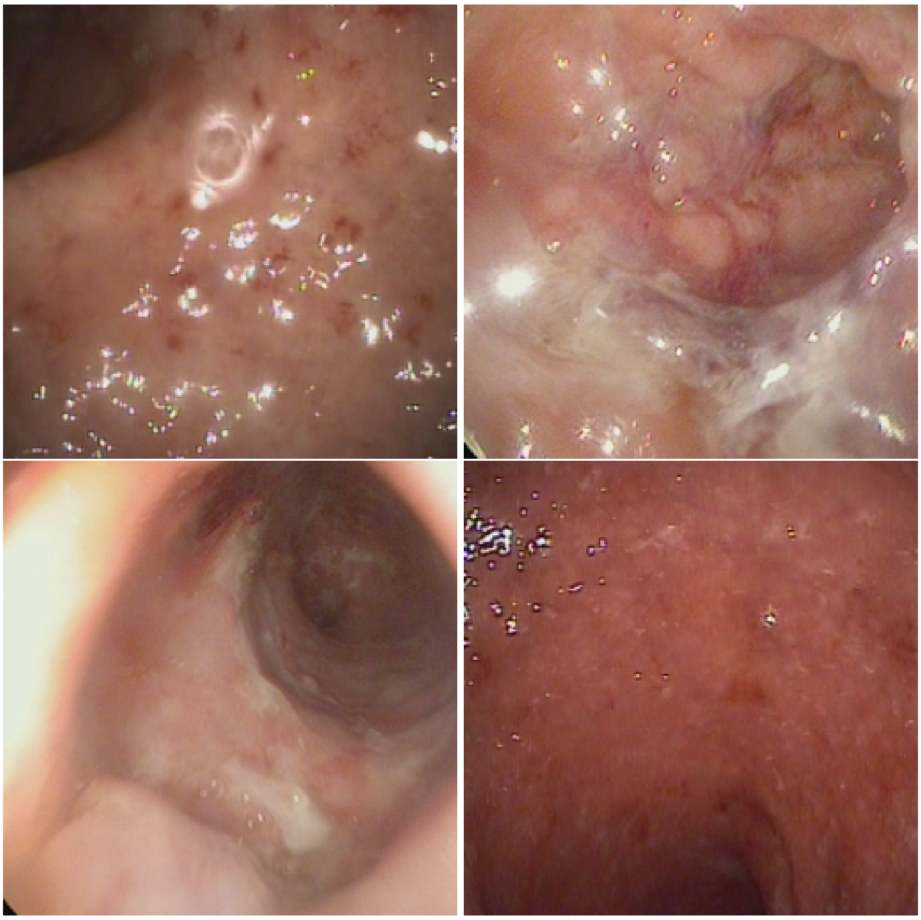

Figure 34: EMS 2 Images incorrectly classified as other EMS classes.

Finally there were only one problematic image for EMS 3 class as shown in Fig. 35. It was classified as EMS 2 class.

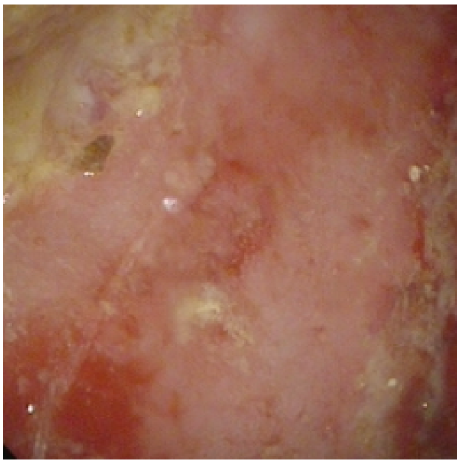

Figure 35: The problematic EMS 3 Image incorrectly classified as other EMS 2 class.

Appendix C

ENERGY CONSUMPTION

C.1 Energy Consumption of the Experiments

The experiments in the thesis consist of two major model training. First, the Generative Adversarial Networks were trained with limited data subsets and a full dataset. Second, the image classification network of ResNet18 was trained to evaluate classification performance with and without generative data augmentation.

Overall training duration of this thesis work was 397.4 days on a single A100 GPU. With the help of parallel training infrastructure provided by TRUBA, the training has been completed in a shorter time. 222.4 days corresponding to 56% of the total training duration were allocated to GAN training. 175 days corresponding to 44% of the total training duration were allocated to ResNet18 Classification network training. 48.7 days of training were omitted from the final results as these training were done with an obsolete dataset containing severe flaws. In the end, 348.7 days of training could be used as the final results, i.e., 12% of the training were discarded, resulting in 88% efficiency in training runs.

The majority of the training was done on NVIDIA A100-SXM-80GB GPUs, while some initial training was conducted on NVIDIA GTX3080. Initial classification model and GAN training with the old dataset with 7.1 and 41.6 days, respectively, were not used in final reports. This training can be considered proof of concept work, and no more contribution from this training can be validated as the dataset used in them was older and contained serious flaws such as label conflicts and artificial instruments. Details about the training cost of this thesis work can be observed from Table. 16

The computation cost of this thesis was measured in time and electricity. While time cost was directly logged and monitored, the electricity cost was only approximated as the real cost of a high performance computing (HPC) operation can vary depending on the jobs submitted to a node, and every node will most probably serve different jobs at the same time. The GPU power utilization percentage of classification network training and GAN training were in the range of 70% to 80% of the rated power per GPU. The palamut node of TRUBA HPC was equipped with NVIDIA A100 GPUs with a rated power of 300 Watts.

Table 16: Experimentation training costs. GAN training was done with four A100 GPUs, initial training was done on a single NVIDIA GTX3080, and all other classification model training was done on a single A100 GPU. GAN training duration was converted to a single GPU by scaling with the number of GPUs used.

CLASSIFICATION TRAININGS			
Configuration	**Experiment Type**	**Training Duration (Days)**	**Number of Experiments**
Initial Dataset	Subset saturation	2.3	1995
	Subset generative augmentation	2.6	1222
	Dataset generative augmentation	2.2	2050
	Total initial trainings:	**7.1**	**5267**
New Dataset	Subset generative augmentation	17.0	8697
	Subset saturation	9.9	2629
	Dataset generative augmentation	4.4	1237
	Subset saturation	11.8	2563
New Dataset + Best FID	Subset generative augmentation	8.5	2298
	Dataset generative augmentation	17.8	1596
New Dataset + Best FID + Pretrained GANs	Dataset generative augmentation	2.7	324
	Dataset generative augmentation	14.1	938
	Subset generative augmentation	74.5	10152
	Subset synthetic only	7.3	395
	Total classification trainings:	**167.9**	**30829**
GAN TRAININGS			
Class Conditional	Old Dataset GAN training	41.6	5
	New Dataset GAN training	78.8	11
Transfer Learning	GAN transfer learning training	102.0	56
	Total GAN trainings:	**222.4**	**72**
GRAND TOTAL TRAININGS:		**397.4**	**36168**

According to the TRUBA HPC resource distribution rules [47], each GPU has to be accompanied by 16 CPU cores. The CPU model of the HPC was AMD EPYC 7742, which has 64 cores and 225 Watt rated power. The GAN and classification networks utilized around 10% CPU throughout their training.

According to the rated power data of CPU and GPU, the rated power of GAN training was found to be 262.5 Watts and the rated power of classification network training to be 232.5 Watts. The total electricity consumption for all classification networks was 976 kilowatt-hours (kWh), and for GAN training, 1401 kilowatt-hours (kWh). The grand total electricity cost of the comprehensive training in this thesis was approximately 2.4 megawatt-hours (MWh).