Manifold Learning of Latent Space Vectors in Generative Adversarial Networks for Image Synthesis.

Article · May 2020

CITATIONS
0

READS
706

1 author:

Mohamed Ayoob
University of Westminster
2 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Informatics Institute of Technology
In Collaboration With

University of Westminster, UK

University of Westminster, Coat of Arms

Manifold Learning of Latent Space Vectors in GAN for Image Synthesis.

A dissertation by
Mr. Mohamed Ayoob

Supervised by
Mr. Guhanathan Poravi

Submitted in partial fulfillment of the requirements for the
BEng (Hons) Software Engineering Degree Department of Computing

May 2020

You can never stop the rain, but you can always carry an umbrella.

I dedicate this to my mother who's always believed in me, my father who's worked hard to for my wellbeing, my teachers who's been an inspiration to me in ways even they do not know, my brother for supporting me and to all the people who made my life special.

Abstract

Generative Adversarial Networks (GAN) have attained remarkable results in image augmentation and generation. GAN models consist of Latent space vectors. Latent space vectors are interpreted as inputs for the generator to generate images. Mathematically they are a hypersphere consisting of many dimensions, containing the learnt representation. However, various GAN variants use different techniques to model latent space vectors. This dissertation dissects the latent space vectors of state-of-the-arts GAN variants, for image synthesis. We also compare and contrast optimization methods of latent space vectors in existing literature. In this dissertation, we undertake a comparative analysis of latent space vectors and optimization methods of the popular image synthesizing GAN models.

Based on the analysis we were able to choose appropriate existing works to carry out the research. We choose the DCGAN as the model and ClusterGAN as the model optimization technique. Based on a review of mathematical methods we choose manifold learning techniques to cluster the latent space. The results of the clustering allowed us to determine the sparse and dense features of a Euclidean image distribution.

The research on the image augmentation techniques allowed the authors to make a deployable facial recognition system utilizing advanced image models. While there are other facial recognition models, they are not deployable in a business environment, they are procedural in nature. The authors had to review existing work on face recognition such as FaceNet model to make the system. The authors also used concurrent programming principles to optimize the model to train in reasonable amounts of time. This will be useful with pandemics such as the COVID-19 as well. We need more robustness in non-invasive (touchless) authentication systems.

Keywords: Image Generative Model; Generative Adversarial Networks; Latent Space Vectors; Mathematical optimization; Image synthesis, Face recognition systems.

Declaration

This dissertation is the result of my own work and includes nothing, which is the outcome of work done in collaboration except where specifically indicated in the text. It has not been previously submitted, in part or whole, to any university of institution for any degree, diploma, or other qualification.

I, Mohamed Ayoob of Informatics Institute of Technology, being a candidate for the B.Eng. (Hons) in Software Engineering, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

Full Name of Student: Mohamed Ayoob
Registration No: 2016343 / w1654551

Signature: ***Mohamed Ayoob*** Date: 6th of May, 2019

Acknowledgement

I consider myself fortunate and I am grateful to a great number of people who made my life special. I would not have been able to complete this research without the guidance and support of the people who was there to reach out and help me bloom. I want to thank my project supervisor, Mr Guhanathan Poravi. It was with his guidance and directions, I was able to grasp what research in computer science means. I would like to express my sincere gratitude to him. If not for his ceaseless reviews on the project topics I would have been piecing together software like building blocks. His research paradigms showed me to critically review existing work. I would also like to express my sincere thanks to the management of my campus, my lecturers, and my colleagues for the support and encouragement during this project. Finally, I would like to thank my parents and brother, they have been the most supportive people in life.

Publications

1. A Survey of Latent Space Vectors in Generative Adversarial Networks for Image Synthesis

Conference: 5th International Conference on Robotics and Vision

Hosted by: Wadham College, University of Oxford, United Kingdom

Publication: ICRV-ACM conference proceedings

Content: Survey of the existing GAN Latent Space Vectors in terms of design, architecture, features and performance.

Table of Contents

List of Tables

List of Figures

CHAPTER 1 : INTRODUCTION

1.1 Chapter Overview

This introductory chapter provides the reader an overview of the project undertaken. Initially the project circumstances are defined and the research context are explicitly established. The context of the research challenge as well as a use of the research product is explored. The research objectives, learning outcomes, operational objectives are established firmly. Since the research project is focused only on a component of an architecture known as the Generative Adversarial Network, the scope of the project is fixated within here as well. This chapter will culminate with the discussion on the resource requirements of this project.

In summary this chapter discourses on the contributing factors that inspired the author to undertake research in this topic.

1.2 Problem Background

Works done on optimizing latent space vectors of Generative Adversarial Networks are a niche in academia. However, the important of carrying out such a research has been outlined in many recent papers and journal articles. This proposal will be addressing the gaps of knowledge pertaining to the use of manifold learning to optimise the n-dimensional latent space vectors. This proposal also asserts to ratify the insight obtained from the research to further strengthen the robustness of person identification systems.

1.2.1 Artificial Intelligence

Artificial intelligence computer programs have gained plenty of traction and progress in the recent years. That's due to the reduction in the cost of compute power and increased scrutiny of AI methods by researchers (Omar et al., 2017). According to (Bonabeau et al., 1999) the applications are of Artificial Intelligence are expected to be based on structures and algorithms inspired by biology in addition to systems inspired mathematical conjectures. Technologies and frameworks for collective intelligence and Swarm Intelligence is expected to soar in industries like retail, defense, and transportation.

According to many researchers and endorsed by Ray Kurzweil's Book "The Singularity Is Near: When Humans Transcend Biology" (Kurzweil, 2006) the direction that the domain of Artificial Intelligence is geared to is the fusion of empirical spheres of mathematics biology and neurotechnology to inspire "superintelligence" in computing machines. Hence, comes the quintessential query of this research, **what results can be obtained by clustering the latent space vectors in a GAN?**

1.2.2 Machine Learning and Deep Learning

Machine learning and deep learning are derivatives of the broad umbrella domain of AI. They are also one of the most branches of AI in both academia and the industry alike. Machine learning is

defined best by (Simon, 2013) in his book "Too Big to Ignore: The Business Case for Big Data". He describes it as "field of study that gives computers ability to learn without being explicitly programmed". Deep learning is loosely synonymous with how the neuron cells in the brain works and no doubt was the inspiration for the ground-breaking paper in Deep Learning that positively impacted many researches and industry applications (LeCun et al., 2015). These insights gave rise to novel neural network models such as Convolutional Neural Networks, Recurrent Neural Networks as well as the ability to ingress images, sounds and even LiDAR point clouds. Significant Researches such as YOLO model (Redmon et al., 2015) have been undertaken to analyze and provide meaningful object detection context in real time on low powered computing devices.

In short however machine learning or deep learning can be described as fitting a line of best fit in a multi-dimensional hyperspace to find the best matrices of weight and biases that yields the least aggregate magnitude of error with respect to the loss function used in the model. This is done by the forward propagation and the back propagation of the weights and biases of neural networks as demonstrated here by Lecun, (1988). This led to the largest induction of neural networks in the industry ever known. It went on the be researched and used in various subfields that spawned its own challenges and solutions and went on to became to be used in the many specialized tasks in many fields.

1.2.3 Image Generative Models

Image Generative models are mechanisms of Deep Learning where we could generate new images based on the features of previously learnt images. According to the research done by Zeiler and Fergus (Zeiler and Fergus, 2013), on visualizing features of convolutional neural networks, they concluded the existence of a representation of learnt images within the inner layers of a model. DistillPub, a medium to present academic research (Odena, 2019) classifies 3 main types of image generative models.

1. Generative adversarial networks
2. Variational Auto Encoders
3. Flow based generative models

In addition to these a recent paper "Generating Long Sentences with Sparse Transformers" (Child et al., 2019) by OpenAI, introduces a new efficient generative model inspired by Transformer networks. This paper has been shown to have significant potential on generative applications in addition to the widespread acclimation of GANs networks. The following is the comparative analysis done on 3 generative models for images.

Models	**Parallel**	**Efficient**	**Reversible**
GANs	Yes	Yes	**No**
VAEs	**No**	Yes	Yes
Flow Based Model	Yes	**No**	Yes

Table 1.1: Comparison between existing models for image generation.

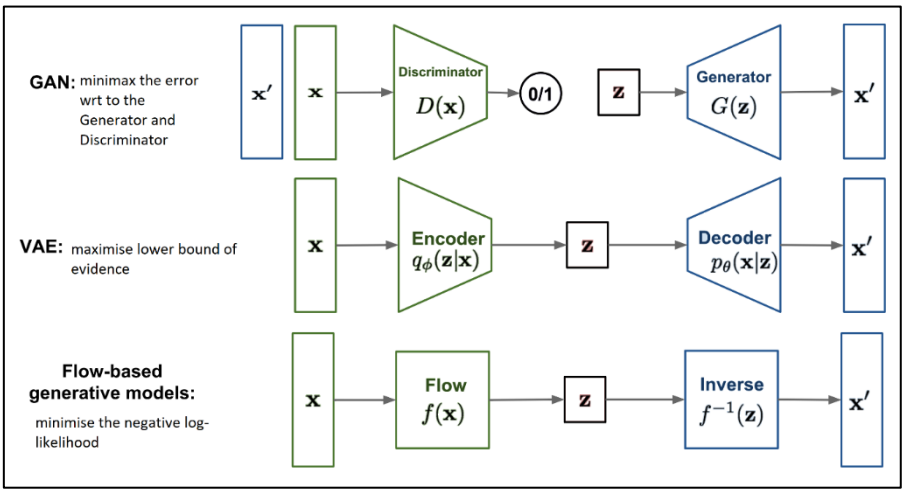


Figure 1.1: Architectural differences between 3 main generative models (Goodfellow et al., 2014a)

The above in Fig 2 is an architecture of the 3 main generative models experimented in the industry. Furthermore, below I have annotated the main difference in data ingress mechanisms and training mechanisms between the 3 models.

1. GANs – minimax the error with respect to the generator and discriminator.
2. VAEs – maximise lower bound of evidence.
3. Flow-based generative models – minimize the negative log-likelihood.

Cursory reading and reviews prove that for generative models one of the best to be used is GANs. Sparse Transformers are shown to have exceptional qualities in the recent paper (Child et al., 2019), however this proposal will not propose to conduct a qualitative study on Sparse Transformer models as it's a recent addition and not well researched due to being a recent introduction in the field of generative modelling.

This proposal focuses on Generative Adversarial Networks. GANs networks have been used extensively by the industry.

1.3 Problem Domain

Generative Adversarial Networks (GANs) has shown exceptional performance in many areas and tasks. In the article (Brownlee, 2019) discusses of 18 tasks that GANs are being actively used. We could summarise those tasks as follows, GANs model is being used for Image generation (Karras et al., 2017), image-to-image translation (Isola et al., 2016), image super resolution (Ledig et al., 2016) to name a few. While most of them use a variant of the original GANs network, the sole object and the architecture of the network remains the same.

The rich picture of the project would be employing techniques to optimize the latent space of GANs to get better and controlled image synthesis. The following is a succinct and concise study of the original GANs network.

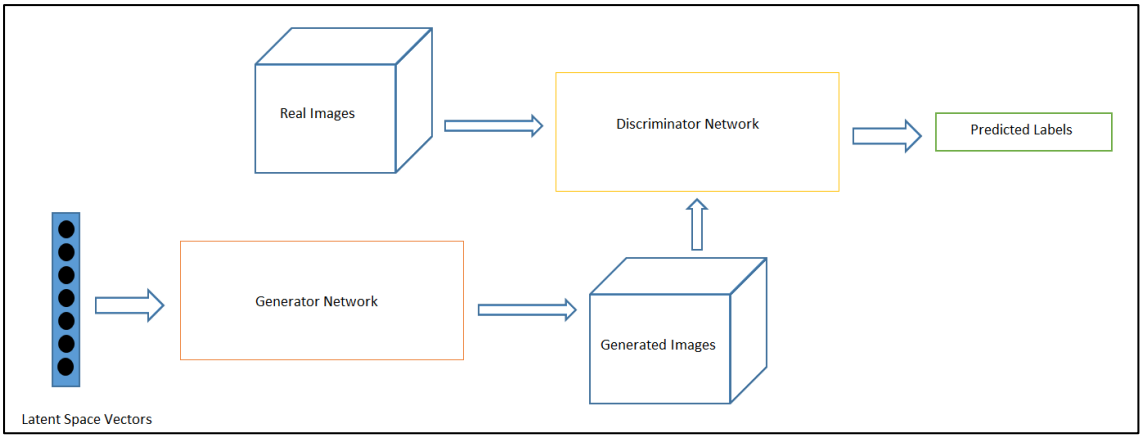


Figure 1.2: GANS architecture and rich picture of the latent space vectors (Goodfellow et al., 2014a)

Above illustrated is the original GANs architecture proposed. The GANs network consists of 2 adversarial models playing a minimax game based on Game Theory in mathematics. This ensures that rather than using the cross entropy to calculate the scaled difference between predicted and real label, GANs uses a minimax loss function that ensures that both the networks are trained in parallel.

According to the paper (Karras et al., 2018), which was published as late as March 2019 the Latent Space Vectors in GANs is still underexplored and poorly understood. Latent Space Vectors exploration and optimization techniques will be explored in this research.

1.4 Related Work

There has been progressing interest by researchers in the field of latent space vectors. We can broadly classify the works on done latent space vector optimization in 2 approaches.

1) Image generative latent space optimization – pure image generative models
2) Conditional Image generative latent space optimization – processed image generative models (text-to-image, image-to-image etc.)

1.4.1 Quantitative analysis on related works in Image Generative latent space vector optimization techniques

Perhaps the most preliminary work involving latent space manipulation was done by Radford (Radford et al., 2015) in their paper “Unsupervised Representation Learning with deep convolutional generative adversarial networks”. They introduced DCGAN, which was able to successfully manipulate the latent space vectors of human faces.

Following the paper on InterFaceGAN (Shen et al., 2019) in 2019 we saw even more distinct manipulations of the latent space, that when changed accordingly was able to edit the characteristics of the image such as age, gender, smile and pose, corresponding to the latent space vector that is being manipulated. Similarly, GLO networks (Bojanowski et al., 2017) were proposed parallel to the InterFaceGAN, it concluded that the latent space vectors are a seldom studied subject in GANs network. Several other works have demonstrated a generic use case of latent space manipulation to features specific to the dataset their models were trained.

To properly understand latent space vectors manipulation, I would like to take on a preliminary study on how certain works have influenced the mapping between the text caption of an image to the features of the image exhibited in the latent space vectors. This would ensure distinctiveness in research. Understanding their approach would be important to make sure that my domain of research doesn't have any overlaps with already existing published works.

Research	Technique Used	Improvements	Limitations
InterfaceGAN (Shen et al., 2019)	Leverages the semantics hidden in the latent space of GANs	Conditional manipulation used to further dissect independent attributes	Trial and error process of attribute editing.
GLO (Bojanowski et al., 2017)	Introduce a new error function to the latent space vectors to reduce reconstruction error.	More sophisticated sampling methods after training will improve visual quality.	Trivial sampling methods used, and empirically tested with only few loss functions.

Table 1.2: Comparison of latent space optimization and visualization

1.4.2 Preliminary Study on Latent Space vector Mapping from text to image

Works from text to image include mapping objects from the latent space for a text classifier to the latent space vectors in the image classifiers. StackedGAN introduced by (Huang et al., 2016) in their paper serves as a frontline in the text-to-image generation. The implementation can be described as two GANs models which are stacked on each other that maps the image caption vectors along with the image feature vectors.

Paper on "Text-Adaptive Generative Adversarial Networks: Manipulating Images with Natural Language" by (Nam et al., 2018) was based on Stacked GANs however it used a text based discriminator that creates its own local discriminators. This particular component enables the models to generate both coarse and fine visual features by separating the scales of the attributes in the latent space.

Now that we have a brief but clear understanding on how the latent spaces are designed with different systems of generative adversarial network, we can conclude that the latent spaces are treated as Riemannian manifolds, however they haven't been treated as diffeomorphic groups on or in manifold learning techniques.

Research	Year	Technique Used	Improvements	Limitations
StackedGAN (Huang et al., 2016)	2019	Stacked 2 GANs networks recursively. There are 2 networks working in conjunction with the minmax loss function	Decomposes the problem of approximating image distribution into multiple representations.	2 GAN networks required to synthesize images. Computationally expensive.
TAGAN (Bojanowski et al., 2017)	2017	Opted for a text adaptive discriminator, in the Discriminator network of GANs.	Successfully enable images to be manipulated semantically to the corresponding word.	No specified limitations.

Table 1.3: Comparison of complex latent space compositions

1.5 Problem Definition

Latent Space Vectors are a module in Generative Adversarial Networks. They are a N-Dimensional Hyperspace all of which are vectors. Initially they are initialized randomly. However, after the training process of the GANs network (ie- when the Discriminator function outputs values ideally close to 0.5) the Latent Space Vectors are clustered in the inter-dimensional space corresponding to features they are designated in the learning process. Several attempts were made on exploring the latent space vectors. Particularly in the work done by Radford, A., Metz, L., Chintala, S., (2015), the authors concluded the feasibility of hyperspace exploration, and specifically examined vector arithmetic of learned vectors to make various interpolations of vectors that was instrumental in facial feature representation. Similarly works by Zhu, J.-Y., Krähenbühl, P., Shechtman, E., Efros, A.A., (2016) observes how the results of the latent space vectors change correspondingly with their Latent Space Vectors.

1.5.1 Problem Statement

Latent Space Vectors in Generative Adversarial Models are still yet and underexplored object of study. According to the few published researches (such as above) conclude that the latent space vector manipulation and exploration is a dire need to fully understand and utilize generative models in the academic research circles (Karras et al., 2018). Furthermore, Bojanowski et al., (2017) concludes that latent space vectors have not been tested and proven on empirical factors such as varying loss functions, different model architectures and progressive generation.

1.6 Research Motivation

Generative modelling using neural network architectures have been a seldom researched topic compared to the rest of Computer Science research. Furthermore, Generative Adversarial Networks as theorized in the original paper is (Goodfellow et al., 2014a) meant for a generation of researchers to come up with a plethora of research content. Latent space optimization is an under explored extension of GANs. The works of (Laine, 2018) on the paper "Feature Based Metrics For exploring the latent space of generative models", came to the conclusion that a dissection of the dimensions and geometry of the latent space is necessary and would greatly benefit from use of an external feature based metric.

The author of this research proposes to take on a mathematical approach to carry out the optimization using Manifold Learning techniques. I identified that the latent space vectors can be treated as diffeomorphic groups of numbers and I'm inspired by the works on abstract algebra .and clustering algorithms.

1.7 Research Contribution

This proposal proposes extend on the work done on the ClusterGAN by (Mukherjee et al., 2019), the possibilities for research has been mentioned in the discussion chapter of their paper. This proposal proposes to experiment on the relevance of various loss function, model architectures and progressive models on the latent space. This proposal also proposes to ratify the effectiveness of pruning the latent space vectors.

The application of this research would be able propose an open source easy to use person identification system with security features based on the learnt representations.

1.8 Research Question

These questions are part of a whole question and they are correlated with each other, they are not separate researches.

Q1: How are the Latent Space Vectors of various GANs varieties compare with each other? (Which are the most optimum to explore and interpolate? What effects do pruning them exhibit? , What would varying model architectures and loss functions exhibit in the Latent Space?)

Q2: How the Latent Space Vectors that is treated as a mathematical conjecture such as diffeomorphic group or Manifold Learning be optimized within the existing frameworks of the theory?

The Whole question would be conjectured as below.

Main Q: How can Image Synthesis be controlled by interpolating Latent Space Vectors, and which are the best GANs varieties for such interpolation after optimizing?

1.9 Research Aim

This research aims to design, develop and evaluate a person identification system robust to adversarial attacks by exploring the various properties of latent space vectors in GANs, and optimize it using Manifold Learning for controlled Image Synthesis. Furthermore, this research proposes to use the knowledge and intuition gained from the research to further stabilize person recognition systems.

To further elaborate on the aim, this research undertakes a review of LSVs of popular GANs networks, after exploring their respective properties and varying parameters of the neural networks used in the generator network the vectors would be subjected to interpolation to see the most semantically meaningful results based on the context of the dataset.

The acquired knowledge would be used to stabilize person recognition systems to make sure that the system is resistant to adversarial-like attacks and increase the convenience of the said system. The system will also be able to perform in a relatively low powered computer.

1.10 Research Objectives

Based on the research aim and research questions, the following objectives were defined for the project.

1.10.1 Research Objectives:

Research objectives are milestones to be reached to complete the research successfully.

Objective	Description
Provisional Literature Review	Carrying out an in-depth literature review of the following domains, • RO 1: Conduct a preliminary study on existing generative modelling techniques. • RO 2: Analyze the perception of generative modelling using generative adversarial networks in the deep and representation learning domain. • RO 3: Latent Spaces of Existing types of GANs networks. • RO 4: Perform a qualitative analysis on the types of GANs suitable for latent space optimization, interpolation and manipulation. • RO 5: Elaborate on the research gap on the existing latent space clustering. • RO 6: Conduct a preliminary study on multidimensional mathematical optimization techniques. • RO 7: Ratify viability of manifold learning as an optimization technique.
Experimental Ratification	Carrying out an in-depth experimentation and benchmarks on the existing solutions, • RO 1: Verify latent space with arithmetic vectors and review results • RO 2: Try interpolating with 3 or more points and review the results of the faces • RO 3: Update the GAN model for stability and review the quality of images.

	• RO 4: Update the latent space to use a larger or smaller latent space and compare the quality of results. • IMPORTANT: compare and benchmark speed of training in the above scenarios.
Design	Designing a mathematical conjecture to make manifold learning techniques to optimize the latent space • RO 1: Learn Abstract algebra. • RO 2: Experiment parallels between optimizing multi-dimensional spaces. • RO 3: Design a mathematical module to visualize individual latent space vectors.
Development	Developing a mathematical conjecture to optimize the latent space using manifold learning. • To develop the manifold learning visualization infrastructure and the geometric learning components to facilitate the cursory model. • To develop the latent space vector optimization framework. • To develop algorithms to compress the above models to acceptable inference speed and model size.
Evaluation Metrics	Ratification metrics for generative models include inception score and visual ratification and comparison. • RO 1: For a scientific study though the metrics of evaluation mentioned specifically for deep image generative models (Chen et al., 2017) will utilized. • RO 2: Produce structured and coherent literature for the academic community.

Table 1.3: Research Objectives

1.11 Project Scope

Based on the objectives of the research and review of existing works, we can surmise the project scope as follows. We are not including the data ingress, feature extraction, dimensionality reduction, and feature cleaning in the scope. All the preparatory steps and the extra features pertaining to each dataset are abstracted away. Here we will be focusing on the core functionalities and the research gaps in which the algorithms are to be developed.

1.11.1 In-scope:

The scope this research considers as focal points is as follows,

- Latent Space Vectors optimization, interpolation and manipulation of image generative models, in particular GANs.

- Reviewing latent space vectors of various GANs models (we have to make many educated guesses and conscious decisions)
- Varying loss functions, distinctive architectures and models to test and evaluate the system is an intensive undertaking, it will require loads of patience and intuition.
- Pruning the latent space vectors is a still unexplored field in the research arena, this is another empirical benchmark this would be undertaking.
- Disentangling the training process for the data collection process for developing the app interface.
- Optimization Latent Space Vectors using manifolds needs special attention, as it will be having to be explored from the domain of abstract algebra.
- Evaluating the results of the system based on existing metrics for generative models.

1.11.2 Out-scope:

The parts of the topic outside the focal point is listed below,

- The optimization is limited to Image generative models, and not audio, text or time series will be tested or evaluated. (NOTE: it may be tested and included in the thesis but will not be formally evaluated)
- While there are many types of GANs available we will be testing only few particular models. Models with promising results in image synthesis and few other interesting models that have reversibility as a virtue.
- Loss functions, model architectures and progressive models are also limited to those of that, which show promising results in Image Synthesis and not all of them will be tested and evaluated.

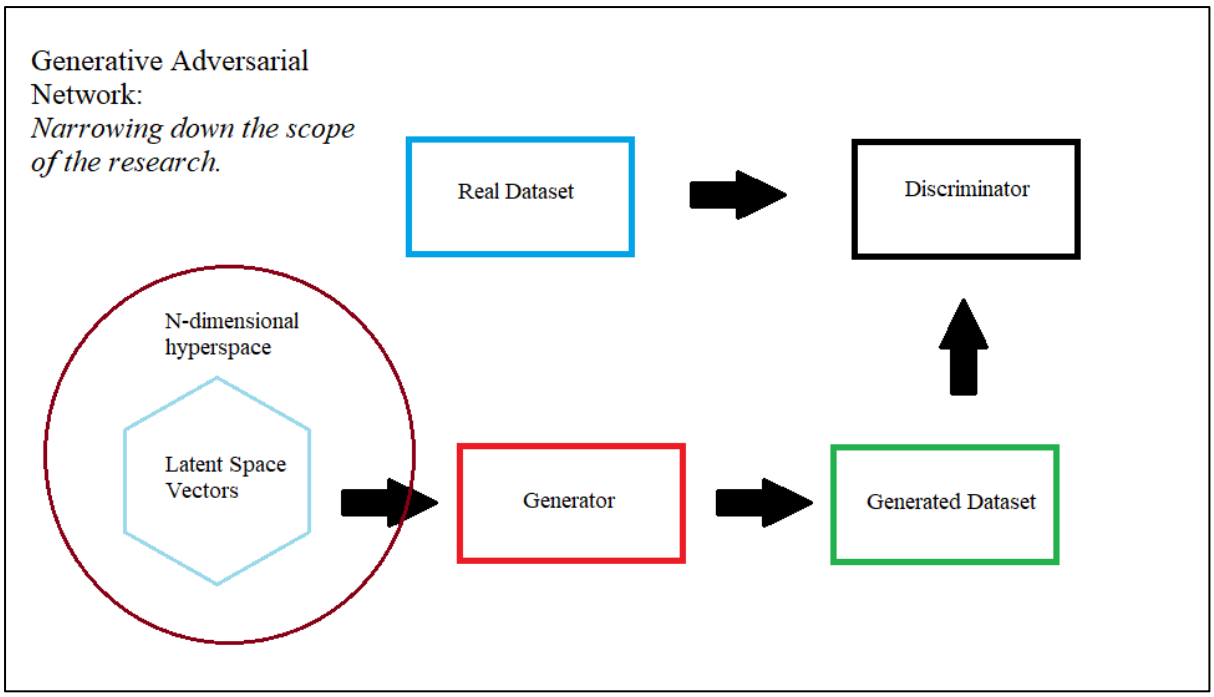


Figure 1.3: Scope of the project in the GAN architecture (circled)

1.12 Resource Requirement

Based on the software, hardware and data resources needed based on the development, testing and ratification Requirements are listed below.

1.12.1 Software Requirements

- Operating System – Linux has always been the performance savvy to DL crowd, hence a 64-bit version of Ubuntu 18.04 will be used.
- PyTorch – A researcher friendly pythonic framework for deep learning made by Facebook.
- Tensorflow GPU – A industry friendly framework for deep learning and development maintained by Google.
- PyTorch Geometric – A geometric deep learning framework for ratification and graph modelling.
- Zotero – Tool used to manage citations
- MS Office Package – To create reports and documentations.
- Google Drive – To backup files and datasets need for the project.
- Google Colabatory – To easily experiment models on the cloud without need for a powerful hardware.

1.12.2 Hardware Requirements

- Core i5 processor (9th Generation) – Perform intensive compute workloads.
- 16 GB RAM – Load datasets and keep them in memory
- Disk space of about 30 GB – Store the codebase and training datasets.

1.12.3 Data Requirements

- Open Image Dataset, Microsoft Open Dataset
- CIFAR 10 and CIFAR 100
- COCO Datasets and ImageNet
- University of California, Stanford Datasets
- ONNX Models repository
- Tensorflow Model hub

1.12.4 Skills Requirements

- Abstract Algebra and mathematical manifold
- Fluency in Deep Image Generative models
- Experience in evaluating generative models
- Experience in Graph Neural Networks
- Research writing skills

1.13 Document Structure

This dissertation discourses on the introduction to the problem, the grounds for the solution proposed, primordial project planning, research objectives, and evaluation strategies. Project scope definitions and literature review to identify research gaps are also detailed.

Chapter 2 documents the literature review. The existing researches and experimentation on latent space vector optimization, existing projects on image generative models, latent space clustering and model compression.

Chapter 3 documents the methodologies, paradigms of the research and the management of the project. The activity diagrams, scheduling work breakdowns, risk assessments, management and development methodologies are covered.

Chapter 4 details the requirements gathering plans followed. The different sources to gather requirement and surveyed requirements are analyzed.

Chapter 5 covers the design decisions of the project. Design plans, diagrams of class, sequence diagrams, domain, contexts, eagle eye overview of the architecture of research and the project are detailed.

Chapter 6 discourses on the implementation of the project. The languages and technologies utilized, development decisions taken, blockers faced, algorithms developed, and appropriate justifications for the implementation decision are documented. **Any deviations from the project design specifications are detailed here along with valid grounding for the said differences.**

Chapter 7 covers the testing of the research. Internal benchmarking, external benchmarking, unit tests, and integration tests are covered here.

Chapter 8 covers the evaluation of the system. Different evaluators review the project based on many empirical factors, the reasons for the evaluations, evaluations criterions, and discussions are covered.

Chapter 9 covers the concluding remarks of the author. Introspection on the research and the topics explored, ethical and social policies respected, and future works are detailed.

1.14 Reflection

The successful conclusion of this research would mean, the latent space vectors in a select few GANs networks would have an empirical study and review. Application of an extended manifold learning theory would be explored as well. Using the insights on latent space interpolation we would have a Open Source project for developers to develop person identification systems that are better equipped to adversarial attacks.

CHAPTER 2 : LITERATURE REVIEW

2.1 Chapter Overview

Image generative models have progressed immensely in the past few years. The last ten years have bought about more progress and innovation in the field than the last fifty years of computer science. Image generative models can be grossly classified into 2 main categories. Conditional Image generative models and unconditional image generative models. However recent researches have yielded another category of generative models called controlled image generative models. This chapter will be taking on a literature survey on the above various categories of image generative models. This chapter also will be taking into account a few mathematical conjectures when outlining any research gaps.

2.2 Concept Graph and Mind Map

The complete scope of the algorithm analysis defined in the literature review is displayed in a graphical form. This graph is a converging graph that takes a funnel-based approach. It begins with the broad artificial intelligence scope and converges to each sub domain in detail. We will be having an eagle eye view of the domains of artificial intelligence. This graph is found in **Appendix A.**

The seminal work of the researches that lead to this research will be defined in another graph as well. This details on the various researches that lead to the topic of my dissertation. This graph is appended in **Appendix B**

2.3 Literature Review of Image Generative Models

2.3.1 Introduction to Machine Learning and Generative Models

Machine learning is defined as “a set of algorithms that enable a computer to model and predict a dataset, with reasonable levels of accuracy in the predictions”. Machine learning spawned many researches using many different types of datasets. Currently machine learning is being used in myriads of tasks and industries. Now that we understand what machine learning is, let’s analyze neural models in machine learning.

Deep Learning became usable after a crucial paper on backpropagation was published by Yann Lecun. The paper discourses on the importance of having a concrete backpropagation framework. It details on previous work done on biological computing and neurological models on supervised learning (Lecun, 1988). The derivation of backpropagation algorithm spawned various researches in deep learning. Among the many contemporary offshoots of deep learning, one that has caught the attention of researchers in the present times is Image generation or Image synthesis models.

2.3.1.1 What are image generative models?

Image generative models are simply models that learn from a dataset of images and is able generate similar, meaningful images. One example is after going through all the 10 classes in the CIFAR10 dataset, a generative model, would be able to generate pictures of airplanes that are not present in the dataset. Image generation goes further than just generating new images, they are also used

in preventing adversarial attacks (Ian Goodfellow and Johnathan Schens, 2014b). Current image generation trends go up to generating 3D objects using capsule networks or 3D convolutional networks. (Jiajun Wu and Chengkai Zhang, 2016). The most popular image generative models are Flow based models, Variational Autoencoders (Kingma and Welling, 2019), and Generative Adversarial Networks (Goodfellow and Bing Xu, 2014a).

Recent signs of progress in deep learning and reinforcement learning have produced enormous programs that can outsmart even the best if humans in games such as Go (Tian and Zhu, 2016), and achieve outstanding accuracy in image classification (Xie and Long, 2020) (Krizhevsky and Hinton, 2012) (Simonyan and Zisserman, 2015), semantic segmentation (Yuan, Chen and Wang., 2019), and object detection (Liu et al., 2019). Image synthesising models or image generative models are an integral component of a number of computer vision and image processing programs.

2.3.1.2 What is GAN?

GAN is a type of image generative model. In 2014, Ian Goodfellow proposed GAN, a novel method of generative model, which gained enormous popularity in both spheres of academia and industry. Many papers on variants of the original GAN have been published. GAN is used for a number of purposes in addition to image synthesis (Brownlee, 2019) . A summarized view of the utilization of the GAN network would be as such. GAN is used for image generation (Karras, Laine and Lehtinen, 2017), image-to-image translation (Isola, Zhu and Efros, 2016), and image super-resolution (Ledig , 2016) to name a few. Despite GAN prioritizing images, there have been many works on text data as well. The progress of the GAN and its relevance in the future is well documented in the existing literature as well as its implications in the future (Brundage et al., 2018).

GAN networks consist of two adversarial neural networks, a discriminator and generator. The two networks play a minimax game inspired by "Game Theory" in mathematics (Goodfellow and Hinton, 2014a). Unlike traditional models which use loss functions such as cross-entropy and difference of squares between the predicted and real labels, GAN uses a minimax loss function that ensures both the networks are trained parallelly. The generator attempt to generate fake images and the discriminator attempts to spot fake images from real ones. The feedback loop from the discriminator helps the generator to generate more realistic images while the improvement of the generator, in turn, induces the discriminator to better distinguish the fake images (images generated by the generator) from real ones.

2.3.1.3 Why GAN is selected in this research?

Latent space vectors are an integral component of the generator. Latent space vectors are generally interpreted as the inputs to the generator. Structurally they are composed of a multi-dimensional hypersphere. Assorted variants of GAN implement the latent space vectors in numerous mathematical modelling methods. "*Science is organized knowledge*", said the famous scientist Herbert Spencer. To put this quote in the case of latent space vectors, they are an inherent structure of organized knowledge or in other words a low-dimensional "representation" of trained data (Bau et al., 2018).

Latent space vectors are prevalent in other image generative models too. However, GAN has a high dimensional latent space, where there's room for much more experiments. Latent space vectors in GAN is also huge in size compared with LSVs of variational autoencoders.

2.3.2 Increase face detection system robustness using latent space vectors

Face detection systems are system that enable the identification of different faces. However, faces are different in nature. But we can't have the same classifier to train on faces. Faces are subject to change. They can be due to a number of factors.

1. Male faces can grow various types of beards and moustaches.
2. Faces can grow freckles and pimples.
3. Normalizing the smiles of faces
4. Wearing of accessories such as hats, tattoos, sunglasses or even olfactory masks.
5. Orthogonal direction the person is facing.

Due to the above reasons we need to generate images of persons sporting different features of their faces on facial detection systems. We also need to make the models as small and concise as possible. Therefore, we will be carrying out the literature review from the following vantage points. These points are postulated after a thorough review

1. We will be taking a review of all current image generative GAN models and doing a thorough review of them to single out a GAN network to carry out our research.
2. We will be taking the GAN and finding a good latent space optimization technique used from existing researches to find out an optimum optimization technique.
3. We will be undertaking a study on model compression techniques to make our model compact.
4. We will use an appropriate clustering technique to cluster our results.

The image below summarizes the action plan. This process of literature review is only meant for the research aspect of the project. The below architecture is based on the product aspects of the project. Since the project is supposed to be developed in tandem with the research, I have made a mini literature review on making a facial detection system as well. This includes literature on face verification, face feature extraction and intermediate bottleneck layers.

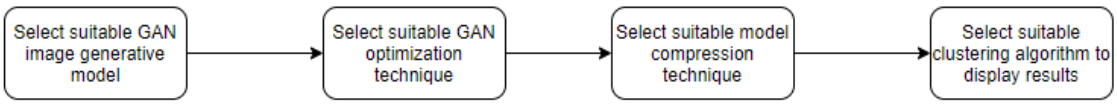


Figure 2.1 : Action plan to be undertaken in the literature review

2.3.3 Architecture of a face detection system

On our pursuit of face detection systems, we have many checklists to be considering, for a successful deployment of the said system. Based on existing works we defined the following architecture for a facial recognition system. We survey a popular facial recognition system called FaceNet (Schroff and Kalenichenko, 2015). We also have outlined algorithms used for the subtasks. We will not be surveying these algorithms extensively due to it being out of scope of this research. We have also derived these features from researches on Face tracking as well.(Zheng and Yu, 2017) and (Di and Patel, 2018).

2.3.3.1 Face Detection and Tracking algorithms

- Haar Feature Selection, features derived from Haar wavelets
- Create integral image
- Adaboost Training
- Cascading Classifiers
- Centroid tracking
- Kernel Correlation filters

2.3.3.2 Geometric Normalization

- Histogram analysis

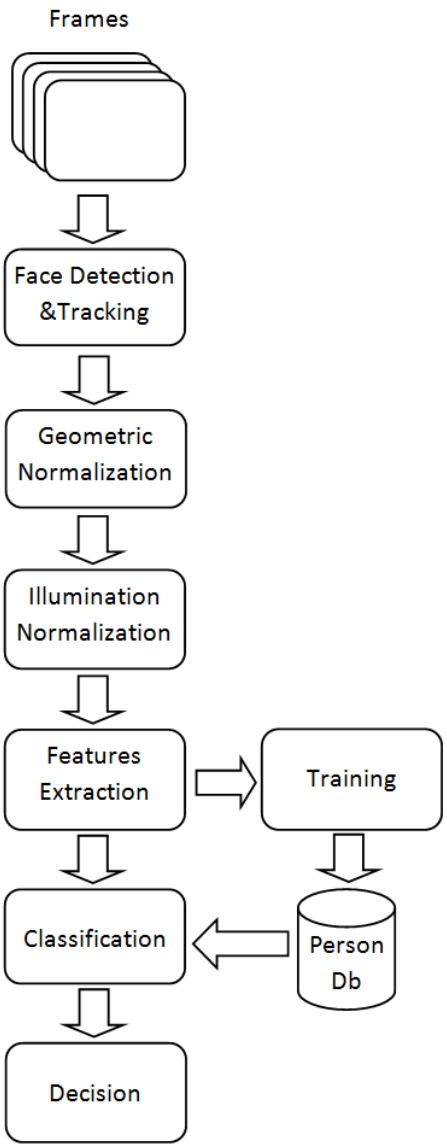


Figure 2.2: Composition of a facial detection system

2.3.3.3 Illumination Normalization

- Illumination Compensation based on Multiple Regression Model (Ko, Byung and Kim, 2002)

2.3.3.4 Feature extraction

The following types of features are possible to be extracted and consolidated for later use from a data distribution.

- Numerical features
- Pairwise features
- Categorical features
- Spatio-Temporal features

2.4 Literature Review of Existing works in Latent Space optimization.

2.4.1 Systematic review of GAN for image generation

There is a vast corpus of GAN variants. Reviewing all of them is out of the focal points of this research. Hence, we will be reviewing a carefully curated list of GAN variants that are optimum for image generation. The following is a comprehensive analysis of latent space vectors of GAN variants popular for image synthesis. We initially dissect each network individually, and then, compile a comparison table on each network. The networks were chosen based on their popularity, current inception scores, and recency. Table 1 lists down instances of GAN variants popular for image synthesis, their latent space vectors and properties.

2.4.1.1 COCO-GAN

In Conditional-Coordinate GAN (COCO-GAN) is a recent proposition in image synthesis (Lin ,Chang, Chen and Wei, 2020). It fragments images into pieces during the training process. The generator generates fragments of images based on spatial coordinates and the discriminator learns to normalise the fragments into complete images. COCO-GAN is unique in nature due to the global coherence, and edge crossing continuity of fragmented images. COCO-GAN has a unique set of latent space vectors. It uses multiple duplicated representations concatenated with micro spatial coordinates of image fragments. This enables this network to perform procedures like “beyond-boundary generation”, “patch-guided generation” and “generation by parts”.

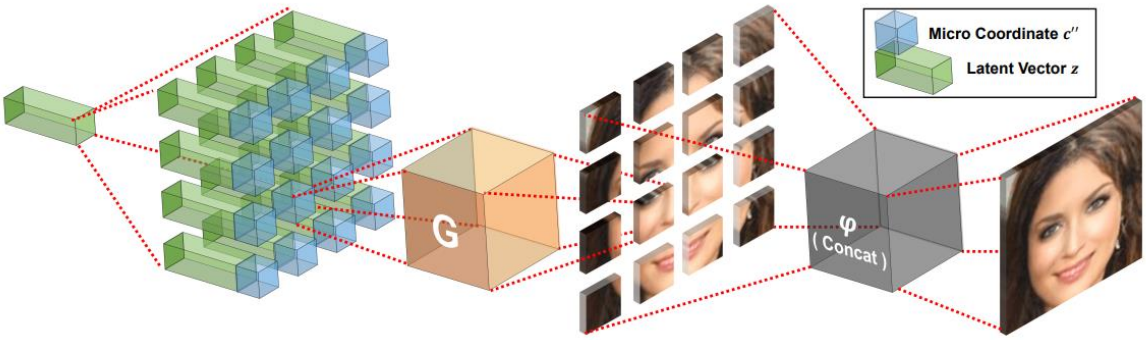


Figure 2.3 : COCOGAN architecture (Lin et al., 2020)

2.4.1.2 StyleGAN

StyleGAN proposes an alternate generator architecture for the GAN (Karras, Laine and Aila, 2018). This implementation of generator leads to autonomous learned, unsupervised segmentation of high-level attributes (depending on the dataset) and random variation in the features of generated images. This means in a dataset of human faces, it is possible to obtain representations of various features of human faces (e.g., eye colour, hairstyle). The latent space vectors are implemented slightly differently by StyleGAN. The latent space vectors are transformed into a representation by a mapping network. The transformed vector is plugged into the convolutional layer of the generator using an adaptive instance normalization layer (AdaIN).

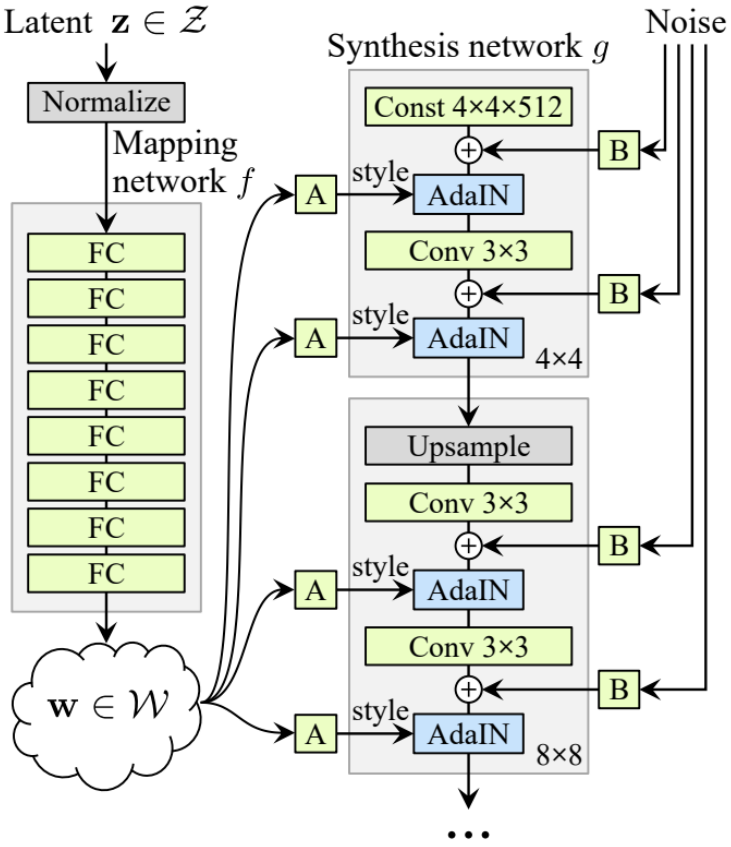


Figure 2.4 : StyleGAN architecture GAN (Karras, et al, 2018)

2.4.1.3 MeRGANs

A frequent drawback of sequential learning in any network is forgetting. As the variations of data points increase neurons loose previously learnt representations, in lieu of generalization. This affects the image generative prowess of a model. This is called the catastrophic forgetting problem. Memory Replay GAN (MeRGAN) is a novel variant of GAN that focusses on the above problem (Wu, Herranz and Wang, 2019). As a conditional generator, the latent space vectors of this model are specialized to capture different categories of features. This is similar to the "Learning without forgetting" approach to minimize feature forgetting (Li and Hoiem, 2017). This consists of two generators, a replay generator, and a regular GAN generator. The latent space vectors are uniform gaussian distributions and are plugged into the generator along with the category conditionals.

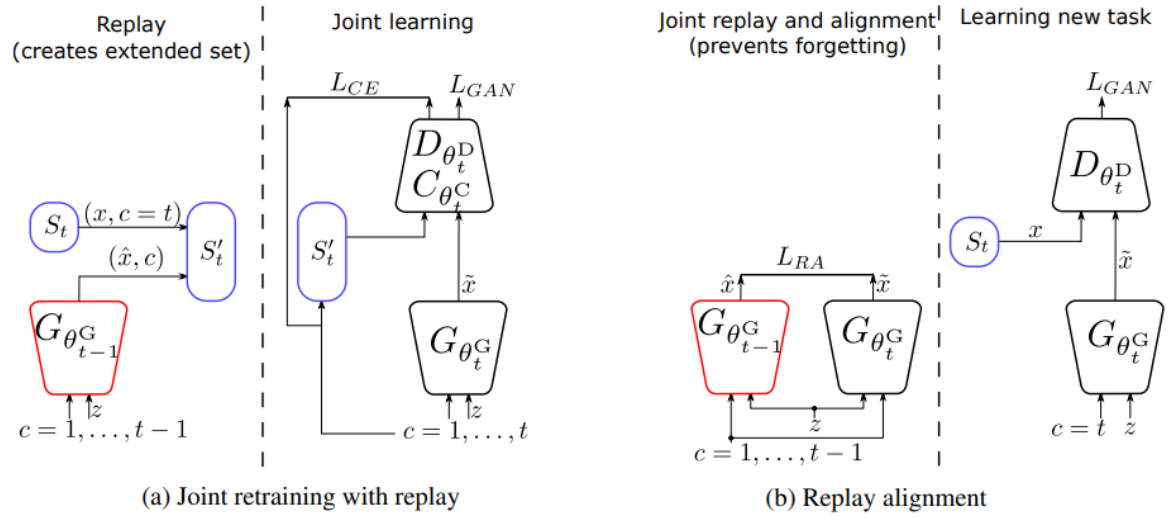


Figure 2.5 : MeRGAN architecture (Wu et al, 2019)

2.4.1.4 BigGAN

BigGAN is a popular conditional image generative model (Brock, Donahue and Simonyan, 2019). It can generate images with phenomenal details and resolution. BigGAN offers numerous improvements over existing literature. It can synthesis images meaningful images with remarkable details. BigGAN also has robust latent space vector interpolation results. We can get intermediate representations of images. Latent space vectors here are connected to multiple layers of the generator network. This permits the generator to manipulate features at different resolutions and levels of hierarchy. Latent space vectors in BigGAN are split into chunks for each resolution and then each chunk is merged to the conditional vector of the image.

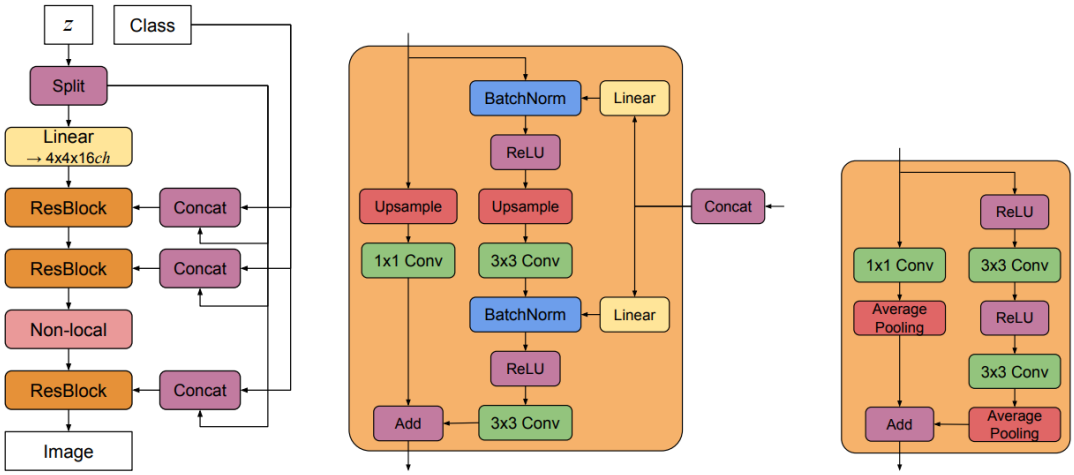


Figure 2.6 : BigGAN architecture (Brock et al, 2019)

2.4.1.5 StyleGAN2

As the name suggests StyleGAN2 was derived from the StyleGAN. The implemented changes to the StyleGAN2 resulted in improvements in model architecture and training methods (Karras, Laine and Aittala, 2019). The generator, in particular, was redesigned. Normalization and regularization were done to felicitate good mapping between latent space vectors and images. Latent space vectors are not different from the StyleGAN (Karras et al., 2018), however, the process of plugging the transformed vector is different. StyleGAN2 uses a normalization layer, which modulates both mean and standard deviation on each convolutional layer.

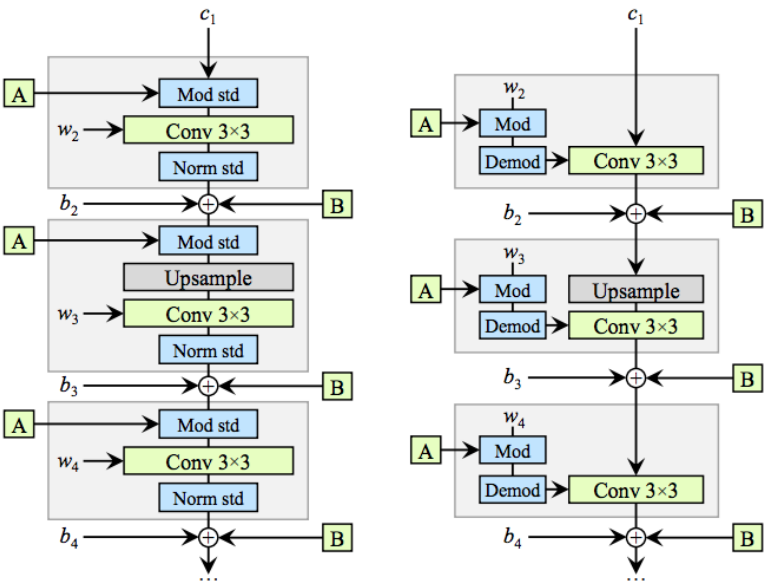


Figure 2.7 : StyleGAN2 architecture (Karras et al,2019)

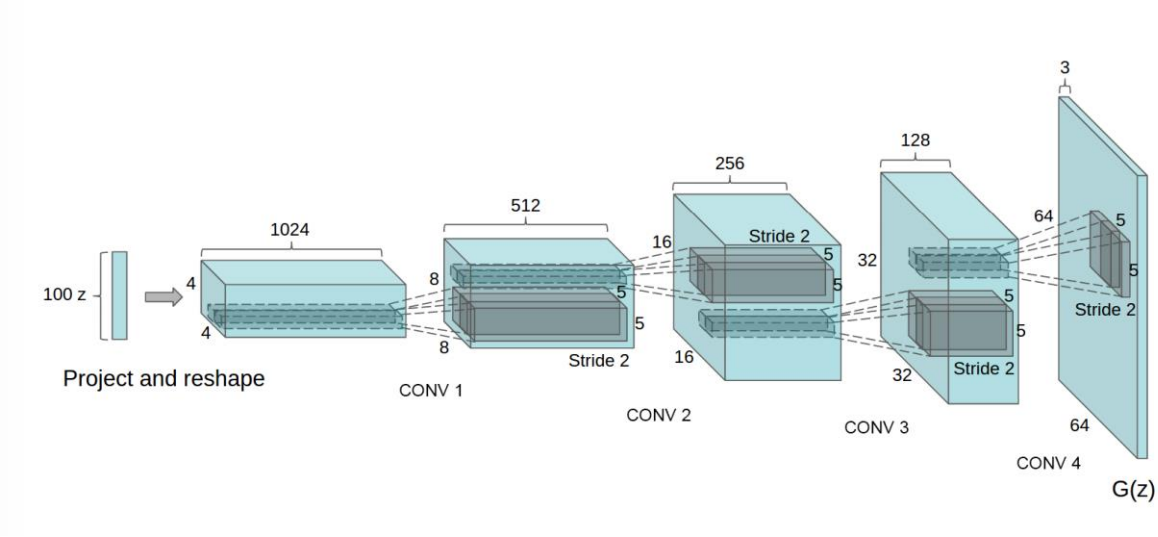


Figure 2.8 : DCGAN architecture (Radford et al,2015)

2.4.1.6 DCGAN

Deep Convolutional Generative Adversarial Network (DCGAN) is one of the first papers on unsupervised that explored latent space vectors academically (Radford and Chintala, 2015). DCGAN consists of latent space vectors that are 100 dimensions in order and uniform in distribution. These vectors reshape into various consequential layers of the generator. The nodes of these layers consist of feature maps and saliency maps of the training data. DCGAN does not contain fully connected layers or pooling layers. They feature sparse layers with reasonable dropouts for better generalization. DCGAN in its debut was a network that became a foundation to many derived GAN researches and a benchmark to other image synthesis models.

2.4.1.7 ProgressiveGAN

Progressive GAN exhibits a dynamic training strategy (Karras et al., 2017). It starts with a low-resolution generator and discriminator and grows both of them in resolution progressively. The model doesn't explore the entire sample of training data, this property of the network speeds up the training speed. Progressive GANs is among the best models for high-resolution image synthesis. Progressive GANs introduces a generative model evaluating metric, for GAN evaluation, both the image quality and the meaningful variations of the latent space vectors.

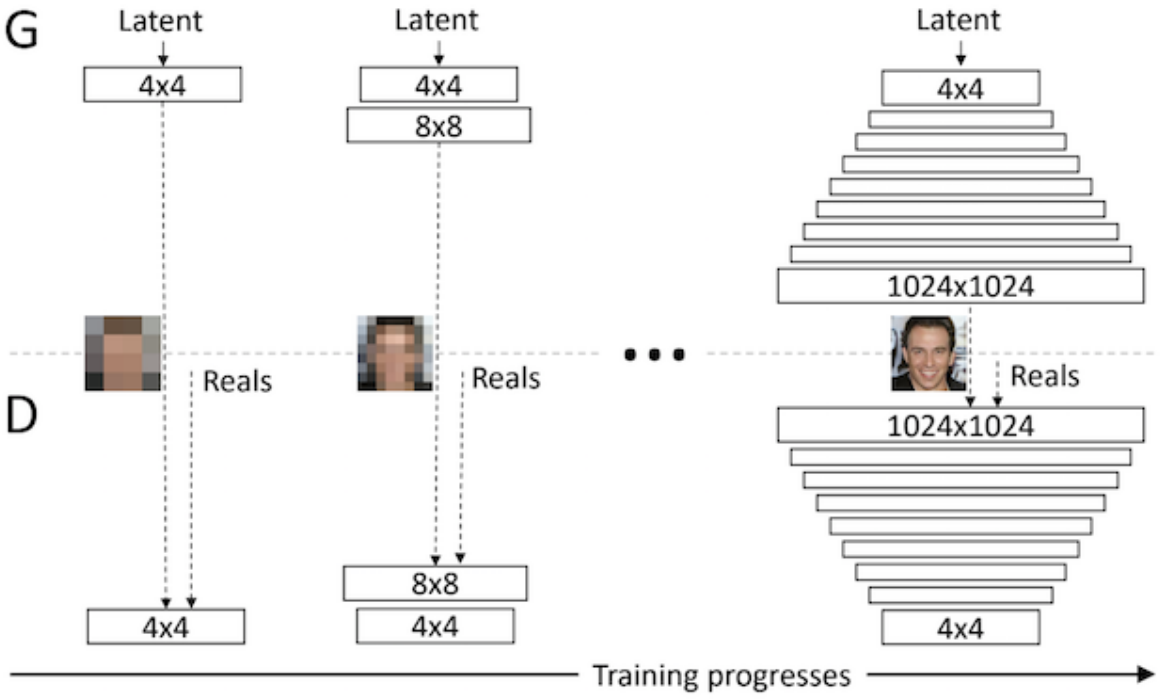


Figure 2.9 : Progressive GAN architecture (Karras et al.,2017)

2.4.2 Summary of Latent Space Vectors in GAN variants

GAN Variant	Latent Space Vector Structure	LSV Distribution	Best Demonstration	Future work
COCO-GAN (2020)	Multiple duplicated and concatenated with micro spatial coordinates of image fragments	Forward-propagates a coordinate manifold that is orthogonal to the latent space vector distribution manifold. (non-gaussian distribution)	• Competes with state-of-the-arts networks in FID scores. • Fragmented image training results in one of the best edge crossing continuity in generated images.	• Extending beyond-boundary generation property to textual and temporal data distributions. • Studies on the correlation between patch size and generation stability.

StyleGAN / SB-GAN (2019)	Transformed by the mapping network to a representation. The resulting representation is then plugged into multiple convolutional layers of the generator network using a blending layer called AdaIN (Adaptive Instance Normalization).	Non-gaussian / able to take any set of distributions	• Standard setter for facial feature recognition. • State-of-the-art facial feature interpolation properties.	• Dynamically shaping the intermediate latent space vectors during training.
MeRGANs (2019)	Consists of two generators. A regular GAN generator and a replay generator. As this is a conditional image generator, both the latent space vector and the category are plugged into the generator along with the category conditionals.	Uniform gaussian distribution	• Novel memory replay architecture. • Ability to display multiple latent vectors simultaneously.	• Applications of MeRGANs in tasks other than image generation.
BigGAN (2019)	Latent space vectors here are connected to	Normal distribution (o,1)	• Class leading image vector	• None specified.

	multiple layers of the generator network. Latent space vectors in BigGAN are split into chunks for each resolution and then merged back to the conditional vector of the image.		interpolation results. • High-resolution detailed image synthesis. • Highly scalable with various datasets.	
BigGAN-deep (2019)	Latent space vectors are not split into chunks and merged into the conditional vector.			
DCGAN (2016)	Consists of latent space vectors that are 100 dimensions in order and uniform in distribution. These vectors reshape into various consequential layers of the generator.	Normal distribution	• One of the first papers to explore latent space vectors. • Showed remarkable results in unsupervised feature representations. • Many future GAN networks were based on DCGAN for its stability and convergence (Liu and Sun,	• Network stability issues persist. • Further investigation into the properties of the latent space vectors. • Applications of DCGAN on video frame prediction and speech synthesis.

			2018) (Donahue and Lipton, 2018).	
StyleGAN 2 (2019)	Transformed by the mapping network to a representation. The resulting representation is then plugged into multiple convolutionsl layers of the generator network using a normalization layer, which modulates both mean and standard deviation on each convolutional layer.	Non-gaussian / able to take any set of distributions	• Improved the quality if images synthesized from StyleGAN. • Significant improvements in dynamic images such as videos. • Trains faster than StyleGAN.	• Study improvements to path length regularization techniques.

Progressive GAN (2018)	A set of Latent space vectors cater to a progressively increasing resolution of the generator. The network increases the resolution progressively, unlike other networks that correlate a mapping from latent space vectors to representations	Normal distribution	• One of the most stable GAN in convergence. • Introduces a new framework of ratifying GAN by considering Multi-scale statistical similarity of batch images encoded as a Laplacian pyramid representation(Burt and Adelson, 1987).	• Curved objects aren't generated well. • Images generated from the LSUN dataset lack photorealism, • There's plenty to be desired in the microstructure of the generated images.

Table 2.1: Summary of Latent Space Vectors in GAN variants

2.4.2.1 Summary of the Latent Space Vector distribution

A brief overview of the latent space vector distributions in existing literature, popular for image synthesis reveals that the most common distributions used as latent space vectors are gaussian distributions. Non-Gaussian distributions exist as well, however, they are limited to the dataset they are trained (Karras et al., 2018) (Karras et al., 2019). Complicated distributions while rare, are found in recent papers (Lin et al., 2020). Complicated implementation of latent space vectors suggests directions for potential future works and experimentation.

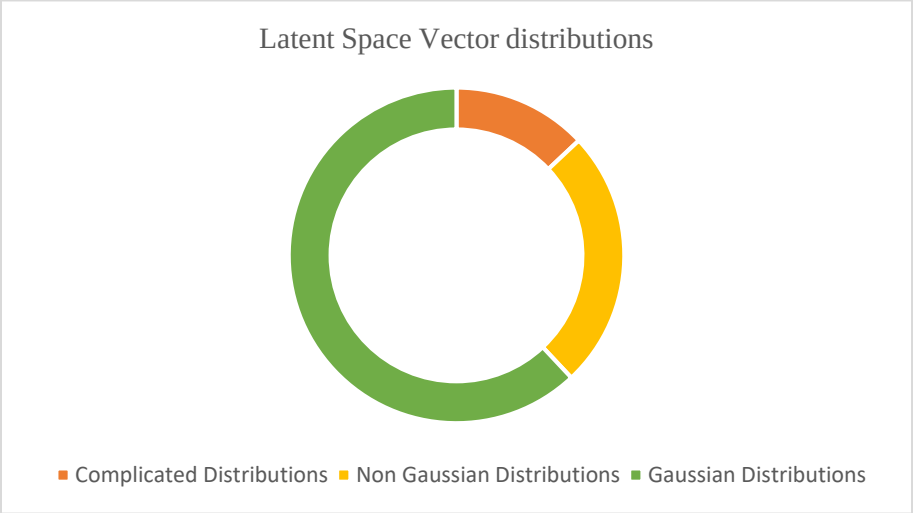


Figure 2.10 : Summary of Latent Space Vectors

2.4.3 Model Compression Techniques Literature Review

A deep learning model is huge in size. This make inference and serialization extremely difficult. Therefore, this section of the literature review we will be reviewing model compression techniques.

2.4.3.1 Runtime Neural Pruning

Facilitates to delete neurons in a layer that don't meaningfully contribute to the training or inference process. Modern works facilitate the pruning process dynamically while runtime (Lin and Rao, 2017).

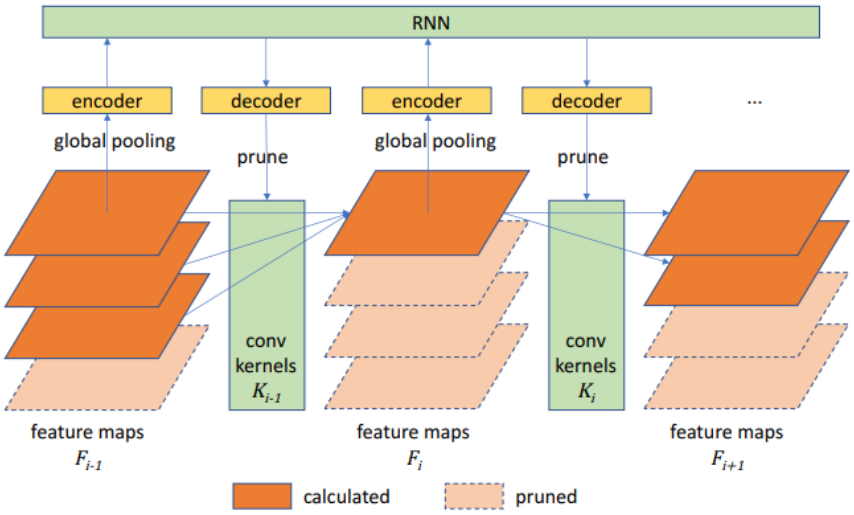
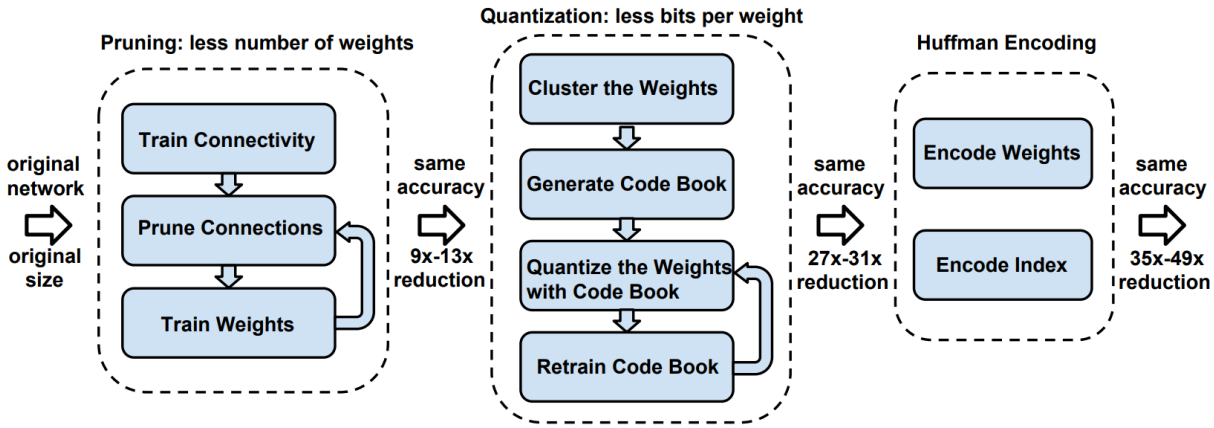


Figure 2.11 : Runtime Neural Pruning architecture (Lin et al., 2017)

2.4.3.2 Deep Neural Compression

Deep Compression (Han and Mao., 2016) was introduced in 2016 shows a novel way to decrease the network size without affecting the accuracy and the precision of the network. The paper introduces a 3-phase process. The method first prunes the network by learning only the important connections. Next, the method quantizes the weights to enforce weight sharing. Finally, the method uses Huffman encoding. After the first two steps, the authors retrain the network to fine-tune the



remaining connections and the quantized centroids. Pruning reduces the number of connections by 9 to 13 times. Quantization then reduces the number of bits that represent each connection from 32 to 5.

Figure 2.12 : Deep compression architecture (Han et al., 2016)

2.4.3.3 Data Neural Quantization Strategy

The authors proposed a dynamic precision data quantization method to improve bandwidth and resource utilization (Qiu, 2016). The method proposed demonstrates remarkably shorter representations of models with still achieving comparable accuracy. Their work is best demonstrated on Field Programmable Gate Array or FPGA.

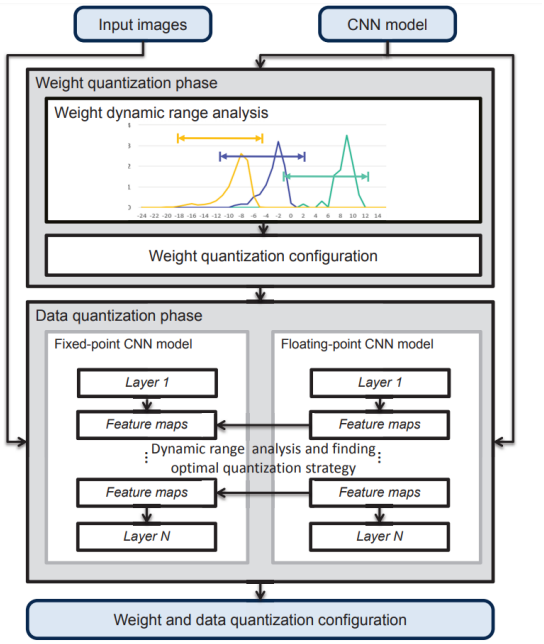


Figure 2.13 : Data Quantization architecture (Qiu et al., 2016)

2.4.3.4 SqueezeNet

Smaller models have faster inference. This work demonstrates AlexNet level accuracy on imagenet dataset with 50 time less parameters (Iandola, Ashraf and Han, 2016). This model compresses AlexNet to less than 0.5 MB in size (more than 500 time the original size).

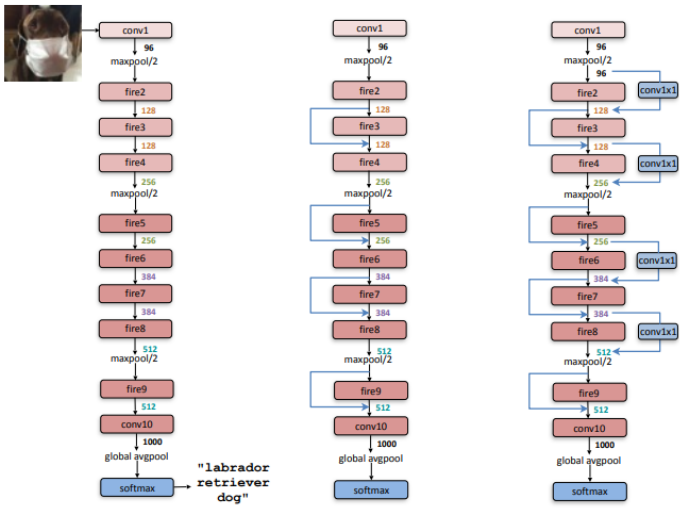


Figure 2.14 : SqueezeNet architecture (Iandola et al., 2016)

2.4.3.5 SqueezeDet

SqueezeDet is a derivative of SqueezeNet (B. Wu, Wan and Iandola., 2019). This work hones its skills towards the real-time inference speed of high accuracy models. This is used in autonomous cars. This is not only faster but also smaller and consumes less energy than previous models. The above models SqueezeNet and SqueezeDet are ideal for deploying in FPGAs and other resource constrained hardware.

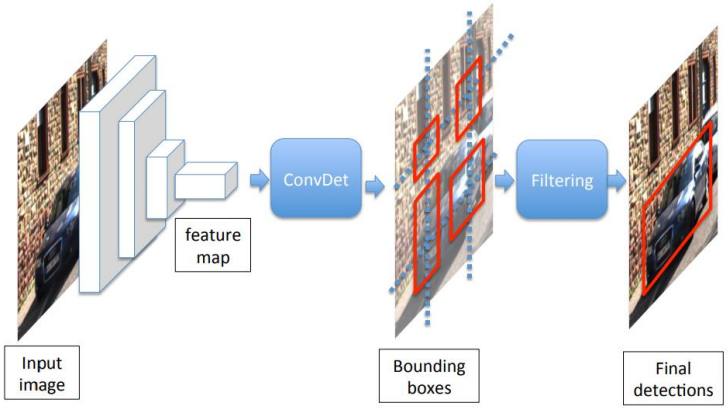


Figure 2.15 : SqueezeDet architecture (B. Wu, et al., 2019)

2.4.3.6 EfficientNet

Efficient Net is a resource tight model. It uses novel algorithms for scaling the image up. They also use neural architecture search to design and develop a unique baseline network to obtain ensemble models that can achieve much better accuracy and efficiency than previous Convnets.

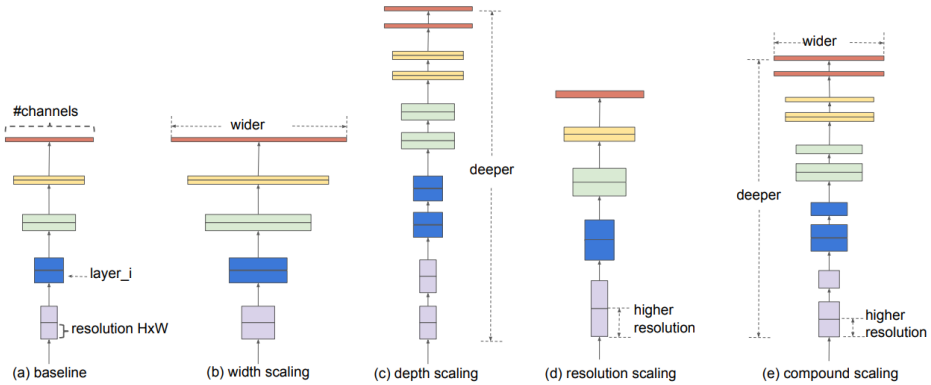


Figure 2.16 : EfficientNet architecture

2.4.3.6 Comparison of model compression techniques

Model Compression Name	Techniques Utilized	Model compression factor (approximation)
Model Neuron Pruning	Uses a simple neuron dropping schema. (Much like dropout layers)	1.5 – 3 times
Deep Compression	Huffman encoding and model weight sharing.	9 – 13 times

Deep Data Quantization	Fischer vector and dynamic precision data quantization method (compiler level).	N/A
SqueezeNet	Uses novel fire module and downsample late so that the CNN layers have large activations at the end of outputs.	~ 500 times
SqueezeDet	Derived from squeezeNet is smaller and faster and consumes less energy.	> 500 times
EfficientNet	Uses neural architecture search to reduce the model size.	8 – 9 times

Table 2.2 : Comparison of model compression techniques

2.4.4 Manifold Learning algorithm analysis

Manifold learning is a process of dimensionality reduction. It utilized when the dimensionality of various datasets is diverged artificially. Mostly data dimension encoding is done by neural networks in respective layers. Autoencoders is another candidate that facilitate the dimensionality reduction. After exploring a survey of manifold learning methods, we were able to single out certain exceptional algorithms for our use case of high dimensional hypersphere clustering (Huo and Smith, 2008).

2.4.4.1 t-SNE algorithm

t-SNE algorithm is a high dimensional data visualization technique for each datapoint (Maaten and Hinton, 2008). It is easier to optimize and visualize data points in a high dimensional tensor. Visualizations produced by t-SNE is much richer and more varied, than existing techniques.

2.4.4.2 Local Tangent Space Alignment

LTSA algorithm is a high dimensional data visualization technique for each datapoint (Zhang and Zha, 2002). It is easier to optimize and visualize data points in a high dimensional tensor. Visualizations produced by LTSA is by making a tangent space for each data point. This is similar to making a vector for each data point.

2.5 Literature Review of mathematical methods of optimization

This is a detailed overview of mathematical methods used to optimize the latent space vectors.

2.5.1 Mathematical methods in hypersphere optimization

GAN variants above, were surveyed in the chronological order. Since the introduction of the GAN in 2014 to the present day, it can be seen that there has been increasing interest shown by the researchers in the latent space vectors of GAN. Authors also state that latent space and its properties are trivially understood (Karras et al., 2018). Latent space vector interpolation and refinement is an interesting avenue in contemporary researches. Works such as COCO-GAN (Lin et al., 2020) have displayed advantages of complex latent space vector implementations.

2.5.1.1 Mathematical Representation Theory

While surveying it was observed that there have been very few researches focused on fundamental mathematical theories. Fundamental researches can explore the mathematical properties of the latent space vectors. A notable characteristic of latent space vectors is their composed structure. They are composed as multidimensional hyperspheres, containing compressed information about the dataset. Representation Theory has interesting applications concerning latent space vectors. Representation Theory suggests representing abstract algebraic structures as linear transformations of vector spaces ("Linear representations of finite groups : Serre, Jean Pierre : Free Download, Borrow, and Streaming : Internet Archive," n.d.). Latent space vectors could be investigated under the lens of representation theory. Furthermore, since representation theory fundamentally classifies all representations of a group up to isomorphism (reversible property) ("Representation theory of finite groups and associative algebras : Curtis, Charles W : Free Download, Borrow, and Streaming : Internet Archive," n.d.), it might give GAN latent space vectors reversible properties that can be reduced to a function.

2.5.1.2 Mathematical Groupoids from Abstract Algebra Theory

Another candidate for latent space exploration using fundamentals would be considering them as a groupoid. Groupoids should be considered as a mathematical group with many objects or identities (Brown, 1987) . Groupoids have many properties conducive to latent space vector exploration. Groupoids have a partial multiplication property which can reduce the algebraic structure into a reduced low-dimensional form known as partial algebraic structures. Topologies and representations in the latent space vectors can be studied better as a groupoid. Many applications of groupoids have been discussed concisely by mathematician Alan Weinstein ("Groupoids: Unifying Internal and External Symmetry," 1996).

2.5.1.3 Ergodic Theory

Ergodic Theory is a class of mathematics that studies the statistical properties of deterministic dynamic systems ("Ergodic Theory - an overview | ScienceDirect Topics," n.d.) . A unique application of ergodic theory to random processes is it various notions of entropy for dynamic tensors. Since Ergodic theory is also utilized to research geodesic flow on Riemannian manifolds. Since latent space vectors are already being treated as Riemannian manifolds this can be a build on module on an existing research (Denker et al., 2006).

2.5.2 Clustering Algorithms survey

A survey on clustering algorithms goes into many modern approaches of clustering (Xu and Tian, 2015). While most of the clustering algorithms implemented are dated, we decided to research some recent algorithms, which are tailor made to high dimensional data. We will look at one algorithm in particular as it seems very relevant for the future in high dimensional clustering.

2.5.2.1 Dynamic Quantum Clustering

A clustering algorithm based on schrodinger equation, the potential is calculated from the data (Weinstein and Horn, 2009). This is theorized to give good performance in clustering. However, the authors extend that process to a temporal schrodinger equation.

2.5.3 Latent Space Optimization algorithms in existing research

After a comprehensive analysis of GAN variants and their latent space vectors, we can now look at how latent space vectors are optimized in existing literature.

2.5.3.1 Generative Latent Optimization Networks

Generative Latent Optimization (GLO) Network is not a derivative of GAN networks (Bojanowski and Joulin, 2017). GLO Network, unlike GAN, does not train two adversarial networks. GLO networks learn to map representations of images in a dataset to latent space vectors by minimizing the image reconstruction loss. Due to the efficiency in optimization, the network is able to generate remarkable samples from the latent space vectors sampled from the dataset.

2.5.3.2 Latent Optimized Generative Adversarial Networks

Latent Optimisation for Generative Adversarial Networks (LOGAN), (Y. Wu and Landola, 2019) makes notable improvements to GAN training for image synthesis by optimizing the latent space vectors. LOGAN by principle improves the adversarial training mechanisms. This is achieved by using knowledge from the discriminator network to guide updates to the latent space vectors. The extensive theoretical study on the model architectures is expected to impact other datasets of adversarial training like text, audio, and video.

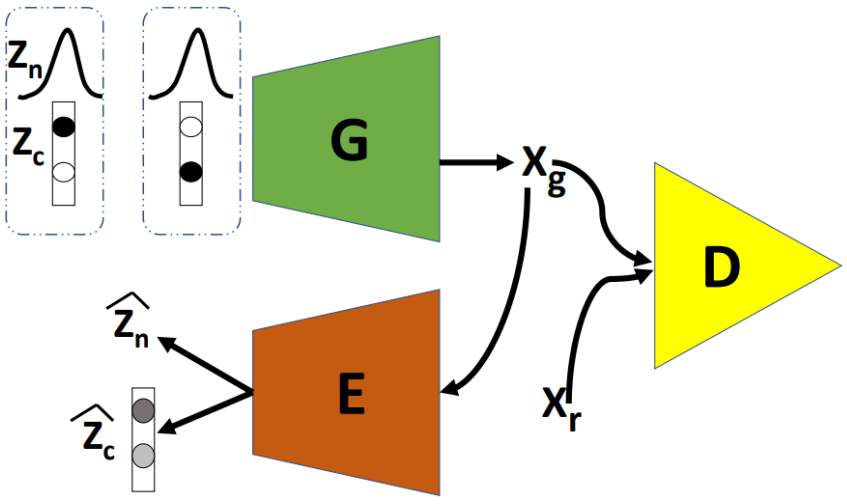


Figure 2.17 : ClusterGAN architecture

2.5.3.3 ClusterGAN

ClusterGAN cluster the latent space during the training process. ClusterGAN, unlike traditional GAN, uses three networks simultaneously. It comprises of a discriminator, generator and an encoder

drawing inspiration from architectures such as Wasserstein Autoencoder (Tolstikhin and Bousquet, 2019) and Adversarial autoencoder (Makhzani and Shlens, 2016). ClusterGAN also consists of a novel backpropagation algorithm. ClusterGAN clusters the latent space vectors using a variety of algorithms to great effect.

2.5.3.4 Summary of Latent Space Optimization algorithms

Optimization Name	**Optimization Algorithm Used**	**Year**	**Future Works**
GLO Networks	Models a latent space distribution from a target dataset of images. The images and the vectors form a bijective function, which is then minimized using a loss function.	2019	• Use sophisticated sampling methods. • Empirically testing with other loss functions, models architectures and progressive generation as in Progressive GAN (Karras et al., 2017).
LOGAN	Uses knowledge (or representations) from the discriminator network to guide updates to the latent space vectors in the generator network.	2019	• Applications on data types other than images, such as text, audio, and video.
ClusterGAN	Agglomerative Clustering (Zhang et al., 2012). K-Means. Spectral Clustering. Non-negative matrix clustering (Lee and Seung, 1999).	2019	• Improve results for compressed sensing. • Probe data-driven priors for latent space vectors.

Table 2.3 : Summary of Latent Space Vector optimization techniques

2.6 Chapter Summary

Based on the above propositions, we presume that latent space vector optimization is a scarcely researched avenue. This paper shows how much work has been done on GAN. It has also shown how much more is yet to be done to understand them better. GAN has a promising future in many tasks involving many types of data. GAN is also a versatile model that can be used both as a classification and generation model. Both applied and fundamental types of research can be carried out as well. Therefore, this literature review of the GAN variants for image generations, can be concluded by stating,

researches on applications of manifold learning methods, groupoids or representation theory on latent space vectors would be useful in understanding GAN latent space vectors better.

The future work on this survey will be to research mathematical methods, particularly focusing on the dynamics of algebraic topologies, groupoids and representation theory. We will also make it our focal point to investigate literature on non-image synthesizing GAN variants to learn deeply about their composition of latent space vectors. Our elemental idea is to utilize manifold learning methods to bring a better understanding of the latent space vectors of GAN.

CHAPTER 3 : METHODOLOGIES

3.1 Chapter Overview

This chapter discourses the methodologies and paradigms of research, project management and product development. The methodologies and management paradigms are selected based on comparisons between existing units. The contributing factors towards decision and the diagrams depicting an action plan to carry out the research is explored.

3.2 Selected Methodologies

The quality of a research is determined by three aspects; feasibility, practicality and scalability, all of which needs to be addressed in an eloquent discipline throughout the research. We also need to scientifically methodize and compartmentalize the research process, such that we follow the scientific methodology in solving a problem. This and the fact that we are carrying out a scientific research unlike many contemporaries in computer science, the exclusion of a concrete scientific framework of assessment is no less than sacrilege.

Scientific Research Methodology	Philosophy	Philosophy is the trigger of any research. A research can be philosophized in 2 main ways. Inductive and Deductive methodologies. They mean an observation-based problem or a survey-based problem respectively. The philosophy of this research is of the **deductive** methodology. Furthermore, this is not an applied research and is a **fundamental research**. The research paradigm is **mixed mode.**
	Preliminary Study	A research fundamentally depends on the relevance of the research **hypothesis** to the domain. As such there has to be an initial foraging of existing theories and systems in use by the researcher. Preliminary study may involve studying existing systems, comparing, contrasting and scoring them based on a common metric. This study is of **qualitative in nature** that takes into account the qualitative results exhibited by the various subsystems of GANs network.
	Problem Domain	Approach to the problem is quantified. Intensive literature review has to be done in order to be confident and move forward in the field chosen. The problem domain dictates if or whether any quantifiable research artifacts can be published in the chosen domain that can be verified by peer reviews, interviews, questionnaires etc.

	Theoretical Framework	Theoretical frameworks of existing researches will be used as the basis in which the research topic and domain will be built upon, the existing theoretical framework for this particular research in general would include the optimization techniques used already on Riemannian manifold. In addition, I will be needing **theoretical foundations on abstract algebra** to further my proposition on manifold learning techniques.
	Hypothesis	Hypothesis is proposed after going through the above three methodology, this is the concrete statement of the problem statement. The hypothesis of any research **guides the direction of the research based on the philosophy** and the type of research question.
	Experimental Design and Ratification	Generative Models are exclusive technologies in artificial intelligence domain. As such we can't use traditional software development. Unit tests and integration tests won't be acceptable metrics. The metrics for generative models **(Chen et al., 2017)** will utilized in experimental design and ratification.

Table 3.1 : Scientific Research methodology

By following the scientific research methodology, the following aspects of the research are determined.

Research Hypothesis: The hypothesis of the research is that by optimizing the latent space vectors of GANs networks using mathematical manifolds from abstract algebra we can get a better control at the features we want from image synthesis.

Research Process: Optimizing the latent space vectors in GANs network for controlled image synthesis.

Prototype Output: The generative model can be used for generating diverse images of a particular class to be classified.

Prototype Features:

1. Person Identification using deep learning
2. Robustness to adversarial attacks
3. Disentangled Training process and data collection process.
4. Graphical User Interface for easy development by developers of Facial Recognition System.

3.3 Work Plan

This following is a workplan based on the literature review and project scope defined and decided by the author and the supervisor. The elemental units in the research such as the artifacts and

the documentation will be elicited in and from the workplan. The following table shows the expected action plan.

3.3.2 Activity Schedule

	TASK	START	END	DAYS
1	Project Initiation	2 Sep 2019	15 Oct 2019	59
2	Literature Review	2 Sep 2019	09 Apr 2020	218
3	Requirement Specification	09 Sep 2019	08 Dec 2019	70
4	Design Specification	09 Sep 2019	08 Jan 2020	90
5	Development Specification	09 Nov 2019	08 Jan 2020	40
6	Prototype Development	09 Dec 2019	19 Apr 2020	89
7	Testing and Evaluation	2 Mar 2020	29 Apr 2020	40
8	Documentation and Dissertation	09 Jan 2020	2 Jun 2020	100

Table 3.2: Activity Schedule

An extended activity schedule is presented in **Appendix C**.

3.3.3 Work Breakdown Structure

Refer **Appendix D** for Work Breakdown structural diagram

3.3.4 Gantt Chart

Refer **Appendix E** for Gantt chart on project timelines

3.3.5 Deliverables

The deliverable that will be resulted from the project were defined as follows,

Deliverable	Date
Project Initiation Document	20th November 2019
Literature Review Review of existing researches and sub domains.	11th December 2019
Software Requirement Specification Requirements for the prototype of the software to be developed.	15th January 2020
System Design Document Based on the literature review, the architectural diagrams, eagle-eye view maps and so on are specified here.	01st February 2020
Prototype Main prototype with core features.	15th February 2020
Thesis Dissertation	01st April 2020
Review Research Paper A research paper reviewing existing GAN variants for image synthesis.	01st March 2020

Public Package / Library A publicly accessible repository containing working code for public facial recognition deployment.	20th April 2020

Table 3.3: Deliverable artifacts and schedules

3.4 Deviations and Risk Mitigation

A project is always associated with risks to be undertaken. It could be technical vulnerabilities. Theoretical risks, or even practical implementation faults. This section will cover risk alleviation strategies to be used if anything expected goes wrong. Of course, there could be things that unexpectedly go wrong as well. The table below displays the risks expected from the research and the product and its mitigation plans.

Risk	Level	Frequency	Risk Alleviation Strategy
Changing requirements of the product. The requirement can change based on many factors. Technical feasibility, theoretical feasibility and resource requirements.	Medium	High	Following a recursive software development methodology such as Agile will help\in alleviating this. This means that as changes occur, we can modularize the changes isolated with that particular system separately.
Deep domain knowledge to be required. This project deals with image generative modelling. This is a fairly complex topic, with a lot of theoretical groundwork. So, such groundwork has to be laid first.	Medium	High	There should be a proportional assignment to the literature review. Reviewing lots of literature on the subject is a surefire was to alleviate the risk in lack of domain knowledge. Also, more time should be assigned to reviewing literature and existing researches. We also took a comprehensive review on subcomponents of the system in the literature review chapter.
Deep generative models' unavailability. This project depends on generative models, both pretrained ones and untrained ones. Lack of the models mentioned in the papers is a serious risk associated with this project.	High	Medium	While there are many model repositories such as ONNX, we can explore them for the models. There are also unofficial models build by the community members on certain papers. Exploring the github repositories can be done in that case. We can also download pre-trained models from github in PKL format.
Machine learning hardware. This is another concern faced by machine learning researchers. Machine learning programs take hours, in some cases days to finish.	High	High	There are many cloud based services that offer discounted rates and free tier memberships to student. Google ML Engine and Amazon SageMaker are good examples of them. And for free tiers Google Colab is a viable option.
Possible Research Veracity. There can be cases where researches are done, that are very	Low	Medium	It is advisable to stay in touch in the field of research. This would mean there is less

closely related to our research gaps and contributions. This can cause problems in conflicting solutions to a research problem			chances of missing out researches similar to the one the authors are undertaking.
Any unprecedented issues. Any other external issues such as political instability, pandemics or other natural disasters can hinder the development of the product.	High	Medium	Practicing good personal hygiene and following the official instructions by the authority during crisis's can help mitigate this risk.

Table 3.4: Risk and Mitigations

3.5 Software Development methodology

We choose the Agile software development lifecycle method due to the iterative development that is necessary in facilitating the research. We also want to develop a product. So, waterfall or spiral won't be useful here because they involve a linear timeline. Rapid Application Development (RAD) won't be relevant here, because this is not a software sprint motivated product.

3.6 PRINCE2 methodology in management

Full form of Prince2 is Project IN Controlled Environments. It's a process centric method to managing a project. It's mostly used for the government projects and private projects in the United Kingdom. The PRINCE2 methodology organizes the project development into logical compartmentalized units. It has a 7-step process.

1. Start the project – PID
2. Initiate the project – Literature review
3. Direct the project – Methodologies and management
4. Control stages – SRS and SDS
5. Manage Product Delivery – Testing
6. Manage Stage Boundary – Evaluations
7. Close the project – Conclusions

This way we can compartmentalize the project components too. We can divide the chapters into stages defined by the PRINCE2 project management methodology.

3.7 Compliance with BCS Code of conduct

BCS code of conduct enforces fours main principles to be followed in the pursuit of enriching the information technological workspace globally. The four principles are listed below.

3.7.1 IT for everyone

This means universal access to information technology. This postulated that any means of sharing information to the public, regarding the health, privacy and security of society should be shared. Since my project a public facial recognition system to enforce security among the society.

3.7.2 Show what you know and learn what you don't

This principle says that we need to be hungry and passionate in the quest for knowledge. We should also show equal footing in teaching the knowledge we know. That was achieved in the project when we reviewed the existing literature in search of a research gap and selected an appropriate research gap. We also summarized and showed the review results in graphical and tabular forms for easy perusal of others in the same domain. This also shows humility and humbleness in research is of paramount importance.

3.7.3 Respect the organization or individual you work for.

This unit mentions that we have to respect the organization or in this case the university that we learn. One of the ways to show respect towards a university we learn from is to follow its values and traditions. We also should be taking responsibility towards our work and not pass the blame on others. We should be ethical in terms of privacy, and basic human rights of others.

3.7.4 Keep IT real. Keep IT professional. Pass IT on.

This means to show professionalism in helping others, passing knowledge and promoting positivity and transparency in information. We should always uphold the sanctity of our work. And should not abuse rights of others. We should also act with integrity and not develop products that would be destructive to the society.

3.8 SLEP Issues

There were ethical guidelines to be followed when designing the system, as the system was processing biometric data. This research project pertains to Class 1: research with no or minimal ethical implications mentioned in the University of Westminster Code of Practice Governing the Ethical Conduct of Research. SLEP principles followed are given below.

Social	Legal
• Evaluators were asked for consent to be included by name in the evaluation chapters. • Personal data are not sent to private servers or through unencrypted channels. • There was no ethnical, religious and political bias in this system.	• The languages and the libraries that were used are under the GPL license. This includes Python, PyTorch and Tensorflow. • Biometric data was collected in a deserialized manner. • Only the information needed for the system was extracted and the rest of the information was safely discarded. The needed information was safely stored.

	• Publication produced during research was copyrighted and transferred to the publishers. • GDPR was followed throughout this project.
Ethical	**Professional**
• The privacy of the evaluators was well respected. The evaluators were made aware of the fact that their feedback is for the evaluation chapter of the thesis. • There's no plagiarism in this thesis, all of the citations were properly referenced. The publications authored has appropriate citations as requested by the conference proceedings. • The biometric data taken in the project for testing is also securely utilized. Everything from processing to data consolidation, ethical guidelines on biometric data usage was followed.	• Standard software engineering practices and industry guidelines were followed for extensibility of the application. • Computers used for the development were password protected and updated with recent security updates. • No pirated software was used to make this research. All the softwares used are with an open source license or with a student license offered by the university. • Results of the research were produced without any manipulation or misleading remarks.

3.9 Chapter Summery

This chapter explored This chapter explored the research paradigms and the scientific research methodologies. We also looked into the project management plans and the temporal feasibility of the research. We choose the Agile software development lifecycle method due to the iterative development that is necessary in facilitating the research. The work plan was thoroughly incorporated with timelines and scheduling. The rest of the project will grossly follow the paradigms and methodologies defined in the above chapter.

CHAPTER 4 : REQUIREMENT GATHERING

4.1 Chapter Overview

This chapter explores the paradigms followed in gathering the requirements for research and the product development. We also analyses the gathered decision to make statistically coherent decisions, that would be crucial in the next system design chapter. We identify the stakeholders relate to the system. Then we will be exploring some of the requirement gathering techniques. After analyzing the data obtained from the requirement gathering, we will be specifying use case descriptions and diagrams. Finally, the requirements with relevance to the functional and non-functional requirements are described.

4.2 Stakeholder Analysis

The stakeholder involved in the product are specified in the onion diagram. The descriptions and the roles undertaken by the stakeholders are specified below as well.

4.2.1 Stakeholder Onion Model

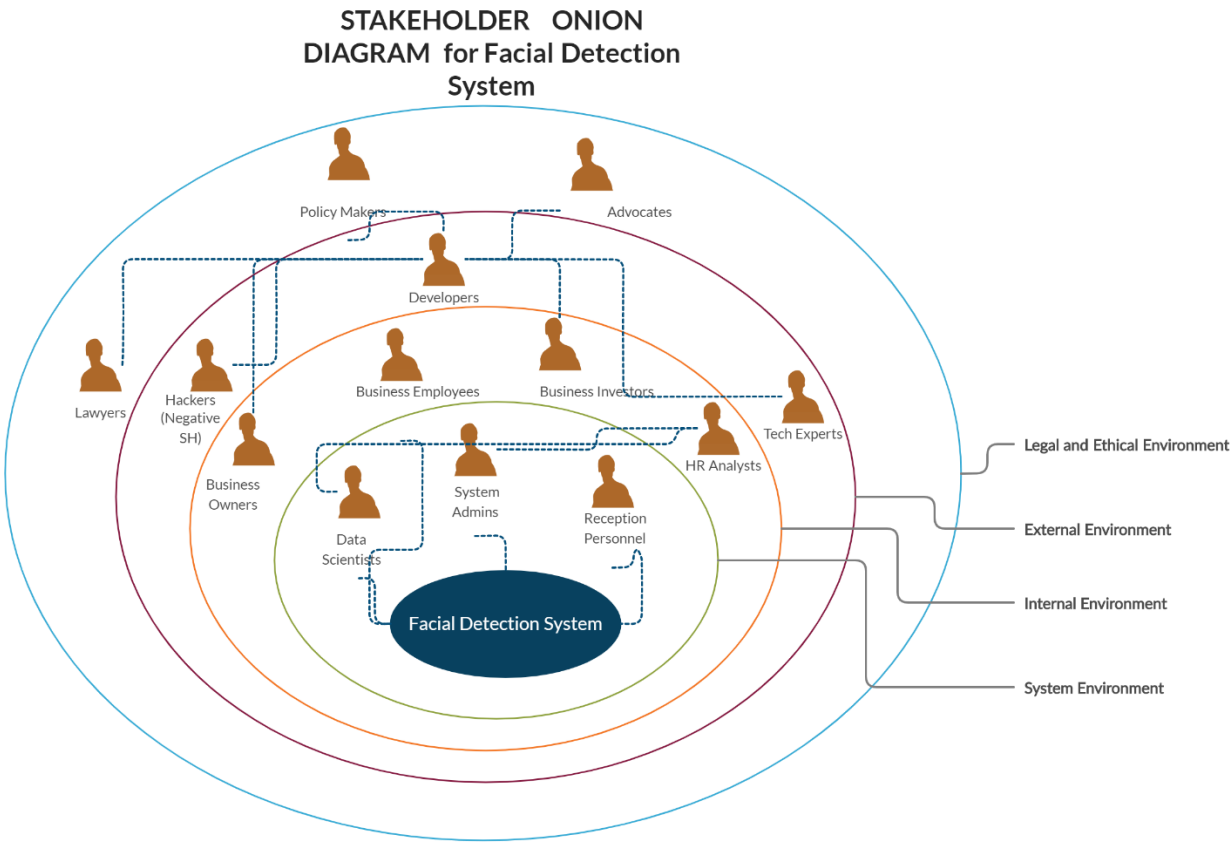


Figure 4.1: Onion model of the system

Stakeholder	Role	Descriptions
System Stakeholders		
Software engineers and data scientists	Product technical maintenance and deployments	Maintains the software. Adds any incremental updates and deploys the software.
System Administrators	Product operational staff	Facilitates the c-level staff and aid them with any technical difficulties.
Reception Personnel	Operational staff	Uses the deployed product in the specified location.
Internal Environment stakeholders		
Business owner	Operational benefactors	Owner of the business. The main stakeholder of the products being developed for.
Business Employees		Employees are the ones that have their biometric data on the system. Their performances are monitored and analyzed.
Business Investors	Financial benefactors	Angel investors, they invest funds in the business to facilitate one or more services.
Human Resources analysts	Data gathering benefactors	Analyze the performances and the work hours of an employee.
External Environment stakeholders		
Product Developer	Engineering staff	Develops the software bases and production pipelines.
Technical Experts	Evaluating staff	Since this deal with biometric recognition software, technical experts evaluate the ethical grounds and clearances.
Hacker	External penetrator	Attempts to cause disruptive behaviors of the system.
Legal and ethical stakeholders		
Lawyers	Legal, ethical and governmental policy makers of biometric data processing systems and methods.	Decides the biometric data processing techniques and ethical guidelines in regulating biometric data storage system. Decides policies to be made with a geographically constrained electorate. Decides the legal guidelines to be followed when deploying such a system to be collecting data from a sample of population for daily recurrent usage.
Policy Makers		
Advocates		

Table 4.1: Roles of each stakeholder in the system

4.3 Analysis of Requirement engineering methodologies

Requirement engineering is a process of foraging requirements for a software project. Requirements can be gathered from the following sources of information. By observing similar systems in the literature review, conducting surveys to relevant individuals, one-to-one interviews and experimentation done on the domain.

4.3.1 Findings from the Literature Review and Existing Systems

We can consider the first step in requirement engineering as the knowledge obtained from the literature review on existing systems. The research gaps in existing researches helps to create more requirement for this project.

Advantages	**Disadvantages**
• Compartmentalized systems can be easily narrowed down • Have a framework for benchmarking. • Limitations and future work can be easily explored. • Apples to apples comparison possible.	• Time and expertise required to go through the papers and documentation one by one.
Findings	
Regarding facial recognition we found that we have many researches that ensures facial recognition features. However, most of the researches are just plain scripts of code. They are not usable in production systems. We found out that we need a good GUI and simplify the system as much as possible for ease of use by no technical professionals.	

Table 4.2: Analysis of literature review

4.3.3 Interviews

Interviews are conducted by contacting domain experts on GAN and image synthesis. Domain experts in the field were reached out and interviewed. Dr.Rukshan Batuwita, Dr. Marek Rei, Dr Hansa Perera, Dr Latefa Elboujdaini, Dr Chrisantha Fernando, and Mr Bathiya Lakmal Priyadharshana are few interviewees selected.

Advantages	**Disadvantages**
• Common fallbacks in image generative models are easily relatable with domain experts on their respective field. • Interviews are more fruitful in discussion and forming ideas.	• Interviews can be personal experiences as opposed to peer reviewed opinions. • Interviews also mean we take opinions from niche group of individuals.
Findings	
The domain experts were interviewed on the research aspects of the project. They commented on the research contribution and the project viability. They commented on the research gap and development strategies. They also commented on the research areas.	

Table 4.3: Analysis of summarized interviews

4.3.4 Observation

Observation and self-analysis were carried out by the author during the course of the research and dissertation.

Advantages	Disadvantages
• Independent research gives the ability to the narrow down the research gap. It also gave the author the ideas to make a prototype.	• The topic of the dissertation and the content of research includes deep generative models. It might be overwhelming for a single person to figure out.
Findings	
Many design decisions of the project were made by observation and brainstorming. The project supervisor, class mates were consulted in the brainstorming sessions. The SRS, SDS and review papers and the final report all were results of deep counselling and brainstorming sessions.	

Table 4.4: Requirements gathered by observation

4.3.5 Distributing Questionnaires

Questionnaires are another way to gather information on requirements. Questionnaires were made based on standard data gathering techniques such as the Elo Rating. Questionnaires were made using Google forms and dispersed online.

Advantages	Disadvantages
• Wide reach of audience. • Less time taken for data input. • Easy analysis on Google forms itself. • Can target a wide range of questions.	• Information put for by the recipients has to be verified if were using them to make design decisions. .

Table 4.5: Analysis of data gathered from questionnaires

4.4 Questionnaire Findings

Question	**How would you describe yourself in the field of machine learning?**
Aim of question	User expertise Identification
Observations	

As expected, the number of participants who was experienced in machine learning were a majority. 25% of participants were had not tried out machine learning and 15% were having experience in machine learning and 60 were beginners interested in machine learning.

Conclusion	
This was useful in determining the useful participants in the questionnaire. Nearly 75 percent of the participants had some experience with machine learning. Therefore statistically we can consider the follow up questions with the same priors.	
Question	**What tools and technologies have you used in machine learning? (Just names of technologies would suffice) Ex. Python, Tensorflow, PyTorch, Apache Spark etc.**
Aim of question	To see the technical expertise
Observations	

PyTorch

Tensorflow, Python, Keras

Python , tensorflow

Python , tensorflow,

Not used anything yet. Planning to use python, tensorflow and also tableau. Yet to start learning

Nothing

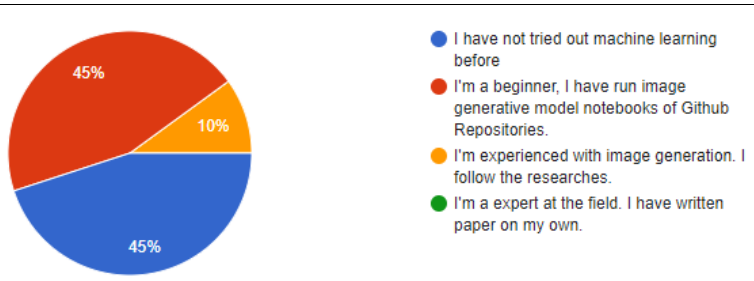
Python, tensorflow

Matlab

Python Vader not scikit learn

Most of the recipients are well versed in the common frameworks of deep learning such as Tensorflow and PyTorch. They are also good with their Python programming. Some have experience in Apache Spark and Sci kit Learn as well.

Conclusion	
Most of the participants are well versed with deep learning tools and technologies. One of the participants was good with even Big Data technologies such as Apache Spark and scientific research software like MATLAB.	
Question	**Have you worked with image generative models?**
Aim of question	Image generative model expertise
Observations	



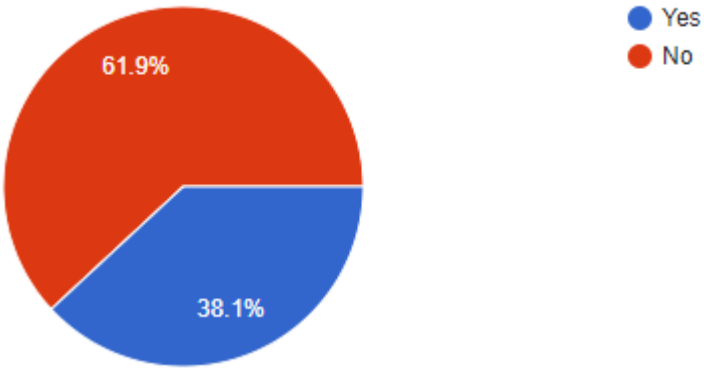
45 percent of the participants have not used imae generative models. 55 percent of the participants have used it to some degree. 45 percent of them understand them nominally and 10 percent of them follow recent researches in the generative models.

Conclusion

More than half of the participants use the image generative models. 10 percent follow recent researches. This is useful in establishing the veracity of the answers that follow this.

Question	**Do you know GAN has a structure called Latent Space Vectors?**
Aim of question	High dimensional tensors

Observations



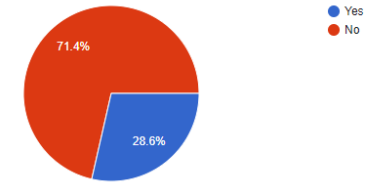
GAN is a niche subject among generative models. Latent space vectors of GAN is more exclusive in terms of exclusivity. Around 40 percent of the participants have said that they know a structure exists in GAN known as latent space vectors

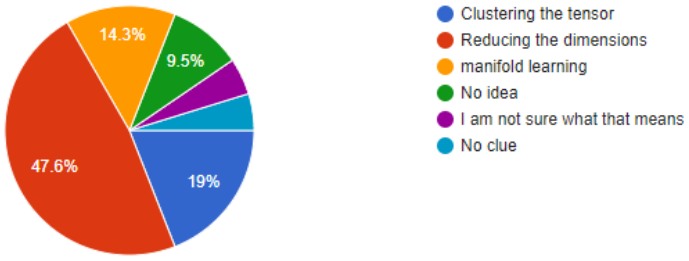
Conclusion

Nearly half of the participants knowing the latent space vectors is welcome news for the research. This will make sure that the scope of the questionnaire was well prepared and the participants were well chosen.

Question	**Are you aware of Latent Space optimization techniques and its effects in model inference?**
Aim of question	Identification of high dimensional tensor optimization techniques and the effects of such optimization on model inference.

Observations

<table>
<tr><td colspan="2"> 

Nearly 30 percent were aware of optimization techniques on high dimensional latent vectors.</td></tr>
<tr><td colspan="2">Conclusion</td></tr>
<tr><td colspan="2">Nearly 30 percent of the recipients were aware of the optimization techniques.</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Question</td><td>If you can think of any ways to optimize a high dimensional tensor what would you prefer?</td></tr>
<tr><td>Aim of question</td><td>Expertise in mathematical optimization</td></tr>
<tr><td>Observations</td><td></td></tr>
<tr><td colspan="2"> 

Nearly 50 percent of the participants where interested in reducing the dimensions. Nearly 20 percent wanted to cluster the tensor. And the 15 percent wanted to consider the tensor as a manifold.</td></tr>
<tr><td colspan="2">Conclusion</td></tr>
<tr><td colspan="2">Evaluation of this question is irrelevant. This serves as a placeholder for future experimentation. We can consider this as a benchmark for mathematical optimization.</td></tr>
</table>

Table 4.6: Results of questionnaires

4.5 Summary of Findings

A short summary of requirements and its sources are listed below.

Finding	Literature Review	Questionnaire	Observation	Interviews
Existence of research gaps in the GAN latent space optimization	X			
Requirement of a model compression algorithm for the GAN.	X			X
Increase of inference speed due to model compression.				X
Architecture of facial recognition system can be formalized.			X	X
Should have an interface with an easy learning curve		X		X
Finalized the tools and technology to be used.	X		X	
Provide GUI on the final application			X	X
Using research models into production is to be taken with care and lots of testing.		X	X	

Should be prone to user defects such as concurrent usage.		X		
Experiment with different techniques to make the accuracy, training time and inference speed as acceptable as possible.	X			

Table 4.7: Summary of requirement gathering

4.6 Context Diagram

The data flow plan has been visualized by the following diagram. This diagram shows the flow of data between the different modules of development.

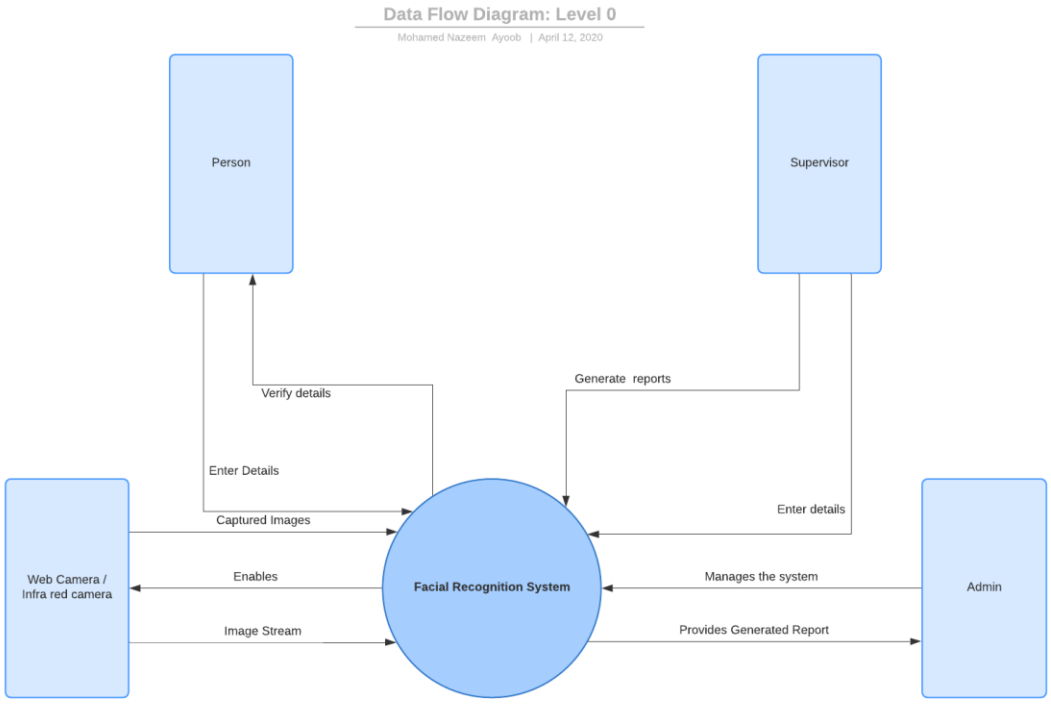


Figure 4.2: Dataflow diagram

4.7 Use Case Diagram

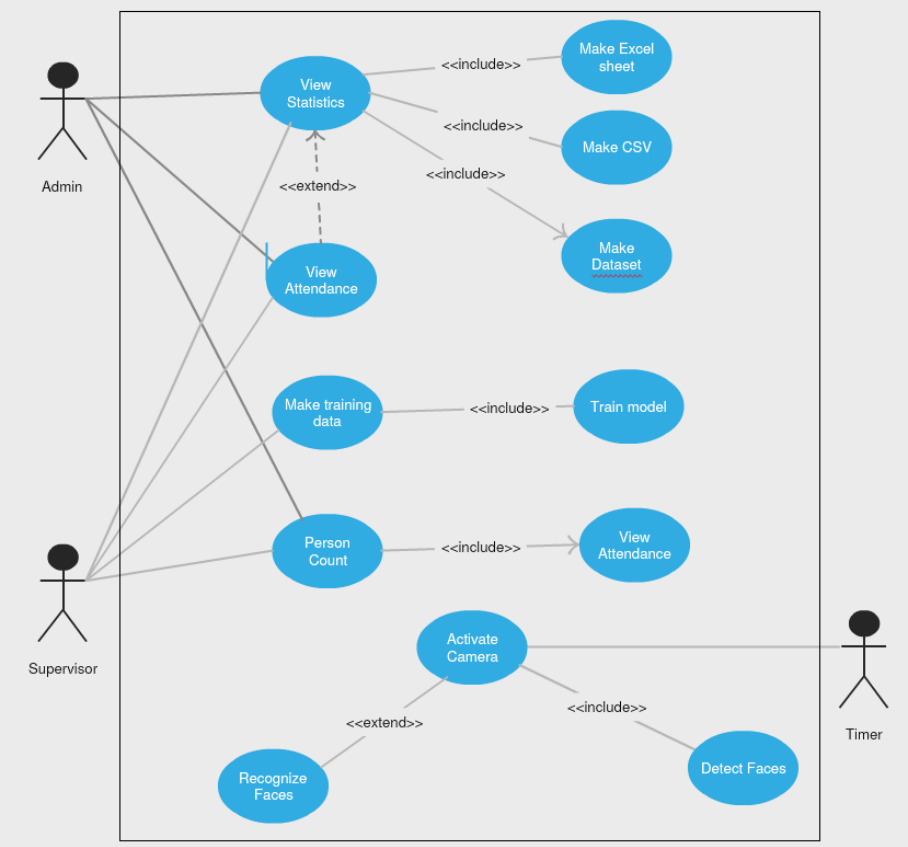


Figure 4.3: Use case diagrams

4.7.1 Use Case Descriptions

Use case of creating statistic viewing plan.

Use Case Name	Get Statistics on attendance
Description	Helps the managers get mores statistics on the attendance or person
Actors	Reception personnel, system admins, Data scientist
Preconditions	Existence of persons attendance data on the system Aggregation of data on the system
Image processing	Input; persons attendance
Main flow	1. System admin or the managers is able to access data 2. Consolidated data being accessed 3. Data is processed visually
Exceptional flows	If there is no data the data is not visualized
Post conditions	Tabular visualization of the data

Use case of viewing all persons and persons counts.

Use Case Name	View person and person count
Description	Get statistics of person count in a particular building
Actors	Data scientists, Human resource analysts
Preconditions	• Counting infrastructure should have been incorporated

Included use cases	Get data preprocessing recommendations; Get learning algorithm candidates
Main flow	1. Data analyst check for the attendance before making reports. 2. Suitable preprocessing techniques are done
Exceptional flows	If there is no data the data is not visualized
Post conditions	The visualization and the report will be utilized in summarizing the attendance statistics.

Use case of activating camera plan.

Use Case Name	Activating camera plan
Description	Get camera feed connected to the ML classifier.
Actors	Data scientists
Preconditions	• Need a valid face classifier.
Included use cases	Train models the classifier with the collected fac images.
Main flow	1. Data scientist trains the classifier.
Exceptional flows	This depends on the camera feed from the camera module.
Post conditions	Enables the camera module to connect to the application

Table 4.8: Use case descriptions

4.8 Requirements Priority Specification

System requirements are classified according to the following three levels of severity.

Priority Level	Description
Critical	Core functionality to the system.
Important	Essential feature to the system, but of less importance.
Desirable	Requirements that are optional to the project.

4.8.1 Functional Requirements

Functional requirements of the system along with its respective priority is listed below.

#	Requirement title and description		Priority Level
FR1	Evaluate classifying algorithms		Critical
	Has to be able to switch from different algorithms for classification based on the meta features.		
FR2	Data preprocessing steps should be outlined properly.		Critical
	Preprocess data before training with different classifiers.		
FR3	Training on the above-mentioned algorithms		Critical
	System should be able to ingress datasets.		
FR4	Activating camera		Important
	System should be able to activate the camera feed to create a dataset.		
FR5	Proper testing should be done behind the scene		Critical
	The model testing accuracies should be benchmarked separately.		
FR6	Be able to view data individually		Critical
	Individual viewing of data should be allowed.		
FR7	Registrars should be able to get statistics		Critical

	The registrars should be able to see the attendance charts.		
FR8	Should be able to persist user data.		Desirable
	The all the data should be consolidated locally or in the cloud.		
FR9	User login information should be stored very securely.		Desirable
	Encrypt passwords		
FR10	Should have an option to stop the training process		Important
	Concurrency rules have to be followed in this regard.		
FR11	Should have a GUI to the app		Important
FR12	Should be accessible publicly to download		Desirable
FR13	Should be updated with the latest python stable wheels.		Critical
FR14	Should have documentation		Important

Table 4.9: Functional Requirements

4.8.2 Non-functional Requirements

Non-functional requirements are in the table below.

#	Requirement title and description	Specification	Priority Level
NFR1	GUI developed should be slick and easy to use	Inception Score	Important
NFR2	Classifying algorithms should be deployable in operating systems with a lot of memory for paging	Performance	Important
NFR3	CPU throttling should be considered when high computer workloads. Try to maintain the clock speed between 60-80 percent of the CPU core clock range	Performance	Important
NFR4	The code should be modularized and properly compartmentalized to ease development learning curve.	Code reuse and modularity	Desirable
NFR5	The system should be plug and play by using libraries such as py2exe and py2app. It would be better if running in on some container-based service like kubernetes or docker.	Code reuse and modularity	Desirable
NFR6	Program should be mindful about the requests that are being sent and received.	Scalability	Important
NFR7	The system should have both user documentation and developer documentation.	Usability	Important

Table 4.10: Nonfunctional Requirements

4.9 Chapter Summary

This chapter discoursed on the narrowing down of stakeholders and their roles in the project. Then we reviewed different requirement foraging techniques. We summarized the requirements based on the various different requirements and the data we received. Using the above data, we are now able to get the functional and non-functional requirements of the project. We also defined the use cases and its descriptions in the project.

CHAPTER 5 : DESIGN

5.1 Chapter Overview

This chapter explores the design decisions undertaken. We also thoroughly go after the system design diagrams such as the high-level, low-level, architecture of the system, class diagrams, sequence diagrams, and the UI diagrams. These design decisions were made based on the requirements that were gathered during the requirement gathering phase of this research. We also go after the rationale behind certain design decisions.

5.2 Design Goals

Design goals should be designated before carrying out any design decisions.

Design Goal	Description
Ease of use	A system has to always be easy to use. It has to be developed with ease of use for the intended users as well as the ease of development for future developers who are adding features.
Performance	Performance is another aspect to be considered when designing a system. It needs to be considered that not only the tertiary performance features such as animations and the transitions but also the core functionalities of the system has to be developed relatively smoothly. Concurrent UI design patterns should be followed to make sure the application runs smoothly without hiccups from both the logical end and the View end. MVC pattern has to be followed in this regard.
Scalability	The system should be scalable when the dataset sizes are increasing. Acceptable Big-O notations would be polynomial time, linear time and logarithmic time. Unacceptable ones are exponential time and factorial times. We should be able to deploy the application as a web server without having any issues with callbacks, request buffer overflow or model inference delays.
Reusability	The software component should be built in a modular way. We should be able to use the modules in a plug and play fashion, with very few dependencies among each other.
Adaptability	This goes hand in hand with the above goal. Adaptability ensures that any new features or changes to the system can be made relatively quickly. Any critical changes should be rapidly addressed.

Table 5.1: Prioritized design goals

5.3 Rich Picture

The system to be designed is fundamentally a face classifier that can be used in a facial recognition framework. The core of the research undertaken is, however, using mathematical methods surveyed from the literature review to cluster the latent space vectors of ClusterGAN

Core research component of the system is an empirical report on the use of mathematical modeling methods on the latent space vectors. The system that is being designed is the usage of the ClusterGAN model as a face classifier to facilitate better and more robust models. The system developed will be tracking the number of people in a particular space and give reports on attendance persons to a registrar. The following rich picture displays the scope of the research and the product developed.

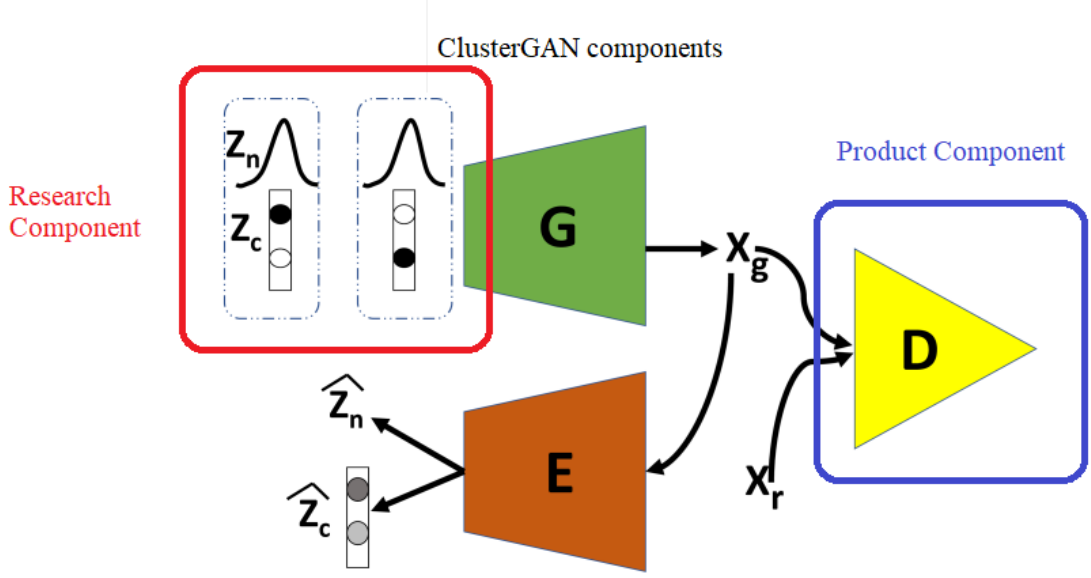


Figure 5.1: Rich picture of the research and product (product emphasized)

5.4 High-Level System Architecture

The architecture of the system is displayed below. It follows the 3 parallelised approach of MVC pattern in app architecture. The research contribution lies in the model layer while the view and controller are application controllers. The architecture is highly modular and many software engineering principles are followed. Scalability, performance-oriented, modular and providing a good user interface to the user in a simple easy to use interface.

The endgame of the system is that the system will be a python application. The following diagram will reflect and have one-to-one mappings with the python applications.

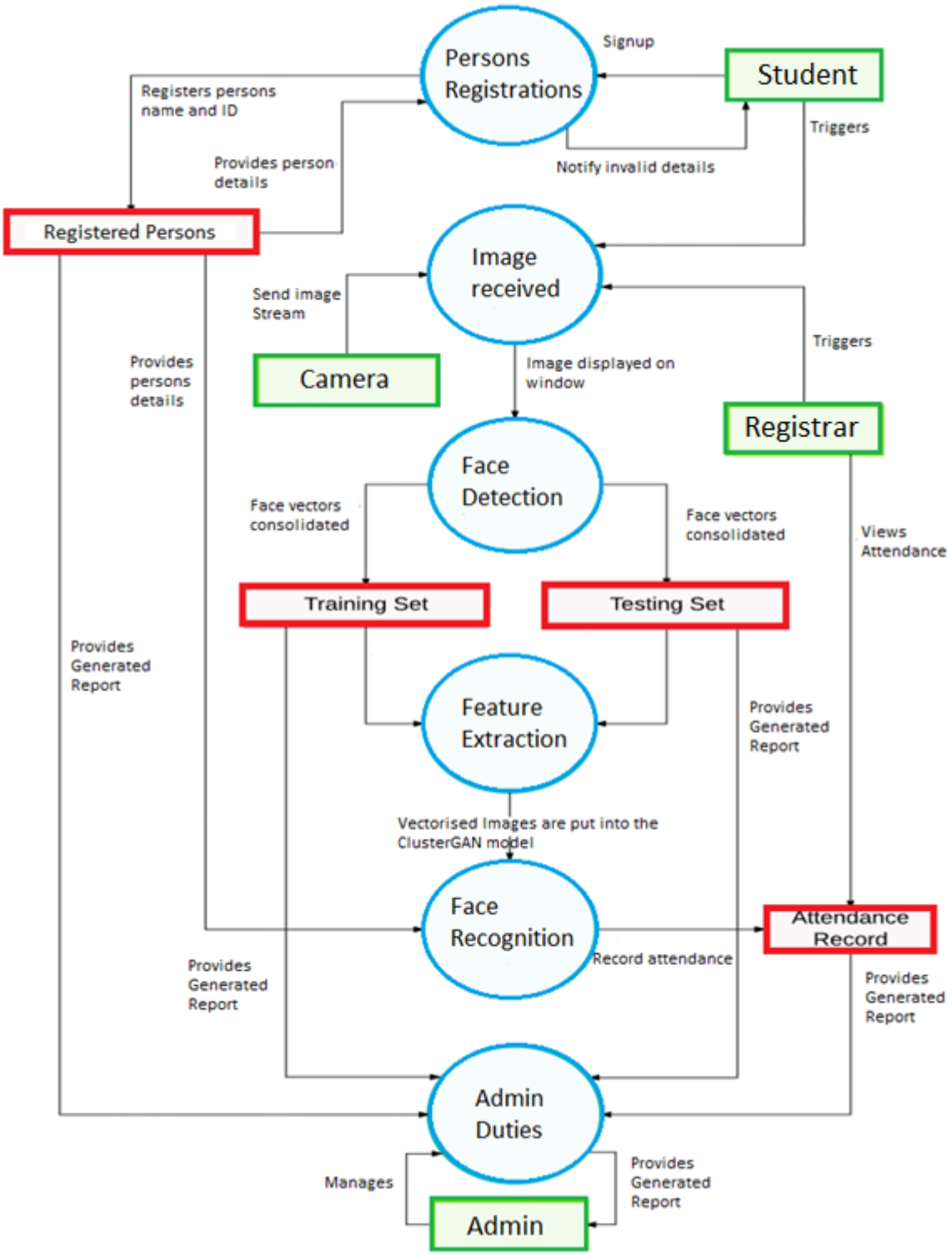


Figure 5.2: high-level architecture

This diagram describes the architecture of the system. Initially, we have person registration. Here the person could be an employee, student or teacher. This is a person recognition system that can count the number of people in a building. After person registrations, we can switch the camera on to identify features of the person. This will make sure the face is detected and cropped and transformed appropriately to feature vectors. This will be feed into a neural network model that can be used to

classify the dataset we have collected. The trained model can give predictions. This also facilitates in an admin and a registrar, and certain roles to them such as seeing attendance records.

5.5 Dataflow Diagram

Components were identified from the architecture diagram. They were compartmentalized into components and modules. The dataflow diagram shows the relationships between the components and the data flow directions.

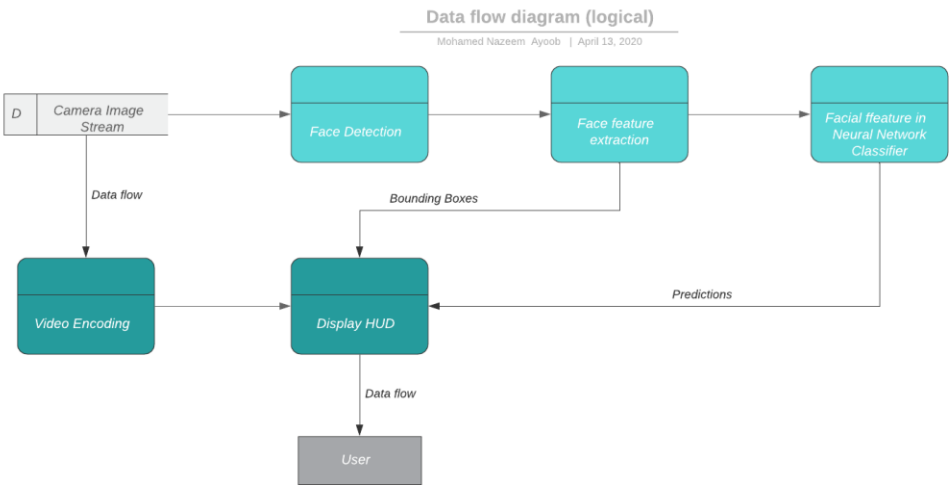


Figure 5.3: Dataflow diagram

5.6 Class diagram

Image generation is not a deterministic paradigm in software engineering . The whole paradigm of machine learning and generative models is non-deterministic. As such one-to-one mapping it to object-oriented principles is a challenge. Additionally, the fact that we are designing a biometric facial recognition system, means that the object-oriented principles have to be followed. Though the application is going to be build using functional programming principles, the diagram below roughly shows an object-oriented perspective of the system we are going to develop. Here we go through the individual features the classifier hopes to classify.

Additionally, the machine learning face detection strategies are documented in the diagram below as well.

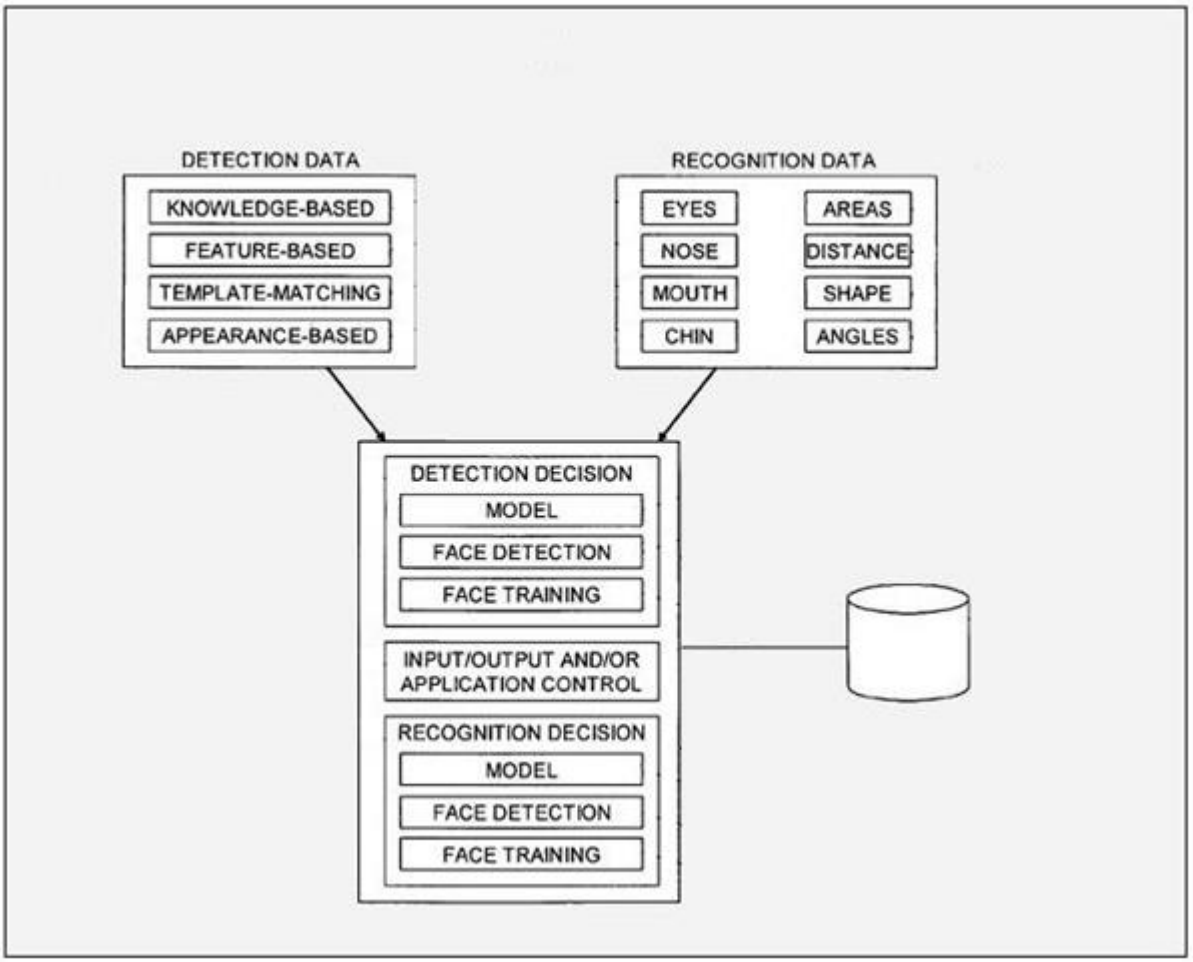


Figure 5.4: Class diagram of the system

Now that we have an understanding of how classes interact with each other we can start looking at sequence diagrams for the application architecture.

5.7 Sequence Diagrams

5.7.1 Sequence Diagram of Facial Recognition system

Below the entities in this system are divided into five distinctive categories. The various interactions between the entities are displayed in the below diagram. This sequence diagram explains the creation of a dataset, the training process and the prediction phase of the application.

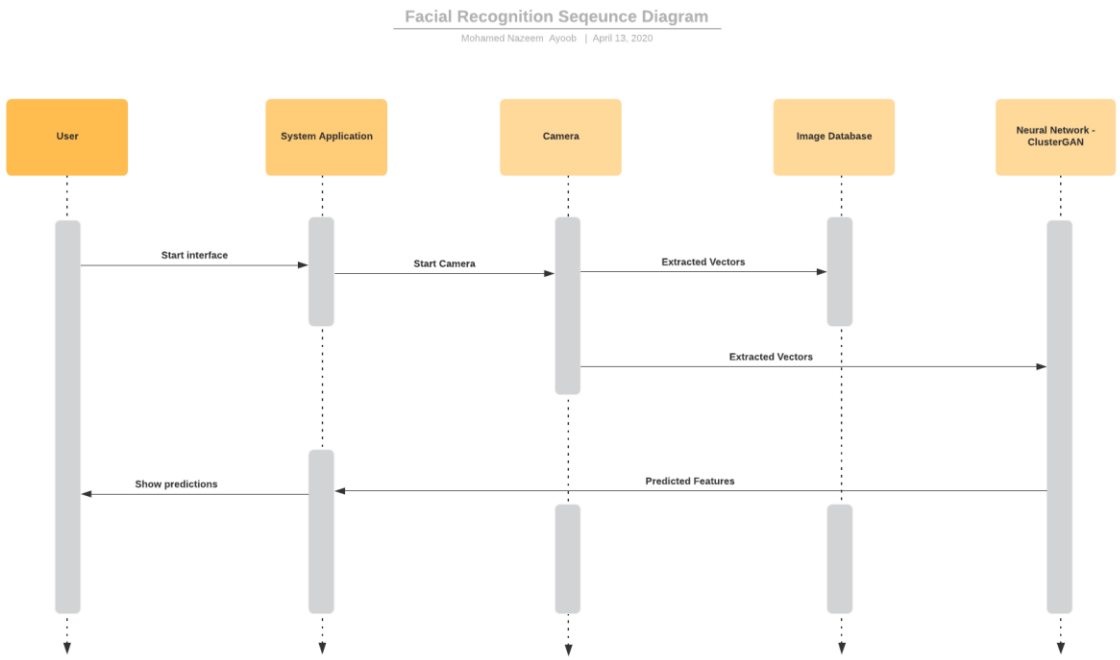


Figure 5.5: Sequence Diagram

5.8 System Process Flowchart

The flowchart below displays the flow of the application and its algorithms and certain design decisions. The system is written using functional programming principles, the chart below explains the most crucial parts of the system.

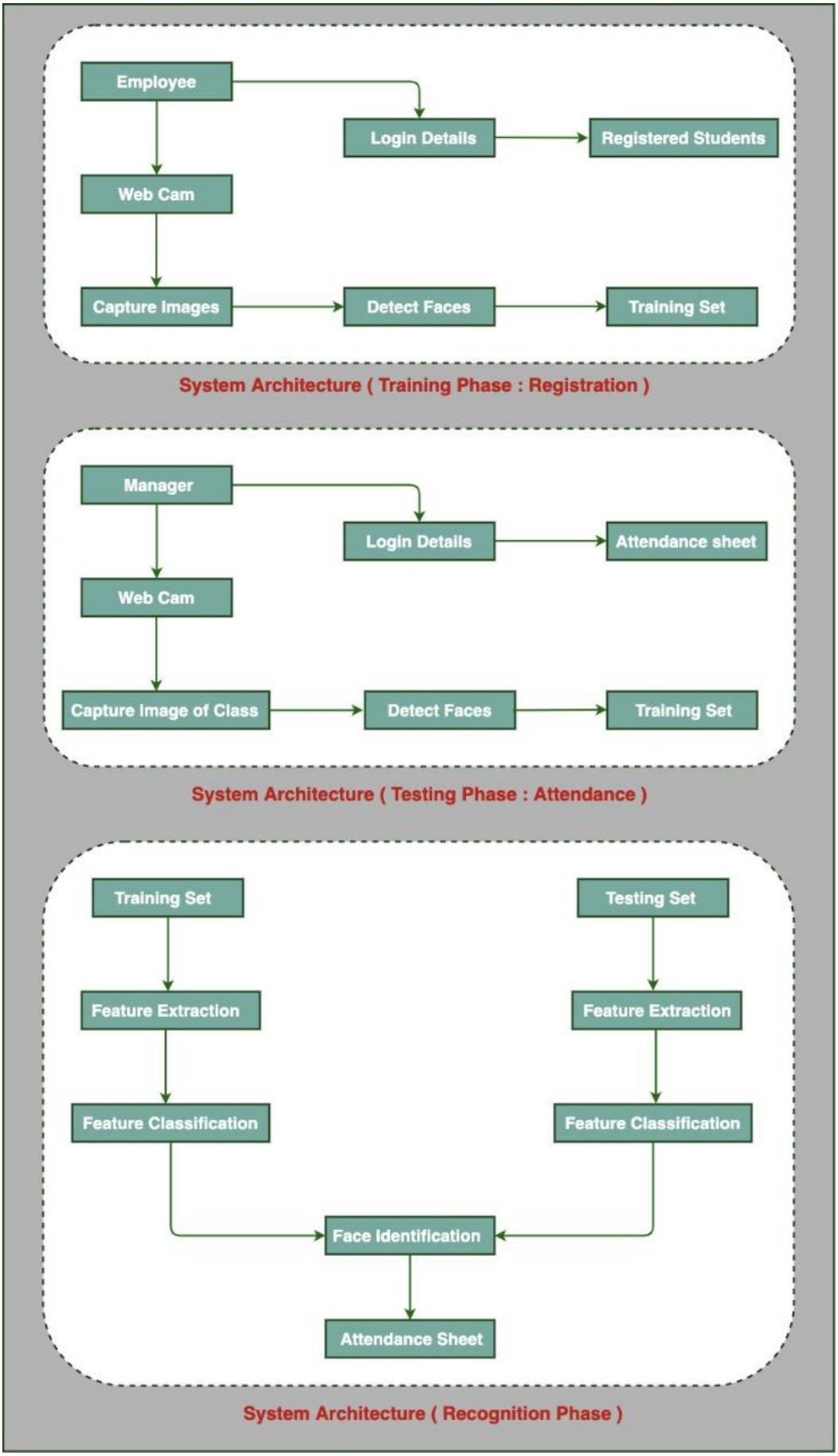


Figure 5.6: Process flow of the system

5.9 High Fidelity Mockups

The system will contain a GUI for easy usage by C-Level staff. As such the user interfaces were made using extremely simple features that exhibit the core functionalities of the application. As the application is a desktop application, the wireframe was developed to model desktop apps.

The start of the application is shown below.

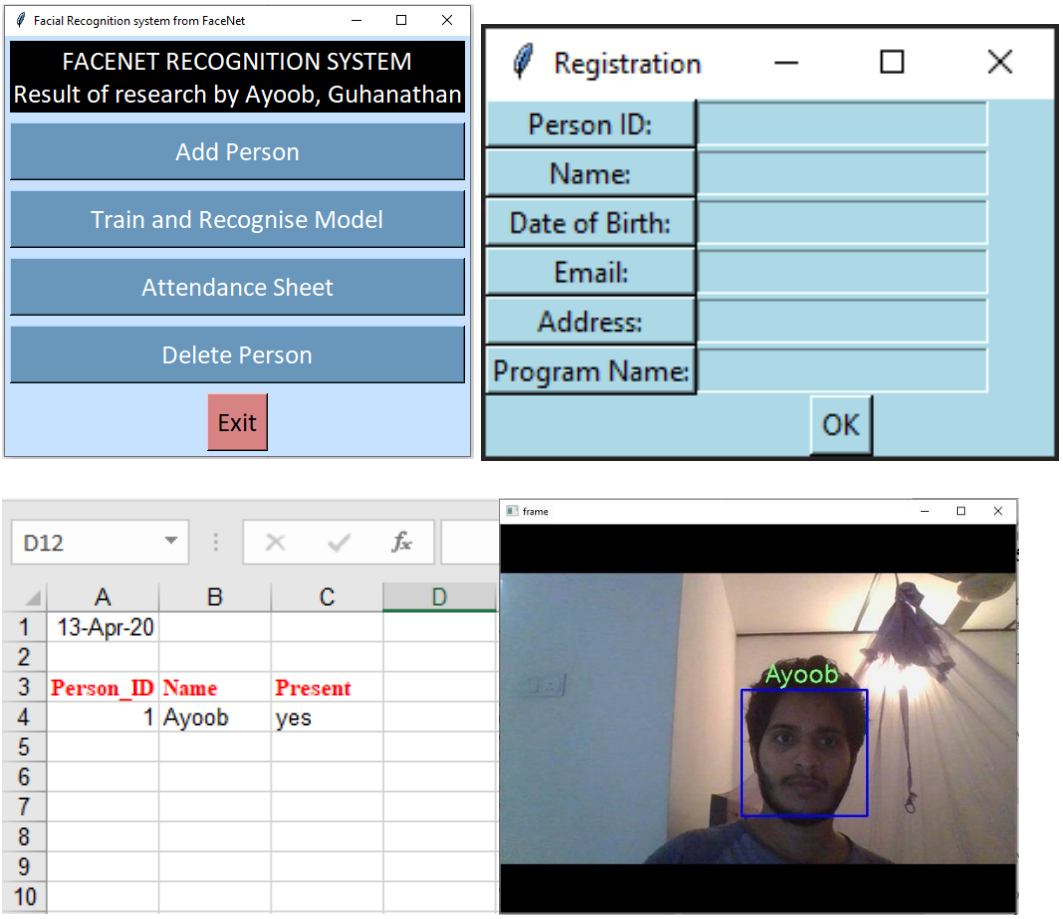


Figure 5.7: Wireframe of input wizard

The above are the screenshots of the product developed.

5.10 Chapter Summary

This chapter discoursed the design and architectural units of the application about to be developed. First, the gathered requirements were analyzed and the design goals were instantiated. Then the rich picture and the high-level architecture diagrams were drafted. Then we narrowed down on the sub-components of the system. The component diagram, sequence diagram, and the class diagrams were defined. Then finally the processes were shown as flow charts and UI wireframes were used.

CHAPTER 6 : IMPLEMENTATION

6.1 Chapter Overview

This chapter will be exploring the various aspects of implementing the project and the research. This chapter is the culmination of the last 3 chapters, which were the literature review, requirements gathering and the project design. Here we will take a look at how the design decisions that we developed will be transformed to runnable code. However, we can expect there will be implementation decisions where we will be deviating from the design decisions. Therefore, such decisions will be curated in this chapter as well. Any implementation blockers and code snippets will be accompanied in this chapter.

6.2 Selection of Languages and Tools

6.2.1 Dataset Selections

Deep generative models, or commonly called image synthesis models, are systems of generative data. Before looking into pretrained models we would like to take a look at the image dataset requirement fulfillment. There are many image datasets that can be utilized for the purpose of deep learning. However, the following few are selected fue to the wide availability and pre-existing data pre-processing techniques. The dataset requirements for our project would ideally be as listed below.

1. Datasets should be modelling the objects in the dataset.
2. Dataset should be Euclidean in nature.
3. Dataset should already be used in many deep learning tasks and importantly for image generation.

After a cursory exploration on image dataset the following datasets were chosen to be included in the project.

1. MS COCO dataset
2. Imagenet
3. Open Image Dataset V5
4. Yelp Review
5. Microsoft Open Dataset
6. University of California Dataset.

These datasets were foraged for the task. These contain Open datasets from Universities and Industries as well. All of this dataset is publicly available at the time of writing.

.

6.2.2 Programming Language

The languages used to research and build machine learning tasks are given in the following table. After through research Python was used to carry out the project.

Language	**Deep Learning Libraries**	**Advantages**	**Disadvantages**
Java	Weka, Deeplearning4j	• Easy to use • GUI • Conform to Java's write once run anywhere philosophy	• Not Scalable • Not used a lot by researchers. • Java has a steep learning curve • Not easy to find pre built models
C++	Dlib, Tensorflow, torch	• Close to the hardware • Doesn't need a lot of dependencies. • C++ is very fast compared to Java and Python.	• Not easy to find built in models • Very recent focus on deep learning by C++. • Community has less support.
Python	Tensorflow, PyTorch, SkLearn, Keras	• Vast community support. • Support for distributed machine learning. • Python has an easy language learning curve. • Easily Scalable. • Many prebuilt models are shared in Python.	• Python is slower than C++.

Table 6.1: Programming language comparison

As such the language of development was choose to be **Python.**

6.2.3 Deep learning library

Python got many deep learning libraries. The most prominent are Tensorflow, Keras and PyTorch. The table below lists. PyTorch was chosen as the library, due to the widespread use and popularity of it. Pytorch is also very Pythonic in nature, which make extending the program much easier when compared to other off-the-shelf alternatives.

Name	Description	Special Features
Keras	Open-Source neural network library. It is easier than TensorFlow, and very beginner friendly.	Easy implementation. Sequential Class very useful in debugging.
TensorFlow	Developed by Google. It is one of the most stable libraries for deep learning. It has a system of operations graph that is statically assigned during the compile time.	Very stable. TensorflowX or Tensorflow Extended is miles ahead in industry applications.
PyTorch	Developed by Facebook. It's built using C++ and CuDNN from Nvidia. It has a dynamic graph of operations. This makes the kernel stable. Kernel panics from PyTorch are far less than Tensorflow or Keras. Dynamic length graphs are specially useful when using character encodings with Recurrent Neural Networks.	Easy to learn and implement. Very pythonic in code nature. Has JIT compiler which make model tracing easy.

Table 6.2 : Comparison of machine learning libraries

6.2.4 IDE

Google Colab was choosen to be the IDE in the research component development. PyCharm was choosen as the IDE for the product development. Google Colab is free for use publicly. PyCharm Professional is free for student to use.

6.2.5 Summary of the implementation tools

Component	Tool
Programming Language	Python
Deep Learning Library – Research	PyTorch
Deep Learning Library – Product	Tensorflow
UI library	TKInter
IDE – Research	Google Colab
IDE – Product	PyCharm Professional
Version Control System	Git
Image Processing Library	OpenCV

Table 6.3: Summary of the implementation technologies

6.2.6 Technology Stack

The technology stack decided to be used in the different layers of the system are as follows.

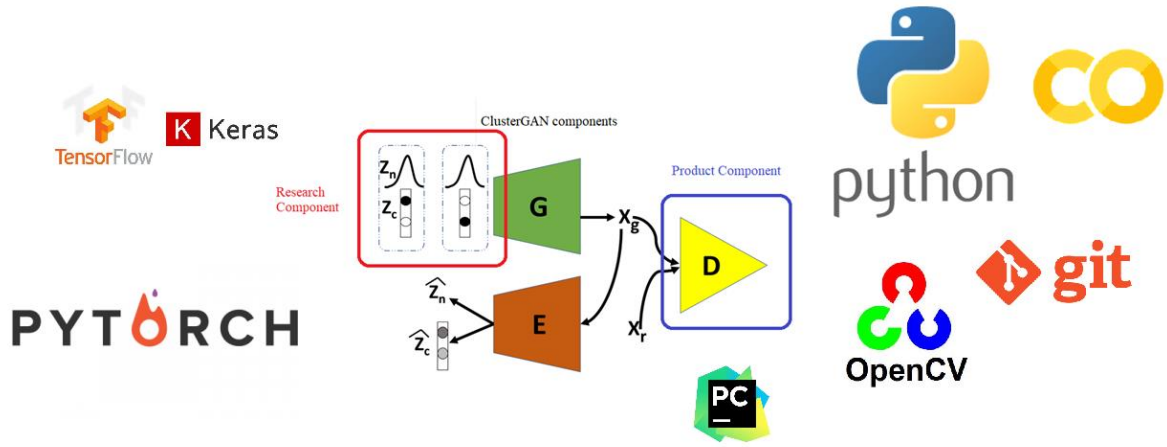


Figure 6.1: Technology Stack utilized by the research component and the product component

6.3 Core Functionality of Product

This will display the code snippets from the product developed. That is the facial recognition system.

6.3.1 Dataset Collection

```
# Loop for faces until num of faces saved.
while (True):

    # Analyse Web cam video feed
    _, single_image = web_cam.read()

    # Remove color channels
    remove_color_channel = cv2.cvtColor(single_image, cv2.COLOR_BGR2GRAY)

    # Detect number of faces in the image
    list_of_faces = init_face_crop.detectMultiScale(remove_color_channel, 1.3, 5)

    # for each face in list_of_faces
    for (x, y, w, h) in list_of_faces:
        # Crop and vectorise the image
        cv2.rectangle(single_image, (x, y), (x + w, y + h), (255, 0, 0), 2)
        # label text
        cv2.putText(single_image, decrypt(enc), (x, y - 10), font, 0.5, (120, 255, 120), 2, 1)

        # num of faces
        num_of_faces += 1

        if num_of_faces == 1:
            # create training data.
            cv2.imwrite("images/" + name_id +"_" + str(face_id) +".jpg", remove_color_channel[y:y + h, x:x + w])

        # Display face with bouding boxes
        cv2.imshow('frame', single_image)

    # Stop video frame press q
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

    elif num_of_faces >= 30:
        print("Successfully Captured")
        break

# Webcam feed ended
web_cam.release()

# Close windows
cv2.destroyAllWindows()
```

This code snippet is for the dataset collection in the facial recognition system. We use the standard way to collect image data.

6.3.2 Image Training

```
'''
The loss function from the thesis implement here.
FaceRecoModel was ported and optimised to run on average computing resources.
'''
def own_loss_function(y_true, y_pred, alpha = 0.3):

    differential, pos, neg = y_pred[0], y_pred[1], y_pred[2]

    # Step 1: Reduce distance between the differential and positive
    positive_distance = tf.reduce_sum(tf.square(tf.subtract(differential, pos)), axis=-1) # axis -1 IMPORTANT
    # Step 2: Reduce distance between the differential and negative
    negative_distance = tf.reduce_sum(tf.square(tf.subtract(differential, neg)), axis=-1) # axis -1 IMPORTANT
    # Step 3: get the difference and add alpha
    intermediate_pooling_loss = tf.add(tf.subtract(positive_distance, negative_distance), alpha)
    # Step 4: Argmax of basic loss and 0
    final_loss = tf.reduce_sum(tf.maximum(intermediate_pooling_loss, 0.0))

    return final_loss

facial_recognition_model.compile(optimizer ='adam', loss = own_loss_function, metrics = ['accuracy'])

load_weights_from_FaceNet(facial_recognition_model)
```

This code snippet is for image training. This particular model Cv2 Classifier is used for just demonstration purposes. The ClusterGAN takes 2 hours to train.

6.3.3 Image Predicting

```
def find_person_identity(img, frame, x1_new, y1_new, x2_new, y2_new):

    h_max, w_max, color_channels = frame.shape
    # Display the bounding box
    part_video_frame = frame[max(0, y1_new):min(h_max, y2_new), max(0, x1_new):min(w_max, x2_new)]

    return who_is_the_person(img, part_video_frame, database, facial_recognition_model, x1_new, y1_new)

def who_is_the_person(frame, image, database, facenet, x1, y1):
    encoding = img_to_encoding(image, facenet)

    minimum_distance = 100
    persons_id = None

    # for each name in database
    for (name, database_encoding) in database.items():

        # calculate euclidean distance
        encoding_distance = np.linalg.norm(database_encoding - encoding)
        cv2.putText(frame, name, (x1 + 30, y1 - 10), font, 1, (120, 255, 120), 2, 1)
        print('distance for %s is %s' %(name, encoding_distance))

        # If this distance is less than the min_dist, then set min_dist to dist, and identity to name
        if encoding_distance < minimum_distance:
            minimum_distance = encoding_distance
            persons_id = name

    if minimum_distance > 0.49:
        return None
    else:
        return str(persons_id)
```

This code snippet is for image predicting image in real time.

6.4 Core Functionality of Research

The following details the core functionality of the research. This will display code snippets from the ClusterGAN system.

6.4.1 Generator and Discriminator

```
class Generator(nn.Module):
    def __init__(self, latent_dim, nr_classes,img_shape = (1,28,28)):
        super(Generator, self).__init__()
        self.img_shape = img_shape
        self.ishape = (128,7,7)
        self.dim = int(np.prod(self.ishape))
        self.network = nn.Sequential(nn.Linear(latent_dim + nr_classes, 1024),
                                     nn.LeakyReLU(0.2, inplace = True),
                                     nn.BatchNorm1d(1024),
                                     nn.Linear(1024, self.dim),
                                     nn.LeakyReLU(0.2, inplace = True),
                                     nn.BatchNorm1d(self.dim),
                                     View(self.ishape), #Reshape
                                     nn.ConvTranspose2d(in_channels = 128, out_channels = 64, kernel_size = 4, stride=2, padding=1),
                                     nn.LeakyReLU(0.2, inplace = True),
                                     nn.BatchNorm2d(64),
                                     nn.ConvTranspose2d(in_channels = 64, out_channels = 1, kernel_size = 4, stride=2, padding=1),
                                     nn.Sigmoid())
        initialize_weights(self)

    def forward(self, zn, zc):
        x = torch.cat((zn,zc), dim = 1)
        x = self.network(x)
        # reshape
        x = x.view(x.size(0), *self.img_shape)
        return x
```

```
class Discriminator(nn.Module):
    def __init__(self):
        super(Discriminator, self).__init__()
        self.logits_out = int(np.prod((128,5,5)))
        self.logits_conv_shape = (self.logits_out,)
        self.network = nn.Sequential(nn.Conv2d(in_channels = 1, out_channels = 64, kernel_size = 4, stride = 2 ),
                                     nn.LeakyReLU(0.2,inplace = True),
                                     nn.Conv2d(in_channels = 64, out_channels = 128, kernel_size = 4, stride = 2),
                                     nn.LeakyReLU(0.2,inplace = True),

                                     View(self.logits_conv_shape),

                                     nn.Linear(self.logits_out,1024),
                                     nn.LeakyReLU(0.2,inplace = True),
                                     nn.Linear(1024,1))
        initialize_weights(self)
    def forward(self, x):
        return self.network(x)
```

Figure 6.2 : Generator and Discriminator

6.4.2 Encoder

```
class Encoder(nn.Module):
    def __init__(self, latent_dim, nr_classes):
        super(Encoder, self).__init__()
        # bekommt 28x28 input image
        self.latent_dim = latent_dim
        self.logits_out = int(np.prod((128,5,5)))
        self.logits_conv_shape = (self.logits_out,)
        self.encoder = nn.Sequential(nn.Conv2d(in_channels = 1, out_channels = 64, kernel_size = 4, stride = 2 ),
                                     nn.LeakyReLU(0.2,inplace = True),
                                     nn.Conv2d(in_channels = 64, out_channels = 128, kernel_size= 4, stride = 2),
                                     nn.LeakyReLU(0.2,inplace = True),
                                     View(self.logits_conv_shape),
                                     nn.Linear(self.logits_out,1024),
                                     nn.LeakyReLU(0.2,inplace = True),
                                     nn.Linear(1024,self.latent_dim + nr_classes))
        initialize_weights(self)

    def forward(self, x):
        z_img = self.encoder(x)
        z = z_img.view(z_img.shape[0],-1)
        # Seperating continous and one-hot vectors
        zn = z[:, :self.latent_dim]
        zc_logits = z[:, self.latent_dim:]
        # Softmax on last 10 (nr classes) to obtain zc
        zc = torch.softmax(zc_logits, dim = 1)
        return zn, zc, zc_logits
```

Figure 6.3 : Encoder

The ClusterGAN network described above works, slightly different to the traditional GAN models. The workings of the traditional GAN models are thoroughly explored in the literature review. Cluster GAN has three networks as players in an adversarial game.

6.4.3 Manifold Learning techniques (Core)

```
from time import time
def make_manifolds(encoder_manifold, labels, title=None):
    manifold_discriminant, manifold_normal = np.min(encoder_manifold, axis=0), np.max(encoder_manifold, axis=0)
    encoder_manifold = (encoder_manifold - manifold_discriminant) / (manifold_normal - manifold_discriminant)

    plt.figure(figsize=(6, 4))
    for idx in range(encoder_manifold.shape[0]):
        plt.text(encoder_manifold[idx, 0], encoder_manifold[idx, 1], 'x',
                 color=plt.cm.nipy_spectral(labels[idx] / 10.),
                 fontdict={'weight': 'bold', 'size': 9})

    plt.xticks([])
    plt.yticks([])
    if title is not None:
        plt.title(title, size=17)
    plt.axis('off')
    plt.tight_layout(rect=[0, 0.03, 1, 0.95])


print("Generating embedding")
encoder_latent_z = manifold.SpectralEmbedding(n_components=2).fit_transform(enc)


from sklearn.cluster import AgglomerativeClustering

for linkage in ('ward', 'average', 'complete', 'single'):
    manifold_clustering = AgglomerativeClustering(linkage=linkage, n_clusters=10)
    t0 = time()
    manifold_clustering.fit(encoder_latent_z)
    print("%s :\t%.2fs" % (linkage, time() - t0))

    make_manifolds(encoder_latent_z, manifold_clustering.labels_, "%s linkage" % linkage)


plt.show()
```

```
import time
import matplotlib.pyplot as plt


from sklearn.cluster import AgglomerativeClustering
from sklearn.neighbors import kneighbors_graph
knn_graph = kneighbors_graph(enc, 10, include_self=False)

for connectivity in (None, knn_graph):
    for n_clusters in range(40):
        plt.figure(figsize=(10, 4))
        for index, linkage in enumerate(('average',
                                         'complete',
                                         'ward',
                                         'single')):
            plt.subplot(1, 4, index + 1)
            model = AgglomerativeClustering(linkage=linkage,
                                            connectivity=connectivity,
                                            n_clusters=10)
            t0 = time.time()
            model.fit(enc[:,0:2])
            elapsed_time = time.time() - t0
            if n_clusters == 39:
              break
            else:
              plt.scatter(enc[:, n_clusters], enc[:, n_clusters+1], c=model.labels_,
                          cmap=plt.cm.nipy_spectral)
              plt.title('linkage=%s\n(time %.2fs)' % (linkage, elapsed_time),
                        fontdict=dict(verticalalignment='top'))
              plt.axis('equal')
              plt.axis('off')

              plt.subplots_adjust(bottom=0, top=.89, wspace=0,
                                  left=0, right=1)
              plt.suptitle('n_cluster=%i, connectivity=%r' %
                           (n_clusters, connectivity is not None), size=17)


plt.show()
```

Manifold learning techniques such as agglomerative clustering, mean shift clustering and tSNE algorithms were used to achieve the latent space clustering results.

$$\min_{\Theta_G,\Theta_E} \max_{\Theta_D} \mathop{\mathbf{E}}_{x\sim\mathbb{P}_x^r} q(\mathcal{D}(x)) + \mathop{\mathbf{E}}_{z\sim\mathbb{P}_z} q(1-\mathcal{D}(\mathcal{G}(z))) + \beta_n \mathop{\mathbf{E}}_{z\sim\mathbb{P}_z} \|z_n - \mathcal{E}(\mathcal{G}(z_n))\|_2^2 + \beta_c \mathop{\mathbf{E}}_{z\sim\mathbb{P}_z} \mathcal{H}(z_c, \mathcal{E}(\mathcal{G}(z_c)))$$

Figure 6.4 : Novel triplet loss function for ClusterGAN

Algorithm

for i in ***(number_of_dimensions_of_latent_space – plotting_dimension)****:*

make_manifolds(encoder_latent_space)

plot_latent_space

6.4.3 Research Artifact Structure (Core)

When researching in deep learning, one of the most important things for other researchers to replicate your results is publicly share the results and research artifact structure. The structure of the artifact is really important in researches involving deep generative models for other researchers to quickly understand the research as well.

6.5 Research Documentation

The steps of the research were detailed as a narrative in the Google Colab notebook. This will be open sourced in a publicly accessible github repository. This will be pushed as a part of the code.

YOU: **So what's the deal with optimizing mathematically ?**

Most of the data we have are euclidean in nature. Let's take a simple Dogs-Cats dataset. We put those thousands of such data points into a function approximator (neural network) and train on deep CNN classifier. **Peanuts right !**.

Dozens of batches forward propagating and backward propagating and we have an epoch. A few more epochs later we have an acceptable error value. (Viola we found the lowest error value)

Take Case 1: This Dog-Cat classifier learn what makes an input a dog or a cat

Mathematically: What makes input image X to be of class Y1 (cat) or Y2 (dog)

By optimizing latent space vectors we can understand what makes the hidden representations of an image class similar across other images of same class. (with me ? if not take a look at the example below)

Take Case 2: The classifier took an input and wanted to know whether that input is a dog or a cat. But latent spaces of an image ask a different question. It look at a bunch of cat pictures and dogs pictures and asks what makes an input a cat, dog, or dinosaur etc.

Mathematically: What makes input image X1 and X2 similar OR image X1 and X3 different. What predominant features are present in images of a particular class.

What makes a bunch of dog images similar? OR what features of the images are retained in the saliency maps.

1. Is is pointed nose ?
2. Is it long mouth ? etc.

In Comparison with classifiers they ask a question "is this input image a cat or dog"?

Latent Space vectors ask a question "what makes 2 or more dog images similar" what are features of cat and dog ?

BUT yout get the point right. That is the fundemental question asked in almost any latent space vector based research.

YOU: **What's latent space optimization?**

Latent space vectors are the data structures that store information on WHAT makes an input a cat or a dog. By using mathematical methods to optimized them we can make the network do 2 things

1. Learn WHAT makes the inputs of similar classes.
2. Reveal learned information on classes better.

I'm doing the latter in my research.

YOU: "*shuffle nervously*" **Err so what's latent space interpolation?**

I AM Glad you asked. So we left of from the previous question with the idea that the latent spaces learn similar features of a same class.

Ex Dog - long nose, long mouth

Ex Cat - pointed ears, prominent whiskers

By interpolating latent spaces we can give one feature of a class to a another feature.

For Example: We can synthesise (or generate) an image of a dog with prominent whiskers.

YOU: "*smiles*" **Big Deal!! You're making a dog with whiskers LOL.What are other applications ? Giving cats dinosaur bones?**

Dinosaur bones we can discuss later. But think of it. After looking at a bunch of pictures of people. We can interpolate the latent spaces to generate images of people with beards, sunglasses etc.

YOU: **But why, why would you do tha..**

Alright think of it this way! By generating pics of a person with sunglasses or beards, or maybe aged versions of themselves we can get more data on a person. This will give us more data to train a classifier that recognises faces. By making sure that we have information on how the person might be looking with a beard, or maybe with a different haircut we can make the facial recognition systems more robust.

This will increase security as facial recognition system is a touchless system, or non invasive system. With other systems like fingerprint access system you have to physically touch the sensor.

This will be useful with pandemics such as the COVID-19 as well. We need more robustness in non invasive (touchless) authentication systems.

Figure 6.5: Research documentation

6.5 Problems Faced

This section documents the problems faced while developing the system. The problems and the way those were handles are as follows.

Problem	Solution
Extremely convoluted Model. The model of ClusterGAN was very complex in nature. Even though it had been previously implemented in PyTorch, it was complex to get it run smoothly in the Jupyter Kernels	A lot of time was spent exploring the code and various internal dependencies. The external dependencies were handled by using a online deep learning development environment such as google colab.
Extremely high training time. The training time required for the model was extremely high. It took about 3 hours to run the model on different datasets, with a model compression technique applied.	Model compression techniques were implemented to counteract the model size and inference speed. Tried to prune neuronal layers randomly to generalize the image as well.
OpenCV is a troublesome library to run. OpenCV is a C++ library that has a python wrapper. As a result, it was complicated to get up and running even with documentation.	The author explored the bug fixes on stackoverflow during implementation of the OpenCV modules. It appeared that the PIP has 4 versions of opencv for python. (https://pypi.org/project/opencv-python/). This means the pip wheels gets rebuilt every time a version gets installed. So, we removed the unnecessary wheels and installed the correct one.

Table 6.4: Problems faced while development

6.6 Chapter Summary

This chapter is all about the implementation details and decisions. There will be implementation decisions that go beyond the design decisions. We started by surveying available image datasets. Then programming languages and deep learning libraries were surveyed. We then selected the IDEs and version control systems for development. Then we showed a few snippets of code from the research and product developed. The documentation of the research was displayed. Finally, the chapter closed off with the problems and blockers faced while implementations.

CHAPTER 7 : TESTING

7.1 Chapter Overview

This chapter discourses on the testing methods for this dissertation. Since this research is a machine learning research and we are dealing with non-deterministic priors we will be using similar metrics to test and evaluate this research and system. As we did in the design and implementation chapter, we will dissect the testing into 2 parts. One dealing with the testbench for the research and the other for the application developed.

7.2 Goals and Objectives of Testing

The objectives of testing are twofold. We have to ratify and see the results of the research, and we also have to see whether the system developed is functioning as expected. The following goals were defined to compartmentalize the testing procedure.

1. Ratify functional requirements of the system.
2. Ratify non-functional requirements of the system.
3. Reduce bugs and feature defects in the code developed.
4. Enhance input and output validations.

7.2.1 Testing Criterion

Software can be tested following two paradigms of criterion.

1. Software functional testing – testing to see if software functions as expected.
2. Code structural tests – testing to see if the code was developed in a orderly manner following best practices in software engineering.

7.2.2 Testing methodologies

Software testing can have various metrics of testing. A few of the popular metrics are mentioned below.

Test Name	Description	Black / White box testing	Status
Unit testing	to check individual modules of software	White	Checked
Integration testing	to check whether combined systems of individual modules work properly	Black and White	Checked
Stress testing	Check how system performs in unfavorable conditions (slow network, high RAM usage, CPU throttling)	Black	Checked
Performance testing	Check the speed and waiting times by the system	Black	Checked
Usability testing	To check in the perspective of a client	Black	Checked
Regression testing	Regression testing is carried out to check of any changes in the system is not destructive to the other modules in the system	White	Checked
Beta testing	Done by end users	Black	Not Checked
Legal and Ethical testing	A project on biometric data should be tested for compliance with General data protection regulation act (GDPR)	White	Checked

Table 7.1 : testing methodologies

7.3 Functional Testing

The test plan of the black box testing and its results are as follows,

Test case #	Description	Input Data / User action	Expected outcome	Actual outcome	Status
Testing Facial Detection System					
1	Register person	Enter use data	User data validated and saved	Same as adjacent	Pass
2	Web camera or any infra-red camera feed	Camera connected to the system	Display bounding boxes	Display bounding boxes	Pass
3	Face detection	Face detected, and cropped from camera feed	Display bounding boxes	Display bounding boxes	Pass
4	Face feature extraction and consolidation	Face cropped images	Vectorise features and Create dataset	Vectorise features and Create dataset	Pass
5	Train the model	Click train model	Train model on face features	Train model on face features	Pass
6	Recognize faces	Recognize faces	Recognize faces	Recognize faces	Pass
7	Delete person data	To be compliant with GDPR	User data deleted	User data deleted	Pass
8	Generate attendance report	Click to generate report	Report Generated	Report Generated	Pass
Testing Research Component					
9	Spectral Clustering	Run Colab cell	Generate 3D latent space for each individual component in the latent space.	Generate 3D latent space for each individual component in the latent space.	Pass
10	Mean-Shift Clustering	Run Colab cell	Generate 3D latent space for each individual component in the latent space.	Generate 3D latent space for each individual component in the latent space.	Pass
11	Agglomerative clustering	Run Colab cell	Generate results for each separate cluster, in for different vantage points. (Average, Complete, Ward, Single)	Generate results for each separate cluster, in for different vantage points. (Average, Complete, Ward, Single)	Pass
12	Manifold Learning on	Run Colab cell	Generate Spectral embedding for	Generate Spectral embedding for each	Pass

	latent space vectors		each class of classification.	class of classification.	
13	Generate Model files for the three developed networks	Run Colab cell	Creates ONNX models and PTH models.	Creates ONNX models and PTH models.	Pass
14	Extend future works.	Run Colab cell	Explores future work	Explores future work	Pass

Table 7.2: Functional Testing Plan

$$\text{Successful functional tests} = \frac{14}{14} = 100\%$$

7.4 Nonfunctional testing

We will be taking a look at the nonfunctional testing from 2 vantage points. Research findings and performance.

7.4.1 Performance

The research carried out is a high end research requiring extremely specialized and high end hardware. Here we will only be using the CPU and the RAM but we will also be using the Tesla K80 GPU with 11GB of VRAM from the Google Colab. Most of the development time is taken to develop the appropriate code and loss functions. The most time-consuming part of running the code is training the system.

The two figures below depict the CPU and GPU usages respectively.

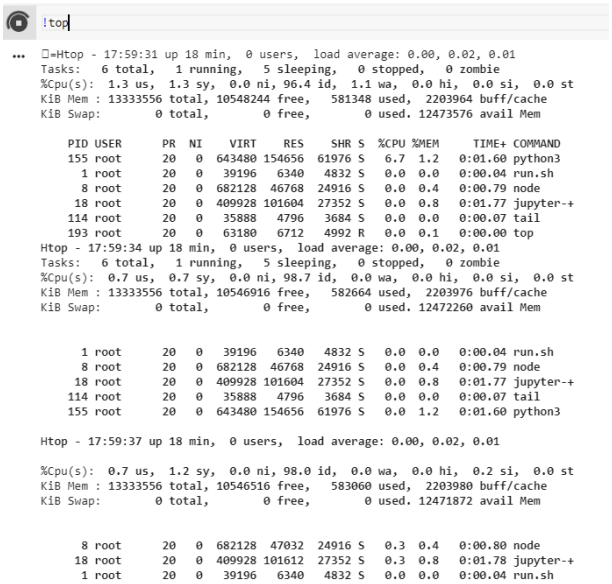


Figure 7.1 : CPU Usage profile

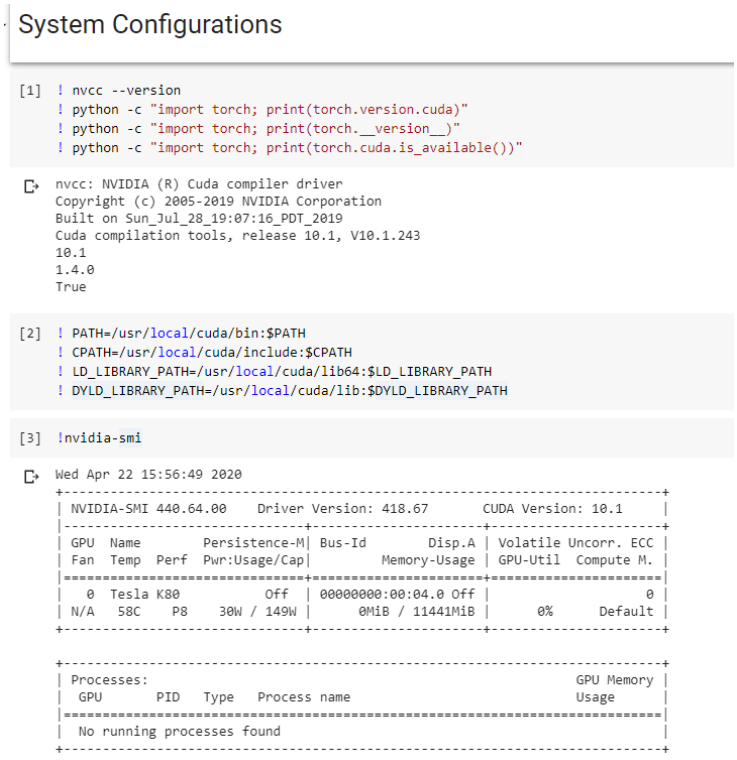


Figure 7.2: GPU profile

The images above sorry show the CPU utilization and the GPU memory utilization for the research component of the project. ClusterGAN is an extremely taxing program in terms of resource allocation. The GPU VRAM usage is nearly 8 GB/ 11 GB., and the GPU utilization is at a maximum most of the time.

Most computationally expensive module in the program is the training. Consolidation of certain data takes more time as well. Overall the system is performing as expected. Idle times are typically similar to what can be expected from a remote Linux computer.

7.4.2 Performance of Facial Detection System

The project developed is a Facial Detection system. The stress test and the performance test is defined as below. Stress test was conducted by keeping a resource intensive application (Google

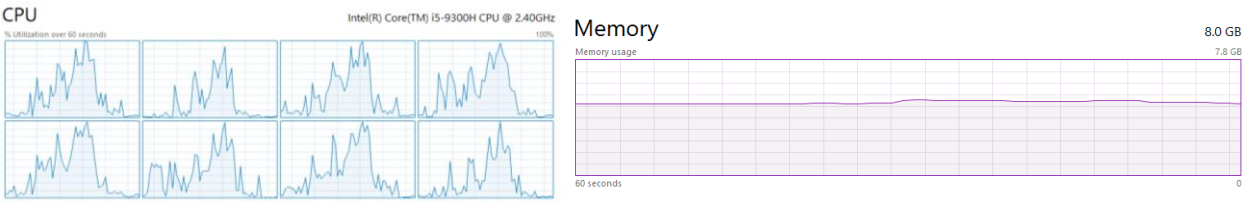


Figure 7.3 : CPU and RAM usage profile

Chrome) open while running the application. Since its doesn't depend on a network, the stress test was limited to only local resources of the host machine. From the images above, it is evident that the application causes expected spikes in CPU usage, and barely causes any spike in memory.

7.5 Integration Testing

Modules	Input	Expected Output	Actual Output	Status
Person data Input Validation	Details	System checks for any fraudulent input	System checks for any fraudulent input	Pass
Feature Preprocessing	Face images	Vectorize images	Vectorize images	Pass
Dataset multiplicity	Feature vectors	Multiple ways to store data should be facilitated in the same program	Only one way to store data is facilitated in the same program	Fail
Captured Images Display	Images	Display Captured images as a carousel	Displaying captured images from memory is a concurrent operation with unstable results	Pass
Reports on the attendance generated	Facial recognition data	Generates a CSV based on the faces detected.	Generates a CSV based on the faces detected.	Pass

Table 7.3: Integration Testing Plan

$$\text{Successful Integration tests} = \frac{4}{5} = 80\%$$

7.6 Testing GAN

GAN evaluation metrics are covered in a modern approach (Chen et al., 2017). Here the authors compare the various ways to evaluate image generative models. They also comment about the latent spaces. The authors concur that training a GAN with a Euclidean dataset of low informational density will make the latent space densely covered. This makes this work the perfect test to evaluate this research. The main way to evaluate this type of unsupervised manifold learning model is to implement the Frechet Inception Distance or FID Score for the model.

1. Comparative Benchmarking – Comparing with existing researches.
2. Research Objective Benchmarking – Verifying research results for assertion.

7.6.1 Competitive Benchmarking

Here we will be covering some of the researches that use similar metrics. The table below displays the FID scores of existing work and comparisons drawn with ClusterGAN.

Research Name	Dataset		
	MNIST	Fashion MNIST	CIFAR-10
ClusterGAN	0.81	0.91	

CR-GAN	N/A	N/A	14.56
WGAN – improved (one-hot)	0.94	6.14	24.8
Wassertein GAN	0.88	0.95	29.3
Info GAN	1.88	11.04	N/A

Table 7.4 : Comparison of FID scores of GAN

7.6.2 Benchmark Metric Explanation

This is an unsupervised learning approach on image generative models and synthesized images. Since we have used to benchmark our model in the previous section, we need to explore the Frechet Inception Score or FID score in detail to justify our claims of benchmark. I used 5 runs to estimate the above score, to eliminate any outliers.

We will take a mathematical approach FID score was initially introduced in 2017 (Heusel et al., 2018). The FID function is described here.

$$d^2((\boldsymbol{m}, \boldsymbol{C}), (\boldsymbol{m}_w, \boldsymbol{C}_w)) = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \mathrm{Tr}\big(\boldsymbol{C} + \boldsymbol{C}_w - 2\big(\boldsymbol{C}\boldsymbol{C}_w\big)^{1/2}\big) \ .$$

7.6.3 Research Objective Benchmarking

Here we will be displaying some results obtained by manifold learning on the MNIST dataset by ClusterGAN using the algorithm we proposed. The research results are appended in the **Appendix F and Appendix J.**

7.8 Chapter Summary

This chapter discourses on testing the developed product and research. We defined objectives and methodologies for testing. We showed results of functional and non-functional tests. We conducted performance and stress testing of the developed application as well. We were also able to bring the training time down to less than 5 minutes using model compression and multi-processing. We benchmarked the research against existing researches and justified the benchmarking metric.

CHAPTER 8 : EVALUATION

8.1 Chapter Overview

The evaluation chapter discourses on evaluations of the research by the domain experts. They comment on various aspects of the research including the product developed, research decisions and conclusions reached. This chapter contains reviews and comments from domain experts regarding this research.

8.2 Evaluation Methodology and Approach

This research has to be evaluated in both qualitative and quantitative approaches. In the previous chapter on testing, we evaluated the research against other researches qualitatively. In this chapter we will be evaluating the research carried out quantitatively by getting feedback from experts and researchers in this domain.

8.3 Evaluation Criteria

Quantitative evaluation will be carried out. The criterion will be outlined below.

Criteria	Purpose to evaluate
Overall project concept	This is done to ratify the project topic is a relevant topic in the domain.
Scope and depth of the project	The scope is ratified by the experts to validate the project.
Theoretical Literature Review	Review theoretical concepts before implementing to justify the theoretical grounding.
System design, architecture and implementation	To see whether the architecture of the system implemented conforms to the general software engineering practices of the domain.
Extensibility of the application	The product component of the system is a public repository for other developers to use it as a boilerplate code.

Table 8.1: Evaluation Criteria

Despite the focus of this feedback form being quantitative in nature, we will also collect data which is qualitative in nature.

8.4 Selected Evaluators

The individuals chosen to evaluate this research were compartmentalized into the following 4 categories.

1. Beginners in image generative models
2. Intermediate AI researchers of generative models (people with <= 2 years of experience)

3. Industry Expert AI researchers of generative models (people with experience)
4. Expert AI researches with publications to their name (mostly people with PhDs in ML)

Based on these categories, the evaluators were reached out as follows.

Group	Details	Reason
Academia Experts	**Dr. Rukshan Batuwita** PhD in Machine Learning (Oxford), Data Scientist at Google.	Dr. Rukshan has been involved in machine learning for nearly the last 15 years. He has a PhD from Oxford in machine learning and is currently working as a data scientist in Google. He has also been the Lead Data Scientist in Woolsworths Group in the past.
	Hansa Perera Mathematician and Machine Learning Researcher, University of Colombo Sri Lanka	Hansa is a mathematician. He has been working as a Data Scientist for the past 15 years. He is well versed in data science practices and methodologies.
	Latefa Elboujdaini PhD Student Faculté des Sciences Oujda Morocco	Latefa is a PhD student in the artificial intelligence and energy conservation in solar panels.
Industry Experts	**Sharmilan Somasundaram** MSc Big Data Analytics Visiting Lecturer, Senior Software Engineer 99X.	Sharmilan is a MSc in Big Data Analytics. He is also a visiting lecturer to many institutes including IIT. He is a project supervisor as well.
	Raveen Savinda Rathnayak Software Engineer WSO2	Raveen is a Software Engineer at WSO2.
Intermediate Researchers	**Savindu Dias** Software Engineer WSO2	Savindu is a Software Engineer at WSO2.
	Gayashan Bombuwala Software Engineer WSO2	Gayashan is a Software Engineer at WSO2.
Beginner Researchers	**Hasal Fernando** DataScience Intern Zone24x7	Data Science intern in Zone24X7
	Safiyyah Thur Rahman 2nd year Student, IIT	Sophomore student in IIT.

Table 8.2: Selected Evaluators

Evaluation was performed by these evaluators.

8.5 Evaluation Results

8.5.1 Project Scope and Depth Concept

Question	
What do you think of the depth of this research?	
Person	**Feedback**
Rukshan Batuwita	The paper presents a comparative analysis of latent space vectors and optimization methods of the popular image synthesizing GAN models. The paper starts by introducing GANs, their applications and popular architectures. Then it presents a thorough analysis of different latest space vector representation methods used in popular GAN architectures such as COCO-GAN, Style-GAN etc. Then it talks about existing gaps identified in the area in terms of both applied and fundamental research. Finally the paper concludes by talking about the gaps in this survey and future research directions. Overall the paper is written well and presents a very clear, concise and useful survey of the topic being addressed. The information summarized in Table 1, 2 and 3 is very helpful, not only for someone entering in to this area of research but also for existing researchers. Authors seems to have analyzed the literature in the area deeply and managed to articulate their findings successfully. Very well done!
Hansa Perera	Good comparison of different variants explained in a manner to understand quickly.
Latefa Elboujdaini	This research is interesting and deal with future improvement.
Sharmilan Somasundaram	It has a good research depth.
Raveen Savinda Ratnayake	The researcher has done a good amount of research and he has good understanding on the research area. He has critically evaluated the existing methods and found a reasonable

	research gap. As a continuation of this research, I would like to see more a more mathematical analysis of the research area.
Savindu Dias	I do not have a string understanding on GANs. However, the paper produces a thorough overview and analysis in comparing LSVs in popular GAN variants. Additionally, the explanations of LSVs provide a clear understanding to any reader who may be unfamiliar with the topic.
Gayshan Bombuwala	The depth of your research seems a perfect fit for a bachelors degree.
Hasal Fernando	The latent space vectors of state-of-the-arts GAN variants are very thoroughly explored in this paper. Latent space vector distributions of the relevant GAN variants are also properly mentioned. Writer seems to have explored the different techniques used by different GAN variants to model latent space vectors and the optimization methods of latent space vectors. The writer has stated them so clearly, that the reader would have a clear understanding of what those techniques and optimization methods are capable of. In summary, it looks that the depth of research is so remarkable, that it covers most of the important state of the art GAN variants, their distributions, the different techniques and optimization methods that are used in a those variants, in a felicitous way.
Safiyyah Thur Rahman	It is appropriate for an undergraduate.

Table 8.3: Research depth evaluation

Evaluation Summary

The evaluators felt that the project depth and research depth is more than adequate for research on image generative models. In particular one evaluator reviewed the paper the author wrote and gave positive feedback.

8.5.2 Suggested improvements on the research

Question	
Any suggestions to improve and continue the research?	
Person	**Feedback**
Hansa Perera	Mathematical explanations would add more value to the paper
Rukshan Batuwita	Although overall the paper has been written well, you may consider to readability of the paper further. For example: 1. In Abstract: "This paper dissects the latent space vectors of state-of-the-arts GAN variants, for image synthesis. We also compare and contrast optimisation methods of latent space vectors in existing literature. In this paper, we undertake a comparative analysis of latent space vectors and optimisation methods of the popular image synthesising GAN models." --> Seems like the same information is repeated twice. 2. 'Table 1 Summary of latent space vectors of GAN variants' --> include this together with the Table 1, not the table Title is in a separate page. 3. State clearly the 'Future Work' in Table 1 are taken from those papers and not something you came up with (if I understand correctly).
Gayashan Bombuwala	Try to write a couple more papers based on your research if possible.
Latifa Elboujdaini	- Improve this survey to a review - Discuss more about the proposed models and approaches

Table 8.4: Improvements evaluation

Evaluation Summary

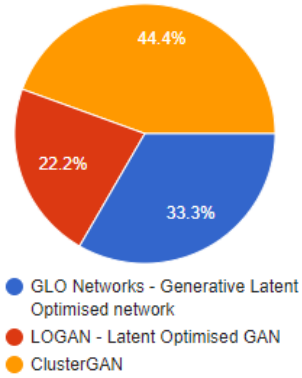
Some evaluators surmised that certain improvements could be made to the research, some of the suggested changes were facilitated.

8.5.5 Limitations

Evaluation on limitations are presented in **Appendix I**

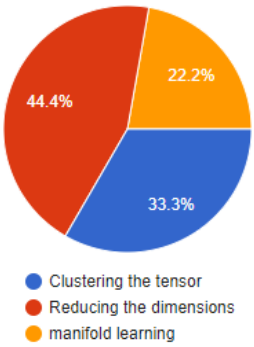
8.6 Quantitative Results

8.6.1 The author has reviewed 3 existing latent space optimization techniques. Which do you feel is the best in terms of latent vector optimization?



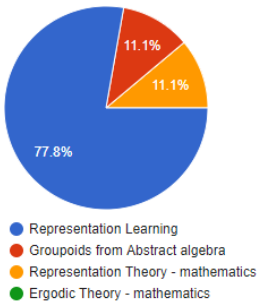
All the evaluators have answered this query in the form. Most of them especially the experienced experts from the industry and academia are concluding that ClusterGAN is the best in the 3 to continue the research. This gives us good statistical priors to carry out the research. The evaluators have provided nearly 44% of votes to ClusterGAN

8.6.2 Any ways to optimize a high dimensional tensor? what would you prefer?



The question asked was whether the evaluators knew anyways of optimizing a high dimensional tensor. The options provided to the evaluators were manifold learning, dimensionality reduction and clustering the tensor. We can see that nearly 44% of the evaluators have voted in favour of reducing the dimensions and 33% to clustering the tensors and just 22% in manifold learning.

8.6.3 Of the few research GAPs in the future work section of the paper, which one appeals to you most? Or do you have any other suggestions?



The question was on future research suggestions for latent space vector optimization. The option given were representation learning, groupoids from abstract algebra, representation theory from mathematics and ergodic theory from mathematics. The use of any 4 of the above would facilitate newer researches in latent space vector optimization. We can see the most of the evaluators have chosen representation learning as the future of this particular field.

8.9 Self-evaluation of the author

Reflecting on the work I've done since I have started the research almost a year back, I feel like I have come a long way. There were of course setbacks, the path to this research was not easy at all, at times I felt being stuck and claustrophobic. Still there are many aspects of the research and the product that was developed that could be improved. There were many decisions taken by the author out of necessity. The following is the author's evaluation.

Criteria	Author's Evaluation
Overall project concept	The concept of tweaking latent space vectors of images is a recent field that cropped up. There were very few researches by the time the author started his research. Similar researches were cropping up along while the author was researching. Overall the author decided to use the research component to make an easy to use and extendable person detection system. There were person detection systems available, but they were not usable by users they just ran out of a single script.
Scope and depth of the project	The depth of the research was thoroughly commended by the evaluators. But the knowledge and insight taken of this research was worth more than it could be express.
Theoretical Literature Review	This project needed extremely intensive literature review. It had to be done for various different components and modules both for the theoretical applications and implementational application. This criterion was crucial in developing research skills and criticizing existing works.
System design, architecture and implementation	The prototype was designed by following standard practices in software engineering. OOP principles and SOLID principles were followed. When there was a necessary concurrent design patterns were followed as well.
Extensibility of the application	The prototype was made as a boilerplate code for other developers. Therefore, we made it fully modularized and followed all the standard practices in software engineering. This will make it easier for others.

Table 8.5: Authors Evaluation

8.10 Chapter Summary

This chapter started with defining evaluation criterions. Then it portrayed the comments by varying levels of researchers. The evaluations were in respect to the research concept, research scope, and the prototype. Some limitations were outlined. Finally, the chapter end with the self-evaluation presented by the author.

CHAPTER 9 : CONCLUSION

9.1 Chapter Overview

This chapter discourses on the conclusionary remarks about the project. We go through make reviewing remarks on the initial goals and objectives, challenges overcome, usage of skills by the author, contribution by the university courses and new knowledge and skills gained. The contribution to the research and product development community is also documented here.

9.2 Achievement of project aim and objectives

9.2.1 Aim of the project

This research aims to design, develop and evaluate a person identification system robust to adversarial attacks by exploring the various properties of latent space vectors in GANs, and optimize it using Manifold Learning for controlled Image Synthesis. Furthermore, this research proposes to use the knowledge and intuition gained from the research to further stabilize person recognition systems.

The aim of the project was achieved from 2 vantage points. First the research component of this project achieves to extract the useful features in a dataset of images. Then the product component of this research makes use of such a model to stabilize and increase the robustness of person / facial recognition systems.

9.2.2 Contribution and advantages of the system relevant to current times

Currently the world is facing a pandemic. Hence, we need more non-invasive, touchless systems. This research gives an open source model of secure person recognition system that operates touchless. This means that current fingerprint registration systems are compromised to the pandemic COVID-19 virus. However, this system can be used to mitigate fingerprint access and yet still have a secure recognition system.

9.2.3 Completion of objectives of the project

Description	Status
Literature Review	
Intensive literature review on project components	**Completed**
System Requirement Gathering	
Carry out appropriate SRS to develop the person recognition system	**Completed**
System Design Specification	
Carry out appropriate SDS to develop the person recognition system	**Completed**
System Implementation	
Develop the system to be open-sourced and public access	**Completed**

Testing and Evaluation	
Test and Evaluate the system	**Completed**

Table 9.1: Completion of objectives of the project

9.3 Utilizing experience from modules in the degree

Module	**Description**
Programming Principles I & II	These modules taught the basic programming principles. Basic data structures and algorithms were covered here. This module was instrumental in choosing the appropriate language and design decisions.
Computer Science Practice	This module was instrumental in making some of the research interests of this research. This module introduced to concepts such as OLAP and OLTP systems. This was important in this research. This module also helped the oral presentation skills.
Object Oriented Programming	This module was important in modularizing the application. This module introduced to OOP design principles and design patterns. The coursework was important to understand many things in this module.
Algorithms: Theory, Design and Implementations.	The module taught more heavy algorithms including searching, sorting and graph traversal. This experience was important in the programming experience. This enabled us to think of space and time complexity before implementing algorithms.
Software Development Group Project	This module was a research module. It was a nice prelude to the final year project. We developed a system together. We also learnt to work as a team.
Concurrent Programming, Database Systems	The data storing concepts followed in the system were from some of the knowledge gained from this module. Even though the system doesn't require database stores, the data management were necessary.
Computer Science Fundamentals, Client-Server architecture.	This module was important in teaching the networking fundamentals and the webservers. Here we understood the web architecture which was important in understanding many Linux systems.

3D Graphic Programming	Although 3D Graphic Programming is not part of this domain, the linear algebra and matrix mathematics learnt in this module proved important for deep learning research.

Table 9.2: Modules in the degree

9.4 New Skills Acquired

The following are the new skills obtained.

Deep Learning – Author was selected on a scholarship by Facebook and Udacity to a special Nanodegree program on deep learning and image generative models. This proved instrumental in the author reaching this particular research gap.

Mathematics – Author learned mathematics as a hobby. He learnt them from completely free resources such as 3blue1brown, Khan academy and may varied sources.

GAN networks and Representation learning – The author gained valuable expertise in image generative models.

9.5 Learning Outcomes

Learned Skills Descriptions	**Learning Outcomes**
The most important thing I am taking away from this module is the mindset of a researcher. The research methodologies and problem analyzing are vastly important the in present job market. This module gave skills to properly review existing literature critically and semantically.	LO1, LO2
This project was on image generative models. This meant the author had to learn mathematical concepts to understand before carrying out any changes. Image generative models are also a niche in deep learning. So, the author had to learn data preprocessing, model hyper parameter tuning and neural architecture search.	LO3, LO4
This module was instrumental in my understanding and ability to write research papers. Academic writing and academic reading were an important skill developed during this time.	LO5, LO6
The author was able to independently search, diagnose and solve problems. The process of this research was not easy at all, but I think that's what made this so challenging and rewarding in the end.	LO3, LO4
This module also taught me how to formerly make questionnaires for quantitative research. The survey techniques, information gathering techniques were very useful and will be useful to me as a researcher.	LO7, LO8

Table 9.3 : Learning outcomes Acquired

9.6 Limitations and Challenges faced

There were many challenges faced during the development and research of the project. Some were easily handled, but others needed more focused attention.

Limitation / Challenge	Description	Solution
Challenges		
Expansive Scope	Extremely challenging and mathematical scope.	To manage the huge scope intensive literature review was done.
Huge learning curve	Learning curve was huge.	Started development a month early
Hardware Requirements	Image generative models needed huge hardware requirements.	A well-equipped laptop was acquired, Google Colab was made use of as well.
Training time	Generative models require long training time.	Planned the training tests overnight. Used the student AWS account well.
Evaluation personnel	This field has few experts to get feedback and advice from.	Consulted campus lecturers and supervisor. Found people working in this field.
Project Time	Project has time constraint considering the huge scope.	Reduce scope and managed time properly.
COVID-19	The global pandemic caused wide spread quarantine. Being shut inside takes a toll on the mind.	Start work early in a timely manner to manage unprecedented situations.
Limitations		
Only Image generative models	Only images considered in the research	Could consider other data types such as audio, time series, video and relational data.
Flow-Based models	Networks such as transformers weren't taken due to small LSV.	Future work could consider this. There was lack of literature in 2019.
Disentangled models	Disentangled representations are interesting.	Huge topic and out of the scope of this project.

Table 9.4 : Challenges and Limitations

9.7 Future Enhancements

All researches and products have room for improvement. The following listed ideas are ways to improve the both the research and product components. It is also a crucial part of research to identify the limitations and future enhancements.

- Review more GAN variants for different types of datasets, such as audio video and relational data.
- Empirically test with more loss functions, model architectures and optimization algorithms.
- Explore recent generative models such as transformer models, and the variants that follow it. Reformer model shows a lot of promise in being used as generative models. More evolutionary models could have been explored as well. Evolutionary computing makes use of automata theory to find patterns and behaviors. We could utilize such systems to some hyperparameter optimization or some neural architecture optimizations.
- We looked into improving the performance and accuracy of person recognition systems in a theoretical way. There could be other "hacks" that could be utilized as well.
- We used model compression techniques to compress the model to be able to deploy. However, we could look into further compression techniques to be able to deploy in resource constrained devices such as FPGAs.
- More work on mathematical models such as Ergodic theory, Representation theory, abstract algebra and manifold learning research.
- More python libraries such as Numba and Cython could be used to make the GUI faster
- Data collection mechanism could be refined for use with a varied data ingression formats of different models.

9.8 Introspection on research

It is said that life is lived forwards, but understood backwards. That is true for research as well. Looking back, I have come a long way forward. I have learnt stuff that I wanted to learn during this research. I have also learnt skills that I didn't think that would be needed during this research. In the end I'm satisfied with the artifacts of this module that I was able to produce.

I made research contributions in utilizing manifold learning for the latent space vectors of GAN models. I was able to find out which feature vectors are densely or sparsely populated. I was also able to make use of a component of my research to make a secure person detection system. This will be particularly useful as most biometric detection systems use fingerprint. Fingerprint is a touch-based authentication. During pandemics such as the COVID-19 it would pose

additional risks to the users. I was able to publicly open source the system for other developers to start making their own secure recognition systems.

I also improved some dev-ops skills. I'm now able to pipe processes and handle concurrent operations with more confidence and expertise. All this exposure and experience has me looking forward to continue my career and research.

9.9 Chapter Summary

This chapter concludes this research project. Here we ratify whether the aims and objectives of the research are achieved. We also discuss how the degree and other resources helped the author to conduct this research. We also describe the learning outcomes of this research, limitations and challenges faced by the author. Finally, we discuss future directions for research and the social, legal, ethical and professional implications of this research. Finally, we close off by having a look at the authors introspections on doing this project.

References

Bau, D., Zhu, J.-Y., Strobelt, H., Zhou, B., Tenenbaum, J.B., Freeman, W.T., Torralba, A., 2018. GAN Dissection: Visualizing and Understanding Generative Adversarial Networks. arXiv:1811.10597 [cs].

Bojanowski, P., Joulin, A., Lopez-Paz, D., Szlam, A., 2017. Optimizing the Latent Space of Generative Networks. arXiv:1707.05776 [cs, stat].

Bonabeau, E., Dorigo, M., Theraulaz, G., 1999. From Natural to Artificial Swarm Intelligence. Oxford University Press, Inc., New York, NY, USA.

Brock, A., Donahue, J., Simonyan, K., 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. arXiv:1809.11096 [cs, stat].

Brown, R., 1987. From Groups to Groupoids: a Brief Survey. Bulletin of the London Mathematical Society 19, 113–134. https://doi.org/10.1112/blms/19.2.113

Brownlee, J., 2019. 18 Impressive Applications of Generative Adversarial Networks (GANs). Machine Learning Mastery. URL https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/ (accessed 10.9.19).

Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B., Dafoe, A., Scharre, P., Zeitzoff, T., Filar, B., Anderson, H., Roff, H., Allen, G.C., Steinhardt, J., Flynn, C., hÉigeartaigh, S.Ó., Beard, S., Belfield, H., Farquhar, S., Lyle, C., Crootof, R., Evans, O., Page, M., Bryson, J., Yampolskiy, R., Amodei, D., 2018. The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation. arXiv:1802.07228 [cs].

Burt, P.J., Adelson, E.H., 1987. The Laplacian pyramid as a compact image code, in: Readings in Computer Vision: Issues, Problems, Principles, and Paradigms. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 671–679.

Chen, N., Klushyn, A., Kurle, R., Jiang, X., Bayer, J., van der Smagt, P., 2017. Metrics for Deep Generative Models. arXiv:1711.01204 [cs, stat].

Child, R., Gray, S., Radford, A., Sutskever, I., 2019. Generating Long Sequences with Sparse Transformers. arXiv:1904.10509 [cs, stat].

Denker, M., Grillenberger, C., Sigmund, K., 2006. Ergodic Theory on Compact Spaces. Springer.

Di, X., Patel, V.M., 2018. Face Synthesis from Visual Attributes via Sketch using Conditional VAEs and GANs [WWW Document]. undefined. URL https://www.semanticscholar.org/paper/Face-Synthesis-from-Visual-Attributes-via-Sketch-Di-Patel/91e9e19a06614197b5431410cecd29762223e04e#citing-papers (accessed 4.9.20).

Donahue, C., Lipton, Z.C., Balsubramani, A., McAuley, J., 2018. Semantically Decomposing the Latent Spaces of Generative Adversarial Networks. arXiv:1705.07904 [cs, stat].

Ergodic Theory - an overview | ScienceDirect Topics [WWW Document], n.d. URL https://www.sciencedirect.com/topics/mathematics/ergodic-theory (accessed 4.10.20).

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014a. Generative Adversarial Networks. arXiv:1406.2661 [cs, stat].

Goodfellow, I.J., Shlens, J., Szegedy, C., 2014b. Explaining and Harnessing Adversarial Examples. arXiv:1412.6572 [cs, stat].

Groupoids: Unifying Internal and External Symmetry, 1996. 43, 9.

Han, S., Mao, H., Dally, W.J., 2016. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. arXiv:1510.00149 [cs].

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S., 2018. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. arXiv:1706.08500 [cs, stat].

Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., Belongie, S., 2016. Stacked Generative Adversarial Networks. arXiv:1612.04357 [cs, stat].

Huo, X., Smith, A., 2008. A Survey of Manifold-Based Learning Methods. Recent Advances in Data Mining of Enterprise Data. https://doi.org/10.1142/9789812779861_0015

Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K., 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv:1602.07360 [cs].

Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A., 2016. Image-to-Image Translation with Conditional Adversarial Networks. arXiv:1611.07004 [cs].

Karras, T., Aila, T., Laine, S., Lehtinen, J., 2017. Progressive Growing of GANs for Improved Quality, Stability, and Variation. arXiv:1710.10196 [cs, stat].

Karras, T., Laine, S., Aila, T., 2018. A Style-Based Generator Architecture for Generative Adversarial Networks. arXiv:1812.04948 [cs, stat].

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T., 2019. Analyzing and Improving the Image Quality of StyleGAN. arXiv:1912.04958 [cs, eess, stat].

Kingma, D.P., Welling, M., 2019. An Introduction to Variational Autoencoders. FNT in Machine Learning 12, 307–392. https://doi.org/10.1561/2200000056

Ko, J., Kim, E., Byun, H., 2002. A Simple Illumination Normalization Algorithm for Face Recognition, in: Ishizuka, M., Sattar, A. (Eds.), PRICAI 2002: Trends in Artificial Intelligence, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 532–541. https://doi.org/10.1007/3-540-45683-X_57

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks, in: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 25. Curran Associates, Inc., pp. 1097–1105.

Kurzweil, R., 2006. The Singularity Is Near: When Humans Transcend Biology. Penguin (Non-Classics).

Laine, S., 2018. Feature-Based Metrics for Exploring the Latent Space of Generative Models.

Lecun, Y., 1988. A theoretical framework for back-propagation. Proceedings of the 1988 Connectionist Models Summer School, CMU, Pittsburg, PA 21–28.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539

Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W., 2016. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. arXiv:1609.04802 [cs, stat].

Lee, D.D., Seung, H.S., 1999. Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791. https://doi.org/10.1038/44565

Li, Z., Hoiem, D., 2017. Learning without Forgetting. arXiv:1606.09282 [cs, stat].

Lin, C.H., Chang, C.-C., Chen, Y.-S., Juan, D.-C., Wei, W., Chen, H.-T., 2020. COCO-GAN: Generation by Parts via Conditional Coordinating. arXiv:1904.00284 [cs, stat].

Lin, J., Rao, Y., Lu, J., Zhou, J., 2017. Runtime Neural Pruning, in: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 30. Curran Associates, Inc., pp. 2181–2191.

Linear representations of finite groups : Serre, Jean Pierre : Free Download, Borrow, and Streaming : Internet Archive [WWW Document], n.d. URL https://archive.org/details/linearrepresenta1977serr (accessed 2.25.20).

Liu, S., Sun, Y., Zhu, D., Bao, R., Wang, W., Shu, X., Yan, S., 2018. Face Aging with Contextual Generative Adversarial Nets. arXiv:1802.00237 [cs].

Liu, Y., Wang, Y., Wang, S., Liang, T., Zhao, Q., Tang, Z., Ling, H., 2019. CBNet: A Novel Composite Backbone Network Architecture for Object Detection. arXiv:1909.03625 [cs].

Maaten, L. van der, Hinton, G., 2008. Visualizing Data using t-SNE. Journal of Machine Learning Research 9, 2579–2605.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B., 2016. Adversarial Autoencoders. arXiv:1511.05644 [cs].

Mukherjee, S., Asnani, H., Lin, E., Kannan, S., 2019. ClusterGAN : Latent Space Clustering in Generative Adversarial Networks. arXiv:1809.03627 [cs, stat].

Nam, S., Kim, Y., Kim, S.J., 2018. Text-Adaptive Generative Adversarial Networks: Manipulating Images with Natural Language, in: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 31. Curran Associates, Inc., pp. 42–51.

Odena, A., 2019. Open Questions about Generative Adversarial Networks. Distill 4, e18. https://doi.org/10.23915/distill.00018

Omar, M., Mehmood, A., Choi, G.S., Park, H.W., 2017. Global Mapping of Artificial Intelligence in Google and Google Scholar. Scientometrics 113, 1269–1305. https://doi.org/10.1007/s11192-017-2534-4

Qiu, J., Song, S., Wang, Y., Yang, H., Wang, J., Yao, S., Guo, K., Li, B., Zhou, E., Yu, J., Tang, T., Xu, N., 2016. Going Deeper with Embedded FPGA Platform for Convolutional Neural Network. pp. 26–35. https://doi.org/10.1145/2847263.2847265

Radford, A., Metz, L., Chintala, S., 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv:1511.06434 [cs].

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2015. You Only Look Once: Unified, Real-Time Object Detection. arXiv:1506.02640 [cs].

Representation theory of finite groups and associative algebras : Curtis, Charles W : Free Download, Borrow, and Streaming : Internet Archive [WWW Document], n.d. URL https://archive.org/details/representationth11curt (accessed 2.25.20).

Schroff, F., Kalenichenko, D., Philbin, J., 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 815–823. https://doi.org/10.1109/CVPR.2015.7298682

Shen, Y., Gu, J., Tang, X., Zhou, B., 2019. Interpreting the Latent Space of GANs for Semantic Face Editing. arXiv:1907.10786 [cs].

Simon, P., 2013. Too Big to Ignore: The Business Case for Big Data, 1st ed. Wiley Publishing.

Simonyan, K., Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs].

Tian, Y., Zhu, Y., 2016. Better Computer Go Player with Neural Network and Long-term Prediction. arXiv:1511.06410 [cs].

Tolstikhin, I., Bousquet, O., Gelly, S., Schoelkopf, B., 2019. Wasserstein Auto-Encoders. arXiv:1711.01558 [cs, stat].

Weinstein, M., Horn, D., 2009. Dynamic quantum clustering: A method for visual exploration of structures in data. Phys. Rev. E 80, 066117. https://doi.org/10.1103/PhysRevE.80.066117

Wu, B., Wan, A., Iandola, F., Jin, P.H., Keutzer, K., 2019. SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving. arXiv:1612.01051 [cs].

Wu, C., Herranz, L., Liu, X., Wang, Y., van de Weijer, J., Raducanu, B., 2019. Memory Replay GANs: learning to generate images from new categories without forgetting. arXiv:1809.02058 [cs].

Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J., n.d. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling 9.

Wu, Y., Donahue, J., Balduzzi, D., Simonyan, K., Lillicrap, T., 2019. LOGAN: Latent Optimisation for Generative Adversarial Networks. arXiv:1912.00953 [cs, stat].

Xie, Q., Luong, M.-T., Hovy, E., Le, Q.V., 2020. Self-training with Noisy Student improves ImageNet classification. arXiv:1911.04252 [cs, stat].

Xu, D., Tian, Y., 2015. A Comprehensive Survey of Clustering Algorithms. Ann. Data. Sci. 2, 165–193. https://doi.org/10.1007/s40745-015-0040-1

Yuan, Y., Chen, X., Wang, J., 2019. Object-Contextual Representations for Semantic Segmentation. arXiv:1909.11065 [cs].

Zeiler, M.D., Fergus, R., 2013. Visualizing and Understanding Convolutional Networks. arXiv:1311.2901 [cs].

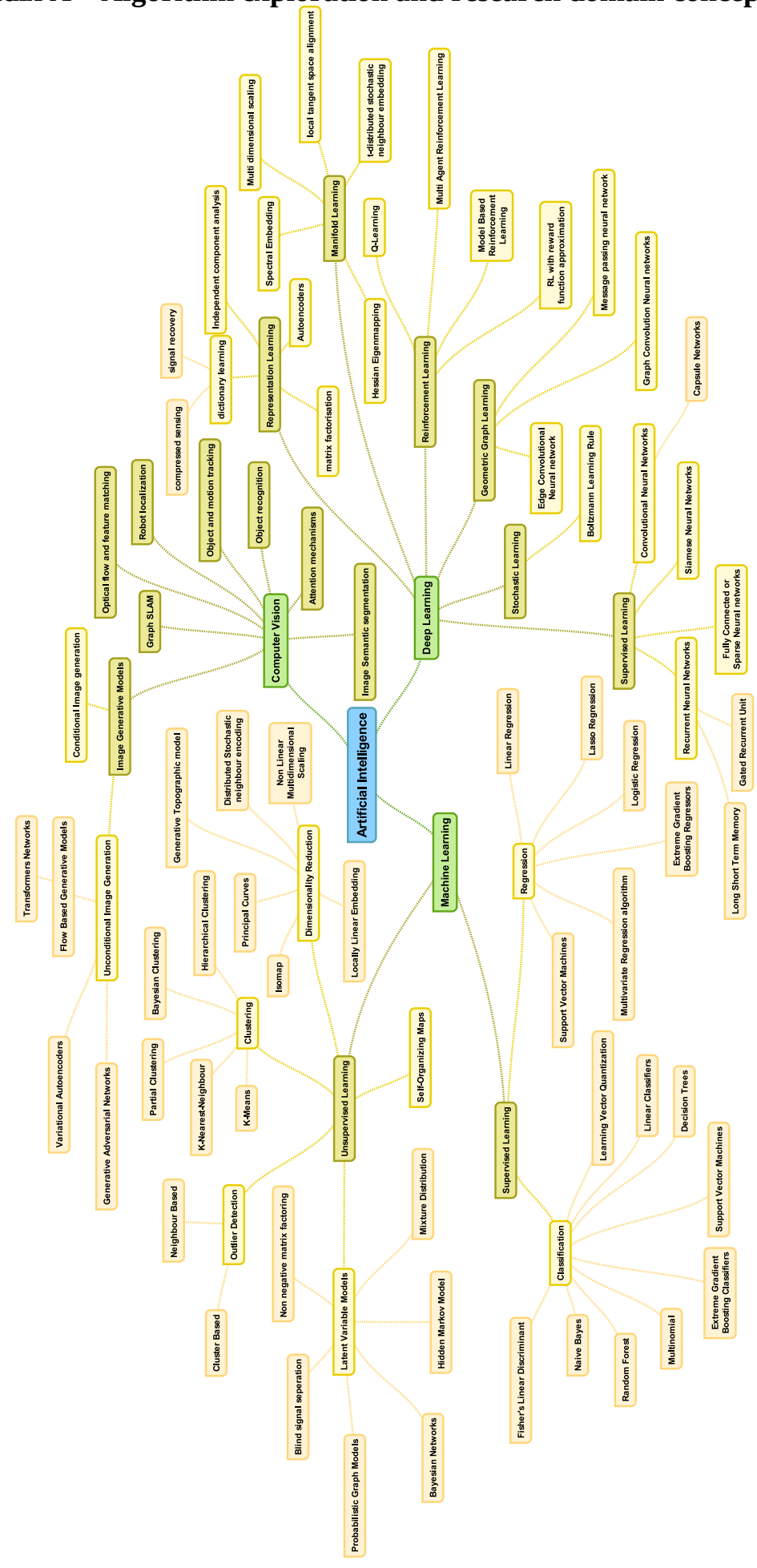
Zhang, W., Wang, X., Zhao, D., Tang, X., 2012. Graph Degree Linkage: Agglomerative Clustering on a Directed Graph. arXiv:1208.5092 [cs, stat].

Zhang, Z., Zha, H., 2002. Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment. SIAM Journal of Scientific Computing 26, 313–338.

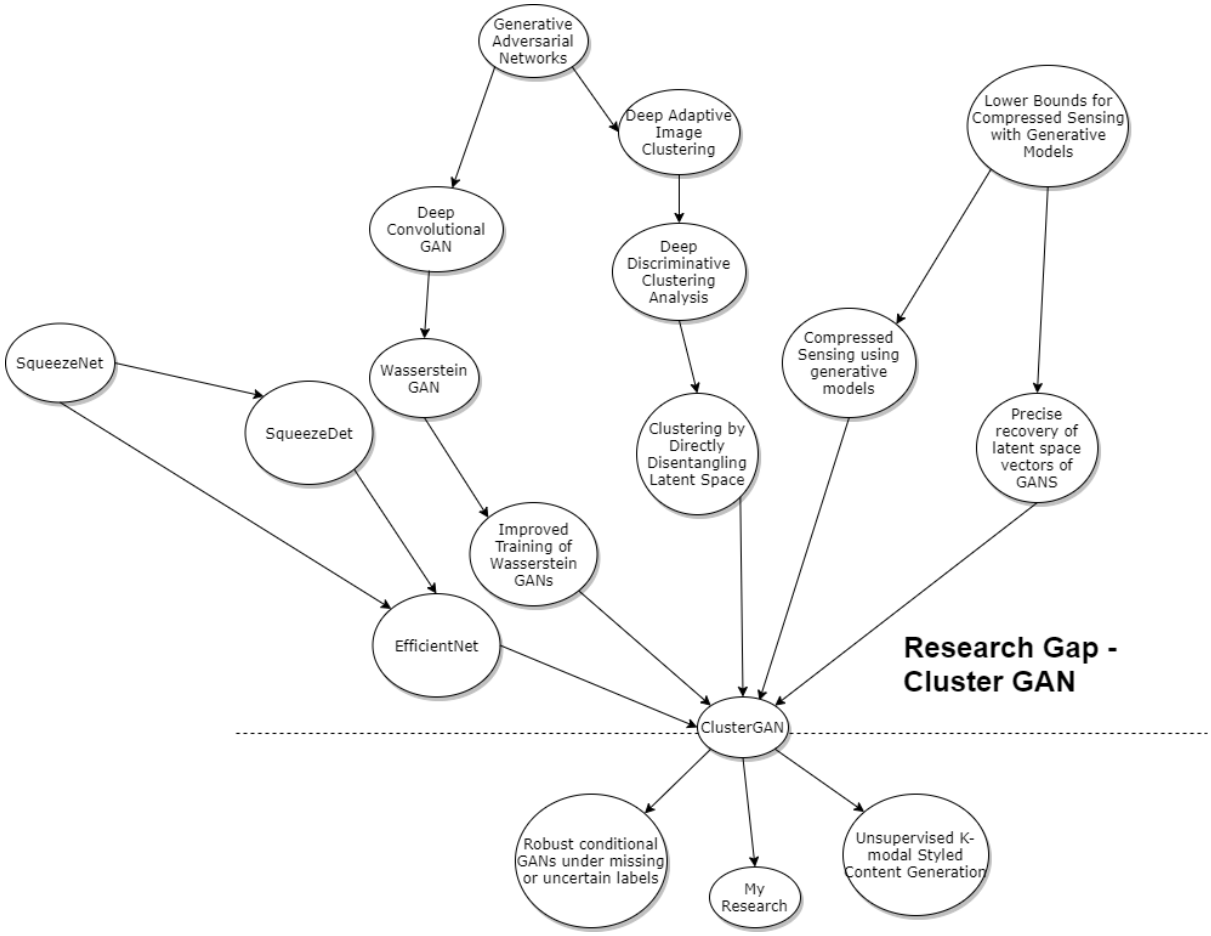
Zheng, Z., Yu, Z., Zheng, H., Wang, C.C., Wang, N., 2017. Pipeline Generative Adversarial Networks for Facial Images Generation with Multiple Attributes [WWW Document]. undefined. URL https://www.semanticscholar.org/paper/Pipeline-Generative-Adversarial-Networks-for-Facial-Zheng-Yu/83fe3f690a9a17b798f3fed86a59e10b03c5d285#references (accessed 4.9.20).

Zhu, J.-Y., Krähenbühl, P., Shechtman, E., Efros, A.A., 2016. Generative Visual Manipulation on the Natural Image Manifold. arXiv:1609.03552 [cs].

Appendix A – Algorithm exploration and research domain concept graph.



Appendix B – Research gap emergence. (seminal researches leading to this dissertation)



Appendix C - Detailed Activity Schedule

	TASK	START	END
1	**Project Initiation**	**Sep 2019**	**Oct 2019**
1.1	Research domain exploration	Sep week 1	Sep week 2
1.2	Selecting research gap	Sep week 2	Sep week 3
1.3	Exploring with research gap domain	Sep week 1	Oct week 4
1.4	Ratifying research aim and objectives	Sep week 2	Oct week 3
1.5	Drafting provisional literature survey	Sep week 1	Oct week 4
1.6	Drafting Project Initiation Document (PID)	Oct week 1	Oct week 4
1.7	PID - review mentor	Oct week 2	Oct week 4

1.8	Refinement of PID	Oct week 2	Oct week 4
1.9	Finalizing PID	Oct week 4	Oct week 4

2	**Literature Review**	**Sep 2019**	**Apr 2020**
2.1	Exploring the research domain	Oct week 1	1 Nov week 1
2.2	Exploring the product domain	Oct week 1	Nov week 2
2.3	GAN variants for image synthesis	Oct week 1	Nov week 4
2.4	Latent space optimization techniques	Oct week 3	Nov week 4
2.5	Model compression techniques	Oct week 2	Nov week 4
2.6	Mathematical methods	Nov week 1	Nov week 2
2.7	Drafting LR	Nov week 3	Nov week 4
2.8	LR – review mentor	Nov week 4	Nov week 4
2.9	Finalizing LR	Sep week 1	Apr week 2

3	**Requirement Gathering**	**Sep 2019**	**Dec 2019**
3.1	Explore requirements	Sep week 2	Dec week 2
3.2	Create Questionnaires	Sep week 3	Dec week 1
3.3	Conduct questionnaires sessions	Oct week 1	Dec week 2
3.4	Preparing questions for interview	Sep week 2	Dec week 1
3.5	Conducting interviews	Oct week 1	Dec week 2
3.6	Refining the rechecking data collection methods	Nov week 1	Dec week 2
3.7	Analyzing received data	Nov week 2	Nov week 4
3.8	Creating SRS	Nov week 1	Dec week 3
3.9	SRS – review mentors	Dec week 1	Dec week 2
3.10	Finalize SRS	Dec week 2	Dec week 2

4	**Designing of the System**	**Sep 2019**	**Jan 2020**
4.1	Identifying components of the existing facial recog systems	Sep week 2	Dec week 2
4.2	Implement framework and the scope of the prototype	Oct week 2	Dec week 3
4.3	High level architecture of the system	Dec week 3	Dec week 3
4.4	High-level architecture – review mentor	Dec week 4	Jan week 4

4.5	Drawing system design diagram	Dec week 4	Jan week 4
4.6	Drafting SDS	Dec week 4	Jan week 1
4.7	Mentor review of SDS	Jan week 2	Jan week 1

5	**Selection of Tools and Technology**	**Nov 2019**	**Jan 2020**
5.1	Reviewing different tools and technology	Nov week 2	Jan week 2
5.2	Selection of appropriate tools and technology	Jan week 1	Jan week 2
5.3	Mentor review of tools and technologies	Jan week 3	Jan week 2

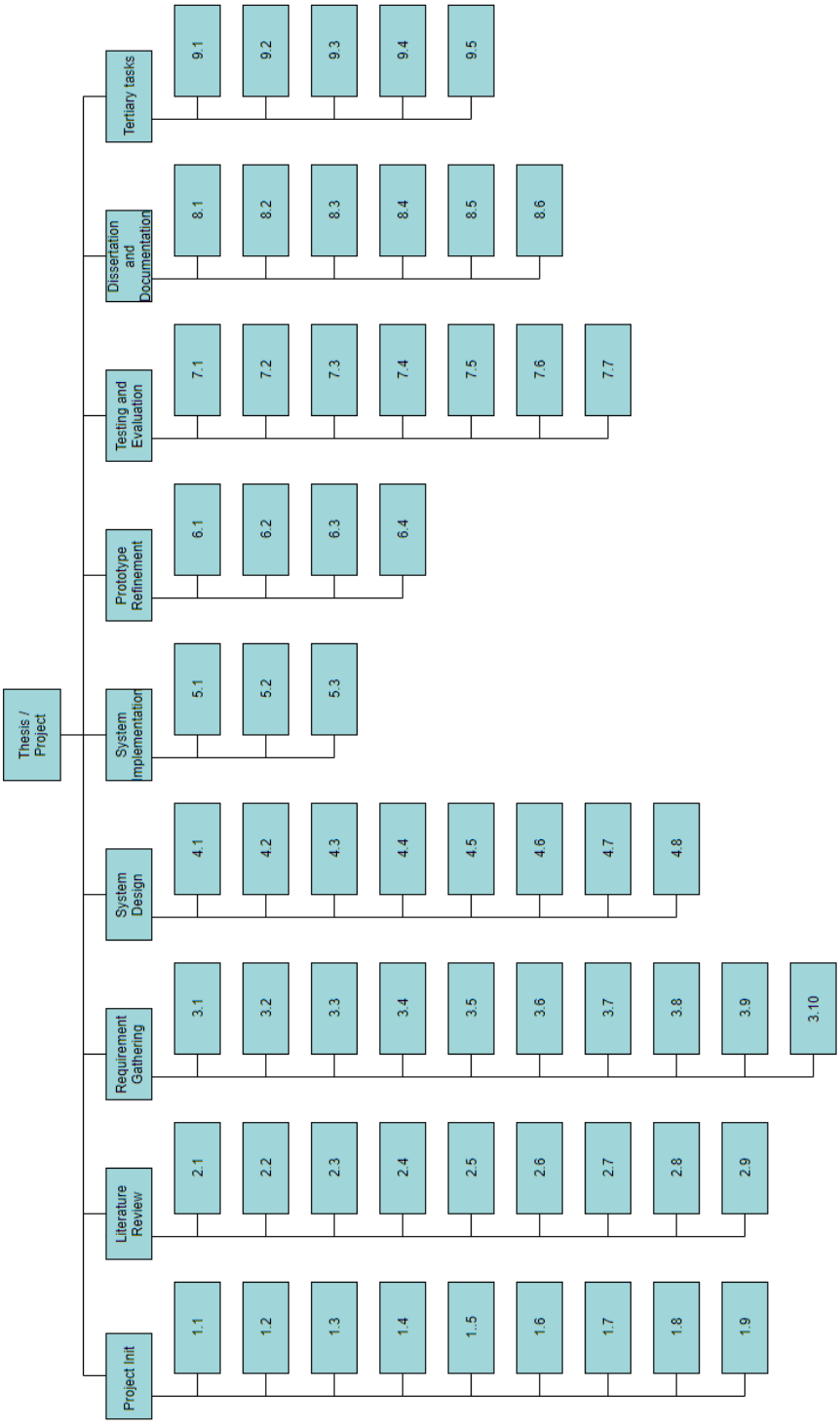
6	**Prototype Implementation**	**Dec 2019**	**Apr 2020**
6.1	Define deliverable artifacts	Dec week 3	Jan week 2
6.2	Implementing core features	Jan week 3	Mar week 2
6.3	Integrating compartments of the app	Feb week 2	Mar week 3
6.4	Ratify the prototype	Mar week 3	Apr week 2

7	**Testing and Evaluation**	**Mar 2019**	**Apr 2019**
7.1	Create test plans	Mar week 1	Apr week 2
7.2	Identifying functional and nonfunctional requirements	Mar week 2	Apr week 2
7.3	Identify Unit component	Apr week 1	Apr week 4
7.4	Writing unit tests	Apr week 1	Apr week 4
7.5	Writing integration test	Apr week 1	Apr week 4
7.6	Writing usability test	Apr week 1	Apr week 4
7.7	Final evaluation of research artifact	Apr week 3	Apr week 4

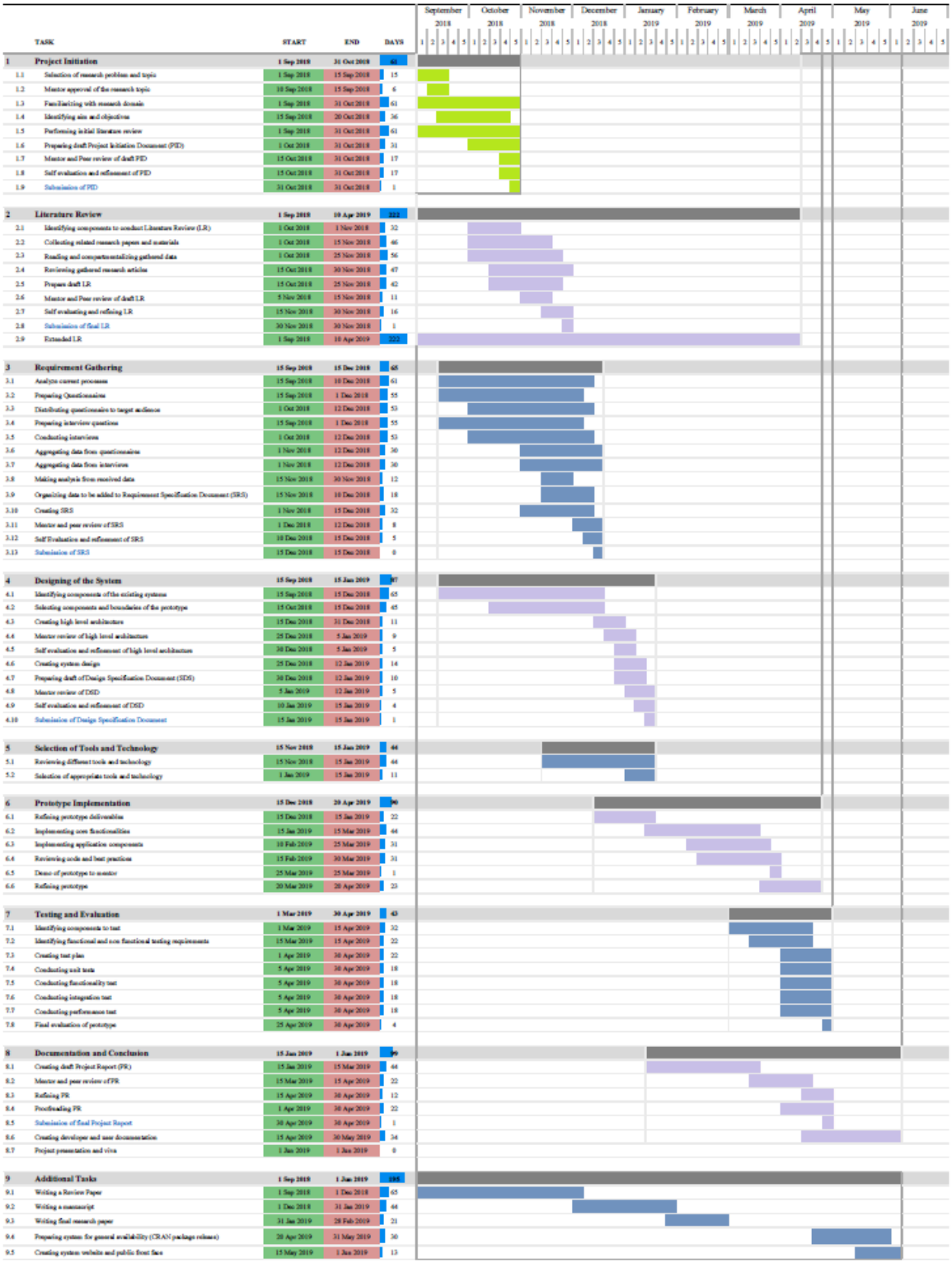
8	**Dissertation and Documentation**	**Jan 2020**	**Jun 2020**
8.1	Drafting the dissertation	Jan week 1	Mar week 2
8.2	Mentor review dissertation	Mar week 2	Apr week 2
8.3	Refining dissertation	Apr week 2	Apr week 4
8.4	Submission of final dissertation	Apr week 4	Apr week 4
8.5	Writing documentation for the product	Apr week 2	May week 4
8.6	Project presentation and viva	Jun week 1	Jun week 1

9	**Tertiary Tasks**	**Sep 2019**	**Jun 2020**
9.1	Writing a manuscript review paper	Sep week 1	Dec week 2
9.2	Refining the review paper	Dec week 1	Jan week 4
9.3	Write the dissertation	Jan week 4	Feb week 4
9.4	Develop system for public release	Apr week 3	May week 4
9.5	Designing the system	May week 2	Jun week 1

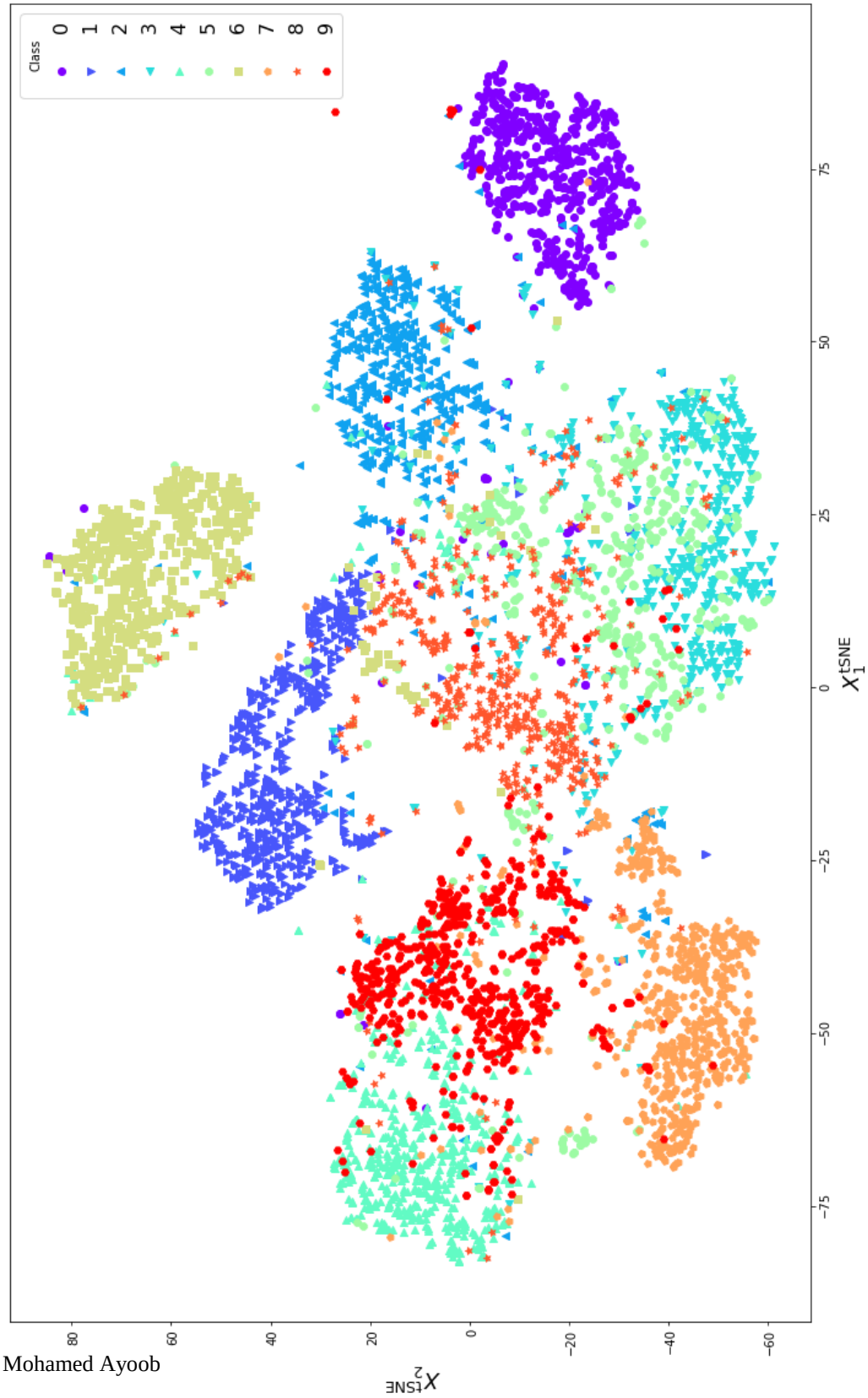
Appendix D - Work Breakdown Structure



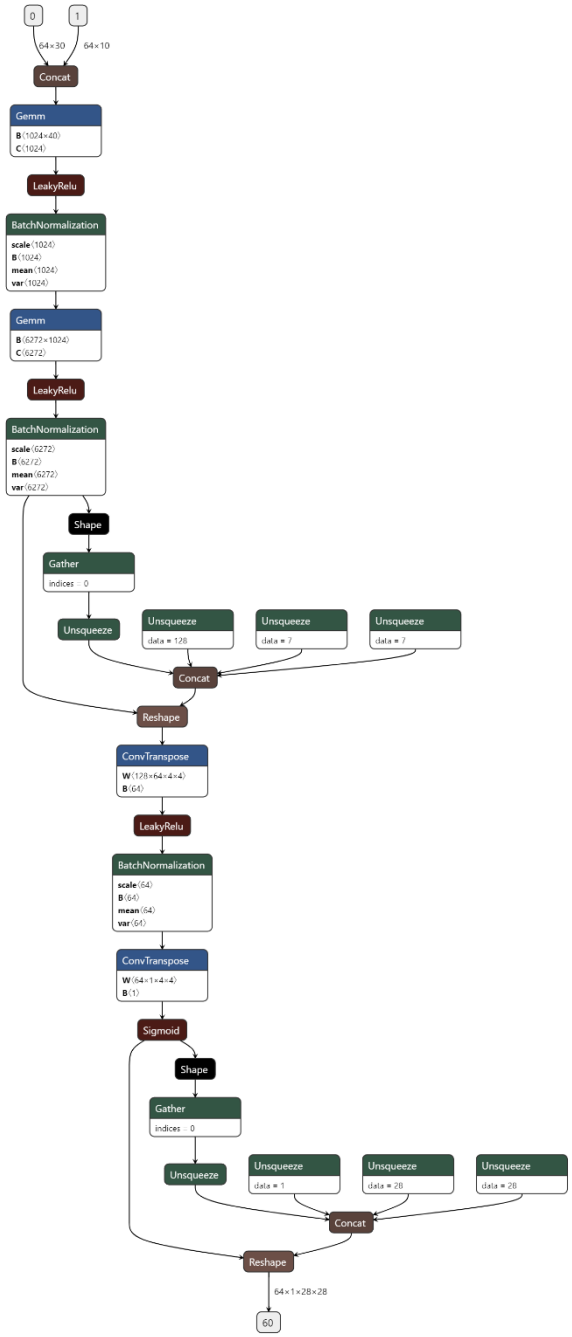
Appendix E - Gantt Chart



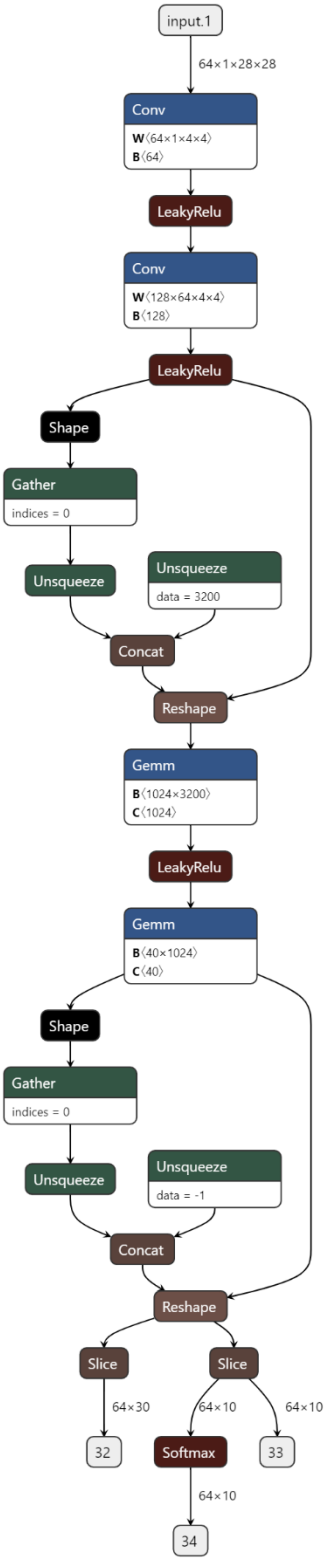
Appendix F – Manifold Learning Results



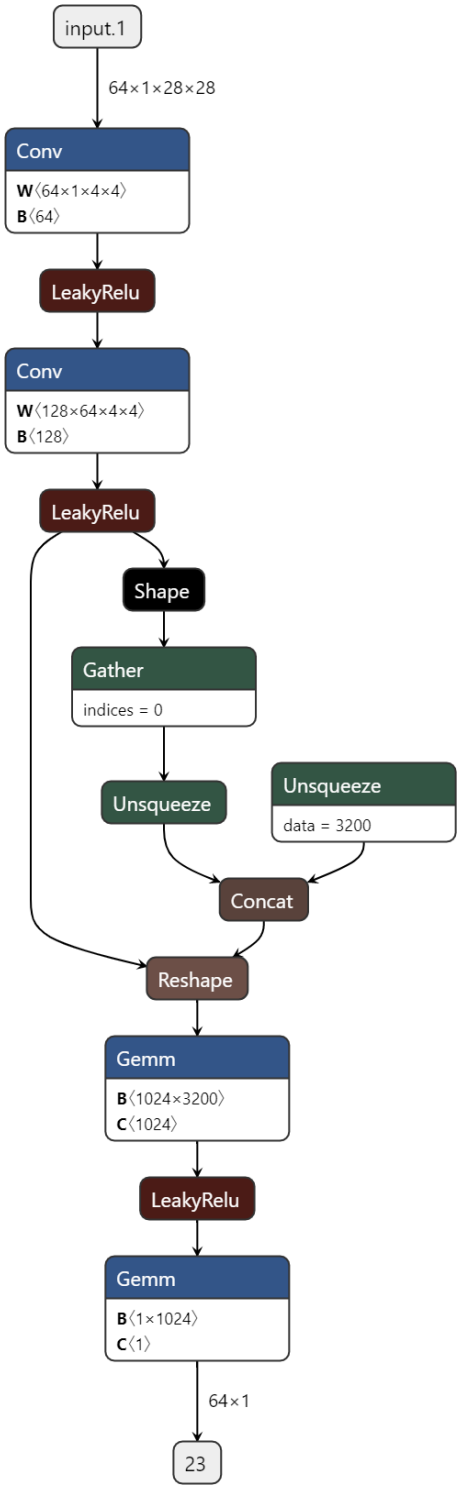
Appendix G – Architectures of the generator, discriminator and encoder developed



Generator



Encoder



Discriminator

Appendix H - Evaluation Form

Evaluation of the research - Latent space vectors of GAN

This Questionnaire is based on image generative models. In particular the latent space vectors of GAN. This is for my research project scope evaluation.

* Required

Name *

Your answer

Profession *

Your answer

Company / University *

Your answer

How would you describe your self in the field of machine learning? *

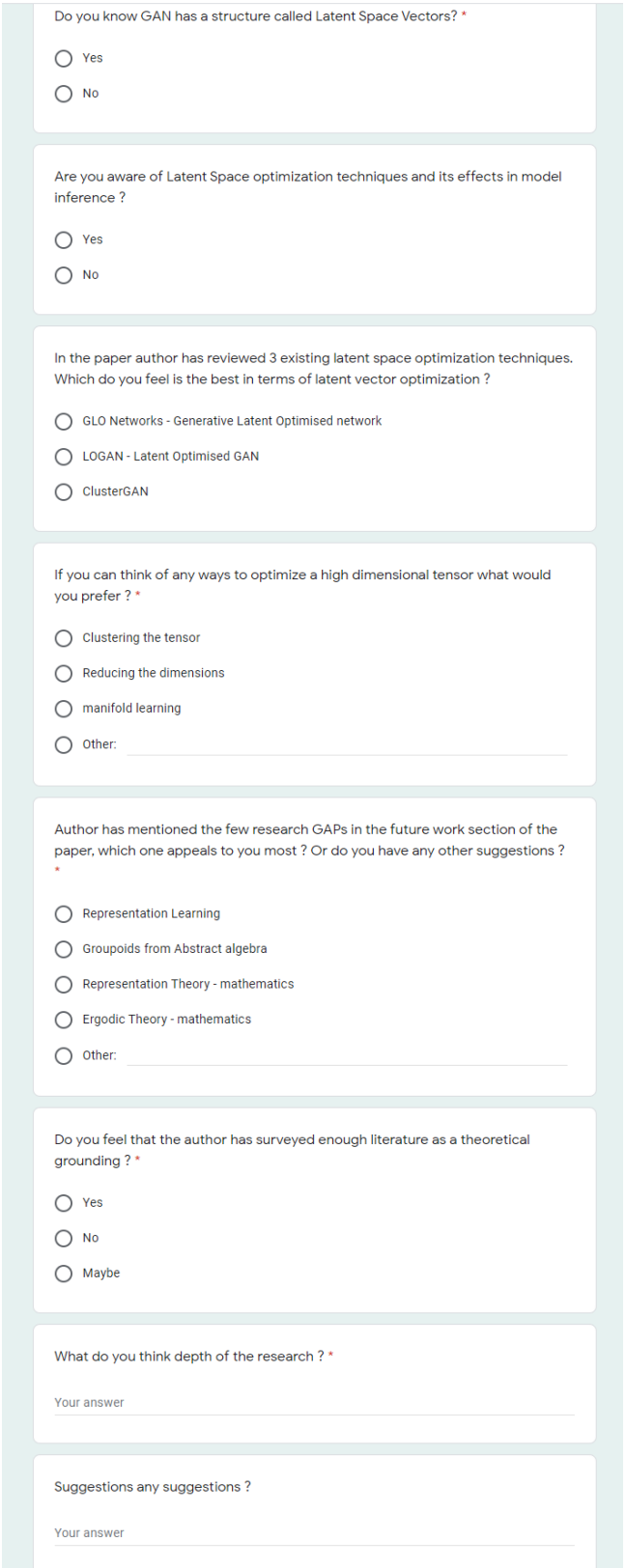
- ◯ I have not tried out machine learning before
- ◯ I'm a beginner.
- ◯ I'm experienced with machine learning. I have been developing models and pipelines for sometime now.
- ◯ I'm a expert at the field. I have read research papers and implemented them on my own.

What tools and technologies have you used in machine learning? (Just names of technologies would suffice) Ex. Python, Tensorflow, PyTorch, Apache Spark etc. *

- ☐ PyTorch
- ☐ Tensorflow
- ☐ Apache Spark MLLib
- ☐ MATLAB
- ☐ Cloud ML services (Google ML Engine, AWS SageMaker)
- ☐ Other:

Have you worked with image generative models? *

- ◯ I have not tried out machine learning before
- ◯ I'm a beginner, I have run image generative model notebooks of Github Repositories.
- ◯ I'm experienced with image generation. I follow the researches.
- ◯ I'm a expert at the field. I have written paper/s on my own.



Do you know GAN has a structure called Latent Space Vectors? *

- Yes
- No

Are you aware of Latent Space optimization techniques and its effects in model inference ?

- Yes
- No

In the paper author has reviewed 3 existing latent space optimization techniques. Which do you feel is the best in terms of latent vector optimization ?

- GLO Networks - Generative Latent Optimised network
- LOGAN - Latent Optimised GAN
- ClusterGAN

If you can think of any ways to optimize a high dimensional tensor what would you prefer ? *

- Clustering the tensor
- Reducing the dimensions
- manifold learning
- Other:

Author has mentioned the few research GAPs in the future work section of the paper, which one appeals to you most ? Or do you have any other suggestions ? *

- Representation Learning
- Groupoids from Abstract algebra
- Representation Theory - mathematics
- Ergodic Theory - mathematics
- Other:

Do you feel that the author has surveyed enough literature as a theoretical grounding ? *

- Yes
- No
- Maybe

What do you think depth of the research ? *

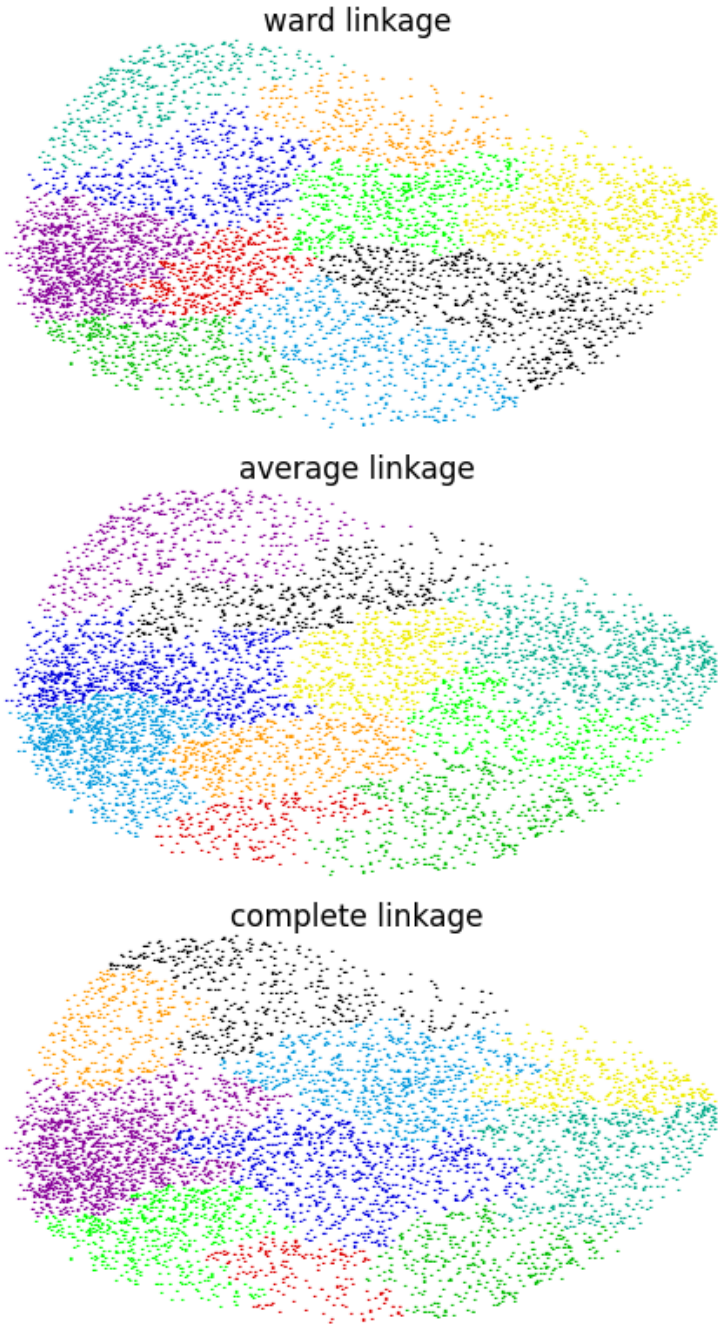
Your answer

Suggestions any suggestions ?

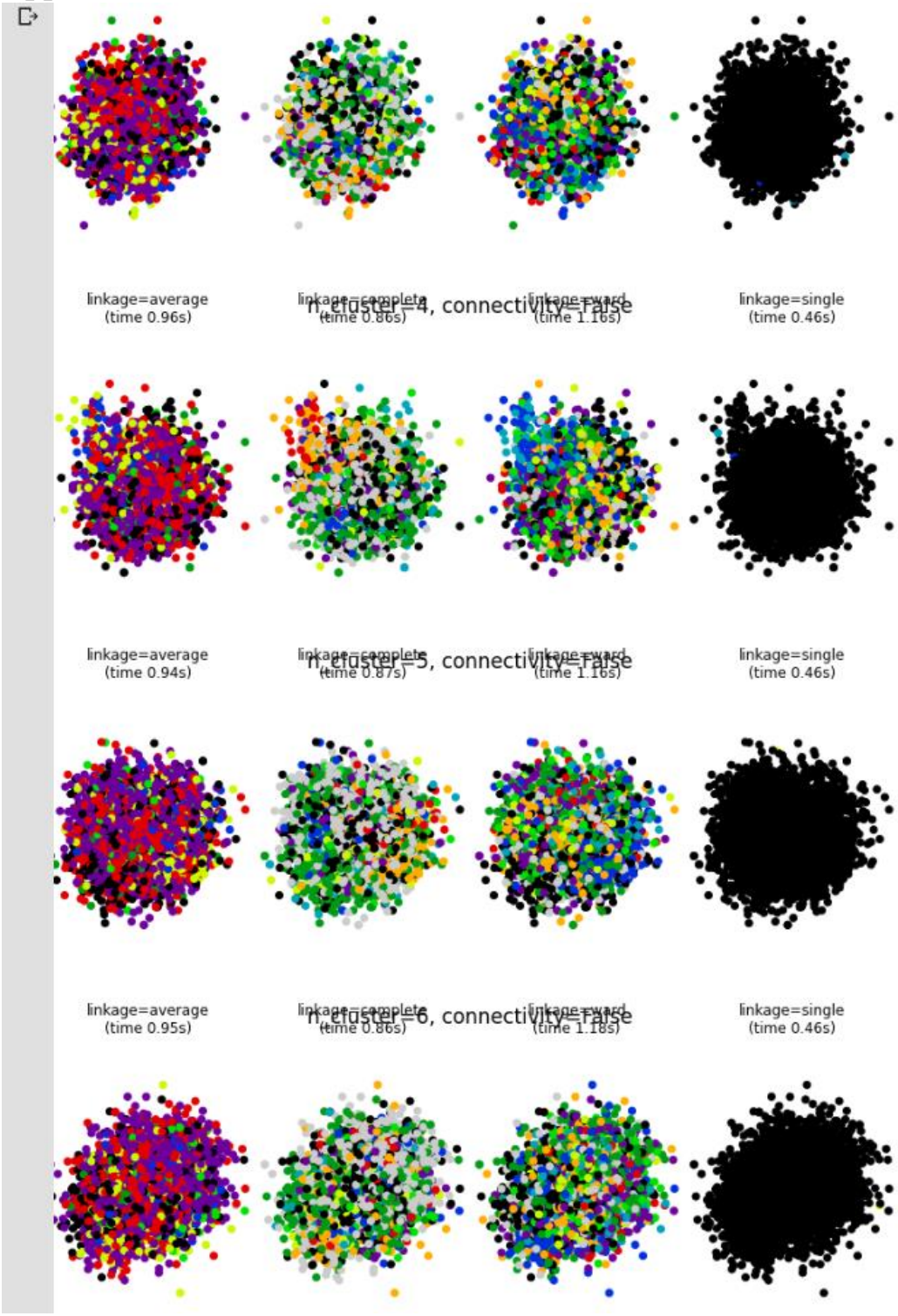
Your answer

Appendix I – Research results - I

```
Generating embedding
Dimensions for the manifold reduction : 2
ward :  1.14s
average :       0.90s
complete :      0.86s
single :        0.39s
```

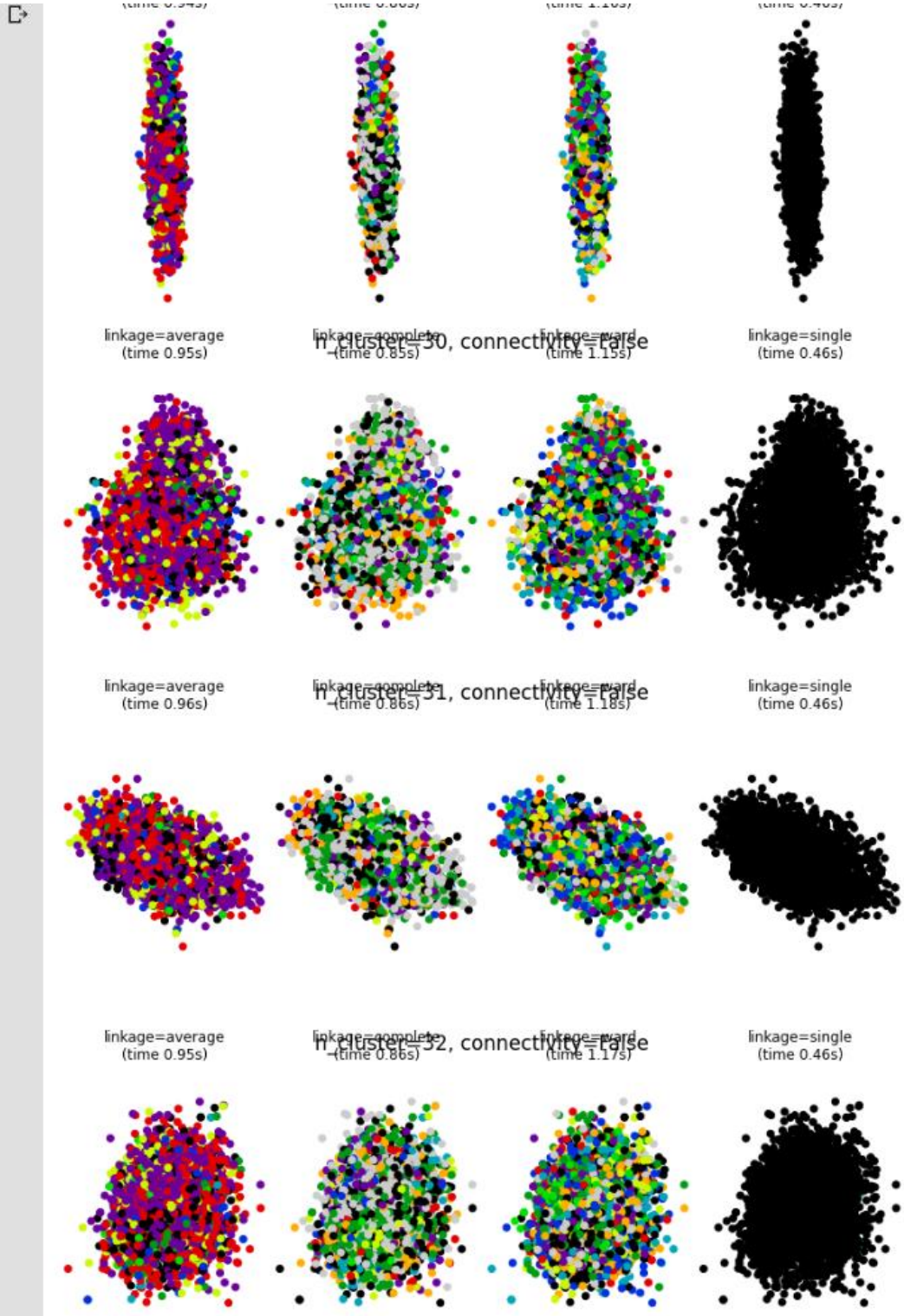


Appendix J – Research results – II



Latent space vectors that are densely populated by the dataset

Manifold Learning of Latent Space Vectors in GANs for image synthesis.



Latent space vectors that are sparsely populated and in need of more scrutiny.

xx

Mohamed Ayoob

Appendix K – Plagiarism Report