INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER, UK

University of Westminster, Coat of Arms

GANSAN: Towards Efficient Image Restoration Using Optimised Generative Adversarial Networks

A dissertation by

Mr. Ruchira Sahabandu

w1790179 | 2019724

Supervised by

Mr. Guhanathan Poravi

Submitted in partial fulfilment of the requirements for

the BSc (Hons) Computer Science degree at the University of Westminster.

May 2023

ABSTRACT

Restoration of degraded images is a naturally ill-posed problem which has been an open research problem for many years in the domain of computer vision. Whilst analogue images were primarily affected by physical degradations, digital imaging systems still struggle with visual corruptions and degradations such as blurring, image noise, low resolution, and haze. Since imaging systems are used in many other cutting-edge applications such as autonomous vehicles, surveillance systems and robotics, it is essential to develop efficient and robust image restoration systems capable of dealing with different types of visual degradations.

The proposed image restoration model is a result of experimenting with NAS to optimise a base U-Net generator, the introduction of a novel PatchGAN based discriminator with self-attention, and subsequently, the usage of automatic hyperparameter optimisation to refine the compiled GAN model through iterative training, using a reinforcement-learning like approach for fine-tuning. Additionally, improvements made by tweaking the normalisations applied to both the generator and discriminator have allowed the model to achieve better overall performance and increased generalisability across different types and levels of degradations whilst minimising visual artifacts in the output.

After training for 500 epochs on relatively light datasets with less than 2000 training image pairs each, GANSAN is capable of denoising noisy images with average output [PSNR/SSIM] scores of (30.58 dB/0.885) and is also capable of restoring low-res blurry images with average output [PSNR/SSIM] scores of (32.19 dB/0.894) as tested on popular benchmark datasets, CBSD68 and Set5 respectively.

Keywords: Image restoration, Image denoising, Image deblurring, Computer Vision, Deep Learning, Data Science

Subject Descriptors

- Computing methodologies → Artificial intelligence → Computer vision
- Computing methodologies → Machine learning → Machine learning approaches → Neural networks
- Computing methodologies → Machine learning → Machine learning approaches → Bio-inspired approaches → Genetic algorithms
- Computing methodologies → Machine learning → Machine learning approaches → Bio-inspired approaches → Generative and developmental approaches

DECLARATION

I hereby certify that this dissertation and all its related subcomponents are a result of my own work, and none of them have been or are currently being submitted/presented as contents of any other degree or qualification program to any other university or institution.
All facts and figures obtained from existing external sources have been appropriately cited and referenced.

Student Name: Ruchira Sahabandu
Registration Number: w1790179 / 2019724

Signature: ------------------------------ **Date:** 05/05/2023

ACKNOWLEDGEMENT

This thesis stands not only as an amalgamation of the dedication, and hard work of my lone self, but rather as that of the culmination of the continued positive influence and support from a number of very special people in my life.

First and foremost, I'd like to extend my utmost appreciation and thanks towards my supervisor, Mr. Guhanathan Poravi without whom this project would not have been possible. I'm forever grateful to him for his enduring guidance and relentless push for perfection throughout the entire research process which empowered me to venture beyond my comfort zone and produce a work of this calibre. I would also like to especially thank Mr. K. T. Yasas Mahima for continuously supporting and directing me in successfully navigating through all the inherent technical intricacies and challenges of this complex area of research. I must also extend a huge thanks to all the academic and industrial experts who helped me better shape the course of this research with their invaluable advice and expertise through each phase of this project. I would also like to thank all my lecturers and tutors as well as the management at the Informatics Institute of Technology for all the support, guidance and opportunities presented to me throughout my undergraduate life.

To all my closest friends, thank you all so very much for your continued love, friendship, and support during the good times and the bad – I would have not been able to come this far without you. And last but not least, to my beloved parents, thank you both from the bottom of my heart for everything you do. Thank you for being the unfailing pillar of strength that keeps me going to pursue all my dreams. I would not be here if not for the unwavering, love, kindness, and guidance that you've so selflessly shown me through every single turn of my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Acronym	Description
GAN	Generative Adversarial Network
NAS	Neural Architecture Search
AutoHPO	Automated Hyperparameter Optimisation
AutoML	Automated Machine Learning
CNN	Convolutional Neural Network
ANN	Artificial Neural Network
RNN	Recurrent Neural Network
CAE	Convolutional Autoencoder
SOTA	State of the Art
ML	Machine Learning
DL	Deep Learning
RL	Reinforcement Learning
IDE	Integrated Development Environment

1 INTRODUCTION

1.1 Chapter Overview

Marking the inception of this project document, this chapter aims to provide the reader with a general introduction to the author's research. Starting at the problem domain, it will lay the foundation by introducing the problem statement, the author's motivation for their selection, identified gap based on the initial literature review, aims and objectives of the research, and the intended contribution to the body of knowledge. It will also explore the expected research challenges and lay out the research questions which the author hopes to address through this work.

1.2 Problem Domain

1.2.1 Image Restoration

Image restoration is a well-established problem in low-level computer vision, where the aim is to recover or hallucinate a high-fidelity sharp image from a degraded (e.g., blurred, or low-resolution) input observation. This has been a long-standing area of research interest in the digital imaging space with attempted solutions ranging from simple mathematical models during the early days, to more advanced and complex deep learning (DL) approaches using various types of neural network (NN) based models and algorithms developed within the past decade. (Su, Xu and Yin, 2022)

Image restoration has been identified as a requirement in a multitude of domains such as autonomous driving and navigation, video surveillance systems, medical imaging, astronomy, and microscopy where it is common to receive degraded inputs due to environmental and other external factors which cannot be controlled. (Hu et al., 2020; Su, Xu and Yin, 2022)

1.2.2 Generative Adversarial Networks

GANs, originally introduced for image generation, is an interesting and rapidly evolving area of research within the DL space and has received wide attention across multiple problem domains, with arguably the most significant impact in the field of computer vision and image processing (Wang, She and Ward, 2022). The basis of a typical GAN architecture consists of a generator network G and a discriminator network D, where the generator attempts to generate new samples based on the training dataset whilst the discriminator attempts to estimate the probability that a sample is from the training data rather than G (Goodfellow et al., 2014).

In the context of image restoration, the application of adversarial training via GAN models has proven to achieve better visual performance and efficiency with tasks such as image deblurring and super-resolution (Fu et al., 2020; Zhang et al., 2022). However, similar to other DL based models, traditional GANs make use of manually designed network architectures created through trial-and-error. The difficulty of creating optimal network architectures utilized in GAN models is pronounced further due to the possibility of encountering unique issues such as mode-collapse during training (Goodfellow, 2017).

1.2.3 Neural Architecture Search

NAS, which is a subfield of AutoML, focuses on automating the process of architecture engineering for NNs and is seen as a logical next-step of automating machine learning (Elsken, Metzen and Hutter, 2019). The basis of NAS could be divided into 3 main components: search space, search strategy and performance estimation/evaluation strategy and could be employed to optimize a multitude of DL models based on NNs.

NAS has been successfully utilized in designing efficient image restoration models based on comparatively simple approaches such as Convolutional Autoencoders (CAE), and Convolutional Neural Networks (CNN) /U-Net models (Suganuma, Ozay and Okatani, 2018; Chen et al., 2020; Arican et al., 2022). Furthermore, NAS has also been successfully applied towards the automation of GAN architecture search for computer vision tasks such as conditional and unconditional image generation and image segmentation (Gong et al., 2019; Gao et al., 2020; Ganepola and Wirasingha, 2021a).

1.3 Problem Definition

The core aim of image restoration is to recover clean, latent images from degraded inputs (e.g., blurred/low resolution images, images with high noise) via methods such as deblurring, super-resolution and denoising. Due to the infinite possible mappings between a degraded observation and its corresponding restored image, image restoration is defined as an ill-posed inverse problem and is still considered to be challenging to tackle (Su, Xu and Yin, 2022).

Conventional approaches towards solving this problem have been based on probabilistic models that solve inverse problems. However, within the past decade, rapid developments in the DL space as well as its various applications in the domain of computer vision has given rise to the introduction of various image restoration models based on Neural Networks (NNs). More recently, the popularity of GAN models in the field of image generation has led to their adoption towards image restoration tasks where a notable increase in performance over existing generic NN based methods has been observed, raising the bar further for State of The Art (SOTA) performance. (Su, Xu and Yin, 2022)

Whilst much progress has been observed in other computer vision tasks employing such GAN models, by optimizing the underlying network architectures using approaches such as NAS, there is a severe lack of such efforts that have been made towards improving GAN models used for image restoration purposes (Gong et al., 2019; Gao et al., 2020; Ganepola and Wirasingha, 2021a).

1.3.1 Problem Statement

Image restoration, being a well-researched area in classic low-level computer vision consist of problems ill-posed by nature, giving rise to the development of a multitude of image restoration networks, many of which employ adversarial training via GAN models which are built upon NN architectures that are difficult to design manually, to be as optimised and efficient as possible.

1.4 Research Motivation

Given the widespread adoption of AI and autonomous technologies that rely on computer vision systems, such as autonomous vehicles, computational photography, and autonomous surveillance systems, it is essential to improve their integrated image restoration methods used to boost performance, and the approaches followed to design and develop them. The process of designing these systems involves developing generative models that can efficiently remove noise, blur, and other distortions while preserving important details. As these technologies become more advanced, the need for sophisticated image restoration techniques will only increase, making this an essential area of research for the future of computer vision.

1.5 Research Gap

- Grounded on the existing work, it is found that adversarial training (via GANs) helps to boost results of image restoration processes (Kupyn et al., 2018) compared to other lightweight deep learning approaches.
- Whilst it has been found that deeper networks in GAN models increase performance further, training and testing times also rise exponentially. Additionally, The appearance of high-frequency artifacts makes training certain GAN variants (e.g. SRGAN) with deeper networks increasingly complex (Ledig et al., 2017).

- Similarly, the experimentation of using NAS to create image restoration networks and deep image priors have proved to be successful and capable of surpassing the performance of most human-designed image restoration networks (Suganuma, Ozay and Okatani, 2018; Zhang et al., 2021; Arican et al., 2022).
- However, a combined approach of NAS with GANs towards image restoration problems, is still an open research problem (Ganepola and Wirasingha, 2021b). A successful completion of such study could aid in improving the efficiency and accuracy of image restoration tasks through GANs as proved by the success of NAS optimised GAN architectures in image synthesis and segmentations tasks (Gong et al., 2019; Ganepola and Wirasingha, 2021b).

1.6 Contribution to the Body of Knowledge

1.6.1 Contribution to the Problem Domain

A majority of State of the Art (SOTA) models used for image restoration tasks such as image deblurring and image super-resolution employ some form of GAN based architecture to increase their accuracy and the quality of the output. The neural networks used for such GAN models are designed, architected, and optimized manually, which is quite a time consuming and difficult process.

The introduction of novel approaches such as NAS as a means of architecting such networks would help catalyse the development of more optimized neural networks and GAN models for image restoration tasks and would allow future researchers to develop more approaches to tackle such low-level vision problems with greater accuracy and efficiency.

1.6.2 Contribution to the Research Domain

This project aims to explore a means of harnessing the versatility of NAS as an approach to architecting efficient and optimized neural networks, towards the purpose of designing optimized GAN models for image restoration tasks.

Existing research has explored the usage of GANs (Xu et al., 2017; Gilbert, Messingher and Patel, 2018; Kupyn et al., 2018) and NAS (Chen et al., 2020; Arican et al., 2022) separately for various image restoration tasks, as well as using NAS to architect/optimise GANs for other areas of image processing such as image generation and semantic segmentation. However, no such attempt has been made to optimise GAN models used for image restoration problems.

Therefore, the construction of such an optimised and efficient GAN model for image restoration tasks would be the primary technological contribution of this research.
Furthermore, extending NAS and GANs towards image restoration and other image-to-image translation tasks (e.g.: Image colour-transfer, Image style-transfer) could be defined as another potential contribution of this project.

1.7 Research Challenge

This research aims at contributing towards the improvement of GANs through the integration of novel means such as NAS, for image restoration tasks such as image denoising and sr-deblurring which are notorious for being ill-posed and challenging by nature (Xu et al., 2017). As such, the following areas have been identified to pose key challenges for this research.
Image restoration via deep learning networks – The continuous research and contributions made towards image restoration tasks such as image deblurring and image super-resolution has led to the creation of a multitude of different types of image restoration networks with varying levels of complexities and vastly different approaches to training and evaluation. Given the rapidly evolving nature of this domain, successfully understanding, and identifying the most relevant and efficient approaches would be a challenge.

Generative Adversarial Networks – GANs offer a novel approach to training generative networks through the use of an adversarial process (Goodfellow et al., 2014), where the generator *G* competes against a discriminator *D* to generate results indistinguishable from the ground truth. Since both networks are trained simultaneously to convergence, the process is resource intensive and makes for a challenging training process with possibilities of premature failures such as mode-collapse (Goodfellow et al., 2014; Goodfellow, 2017).

Neural Architecture Search – NAS focuses on automating the architecting process for neural networks (Elsken, Metzen and Hutter, 2019), with the aim of efficiently producing optimal network architectures. Defining the optimal search space and search algorithm for a selected image restoration network would pose its own challenges due to the aforementioned complex nature of SOTA networks.
For the success of this research, the author would have to deal with the above challenges as part of a cohesive system for image restoration as well as the uncertainties and complications

that would arise thereof. Further, the author would also have to tackle diverse and challenging learning areas such as math and optimizations throughout the project's duration.

1.8 Research Questions

RQ1: What properties of image restoration could be improved by using GANs?

RQ2: How could an optimized GAN model be built for general image restoration tasks?

RQ3: How could NAS be used with GAN based approaches towards image restoration?

RQ4: What are the potential ways to evaluate image restoration tasks based on GAN?

1.9 Aims and Objectives

1.9.1 Aim

This research project aims to design, develop, and evaluate an efficient and optimized image restoration Generative Adversarial Network architecture by experimenting with Neural Architecture Search, to efficiently restore degraded images (E.g.: Deblurring, Super-resolving, Denoising).

To elaborate further, the focus of this project will centre on developing an efficient deep learning approach towards restoring degraded images by experimenting with NAS and other autoML approaches to help architect an optimised generator network for a selected GAN model. Ideally the conceptualised approach would be capable of rivalling such existing human-designed network architectures, some of which employ adversarial training processes with complex loss functions and high resource requirements for training.

1.9.2 Research Objectives

The following table represents a list of atomic activities that are needed to be carried out to achieve the aim of this research.

Research Objectives	Explanation	Learning Outcome	Research Question
Problem Identification	**RO1 –** Brainstorm within domains of interest, with the support of existing literature to identify research areas with existing problems of interest. **RO2 –** Continue further reading into the selected problem domain to identify problems addressable via a research project.	LO1 LO4 LO8	RQ1

	RO3 – Review all available options and decide on a problem with good research impact		
Literature Review	A thorough literature survey is conducted within the image processing domain to, **RO1 –** Gain insight into the applications of deep learning networks within the image-processing domain. **RO2 –** Identify the deep learning approaches used towards restoring degraded and blurred images. **RO3 –** Learn about the prominence for the increased usage of GAN based models when approaching image restoration tasks. **RO4 –** Understand the usage and impact of NAS in designing neural networks, its applications towards developing optimized image restoration networks. **RO5 –** Analyse and understand the means of designing optimised GAN models using techniques such as NAS.	LO1 LO4 LO5 LO8	RQ1 RQ2 RQ3 RQ4
Data Gathering and Requirement Analysis	Relevant data is gathered, and a thorough requirement analysis is conducted to, **RO1 –** Get an understanding on existing systems and identify the functional requirements of the system being developed. **RO2 –** Gain insights on the requirements and opinions of domain experts and researchers on the expectations for the proposed system via questionnaires and interviews. **RO3 –** Understand and define the core features and functionality expected of the prototype software by users through surveys.	LO1 LO2 LO3 LO6 LO8	RQ3 RQ4
Design	Designing of the proposed image restoration approach by,	LO1 LO2	RQ1 RQ2

	RO1 – Designing and optimising a neural network for image restoration using optimisation approaches such as NAS. **RO2 –** Designing an optimized image restoration GAN model by integrating such efficient and versatile network architectures.	LO5 LO8	RQ3
Implementation	Development of the prototype using the proposed approach towards image restoration based on the architectures and approaches defined in the design phase to, **RO1 –** Effectively restore degraded images using the optimized models. **RO2** – Build efficient and functional GAN models for such image restoration tasks.	LO1 LO2 LO5 LO7	RQ2 RQ3
Testing and Evaluation	Testing and evaluation of the developed prototype with feedback and input of domain experts to, **RO1** – Test and verify the functionality of each component of the developed prototype. **RO2** – Review the performance of the system when it comes to restoring various types of images. **RO3** – Validate the effectiveness of the developed system in relation to the initially identified system requirements.	LO1 LO4 LO8	RQ4

Table 1-1: Research Objectives

1.10 Chapter Summary

Starting with an introduction to the problem domain, this chapter provided a detailed overview of the project, along with a clear definition of the problem statement, aims and objectives of the research, planned contributions as well as the expected challenges over the course of its lifecycle. The upcoming chapters will go over the requirement specifications of the proposed system, conceptualized design, and initial implementation details.

2 LITERATURE REVIEW

2.1 Chapter Overview

This chapter provides a comprehensive review on the problem domain of image restoration, current state-of-the-art techniques, relevant tools, technologies, and approaches, and evaluation methods used to assess the performance of image restoration systems. Here, we will also analyse deep learning-based approaches such as GANs and existing integrations of such systems with NAS methods in other computer vision tasks. Finally, the evaluation methods section will discuss metrics used to measure the quality of restored images, which serve as a foundation for the proposed methodology presented and evaluated in subsequent chapters.

2.2 Concept Map

A "concept map" [see **APPENDIX – I**] was created as a graphical representation of the entire project scope, including existing works, technologies, evaluation metrics, optimization methods, datasets, and key applications within the domain of computer vision and a few others.

2.3 Problem Domain

2.3.1 Introduction to Image Restoration

Image restoration is a blanket term that refers to the process of recovering clean, high-fidelity images from damaged or degraded input observations. This has been an area of active research in computer vision for the past few decades. Due to the theoretical existence of an infinite number of mappings between any given degraded input and its high-fidelity counterpart, image restoration is regarded as an ill-posed inverse problem of high complexity. Since the creation of generalized mappings applicable to a wide range of images is challenging, researchers have focused on developing learning schemes and analytical models to derive approximations of exact mappings between degraded and hi-fi images. Whilst early solutions in this space employed advanced math-based approaches, deep-learning based approaches have seen a rise in popularity in recent years due to the progress made in model architectures and learning techniques (Su, Xu and Yin, 2022).

Discussed below are some of the most encountered image restoration tasks.

2.3.1.1 Image Deblurring

Image deblurring, which happens to be a classic low-level computer vision task, describes the act of recovering a sharp image from a blurred input. Blur is commonly caused due to the absence of camera focus, due to camera shake or relative motion between the capture device and the target. These could primarily be divided into two categories, namely defocus blur and motion blur (Zhang et al., 2022).

Traditional approaches to image deblurring were not based on deep-learning centred approaches. Instead, the task was formulated like an inverse-filtering problem. Such approaches model the blurred input as a result of convolutions with spatially variant or invariant blur kernels. Various regularisation constraints are applied during this process since deblurring is by extension, an ill-posed problem (Zhang et al., 2022).

Modern deep-learning based approaches to image deblurring employ the power and versatility of neural networks and novel training methods to achieve state-of-the-art results in single-image as well as video deblurring. However, due to the continuous improvements and changes made to deblurring approaches and related network architectures, obtaining a clear overview of this domain is a challenging task (Zhang et al., 2022).

2.3.1.2 Image Super-resolution

The process of retrieving high-resolution output images from low-resolution inputs is known as image super-resolution. This happens to be one of the most important, and therefore, one of the most researched areas in image processing and computer vision, partly thanks to its application in a wide range of auxiliary domains such as medical imaging, security and surveillance and autonomous vehicles. Furthermore, advancements made in image super-resolution contributes towards the improvement of other areas of computer vision such as image classification and segmentation. Similar to the other image restoration methods discussed, image super-resolution is an ill-posed problem due to the theoretical existence of an infinite number of potential high-res images corresponding to a given low-res input (Wang, Chen and Hoi, 2021).

Image super-resolution could primarily be divided into two classes, single-image super-resolution (SISR) and multi-frame super-resolution (MFSR). The prior which refers to the act of generating a high-res image from a single low-res input has proven to be of greater practical value compared to the latter which consists of fusing information of a series of complementary, correlated images from a single scene (Chen et al., 2022).

Additionally, image super-resolution could be sub-divided based on the state of knowledge on the image degradation. Where the degradation causing the resolution drop is known, solutions are collectively known as non-blind super-resolution whilst approaches used to remedy unknown degradations are known as blind super-resolution. Most real-world images suffer from unknown and non-uniform resolution degradations making blind SR more useful and versatile (Liu et al., 2021).

2.3.1.3 Image Denoising

Due to certain physical limitations of capturing devices, during acquisition (capturing), images have a tendency to manifest "image noise" which could be defined as a form of basic signal distortion. Image noise hinders the observability of images and affects auxiliary processes such as information extraction and analysis due to loss of detail and visual definition. Therefore, the suppression of all forms of image noise, also known as "denoising", is a quintessential part of image processing and computer vision (Goyal et al., 2020).

Due to the nature of image noise, denoising poses a few inherent challenges which should be considered when formulating generalized solutions. Noise, similar to edge and texture details, manifests itself as a high-frequency visual component. Therefore the process of, distinguishing and eradicating noise has a tendency to remove such important details from the original image (Fan et al., 2019). Additionally, similar to the other image restoration problems, from a mathematical perspective, image denoising presents itself as an inverse problem where the solution to a given input is not unique. It is an ill posed problem in computer vision which has been under active research throughout the past few decades (Fan et al., 2019).

Additive White Gaussian Noise (AWGN), quantisation noise, poisson noise, and salt and pepper noise are some of the most commonly discussed image noise types. Whilst early denoising solutions employed specific image filters, the rise in popularity of CNNs has caused a shift in attention towards employing deep learning networks for denoising to solve most of the drawbacks of the early methods. These DL based methods have proven to be a lot more flexible and generalizable, catering to a wide range of noise types and image domains (Ilesanmi and Ilesanmi, 2021).

2.3.1.4 Other Image restoration methods

Whilst the above discussed image restoration methods address degradations caused primarily due to limitations of the capturing device, some restoration methods focus on improving the perceptual quality of images suffering from scene degradations and post-capture damages. Image deraining, dehazing and inpainting are few such examples.

Deraining is the process of removing rain-related visual degradations such as rain drops, streaks or accumulation from images captured under inclement weather conditions. Due to its ability to boost the performance of peripheral high-level vision tasks such as image recognition, object detection and saliency detection, deraining proves to be particularly useful for outdoor vision systems such as ones found on autonomous vehicles, security and surveillance systems and military applications. Rain information (and subsequent degradations) are complex and irregular in nature, which frames deraining as a rather complex, ill-posed computer vision problem. This complexity is heightened further when performing single image deraining instead of video deraining, since the system is forced to rely on spatial and visual properties gathered from neighbouring pixels and general rain and background-scene data (Zhang et al., 2023).

Dehazing deals with improving the perceptual quality of images suffering from low visibility caused due to haze, fog, smoke, or mist in the air during time of capture. Like rain-streaks, haze primarily affects vision systems operating in outdoor environments (Singh and Kumar, 2019). Haze and other similar atmospheric degradations cause blurring of textures and edges in images which is evident when comparing their corresponding pixel histograms. This adversely affects the applicability and performance of other computer vision tasks such as image segmentation and object detection (Gui, Cong, et al., 2022). Therefore, dehazing systems are utilized extensively to supplement both civil and military vision applications such as self-driving vehicles and visual targeting systems (Singh and Kumar, 2019).

"Inpainting" is an umbrella term that refers to the process of restoring and repairing deteriorated images with damaged, missing or altered pixels, to achieve a high visual quality and close semantic approximation to their original versions (Elharrouss et al., 2020; Qin et al., 2021). Due to its diverse applicability towards various computer vision tasks, inpainting has been an area under active research throughout the past couple of decades. Whilst early approaches to inpainting used mathematical models and statistical probability based ML models, advances made in the deep learning space has resulted in its rapid adoption and subsequent success in this domain. As such, existing inpainting techniques could be divided into 3 main subcategories; sequence-based methods, CNN-based, methods and GAN based methods (Elharrouss et al., 2020). Modern CNN and GAN based inpainting models have proven to be a great deal more versatile compared to traditional methods. This has allowed for the development of inpainting methods capable of removing distortions, masks, and other visual contaminants such as text from images. Inpainting is regarded as a countermeasure against image forgery, commonly encountered in the field of security and forensics (Elharrouss et al., 2020).

2.4 Existing Work

2.4.1 Deep Learning based Image Restoration techniques

Over the past decade, the progress made in Deep Learning techniques has had a sizeable impact on many computer vision tasks including Image Restoration. Amongst the plethora of different approaches that have been found within the deep learning space, towards solving digital image restoration problems, learning restoration kernels or image priors via deep learning neural networks has been identified as a popular method (Su, Xu and Yin, 2022).

A work by (Ulyanov, Vedaldi and Lempitsky, 2020) strikes as particularly interesting in this area due to their rather unorthodox approach towards crafting deep image priors using neural networks. The core of this work lies in its functioning proof that a neural network which is randomly initialized is capable of producing excellent results when used as a handcrafted prior, for standard inverse problems such as inpainting, super-resolution and denoising. Contrary to popular belief which follows the school of thought that learning on large datasets alone is sufficient to produce greater performance, this work demonstrates that, independent of learning, a large amount of image statistics could be captured by the structure of a given convolutional image generator. This has been achieved via a generator network fitted to a single degraded image. This approach shows us that all the information required to solve such a problem is contained within the reconstruction network's handcrafted structure, and the degraded input image itself. However, it must be noted that despite the ability of this rather simple formulation to compete with SOTA denoising, inpainting and super-resolution approaches, it is quite slow in its current state, requiring up to several GPU minutes of computation per degraded image.

Taking on a more novel, cutting edge approach, (Zamir et al., 2022) has proposed *Restormer* which has been defined as an "efficient transformer for high-res image restoration". The usage of a transformer model here, allows to overcome shortcomings of traditional CNNs, such as their inadaptability to varying inputs as well as their limited receptive field. To avoid the performance overhead due to the quadratic growth of computational complexity with spatial resolution of images, an efficient transformer model is built with multiple key changes to allow the efficient capturing of long-range interactions of pixels whilst remaining scalable for use with larger images. This system is built upon an encoder-decoder transformer, along with two additional modules to aggregate local and global pixel interactions and perform controlled feature transformations. The demonstrated versatility and the system's capability to achieve

SOTA results in many image restoration tasks such as defocus and motion deblurring, image denoising, as well as image deraining is noteworthy.

2.4.2 Usage of GANs for Image Restoration tasks

(Pan et al., 2020) proposed an approach to exploiting image priors captured via GANs which are trained on large-scale datasets of natural images. The proposed deep generative prior has proven itself capable of restoring certain missing semantics of degraded images, such as colour, resolution, and patch. In contrast to GAN-inversion methods in existence, the generator in this system has provisions for progressive and regularized on-the-fly fine-tuning using the feature-distance retrieved through the discriminator. This allows for the execution of highly flexible image manipulations and restorations including category transfers, image morphing as well as random jittering. Most importantly, it must be noted that the improvements made in this system lead to more faithful and precise reconstructions of real images.

When looking at image restoration systems that focus on a single restoration task such as deblurring, or de-noising, the work by (Kupyn et al., 2018) stands out due to its approach of utilizing GANs for image deblurring as well as its marked improvement in performance over the nearest competitor at the time of publishing. The system dubbed DeblurGAN, focuses on motion deblurring using an end-to-end, content loss and conditional GAN based learned method. The proposed system manages to achieve SOTA results in both visual appearance as well as structural similarity measures. Further, the authors have turned to a novel way of evaluating the quality of deblurring by performing object detection on the deblurred images, which ensures that the visual structure and composition of each individual visual element is preserved. DeblurGAN has proven itself to be 5 times faster than its closes competitor, DeepDeblur, which is undoubtedly a significant achievement within the image deblurring space.

SRGAN, a work by (Ledig et al., 2017) has proven to be one of the most significant developments in the image super-resolution space using GANs. Built using ResNet, a deep residual network with skip connections, SRGAN was the first framework with the capability to infer photorealistic natural images with an upscaling factor of up to 4x. The authors employed a perceptual loss function defined using the VGG network's high level feature maps, consisting of an adversarial as well as content loss. Tests using Mean-Opinion-Score (MOS) have shown significant perceptual quality gains in the outputs obtained through SRGAN, with values closer to those of the original high-res images than any other SOTA model at the time of publishing. Whilst the authors have found that deeper networks increase performance further, training and testing times also rise. The appearance of high-frequency artifacts makes training SRGAN variants with deeper networks increasingly complex.

A work by (Zhang, Sindagi and Patel, 2019) proposes a single-image deraining system using GANs to restore weather-based visual degradations (rain streaks). The authors have approached this problem by focusing on obtaining a restored image which is indistinguishable from a high-fidelity image to a given discriminator. This has been achieved by incorporating the above criteria directly into the optimization framework using a conditional GANs. Using a new refined loss function, visual artifacts are reduced and resulting visual quality is enhanced. The system dubbed Image De-raining Conditional GAN (ID-CGAN), uses a new perceptual loss to aid the proposed network to generate outputs which are artifact-free and visually pleasing. Despite its ability to outperform many SOTA single image deraining methods at the time of publishing, it suffers from an unexpected visual artifact where certain white, round rain streaks are enhanced in the final output. This could be due to the lack of diversity in the training data as well as the nature of the CNN and the subsequent enhancement by the perceptual loss function as well.

2.4.3 Usage of NAS in Image Restoration tasks

A work titled NAS-DIP by (Chen et al., 2020) proposes the usage of Neural Architecture Search to capture strong image priors for image restoration in place of hand-designed image prior architectures. The authors build upon a generic U-Net by creating new search spaces for an up-sampling cell and a cross-scale residual connection pattern. The authors have used an existing NAS algorithm involving reinforcement learning to search for improved network architectures. The effectiveness of the developed system has been validated through its successful applications towards various image restoration tasks such as denoising and super-resolution. Despite taking 3 – 5 GPU days to discover suitable network architectures, NAS-DIP performs favourably against SOTA learning-free approaches and can compete with some learning-based methods as well.

The work of (Arican et al., 2022) is one of the most recent and interesting approaches towards leveraging NAS for image restoration. Unlike previous works where an architecture is searched for an image dataset, this approach, dubbed ISNAS-DIP, searches for a DIP framework using an image-specific NAS strategy. The authors show the necessity to identify image-specific models to augment the DIP's quality and present their own image-dependent metrics for finding DIP architectures. This means that only a randomly initialized CNN network is required prior to running the NAS algorithm. Architectures are ranked without lengthy optimization processes within any search space. Through extensive experiments ranging from image denoising to super-resolution, this work proves that image-specific metrics reduce the search space to a small group of models, with the best of them outperforming existing image restoration NAS approaches. However, determining the specific top-1 model and defining a suitable early stopping point for training selected models are yet to be resolved.

(Zhang et al., 2021) proposed an image restoration strategy using a memory-efficient hierarchical NAS (dubbed HiNAS) by focusing on two image restoration tasks: Image denoising and Image super-resolution. Inner and outer search spaces were built as part of a flexible hierarchical search space using gradient based search strategies. The inner search space uses a “layer wise architecture sharing strategy” (LWAS). The outer search space is designed using a memory efficient cell-sharing strategy boosting the search speed considerably. The main consideration of this work was to boost the often-overlooked inference speed and computational/memory efficiency, which is of very high importance. HiNAS can search for a denoising network in 1 hour, and a super-resolution structure in 3.5 hours using its novel search strategy. It’s capability to compete with other SOTA methods is also quite remarkable.

2.5 Review of Technologies

2.5.1 Deep Learning and Neural Networks

Deep learning (DL) could be generally defined as an advanced branch of machine learning (ML) with specialised capabilities in discovering intricate patterns and structures embedded within large, high-dimensional datasets (LeCun, Bengio and Hinton, 2015). ML systems allow for computers to perform specific tasks in the absence of explicit programming, via the usage of mathematical algorithms and statistical models (Mahesh, 2018). Whilst ML based solutions had proven themselves effective in areas such as pattern matching/prediction, and object identification, conventional ML techniques were limited in their ability to process raw, natural data. Representation learning (RL), which describes a collection of methods that allow machines to overcome some of these limitations, was introduced as an evolution of traditional ML approaches and coincidentally formed the basis for DL. At its core, DL consists of RL methods with multiple representation levels. The key defining factor of DL is that it is capable of designing its feature layers through general purpose learning procedures using training data (LeCun, Bengio and Hinton, 2015).

The backbone of typical deep learning algorithms are built upon neural networks. In particular, convolutional neural networks (CNNs) have proven to be the most versatile and common flavour of NNs with use cases ranging from computer vision to natural language processing (Li et al., 2022). Inspired by the visual cortex of animals and initially adopted for object detection/image classification, CNNs typically consist of a collection of neurons held in an acyclic-graph arrangement (Aloysius and Geetha, 2017). All CNNs comprise of three types of layers, each performing a different role:

i. **Convolutional layers –** These form the basic unit of a CNN. A set of kernels or learnable filters form the parameters of such layers. Most computations in a CNN model take place in convolutional layers.
ii. **Pooling layers –** The main function of these layers is to reduce the spatial dimension of activation maps and parameter count in the network to reduce overall computational complexity.
iii. **Fully connected layers –** These deal with high-level reasoning tasks. Neurons in a fully connected layer are connected to all neurons in the preceding layer. Since neurons in these layers aren't spatially arranged, conv layers aren't allowed to be positioned below a fully connected layer (Aloysius and Geetha, 2017).

2.5.2 Generative Adversarial Networks

2.5.2.1 Introduction to GANs

Generative Adversarial Networks (GANs) are a class of deep learning models, first introduced by with the core aim of improving the estimation procedure for generative networks. A typical GAN model consists of two sub-models; a generator and a discriminator, both of which are usually implemented using some form of neural networks. For each training loop, the generator attempts to create fresh data examples aligned with the training dataset, whilst the discriminator attempts to successfully differentiate between generated data and true data from the dataset (Gui, Sun, et al., 2022).

GANs hold an inherent advantage in semi-supervised and unsupervised learning tasks since they provide a means to learn deep representations in the absence of training data which is extensively annotated. Typically, the generator doesn't have direct access to real data, making its interaction with the discriminator being the only source of learning. In contrast, the discriminator holds access to both real data and generated data samples, with its functionality being similar to that of a binary classifier (Creswell et al., 2018; Gui, Sun, et al., 2022).

Representations learned via GANs are useful for a wide variety of domains and applications including image generation, image editing, image style transfer, image restoration (super-resolution, deblurring, denoising etc.), audio generation and many natural language processing tasks (Creswell et al., 2018).

Optimisation of GANs presents itself as a minmax optimisation problem. The optimisation termination is reached at a point at which a minimum is formed with respect to the generator and a maximum is formed with respect to the discriminator. Simply put, the optimisation goal of any GAN model is to reach Nash equilibrium, i.e. a steady state (Gui, Sun, et al., 2022).

2.5.2.2 An Architectural Breakdown of GANs

As discussed above, the basis of any GAN model lies upon two competing neural networks; a generator G and discriminator D. Early GAN models employed a class of fully connected feedforward ANNs called multilayer perceptrons (MLPs) since they proved to be the most straightforward to implement in the given context (Goodfellow et al., 2014).

To learn the distribution of the generator p_g, over data x, a prior is defined on input noise variables $p_z(z)$. Next, a mapping to the data space is demonstrated using $G(z; \theta_g)$, where G refers to a differentiable function represented through an MLP with parameters θ_g (Goodfellow et al., 2014).

Next, a second MLP $D(x;\ \theta d)$ that produces a solitary scalar is introduced. The purpose of $D(x)$ is to represent the probability of x originating from the dataset and not p_g.

D is trained to maximise the likelihood of assigning the correct label to samples from G and those from the training set.

G is trained simultaneously to minimise $log(1 - D(G(z)))$.

In simpler terms, G and D engage in the following 2-player minimax game where the value function is $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[logD(x)] + \mathbb{E}_{x \sim p_z(z)}[\log(1 - D(G(z)))]$$

Fully optimizing D within the inner loop of training is not feasible due to computational limitations and would cause overfitting on finite datasets. Thus, the optimization process is alternated between k steps for D and a single step for G. This allows D to remain near its optimal solution as long as G doesn't change too rapidly.

2.5.2.3 GANs Architectures in Image Restoration

2.5.2.3.1 Pix2Pix GAN

Introduced by (Isola et al., 2017), Pix2Pix serves as a general-purpose solution based upon conditional adversarial networks (cGANs), towards image-to-image translation tasks. Since cGANs hold the ability to learn unique loss functions coupled to the mapping between input and expected output, this approach proves to be extremely versatile and hence applicable to a multitude of different problems. Pix2Pix uses a general U-Net shaped network with skip connections as the generator to efficiently shuttle low-level information through the network. The discriminator, dubbed "PatchGAN", focuses on modelling high frequency structures by only penalizing the output at a patch-scale, allowing the L1 loss term to deal with correctness at low-frequencies.

2.5.2.3.2 DeblurGAN

This work by (Kupyn et al., 2018), based upon the findings of (Isola et al., 2017) focuses on eliminating motion blur from images using a cGAN. The generator is composed of two sets of strided convolution blocks that use a stride of 12, nine residual blocks as well as two transposed convolution blocks. In addition, a global skip connection named ResOut is also used. A Wasserstein GAN with gradient penalty, dubbed "WGAN-GP" is used as the discriminator network during training. This is said to be architecturally similar to the PatchGAN discriminator used in Pix2Pix.

2.5.2.3.3 SRGAN

SRGAN by (Ledig et al., 2017) claims to be the first super-resolution framework capable of inferring photo-realistic results by upscaling natural images by a factor of 4. A unique perceptual loss function combining both content loss and adversarial loss is used to boost performance. The generator employs a pair of convolutional layers that have 64 feature maps and use small 3 × 3 kernels. Batch-normalization layers are applied next, along with ReLU activation functions. The network has eight convolutional layers with a growing number of filter kernels from 64 to 512, similar to VGG. Stridden convolutions are used to reduce image resolution. After generating 512 feature maps, the network uses two dense layers and a sigmoid activation function to obtain sample classification probability.

2.5.3 Neural Architecture Search for architecting Neural Networks

2.5.3.1 Introduction to NAS

Architecture engineering has seen a considerable rise in demand in the deep learning space throughout the past couple of years. This stems from the success enjoyed by applications of deep learning in perceptual tasks, largely thanks to its automation of the feature engineering process. Neural Architecture Search (NAS), a subfield of AutoML which automates the architecture engineering process, has therefore been introduced as a logical next-step towards automating and further refining machine learning workflows (Elsken, Metzen and Hutter, 2019).

To elaborate further, the core aim of NAS is to automatically search for optimal network architectures, which traditionally is considered to be a delicate, time-consuming, and error-prone process requiring a great deal of domain knowledge and expertise. In broader terms, this automation process is defined as a search problem that iterates over a decision set defining various components of a neural network. Whilst early NAS systems utilised reinforcement learning as part of their search algorithms, they proved to be extremely resource intensive, consuming hundreds of GPU hours to discover an optimised architecture. Subsequent research attempts succeeded in reducing this computational burden considerably through the usage of techniques akin to transfer learning (Wistuba, Rawat and Pedapati, 2019).

NAS systems effectively consist of 3 main components: Search space, search strategy and performance estimation strategy. The search space refers to the types of architectures and components represented and considered in the optimisation process, search strategy describes method of exploration of the search space, and performance estimation strategy simply refers to the means of estimating the performance of searched architectures. It must be noted that whilst discovering an optimised architecture quickly is desirable, premature convergence of the search process should be avoided as it may result in producing a suboptimal architecture (Elsken, Metzen and Hutter, 2019). NAS systems have proven to be capable of matching and even outperforming SOTA manually designed architectures for tasks such as image segmentation, classification and object detection (Elsken, Metzen and Hutter, 2019).

2.5.3.2 An Architectural Breakdown of NAS

2.5.3.2.1 Search Space

The range of possibilities for creating architectures is determined by the search space. By including information about common characteristics of effective architectures for a particular task, the search space can be narrowed down and made less complex. However, this approach also brings in a human perspective that could limit the discovery of innovative architectural elements that are not yet known to the research community (Elsken, Metzen and Hutter, 2019). Discussed below are few such search spaces described in recent NAS research.

1. Chain Structured Neural Network Search Spaces

This is a relatively simple search space. The entire architecture could be represented as an n layer sequence where L_i (layer i) receives inputs from layer $i - 1$ and passes on its output as the input for layer $i + 1$ (Elsken, Metzen and Hutter, 2019).

The search space is defined by several parameters, which include, (i) the maximum number of layers (which can be unlimited), (ii) the type of operation that each layer performs (such as pooling, convolution, or more complex methods like depth wise separable convolutions or dilated convolutions), and (iii) the hyperparameters that are associated with these operations (Elsken, Metzen and Hutter, 2019).

2. Cell-Based Representation

Inspired by architectures built with repeated motifs, (Zoph and Le, 2017) proposed to search for such segments called *cells (or blocks)* instead of entire architectures. The optimization process involves two types of cells: a normal cell that maintains input dimensionality, and a reduction cell that decreases spatial dimension. These cells are combined in a predetermined pattern to create the final architecture. This approach holds three main advantages over other conventional approaches:

i. The size of the search space is significantly decreased.
ii. Models created using cells can be easily transferred or adapted to different datasets by adjusting the number of cells and filters used in the model.
iii. It provides strong evidence of the effectiveness of the design pattern of stacking modules repeatedly in architecture engineering (e.g.: Repeating LSTM blocks in RNNs) (Elsken, Metzen and Hutter, 2019).

3. Hierarchical Search Space

To make use of previously discovered and well-defined network motifs, the NAS search space can be constrained as a hierarchical structure. Most notably, this type of search space was adopted by (Zhang et al., 2021) where an inner search space was used to search for topological architectures of inner cells whilst an outer search space was tasked with searching for optimal cell-widths, for a network intended for image restoration.

2.5.3.2.2 Search Strategy

The way of navigating through the search space (the size of which is typically extensive and potentially endless) is described by the search strategy. This involves balancing the trade-off between exploration and exploitation, where it's important to discover high-performing architectures quickly but also avoid getting stuck in a suboptimal region resulting in premature convergence (Elsken, Metzen and Hutter, 2019).

2.5.3.2.3 Performance Estimation Strategy

The main goal of NAS is usually to discover architectures that exhibit excellent predictive performance on new data. To evaluate this performance, a process known as Performance Estimation is utilized. The basic approach involves training and validating the architecture on data, but this can be time-consuming and restricts the exploration of numerous architectures. Consequently, current studies concentrate on developing techniques that lower the cost of these performance estimations (Elsken, Metzen and Hutter, 2019).

2.5.4 NAS integration with GAN models

2.5.4.1 AutoGAN

This work by (Gong et al., 2019) presents itself as the first attempt towards automating the process of architecting a GAN model using NAS. The authors proposed a NAS algorithm for improving the generator network architecture of GANs instead of attempting to architect both G and D simultaneously due to the inherently complex nature of these models. Training GAN models is a generally unstable task, and the proposed algorithm addresses this by using a search strategy spanning multiple levels, guided by an RNN controller updated using reinforcement learning with the Inception Score used as the reward. The algorithm generates multiple generator architectures, and the one with the highest Inception Score is selected as the final architecture (Gong et al., 2019).

2.5.4.2 AlphaGAN

AlphaGAN by (Lutz, Amplianitis and Smolic, 2018) proposes a novel approach for improving GANs from the perspective of NAS through inspiration drawn from game-theory concepts. Unlike most NAS methods, which rely on validation sets and evaluation metrics for architecture search, AlphaGAN uses a differential evaluation metric to guide the search towards achieving the pure Nash Equilibrium between the generator G and discriminator D. The proposed search framework is a bi-level minimax optimization problem solved using stochastic gradient methods, where the outer-level objective optimizes the generator architecture parameters, and the inner-level objective optimizes the weight parameters of the searched architecture. AlphaGAN extends the Differentiable Architecture Search (DARTS) method, which aims to find the best architectures that can achieve high performance on validation sets trained on network parameters of the training set (Lutz, Amplianitis and Smolic, 2018).

2.5.4.3 AdversarialNAS

AdversarialNAS presents a novel approach introduced by (Gao et al., 2020) to simultaneously search for superior generator and discriminator architectures of unconditional image generation GAN models. A large initial search space is designed and then relaxed, allowing for continuous search. The discriminator architecture is directly used to evaluate the generator architecture, guiding the search direction, and updating the generator architecture's distribution using gradient descent. The discriminator architecture is actively changed during the search process to maintain balance. Similarly, the generator computes the loss and updates the discriminator through stochastic gradient ascent, thus creating an adversarial mechanism where both architectures compete to continuously improve their performance. This removes the requirement to calculate evaluation metrics and guarantees a balanced approach, leading to an improvement in the search for an enhanced generative model (Gao et al., 2020).

2.6 Evaluation Methods

2.6.1 Evaluation Metrics for Image Restoration Systems

To assess the effectiveness of image restoration models on individual images, a range of quantitative measures are utilized within the field. Mean Squared Error (MSE), Structural Similarity Index (SSIM), Peak Signal-to-Noise Ratio (PSNR), and Feature Similarity Indexing Method (FSIM), are some of the most notable and popular such evaluation metrics (Sara, Akter and Uddin, 2019).

Whilst the presence of a reference ground truth image is essential for a truly authentic evaluation of image quality, such benchmarks are often unavailable in real-world use cases. This leaves full-reference metrics such as those mentioned above as the sole measurement of performance for image restoration systems. Unlike MSE and PSNR, SSIM and FSIM metrics are normalized from a representation perspective. Therefore, the error values provided by SSIM and FSIM which are based on perception and saliency are more comprehensible to us humans compared to the absolute error values provided by MSE and PSNR scores.

However, both PSNR and SSIM have been used as the main evaluation metrics of choice by a large majority of recent studies in pursuit of a more holistic evaluation (Sara, Akter and Uddin, 2019).

2.6.1.1 Peak Signal-to-Noise Ratio (PSNR)

The primary metric utilized to assess the quality of an image in relation to the noise that may distort it is PSNR. This metric determines the signal as the original image and treats any form of distortion or noise as an error. PSNR is a reliable indicator of image quality as it closely approximates the human perspective. Since the range of values can be quite extensive (peak value could tend towards infinity), PSNR is calculated in decibels using a logarithmic scale (Sara, Akter and Uddin, 2019).

Typical PSNR values relating to the quality degradation of image and video compression for 8-bit data could vary between 30 – 50 dB, and from 60 – 80 dB for 16 bit data.

The formula for PSNR, where "peakval" represents the highest value in image data and MSE represents the Mean Square Error, could be given as follows:

$$PSNR \ = \ 10 \ log10 \ (peakval2)/MSE$$

2.6.1.2 Structural Similarity Index (SSIM)

SSIM measures changes in perception caused by image degradation by comparing the interdependent pixel values. This evaluation between the original image and the restored image can reveal important details about the contents of the images. SSIM is a composite measure that incorporates luminance (l), contrast (c), and structure (s) (Sara, Akter and Uddin, 2019). Using these terms, and with x representing the original image, y representing the generated image, and α, β, and γ being parameters that adjust the relative importance of each component, SSIM can be mathematically represented as:

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

SSIM extends itself to a couple of different flavours, namely, Multi-Scale Structural Similarity Index Method (MS-SSIM) and Three-Component SSIM (3-SSIM). The prior focuses on comparing the structural similarity of images at different image scales whilst the latter attempts to follow the natural human vision system where details in textured regions are observed more accurately than those in smooth regions. 3-SSIM splits images according to three important sections; edges, textures, and smooth regions. The final score achieved is the average of 3 separate weighted scores (0.5 for edges, 0.2 for texture and 0.25 for smooth regions) (Ran and Farvardin, 1995; Sara, Akter and Uddin, 2019).

2.6.1.3 Mean Square Error (MSE)

MSE is a widely used full-reference metric to measure image quality, where values closer to zero indicate better results. It incorporates the variance and bias of the estimator and is the second moment of the error. The Root-Mean-Square Error (RMSE) or Root-Mean-Square Deviation is a concept introduced by the Mean Squared Error (MSE), which is often referred to as the standard deviation of the variance. MSE is a function of risk and measures the average of the square of errors between the estimator and the estimated outcome of an unobserved quantity in the image.

The MSE between two images $g(x, y)$ and $\hat{g}(x, y)$, where M and N represent the dimensions of the image respectively (in pixels), and (x, y) refers to a specific pixel location could be represented as follows:

$$MSE = \frac{1}{MN}\sum_{n=0}^{M}\sum_{m=1}^{N}[\,\hat{g}(n,m) - g(n,m)\,]^2$$

2.6.1.4 Features Similarity Index Matrix (FSIM)

The Feature Similarity Index Method measures the similarity between two images by mapping their features. It employs two criteria: Phase Congruency (PC) and Gradient Magnitude (GM). Phase Congruency is a method for detecting image features that is invariant to light variation and can detect interesting features in the frequency domain. Gradient Magnitude measures the gradients of an image using convolution masks and can be defined as the magnitude of the image's horizontal and vertical gradients.

2.7 Chapter Summary

To summarise, this chapter provided a thorough overview of image restoration, covering the problem domain, existing work, technological review, and relevant evaluation methods. This chapter highlighted the limitations of current methods and emphasized the need for more advanced and effective techniques. It also analysed the various technologies used in this domain and discussed the cross integration of GANs and NAS in other computer vision tasks. Evaluation methods used to assess the performance of image restoration models were also covered. Overall, this chapter provided a foundation for the proposed methodology presented in subsequent chapters and helps to contextualize the research within the larger field of image restoration.

3 METHODOLOGY

3.1 Chapter Overview

The success of any project hinges on the various methodologies followed to tackle different aspects of the venture. As such, this chapter focuses on discussing those methodologies selected for the research, development, and project management components of this research project. Furthermore, a general timeline relating to the deliverables of this project is provided at the end of this chapter.

3.2 Research Methodology

The following table provides an overview of the selected scientific research methodology based on the Research 'Onion' Model presented by (Saunders, Lewis and Thornhill, 2009)

Research Philosophy	From amongst the 4 main available research philosophies, positivism, interpretivism, pragmatism and realism, **Pragmatism** was selected. The reasoning behind this decision was the fact that image restoration tasks such as image deblurring and super-resolution are ill-posed problems, and to find an optimal approach to reach the research goal would require the author to experiment and investigate various methodologies along the hypothesis.
Research Approach	Out of the available research approaches, namely, deductive, and inductive, the **deductive** approach was chosen for this project. This was selected as this is a **quantitative and qualitative** research where the author would have to experiment with different GANs and other deep learning approaches used for image restoration in order to optimize the proposed system to prove the original hypothesis.
Research Strategy	The next layer of the Onion model deals with research strategies. Research strategy provides the general direction and execution of a given research project. As such, **experiments** and **surveys** shall be used as primary research strategies as quantitative data is paramount when fine-tuning the Machine Learning (ML)/Deep Learning (DL) models. Interviews may be used during latter stages of the project as a means of collecting qualitative data.

Research Choice	From the available research choices, the mono-method, mixed-methods and multi-methods, **mixed-methods** has been selected. This is due to the fact that both **quantitative** (experiments and surveys) and **qualitative** (interviews) approaches being used as means of data gathering throughout the research process.
Time Horizon	Out of the available time horizons, longitudinal and cross sectional, the **cross-sectional** time horizon was selected for this research as data gathering would be done at a single point in time in order to conduct evaluations efficiently.
Techniques and Procedures	For the purpose of data gathering and analysis, a multitude of techniques such as trial-and-error observations, surveys, interviews, reviewing literature of similar work would be utilized.

Table 3-1: Research Methodology

3.3 Development Methodology

Out of the available Software Development models, the **Prototyping Model** was chosen as it allows to seamlessly develop, evaluate, and re-iterate the prototype as required. Being a research project, this model is best suited for the purpose as it accommodates for improvements and changes that would be made throughout the project's duration, based on new research findings, self-evaluations as well as evaluations by industry experts.

3.4 Project Management Methodology

The **Prince 2 Agile** hybrid methodology was selected for the purpose of managing this project, out of the many project management methodologies available. Prince 2 follows a strategic-level project management methodology whilst Agile focuses on being more flexible with incremental improvements to the project via constant reassessments and execution. The combination of the above would provide for a more holistic approach to manage this research project.

3.4.1 Schedule

The **Gantt chart** outlining the timeline of this project is presented in **APPENDIX - II**

3.4.1.1 Deliverables

Deliverables, milestones, and dates of deliverables

Deliverable	Date
Literature Review Critical review of the existing work in the domain.	27th Oct 2022
Final Project Proposal and Ethics Form Finalized Initial Project Proposal.	3rd Nov 2022
Software Requirement Specification Document specifying the requirements to be satisfied by the prototype.	24th Nov 2022
Proof Of Concept Initial proof-of-concept of the proposed work.	5th Dec 2022
Initial Project Specifications Design and Prototype (PSDP) Submission	23rd Jan 2023
Final Project Specifications Design and Prototype (PSDP)	23rd Feb 2023
Test and Evaluation Report Document listing the findings of evaluations and testing of the PoC	23rd Mar 2023
Submission of Draft Project Reports Submission of the initial draft of the thesis to obtain feedback from supervisors.	30th Mar 2023
Final Thesis Final Thesis document detailing all processes and findings related to the completed research project. Submitting a conference paper based on research findings.	27th Apr 2023

Table 3-2: Deliverables

3.5 Resource Management

Based on the objectives, methodologies and functionalities defined above, the identified hardware, software, data, and skill requirements for this project along with their justifications are as follows:

3.5.1.1 Hardware Requirements

- **Intel 8th gen Core i5 processor or above –** For computationally intensive ML model training purposes.
- **Nvidia GTX 1660 GPU or above (With CUDA support) –** For GPU intensive training of Deep Learning models.
- **50GB Disk space or more –** To store datasets and trained models.

- **16GB RAM or more –** To manage large datasets as well as trained models.

Ps: If hardware requirements of the local machine (as mentioned above) prove to be insufficient Google Colab, or a similar cloud environment would be used.

3.5.1.2 Software Requirements

- **Operating System (Windows / MacOS / Linux) –** An operating system is vital to run essential programs such as Integrated Development Environments (IDEs), web browsers and word processing software to complete various deliverables of the project.
- **Python –** Python would be used as the primary programming language to build and train the required ML models. The reason for this is the requirement to use certain frameworks such as PyTorch which are Python-based.
- Jupyter NoteBook / Pycharm / Visual Studio Code – IDEs and text editors used for development
- **GitHub** – Version Control System used to store all project-implementation related files.
- **Zotero/Mendeley** – Document and reference management tools used to manage and view research documents related to the project.
- **MS Word/Google Docs** – Word processing software used for documentation purposes.
- **Google Drive/DropBox** – Cloud storage used to manage project files and documentation (unrelated to implementation).

3.5.1.3 Data Requirements

The following datasets have been identified as potential candidates for training and testing the proposed system:

- RealBlur Dataset (Jaesung Rim, 2020)
- GoPro – V7 Dataset (Nah, Kim and Lee, 2017)
- NAS for DIP Dataset (Arican* et al., 2022)
- DIV2K Dataset (Agustsson and Timofte, 2017)
- BSD/CBSD Dataset (Martin et al., 2001)

3.5.1.4 Skills Requirements

- **Machine Learning/Deep Learning knowledge –** Since the project is based on ML/DL concepts, a thorough knowledge on Machine Learning/Deep learning techniques, models, and algorithms, especially related to the Image Processing domain would be required.

- **Design Skills –** Since the early stages of the project consists of detailed design steps (both Technical and Non-Technical), knowledge and skills necessary to create various design and technical diagrams would be required.
- **Academic Writing Skills –** Since the project deliverables consist of multiple document submissions, including the final thesis, good academic writing skills would be essential for successful completion of the same.
- **Communication Skills –** For Requirement gathering, and presentations.

3.6 Risk Management

The following table outlines the potential risks identified prior to the commencement of this project along with their predicted severity, frequency and actionable mitigation plans.

Risk Item	Severity	Frequency	Mitigation Plan
Changing nature of project requirements – Based on the continuous research and new knowledge gained thereof, initially identified requirements may change.	4	5	Use an iterative development path and maintain consistent contact with the project supervisor and other experts for constructive feedback and advice.
Lack of technological expertise as well as domain knowledge – Since the domain of Image Processing, ML and DL are unfamiliar to the author, they may present themselves as limiting factors.	4	4	Continue learning and expanding the knowledge base on the required technologies and techniques to implement the project. Resources such as YouTube, online courses, StackOverflow, academic research as well as input from domain experts could be used.
Hardware resources being insufficient to complete implementation – Locally available resources may prove to be insufficient to	4	4	If the available resources prove to be insufficient, cloud-based development environments and resources (such as Google Colab/Vertex-

train the models and execute other implementation components.			AI/QBlocks/Amazon Sagemaker) could be used.
Losing valuable project documentation – The risk of losing project documentation or progress due to unforeseen storage-related issues or data corruption is present.	5	3	Backup the documentation and any progress made composing them to cloud.
Hypothesis being Invalid – New research conducted in the domain may invalidate the hypothesis.	4	3	Continuous research will be conducted throughout the project's duration to tweak the final result as required, to ensure a contribution to the domain is made.
Other unpredictable risks – There could be unpredictable risks related to external environmental factors such as power-cuts which could affect the progression of the project.	3	3	Be methodical with the process and manage work according to a timetable with weekly/daily goals. This would help minimize risks and allow for some breathing room should something of such nature occur.

Table 3-3: Risk Management

3.7 Chapter Summary

In this chapter, the methodologies for research, development, and project management were discussed, along with the reasoning for selecting each requirement and plans for mitigating any potential risks that may arise.

4 SOFTWARE REQUIREMENT SPECIFICATION

4.1 Chapter Overview

This chapter provides an in-depth view of the entire requirement gathering process of this research project, spanning from the core research component to the development of the prototype. At the outset, stakeholders of to the proposed system are identified and illustrated through graphical means along with their roles and relationships. Next, potential requirement elicitation techniques are discussed and analysed with detailed reasoning for their selection. The gathered requirements are then presented in tabular form and use case diagrams with supporting descriptions are produced. Ultimately, the findings made through the requirement elicitation process are interpreted and summarized.

4.2 Rich Picture Diagram

The purpose of a rich picture diagram is to demonstrate the relationships that would exist between the proposed system and its wider environment. These entities may extend beyond the system's own domain through tertiary relationships. Additionally, this rich picture diagram also demonstrates potential threats and vulnerabilities that may affect the proposed system due to activities of malicious actors and lawful competitors alike.

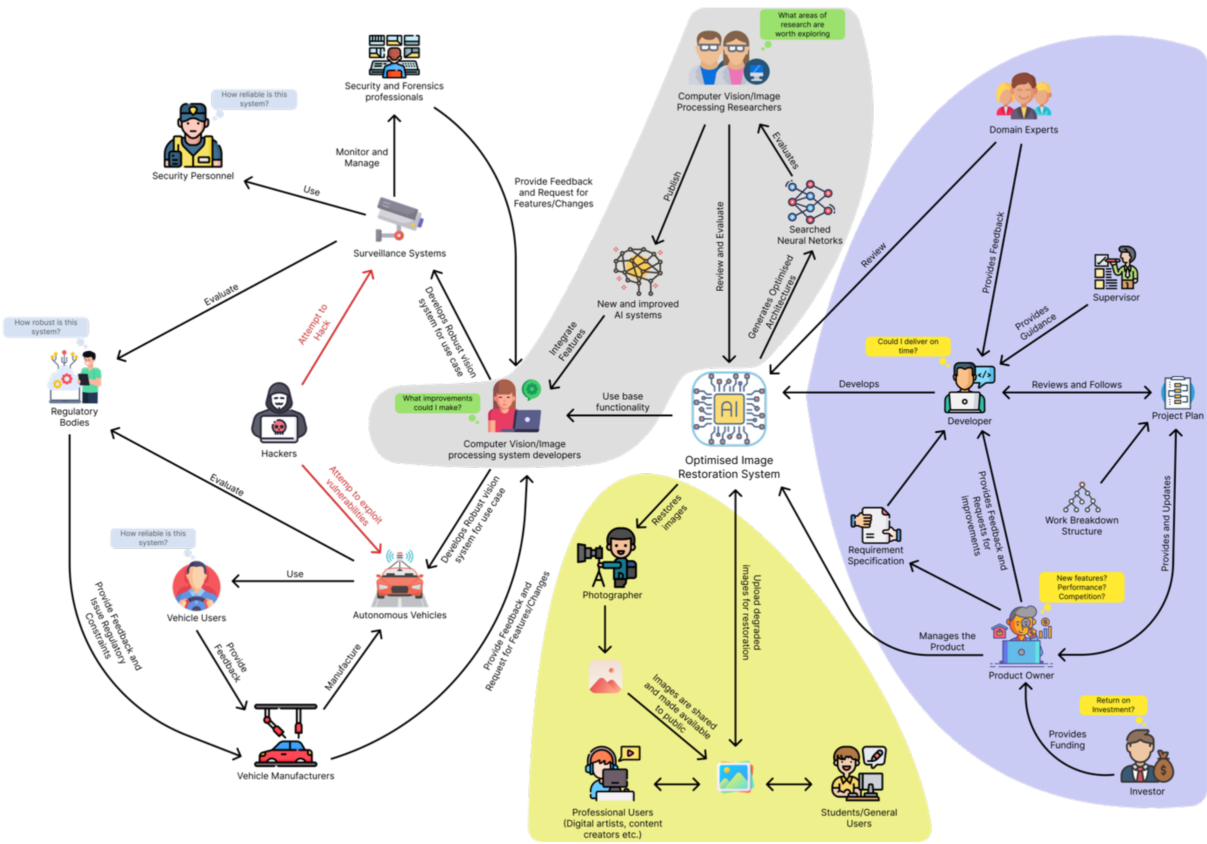


Figure 4-1: Rich Picture Diagram (Self-Composed)

4.3 Stakeholder Analysis

The main stakeholders who participate and interact with the proposed prototype system, identified through the creation of the above rich-picture diagram are presented here within an onion-model diagram. Further, a detailed description of each stakeholder is presented in tabular form for better clarity and understanding.

4.3.1 Stakeholder Onion Model

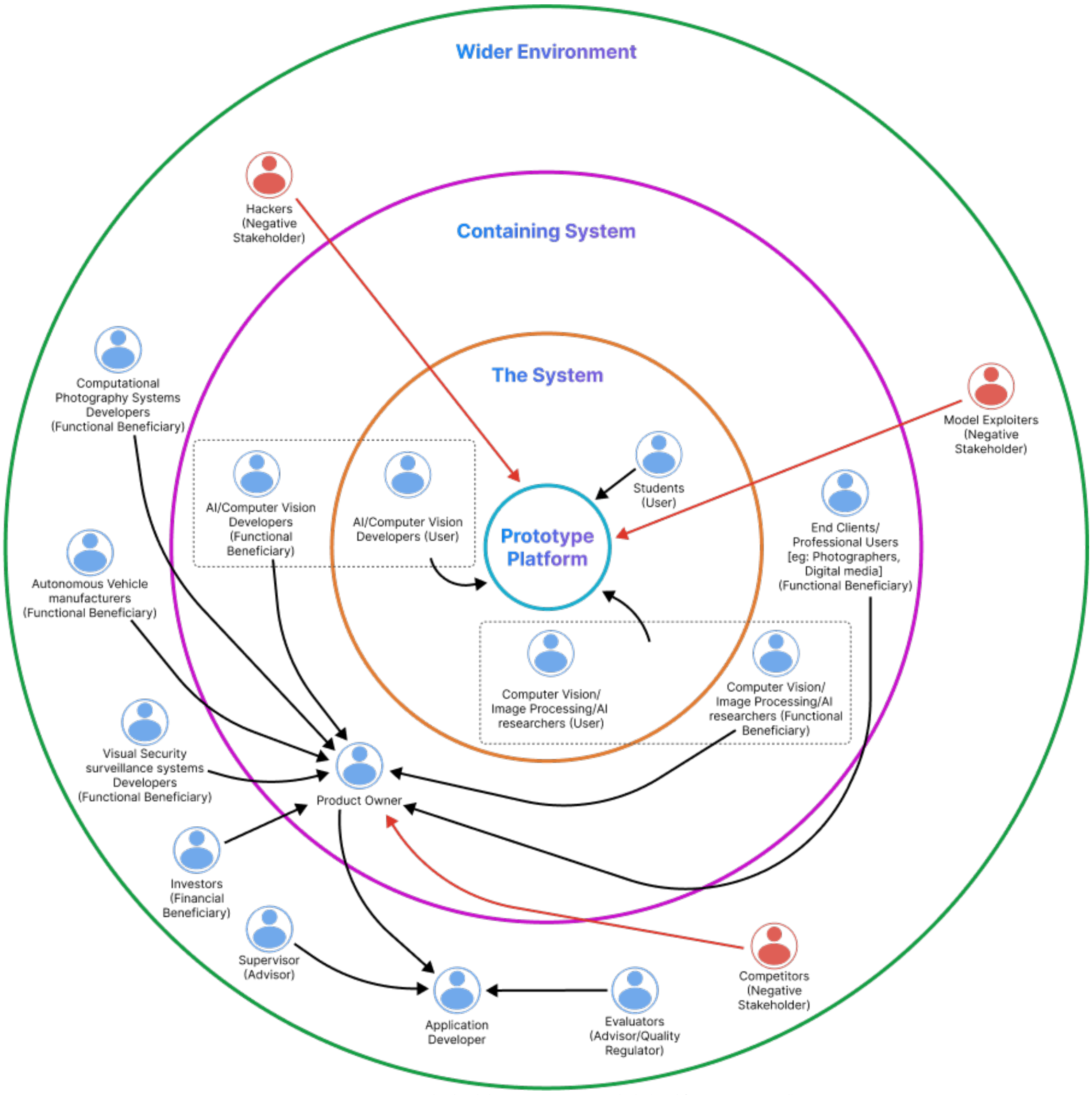


Figure 4-2: Stakeholder Onion Model (Self-Composed)

4.3.2 Stakeholder Viewpoints

Stakeholder	Role	Role Description
Core System		

Students	User/Functional beneficiary	Uses the system and observes how the proposed research project could help improve architecting image restoration neural networks and subsequent GAN models with which they're integrated.
AI/Computer Vision developers	User	Uses the research concepts to develop improved image restoration systems embedded within other complex applications such as autonomous vehicles and visual surveillance systems.
Computer Vision/Image Processing researchers	User	Critically evaluate the research concept and findings and research/provide suggestions on new ways of further enhancing image restoration and related computer vision tasks.
Containing System		
Product Owner	Operational beneficiary Financial beneficiary	Monitor and manage business process and make decision related to the product based on feedback received from other stakeholders.
AI/Computer Vision developers	Functional beneficiary	Provide feedback to the product owner on potential improvements that could be made to the product.
Computer Vision/Image Processing researchers	Functional beneficiary	Provide feedback on improvements that could be made or come up with improved approaches to enhance the core features of the product.

End Clients/Professional Users	Functional beneficiary	Use the product for personal use cases or as a tool for professional requirements.
Investors	Financial beneficiary	Invests in the product and provides funds for further improvements of its features as well as for further research which could help boost performance and efficiency further.
The Wider Environment		
Application Developer	Development and operational staff	Develops the core functionalities of the system and makes improvements/fixes issues wherever possible.
Computational photography systems developers	Functional beneficiary	Uses the research findings to improves capabilities of their own products.
Autonomous Vehicle vision system developers	Functional beneficiary	Uses the research findings to improves capabilities of their own products.
Visual surveillance systems developers	Functional beneficiary	Uses the research findings to improves capabilities of their own products.
Supervisor	Advisor	Guides the developer throughout the conceptualization and development phase and provides feedback as required.
Evaluator	Advisor/Quality Regulator	Provides feedback on potential issues and improvements to the concept and prototype system.
Competitors	Negatives stakeholder	Develops competing image restoration systems using similar or different approaches.

Hackers	Negative stakeholder	Attempts to hack the system.
Model Exploiters	Negative stakeholder	Attempts to exploit the generated image restoration models/network architectures for personal gain or other malicious intent.

Table 4-1: Stakeholder Analysis

4.4 Selection of Requirement Elicitation Methodologies

Requirement Elicitation is a vital component of software development which focuses on gathering the requirements for a software project that allows the developers to obtain a holistic understanding of its purpose and ultimate goals. The tables below summarize the selected requirement elicitation methods along with the relevant justifications, weighing in their pros and cons.

Technique 1 – Literature Review
Exploring, studying, and reviewing existing literature works stands as the initial step in gathering requirements for any research project. This is instrumental in obtaining a clear view of the selected field of research and helps in identifying limitations of the corpus of contemporary research work and available future research directions. Additionally, reviewing literature helps to identify potential approaches and techniques that could help successfully navigate rather complex topics such as NAS and GANs in computer vision.
Technique 2 – Survey
Conducting surveys is one of the simplest, yet most effective forms of gathering requirements for any project or task – obtaining the public's opinion. This allows to gather and quantify data regarding the various requirements, expectations and wishes that end users may have regarding the proposed solution/product. A general questionnaire, being one of the most straightforward means of conducting a survey was selected as it allows for ease of distribution (publicly, since general users are the target demographic) and comprehension for all parties involved.
Technique 3 – Structured Interviews
Since AI/Computer Vision developers and Computer Vision/Image Processing researchers are also part of the target demographic, structured interviews were selected as a requirement gathering instrument. One-to-one structured interview sessions allows for better communication of complex concepts, ideas and points of advice and is arguably the fastest and most efficient means of gathering requirements from such personnel. Furthermore,

obtaining expert feedback to assess and validate the proposed architecture and prototype features is an essential step moving forward. However, features included in the final implementation may vary based on many other factors.
Technique 4 – Prototyping
Exploratory prototyping was selected as the main development approach for the project. This allows for the continuous evaluation, validation, and improvement of the proposed system throughout its development cycle. Since the aim of this research is to develop a novel image restoration model using NAS with GANs, an exploratory prototyping approach would serve well towards identifying potential improvements, optimizations, and adjustments in the design and development of the core research component. Feedback could be obtained from experts throughout the development process once the initial prototype is built.

Table 4-2: Requirement Elicitation Methods

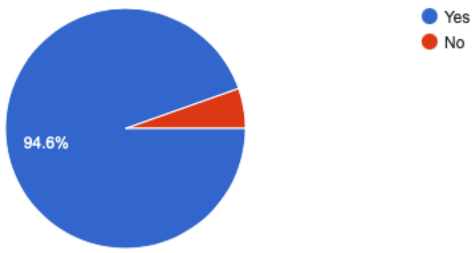
4.5 Discussion of Findings

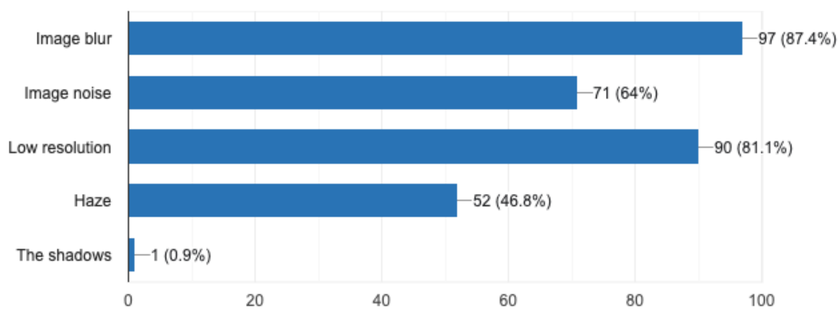
4.5.1 Literature Review

Citation	Findings
(Su, Xu and Yin, 2022)	Most modern methods used for image restoration use deep learning-based approaches to restore images. GANs and other discriminative learning methods have been experimented with
(Chen et al., 2020; Ulyanov, Vedaldi and Lempitsky, 2020)	Deep image priors are capable of efficiently solving standard inverse image restoration problems such as deblurring and super-resolution. Their U-net architecture with Encoder and Decoder blocks play a key role in this process. Moreover, using NAS to architect such U-Net models allows to capture stronger priors from images.
(Kupyn et al., 2018; Pan et al., 2020)	GAN models are equally capable of optimizing image restoration models built for multiple restorations tasks (such as DIP's) as well as addressing specific restoration problems such as de-blurring and de-noising.
(Gong et al., 2019; Ganepola and Wirasingha, 2021b)	The usage of NAS with GAN models designed for computer vision tasks allow for an increase in performance and efficiency. This is reflected both whilst building and testing such deep learning models. The literature has demonstrated that using NAS with GANs for image restoration could, therefore, yield better results.

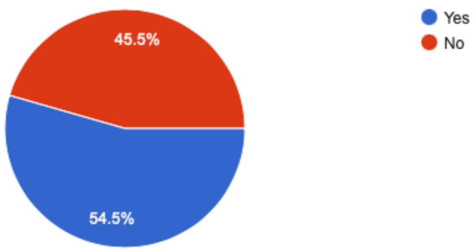
Table 4-3: Requirement Elicitation - Literature Review

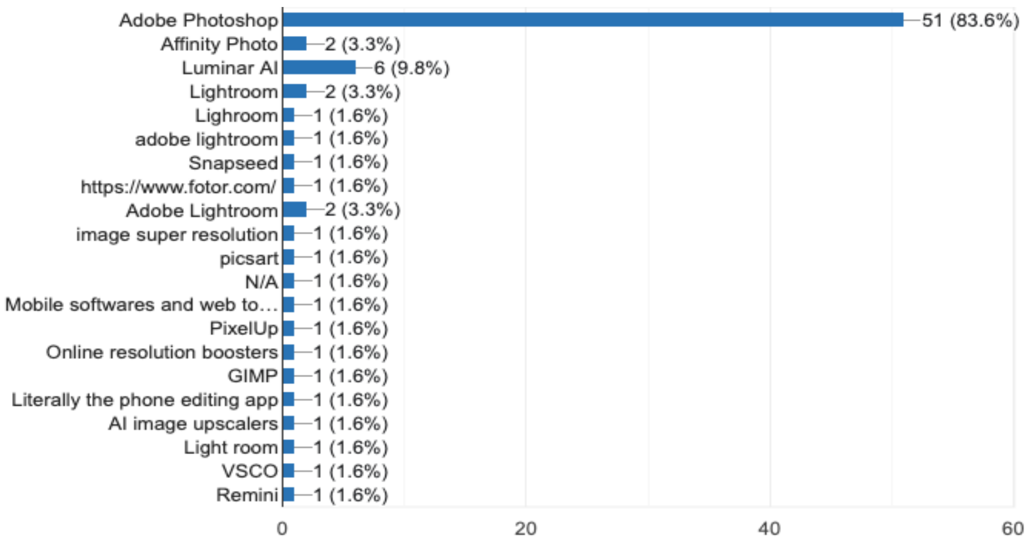
4.5.2 Survey

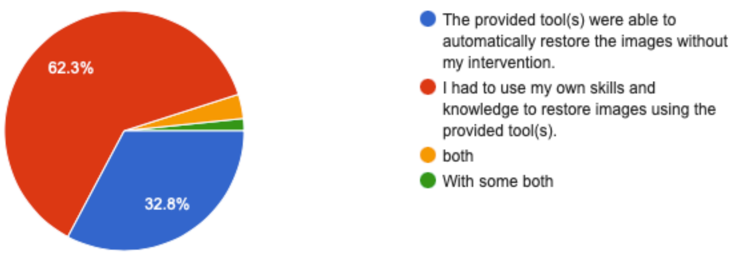
<table>
<tr><td>Question</td><td>Do you take photos?</td></tr>
<tr><td>Aim of Question</td><td>Identification and filtering of participants</td></tr>
<tr><td colspan="2">Findings


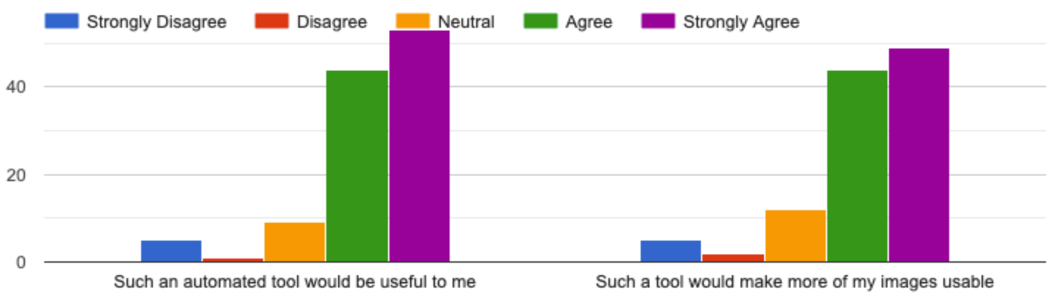
As expected, almost all survey participants (close to 95%) were individuals who took photos at some capacity in their daily lives. This validates the fact that almost all responses received for this survey were through individuals with some form of vested interest in digital images and therefore were familiar with the broader domain to which the proposed system would apply. However, the survey continues for all parties (including those who answered “no”) to obtain a more holistic understanding of the general consensus towards the project.</td></tr>
<tr><td>Question</td><td>Have you ever noticed any of the following visual degradations in images captured by you or someone else?
1. Image blur
2. Image noise
3. Low resolution
4. Haze</td></tr>
<tr><td>Aim of Question</td><td>Validation that the participants were aware of common visual degradations in images and quantification of the most perceived visual degradations.</td></tr>
<tr><td colspan="2">Findings


The most common type of image degradation experience by the participants turns out to be image blur (over 87%) whilst low resolution in images trails at a close second (still over 80%). It could be safely assumed that this observation is interlinked with various compressions that images are subject to when being stored and shared between devices.</td></tr>
</table>

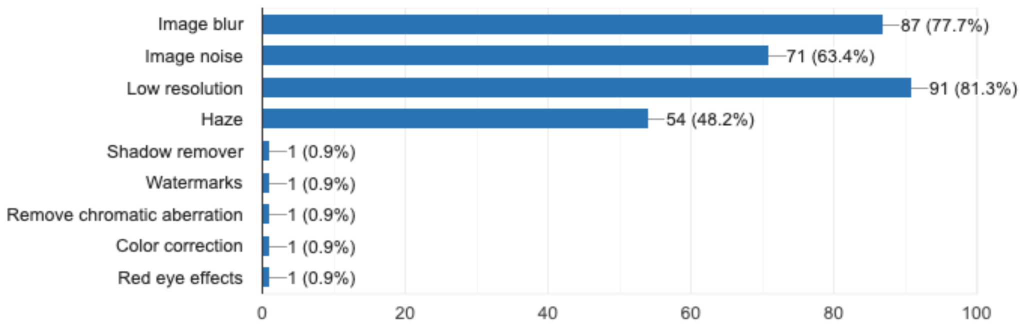
<table>
<tr><td colspan="2">Image noise and haze also have been selected by a considerable number of respondents as commonly noticed visual degradations. Additionally, one respondent has included shadows in images as a type of visual degradation, which could be the result of the drop in visual clarity in low-light instances.</td></tr>
<tr><td>Question</td><td>Have you ever used any software tools to remove such visual degradations from images?</td></tr>
<tr><td>Aim of Question</td><td>Assessing the familiarity of the participants with available image manipulation tools</td></tr>
<tr><td colspan="2">Findings


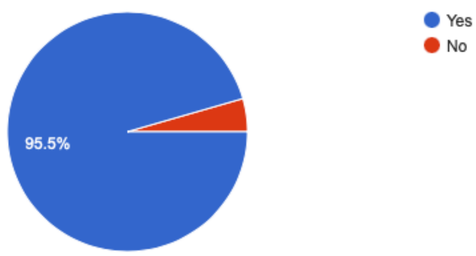
The responses to this have been quite evenly matched with a slight majority (almost 55%) showing familiarity with image manipulation tools that could be used to alleviate certain image degradations.</td></tr>
<tr><td>Question</td><td>Which of the following applications have you used to remove such image degradations?
1. Adobe Photoshop
2. Affinity Photo
3. Luminar AI</td></tr>
<tr><td>Aim of Question</td><td>Identification of available solutions within the problem domain</td></tr>
<tr><td colspan="2">Findings

</td></tr>
</table>

<table>
<tr><td colspan="2">An overwhelming majority (83%) of participants who had shown familiarity with image manipulation tools have used Photoshop, one of the most popular applications for digital art and image manipulation to restore images with visual degradations. Image manipulation tools which use AI and other such techniques have also been used by a small minority, which could be indicators of different limitations of said systems.</td></tr>
<tr><td>Question</td><td>Were the methods/tools used to restore images using the aforementioned application(s), automatic or manual?</td></tr>
<tr><td>Aim of Question</td><td>Assessment of the penetration of AI based approaches towards solving image restoration problems.</td></tr>
<tr><td colspan="2">Findings


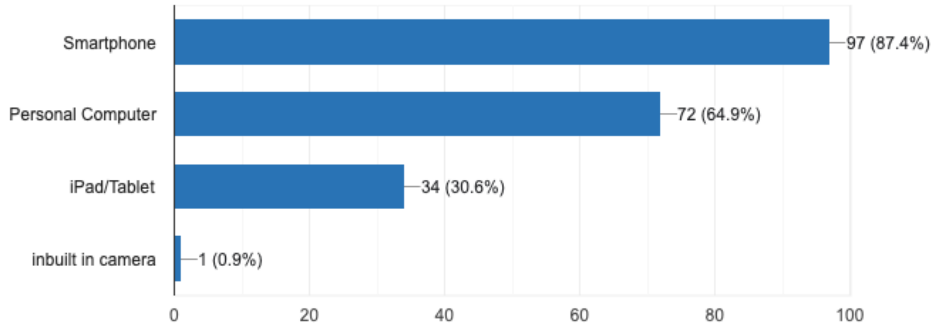
The majority of users (over 62%) have had to use completely manual methods to rectify degraded whilst a small fraction of participants have used applications where a certain level of human intervention had been required. A sizable portion of the respondents have also had experience with completely automated image restoration systems which is a testament to the capability and potential of applying AI based solutions to this domain.</td></tr>
<tr><td>Question</td><td>If there was an automated tool that could remove visual degradations from your images,
1. Would you find it useful?
2. Would it make more of your images usable?</td></tr>
<tr><td>Aim of Question</td><td>Validation of the concept of the proposed system</td></tr>
<tr><td colspan="2">Findings


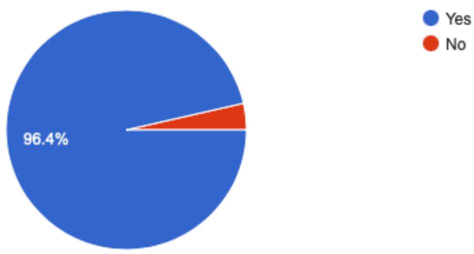
The response to both questions are overwhelmingly positive, with 97/111 participants agreeing or strongly agreeing that an automated image restoration tool of the proposed nature</td></tr>
</table>

<table>
<tr><td colspan="2">would be useful to them. Furthermore, 93/111 participants have also responded that they either agree or strongly agree with the fact that such an image restoration tool would make more of their images usable. This validates that the proposed system would be of great use to the vast majority of general users.</td></tr>
<tr><td>Question</td><td>Which type of image degradation would you like to see addressed by such an automated image restoration tool?</td></tr>
<tr><td>Aim of Question</td><td>Identifying which areas of image restoration should receive more focus, given equal technical feasibility.</td></tr>
<tr><td colspan="2">Findings


Once again, low resolution (over 81%) and image blur (almost 78%) have been selected as the most requested type of image degradation to be addressed. Image noise and haze have also been requested as areas to which the solution could cater by popular demand. Further, watermarks, chromatic aberration as well as color corrections have been listed as areas of image degradations to which a solution would be welcomed. However, the primary focus would be on addressing the main types of image degradations that have been identified and validated through these responses.</td></tr>
<tr><td>Question</td><td>Would you use such an automated image restoration system to restore your own images? (Provided it did not store any of your images or personal data permanently)</td></tr>
<tr><td>Aim of Question</td><td>Validation of the usefulness of the prototype to the participants</td></tr>
<tr><td colspan="2">Findings


Once more, the vast majority of respondents (over 95%) have confirmed that such an automated image restoration system that upholds user privacy would be of personal use to them towards restoring their own degraded images. This validates the core theme</td></tr>
</table>

<table>
<tr><td colspan="2">of this research project and confirms the suitability and usefulness of the conceptualized prototype to demonstrate the proposed novel image restoration approach.</td></tr>
<tr><td>Question</td><td>If yes, what device(s) would you like to use such an application on?
1. Smartphone
2. Personal Computer
3. iPad/Tablet</td></tr>
<tr><td>Aim of Question</td><td>Identification of the most suitable candidate as a prototyping platform</td></tr>
<tr><td colspan="2">Findings


Smartphones appear to be the predominantly preferred platform to use such an application with over 87% of the participants responding in favour of it. However, personal computers as well as iPads/tablets have also been selected as preferred platforms by a considerable number of respondents, with an aggregate of 95.5% of the user base being in favour. (Multiple selections were allowed). A single participant has shown preference towards such an application being implemented as part of the camera itself. Based on the above diverse distribution of preferences, a web-application appears to be most suitable due to the versatility and platform-agnostic nature afforded through the ability to run the application on common web-browsers.</td></tr>
<tr><td>Question</td><td>Would you like such an application to have a simple user interface without too many distracting options and settings?</td></tr>
<tr><td>Aim of Question</td><td>Identification of end user preferences when making design decisions.</td></tr>
<tr><td colspan="2">Findings


With over 95% of the responses being in favor of a simple graphical user interface to interact with the system, a web application with a simple GUI structure is most suitable to demonstrate the core features of the proposed prototype.</td></tr>
</table>

Table 4-4: Requirement Elicitation - Survey

4.5.3 Interview

Codes	Themes	Conclusion
Image Degradations, Visual corruptions	Research Problem	All participants unanimously agreed that the selected research problem is a high-impact area in the domain of computer vision. Further, two of the more experienced participants commended the effort to work on an area of computer vision generally considered to be one of the most complex in the field.
Generative Adversarial Networks, Neural Architecture Search, Image restoration	Research Gap	Using NAS alongside GANs for image restoration was recognized by all interviewees as a highly advanced research gap. They commended the initiative to pursue a project with a heavy research component such as this and highlighted its potential beyond the undergraduate level. However, two of them advised to review existing implementations for both components of the project and to experiment with proven NAS algorithms to avoid unnecessary complications and resource limitations during experimenting and training.
NAS Algorithms, GANs architectures, ML Libraries	Methodology	Due to the inherently resource intensive nature of traditional NAS approaches, it was advised to experiment with memory-efficient NAS algorithms. The definition of the search space and search strategy were highlighted as areas to be experimented with. Since NAS implementations that work with U-net architectures already exist, it was advised to attempt integrating such networks with Pix2Pix or CycleGAN type models which are designed around U-Net architectures. Both PyTorch and Tensorflow were recommended for

		experimenting. SciKit Learn was also indicated as a useful library for this project.
Dataset selection, labelled vs unlabelled data, Evaluation methods	Datasets	All interviewees stressed the importance of using tried and tested datasets for training and testing to ensure the quality of data. Moreover, the usage of such datasets would also assist during evaluation and benchmarking against other available solutions. One participant indicated the importance of finding a dataset with 1:1 matching annotation if experimenting with Pix2Pix GANs. All participants stated that standard evaluation metrics such as PSNR, SSIM and FID (for the GAN model) would be ideal during the testing phase.
User Interface, Application type, data visualizations	Implementation details	Whilst all participants agreed that a web-based UI would be ideal to showcase the proposed system, three of the most senior researchers expressed doubts on the feasibility of developing an end product given the steep learning curve of the research component. However, it was agreed that a simple user interface would be ideal for the task at hand. One participant indicated the integration of Tensorboard with the UI as a potential value addition to the final product.

Table 4-5: Requirement Elicitation – Interviews

- For transcripts of the interview, please refer to **APPENDIX – III**

4.5.4 Prototyping

Prototype Type	Findings
Basic CNN for image denoising	The model is capable of generating a decent output withing a reasonable amount of time. However visual artifacts remain around areas with smooth textures and solid colours.

NAS-DIP with basic U-Net architecture, prior to optimising with NAS.	Quality of outputs for the denoising task were poor, with details being lost mid-way through the training process. Similar results were observed for the super-resolution task. The generated high-resolution image was blurred.
NAS-DIP with basic U-Net based architecture, post NAS optimisation (NAS implementation testing).	The model is capable of generating high-quality outputs. Image denoising takes the least amount of time and resources whilst image inpainting tasks required training for longer.
DeblurGAN (GAN implementation testing)	The model is capable of successfully deblurring images with very high accuracy. However, the GAN training process is difficult and resource/time intensive.
GANSAN Prototype GAN implementation	The U-net architecture of the generator paired with the base PatchGAN discriminator is capable of producing decent results. Training time is not nearly as long as some other GAN implementations. Convergence is achieved at around 800 epochs, but this number could be reduced by optimising further.

Table 4-6: Requirement Elicitation - Prototyping

4.5.5 Summary of Findings

The below table presents the summary of findings made through all the above-mentioned requirement elicitation methods.

ID	Finding	Literature Review	Survey	Interview	Prototyping
1	Validating the research problem and gap	X	X	X	
2	Novelty of the proposed solution and approach (Using NAS with GANs)	X		X	

3	Significance of using existing NAS algorithms to optimize a neural network (due to the cost of time vs reward when training with limited resources).			X	
4	Significance of using an existing GAN architecture as the basis of the image restoration model.			X	
5	Experimenting with different network and model architectures to find best possible fit			X	X
6	Use tried and tested datasets for model training and testing for fair comparison.			X	X
7	Implement a simple GUI which promotes ease of use for general users.		X		
8	Users should be able to restore degraded images without human intervention.		X	X	
9	If possible, include the ability to train the model on custom datasets through the GUI.			X	
10	If possible, include the ability to download the trained model.			X	
11	It would be beneficial to integrate Tensorboard logs into the GUI if the option to train on a custom dataset is provided.			X	X
12	Show evaluation metrics			X	

Table 4-7: Requirement Elicitation - Summary of Findings

4.6 Context Diagram

The context diagram gives an overview into the system's boundaries, interactions with external entities and the data flows between the system and its primary users. As seen below, there are mainly 2 types of users who would typically interact with this system, namely, General Users/Students, Computer Vision Developers/Researchers

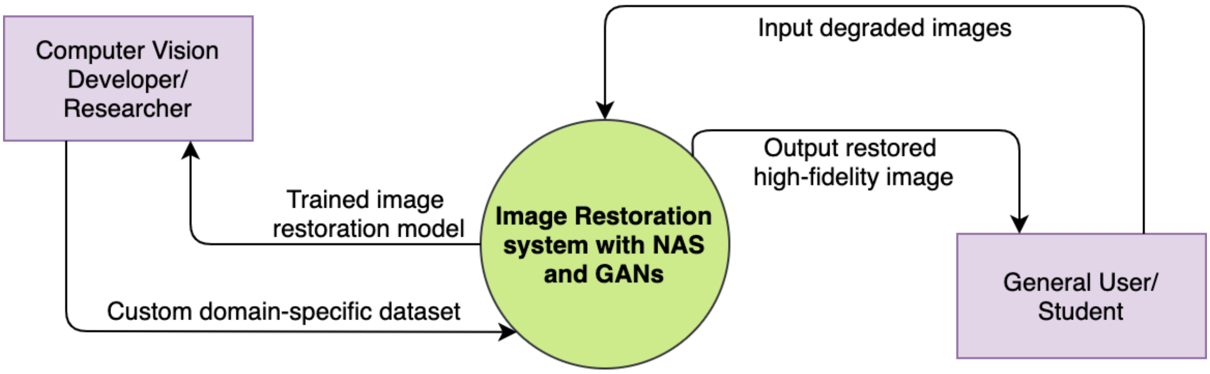


Figure 4-3: Context Diagram (Self-Composed)

4.7 Use Case Diagram

The following diagram illustrates the use-cases bound to the interactions between the system and its most prominent stakeholders. Additionally, relationships between use cases within the system have also been indicated for better understanding.

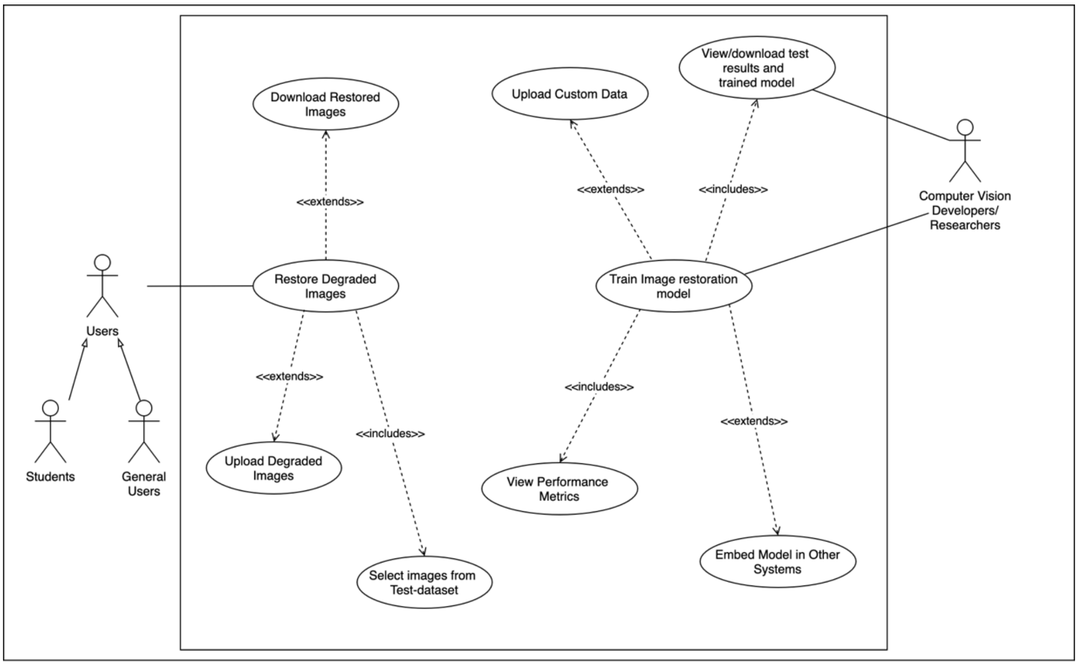


Figure 4-4: Use Case Diagram (Self-Composed)

4.8 Use Case Descriptions

Use Case	**Restore Degraded Images**
Description	This use case describes evaluating the image restoration capabilities of the system.
Participating Actors	General Users
Pre-Conditions	The image restoration model should be accessible long with the degraded image(s) to be restored.

<table>
<tr><td>Extended Use Cases</td><td colspan="2">1. Upload degraded images.
2. Download restored images.</td></tr>
<tr><td>Included Use Cases</td><td colspan="2">Selecting degraded images from the test-dataset</td></tr>
<tr><td rowspan="2">Main Flow</td><td>Actor</td><td>System</td></tr>
<tr><td>1. The user activates the system.
2. Select degraded image(s) to be restored.
4. Perform image restoration using the trained model.</td><td>3. Load the degraded image to the trained model to obtain a restored result.
5. Display the generated result for viewing and/or downloading.
6. Generate and display performance and evaluation metrics.</td></tr>
<tr><td>Alternate Flows</td><td colspan="2">AF1 – If the user requires, they may upload their own degraded images for restoration.</td></tr>
<tr><td>Exceptional Flows</td><td colspan="2">EF1 – Failure to restore the selected image: Display an error message.</td></tr>
<tr><td>Post-Conditions</td><td colspan="2">1. The restored image output is saved to a directory and remains accessible for download.
2. Testing metrics remain available for evaluation purposes.</td></tr>
</table>

Table 4-8: Use Case Description - 1

4.9 Requirements

Identified functional requirements are categorized based on the “MoSCoW” prioritization based on factors such as projected delivery schedules, value addition to the product and stakeholders, and risk factors involved.

Priority Level	Description
(M) – Must have	Mandatory features which form the core of the project and must be implemented.
(S) – Should have	Important features, which even though not mandatory for the functioning of the prototype, would add significant value if implemented.

(C) – Could have	Supplementary features which would add further value to the system but are not mandatory and essential for the functioning of the prototype. Will be implemented based on availability of time.
(W) – Would not have	These requirements are not important for the implementation of the system and would therefore be omitted altogether.

Table 4-9: Requirement Priority Levels

4.9.1 Functional Requirements

FR ID	Requirement Description	Priority Level	Status
FR1	Users must be able to select a degraded image for restoration.	M	Select restoration model
FR2	Users must be able to upload their degraded images.	M	Upload degraded image.
FR3	Users must be able to restore images using the trained model.	M	Perform image restoration
FR4	Users must be able to download the restored images.	M	Downloads restore image.
FR5	Computer vision researchers should be able to train the image restoration model on an existing dataset.	S	Retrain the restoration model
FR6	Computer vision researchers/developers should be able to view performance metrics of the image restoration model.	S	View performance metrics
FR7	Computer vision developers should be able to obtain the test results and the trained model/weights	S	View/Retrieve test results and trained model
FR8	Computer vision developers and researchers could have the option to tweak parameters related to the model prior to training.	C	Train image restoration model
FR9	Computer vision researchers could use their own datasets to train the image restoration model.	C	Upload custom data

FR10	Computer vision researchers could search for an optimized neural network architecture using NAS on an uploaded/available dataset through the GUI.	C	Search for optimal architecture using NAS

Table 4-10: Functional Requirements

4.9.2 Non-Functional Requirements

NFR ID	Requirement Description	Specification	Status
NFR1	Users must be able to restore degraded images using the system within an acceptable timeframe from initiation to final output.	Performance	Important
NFR2	Non-technical users must also be able to restore degraded images.	Usability	Important
NFR3	The codebase of the project should conform to coding best-practices.	Maintainability	Important
NFR4	The system should alert the users if an error occurs while executing any task.	Usability	Desirable
NFR5	Uploading and downloading images/results should be executed efficiently.	Performance	Desirable

Table 4-11: Non-Functional Requirements

4.10 Chapter Summary

This chapter encapsulated the overall process of identifying and gathering requirements for the implementation of the proposed prototype system. Potential stakeholders were initially identified using a rich picture diagram and were filtered and categorized based on the proximity to the system using the Saunder's Onion Model. Next, suitable requirement elicitation instruments were identified, implemented and the gathered findings were presented in a summarized fashion. Subsequently, a context diagram of the system depicting its dataflow as well as a high-level use case diagram pertaining to the interactions between the system and its primary users was presented. Finally, the identified use cases were described in detail and the discovered functional and non-functional requirements of the system were tabulated and presented in-keeping with the "MoSCoW" prioritization method.

5 SOCIAL, LEGAL, ETHICAL AND PROFESSIONAL ISSUES

5.1 Chapter Overview

The aim of this chapter is to identify and address any social, legal, ethical, and professional challenges that may arise during the project, and to outline measures to mitigate those issues.

5.2 Breakdown of SLEP Issues

Social	Legal
• The anonymity of all respondents of the questionnaire has been maintained and all findings have been generalized and presented in a manner devoid of any personal identification. • All interviewees were clearly briefed on the purpose of the interviews and consent was obtained to include their individual responses and designations in the thesis.	• All software tools, frameworks, and programming languages used for the implementation of this project were under open-source license. • All datasets used for this project were obtained under open-source licence.
Ethical	**Professional**
• Participants involved in the requirement elicitation process were clearly briefed on the nature of the project and their contribution through participation. • The thesis contains no instances of fabrication, falsification, or plagiarism. All data, knowledge or facts obtained from other sources have been properly cited and referenced.	• This research was conducted in adherence to both academic and industrial norms and standards. • The outcomes of the project have been presented in an authentic manner, clearly highlighting any limitations present. • The requirements/feedback obtained from the specialists were not altered or tampered with.

Table 5-1: Breakdown of SLEP Issues

5.3 Chapter Summary

This chapter outlined the potential social, legal, ethical, and professional problems relating to the research project alongside the corresponding mitigation strategies followed by the author throughout its duration.

6 DESIGN

6.1 Chapter Overview

This chapter will dissect the design decisions made when creating a suitable architecture for the proposed system based on the requirements gathered in the previous chapter. First, the identified design goals will be tabulated after which, the high-level design, low level design, related design diagrams including a system process flow chart and UI wireframes will be presented with comprehensive reasoning for each design choice.

6.2 Design Goals

Design Goal	Description
Performance	**Research Component:** Since the objective of using NAS is to optimize the image restoration network architecture, it is vital that the results produced by the system are highly performant and efficient compared to industry standards. **Prototype Component:** It is important that the proposed system can restore degraded images efficiently within an acceptable timeframe to maximise user satisfaction.
Adaptability	The system should be designed such that it could be adapted to work with images from different domains. Further, it is also important that the prototype is developed such that it could be used on different platforms based on the resources available to the user.
Usability	Since general users are expected to be able to operate the system with minimal guidance, it is important that the UI is elegant and easy to navigate. The provision of an intuitive and lightweight UX is a priority during implementation.
Scalability	In a production environment, the system should be able to work with large datasets and a diverse range of images/visual degradations.
Correctness	Since the proposed system deals with images, the ability to produce results with minimal errors is of paramount importance. Any shortcomings of the system will be evidently visible in generated outputs, causing the purpose of the image restoration system to be lost.

Table 6-1: Design Goals

6.3 High Level Design

6.3.1 Architecture Diagram

Due to the nature of this project, a tiered model, where the presentation, logic, and data tiers could be physically separated was selected as the system architecture. The following diagram depicts the components and modules contained within each tier. The Presentation tier deals with the UI components with which the end users interact, Data tier deals with storing and managing system and user data, whilst the Logic tier bridges the above two and facilitates the functionalities of the system.

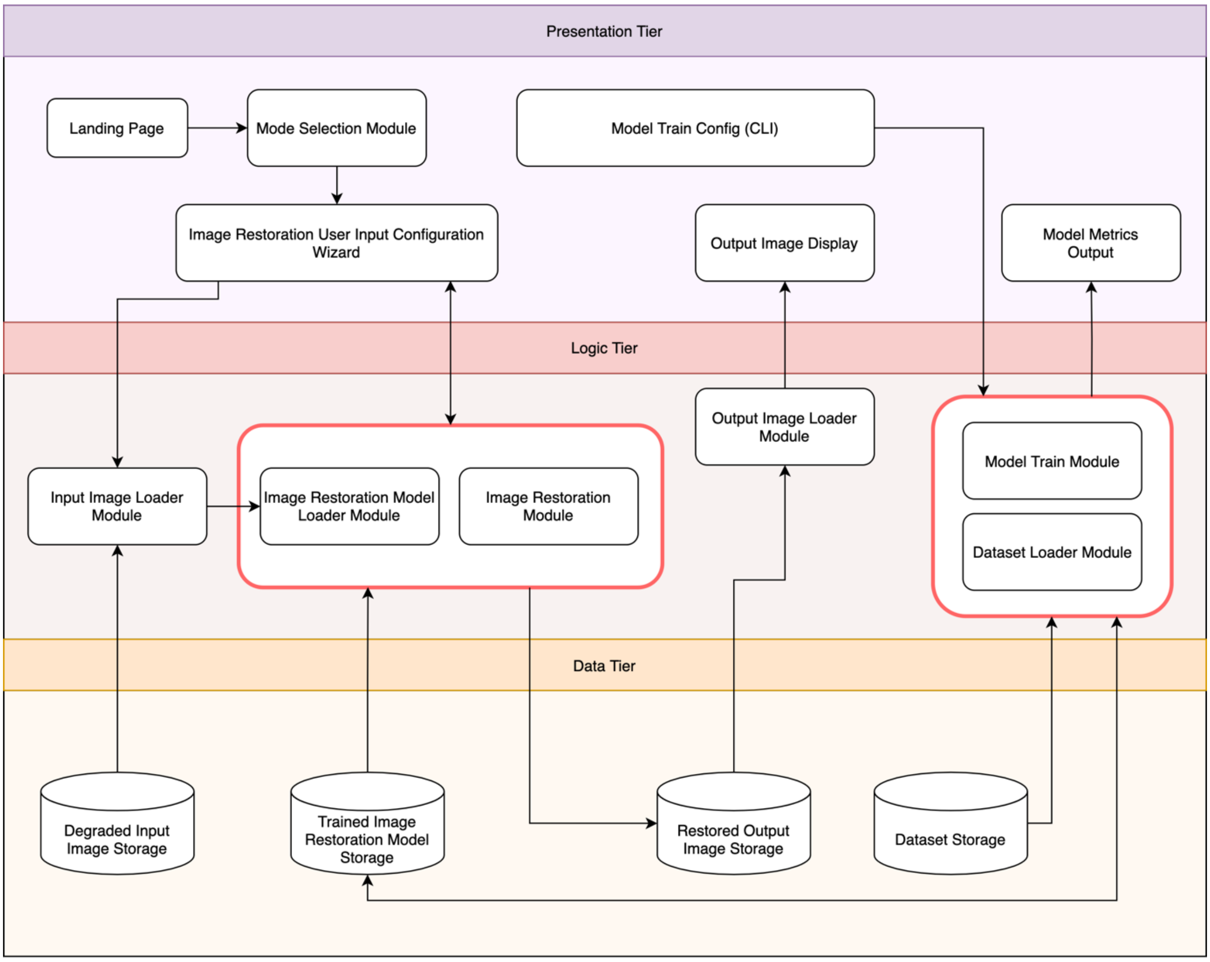


Figure 6-1: High Level Architecture (Self-Composed)

6.3.2 Discussion of Tiers

Data Tier

- **Degraded Input Image Storage** – This module acts as a storage for pre-loaded degraded images which could be selected by the user to test the image restoration capabilities of the system.

- **Trained Image Restoration Model Storage –** This module allows for the storage of the trained optimised image restoration models architected using the approach proposed in this project.
- **Restored Output Image Storage –** This module serves as a temporary storage for images restored using the proposed system, to feed them through to the output image loader module.
- **Dataset Storage –** This module serves as the storage for image datasets used to train the model.

Logic Tier

- **Input Image Loader Module –** The purpose of this module is to load images uploaded by the user or images selected from the degraded input image storage, into the image restoration system.
- **Image Restoration Model Loader Module –** This loads the required image restoration module based on the degradation type selected by the user.
- **Image Restoration Module –** This module executes the image restoration process using a saved restoration model, retrieved from the Trained Image Restoration Model Storage.
- **Dataset Loader Module –** This module loads a training dataset into the model, from the dataset storage or user uploads according to the selected configuration.
- **Model Train Module –** This module executes the model training process when triggered by the user via the CLI.
- **Output Image Loader Module –** Images restored and saved to the temporary storage are retrieved and loaded into the system prior to being displayed on the final output page.

Presentation Tier

- **Landing Page –** This will serve as an introduction to the application and will guide the user to try out the different functionalities of the system.
- **Mode Selection –** This allows the user to select the degradation type they wish to restore.
- **Image Restoration User Input Configuration Wizard –** This component will allow the user to configure the system to restore degraded images (self-uploaded or pre-loaded system images).

- **Model Train Configuration** – The process of configuring datasets and training parameters for model retraining has been made easy by including a common config file.
- **Output Image Display** – This component displays the output image next to the input image for the user to assess the perceptual improvement and download the image.

6.4 Low Level Design

6.4.1 Choice of Design Paradigm

A software design paradigm consists of the general philosophy and approach to designing and developing any software system. Since it provides the basic framework and guidance to structure, organize, and maintain the entire project, it is essential to select an appropriate design paradigm for our specific use-case. The two most commonly used design paradigms are **OOAD** (Object Oriented Analysis and Design) and **SSADM** (Structured Systems Analysis and Design Method).

Due to the complex nature of this research project where rapid changes and improvements must be made based on continuous experimentations to achieve a pre-defined goal, **SSADM** was selected as the design paradigm of choice. Additionally, since Python is used as the main programming language, it is not essential to map the system's design to real-world objects.

6.5 Design Diagrams

6.5.1 Component Diagram

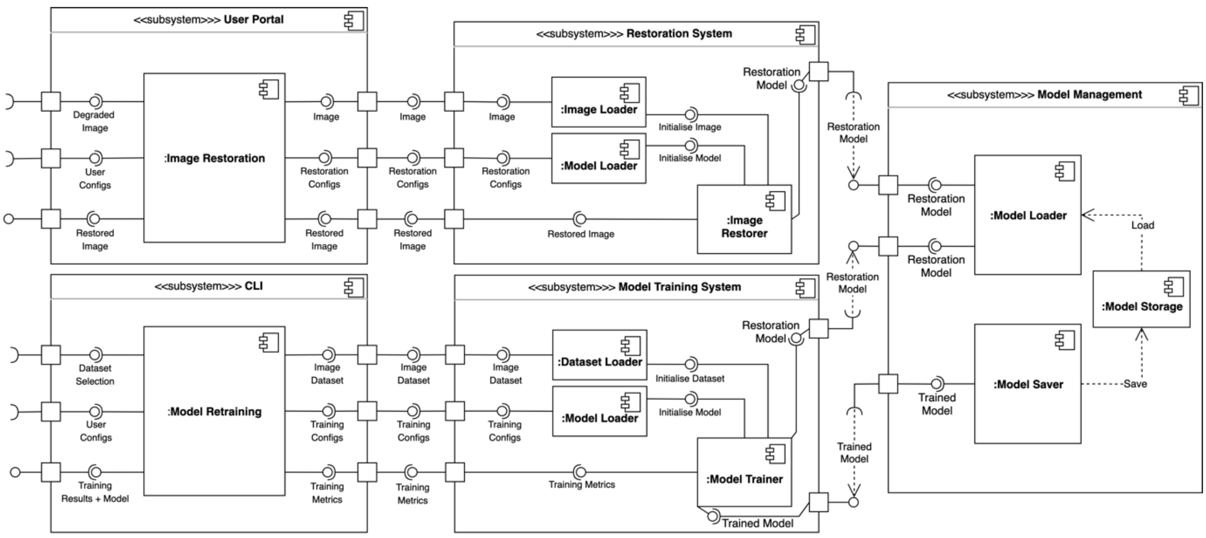


Figure 6-2: Component Diagram (Self-Composed)

* For a full-page landscape view of the Component Diagram, please refer to **APPENDIX – V**

6.5.2 Data Flow Diagram

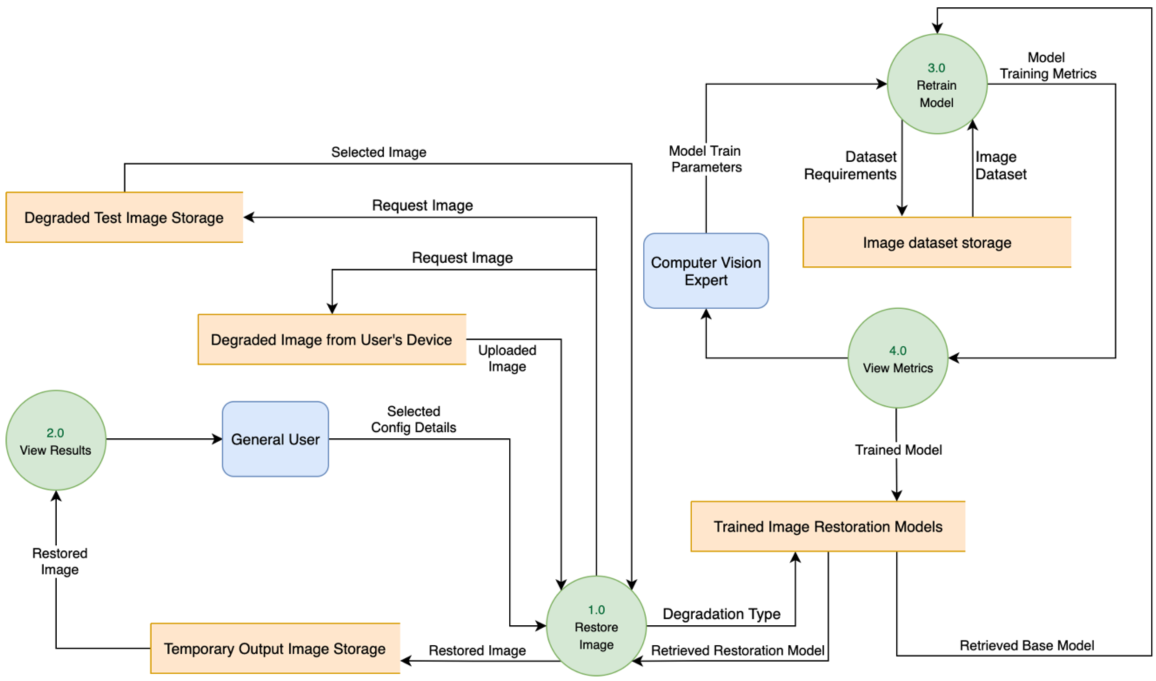


Figure 6-3: Level 1 Data Flow Diagram (Self-Composed)

Depicted through the above data-flow diagram (**Level 1 DFD**) is the relational data flow between various components of the system and external interactors.

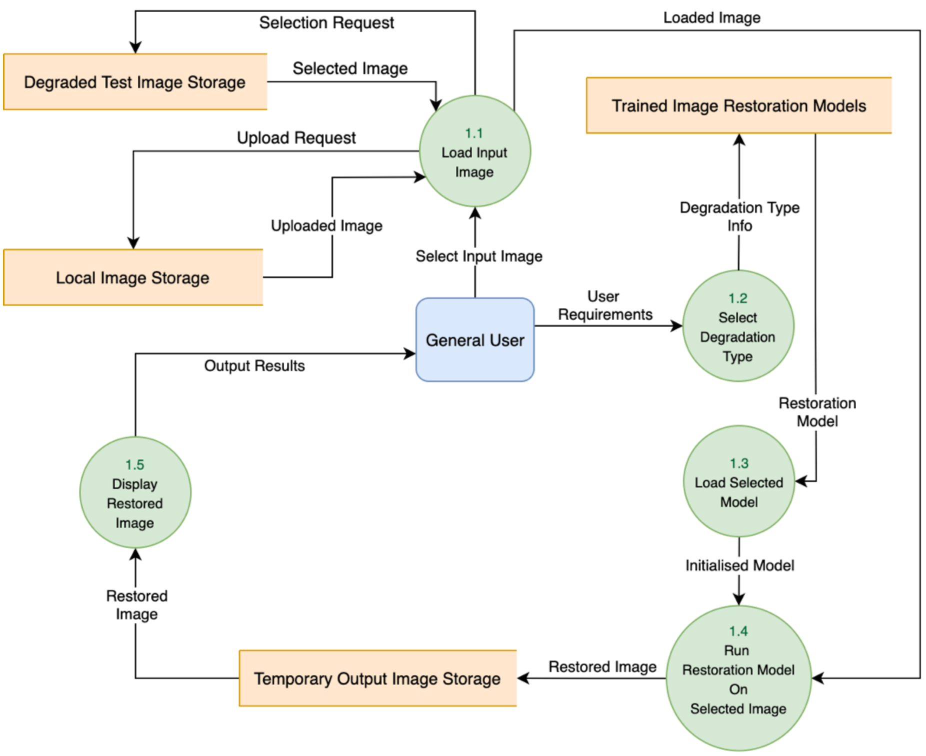


Figure 6-4: Level 2 Data Flow Diagram (Self-Composed)

The above data flow diagram (**Level 2 DFD**) represents an extensive breakdown of the data flow between system components and general users.

6.5.3 System Process Activity Diagram

The below system process activity diagram represents the overall workflow of the prototype system, including inputs, outputs, decision operations, and other processes. It assumes the point of view of a general user of the prototype application.

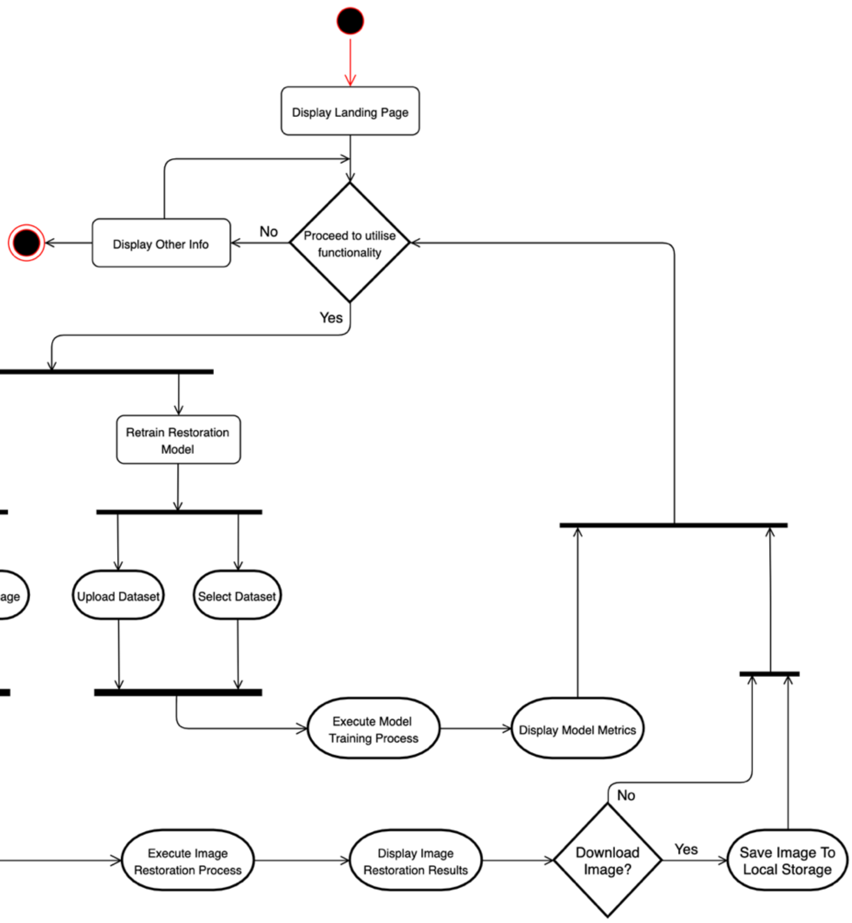


Figure 6-5: Activity Diagram (Self-Composed)

6.6 User Interface Design

- Please refer to **APPENDIX - VI** for wireframes of the proposed UI.

6.7 Chapter Summary

This chapter thoroughly explained the system's design and methodologies used before implementing the proposed solution. The design goals were identified and high-level and low-level architectural components were structured and depicted with relevant design diagrams. The chosen software design paradigm was justified, and all aspects of the system's design were presented through supporting diagrams for its components, data flow, process flow, and user interfaces. The next chapter will discuss system implementation details based on the design outline.

7 IMPLEMENTATION

7.1 Chapter Overview

This chapter documents in detail, all aspects of the implementation process of the designed prototype. Based on the system architecture created with the knowledge gathered through the literature review and requirement elicitation phases, suitable technologies and development frameworks have been selected. Sufficient justification for all technology selections have been provided along with code snippets supporting implementation details of the core functionality of the prototype.

7.2 Technology Selection

7.2.1 Technology Stack

The following figure represents the technologies selected to for the implementation and functionality of the designed 3-tier architecture.

Figure 7-1: Technology Stack (Self-Composed)

7.2.2 Dataset Selection

Based on the knowledge gathered through the literature review and the requirement elicitation process, a mix of generalized and specialised datasets have been selected for this project.

Dataset	Type	Purpose
Div2k	Generalized, high-res images	Used for image restoration GAN and NAS training. Data pre-processing and augmentation methods could be used to generate dataset variants with degradations for training.
GoPro	Blurred and high-res image pairs	Used for image restoration model training. Since image pairs are available, artificial addition of degradations is unnecessary
BSD/CBSD	Generalized images	Used to train/test restoration models. Degradations could be added using data augmentation and pre-processing techniques to generate variants as required.
Other Kaggle datasets	Miscellaneous/Images	Since the aim of the project is to generalize the capabilities of the image restoration model, experiments could be conducted with multiple image datasets from reputed sources such as Kaggle.

Table 7-1: Dataset Selection

7.2.3 Programming Languages

The following table depicts the programming languages selected for the implementation of different components of this project along with supporting reasoning.

Programming Language	Reasoning
Python	Python was chosen as the main programming language to build the data-science/deep-learning component of this project due to the extensive availability of supporting libraries, tooling, and community support.
JavaScript	JavaScript was selected as the main front-end scripting language due to its adoption as the industry standard in building responsive, dynamic web applications.

Table 7-2: Programming Language Selection

In addition to the above, HTML and CSS will also be utilized to build and style front-end components of the application.

7.2.4 Development Frameworks

With Python being selected as the main development language, Flask and Django presented themselves as the go to web-development frameworks.

Both contenders were put subjected to a comparative analysis weighing in their individual pros and cons with respect to the nature of this project. **Flask** was selected as the development framework of choice after careful consideration of the rationale summarized in the table below.

Framework	Rationale
Flask	Being a light-weight Python-based micro-framework with no predefined tooling or libraries, Flask is extremely lightweight, flexible, and considerably easier to learn. Additionally, Flask is faster than Django since it has fewer abstraction layers.

Table 7-3: Development Framework Selection

7.2.5 Libraries/Toolkits

Based on the language and development framework selections made for this project, the following libraries were selected to aid in the implementation of the deep learning models and application features.

Library/Toolkit	Rationale
Torch/PyTorch	Torch has been selected since it works alongside PyTorch (which acts as an extensive framework) when building deep learning models for various domains including computer vision.
TensorFlow	Used during early stages of development and prototyping to test image restoration models and approaches.
NumPy	NumPy was selected to address the requirements related to building mathematical and algorithmic functions.
Matplotlib	Commonly used to create data visualisations in python, Matplotlib was selected to display image data and plot testing and evaluation results of the deep learning models.
OpenCV	OpenCV is one of the most popular libraries used for computer vision and machine learning projects. OpenCV was selected to manage the image data used in this project.

Scikit-Learn	These libraries were selected to pre-process and manage image data used in this project. Additionally, Scikit-Image would is used when calculating PSNR and SSIM scores of images when evaluating the performance of the developed model.
Scikit-Image	
Pillow	
Albumentations	
Bootstrap	Being one of the most popular HTML, CSS and JS libraries, Bootstrap was selected for this project to help build a responsive and dynamic web application. Its community support and extensibility with other 3rd party libraries would greatly aid in the rapid development of the application's front end.

Table 7-4: Library Selection

7.2.6 IDEs

The following IDE's and code editors were selected for the implementation of this project.

IDE/Code Editor	Justification
Google Colab	Colab was selected for initial implementation and testing since it offers cloud resources and a pre-built environment allowing for rapid setup and execution of ML models.
Jupyter Notebook	Jupyter Notebook was chosen since it allows to run components as code-blocks and its overall versatility. Since a local GPU-machine was used for a bulk of the development process, Notebooks proved to be an ideal overall fit.
PyCharm	The extensive tooling and debugging features of PyCharm would allow to build and test components of the system with greater ease.
VS-Code	VS-Code was selected since it functions as a versatile development platform capable of working with a multitude of different languages and frameworks. Furthermore, the extensive 3rd part library support offered is useful when debugging and testing various components of the project.

Table 7-5: IDE Selection

7.2.7 Summary of Technology Selection

All the above technological selections could be summarized as follows:

Component	Tool(s)/Technology(ies)
Programming Language	Python, JavaScript
Development Framework	Flask

Deep Learning Library	Pytorch
UI Framework/Library	Bootstrap, SCSS, AJAX
Other Libraries	TensorFlow, NumPy, Matplotlib, OpenCV, Scikit-Learn, Scikit-image, Pillow, Albumentations
IDEs	Google Colab, Jupyter Notebook, PyCharm, VS-Code
Version Control	Git, GitHub

Table 7-6: Summary of Technological Selections

7.3 Implementation of Core Functionality

7.3.1 Discriminator Architecture

Whilst a standard PatchGAN architecture was used early on, it was later found that the addition of self-attention layers to the discriminator boosts its ability to better differentiate between real and fake images received through the network. As such, the following Self Attention module was written to modify the base discriminator network.

```python
class SelfAttention(nn.Module):
    def __init__(self, in_channels):
        super(SelfAttention, self).__init__()

        # Define query, key and value conv layers
        self.query_conv = nn.Conv2d(in_channels, in_channels//8, kernel_size=1)
        self.key_conv = nn.Conv2d(in_channels, in_channels//8, kernel_size=1)
        self.value_conv = nn.Conv2d(in_channels, in_channels, kernel_size=1)

        self.gamma = nn.Parameter(torch.zeros(1))

        self.softmax = nn.Softmax(dim=-1)

    def forward(self, x):
        batch_size, C, height, width = x.size()
        # Compute query and key tensors
        proj_query = self.query_conv(x).view(batch_size, -1, height*width).permute(0,2,1)
        proj_key = self.key_conv(x).view(batch_size, -1, height*width)

        # Compute attention weights using Softmax
        energy = torch.bmm(proj_query, proj_key)
        attention = self.softmax(energy)

        proj_value = self.value_conv(x).view(batch_size, -1, height*width)

        # Compute output tensor
        out = torch.bmm(proj_value, attention.permute(0,2,1))
        out = out.view(batch_size, C, height, width)

        # Add scaling parameter and add to input feature map
        out = self.gamma*out + x
        return out
```

Figure 7-2: Self-Attention Block

```python
class CNNBlock(nn.Module):
    def __init__(self, in_channels, out_channels, stride=2):
        super().__init__()
        self.conv = nn.Sequential(
            nn.Conv2d(in_channels, out_channels, 4, stride, bias=False, padding=1, padding_mode="reflect"),
            nn.InstanceNorm2d(out_channels, affine=True),
            nn.LeakyReLU(0.2)
        )
        # Add self attention layer
        self.attn = SelfAttention(out_channels)

    def forward(self, x):
        x = self.conv(x)
        x = self.attn(x)
        return x
```

Figure 7-3: Discriminator CNN Block with Self Attention

```python
class Discriminator(nn.Module):
    def __init__(self, in_channels=3, features=[64, 128, 256, 512]):
        super().__init__()
        self.initial = nn.Sequential(
            nn.Conv2d(in_channels*2, features[0], kernel_size=4, stride=2, padding=1, padding_mode="reflect"),
            nn.LeakyReLU(0.2),
        )
        # Iteratively define and add conv layers
        layers = []
        in_channels = features[0]
        for feature in features[1:]:
            layers.append(
                CNNBlock(in_channels, feature, stride=1 if feature == features[-1] else 2),
            )
            in_channels = feature

        layers.append(
            nn.Conv2d(
                in_channels, 1, kernel_size=4, stride=1, padding=1, padding_mode="reflect"
            ),
        )

        self.model = nn.Sequential(*layers)
```

Figure 7-4: Model Discriminator

7.3.2 Generator Architecture

During preliminary experimentations conducted to optimise the generator architecture, the author noticed that the results yielded by NAS based approaches (see Model Testing) did not justify the extended training times/complexity.

Therefore, a baseline U-Net like network architecture based on initial tests was adapted and tweaked with the help of auto hyperparameter optimisation to create a suitable and efficient generator architecture.

```python
class Generator(nn.Module):
    def __init__(self, in_channels=3, features=64):
        super().__init__()
        self.initial_down = nn.Sequential(
            nn.Conv2d(in_channels, features, 4, 2, 1, padding_mode="reflect"),
            nn.LeakyReLU(0.2),
        )
        # Downsampling blocks
        self.down1 = Block(features, features * 2, down=True, act="leaky", use_dropout=False)
        self.down2 = Block(…
        self.down3 = Block(…
        self.down4 = Block(…
        self.down5 = Block(…
        self.down6 = Block(
            features * 8, features * 8, down=True, act="leaky", use_dropout=False
        )
        self.bottleneck = nn.Sequential(
            nn.Conv2d(features * 8, features * 8, 4, 2, 1), nn.ReLU()
        )
        # Upsampling blocks
        self.up1 = Block(features * 8, features * 8, down=False, act="relu", use_dropout=True)
        self.up2 = Block(…
        self.up3 = Block(…
        self.up4 = Block(…
        self.up5 = Block(…
        self.up6 = Block(…
        self.up7 = Block(features * 2 * 2, features, down=False, act="relu", use_dropout=False)
        self.final_up = nn.Sequential(
            nn.ConvTranspose2d(features * 2, in_channels, kernel_size=4, stride=2, padding=1),
            nn.Tanh(),
        )
```

Figure 7-5: Model Generator

Initial tests showed that image artifacts appeared at random in restored image samples. To avoid this, multiple trials were conducted with revisions to the network architectures and Batch normalisation was replaced with Instance normalisation in both the generator and discriminator.

```python
class Block(nn.Module):
    def __init__(self, in_channels, out_channels, down=True, act="relu", use_dropout=False):
        super(Block, self).__init__()
        self.conv = nn.Sequential(
            nn.Conv2d(in_channels, out_channels, 4, 2, 1, bias=False, padding_mode="reflect")
            if down
            else nn.ConvTranspose2d(in_channels, out_channels, 4, 2, 1, bias=False),
            nn.InstanceNorm2d(out_channels, affine=True),
            nn.ReLU() if act == "relu" else nn.LeakyReLU(0.2),
        )

        self.use_dropout = use_dropout
        self.dropout = nn.Dropout(0.5)
        self.down = down
```

Figure 7-6: Generator Block Module

Finally, training is carried out for 500 epochs while saving checkpoints every 5 epochs.

```python
    for epoch in range(config.NUM_EPOCHS):
        train_fn(
            disc, gen, train_loader, opt_disc, opt_gen, L1_LOSS, BCE, g_scaler, d_scaler,
        )

        if config.SAVE_MODEL and epoch % 5 == 0:
            save_checkpoint(gen, opt_gen, filename=config.CHECKPOINT_GEN)
            save_checkpoint(disc, opt_disc, filename=config.CHECKPOINT_DISC)

        save_some_examples(gen, val_loader, epoch, folder="evaluation")
```

Figure 7-7: Training for defined number of epochs

- Please refer to **APPENDIX X** for model configs and dataset manipulations scripts used to synthesize the initial image denoising dataset based on the popular DIV2K dataset.

7.4 User Interface

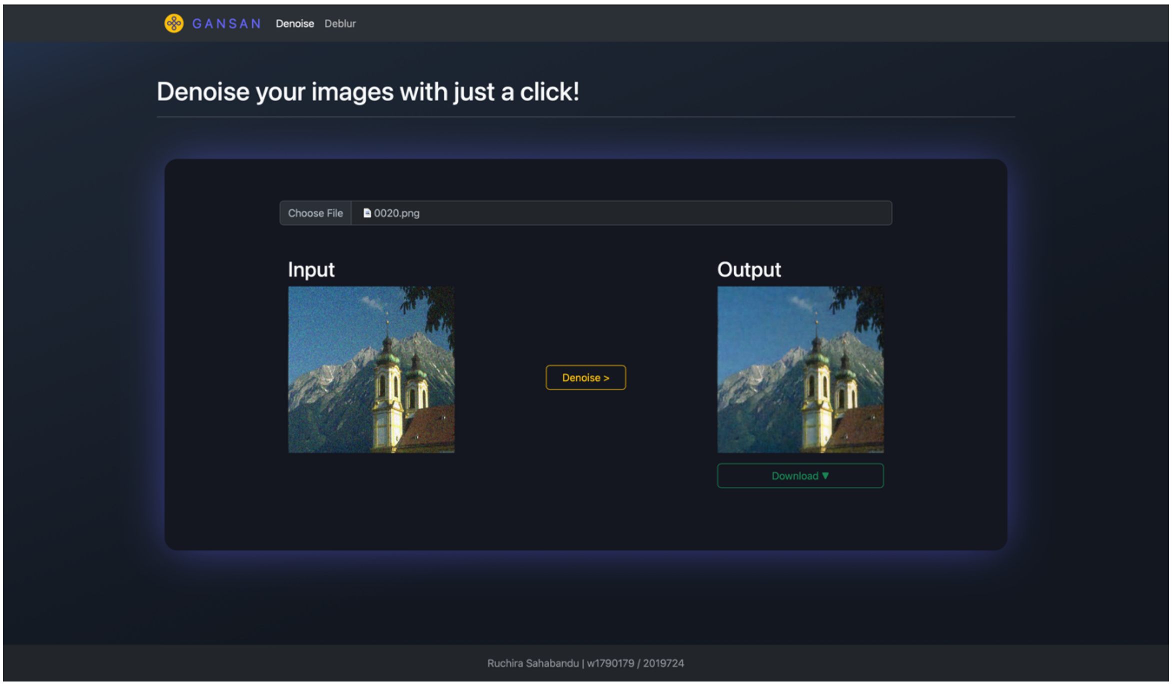


Figure 7-8: Image Denoising Dashboard

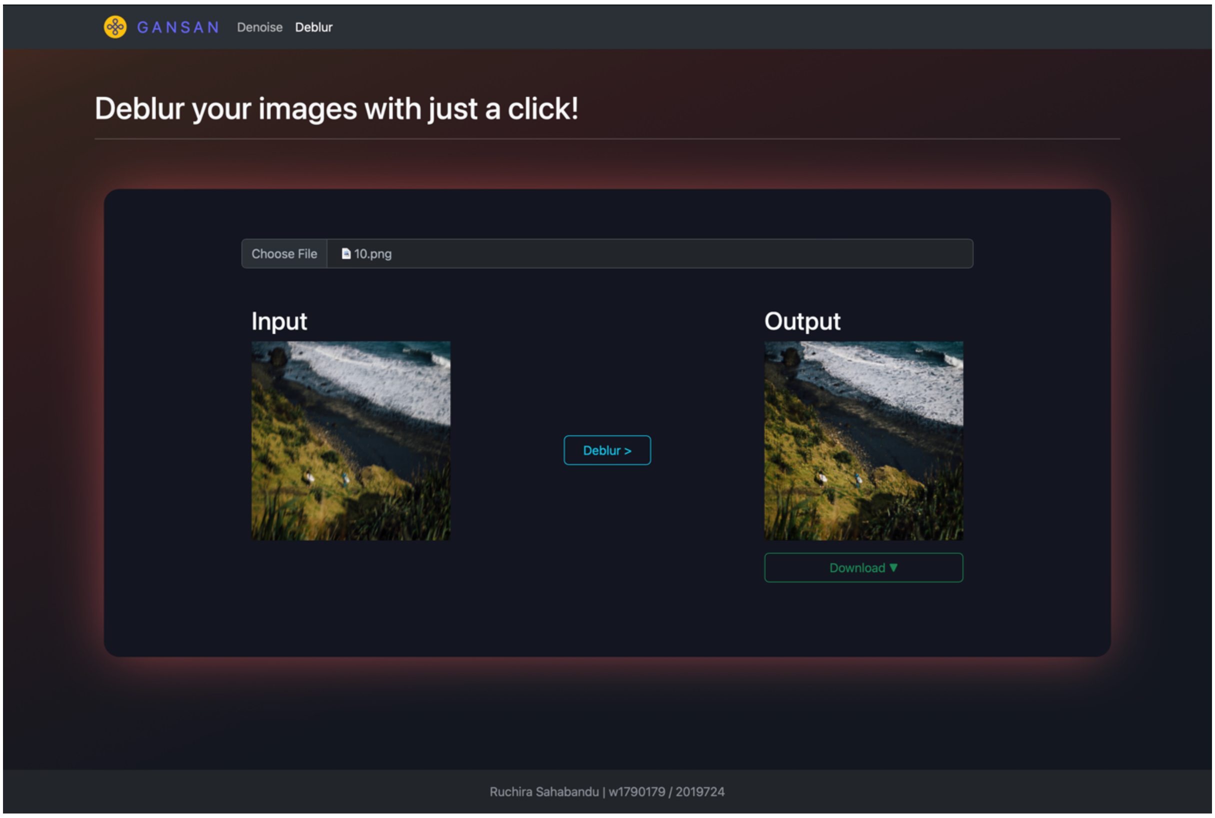


Figure 7-9: Image Super-Res Deblurring Dashboard

7.5 Chapter Summary

This chapter provided a detailed discussion on the specifics involved in the implementation of the proposed prototype system based on the formulated design goals discussed in the preceding chapter. The selection of datasets, development frameworks, programming languages and libraries were justified with supporting reasoning. Code snippets related to the implementation of the core components of the prototype including the novel PatchGAN based discriminator and optimised U-net like generator were presented for further clarity and understanding of the reader. The subsequent chapter will wrap up this document with details on the progress of the project thus far and final concluding remarks.

8 TESTING

8.1 Chapter Overview

The aim of this chapter is to provide a comprehensive understanding of the testing process followed and its significance in ensuring the quality, reliability, and efficiency of the developed prototype. Various aspects of the testing process, such as testing objectives, model testing, benchmarking, functional and non-functional testing, module and integration testing, and limitations of the testing process will be covered in this section.

8.2 Objectives and Goals of Testing

The primary purpose of testing a software product is to assess if the developed system performs as intended and is capable of achieving predefined expectations, based on the gathered requirements.

As such, the main goals, and objectives of GANSAN's testing process are listed below:

- Validating that GANSAN's image restoration models perform as expected and are refined to produce the best possible outcome.
- Identification of means to benchmark the performance of the developed system against similar existing works.
- Verifying if the system satisfies the identified mandatory functional requirements.
- Verifying if the system satisfies the identified important non-functional requirements.
- Verifying that the developed system does not contain any major bugs or defects that may affect its functions or performance.
- Validating that the various components of the system work together, seamlessly, as expected.

8.3 Testing Criteria

Based on the specified system requirements and solution design, the following two testing paradigms were selected to assess the developed prototype application.

Testing Paradigm	Description
Functional Testing	This indicates tests carried out on the system's functional requirements to verify that they have fulfilled the expectations.

Structural Testing	This refers to validating design and code structures to ensure conformity with academic and industrial best practices.

Table 8-1: Testing Criteria

8.4 Model Testing

To evaluate the performance and effectiveness of the designed GAN model, it was trained and tested using a couple of datasets (and dataset combinations) for both designated restoration tasks, i.e., denoising and super-res deblurring. Training details of the selected models are as follows:

1. **Denoising Model**

   Training Details:

   - **Dataset:** Div2k (Train) σ 35 (with brightness shift) + Div2k (Train) σ 25.
   - **Epochs / Training time:** 500 / 2h 46 mins.

2. **SR Deblurring Model**

   Training Details:

   - **Dataset:** Div2k (Train) 4x unknown-deg + Div2k (Train) 2x unknown-deg
   - **Epochs / Training time:** 500 / 3h 22 mins.

Testing was conducted using the initial network architectures found through NAS experiments as well as the latter models tweaked and optimised by the author.

Based on the testing procedures followed in the reviewed literature, it was evident that **PSNR** and **SSIM** scores would be the ideal metrics of choice to test and validate the performance of the produced restoration models.

For each test-set, the above metrics were retrieved for the **best-performing image**, **best average batch,** and **the entire test set**.

8.4.1 Image Denoising

For validation of performance and generalisability of a denoising model, it is important to test its effectiveness against images relating to different domains, with varying noise and brightness levels included in their degradations. To streamline this process with provisions for benchmarking against existing work, the following datasets were selected as the test-sets of choice.

- Div2k (Test) – σ 35
- CBSD68 – σ 25
- CBSD68 – σ 35 (with brightness shift)

The Div2k test set contains 100 test images with a nominal synthetic gaussian noise level of σ 35, whilst both CBSD68 datasets, which have been used as popular denoising benchmarks throughout existing literature, contain 68 test images each, with nominal Additive White Gaussian Noise (AWGN) intensities of σ 25 and σ 35 respectively.

8.4.1.1 Test Results

Results observed for all test sets are as given below.

- For each metric, the **left column** indicates values of the **better performant model tweaked by the author** whilst the right column indicates values of the lesser performant network found through initial NAS experiments.

Div2k Test Results				
Test Type	**PSNR (dB)**		**SSIM**	
Single Image (Best)	**33.37**	29.17	**0.935**	0.885
Average (Batch of 8)	**30.17**	27.66	**0.883**	0.820
Average (Complete Test Set)	**29.01**	24.52	**0.879**	0.809

Table 8-2: Denoise Test Results - Div2k

CBSD68 – σ 25				
Test Type	**PSNR (dB)**		**SSIM**	
Single Image (Best)	**31.98**	27.12	**0.937**	0.878
Average (Batch of 8)	**31.04**	26.95	**0.894**	0.866
Average (Complete Test Set)	**30.58**	25.49	**0.885**	0.845

Table 8-3: Denoise Test Results - CBSD68 (Sigma 25)

CBSD68 – σ 35 (with brightness shift)				
Test Type	**PSNR (dB)**		**SSIM**	
Single Image (Best)	**31.96**	27.24	**0.931**	0.870
Average (Batch of 8)	**30.92**	26.11	**0.891**	0.855
Average (Complete Test Set)	**30.51**	25.53	**0.872**	0.837

Table 8-4: Denoise Test Results - CBSD68 (Sigma 35 with Brightness Shift)

As observed above, both PSNR and SSIM metrics show promising scores on the tweaked model for all test cases, with a slight drop in values as the testing batch-sizes are increased.

The below graphs depict the variation of scores for each metric based on the selected test datasets for the better performing model.

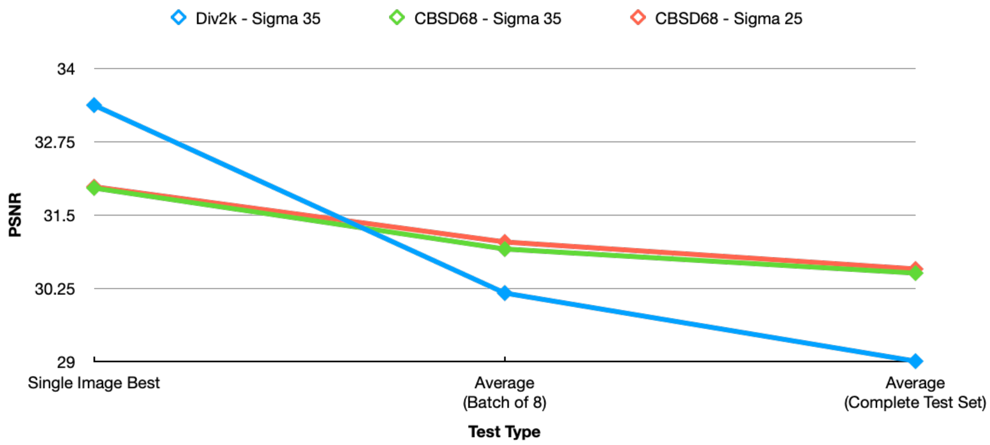


Figure 8-1: Comparison of PSNR Scores for Image Denoising

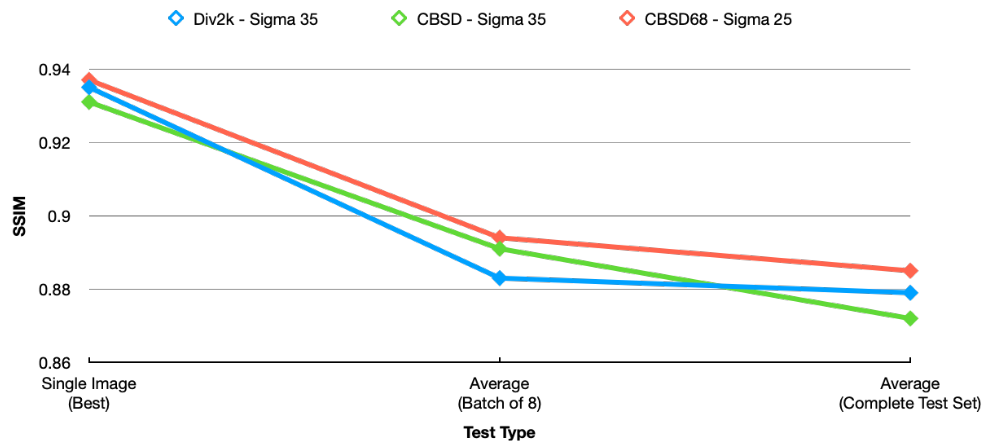


Figure 8-2: Comparison of SSIM Scores for Image Denoising

It must be noted that the PSNR scores between the two noisy CBSD68 datasets are evenly matched, which indicates the generalisability of the model for varying noise and brightness levels between comparable scenarios.

8.4.2 Image Super-res Deblurring

Similar to the testing processes followed for image denoising, the following 3 datasets were used to test the performance of the super-resolution deblurring model whilst maintaining provisions to benchmark against similar works described in the available literature.

- Div2k (Test) – 4x unknown deg.
- Set5 – 4x
- Set14 – 4x

The Div2k test set contains 100 images whilst Set5 and Set14 datasets contain 5 and 14 images respectively. All 3 datasets are commonly used as popular benchmarks for super-res deblurring of images.

8.4.2.1 Test Results

Results observed for all test sets are as given below.

Div2K Test results - 4x Unknown				
Test Type	**PSNR (dB)**		**SSIM**	
Single Image (Best)	**35.10**	31.11	**0.932**	0.865
Average (Batch of 2)	**32.57**	28.90	**0.912**	0.844
Average (Complete Test Set)	**30.60**	26.42	**0.891**	0.821

Table 8-5: SR-Deblur Test Results - Div2k (4x Unknown deg.)

Set5 – 4x				
Test Type	**PSNR (dB)**		**SSIM**	
Single Image (Best)	**34.85**	30.90	**0.943**	0.895
Average (Batch of 2)	**33.26**	28.50	**0.910**	0.872
Average (Complete Test Set)	**32.19**	26.16	**0.894**	0.858

Table 8-6: SR-Deblur Test Results - Set5 (4x)

Set14 – 4x				
Test Type	**PSNR (dB)**		**SSIM**	
Single Image (Best)	**32.08**	29.10	**0.851**	0.799
Average (Batch of 2)	**31.22**	27.32	**0.826**	0.744
Average (Complete Test Set)	**30.66**	27.01	**0.791**	0.698

Table 8-7: SR-Deblur Test Results - Set14 (4x)

The below graphs depict the variation of scores for each metric based on the selected test datasets for the better performing model (tweaked by the author).

Figure 8-3: Comparison of PSNR Scores for Image SR-Deblurring

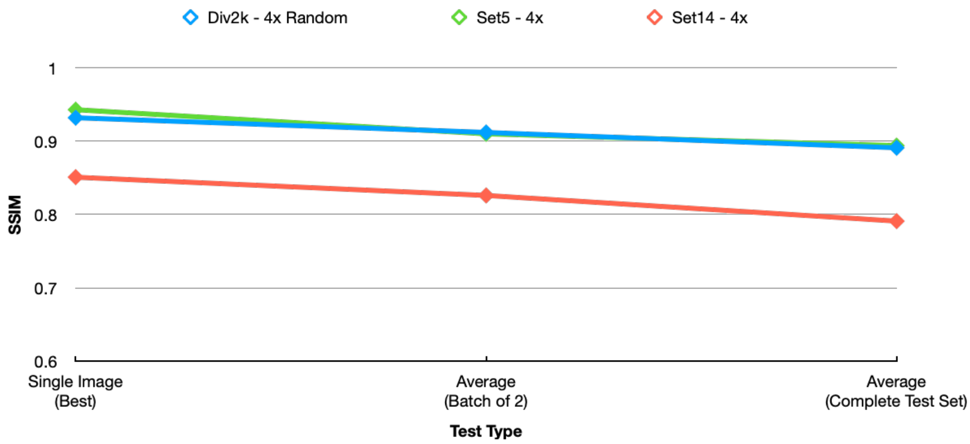


Figure 8-4: Comparison of SSIM Scores for Image SR-Deblurring

The decline in SSIM scores as the batch-sizes of test images are increased from a single image to the entire test set is minimal, indicating the consistency in performance of the restoration model.

8.5 Benchmarking

Once the higher performant finalised models were selected for both image restoration tasks, their performance metrics on standard benchmarking datasets were collected and compared against state-of-the art existing works to establish performance baselines and benchmarks for both image denoising and super-res deblurring tasks. ***For both PSNR and SSIM scores, higher is better.***

8.5.1 Benchmarking for Denoising

Work	PSNR (dB)	SSIM
CBSD68 – σ 25		
GANSAN (Ours)	**30.58**	**0.885**
(Ghahremani et al., 2022)	31.78	-
(Aharon and Ben-Artzi, 2022)	31.41	0.890
(Zhang, Zuo and Zhang, 2018)	31.21	-
(Claus and van Gemert, 2019)	30.99	-
(El Helou and Susstrunk, 2020)	30.76	-
CBSD68 – σ 35		
GANSAN (Ours)	**30.51**	**0.872**
(Ghahremani et al., 2022)	30.24	-
(Aharon and Ben-Artzi, 2022)	29.8	0.850

(Zhang, Zuo and Zhang, 2018)	29.58	-
(Claus and van Gemert, 2019)	29.34	-
(El Helou and Susstrunk, 2020)	28.81	-

Table 8-8: Benchmarks for Image Denoising

As seen above, the proposed system has performed better than comparable works on the CBSD68 – σ 35 dataset as opposed to the CBSD68 – σ 25 dataset which indicates that the tested model generalises better for higher noise levels.

8.5.2 Benchmarking for SR-Deblurring

Work	PSNR (dB)	SSIM
Div2k – 4x (Unknown degradations)		
GANSAN (Ours)	**30.06**	**0.891**
CAR(Sun and Chen, 2020)	32.82	0.884
(Vassilo et al., 2020)	28.08	0.814
(Park, Moon and Cho, 2023)	27.69	0.793
Set5 – 4x		
GANSAN (Ours)	**32.19**	**0.894**
(Anwar and Barnes, 2019)	32.74	0.901
(Zhang and Li, no date)	32.73	0.901
(Ledig et al., 2017)	32.05	0.902
(Chen et al., 2020)	31.09	-
Set14 – 4x		
GANSAN (Ours)	**30.66**	**0.791**
(Anwar and Barnes, 2019)	29.02	0.791
(Zhang and Li, no date)	28.98	0.791
(Ledig et al., 2017)	28.49	0.818
(Chen et al., 2020)	28.37	-

Table 8-9: Benchmarks for Image SR-Deblurring

It could be seen that the proposed model has achieved competitive performance compared to other comparable models, especially on the Set14 dataset where both PSNR and SSIM scores achieved are best-in-class.

8.6 Functional Testing

The Black-Box testing methodology was utilised to test the system against the functional requirements established in chapter 4.

Test Case	FR ID	User Action	Expected Outcome	Actual Outcome	Result Status
1	FR1	User clicks on File input field.	The system opens the file manager for image selection.	The system opens the file manager for image selection.	Passed
2	FR2	User selects an image for upload.	The selected image is uploaded to the system and displayed in the input-image box.	The selected image is uploaded to the system and displayed in the input-image box.	Passed
3	FR3	User clicks on the image restoration button.	The image is passed to the relevant restoration model (based on the selected degradation) and the output is displayed in the output-image box.	The image is passed to the relevant restoration model (based on the selected degradation) and the output is displayed in the output-image box.	Passed
4	FR4	The user clicks on the download-image button.	The restored image is downloaded onto the users' device.	The restored image is downloaded onto the users' device.	Passed
5	FR5	Computer vision researcher/developer selects paired-image dataset to retrain the model.	They must be able to train the model on the selected dataset.	Researchers/Developers could retrain the model by updating configs, using the command line.	Passed
6	FR6	Computer vision researcher/developer	PSNR/SSIM scores for the restored	PSNR/SSIM scores for the restored image/batch of	Passed

		triggers PSNR/SSIM evaluation scripts.	image/batch of images is displayed.	images is displayed via the command line.	
7	FR7	Computer vision researcher/developer triggers model retraining on a selected paired-image dataset.	Model weights are saved automatically and could be used for testing on test images.	Model weights are saved automatically, and testing could be triggered by updating the config script.	Passed
8	FR8	Computer vision researcher/developer uploads their own paired-image dataset to the train/test folders for training.	Model trains as expected without requiring further tweaks.	Model trains as expected without requiring further tweaks as long as the dataset conforms to the expected format.	Passed

Table 8-10: Functional Testing

8.7 Module and Integration Testing

This section deals with testing out the functionality of each module of the application as a complete system.

Please refer to **APPENDIX - VII** for module integration test results.

8.8 Non-Functional Testing

Non-functional requirements of the prototype application were tested based on 3 key areas, namely, performance, usability, and maintainability, which were identified during the requirement elicitation process. This section will mainly focus on performance since both usability and maintainability of the system have been and discussed in the subsequent Evaluation chapter.

8.8.1 Model development performance

Since image restoration is known to be one of the most resource heavy computer vision tasks requiring powerful GPUs, the entire model development and optimisation process was executed on a local desktop machine equipped with an Nvidia RTX 3060 (12GB) with 32GB of system RAM. The below screenshots depict the resource usage of the system during training.

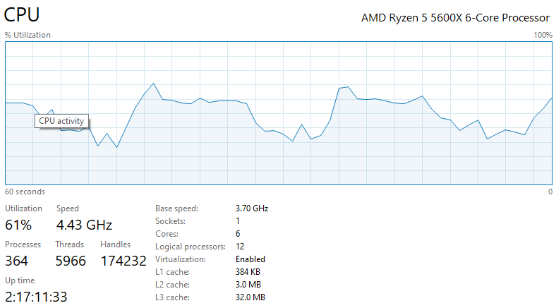


Figure 8-6: CPU Usage During Training

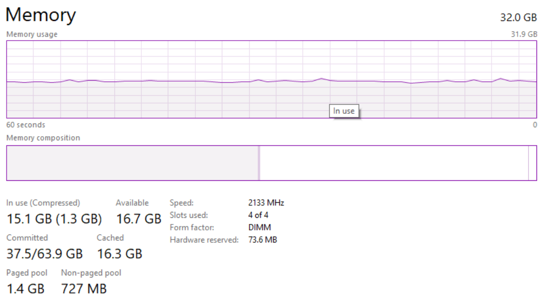


Figure 8-6: Memory Usage During Training

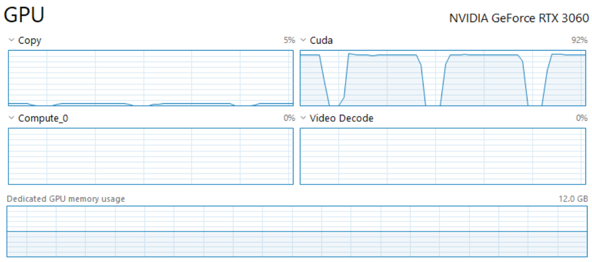


Figure 8-7: GPU Usage During Training

It must be noted that the resource consumption during the training and optimization stage is comparable and even more efficient than those of many existing works which had been trained and tested on large GPU servers. Training times for both models were under 3h 30mins each.

8.8.2 Product performance

The performance of the developed system was tested by measuring the resource usage on a MacBook equipped with an Intel i9 processor, AMD Radeon Pro 5500M (4GB), and 16GB of system RAM.

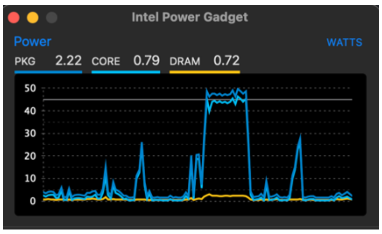


Figure 8-8: Power Utilisation During Product Testing

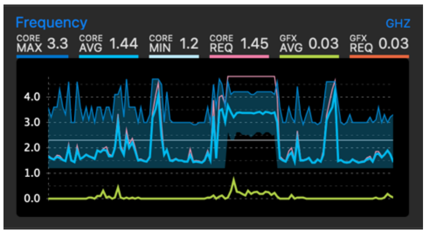


Figure 8-9: Processor Frequency During Product Testing

It must be noted that a couple of other resource-hungry processes such as web browsers and word processors were running simultaneously during the testing process. This proves the effectiveness and general usability of the developed system within resource constrained environments which lack access to CUDA cores traditionally required for complex deep-learning processes.

8.8.3 Usability and Maintainability Testing

- Coding best-practices and industry/academic standards were adhered to throughout the entire development process, along with the appropriate use of versioning tools such as Git/GitHub for ease of maintainability.
- The MVP (Minimum Viable Product) was developed with an intuitive user interface to enable any user with basic knowledge on operating a PC to utilise the functionalities of the system.

Please refer to the Evaluation Results section in the subsequent chapter (Evaluations) for feedback received regarding these two non-functional requirements.

8.9 Limitations of the Testing Process

Most limitations during the testing were experienced while carrying out the benchmarking process. There are two main reasons for this:

1. Lack of model files and/or compatible datasets.

It was found that certain existing works did not provide public access to their codebases and/or pre-trained model files. This proved to be problematic for practical benchmarking since the author was unable to test out the competing approaches themselves, requiring the use of the provided performance metric figures for comparison. Since certain testing datasets mentioned in existing works did not conform to the paired-image structure used in this research, the author was forced to omit them altogether.

2. Lack of comparable GPU resources.

Certain state of the art models mentioned in existing work had been trained using multiple high-performance GPUs for multiple hours. Due to the resource and time constraints applied to this project, the author was unable to conduct fair and meaningful comparisons against such works.

8.10 Chapter Summary

This chapter has presented a detailed examination of the testing process and findings made from the research project. In particular, this section has provided an overview of the testing objectives, criteria, and outcomes for both the main research component and the functional and non-functional requirements of the prototype. Furthermore, limitations faced by the author during the testing process have also been discussed.

9 EVALUATION

9.1 Chapter Overview

Once the prototype had been designed, implemented, and optimised through multiple tests and experiments, it was evaluated based on the requirements collected in the SRS chapter. This chapter details the evaluation process followed, including specifics of the self-evaluation process, and assessments by experts in the technical domain, academia, and the industry in general.

9.2 Evaluation Methodology and Approach

Since this research lies within the domain of computer vision and deals with image restoration, both quantitative and qualitative data were utilized to evaluate the findings. While the previous chapter discussed the results obtained through quantitative evaluations, to gather the qualitative evaluation data, the author has selected interviews and surveys by distributing a questionnaire. Finally, the questionnaire and interview responses have been analysed thematically and presented in a descriptive manner.

9.3 Evaluation Criteria

The following table displays the criteria selected to perform qualitative evaluations on the overall research project, both through means of self-assessments and structured interviews/surveys conducted with industrial and academic experts. These points of reference have been based on themes observed throughout interviews conducted during the requirement elicitation phase.

ID	Criterion	Evaluation Purpose
EC1	Overall concept and choice of research topic	To validate the overall concept of the selected research topic with respect to developments made in the domain, existing work and identified gaps.
EC2	Significance and depth of the work undertaken	To verify the impact, significance and depth of the work undertaken as result of the selected research topic and identified research gap, with respect to the author's level of expertise (undergraduate).

EC3	Quality of research process followed and literature review	To verify that the literature review conducted, and research process followed is of substantial quality with adequate coverage of existing work, state of the art technologies and approaches, and evaluation methods.
EC4	Development approach followed	To validate the quality of the approach followed during the development phase with respect to academic and industry standards, norms, and best practices.
EC5	Demonstrated test results and testing approach	To validate the suitability of the selected performance metrics and results obtained thereof, through the developed prototype system.
EC6	Quality, usability and UI/UX of the MVP	To confirm that the end product developed allows for clear and convenient demonstration of the proposed solution and promotes ease-of-use for end users.

Table 9-1: Evaluation Criteria

9.4 Self-Evaluation

The author's self-evaluation of their own work based on the evaluation criteria discussed above has been presented in the table below.

ID	Self-Evaluation by the Author
EC1	Image restoration is a core area of research within the domain of computer vision, that holds a position of high-impact towards other image-processing tasks as well due to the negative effects caused by degradations found in input data. The research topic selected by the author focuses on optimising the generator network architecture of a GAN model used for image restoration, using an AutoML based approach which includes NAS and AutoHPO. Based on the literature review conducted, this concept presents itself as a novel approach towards tackling the process of architecting effective image restoration GAN models and opens up new pathways towards future research on effective and efficient image restoration systems.
EC2	Since this project requires a thorough understanding of imaging systems, image restoration systems, including traditional and deep learning approaches, as well as a good grasp on recent state-of-the-art developments in the selected domain, (on both conceptual and practical levels), the work undertaken by the author has a significant depth and level of difficulty, especially at an undergraduate level of

	study. Furthermore, since this project entails the completion of a thorough literature review along with designing, developing, and testing a working prototype, complete with an end-application, the amount of work undertaken could also be stated as significant.
EC3	An in-depth literature review has been conducted with wide coverage of the selected problem and research domains, existing work, technologies used for existing solutions as well as those selected as candidates for the proposed solution, along with the most effective and popular evaluation metrics and approaches used in recent years. Additionally, the research document (project thesis) composed is a high-quality piece of work, complete with detailed explanations and diagrams relating to each component of the project.
EC4	The author has followed a standard iterative development approach starting from a sample proof-of concept, which has then progressed towards a prototype implementation of the core functionality, and finally a minimum-viable-product. This allowed the author to successfully overcome the initial challenge posed by the performance deficit of the network searched through early NAS experiments. Academic and industrial best practices have been followed by breaking down the solution into multiple separate modules, neatly organised into a clean folder structure, whilst maximising code readability and reusability.
EC5	Standard image quality assessment metrics popularly used in the domain (PSNR/SSIM scores) have been utilised to test the performance of the system and quality of the produced outputs. Whilst the initial hypothesis of this research hinged on a NAS-based approach towards architecting the proposed optimised GAN model, the model designed tweaked by the author subsequent to the preliminary NAS experiments proved to perform better for both restoration tasks.
EC6	The minimum viable product developed as the final output of this research consists of a minimal and intuitive UI through which users may test the performance of the image restoration system. It meets all requirements of providing a simple interface to interact with the developed restoration model.

Table 9-2: Self Evaluation

9.5 Selection of Evaluators

The categories of evaluators selected to obtain feedback for the project could be broken down as follows:

Category ID	Category
1	Experts in the domain of computer vision, image signal processing and deep learning/machine learning.
2	Experts in research and development from related technical domains.
3	Experts in the field of software engineering.

Table 9-3: Selection of Evaluators

9.6 Evaluation Results

9.6.1 Qualitative Analysis

Once feedback was gathered from experts (See **APPENDIX – VIII** for evaluator details) belonging to the categories mentioned above, a thematic analysis was conducted on the findings (See **APPENDIX – IX**) to determine recurring themes upon which evaluations and comments had been provided.

Criterion	Theme	Summary of Evaluations
Overall concept and choice of research topic	Research topic	All evaluators have stated that the selected research domain is a highly focused and trending area of research at this moment. A majority has confirmed that the choice of research topic is commendable due to the growth in general demand for efficient solutions in this space and the real-world applicability of such systems.
	Overall concept	The overall concept has been recognized as challenging and novel, with the potential for significant contributions as well as an opportunity for the author to greatly further their knowledge.
Significance and depth of the work undertaken	Significance of the undertaking	A majority of the evaluators have agreed that the work undertaken by the author is of great significance due to the complexity and nature of the selected research topic. Once again, the real-world significance of this research has been substantiated.
	Depth of the research work	All evaluators have stated that depth of the work undertaken and completed by the author is substantial and praiseworthy, especially considering that this

		research project has been carried out by a single individual at an undergraduate level of study.
Quality of research process followed and literature review	Quality of the LR	Almost all evaluators have stated that the literature review conducted, including the compiled documentation conforms to high standards. The in-depth coverage and critical analysis of prominent existing works, technologies, and evaluation methods has been pointed out as noteworthy. In particular, one expert has stated that the produced literature review is a “publishable piece of work”.
	Research process followed	All experts have expressed their satisfaction and approval of the research process followed by the author, describing it as “methodical” and “systematic”. The evaluators have recognized the effort exerted by the author towards the successful completion of this project as significant.
Development approach followed	Deep Learning Component	Deep learning experts have stated that the author has successfully identified the requirements of the deep learning component of the project and the solution has been implemented in an efficient and effective manner.
	General Development	Almost all evaluators have agreed that the author has followed a standard development approach on par with existing research work whilst adhering to best coding best practices followed in the industry. The effective usage of existing libraries to streamline the development process and maintenance of a clean folder and code structure has been commended. It has been stated that the author has done a considerable amount of work that is excellent for an undergraduate level of study.
	Output results	A majority of the evaluators have stated that a marked improvement in the visual quality of the input images

Demonstrated test results and testing approach		has been achieved by the developed system. Therefore, the research has been deemed to be successful in terms of achieving the defined objectives. One evaluator has suggested making some improvements to the system to maintain brightness levels between the inputs and outputs. The author has also been encouraged to further elaborate on the performance of the system using quantitative metrics.
	Benchmarking	Two evaluators have encouraged the author to conduct more tests and benchmark the solution against similar work.
Quality, usability and UI/UX of the MVP	Overall UI/UX	The author has been complimented on the overall UI/UX, highlighting the attractive, minimalist and intuitive nature of the created web application.
	Functionalities of the MVP	Almost all evaluators have stated that the final application achieves the goals of an MVP through the provided feature set. One evaluator has suggested the addition of additional features to expand the capabilities of the MVP as a marketable product.

Table 9-4: Evaluation Results – (Qualitative)

9.6.2 Quantitative Analysis

The following graph demonstrates the quantitative feedback provided on the overall research project:

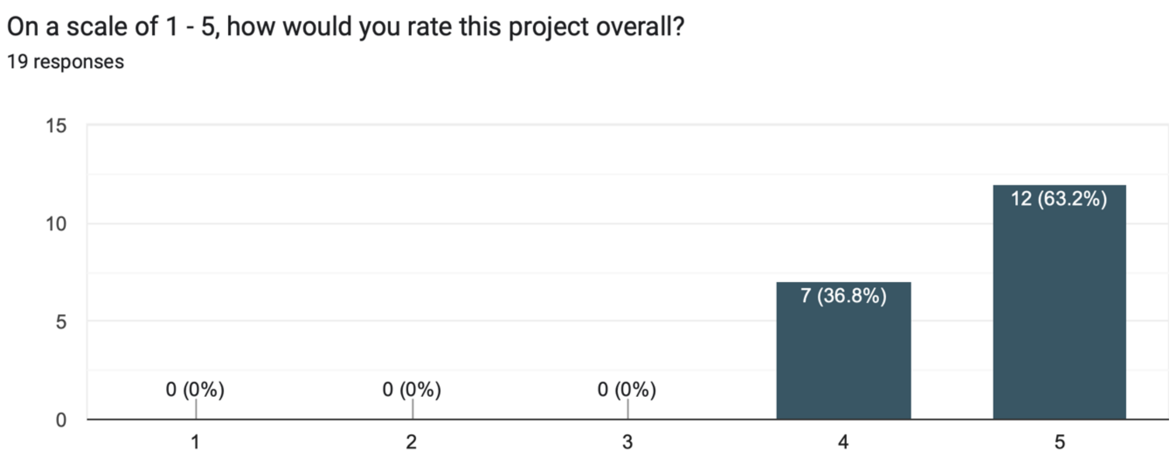


Figure 9-1: Evaluation Results – (Quantitative)

9.7 Limitations of Evaluation

The most significant limitation faced during the evaluation phase was the lack of reachable experts and researchers in the domain of image restoration. Furthermore, all available industrial and academic experts had busy schedules which did not allow for an opportunity for face-to-face interviews to collect feedback. A separate detailed evaluation document had to be created and shared along with a short demo video of the developed MVP. Evaluations had to be collected via a structured google-form. General research and industry experts were also consulted in order to obtain the required amount of feedback.

9.8 Evaluation of Functional Requirements

FR ID	Requirement Description	MoSCoW Priority Level	Evaluation
FR1	Users must be able to select a degraded image for restoration.	M	Implemented
FR2	Users must be able to upload their own degraded images.	M	Implemented
FR3	Users must be able to restore images using the trained model.	M	Implemented
FR4	Users must be able to download the restored images.	M	Implemented
FR5	Computer vision researchers should be able to train the image restoration model on an existing dataset.	S	Implemented
FR6	Computer vision researchers/developers should be able to view performance metrics of the image restoration model.	S	Implemented
FR7	Computer vision developers should be able to obtain the test results and the trained model/weights	S	Implemented
FR8	Computer vision researchers could use their own datasets to train the image restoration model.	C	Implemented

FR9	Computer vision developers and researchers could have the option to tweak parameters related to the model prior to training.	C	Not-Implemented
FR10	Computer vision researchers could search for an optimized neural network architecture using NAS on an uploaded/available dataset using the GUI.	C	Not-Implemented
Functional requirement completion rate = 8/10 * 100 = **80%**			

Table 9-5: Evaluation of Functional Requirements

9.9 Evaluation of Non-Functional Requirements

NFR ID	Description	Priority Level	Evaluation
NFR1	Users must be able to restore degraded images using the system within an acceptable timeframe from initiation to final output.	Important	Implemented
NFR2	Non-technical users must also be able to restore degraded images.	Important	Implemented
NFR3	The codebase of the project should conform to coding best-practices.	Important	Implemented
NFR4	The system should alert the users if an error occurs while executing any task.	Desirable	Implemented-partially
NFR5	Uploading and downloading images/results should be executed efficiently.	Desirable	Implemented
Non-Functional requirement completion rate = 5/5 * 100 = **100%**			

Table 9-6: Evaluation of Non-Functional Requirements

9.10 Chapter Summary

The focus of this chapter was on evaluating the research work carried out during this research project. The criteria considered for evaluation were explained, along with supporting reasoning for their selection. Details of the self-evaluation process followed by the author have also been provided along with the feedback received through external evaluators regrading various aspects of this project. Ultimately, the functional and non-functional requirements were assessed.

10 CONCLUSION

10.1 Chapter Overview

As the concluding chapter of this document, this section discusses the final observations regarding this project. Starting with an assessment of the author's achievement of the research aims and objectives, this chapter will discuss the contribution of the knowledge gained throughout the course program and the author's own self-learning initiatives, towards the successful completion of the research. Furthermore, it will delve into how well the expected learning outcomes have been met, along with a detailed summary of the various deviations, limitations, and potential future enhancements related to this work. Finally, a brief review on the project's contribution to the body of knowledge will be presented prior to wrapping up the document with the author's concluding remarks.

10.2 Achievement of Research Aims and Objectives

This research project aims to design, develop, and evaluate an efficient and optimized image restoration Generative Adversarial Network architecture by experimenting with Neural Architecture Search, to efficiently restore degraded images (E.g.: Deblurring, Super-resolving, Denoising).

The aim of this research has been achieved successfully by experimenting with NAS and AutoHPO, and introducing a novel discriminator network with self-attention along with an efficient generator network for the selected base GAN architecture, as demonstrated through the evidence provided in the testing and evaluation chapters. The effectiveness of the created model has been showcased by applying it towards 2 image restoration tasks, namely, image denoising and sr-deblurring.

10.3 Utilisation of Knowledge from the Course

Module	Description
Programming Principles I and II, Object Oriented Programming.	These two modules could be said to have laid the foundation for the core programming and software design skills used by the author for the successful completion of this project. The theoretical knowledge and practical hands-on experience provided through the assigned coursework had established the groundwork for the overall development process followed here.

Web Design and Development, Server-side Web Development	The knowledge and experience gained through these modules proved especially beneficial in designing and developing the MVP web-application as the practical skills gained in front-end technologies, such as HTML, CSS, and JS, and data communication frameworks like AJAX were directly utilized during its development phase.
Software Development Group Project	This module provided the author with their initial experience in working in the domain of data science and machine learning. Coincidentally, it was also the author's first experience in working on a full-scale end-to-end project with a complex tech-stack, from problem identification, through design, development and testing along with the composition of a complete project documentation.
Applied AI	This module offered the author valuable insights and expertise in constructing and testing machine and deep learning models. The experience in designing/architecting neural networks obtained through this module proved to be beneficial towards this research.
Algorithms, Theory, Design, and Implementation	The knowledge gained through this module was instrumental in understanding and working with various algorithms, data structures and mathematical models involved in the core research component as well as testing components of this project.

Table 10-1: Utilisation of Knowledge from the Course

10.4 Use of Existing Skills

- **ML/DL Skills –** Knowledge gathered through self-learning ventures with the help of online resources such as Udemy, Coursera, YouTube and Deeplearning.AI was instrumental in the synthesis and successful completion of this project.
- **Efficient development and presentation skills –** The experience gained in key areas such as development, documentation, and presentation, through working as a Software Engineering intern at WSO2 during the placement year proved to be greatly beneficial in all areas throughout the duration of this research project.

10.5 Use of New Skills

- **Image-to-Image translation and Generative models –** The author was not previously familiar with image generative models, including GANs and Autoencoders. However, due to the project's requirements, it was essential to investigate and understand how such models performed when applied to in image-restoration tasks.

- **AutoHPO/NAS approaches to building neural networks –** The domain of AutoML in general was completely new to the author at the commencement of this research. To understand methodologies followed in existing work, discover gaps in the body of knowledge and conceptualise a solution approach, gathering skills in this area was required.

10.6 Achievement of Learning Outcomes

Learnings	LOs
The author gained new skills and understanding on conducting meaningful research, starting from selecting a problem domain, to defining a workable scope, gathering the required domain knowledge, and synthesizing a suitable solution.	L01 L02 L03
The author was able to obtain a hands-on experience on conducting a critical in-depth literature review, which happens to be the most important component of any research. A sizeable knowledge base on image restoration, computer vision, and deep-learning models in general was gathered through this process.	L04 L05
The author was given the opportunity to expand their knowledge and understanding on methods used for formal data-gathering whilst prioritising social, legal, ethical, and professional aspects.	L03 L06
The author acquired the necessary skills to synthesise and execute a workable solution approach to tackle the defined research problem. General skills related to self-learning and iterative development were gained in addition to those specific to computer vision and deep learning domains.	L05 L07
The author was able to produce a comprehensive and coherent document with end-to-end coverage of the entire research project, complete with detailed explanations, diagrams, and an extensive, critical literature review, deemed to be a high-quality, publishable piece of work by domain experts.	L08

Table 10-2: Achievement of Learning Outcomes

10.7 Problems and Challenges Faced

Problem/Challenge Faced	Solution

Complex and multi-faceted nature of the research domain.	The selected research topic primarily revolved around the field of image restoration, a subsect within the domain of Computer Vision, and also included elements of image generation, image-to-image translation as well as novel autoML techniques. To conquer this challenge, the author was compelled to conduct continuous, thorough research on all these areas, throughout the duration of the project.
Broad scope of the selected research topic.	Due to the inherently complex nature of the select research topic and the broad implications in synthesizing a generalised solution applicable for multiple image restoration tasks, the project scope was carefully narrowed down to a specified range, practically workable within the available time constraints.
Steep learning curve of the research domain and novelty of the technologies involved.	Due to the novel and complex nature of the technologies involved in this area of research and the author's lack of prior experience working in this domain, multiple experiments were conducted using solutions proposed in existing work, whilst conducting the literature review. Furthermore, development of the core research component of this project was commenced immediately after completing the requirement elicitation process.
Extended training times and the requirement for multiple experiments.	The generator tuning process as well as the overall training process for image restoration GAN models proved to be highly time-consuming. To combat this, the image size used for training was reduced from 512x512 to 256x256. Additionally, the training and testing process was limited to two image restoration tasks in order to focus on achieving better results.
High resource requirements.	Due to the inherently resource hungry nature of generative networks, the architecture optimisation and model training processes required powerful hardware. A desktop machine with a rather powerful GPU was acquired to meet this demand.
Limited availability of suitable paired-image datasets.	Since naturally occurring degraded and high-quality image datasets are rare, the author was compelled to limit the scope of the project to very specific image restoration tasks. A noisy-non noisy paired image dataset was created based on the Div2K dataset using synthetic image noise, following methods described in existing literature.

Table 10-3: Problems and Challenges Faced

10.8 Deviations

Whilst the general direction and aim of the project was maintained throughout its duration, the author was compelled to make the following deviation in the solution approach:

- Due to time, resource, and dataset constraints, and the level of expertise required to integrate a NAS process into the GAN training process being beyond the manageable scope, the author was compelled to experiment with NAS approaches separately using a regular U-net as the base network as a preliminary step in the experimentation process. However, given the less-than-expected quality of the output results and the computational complexity incurred, the author subsequently experimented with a novel discriminator architecture and the integration of auto-hyperparameter optimisation (a comparable autoML alternative) into the model training process in search of further performance improvements.

10.9 Limitations of the Research

The main limitations of this research are as follows:

1. The project has been scoped down to only accommodate the restoration of images degraded with gaussian noise and low-resolution blur, mostly due to the lack of high-quality paired image datasets required for training.
2. The trained GAN models are limited to working with an image resolution of 256x256. This may result in a minor loss of detail in the output due to the impact of the degradation on image details being exacerbated in the input.
3. The models exhibit limited performance when dealing with severely degraded images (belonging to the types of degradations mentioned in the first limitation).
4. Due to time and GPU resource constraints, the developed system has not been tested/benchmarked on certain large-scale datasets used in other state-of-the-art works.

10.10 Future Enhancements

Considering the limitations mentioned above and the fresh avenues of research opened up through this project within the domain of image restoration, numerous potential areas of improvement for the future could be deliberated.

- The real-world applicability of the models could be improved by developing an approach to avoid image resizing, allowing persistence of resolution and aspect ratio.

- Research could be conducted on further generalising the model to be capable of dealing with any type or intensity of image noise (or other selected degradation).
- The base models could be tweaked and further trained to be capable of performing other image restoration tasks as well (eg: image inpainting, colour restoration).
- Improvements could be made to the model to enable multiple successful restorations to be performed on a single degraded image (eg: denoise and then deblur an image).
- Integrate the NAS process into the GAN training process to help increase performance in finding better fitting network architectures.
- Improve performance of self-attention mechanism in the discriminator to reduce training time and GPU memory load. This could also be adopted towards the generator.

10.11 Achievement of the Contribution to the Body of Knowledge

The author has managed to deliver on the pre-set research goals by successfully designing and developing an efficient image restoration GAN model using a generator architecture optimised by experimenting with, NAS and AutoHPO approaches, plus, a novel discriminator with self-attention based on the standard PatchGAN architecture. This development helps open up new pathways in architecting efficient and effective generative models within the sub-domain of image restoration as well as other related domains based on image-to-image translations.

10.12 Concluding Remarks

This research project on image restoration has been a significant undertaking involving extensive research and experimentation towards overcoming a range of novel challenges in the domain of computer vision, and deep learning in general. As indicated in this chapter, the vast range of skills and knowledge imparted through the course program along with the author's own dedication, commitment, and self-motivation towards producing a high-quality piece of work has aided in achieving the pre-defined research objectives and overall project success. The positive feedback received from both industry and academic experts is a testament to this, highlighting the relevance of the final research findings and its potential towards future impact in this domain. In conclusion, this project has been a tremendously challenging yet rewarding journey that has provided the author with fresh perspectives on the importance of innovative research towards the exploration of untapped potential of AI and computer science at large.

End of document.

REFERENCES

Agustsson, E. and Timofte, R. (2017). NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. July 2017. Honolulu, HI, USA: IEEE, 1122–1131. Available from https://doi.org/10.1109/CVPRW.2017.150 [Accessed 31 October 2022].

Aharon, S. and Ben-Artzi, G. (2022). Hypernetwork-Based Adaptive Image Restoration. Available from https://doi.org/10.48550/ARXIV.2206.05970 [Accessed 2 May 2023].

Aloysius, N. and Geetha, M. (2017). A review on deep convolutional neural networks. *2017 International Conference on Communication and Signal Processing (ICCSP)*. April 2017. Chennai: IEEE, 0588–0592. Available from https://doi.org/10.1109/ICCSP.2017.8286426 [Accessed 2 April 2023].

Anwar, S. and Barnes, N. (2019). Densely Residual Laplacian Super-Resolution. Available from http://arxiv.org/abs/1906.12021 [Accessed 2 May 2023].

Arican, M.E. et al. (2022). ISNAS-DIP: Image-Specific Neural Architecture Search for Deep Image Prior. Available from http://arxiv.org/abs/2111.15362 [Accessed 25 September 2022].

Arican*, M.E. et al. (2022). ISNAS-DIP: Image-Specific Neural Architecture Search for Deep Image Prior. Available from https://doi.org/10.5281/ZENODO.6351599 [Accessed 31 October 2022].

Chen, H. et al. (2022). Real-world single image super-resolution: A brief review. *Information Fusion*, 79, 124–145. Available from https://doi.org/10.1016/j.inffus.2021.09.005.

Chen, Y.-C. et al. (2020). NAS-DIP: Learning Deep Image Prior with Neural Architecture Search. In: Vedaldi, A. Bischof, H. Brox, T. et al. (eds.). *Computer Vision – ECCV 2020*. Lecture Notes in Computer Science. Cham: Springer International Publishing, 442–459. Available from https://doi.org/10.1007/978-3-030-58523-5_26 [Accessed 28 September 2022].

Claus, M. and van Gemert, J. (2019). ViDeNN: Deep Blind Video Denoising. Available from http://arxiv.org/abs/1904.10898 [Accessed 2 May 2023].

Creswell, A. et al. (2018). Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*, 35 (1), 53–65. Available from https://doi.org/10.1109/MSP.2017.2765202.

El Helou, M. and Susstrunk, S. (2020). Blind Universal Bayesian Image Denoising With Gaussian Noise Level Learning. *IEEE Transactions on Image Processing*, 29, 4885–4897. Available from https://doi.org/10.1109/TIP.2020.2976814.

Elharrouss, O. et al. (2020). Image Inpainting: A Review. *Neural Processing Letters*, 51 (2), 2007–2028. Available from https://doi.org/10.1007/s11063-019-10163-0.

Elsken, T., Metzen, J.H. and Hutter, F. (2019). Neural Architecture Search: A Survey. Available from http://arxiv.org/abs/1808.05377 [Accessed 28 September 2022].

Fan, L. et al. (2019). Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, 2 (1), 7. Available from https://doi.org/10.1186/s42492-019-0016-7.

Fu, K. et al. (2020). Image Super-Resolution Based on Generative Adversarial Networks: A Brief Review. *Computers, Materials & Continua*, 64 (3), 1977–1997. Available from https://doi.org/10.32604/cmc.2020.09882.

Ganepola, V.V.V. and Wirasingha, T. (2021a). A Novel Framework for Semantic Image Segmentation using Generative Adversarial Networks and Neural Architecture Search. *2021 10th International Conference on Information and Automation for Sustainability (ICIAfS)*. 11 August 2021. Negambo, Sri Lanka: IEEE, 252–257. Available from https://doi.org/10.1109/ICIAfS52090.2021.9605889 [Accessed 12 October 2022].

Ganepola, V.V.V. and Wirasingha, T. (2021b). Automating Generative Adversarial Networks using Neural Architecture Search: A Review. *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*. 5 March 2021. Pune, India: IEEE, 577–582. Available from https://doi.org/10.1109/ESCI50559.2021.9396991 [Accessed 28 September 2022].

Gao, C. et al. (2020). AdversarialNAS: Adversarial Neural Architecture Search for GANs. Available from http://arxiv.org/abs/1912.02037 [Accessed 28 September 2022].

Ghahremani, M. et al. (2022). Adversarial Distortion Learning for Medical Image Denoising. Available from http://arxiv.org/abs/2204.14100 [Accessed 2 May 2023].

Gilbert, A., Messingher, S. and Patel, A. (2018). Non-Blind Image Deblurring using Neural Networks. 7.

Gong, X. et al. (2019). AutoGAN: Neural Architecture Search for Generative Adversarial Networks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. October 2019. Seoul, Korea (South): IEEE, 3223–3233. Available from https://doi.org/10.1109/ICCV.2019.00332 [Accessed 28 September 2022].

Goodfellow, I. (2017). NIPS 2016 Tutorial: Generative Adversarial Networks. Available from http://arxiv.org/abs/1701.00160 [Accessed 18 October 2022].

Goodfellow, I.J. et al. (2014). Generative Adversarial Networks. Available from http://arxiv.org/abs/1406.2661 [Accessed 28 September 2022].

Goyal, B. et al. (2020). Image denoising review: From classical to state-of-the-art approaches. *Information Fusion*, 55, 220–244. Available from https://doi.org/10.1016/j.inffus.2019.09.003.

Gui, J., Cong, X., et al. (2022). A Comprehensive Survey and Taxonomy on Single Image Dehazing Based on Deep Learning. Available from http://arxiv.org/abs/2106.03323 [Accessed 27 February 2023].

Gui, J., Sun, Z., et al. (2022). A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 1–1. Available from https://doi.org/10.1109/TKDE.2021.3130191.

Hu, B. et al. (2020). Subjective and objective quality assessment for image restoration: A critical survey. *Signal Processing: Image Communication*, 85, 115839. Available from https://doi.org/10.1016/j.image.2020.115839.

Ilesanmi, A.E. and Ilesanmi, T.O. (2021). Methods for image denoising using convolutional neural network: a review. *Complex & Intelligent Systems*, 7 (5), 2179–2198. Available from https://doi.org/10.1007/s40747-021-00428-4.

Isola, P. et al. (2017). Image-to-Image Translation with Conditional Adversarial Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017. Honolulu, HI: IEEE, 5967–5976. Available from https://doi.org/10.1109/CVPR.2017.632 [Accessed 5 April 2023].

Jaesung Rim, S.C., Haeyun Lee, Jucheol Won. (2020). Real-World Blur Dataset for Learning and Benchmarking Deblurring Algorithms. *Proceedings of the European Conference on Computer Vision (ECCV)*. 2020.

Kupyn, O. et al. (2018). DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks. Available from http://arxiv.org/abs/1711.07064 [Accessed 19 July 2022].

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. *Nature*, 521 (7553), 436–444. Available from https://doi.org/10.1038/nature14539.

Ledig, C. et al. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. Available from http://arxiv.org/abs/1609.04802 [Accessed 21 December 2022].

Li, Z. et al. (2022). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33 (12), 6999–7019. Available from https://doi.org/10.1109/TNNLS.2021.3084827.

Liu, A. et al. (2021). Blind Image Super-Resolution: A Survey and Beyond. Available from http://arxiv.org/abs/2107.03055 [Accessed 28 September 2022].

Lutz, S., Amplianitis, K. and Smolic, A. (2018). AlphaGAN: Generative adversarial networks for natural image matting. Available from http://arxiv.org/abs/1807.10088 [Accessed 5 April 2023].

Mahesh, B. (2018). Machine Learning Algorithms - A Review. 9 (1).

Martin, D. et al. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Proceedings*

Eighth IEEE International Conference on Computer Vision. ICCV 2001. 2001. Vancouver, BC, Canada: IEEE Comput. Soc, 416–423. Available from https://doi.org/10.1109/ICCV.2001.937655 [Accessed 3 May 2023].

Nah, S., Kim, T.H. and Lee, K.M. (2017). Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017. Honolulu, HI: IEEE, 257–265. Available from https://doi.org/10.1109/CVPR.2017.35 [Accessed 31 October 2022].

Pan, X. et al. (2020). Exploiting Deep Generative Prior for Versatile Image Restoration and Manipulation. Available from http://arxiv.org/abs/2003.13659 [Accessed 28 September 2022].

Park, S.H., Moon, Y.S. and Cho, N.I. (2023). Perception-Oriented Single Image Super-Resolution using Optimal Objective Estimation. Available from http://arxiv.org/abs/2211.13676 [Accessed 2 May 2023].

Qin, Z. et al. (2021). Image inpainting based on deep learning: A review. *Displays*, 69, 102028. Available from https://doi.org/10.1016/j.displa.2021.102028.

Ran, X. and Farvardin, N. (1995). A perceptually motivated three-component image model-Part I: description of the model. *IEEE Transactions on Image Processing*, 4 (4), 401–415. Available from https://doi.org/10.1109/83.370671.

Sara, U., Akter, M. and Uddin, M.S. (2019). Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. *Journal of Computer and Communications*, 07 (03), 8–18. Available from https://doi.org/10.4236/jcc.2019.73002.

Saunders, M.N.K., Lewis, P. and Thornhill, A. (2009). *Research methods for business students*, 5th ed. New York: Prentice Hall.

Singh, D. and Kumar, V. (2019). A Comprehensive Review of Computational Dehazing Techniques. *Archives of Computational Methods in Engineering*, 26 (5), 1395–1413. Available from https://doi.org/10.1007/s11831-018-9294-z.

Su, J., Xu, B. and Yin, H. (2022). A survey of deep learning approaches to image restoration. *Neurocomputing*, 487, 46–65. Available from https://doi.org/10.1016/j.neucom.2022.02.046.

Suganuma, M., Ozay, M. and Okatani, T. (2018). Exploiting the Potential of Standard Convolutional Autoencoders for Image Restoration by Evolutionary Search. 10.

Sun, W. and Chen, Z. (2020). Learned Image Downscaling for Upscaling using Content Adaptive Resampler. *IEEE Transactions on Image Processing*, 29, 4027–4040. Available from https://doi.org/10.1109/TIP.2020.2970248.

Ulyanov, D., Vedaldi, A. and Lempitsky, V. (2020). Deep Image Prior. *International Journal of Computer Vision*, 128 (7), 1867–1888. Available from https://doi.org/10.1007/s11263-020-01303-4.

Vassilo, K. et al. (2020). Multi-Step Reinforcement Learning for Single Image Super-Resolution. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2020. Seattle, WA, USA: IEEE, 2160–2168. Available from https://doi.org/10.1109/CVPRW50498.2020.00264 [Accessed 2 May 2023].

Wang, Z., Chen, J. and Hoi, S.C.H. (2021). Deep Learning for Image Super-Resolution: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43 (10), 3365–3387. Available from https://doi.org/10.1109/TPAMI.2020.2982166.

Wang, Z., She, Q. and Ward, T.E. (2022). Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy. *ACM Computing Surveys*, 54 (2), 1–38. Available from https://doi.org/10.1145/3439723.

Wistuba, M., Rawat, A. and Pedapati, T. (2019). A Survey on Neural Architecture Search. Available from http://arxiv.org/abs/1905.01392 [Accessed 18 October 2022].

Xu, X. et al. (2017). Learning to Super-Resolve Blurry Face and Text Images. 10.

Zamir, S.W. et al. (2022). Restormer: Efficient Transformer for High-Resolution Image Restoration. Available from http://arxiv.org/abs/2111.09881 [Accessed 16 December 2022].

Zhang, H., Sindagi, V. and Patel, V.M. (2019). Image De-raining Using a Conditional Generative Adversarial Network. Available from http://arxiv.org/abs/1701.05957 [Accessed 21 December 2022].

Zhang, H. et al. (2021). Memory-Efficient Hierarchical Neural Architecture Search for Image Restoration. Available from http://arxiv.org/abs/2012.13212 [Accessed 25 September 2022].

Zhang, K., Zuo, W. and Zhang, L. (2018). FFDNet: Toward a Fast and Flexible Solution for CNN based Image Denoising. *IEEE Transactions on Image Processing*, 27 (9), 4608–4622. Available from https://doi.org/10.1109/TIP.2018.2839891.

Zhang, K. et al. (2022). Deep Image Deblurring: A Survey. *International Journal of Computer Vision*, 130 (9), 2103–2130. Available from https://doi.org/10.1007/s11263-022-01633-5.

Zhang, Y. and Li, K. (no date). Image Super-Resolution Using Very Deep Residual Channel Attention Networks.

Zhang, Z. et al. (2023). *Data-Driven Single Image Deraining: A Comprehensive Review and New Perspectives*. Available from https://doi.org/10.36227/techrxiv.14944752.v4 [Accessed 27 February 2023].

Zoph, B. and Le, Q.V. (2017). Neural Architecture Search with Reinforcement Learning. Available from http://arxiv.org/abs/1611.01578 [Accessed 4 April 2023].

APPENDIX I – Concept Map

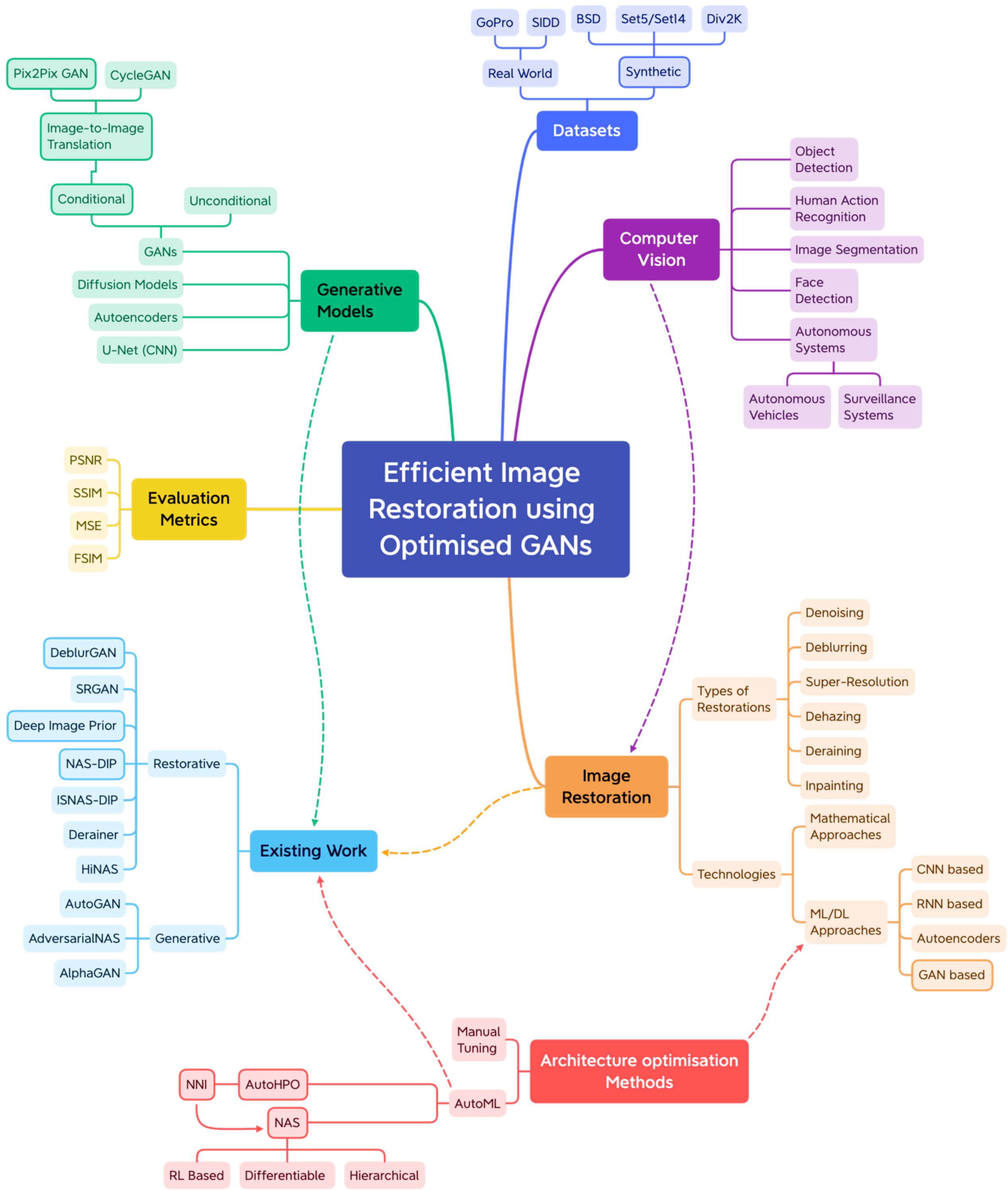


Figure 0-1: Concept Map

APPENDIX II – Gantt Chart

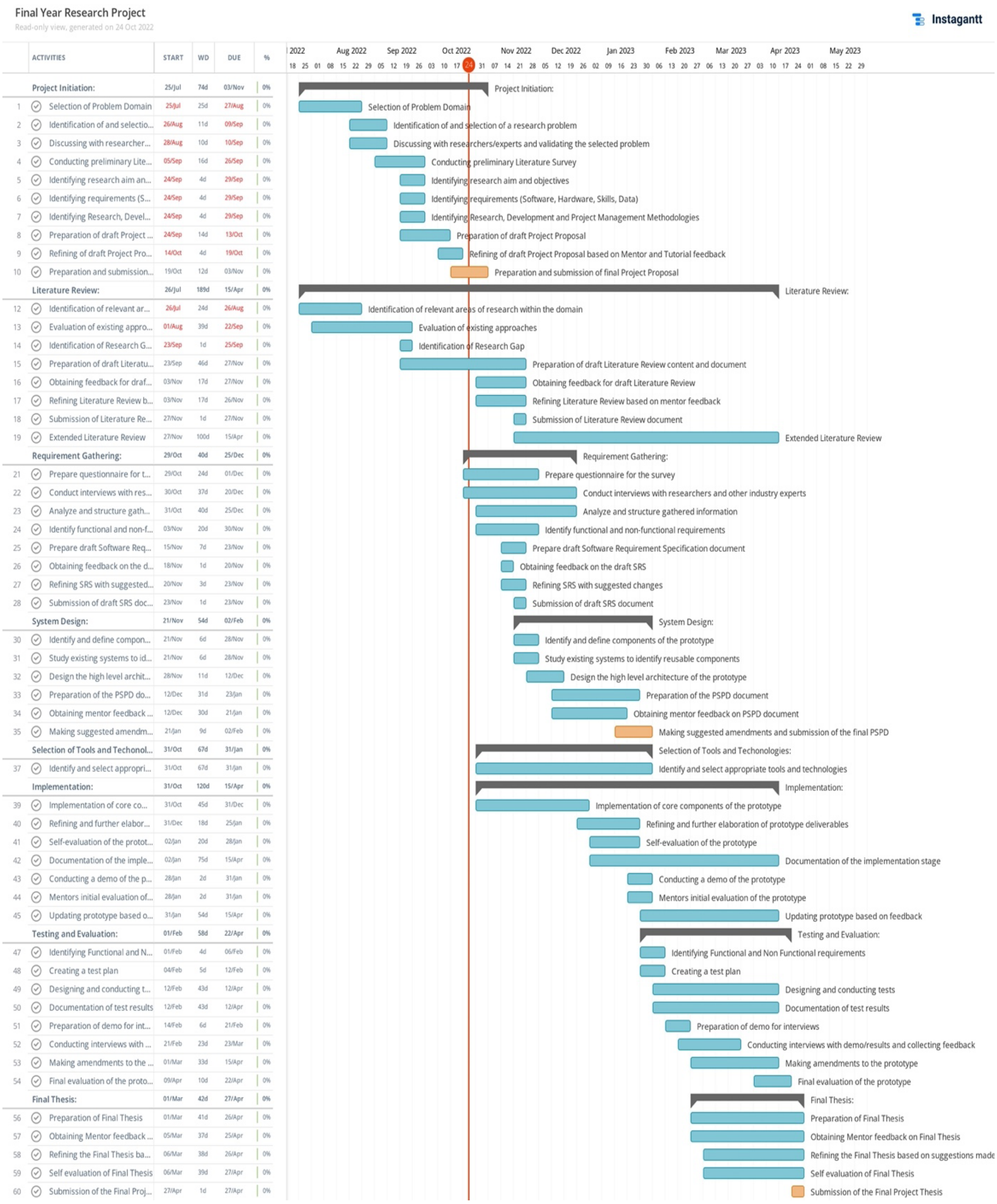


Figure 0-1: Gantt Chart

APPENDIX III – Thematic Analysis (Requirement Elicitation)

Interviewee ID	Name	Designation
P1	Anonymous	Senior Lecturer, Department of Electronic and Telecommunication Engineering, University of Moratuwa
P2	Ms. Vishmi Ganepola	Software Engineer/Researcher, Pearson Lanka
P3	Ms. Senuri De Silva	PhD Student, National University of Singapore
P4	Mr. Thepul Ginige	Senior Lecturer/Programme Coordinator, University College Lanka
P5	Mr. Dulaj Weerakoon	Research Engineer, Singapore Management University
P6	Anonymous	Data Scientist, King's College London

Table 0-1: Appendix A - Interviewee Details

Question	**How useful do you think a system to automate the creation of image restoration neural network architectures and integrating them with GANs would be on a scale of 1-5?**
Responses	
P1	"Concept sounds pretty advanced, especially for an undergraduate. Surely a 4 or 5 out of 5 for me. Working with degraded images is one of the more difficult areas in computer vision"
P2	"I definitely think it would be a 4 or 5 out of 5. It's important to try out new approaches to solve problems because that is the point of having technological developments."
P3	"This idea sounds interesting. I'm not too familiar with image restoration, but the concept sounds novel and unique. I think it's a 4 out of 5"

P4	"Automating this is a bit like AutoML I think. Should be very useful if you can build it. It's not easy to work with visual corruptions – but it's very useful for a lot of fields nowadays. I'd say 4 out of 5"
P5	"GANs is not very easy to work with. Automating the architecture will help a lot of research in this field. 4 out of 5"
P6	"I'm not very familiar with computer vision, but I know GANs is very powerful and complex. Automating part of the architecture of something like that is definitely a 4 out of 5. This will be useful since you're working on image degradations and corruptions"
Question	**What are your thoughts on using Neural Architecture Search to automatically architect such image restoration neural networks?**
Response	
P1	"NAS is very resource intensive. If you can find a way to use it with low resources, then that is great!"
P2	"Using NAS for this is very impressive. Experimenting with applying NAS towards image restoration problems is a commendable research effort."
P3	"Very useful trying out state of the art methods"
P4	"This would be good for Memory-Efficient Hierarchical Neural Architecture Search for Image Restoration"
P5	"Using NAS with GANs is very complex. This has high academic research value"
P6	"Impressive to use technologies like that."
Question	**What are the main technical aspects (if any) you would advise that I must prioritise and consider in relation to the NAS component of this project?**
Response	
P1	"Try to use existing NAS algorithm and integrate it with the restoration model. If you try to build it from scratch it might be too resource intensive."
P2	"Consider existing NAS based systems as a starting point since it would be easier to obtain a better understanding on how such a system functions. Using an existing algorithm would also make it easier to do more experiments and make improvements faster within the available time period."
P3	"I'm not familiar enough with NAS to give a definitive answer"

P4	“Evolutionary neural architecture search, Deep Image Prior to be applied fully automatically”
P5	“Try out with existing algorithms to get more understanding. Mathematically based NAS algorithms might be a lot more efficient than ones using reinforcement learning or similar approaches.”
P6	“I’m not too familiar with NAS to answer this question properly”
Question	**Are there any specific network architectures/GAN models or NAS algorithms that you could recommend for the purpose of this project?**
Response	
P1	“Look into existing NAS algorithms. You could also start with GAN architectures like Pix2Pix or CycleGAN. You must first verify if the selected NAS and GANs components are compatible.”
P2	“Pix2Pix GAN might be very good for this. CycleGAN is good too but could be more complex since there are 4 networks instead of 2”
P3	“Try out some popular and proven GAN architectures first”
P4	“CycleGAN can be used”
P5	“I’m not sure about specific GAN architectures but you can start with some commonly used ones. Pix2Pix might be a good fit”
P6	“Try to select a GAN with lesser complexity because otherwise the research might become too heavy”
Question	**What AI/ML libraries would you recommend experimenting with for the progression of this project?**
Response	
P1	“Standard ones like PyTorch and Tensorflow should do.”
P2	“You can try with PyTorch and also Tensorflow. Tensorboard is useful to view model metrics. Other image processing libraries will also be useful.”
P3	“Both Tensorflow and Pytorch can be tested.”
P4	“Tensorflow, Pytorch, SciKit Learn. There are other libraries as well.”
P5	“Tensorflow or PyTorch. You might have to experiment a little to decide properly.”
P6	“Pytorch and Tensorflow mainly.”

Question	**What web-application tools/framework(s) would you recommend for building a prototype to demonstrate the functionality of the proposed system as an end product?**
Response	
P1	"That is up to you. The most important part here is the core research component because it has a very good novelty and potential. Try to focus on that"
P2	"You can use React for the front end and something like Flask to build the API. Use whatever you're comfortable with"
P3	"Flask or Django depending on your preference. Use any front-end library you like"
P4	"React is good for front end if you're familiar with it. Django or Flask for backend"
P5	"You can use React or Vue if you're comfortable with that, or even Flutter. Backend could be built with Flask. This project has very high academic research value so it's best to focus on that"
P6	"Use any libraries you're comfortable with. Focus on the research component more. This project seems to have very high potential"
Question	**What datasets would you recommend for training the NAS and GAN components of this project?**
Response	
P1	"Use datasets that have been commonly used for other research in this domain. Make sure the data is high quality because it will affect your results"
P2	"NAS for DIP dataset since it has been used for another NAS project. If you're using a Pix2pix GAN, make sure you have 1:1 annotated dataset for training."
P3	"RealBlur, GoPro-V7 datasets"
P4	"RealBlur and GoPro-V7 datasets. Try out other popular datasets as well."
P5	"You could try with tried and tested datasets to make sure the data is of high quality. For initial testing you could use something like the MNIST dataset. Later you could use datasets with complex high-resolution images."
P6	"You can try with Kaggle datasets or other datasets used for similar projects."
Question	**What evaluation metrics would you recommend for evaluating the final image restoration model and resulting outputs?**

Response	
P1	"Standard metrics like PSNR and SSIM"
P2	"PSNR, SSIM scores are useful."
P3	"FID scores could be used since you're working with a GAN model."
P4	"PSNR Score to verify the quality of the output."
P5	"PSNR and SSIM scores. FID score could be used to verify GAN performance."
P6	"PSNR score can be used"
Question	**What features would you recommend to include in the final prototype of this project?**
Response	
P1	"You can show the results by restoring an image. Spend more time on improving the core component of the research. Use a simple UI if you need one"
P2	"Start with the main component, that is restoring degraded images. If you have time you could also add features to allow retraining the model on custom datasets. Also you could integrate Tensorboard into the GUI to show the model performance."
P3	"Nothing in specific. Showcase the performance of the core research component"
P4	"Mainly the image restoration feature. Use a simple UI"
P5	"Include the core feature, that is restoring degraded images. You could also show the model performance metrics on the GUI. Build a simple UI so that you have more time to spend on the core research component."
P6	"Focus on the image restoration component mostly"

Table 0-2: Appendix A - Interview Summary

APPENDIX IV – Additional Use Case(s)

Use Case	**Train Image Restoration Model**	
Description	This use case describes the process of retraining the image restoration model on a given dataset.	
Participating Actors	Computer Vision Developers/Researchers	
Pre-Conditions	The required datasets and pre-trained/untrained model should be available on the running system.	
Extended Use Cases	1. Upload custom datasets. 2. Search for optimal network architecture using NAS.	
Included Use Cases	1. View performance metrics. 2. View/Download test results and trained model.	
Main Flow	Actor	System
	1. The user must select a dataset to retrain the image restoration model. 3. If the user requires, they may view the performance metrics of the newly trained GAN model. 4. If the user requires, they may download the test results and the trained model.	2. Load the training dataset into the model and start training the model when triggered whilst displaying training and performance metrics.
Alternate Flows	**AF1** – If the user requires, they may upload a custom dataset to train the restoration model. **AF2** – If the user requires, they may retrigger a search for the optimal generator architecture using NAS.	
Exceptional Flows	**EF1** – Unable to train the restoration model: Display an error message.	
Post-Conditions	1. If successful, performance metrics of the model based on test data will be displayed. 2. If the model training is successful, the trained model will be available for download to the user.	

Table 0-1: Use Case Description - 2

APPENDIX V – Component Diagram

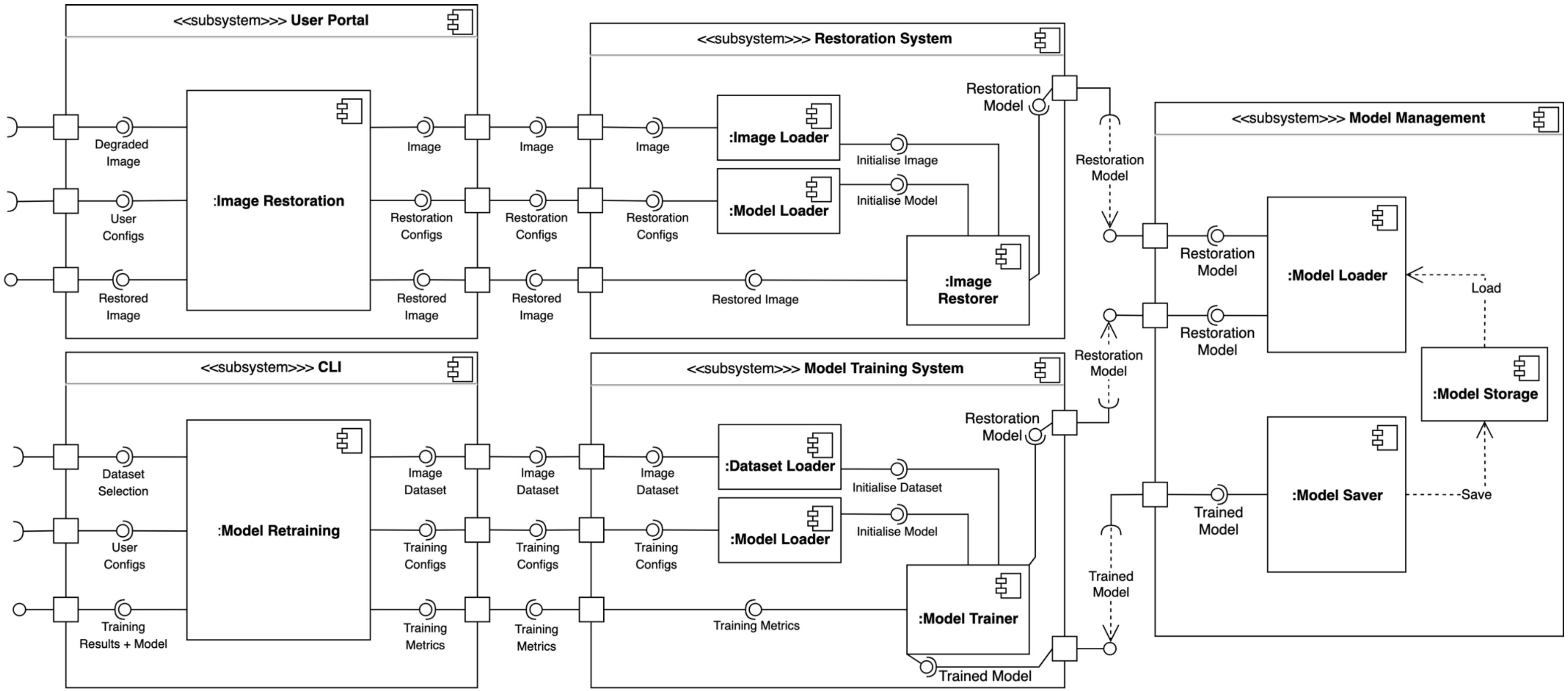


Figure 0-1: Component Diagram (Self-Composed | Landscape)

APPENDIX VI – UI Wireframes

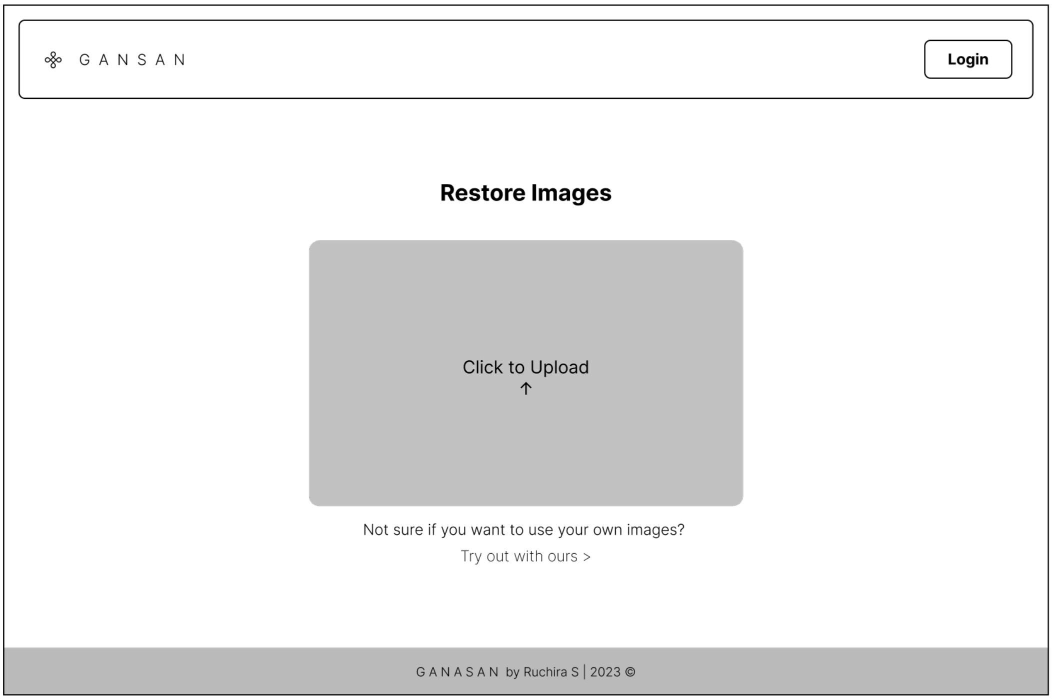


Figure 0-1: Image Upload page

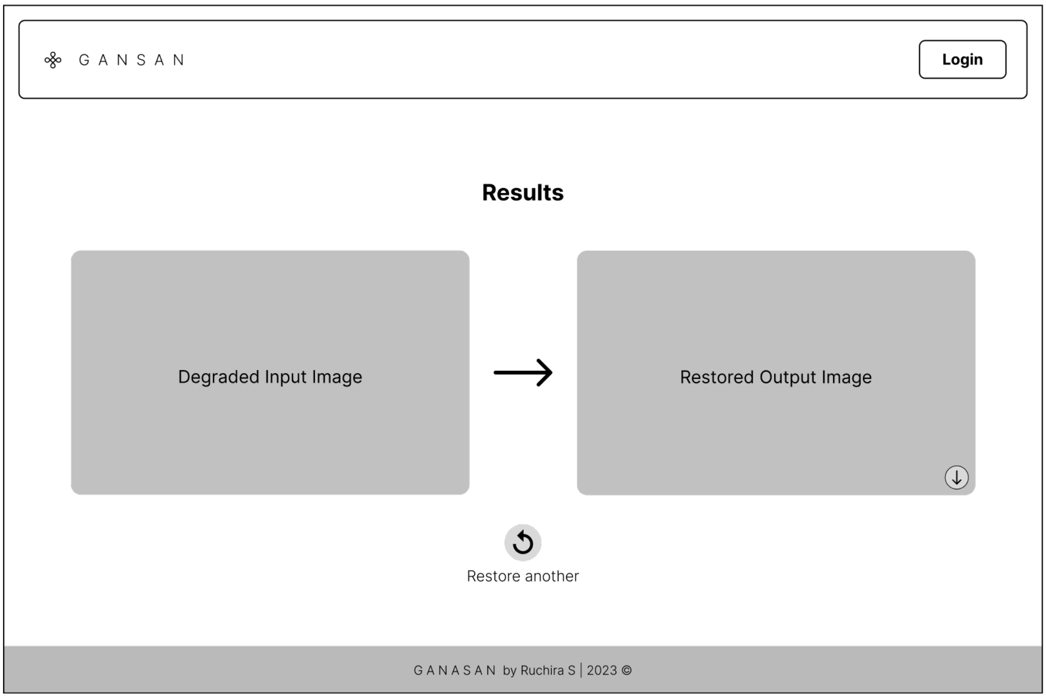


Figure 0-2: Results Output page

APPENDIX VII – Module Integration Test Results

Module	Input	Expected Output	Actual Output	Status
Image input module	Selected Image	Image gets uploaded to the upload folder and gets attached to the corresponding UI component.	Image gets uploaded to the upload folder and gets attached to the corresponding UI component.	Passed
Image restoration module	Uploaded degraded image	The degraded image is passed on to the selected restoration model through the data loader to be restored.	The degraded image is passed on to the selected restoration model through the data loader to be restored.	Passed
Restored image output loader	Restored image output from the model.	Saves the restored image to a temporary internal folder and display using the corresponding UI component.	Saves the restored image to a temporary internal folder and display using the corresponding UI component.	Passed
Restore image downloader	Restored image displayed on the UI	Downloads the restored image to the user's local drive.	Downloads the restored image to the user's local drive.	Passed

Table 0-1: Module and Integration Testing

APPENDIX VIII – Project Evaluators' Details

<table>
<tr><th>ID</th><th>Position</th><th>Affiliation</th></tr>
<tr><td colspan="3">Computer vision, image signal processing, and deep learning/machine learning experts.</td></tr>
<tr><td>E1</td><td>Research Engineer – Computer Vision</td><td>Singapore Management University</td></tr>
<tr><td>E2</td><td>Machine Learning Engineer (Level 2)</td><td>Leather Broadcasting Inc. San Francisco, CA, USA</td></tr>
<tr><td>E3</td><td>PhD Candidate / Research Assistant</td><td>Stony Brook University, NY, USA</td></tr>
<tr><td>E4</td><td>PhD Candidate</td><td>University of New South Wales, Australia</td></tr>
<tr><td>E5</td><td>Researcher, Advisor and Consultant</td><td>Massey University, NZ</td></tr>
<tr><td colspan="3">Experts in research and development from related technical fields</td></tr>
<tr><td>E6</td><td>Senior Research Scientist – BSc (Eng), M. Eng, PhD</td><td>Apple Inc., Cupertino, CA, USA</td></tr>
<tr><td>E7</td><td rowspan="2">Senior Lecturer</td><td rowspan="2">University of Moratuwa</td></tr>
<tr><td>E8</td></tr>
<tr><td>E9</td><td>Director</td><td>Kingslake Engineering Systems (Pvt) Ltd</td></tr>
<tr><td>E10</td><td>Consultant - Embedded Systems</td><td>Oxit LLC</td></tr>
<tr><td>E11</td><td>Former Senior Research Engineer (ML & DS Team Lead)</td><td>4Axis Solutions (pvt) Ltd</td></tr>
<tr><td>E12</td><td>Project CEO</td><td>Nagarro Sri Lanka</td></tr>
<tr><td colspan="3">Experts in the field of software engineering</td></tr>
<tr><td>E13</td><td rowspan="2">Senior Software Engineer</td><td rowspan="2">Axiata Digital Labs (pvt) Ltd</td></tr>
<tr><td>E14</td></tr>
<tr><td>E15</td><td>Head of Engineering</td><td rowspan="2">FCode Labs (pvt) Ltd</td></tr>
<tr><td>E16</td><td>Managing Director</td></tr>
</table>

Table 0-1: Project Evaluators' Details

APPENDIX IX – Project Evaluation Thematic Analysis

The following table presents the mapping between identified codes and derived themes of the evaluation process.

Codes	Theme
Research domain, Research trends, Practical Applications, Related domains	Research Topic
Topic complexity, Research challenge	Overall Concept
Significance of the concept, Significance of the technologies, Significance to the research domain/ community	Significance of the undertaking
Level of study, Self-learning, Research depth, Technical concepts, Output results	Depth of the research work
Critical review of literature, Coverage of concepts and technologies, Presentation of LR	Quality of the LR
Demonstrated understanding, Research approach, LR approach	Research process followed
Understanding of challenges in ML, Usage of libraries, Scope navigation,	Deep learning component
Code quality, Project structure, Technology stack, Solution scalability, Development approach	General development
Quality of outputs, Perceptual improvments, Quality of demonstration, Quality of presentation	Output results
Numerical metrics, Comparison to existing work	Benchmarking
Visual interface design (UI), Perceived Complexity (UI/UX)	Overall UI/UX
Functional processes, Product Features	Functionalities of the MVP

Table 0-1: Evaluation Codes and Themes

The following table presents an abridged summary of all evaluation responses and the relevant/recurring codes identified thereof.

Evaluator	Response (abridged)	Codes
Question: What are your thoughts on the overall concept and choice of research topic?		
E1	"An exciting area of research that has the potential to lead to significant improvements in the quality of restored images". "Challenging task because image restoration involves dealing with different types of noise, artifacts, and distortions, and finding the optimal architecture and hyperparameters of the GANs can be a complex optimization problem". "Important area of research that could have practical applications in various fields".	Research domain, Research trends, Research challenge, Related domains, Topic complexity, Practical applications
E2	"Overall the concept is good."	
E3	"The idea seems novel and is applied in an interesting setting here."	
E4	"The research topic is a trending and highly research-driven area within several safety-critical research domains such as autonomous vehicles. The proposed concept of using NAS to optimise image restoration tasks is a novel contribution with promising results."	
E5	"For a FYP, you have selected a topic to demonstrate your capacity and potential in this evolving field. It is useful in many applications such as surveillance and achieving."	
E6	"For an undergraduate project, the research carried out is very impressive. The concept and the research topic is good."	
E7	"Overall concept of the research topic is very good. The student has taken a courageous decision to take up a contemporary research project of this nature as the final year project."	

E8	"This research is both timely and captivating as it has various applications in diverse fields such as medicine, security, and entertainment where accurate and high-quality images are crucial."	
E9	"Today, image processing with Deep learning techniques are a good area for research with practical applications. Hence the topic is a good choice which will further the theoretical and practical knowledge of the student."	
E10	"The topic chosen is a popular research problem for decades, which is still a challenging topic. There are a lot of real world applications waiting for improved solutions to this problem..."	
E11	"Can be useful in unstable and low light situations."	
E12	"Given the fact that image processing has now become a household activity, this is a topic which will have a widespread application. Even though this has been a research area for a quite a long time I am impressed withn the quality of the images generated."	
E13	"Novel idea with practical usage."	
E14	"The concept of using GAN and NAS for image restoration is very interested and this has the potential to contribute the advancement of image restoration techniques and address the challenges in the field."	
E15	"Image restoration is widely used in industry. Efficient and fast restoration techniques are at utmost importance."	
E16	"Concept is timely because AI has become very popular among both the academia and the general public. In addition, image-based AI techniques like denoising and deblurring are very helpful in a lot of situations."	
Question: What is your assessment on the significance and depth of the work undertaken?		

E1	“The use of GANs and NAS for image restoration is a challenging and complex task, and the fact that Ruchira was able to make progress in this area is commendable”. “Ruchira's work in image restoration using GANs and NAS is significant and has the potential to contribute to the advancement of computer vision and related fields.”	Significance of the concept, Significance of the technologies, Significance to the research domain/ community, Level of study, Self-learning, Research depth, Technical concepts, Output results,
E2	“By doing this sort of work, the student get's the experience of an end to end project. Therefore, student has done a significant work with a considerable depth”	
E3	“The work is significant given its timely nature and usefulness to the greater research community.”	
E4	“The significance and depth of the research project are excellent for undergraduate-level research work. This is because the student has to self-learn state-of-the-art technologies as a proper understanding of each technology is essential for implementing the proposed approach.”	
E5	You have done a broad literature review of artificial intelligence (AI) and its variants in this context. Need little bit deeper and wider analysis of traditional image restoration based on currently available techniques, their limitations and merits. As mentioned earlier, topic is significant and involve less ethical concerns.	
E6	“The research work carried out is significant as this is a common problem that is faced image processing.”	
E7	“Both the significance and the depth of the work undertaken are certainly above average considering this as an undergraduate project.”	
E8	“The profundity and importance of this work meet my expectations, and I am content with the level of significance it carries.”	
E9	“It can be seen that the student has studied the available research material & literature in depth prior to developing a solution.”	

E10	"The research has covered sufficient depth for a time bound undergraduate project. Considering the challenges such as lack of training data sets, the researcher has managed to achieve excellent results."	
E11	"Could be significant for photos and videos taken using mobile phones"	
E12	"I believe this is a highly researched area. The candidate has done a decent job to be in par with the rest."	
E13	"Good."	
E14	"The work undertaken in identifying the limitations of image restoration techniques, researching and implementing advanced methods effectively, and presenting the results in a clear and comprehensive manner holds significant value."	
E15	"Good level for an individual project."	
E16	"A good job has been done on the work as a final year project. The appropriate technologies in machine learning for deblurring and denoising images have been researched extensively. Based on that, the decisions on the machine learning technologies, programing language and frameworks to use in the implementation have been made. The implementation and integration have been done excellently to demonstrate the application very well."	
Question: What are your thoughts on the demonstrated quality of the research process and literature review?		
E1	"Ruchira's literature review demonstrates the quality of the research process and literature review." "[It] shows his thorough understanding of the topic." "Overall, Ruchira's review provides a comprehensive understanding of the state-of-the-art techniques in image restoration using GANs and NAS/AutoHPO."	Critical review of literature, Coverage of concepts and technologies, Demonstrated understanding, Presentation of
E2	"Demonstration of work is excellent and interesting. To come up with a solution with this quality, it's evident that	

	the student has done a literature search up to a considerable extent.”	LR, Research approach, LR approach
E3	“There is some literature review and strong research process.”	
E4	“A thorough literature review was conducted with a critical analysis of the state-of-the-art. A publishable piece of work.”	
E5	“Literature review has been carried out well. Could have included some [more] techniques of image restorations their merits and demerits, which allows new methods, techniques and architectures. It has been done in AI domain itself.”	
E6	“Again, I think it's very impressive for an undergraduate research project.”	
E7	“Literature review is excellent. Student have followed a very methodical approach for the research process.”	
E8	“The effort and dedication to develop and implement advanced image restoration techniques using mathematical models, machine learning, and deep neural networks is evident. A comprehensive review of the literature has been presented.”	
E9	“Very good.”	
E10	“The researcher has conducted sufficient literature review and has demonstrated a methodical approach in meeting the research goal.”	
E11	N/A	
E12	N/A	
E13	“Demonstration is clear and literature review is informative.”	
E14	“It is very informative and nicely presented.”	
E15	“Good.”	
E16	“I think the literature review and research process of the project are very good… demonstrates a thorough	

	understanding of the existing methods and challenges in denoising and deblurring images." "The literature review done provides a comprehensive and critical evaluation of the relevant literature, as well as a detailed description of the data, algorithms, and evaluation metrics used in the application. A systematic research process has followed that ensures the validity and reliability of the findings."	
Question: What is your view on the development approach followed by the student?		
E1	"Overall, Ruchira's approach seems well-considered and demonstrates a good understanding of the key components required for the successful development of an image restoration system."	Understanding of challenges in ML, Usage of libraries, scope navigation, Code quality, Project structure, Technology stack, Solution scalability, Development approach
E2	"Development approach looks great. Student has done a considerable amount of work since it is done from end to end."	
E3	"The student has put in a lot of effort to come up with an innovative outcome."	
E4	"Excellent for an undergraduate individual research project."	
E5	"The development approach by proposing a new architecture, then following best practices in in the development process has led to the demonstratable prototype or solution"	
E6	"The development approach looks good. I believe its a pretty standard development approach to solve this type of problems."	
E7	"Very Good."	
E8	"In general, the development approach for image restoration appears to be effective."	
E9	"The standard methodologies have been followed."	

E10	“He has understood the inherent challenges in the field of ML and managed to overcome them by carefully selecting the problem scope and the most viable solutions.”	
E11	N/A	
E12	“Usage of existing libraries and scalability of the solution can be recognised as good design approach.”	
E13	“Good approach and clean folder architecture in development.”	
E14	“As an undergraduate, your ability to utilise the modern technology stacks for both front and and back-end development to create a robust application is impressive and commendable.”	
E15	“Good.”	
E16	“The project has a good development approach for a final year project…”	
Question: What are your views on the demonstrated test results? (Answered based on the provided quantitative metrics, or qualitative aspects, or both)		
E1	“Was able to successfully demonstrate de-noising and de-blurring of images.” “It's also important to compare the performance of the proposed system with existing state-of-the-art solutions to establish its relative strengths and weaknesses.”	Quality of outputs, Perceptual improvments, Quality of demonstration, Quality of presentation, Numerical metrics, Comparison to existing work
E2	“The demonstrated results looks great! The quality of outcome is evident.”	
E3	“The method obtains improvements measured using identified suitable metric over their selected baselines.”	
E4	“Seems the proposed approach gives the promising results. If possible please benchmark against the other research works.”	
E5	“MSE, PSNR and SSIM objective metrics are the best to evaluate image quality. However, due to practicality, subjective measure MOS (Mean Opinion Score) could not	

	use. Above three metrics are reasonable proxies of MOS in image quality evaluation in general."	
E6	"Looking at the demo, it seems to be giving results as expected. I have to mention that Ruchira gave an excellent presentation. My experience with Sri Lankan undergraduates is that although their technical skills are good, the presentation skills are poor. However, Ruchira seems to have excellent presentation and technical skills."	
E7	"Chosen metrics are good for the purpose. When presenting the effectiveness of the chosen approach against alternative approaches, it could be better if a more complete comparison of relevant values of the metrics be given against available alternative approaches."	
E8	"The test outcomes for both denoising and deblurring are satisfactory."	
E9	"A marked improvement can be seen after the images are processed…" "In my opinion, the final test for overall effectiveness is how clear the images would become, rather than the numerical ratings."	
E10	"De-noising and de-blurring both have shown excellent results." "it is good to introduce brightness normalization step against the original image, so that the application that consider brightness as a critical metric, can also use this solution"	
E11	"Looks good for an image of that resolution. Nice."	
E12	"Impressive."	
E13	"Qualitative analysis is sufficient. However, better to have some more quantitative matrices."	
E14	"The restored images presented in the demonstration show a significant improvement in visual quality, suggesting that the model is exhibiting a high degree of accuracy."	

E15	"Could be more comprehensive… compare with previous research"	
E16	"The achieved results are good for me." "The results were demonstrated by showing different kinds of images for each case, denoise and deblur. Each case had 3-4 images with impressive results."	
Question: What are your thoughts on the quality of the final prototype application?		
E1	"Ruchira's prototype application is simple and easy-to-use. a user-friendly interface."	Visual interface design (UI), Perceived Complexity (UI/UX), Functional processes, Product Features
E2	"The student has wrapped the core work of the project as a quality, functional prototype. UI/UX functionality is also satisfying."	
E3	N/A	
E4	"Good."	
E5	"It is a convenient UI that provide improved UX as the user can select the image and the application shows the restored image in the other window next to the original"	
E6	The UI looks good. I feel that it could have been better. It covers the basic functionality. But in the industry sometimes you need to develop very high quality UI to get wide acceptance.	
E7	"Prototype application is well designed. It hides all the complexity of internal computational processes form the user with a very simple and effective user interface."	
E8	"The user interface is both intuitive and comprehensive, offering easy handling of all available functionalities."	
E9	"It meets the objective of demonstrating the implemented techniques for image restoration. The main requirements - of process selection, uploading, processing and downloading the processed images are met."	
E10	"Good UI design. A general user can intuitively use the controls, which reflects good UX."	
E11	N/A	

E12	"The prototype is good enough to showcase the intended capabilities of the solution. However as a suggestion from the end user perspective, once the images uploaded, the product itself could decide what corrections are needed on the image and suggest the user for his confirmation… This is a very good foundation."	
E13	"Functional features are sufficient for the demonstration, but can improve UI/UX a bit."	
E14	"The prototype application has a user-friendly interface with intuitive navigation and its functionality is efficient, accurate and free of errors."	
E15	"Overall Good. Nice presentation and delivery. Intuitive interface."	
E16	"A prototype application should have minimalistic, intuitive UIs which are easier to use. I think the simple UI that has been developed is doing that job. As I have said earlier, the code quality is also good."	

Table 0-2: Evaluations Responses and Codes

APPENDIX X – Dataset Manipulation Scripts (Denoising)

```
DEVICE = "cuda" if torch.cuda.is_available() else "cpu"
TRAIN_DIR = "data/train"
VAL_DIR = "data/val"
LEARNING_RATE = 2e-4
BATCH_SIZE = 16
NUM_WORKERS = 2
IMAGE_SIZE = 256
CHANNELS_IMG = 3
L1_LAMBDA = 100
LAMBDA_GP = 10
NUM_EPOCHS = 500
LOAD_MODEL = False
SAVE_MODEL = True
CHECKPOINT_DISC = "disc.pth.tar"
CHECKPOINT_GEN = "gen.pth.tar"
```

Figure 0-1: Model configs

add_noise_2.py > ...

```
import numpy as np
import cv2
import glob
import os

mean = 5
var = 2
sigma = var ** 0.9

images = glob.glob('C:\Work\Deep Learning\Lab\Datasets\div2k\\512_val\*.png')
output_dir = ('C:\Work\Deep Learning\Lab\Datasets\div2k\\512_val_noise')

for id, image in enumerate(images):
    img = cv2.imread(image)

    # Define degradation parameters (image noise)
    gaussian = np.random.normal(mean, sigma, (img.shape[0],img.shape[1],img.shape[2]))
    noisy_image = np.zeros(img.shape, np.float32)

    # Add noise to image
    noisy_image = img + img*gaussian

    cv2.normalize(noisy_image, noisy_image, 0, 255, cv2.NORM_MINMAX, dtype=-1)
    noisy_image = noisy_image.astype(np.uint8)

    output_message = 'Noised index' + str(id)
    print(output_message)

    # Save to output directory
    output_filename = os.path.split(image)[1]
    cv2.imwrite(os.path.join(output_dir, output_filename), noisy_image)
```

Figure 0-2: Add Noise to Image Data

combine_data2.py > ...

```
import cv2
import glob
import os

output_dir = 'C:\Work\Deep Learning\Lab\Datasets\div2k\\512_val_combo'

for i in range(0, 800):

    # Define image data input and output directories
    input_dir_1 = 'C:\Work\Deep Learning\Lab\Datasets\div2k\\512_val_noise'
    input_dir_2 = 'C:\Work\Deep Learning\Lab\Datasets\div2k\\512_val'

    input_filename = str(i) + '.png'

    # Concatenate hr and degraded images horizontally
    img_right = cv2.imread(os.path.join(input_dir_1, input_filename))
    img_left = cv2.imread(os.path.join(input_dir_2, input_filename))
    img_combined = cv2.hconcat([img_right, img_left])

    # Save concatenated image files
    cv2.imwrite(os.path.join(output_dir, input_filename), img_combined)
    output_message = 'Combined index:' + str(i)
    print(output_message)
```

Figure 0-3: Concatenate High-Res and Degraded Images