MCAST

Black and White Image Colourisation using Deep Learning Techniques

Fabian Muscat

Supervisor: Thomas Gatt

June - 2023

A dissertation submitted to the Institute of Information and Communication Technology in partial fulfilment of the requirements for the degree of BSc (Hons) Multimedia in Software Development

Authorship Statement

This dissertation is based on the results of research carried out by myself, is my own composition, and has not been previously presented for any other certified or uncertified qualification.

The research was carried out under the supervision of **Mr. Thomas Gatt**

. . . .

Date Signature

Copyright Statement

In submitting this dissertation to the MCAST Institute of Information and Communication Technology, I understand that I am giving permission for it to be made available for use in accordance with the regulations of MCAST and the Library and Learning Resource Centre. I accept that my dissertation may be made publicly available at MCAST's discretion.

....

Date Signature

Abstract

Machine learning has been vastly used in many different aspects of arts and photography in the past few years, such as image restoration and artwork generation from photographs. This study explores the task of image colourisation using an auto-encoder and a generative adversarial network (GAN) and comparing the accuracy and efficiency with other state-of-the-art implementations. To accomplish the colourisation task, a dataset of grayscale images is preprocessed by converting them to grayscale and normalising the pixel values. The auto-encoder structure includes skip-connections to maintain information integrity during decoding. The GAN model comprises a generator and discriminator, utilising the least squares loss function to enhance the colourisation process. The study employs a combination of similarity metrics and colourisation Turing tests to assess the effectiveness of the developed models.

Quantitative evaluation is performed using similarity metrics to compare the colourised images to their respective ground truth counterparts, assessing the accuracy and fidelity of the colourisation process. Additionally, qualitative evaluation is conducted through colourisation Turing tests, where participants identify colourised images for various questions. The results indicate that both the auto-encoder and GAN models achieve satisfactory colourisation outcomes. The auto-encoder demonstrates consistent performance across different similarity metrics, exhibiting plausible results even with only 50 epochs of training. The GAN model shows better SSIM performance but a wider range of MSE values, requiring 150 epochs for optimal accuracy. Certain portrait images with complex backgrounds or diverse colours pose more challenges for accurate colourisation.

Keywords: Image Colourisation, Machine Learning, GAN, CNN, Auto-Encoders.

Acknowledgements

I would like to begin by expressing my heartfelt appreciation to my supervisor, Mr. Thomas Gatt, whose unwavering support, understanding, and patience have been instrumental in guiding me throughout the completion of this dissertation. His invaluable insights and feedback have greatly contributed to the overall quality of this work.

Next, I would like to extend my gratitude to my family for their encouragement and support. I am deeply grateful to my mother, Loranne, and my sister, Anthea, for their continuous belief in my abilities and their understanding during the demanding process of this research.

I would also like to acknowledge the support of my employer, SHIELD Consultants Ltd. Their understanding of the importance of academic pursuits and their flexibility in accommodating my research commitments have been truly remarkable. Their support has provided me with the necessary resources and time to dedicate myself fully to this dissertation.

Lastly, but certainly not least, I would like to express my sincere gratitude to all the participants who willingly dedicated their time and actively took part in this study. Without their cooperation and valuable contributions, this research would not have been possible. Their willingness to engage in the study and provide insightful feedback has greatly enriched the findings and overall quality of this work.

Table of Contents

Acronyms

NN	Neural Network
ML	Machine Learning
DL	Deep Learning
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
MSE	Mean Squared Error
SSIM	Structural Similarity Index Matrix
PSNR	Peak-Signal-to-Noise-Ratio

List of Figures

List of Tables

Introduction

1.1 Description of Theme and Topic Rationale

The theme of this research is improving image colourisation algorithms through the combination and modification of different existing algorithms. Existing image colourisation algorithms have limitations in terms of accuracy and efficiency. By implementing and evaluating different existing algorithms, the strengths and weaknesses can be identified, and a more effective and efficient algorithm could be developed.

1.2 Purpose of Study

The purpose of this study is to investigate and compare different deep learning techniques for colourising black and white images. We explore the performance of Auto-Encoders and Generative Adversarial Networks (GAN) in colourising images. The goal is to determine which algorithm is the most efficient and effective in generating accurate and visually appealing colourisations. To evaluate the performance of these algorithms, we conducted both qualitative and quantitative analyses. Qualitative analysis involves 2 online surveys where participants compare the ground truth colour images with the colourised images generated by different algorithms, and will also be given another set of images to rate the quality of the colourisation. Quantitative analysis involves a comparison between the ground truth images and the colourised images produced by the algorithms

using similarity metrics.

1.3 Research Aim

The aim of this study is to develop a more efficient and accurate image colourisation algorithm through analysis and evaluation of different existing algorithms. The ultimate goal is to contribute to the development of a more effective and practical algorithm which can be used in various fields such as digital art and image processing.

1.4 Problem

Image colourisation is a complex and difficult process, due to the broad range of imaging situations that must be addressed with a single method. The challenge is intrinsically tough because it lacks two of the three dimensions required of a coloured image. While semantic cues can help in some circumstances (for example, grass is mostly green, clouds are predominantly white, and the sky is predominantly blue), such priors are not readily available for many man-made and natural items, such as shirts, vehicles, and flowers. Additionally, the colourisation problem encompasses the typical difficulties encountered in image enhancement, including changes in illumination, variations in viewpoints, and occlusions.

1.5 Overview on Image Colourisation

Colourisation can be divided in two categories; colourisation of grayscale images and videos, and the other is colourisation of grayscale art forms, such as sketches and comics. This dissertation will focus on the colourisation of grayscale images.

The process of adding colour to a grayscale image is known as colourisation. This process allows us to give colour to historical moments and people which have never been seen in colour before. Colourisation is usually done by a human colourist, however, this can be a tedious and long process, involving in researching the historical background of the image and the manual colourisation of the image.

Being a photography and computer vision enthusiast, a machine learning approach is proposed to build a prototype which will automatically colourise images and reduce the trouble one has to go through to manually colourise an image. By combining the strengths of different image colourisation algorithms through comparison and modification can lead to a more efficient and accurate algorithm. This hypothesis brings forward the following questions:

1. What are existing image colourisation algorithms?
2. What types of images are easier/more difficult to colourise?
3. What are the strengths and weaknesses of each algorithm?
4. Can a hybrid algorithm combining the strengths of each algorithm be developed to improve to improve colourisation.

Literature Review

2.1 Introduction

Relevant literature will be used to analyse other researchers viewpoints on the problem of image colourisation. The implemented solutions of current research will be analysed, along with the utilised datasets. The importance of colour space models and different machine learning algorithms will also be discussed in the colourisation of an image.

Image colourisation is the process of adding colour information to grayscale images. Various fields have also been making use of colourisation, such as films, television shows and also the medical field [27]. Various machine algorithms have been implemented to address the problem of colourising black and white images. Such algorithms used are Convolutional Neural Networks (CNN), Generative Adversarial Networks (GAN) Memory Augmented Networks.

2.2 Colour Spaces

Colour spaces are essential in the process of image colourisation. In this discussion, we will delve into the RGB, HSV, LAB, and LUV colour spaces and their significance in the realm of image colourisation.

2.2.1 RGB

The RGB (Red, Green, Blue) colour space is commonly employed in digital images. It consists of three components - red, green, and blue - which determine the colour of each pixel based on their respective intensity values. However, when it comes to grayscale images, the use of the RGB colour space is restricted due to the absence of colour information [11]. Grayscale images are represented solely by shades of gray, with each pixel denoting a specific level of brightness. Since grayscale images lack colour, the three components of the RGB colour space do not hold any meaningful information. The RGB colour channels in grayscale images do not exhibit variations, as they are essentially monochromatic. In grayscale images, the focus is on capturing and representing variations in brightness rather than colour.

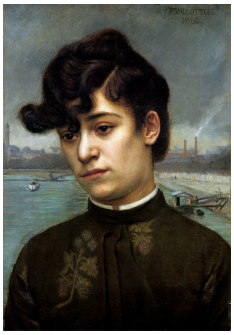

Figure 2.1: Image shown in RGB Colour Space

2.2.2 HSV

The HSV (Hue, Saturation, Value) colour space separates the colour information into three components. The hue component represents the dominant wavelength of the colour, ranging from 0 to 360 degrees. Saturation represents the intensity or purity of the colour, ranging from 0 (gray) to 100 (fully saturated). Value represents the brightness or lightness of the colour, ranging from 0 (black) to 100 (white). The *HSV* colour space is often used in image colourisation tasks as

it provides an intuitive way to manipulate colours based on hue, saturation, and value.

Figure 2.2: Image shown in HSV Colour Space

2.2.3 LAB

The *LAB* colour space shown in Figure 2.5, also known as CIE *LAB* or simply *LAB*, is a three-dimensional colour space that separates the lightness component (L) from the chromatic components (A and B). The L component represents the lightness or brightness of a colour, ranging from 0 to 100. The A component represents colours between red and green, where positive values indicate red and negative values indicate green. The B component represents colours between yellow and blue, with positive values indicating yellow and negative values indicating blue. *LAB* is often preferred in image colourisation tasks because it provides more data to work with compared to other colour spaces.

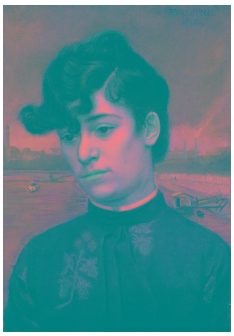

Figure 2.3: Image shown in LAB Colour Space

2.2.4 LUV

The *LUV* colour space, shown in Figure 2.5 is another three-dimensional colour space that separates the lightness component (L) from the chromatic components (U and V). The L component represents the lightness or brightness of a colour, similar to *LAB*. The U and V components represent the chromaticity or colour information. U represents the green-red axis, with negative values indicating green and positive values indicating red. V represents the blue-yellow axis, with negative values indicating blue and positive values indicating yellow. The *LUV* colour space, like *LAB*, is commonly used in image colourisation due to its perceptual uniformity and the ability to represent a wide range of colours.

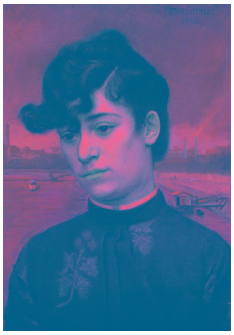

Figure 2.4: Image shown in LUV Colour Space

In summary, different colour spaces have their advantages and are used in image colourisation for various purposes. The RGB colour space is widely used in digital images, while *HSV*, *LAB*, and *LUV* offer alternative representations that provide more flexibility and better perceptual accuracy for colour manipulation in the context of image colourisation.

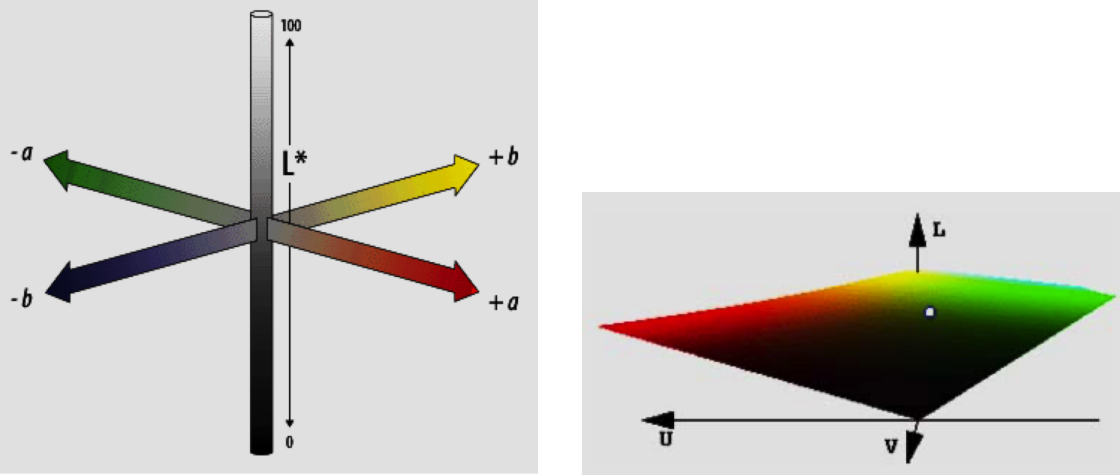

Figure 2.5: CIELAB and CIELUV Colour Spaces

2.2.5 User Involvement

The task of image colourisation has been widely investigated in the literature, with two main approaches identified: user-guided and non-user guided techniques. To guide the colourisation process, user-guided methods require the user to provide manual inputs such as points, strokes, or scribbles. Non-user guided techniques, on the other hand, use an automated approach to colourise images without any manual input from the user [2]. The user-guided techniques require more time and rely on the user's expertise, but they provide greater control over the output. Non-user guided techniques, on the other hand, require less human intervention and are more efficient, but they may lack the accuracy and control that user-guided approaches provide. Despite their differences, these approaches all share the goal of transformation.

2.2.6 Loss and Optimisation Functions

Deep learning models are an essential component of many artificial intelligence applications, and selecting the appropriate loss functions is crucial for achieving high accuracy. The two primary types of loss functions used in deep learning are regression and classification loss functions, which measure the distance between the predicted output and the ground-truth. To improve the performance of deep learning models, optimisation functions are commonly used in conjunction with loss functions, which iteratively updates the parameters of the model based on the loss function to reduce overall loss while enhancing model accuracy.

Binary Cross Entropy Loss Function

The binary cross-entropy is a specific variant of the cross-entropy loss function that is commonly used in machine learning, particularly in classification tasks where the target variable is binary (having two possible values, usually 1 or 0). It is a measure of the dissimilarity between the predicted probabilities and the true binary labels [21].

$$BCE = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \tag{2.1}$$

Semantic Identity Loss Function

The goal of the study by [28] is to build *colorCycleGAN*, a new Generative Adversarial Network (GAN) architecture that intends to solve the single-image colourisation problem. The suggested GAN model uses a two-mirror symmetrical architecture that comprises of a flat convolutional layer after an encoder-decoder framework. Also, a U-net-like structure is incorporated into the network architec-

ture, which aids in maintaining the spatial information of the input image during the colourisation process. The authors use a semantic identity loss function, which makes use of the high-level feature map of the pre-trained VGG network, to further improve the performance of the proposed *colorCycleGAN*. The authors also make use of the Adam optimiser, which reduces the semantic identity loss function and produces a more effective and efficient training.

Cross-Entropy Loss Function

An et al. used a deep convolutional neural network (CNN) with the VGG-16 architecture in their study. The network is made up of numerous convolutional blocks, each of which has two or three convolutional layers. Batch Normalisation and a ReLU activation function are then applied. The mapping between the colour channels and the appropriate ground-truth colours is reversed by the model to accomplish this. The authors experimented with the cross-entropy loss function, which compares the quantised colours of the predicted output with the ground-truth, resulting in a multi-modal cross-entropy loss, in order to minimise the loss between the predicted output and the ground-truth. The network can learn to provide more accurate colourisations with the use of this loss function.

Least Squares Loss Functions

Least Squares Generative Adversarial Networks (LSGANs) were introduced as an alternative to normal GANs in a paper by [17]. The sigmoid cross entropy loss function is typically used for the discriminator in regular GANs, but the authors showed that this can result in disappearing gradients during learning. LSGANs use the least squares loss function for the discriminator to address this issue. LSGANs' primary goal is to minimise divergence by minimising the least squares loss function. LSGANs provide two significant benefits over standard GANs.

Firstly, they can generate higher-quality photos. By penalising samples that are too far away from the decision boundary, even if they are correctly classified, the generator is driven to produce samples that are closer to the real data, resulting in better image quality. Second, LSGANs exhibit more stability during the learning process. When the generator is updated, the penalisation of distant samples generates more gradients, alleviating the problem of disappearing gradients.. This enhanced stability can be attributed to the characteristics of the least squares loss function, which remains relatively flat except at one point, in contrast to the sigmoid cross entropy loss function that saturates for large input values.

2.3 Machine Learning

Machine learning can be divided in four different categories; Supervised, Semi-Supervised, Unsupervised and Reinforcement Learning. In this section, supervised and unsupervised learning will be discussed along with some machine learning algorithm related to image colourisation.

2.3.1 Types

Supervised Learning

Supervised learning requires labels in order to map input data to corresponding output data. The input data can also be referred to as features/predictors whereas the output data would be the target [16]. The goal of supervised learning is to predict outputs of a system given new inputs. During training, the parameters of a network are adjusted to minimise the error between the predicted output and the actual output.

Unsupervised Learning

Unsupervised learning is a machine learning method that analyses unlabelled data. Unsupervised learning acts on datasets that just contain input data, as contrary to supervised learning, which labels the training data with corresponding output values. Since this algorithm must uncover dependent variables, structures, or relationships within the data by itself, the learning process becomes more challenging and unpredictable. Unsupervised learning algorithms are particularly good at completing difficult tasks like anomaly identification and clustering. Unsupervised learning algorithms provide useful insights and enable numerous applications in disciplines such as data mining, pattern identification, and exploratory data analysis by uncovering hidden patterns or grouping similar data points together. [9]

Semi-Supervised Learning

Semi-supervised learning is a category of machine learning that combines clustering and classification. The goal is to improve either task's performance by leveraging information from the other. Unlabelled data can be used to improve classification, whereas knowing that certain data points belong to a single class can assist in the learning procedure in clustering [25].

Reinforcement Learning

Reinforcement learning is a problem-solving technique in which an agent learns how to behave through interaction with a dynamic environment and feedback in the way of rewards or punishments. There are two main techniques to dealing with reinforcement learning problems. To determine the most effective solutions, the first step entails studying diverse behaviours in the environment using

techniques such as genetic algorithms, genetic programming, and advanced search approaches. The second technique includes statistical and dynamic programming approaches to assess the value or utility of specific acts in various states of the world. Through trial and error, these strategies let the agent to learn and improve its behaviour, gradually optimising its behaviours to maximise rewards and achieve desired outcomes [15].

2.3.2 Architectures

Auto-Encoders

Auto-encoders are a family of neural networks, used for unsupervised learning. The main purpose of this architecture is to learn a compressed representation of the input while extracting the most important features [24]. This architecture is made up of a network of encoders and decoders, in which the encoder is used to compress the input into a latent-space representation while the decoder reconstructs the output from this representation.

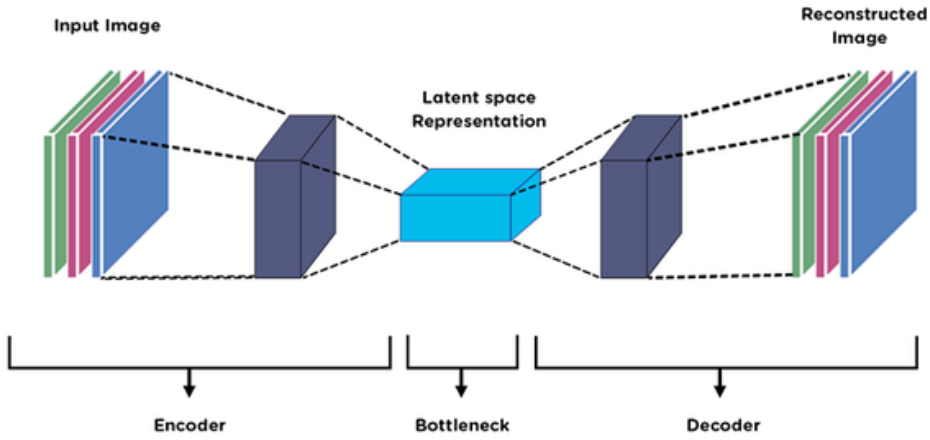

Figure 2.6: Auto-Encoder Architecture

Auto-encoders have been widely used in deep learning for a variety of applica-

tions, including data denoising and dimensionality reduction. These neural networks are made up of several layers of neurons in which the input is encoded into a lower-dimensional representation and then decoded back into the original input using a variety of techniques, including back-propagation [5]. The encoding step acts as a bottleneck for the data, resulting in a compressed representation of the input. This compressed representation can be used for a variety of purposes, including reducing storage space and improving training efficiency. The decoding step seeks to reconstruct the original input by minimising the difference or loss between it and the reconstructed output.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are an extended version of Artificial Neural Networks (ANNs) and are widely known for the impressive analysis in image and video-related tasks such as pattern recognition, classification and segmentation. As explained by [3], a CNN contains several layers which include a convolutional, pooling and fully-connected layer. The first layer convolves through the input image using features extracted by learnable filters and the pooling layer downsamples the extracted features. The final layer results in a fully-connected neural network which is able to perform classification.

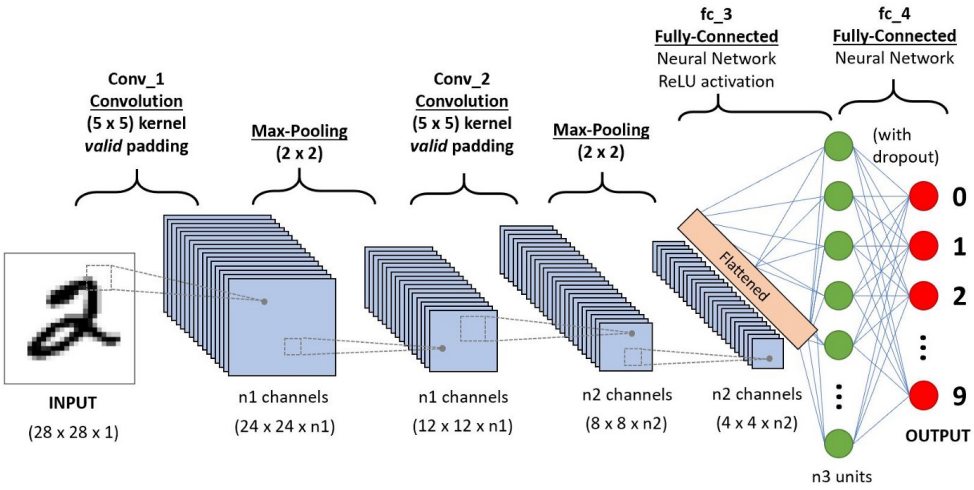

Figure 2.7: CNN Architecture

Generative Adversarial Networks

The job of a Generative Adversarial Network (GAN) is to generate an output which represents the input data, using two smaller networks called the generator and discriminator. As the name implies, the goal of the generator is to generate an output which is indistinguishable from the input, while the discriminator classifies whether the output has been produced from the generator's model or the original input [18]. GANs have been widely used in computer vision scenarios, such as image generation from text and image-to-image translation [12, 19].

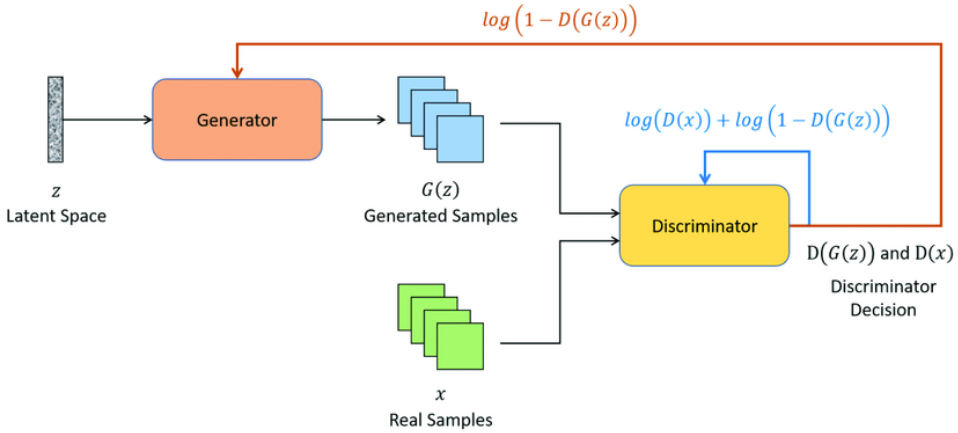

Figure 2.8: GAN Architecture

Memory Augmented Neural Networks

Memory Augmented Neural Networks (MANN) are a new class of learning models which are similar to Recurrent Neural Networks (RNN). MANNs have shown to learn and generalise better than other RNNs [4]. The existence of a memory unit component separates the MANN from other networks. This memory unit can be interacted with using read and write heads and is able to remember uncommon occurrences of colours in images. MANNs have been utilised to store image data for image captioning, generation and summarisation [29].

2.4 Related Work

In the study done by [13], a deep convolutional auto-encoder is implemented. A special multi-skip connections feature is also introduced in this model. The network uses the *YUV* colour space, were the *Y* channel represents the input luminance (gray) image of size 128x128. The *UV* channels represent the output chrominance (hue and saturation) image. The multi-skip connection structure is an implementation of a residual neural network, used to minimise the possibility of the vanishing gradient problem and transmit more data from lower to higher layers. Three types of connections were used in the auto-encoder: no-skip connections, 3-layers-1-skip connections and 6-layers-1-skip connections. A kernel of size 2x2 for the shortcut connections proved to generate more plausible images in terms of colour similarity and details. A combination of the MSE and image gradient loss functions were used to evaluate the predicted images.

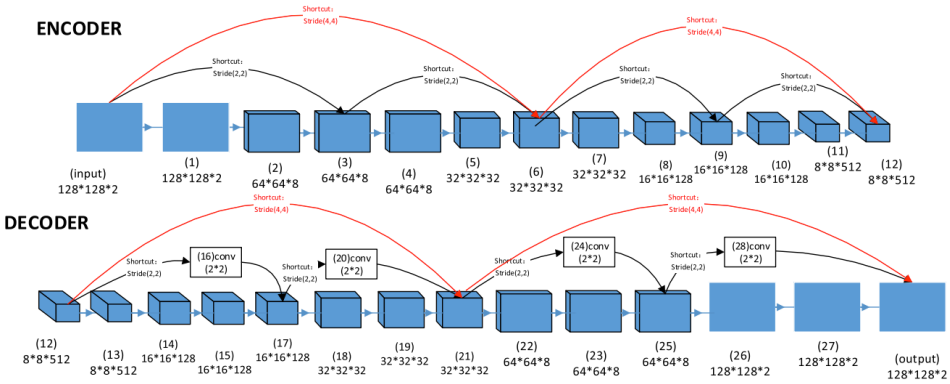

Figure 2.9: Auto-Encoder Model with Multi-Skip Connections

[13] uses four different datasets to train this model, the LFW dataset containing more than 13,000 images of human faces and three datasets from the Large-scale Scene Understanding (LSUN) dataset, particularly the beach, church and bedroom datasets. Each image is cropped to 128 x 128 pixels. Different combinations of the three schemes mentioned above are tested to evaluate the effect of the short-cut connection. Fig 2.10 shows that the adopted multi-skip connection resulted in more useful feature extraction. [13] identified that the ideal kernel size to use with the shortcut connections is 2 x 2 as other kernel sizes resulted in similar colourisations but required more parameters. The use of the mean squared error loss function is included and also uses a gradient with the values 1 and 0.0001 respectively. Results obtained are displayed in Fig 2.11.

Figure 2.10: Feature Extraction using different connections by Jin et al.
(a) Gray Image, (b) Ground Truth, (c) No-Skip (d) 3-layers-1-skip, (e) 3-layers-1-skip and 6-layers-1

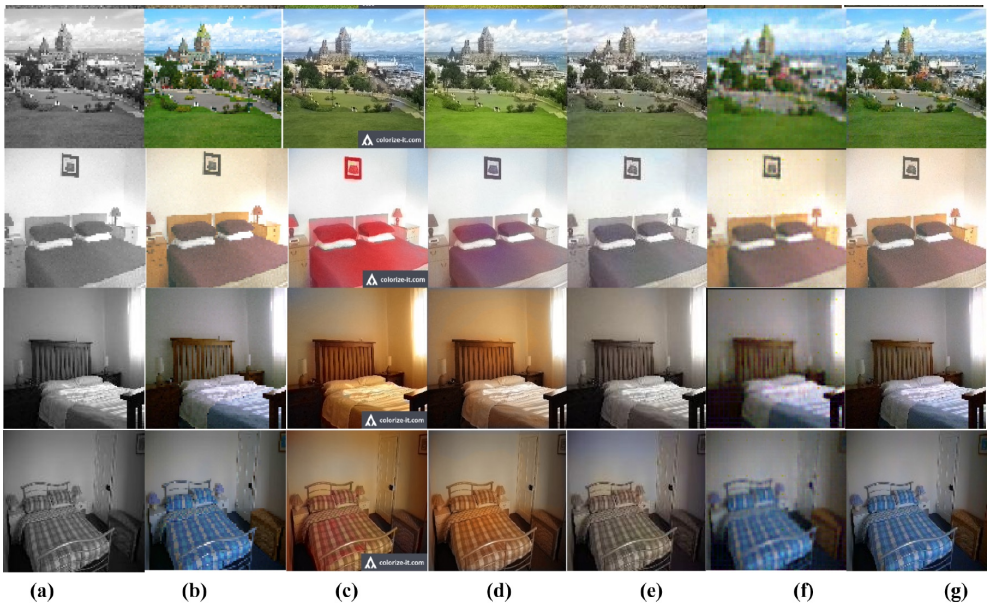

Figure 2.11: Colourisation Results by Jin et al.

Hwang and Zhou conducted a study in which Deep-CNNs were used to colourise black and white images. The image is converted to the CIE *LUV* colour space, where the *L* channel is used as the input, and the *U* and *V* channels are the target values. The CNN generates two arrays corresponding to the *U* and *V* channels. The three channels are then concatenated to form the CIE *LUV* colour space which will represent the predicted image.

The network developed by [30] uses 2 variants of the system, the local and global hints network. The local hints network is responsible for processing the user inputs which is done through sparse points and yielding colour distribution while the global hint network computes global histograms and calculates the average saturation through global statistics. The CIE *Lab* colour space is uses for this system, where the *L* channel is the input image in grayscale and the output is the estimate of the *b* colour channels of the image. [30] adopts the U-Net architecture [20] consisting of 10 convolutional blocks, with each block containing 2 or 3 convolutional layers followed by the ReLU activation function. Each block uses a 3 x 3 kernel and is succeeded with batch normalisation. The first four blocks are continuously halved while the feature dimensions are doubled. The last four blocks repeat the same process but in reverse. Convolution blocks five and six use dilation with a factor of 2 rather than halving the spatial resolution. Another convolutional layer uses a 1 x 1 kernel to map the final convolution block with the output colour. Qualitative and quantitative experiments are carried out in which users were required to colourise images (refer to Fig 2.12) by choosing colours from a suggested colour palette, and utilises PSNR as a measure of similarity between the ground-truth and the colourised image.

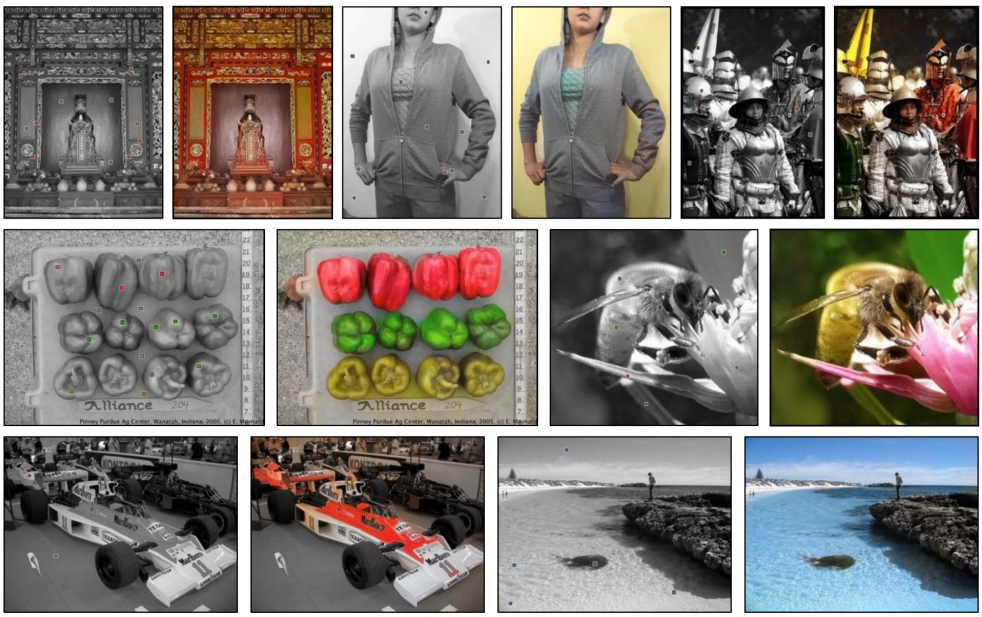

Figure 2.12: User Colourised Results by Zhang et al.

In a study by [14], a custom dataset is created consisting of heritage, historical and cultural image repositories of Nepal. Pre-processing was first performed by collecting a random number of RGB and grayscale images which were used for the train and testing datasets. Low-resolution and degraded images were removed, then cropping and resizing was applied. The images were converted to the CIE *Lab* colour space. The deep learning model used was the CNN, which takes pre-processed grayscale images as inputs. The L component is used as the input during training time, while the *a* and *b* components are the target values. The model generates two arrays corresponding to the *a* and *b* channels. The three channels are then joined together to form the CIE *Lab* representation of the predicted image. The results obtained were good for images which have features such as the sky or river appearing clearly but performed poorly in other images which contained specific objects. Roughly 1,200 images were utilised for training, with 85% allocated for training and the rest for testing. The network's performance was enhanced when particular image attributes, such as skies, trees, and rivers, were present. The mean squared error was employed as the loss function during validation, in conjunction with the Adam optimiser using a learning

rate of 0.001. To enhance accuracy and minimise loss, batch sizes of 10, 20, and 25 were utilised (Fig 2.13). However, a batch size of 20 was ideal as it allowed the network to converge on both loss and accuracy.

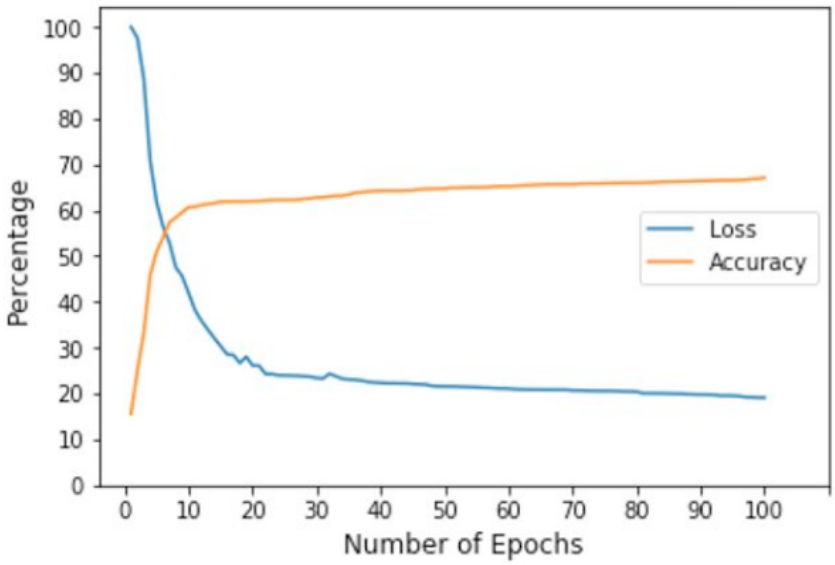

Figure 2.13: MSE/Accuracy vs Epochs - Joshi et al.

In the study ChromaGAN by [26], an end-to-end self-supervised generative adversarial network is developed. [26] utilises the geometric, perceptual and semantic understanding of real scenes to generate a plausible colour image, rather than focusing on aesthetic appeal. The generator is composed of two sub-networks, each receiving a grayscale image of size 224x224 as input. The first sub-network yields the chromanince information, while the second sub-network outputs the computed class distribution vector. The network is composed of three distinct stages. The first stage is shared between both sub-networks and uses the same structure as the VGG-16 while removing the last three fully-connected layers, and uses its pre-trained weights for training without freezing them. During the second stage, each sub-network follows its path. The first path is split into two modules, each combining a convolutional layer followed by Batch Normalisation and the ReLU activation function. The second path is split into four modules with the same structure but followed by three fully-connected layers and provides

class-distribution. The discriminator is used to evaluate how realistic a prediction is. [26] bases their discriminator on the Markovian (PatchGAN) discriminator architecture, which keeps track of high-frequency structural information by focusing on local patches of the image. This is shown in Fig 2.14

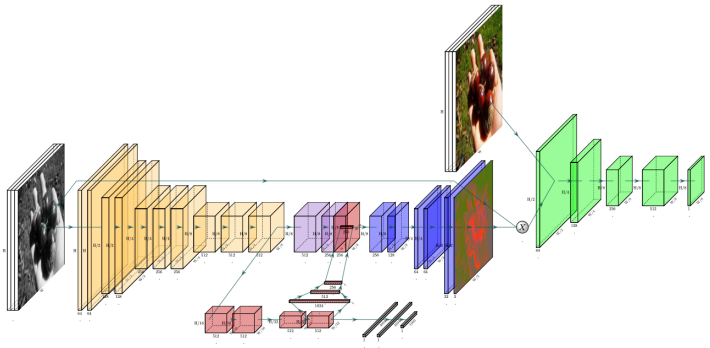

Figure 2.14: Overview of ChromaGAN Architecture

This network is trained on 1.3M images from the ImageNet dataset and uses the Adam optimiser to minimise the objective loss. Joshi et al. performs a quantitative evaluation by using the PSNR measure and comparing it with other state-of-the-art implementations.

The study conducted by Dahlmann et al. quantises an image based on 313 possible AB pairs. The original RGB image is quantised into distinct bins where each pixel corresponds to one of the 313 bins. Each pixel is quantised by converting it to the *Lab* colour space. Once the dataset is quantised, the weighting scheme from [30]'s approach is computed, giving a weight to each bin based on the rarity of the bin. The architecture used is adopted from [30]'s study for use on 64x64 images. A few changes are made to this architecture. The depth of the network outputs at each layer is reduced and one of the convolutional blocks is removed to reduce the number of trainable parameters in the model. Finally, an additional two transpose convolutional blocks were added at the end to up sample the output back to the original width and height. Like the previous paper, the L channel is taken as the input and the a and b channels are the target values. A series of comparison studies were performed to assess the performance of the

model. The two datasets which were used to analyse were a dataset of celebrity faces and a dataset of colourful flowers. Since [6] uses a modified and simpler model architecture and trained for far less time, the results obtained were of a poorer quality. For the second experiment, better results were obtained for the celebrity and flowers datasets, with results comparable to the baseline.

The technique described by Yoo et al.. proposes a model called Memo-Painter, which can produce high-quality colourisations even with limited data. The model consists of a memory network and a colourisation network, where the memory network stores key memory, value memory, and age, and the colourisation network is a conditional GAN. The memory network stores spatial features in the key memory and colour features in the value memory. Two variants of colour features are used: colour distribution and dominant RGB values. The age indicates how long items have been stored in memory without being used. The model outperformed three other baselines, Deep Priors, CIC, and Pix2pix, and consistently produced accurate and vibrant colourisation results, even with extremely limited data.

The Memo-Painter model's success is attributed to its Threshold-Triplet Loss (TTL), a novel approach for training memory networks that stores key and value memories along with age [29]. The model was qualitatively evaluated on five datasets, namely the Oxford102 Flower Dataset, Monster Dataset, Yumi Dataset, Superheroes Dataset, and Pokemon Dataset. MemoPainter was compared to colourisation networks without memory augmentation, and was found to produce superior results, especially in accurately colouring rare instances with only one or a few examples. The study also analysed two hyper-parameters, and found that the model did not overfit to a particular colour threshold or memory size, maintaining stable performance across a range of hyper-parameters measured by Learned Perceptual Image Patch Similarity (LPIPS). The proposed Memo-Painter model outperformed the three other baselines, Pix2Pix, CIC, and Deep Prior, in both user-study evaluations and LPIPS scores for both one-shot and few-shot colouri-

sation scenarios, as shown in Table **??** by Yoo et al.. Specifically, Memo-Painter achieved the highest user-study ratings and the lowest LPIPS scores, indicating its superior performance in accurately colourising images.

	One-Shot		Few-Shot	
	User-study	LPIPS	User-study	LPIPS
Ours	**75%**	**8.48**	**71%**	**1.34**
CIC	10%	9.89	7%	1.80
Pix2Pix	5%	13.47	16%	2.34
Deep Prior	10%	19.26	4%	2.03

Table 2.1: Quantitative Comparisons (LPIPS - lower is better) by Yoo et al.

2.5 Conclusion

The reviewed literature focuses on supervised and unsupervised learning algorithms such as convolutional neural networks (CNN), auto-encoders, memory-augmented neural networks (MANN), and generative adversarial networks, among others (GAN). These algorithms have successfully colourised grayscale images, but certain implementations require a significant amount of computational power to produce high-quality outputs.

As a result, the goal of this research is to find techniques that are less computationally expensive while maintaining or improving output accuracy. The study's goal is to provide insights into less computationally intensive methods that can be applied to colourisation tasks, ultimately making the colourisation process more accessible and practical for wider application in various fields.

Research Methodology

3.1 Introduction

This chapter presents an approach for automatically colourising grayscale images, along with the implementation details and any unsuccessful attempts. In the first part, the chapter explains the process of gathering and preparing the dataset for colourisation. This includes selecting appropriate images, pre-processing the data, and extracting the grayscale and colour images. The second part then describes the implementation of the colourisation approach, which involves training a neural network using auto-encoders and GANs to learn the mapping between grayscale and colour images. The chapter also discusses any challenges encountered during the implementation.

3.2 Pipeline

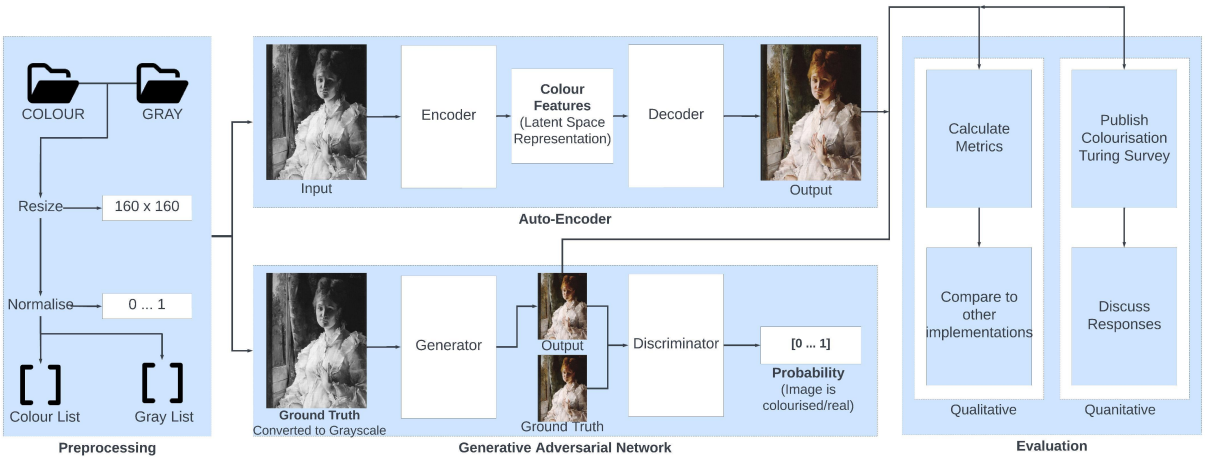

Figure 3.1: Pipeline

The research methodology used in this study follows the pipeline depicted in Figure 3.1. The framework is made up of three major parts: preprocessing, networks (including an auto-encoder and a GAN), and evaluation. Several tasks are performed during the preparation stage, such as image scaling, normalisation, and conversion to an appropriate colour space. These preparations are required to allow the model to be trained effectively using the photos. The second component is concerned with the setup of the networks, which are specifically developed to address the colourisation issue. To achieve the task of black and white image colourisation, two key models, namely an auto-encoder and a GAN (Generative Adversarial Network), are employed. Both models operate on the preprocessed inputs and generate colourised outputs. Auto-encoders are selected for their ability to enhance training efficiency by reducing dimensions and preserving data through skip-connections during decoding. GANs are also utilised due to their popularity in computer vision tasks, leveraging the generator-discriminator framework to validate and refine the model continuously [7]. Lastly, the models are evaluated using both qualitative and quantitative methods in the final stage. The qualitative evaluation involves comparing metrics across different implementations, and the

quantitative evaluation involves survey data analysis.

3.3 Dataset

The availability of training data for converting grayscale images to coloured ones is abundant due to the simplicity of the task. In this study, the primary source of training data was a dataset of war images. However, gathering these images posed challenges as manual curation was required. To address this, the Python library serpapi (Google Images) was utilised to retrieve multiple images through a Google search. Despite this effort, the number of obtained images was considerably reduced, making it insufficient for building a dataset. As a result, the main dataset used for training consisted of 4,117 antique portraits (Figure 3.2) which was retrieved from [1].

Figure 3.2: Sample from Dataset

[1] https://www.kaggle.com/datasets/karnikakapoor/art-portraits?select=Portraits

3.4 Limitations

Several restrictions were encountered during the course of this research. One such limitation was the significant hardware resources required to properly train two colourisation models. Despite using a MacBook Air with an M1 chip, the available processing power was insufficient to satisfy the requirements of training the models, especially the GAN model. To reduce the workload from the MacBook, Google Colab was also utilised in instances were more processing power was required, such as training the GAN. Furthermore, acquiring war images for the dataset was an additional challenge, as these exact images proved difficult to locate. As a result, an alternate dataset had to be picked, even though it still covered a significant percentage of the required image colourisation focus. Finally, the process of hyper-parameter tuning created a significant challenge since it required much more processing power than simple model training. This entailed training the same model many times with different parameters, which increased the processing requirements even more.

3.5 Preprocessing

The preprocessing step is a critical part of preparing data for machine learning models, such as an auto-encoder and a GAN. This step involves loading images from a specified directory and converting them into a float32 format. The function reads images from two folders, namely ”COLOUR” and ”GRAY,” which contain ground-truth images in RGB and corresponding grayscale images, respectively.

To ensure that all images are of a consistent size, the images are resized to a size of 160, if required. Additionally, each image is normalised to a value

between 0 and 1. The normalisation is necessary since the images are stored in float format. By normalising the data, the range of the values in each image is standardised, making it easier to compare images and draw meaningful insights.

After normalisation, the images are added to their respective lists. The first list contains the resized or original RGB images, while the second list contains the resized or original grayscale images. By separating the images into two lists, it becomes easier to train both the auto-encoder and GAN models since each model requires a different input format.

3.6 Proposed Method

This study proposes an implementation of multiple networks by combining the key-parts from other state-of-the-art implementations. By using the implementation from [2] as a foundation for the auto-encoder, the use of auto-encoders and GANs was extended to the colourisation of historical portraits, landscapes and war images.

Both networks were built using Python 3.10. Keras was the library of choice for machine learning due with Tensorflow being used as its backend. Other libraries were also used such as numpy and scikit.

3.6.1 Description of Auto-Encoder

As explained in Section 2.3.2, the implementation of an auto-encoder is comprised of two main parts: the encoder, which reduces the size of the input image, and the decoder, which increases the size of the image to reconstruct the input.

[2] https://www.kaggle.com/code/theblackmamba31/autoencoder-grayscale-to-color-image

This auto-encoder uses a U-Net as its underlying architecture. The encoder is made up of a series of convolutional layers, each applying a set of filters to the input image to extract features. Each layer is interconnected with pooling layers to reduce the size of the input image by a factor of 2 in each dimension.

The decoder is made up of a series of transposed convolutional layers, which increase the size of the image. This decoder uses skip-connections by concatenating the output of each layer with corresponding layer output by the encoder. By making use of skip-connections, high-level features are preserved during the up-sampling process. The input image passed to the encoder is a grayscale image which is compressed, generating the latent space representation, and then passed to the decoder to reconstruct the image in colour.

Architecture

As shown in Figure 3.3, the encoder consists of 5 convolutional layers, each layer using a kernel of size 3 and reducing the image data with a stride of 2. The first 2 layers use 128 filters and do not apply batch normalisation or activation functions. The remaining 3 layers use 256, 512 and 512 filters respectively and apply batch normalisation along with the leaky relu activation function. As mentioned previously, the layers output by the decoder are concatenated with the corresponding outputs from the encoder. The 5 layers in the decoder use transposed convolutional layers with 512, 256, 128 and 128 filters respectively, each of which are doubled in dimensions. The final convolutional layer outputs an RGB image of size 160 x 160.

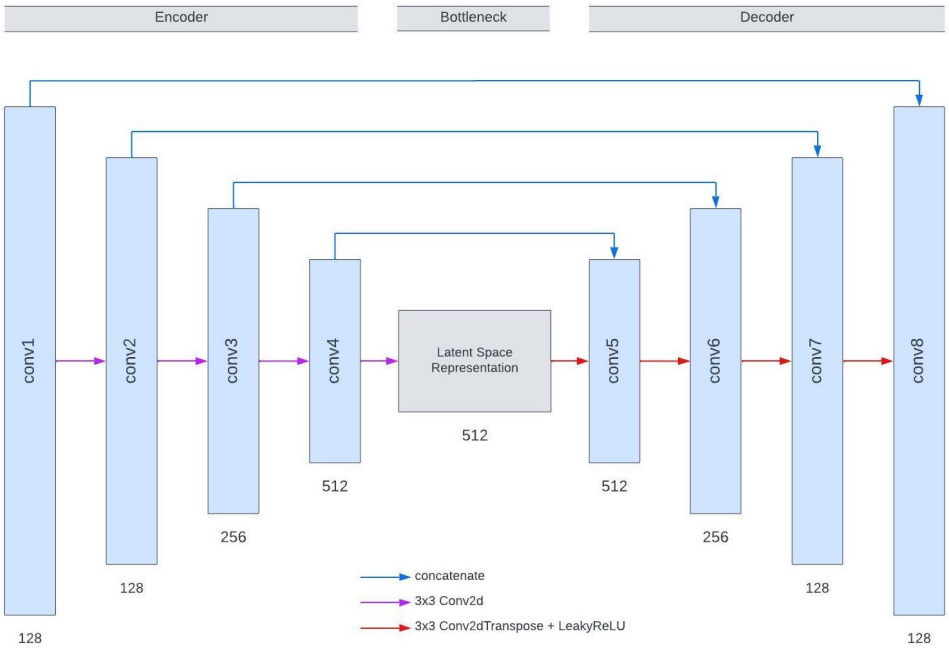

Figure 3.3: Auto-Encoder Model with U-Net Architecture

For this model, the Mean Absolute Error (MAE) is employed as the chosen loss function for the auto-encoder. The MAE measures the average magnitude of the absolute errors by summing them and subsequently dividing by the total number of errors, as denoted in the equation provided:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.1}$$

The resultant computed value signifies the MAE associated with the colourised image. A reduced value of MAE is indicative of heightened accuracy and superior performance, suggesting a closer alignment between the colourised image and the ground truth reference.

3.6.2 Generative Adversarial Network

The GAN is made up of two components: the generator and the discriminator. The generator creates a colourised image, while the discriminator determines whether the output generated by the generator is the true image or a colourised image. The generator is made up of several convolutional blocks, each consisting of three convolutional layers with different kernel sizes and strides for down-sampling.

The bottleneck employs the *tanh* activation function and is followed by several transposed convolutional blocks, with the final layer producing an image with the same size as the input.

On the other hand, the discriminator model comprises four convolutional layers, each followed by another convolutional layer with the same number of filters. The final layer uses the *sigmoid* function and outputs a single value indicating the probability that the input image is genuine.

Architecture

The generator is composed of 3 convolutional blocks, each with a different set of filter sizes: (16, 32, 32), (32, 64, 64), and (64, 128, 128). The first layer in each block has a kernel of 5, while the remaining 2 layers have a kernel size of 3. The bottleneck of the generator has 128 filters and a kernel size of 3. The generator also has 3 transposed convolutional layers that reverse the process and output a generated image.

On the other hand, the discriminator consists of 4 convolutional layers with sizes 32, 64, 128, and 256, followed by a flatten layer and 4 dense layers that result

in a single scalar output.

The generator loss function calculates the mean squared difference between the true target values and the predicted values. This loss function measures the discrepancy between the generated output and the desired output, encouraging the generator to produce more accurate and realistic colourised images. The generator aims to minimise this loss by adjusting its parameters through back-propagation and gradient descent during training.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.2}$$

On the other hand, the discriminator loss function incorporates the least squares loss function [17] in a GAN framework. It calculates the mean squared difference between the true target values of the discriminator and the predicted values. The discriminator's goal is to distinguish between real and generated colourised images. The loss function penalises the discriminator for incorrect classifications of both real and generated images. By minimising this loss, the discriminator learns to differentiate between real and fake images effectively.

3.7 Evaluation

3.7.1 Quantitative Analysis

Several metrics, such as the Mean Squared Error (MSE), Structural Similarity Index Method (SSIM), and Peak Signal-to-Noise Ratio, were used to objectively evaluate the quality of colourisation (PSNR). The results obtained from these metrics will be further discussed in the next chapter.

3.7.2 SSIM

The SSIM metric has become widely used [8] as a means to measure image quality and assess the effectiveness of image processing algorithms and systems. This comprises image elements such as brightness, contrast, and saturation [23]. The SSIM index is calculated as the sum of three variables: brightness, contrast, and saturation. The SSIM formula is denoted as:

$$SSIM(x,y) = (l(x,y) \cdot c(x,y) \cdot s(x,y)) \tag{3.3}$$

where $l(x,y)$ represents the luminance component, $c(x,y)$ represents the contrast component, and $s(x,y)$ represents the saturation component. The overall SSIM index provides a comprehensive assessment of the similarity between the two images.

Figure 3.4: SSIM

Figure 3.4 displays the SSIM value for the auto-encoder, indicating a more accurate colourisation when compared to the GAN.

3.7.3 PSNR

The PSNR measure is commonly used to assess the quality of image and video compression [23]. It calculates the ratio of the greatest feasible signal power to the power of the distorting noise that influences the representation's quality. The PSNR is often reported in decibels (dB) because it accounts for the vast dynamic range of signals, which can vary greatly in quality.

PSNR serves as an approximation of human perception when assessing the quality of reconstructed images in comparison to compression codecs. It quantifies the difference between the original signal (referred to as the "signal") and the error introduced during compression or distortion (referred to as the "noise"). The formula for calculating PSNR is expressed as:

$$PSNR = 10\log_{10}\left(\frac{peakval^2}{MSE}\right) \tag{3.4}$$

Here, *peakval* represents the maximum possible pixel value, and MSE denotes the mean squared error between the original and reconstructed images. This formula allows for a quantitative assessment of the reconstruction quality based on the power ratio between the signal and the distortion.

(Auto-Encoder) PSNR: 27.35

(GAN) PSNR: 27.17

Figure 3.5: PSNR

In Figure 3.5, the PSNR value for the auto-encoder indicates a slight advantage in accuracy compared to the GAN.

3.7.4 Qualitative Analysis

A visual comparison of the colourised images and their corresponding ground-truth photos was undertaken to assess the quality of the colourised images. Two separate online surveys were conducted to evaluate the colourisation performance of the auto-encoder and the GAN models. Each survey consisted of ten questions and aimed to assess the quality of colourisation achieved by the respective models.

During the survey, each question consisted of a set of four images. Among these images, three were ground truth images representing the original state, while the remaining one was a result of colourisation by an algorithm defined in Section 2.3.2 . To ensure fairness and impartiality, a Python script was utilised to randomly select the images for each question, thereby avoiding any potential bias towards favouring certain colourised images over others.

The goal of the test was to determine whether or not participants could tell which of the four photographs shown had been coloured using an algorithm. During the course of a week, 53 and 32 responses were collected for the auto-encoder and GAN surveys respectively. To maintain anonymity, participants' identities were kept private, and no personal information was recorded during the survey.

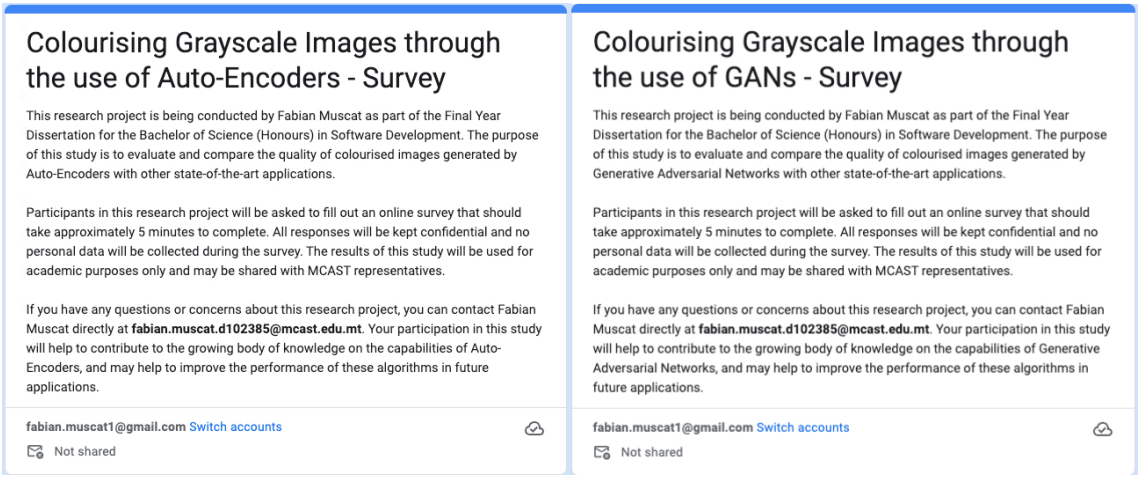

Figure 3.6: Auto-Encoder and GAN Surveys

3.8 Conclusion

In conclusion, an effective approach was employed to address the challenge of colourisation, which involved the implementation of two distinct networks: an auto-encoder and a Generative Adversarial Network (GAN). The implementation process was comprehensively described, outlining the steps taken to achieve successful colourisation results. Additionally, details regarding the dataset utilised for training and evaluation purposes were provided, emphasising its significance in obtaining accurate outcomes.

To thoroughly evaluate the performance of the colourisation models, a combination of quantitative and qualitative analysis techniques was employed. Quantitative assessment involved the utilisation of similarity metrics. Furthermore, a qualitative analysis was conducted through a colourisation Turing test, in the form of an online survey.

Analysis of Results and Discussion

4.1 Introduction

This chapter will present the outcomes derived from the methodologies proposed in the preceding chapter. The objective is to furnish the reader with a comprehensive understanding of the obtained results. This encompasses both quantitative data, acquired through the utilisation of similarity metrics such as MSE, SSIM and PSNR, and qualitative assessments obtained from a colourisation turing test conducted by means of 2 online surveys. Furthermore, visual observations will also be incorporated as a valuable component of the analysis. These results will be effectively organised and presented through the inclusion of tables, graphs, and side-by-side image comparisons. These visual representations aim to offer a concise and accessible means of conveying the key findings of the research.

4.2 Comparison of Results with other Implementations

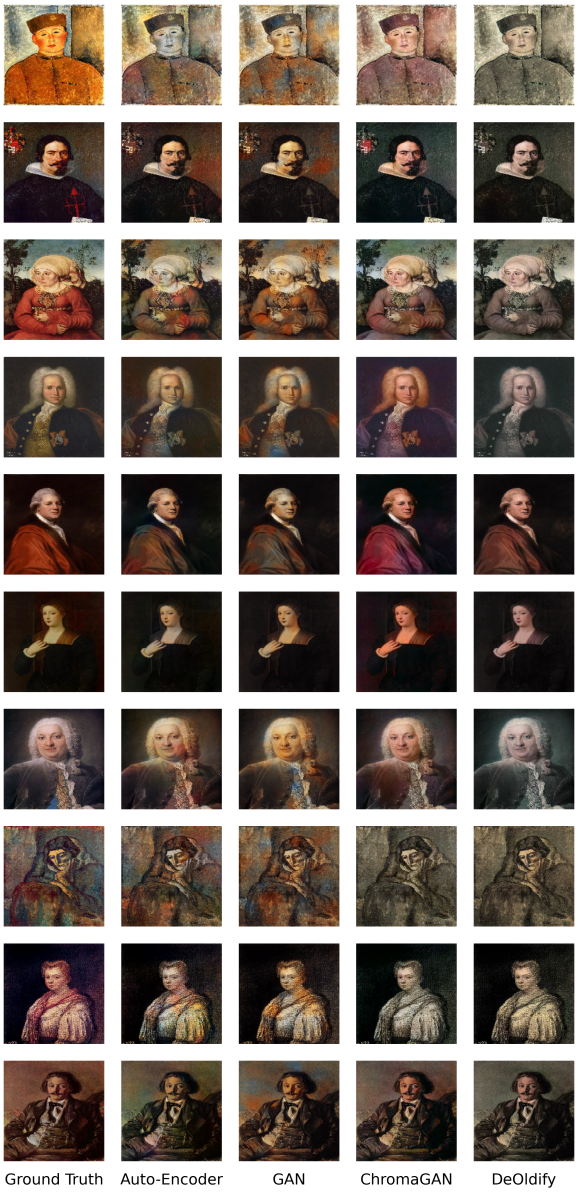

Figure 4.1: A Selection of 10 Colourised Images from Different Models

A set of images from the dataset underwent testing using both the Auto-Encoder and GAN models, as depicted in Figure 4.1. Additionally, to enhance the comparison and evaluation process, these images were also evaluated using alternative implementations [22, 26]. Comparing the results to the ground truth as a refer-

ence point, it is evident that for images 8 and 10, both the auto-encoder and GAN produced more visually appealing colourisations than other implementations such as ChromaGAN and DeOldify. To facilitate an objective comparison, Tables 4.1, 4.2, and 4.3 present the MSE, SSIM, and PSNR values, respectively, for each image when compared to the ground truth.

MSE				
Image	**Auto-Encoder**	**GAN**	**ChromaGAN**	**DeOldify**
1	1335.054	1065.909	1727.778	2175.25
2	127.196	127.522	246.92	200.729
3	330.482	322.562	403.789	662.725
4	73.575	61.333	178.024	129.658
5	178.238	203.852	236.83	184.11
6	47.475	44.157	94.667	102.55
7	126.907	164.073	104.55	93.99
8	176.359	197.614	181.819	157.423
9	151.896	146.426	273.717	200.377
10	109.675	107.328	125.859	204.075

Table 4.1: MSE Values for Images 1 - 10

A lower MSE value suggests that the colourised output closely resembles the ground truth. Upon analysing the MSE values presented in the table, it is apparent that the auto-encoder and DeOldify model show more consistency, as their MSE values consistently fall within a similar range. This consistency implies that the colourisation results produced by these methods are consistently close to the ground truth images, demonstrating their reliability and stability.

In contrast, the MSE values obtained for the GAN model display a slightly greater degree of variation across the different images. This suggests that the GAN model's colourisation performance may exhibit more variability, with some images achieving lower MSE values closer to the ground truth, while others may have relatively higher MSE values indicating a larger deviation from the ground truth.

SSIM				
Image	**Auto-Encoder**	**GAN**	**ChromaGAN**	**DeOldify**
1	0.846	0.847	0.789	0.785
2	0.924	0.917	0.825	0.893
3	0.945	0.935	0.902	0.905
4	0.939	0.938	0.811	0.931
5	0.737	0.743	0.532	0.689
6	0.868	0.845	0.646	0.813
7	0.951	0.936	0.89	0.95
8	0.937	0.929	0.92	0.936
9	0.896	0.871	0.789	0.866
10	0.937	0.916	0.862	0.903

Table 4.2: SSIM Values for Images 1 - 10

Higher SSIM values indicate a greater similarity between the colourised output and the ground truth image in terms of both structural information and visual perception. Upon examining the SSIM values presented in the table, several observations can be made. The auto-encoder and GAN models consistently exhibit high SSIM values across the majority of the images, indicating that their colourisation outputs closely resemble the ground truth images in terms of structural details and visual appearance. This consistency suggests that these models are capable of capturing and preserving the important features and characteristics of the original images during the colourisation process.

Comparing the SSIM values of ChromaGAN and DeOldify with the other models, it is evident that they achieve relatively lower SSIM values for most of the images. This suggests that their colourisation results may deviate slightly from the ground truth images in terms of structural similarity and visual fidelity.

PSNR				
Image	**Auto-Encoder**	**GAN**	**ChromaGAN**	**DeOldify**
1	16.876	17.854	15.756	14.756
2	27.086	27.075	24.205	25.105
3	22.939	23.045	22.069	19.917
4	29.463	30.254	25.626	27.003
5	25.621	25.038	24.386	25.48
6	31.366	31.681	28.369	28.021
7	27.096	25.98	27.938	28.4
8	25.667	25.173	25.534	26.16
9	26.315	26.475	23.758	25.112
10	27.73	27.824	27.132	25.033

Table 4.3: PSNR Values for Images 1 - 10

Higher PSNR values indicate better image quality, as it implies a lower level of distortion or noise in the colourised output. Analysing the PSNR values provided in the table, several observations can be made. The auto-encoder and GAN models consistently exhibit high PSNR values across the majority of the images, indicating that their colourised outputs have relatively low levels of noise and distortion compared to the ground truth images. This suggests that these models are capable of preserving important image details and minimising information loss during the colourisation process.

In contrast, the PSNR values of ChromaGAN and DeOldify tend to be slightly lower for most of the images. This suggests that the colourised outputs generated by these models may have a slightly higher level of noise or distortion compared to the ground truth images.

4.3 Colourisation Turing Test Results

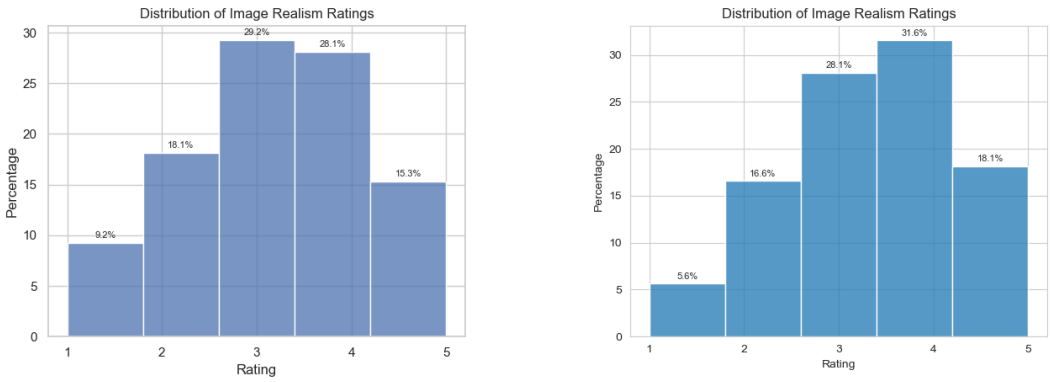

Figure 4.2: Distribution of Image Realism Ratings (Auto-Encoder, GAN)

The histograms in Figure 4.2 illustrate the distribution of ratings provided by participants for a different set of images for the auto-encoder and GAN respectively. It can be observed that both the auto-encoder and GAN display normalised histograms, indicating a relatively balanced distribution of ratings. The average rating of 3 suggests that the majority of participants perceive the images as moderately realistic.

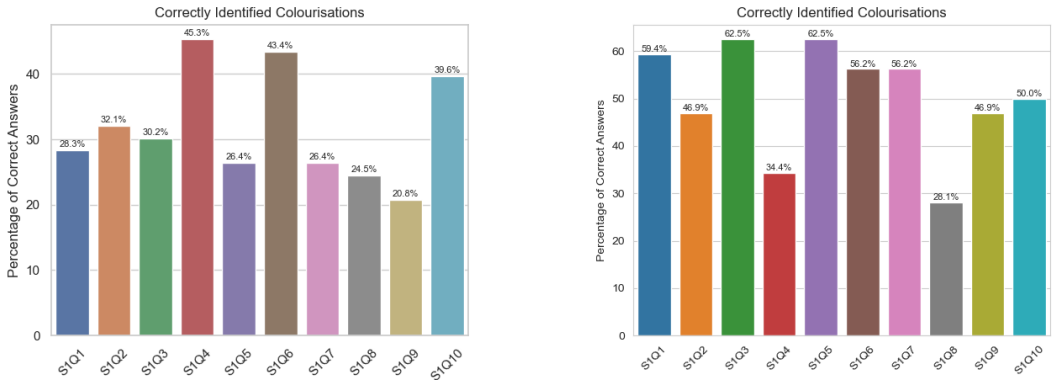

Figure 4.3: Correctly Identified Colourisations (Auto-Encoder, GAN)

Figure 4.3 displays the bar charts representing the accurate identification of colouri-

sations by participants for each question from 1 to 10. Lower values on the chart indicate instances where participants were unable to correctly identify the colourised image, suggesting that the corresponding model generated a realistic colourisation. Regarding the auto-encoder, question 9 received the highest number of incorrect identifications, while for the GAN, it was question 8. This indicates that approximately 79.2% and 71.9% of participants, respectively, provided incorrect guesses for these questions, showing that a more realistic colourisation was produced.

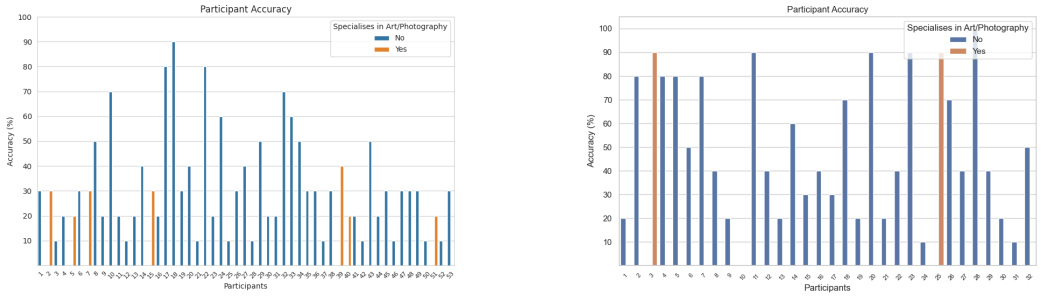

Figure 4.4: Correctly Identified Colourisations for all Images (Auto-Encoder and GAN)

Figure 4.4 displays the bar charts representing the accurate identification of colourisations by participants for each image. Overall, the participants were able to identify colourised images correctly for the GAN with a higher accuracy than the auto-encoder.

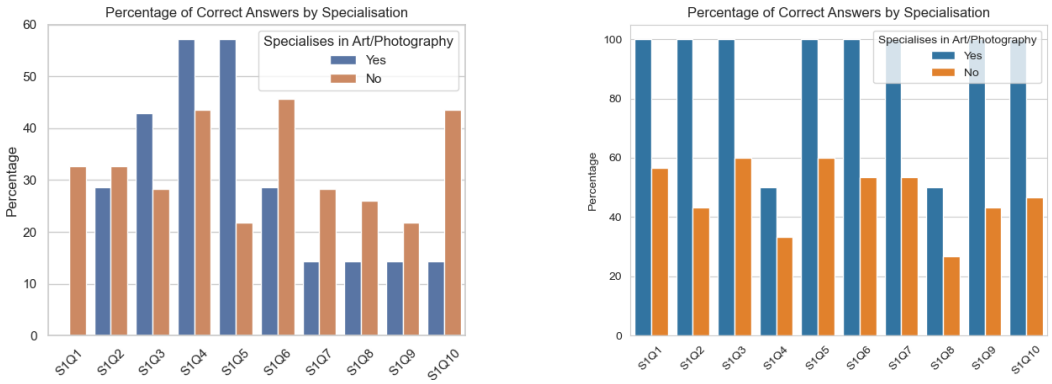

Figure 4.5: Correctly Identified Colourisations grouped by Specialisation (Auto-Encoder and GAN)

The participation of individuals with a specialisation in art or photography was relatively limited compared to those without such specialisation. Specifically, out of the 53 participants in the auto-encoder group, only 7 had a specialisation in the field, while for the GAN group, only 2 out of 32 participants were specialised. Upon examining Figure 4.5, it becomes apparent that non-specialised participants had a higher success rate in correctly identifying colourised images compared to those with specialisation.

4.4 Research Questions Evaluation

1. **What are existing image colourisation algorithms?**
   In Section 2, various algorithms, including auto-encoders, GANs, and MANNs, were examined and explored. Specifically, an auto-encoder with skip-connections and a GAN were implemented to evaluate their performance, alongside other existing implementations such as ChromaGAN and DeOldify [22, 26]. To assess their effectiveness, similarity metrics were employed for comparison purposes.

2. **What types of images are easier/more difficult to colourise?**
   Given the difficulties encountered in obtaining multiple datasets, the models were evaluated using a single dataset. Nevertheless, an analysis of similarity metrics, including MSE, SSIM, and PSNR, revealed that specific portrait images posed greater difficulties for accurate colourisation. For instance, images with complex backgrounds, multiple objects, or diverse colour variations proved to be more challenging. On the other hand, certain images were easier to achieve accurate colourisation.

3. **What are the strengths and weaknesses of each algorithm?**
   Through the evaluation process, the strengths and weaknesses of each algorithm were identified. The auto-encoder showcased consistent performance

and yielded satisfactory results in terms of MSE and SSIM, even with a relatively shorter training duration of only 50 epochs. Despite its limited training, it managed to produce plausible outcomes. On the other hand, the GAN outperformed the auto-encoder in terms of SSIM but displayed a broader range of MSE values. Achieving its desired accuracy required a longer training duration of 150 epochs and was more computationally expensive.

4. **Can a hybrid algorithm combining the strengths of each algorithm be developed to improve to improve colourisation?**

   The dissertation did not specifically explore the development of a hybrid algorithm combining the strengths of existing algorithms. However, the comparison and evaluation of different algorithms provided valuable insights into their respective strengths and weaknesses. This information can serve as a basis for future research and the potential development of a hybrid algorithm that leverages the strengths of each individual algorithm to improve overall colourisation performance.

4.5 Conclusion

This chapter provided a comprehensive analysis and discussion of the results, encompassing both quantitative and qualitative data. The examination involved an assessment of similarity metrics including MSE, SSIM, and PSNR for both the auto-encoder and GAN models. Additionally, a comparison was made between these models and other advanced implementations, namely ChromaGAN and DeOldify [22, 26]. Finally, the research questions posed at the beginning of the study were addressed and evaluated.

Conclusions and Recommendations

5.1 Conclusion

The primary objective of this study was to enhance the efficiency of image colourisation by implementing an auto-encoder and a GAN. The aim was to explore existing algorithms and develop a more effective method for colourising grayscale images. Extensive research was conducted, addressing relevant literature on colour spaces, various algorithms, and state-of-the-art implementations. The study involved several steps, including data preprocessing, model development, and evaluation using a combination of similarity metrics and colourisation turing tests.

The literature review extensively examined the existing body of knowledge on image colourisation. It explored different colour spaces and their applications, providing insights into the complexities involved in the colourisation process. Various algorithms, including auto-encoders, GANs, and other state-of-the-art implementations, were analysed to identify their strengths and weaknesses. The review served as a foundation for the development and evaluation of the auto-encoder and GAN models in this study.

The study employed a systematic approach to achieve its goals. Initially, the dataset of grayscale images was preprocessed, involving the conversion of images to grayscale and normalisation for optimal input to the models. The auto-encoder was designed with skip-connections to mitigate information loss during the decoding process. On the other hand, the GAN model utilised a generator and discriminator architecture and employed the least squares loss function. These

models were trained using appropriate training procedures and optimisation techniques to achieve effective colourisation.

The results obtained from the evaluation of the colourised images were diverse and provided valuable insights into the performance of the developed models. The auto-encoder demonstrated consistency and satisfactory outcomes, as indicated by the similarity metrics such as MSE and SSIM. These results were achieved even with a relatively short training period of 50 epochs. Conversely, the GAN model exhibited better performance in terms of SSIM, but its MSE values were comparatively more widely distributed. Achieving higher accuracy with the GAN required an extended training period of 150 epochs.

The evaluation of the colourised images also involved participant ratings and identification tests. Notably, certain types of images posed greater challenges for accurate colourisation, particularly those with multiple objects in complex backgrounds or varying colours. Conversely, other image types were relatively easier to colourise. Furthermore, the participants' specialisation in art or photography was considered, and it was observed that non-specialised participants tended to have higher accuracy in identifying colourised images compared to those with specialisation.

5.2 Recommendations

1. **Expand Dataset and Training Variations**: By increasing diversity and size of the dataset, the complexities and variations present in real-world images can be better represented. Furthermore, variations in the training process, such as different hyper-parameters, architectures, or loss functions, can be introduced to enable a more comprehensive exploration of the colourisation problem space.

2. **User Feedback Integration**: Incorporate user feedback and iterative improvements into the colourisation process. By collecting feedback from users, including artists, photographers, and general users, it is possible to gather valuable insights on the perceived quality and realism of colourised images. This feedback can be used to refine the models and algorithms, making them more aligned with human perception and preferences. User-driven improvements can enhance the overall quality and satisfaction of the colourisation process.

3. **Focus on Complex Image Types**: Pay specific attention to the challenges associated with colourising complex image types. Images with multiple objects, intricate details, or varying colours often pose difficulties for accurate colourisation. Future research should investigate techniques and algorithms specifically designed to handle these complexities. This may involve incorporating advanced neural network architectures, attention mechanisms, or context-aware models to improve the colourisation of portraits, landscapes, or other visually intricate images.

References

[1] Jiancheng An, Kpeyiton Koffi Gagnon, Qingnan Shi, Hongwei Xie, and Rui Cao. Image colorization with convolutional neural networks. In *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–4. Ieee, 2019.

[2] Saeed Anwar, Muhammad Tahir, Chongyi Li, Ajmal Mian, Fahad Shahbaz Khan, and Abdul Wahab Muzaffar. Image colorization: A survey and dataset. *arXiv preprint arXiv:2008.10774*, 2020.

[3] Belhassen Bayar and Matthew C Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 13(11):2691–2706, 2018.

[4] Mark Collier and Joeran Beel. Memory-augmented neural networks for machine translation. *arXiv preprint arXiv:1909.08314*, 2019.

[5] Musab Coşkun, Özal YILDIRIM, UÇAR Ayşegül, and Yakup Demir. An overview of popular deep learning methods. *European Journal of Technique (EJT)*, 7(2):165–176, 2017.

[6] Annika Dahlmann, Cameron Davis, Cole Hudson, Kyle O'Laughlin, and Nathan Tseng. Back to the future: Building upon image colorization via classification, 2021. URL `https://web.eecs.umich.edu/~justincj/teaching/eecs442/projects/WI2021/pdfs/065.pdf`.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Advances in

neural information processing systems. *Curran Associates, Inc*, 27:2672–2680, 2014.

[8] Mohammed Hassan and Chakravarthy Bhagvati. Structural similarity measure for color images. *International Journal of Computer Applications*, 43(14):7–12, 2012.

[9] Kamal Kant Hiran, Ritesh Kumar Jain, Kamlesh Lakhwani, and Ruchi Doshi. *Machine Learning: Master Supervised and Unsupervised Learning Algorithms with Real Examples (English Edition)*. BPB Publications, 2021.

[10] Jeff Hwang and You Zhou. Image colorization with deep convolutional neural networks. In *Stanford University, Tech. Rep.* 2016.

[11] Noor A Ibraheem, Mokhtar M Hasan, Rafiqul Z Khan, and Pramod K Mishra. Understanding color models: a review. *ARPN Journal of science and technology*, 2(3):265–275, 2012.

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[13] Xin Jin, Yide Di, Qian Jiang, Xing Chu, Qing Duan, Shaowen Yao, and Wei Zhou. Image colorization using deep convolutional auto-encoder with multi-skip connections. *Soft Computing*, 27(6):3037–3052, October 2022. doi: 10.1007/s00500-022-07483-0. URL `https://doi.org/10.1007/s00500-022-07483-0`.

[14] Madhab Raj Joshi, Lewis Nkenyereye, Gyanendra Prasad Joshi, SM Riazul Islam, Mohammad Abdullah-Al-Wadud, and Surendra Shrestha. Auto-colorization of historical images using deep convolutional neural networks. *Mathematics*, 8(12):2258, 2020.

[15] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4: 237–285, 1996.

[16] Bing Liu and Bing Liu. *Supervised learning*. Springer, 2011.

[17] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[18] Kamyar Nazeri, Eric Ng, and Mehran Ebrahimi. Image colorization using generative adversarial networks. In *International conference on articulated motion and deformable objects*, pages 85–94. Springer, 2018.

[19] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. Pmlr, 2016.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[21] Usha Ruby and Vamsidhar Yendapalli. Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng*, 9 (10), 2020.

[22] Antoine Salmona, Lucía Bouza, and Julie Delon. Deoldify: A review and implementation of an automatic colorization method. *Image Processing On Line*, 12:347–368, 2022.

[23] Umme Sara, Morium Akter, and Mohammad Shorif Uddin. Image quality assessment through fsim, ssim, mse and psnr–a comparative study. *Journal of Computer and Communications*, 7(3):8–18, 2019.

[24] Mohit Sewak, Sanjay K Sahay, and Hemant Rathore. An overview of deep learning architecture of deep neural networks and autoencoders. *Journal of Computational and Theoretical Nanoscience*, 17(1):182–188, 2020.

[25] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

[26] Patricia Vitoria, Lara Raad, and Coloma Ballester. Chromagan: Adversarial picture colorization with semantic class distribution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2445–2454, 2020.

[27] Hao Wang and Xuedong Liu. Overview of image colorization and its applications. In *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, volume 5, pages 1561–1565, 2021. doi: 10.1109/iaeac50856.2021.9390626.

[28] Yuxuan Xiao, Aiwen Jiang, Changhong Liu, and Mingwen Wang. Single image colorization via modified cyclegan. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3247–3251. Ieee, 2019.

[29] Seungjoo Yoo, Hyojin Bahng, Sunghyo Chung, Junsoo Lee, Jaehyuk Chang, and Jaegul Choo. Coloring with limited data: Few-shot colorization via memory augmented networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11275–11284, 2019. doi: 10.1109/cvpr.2019.01154.

[30] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017.

Appendices

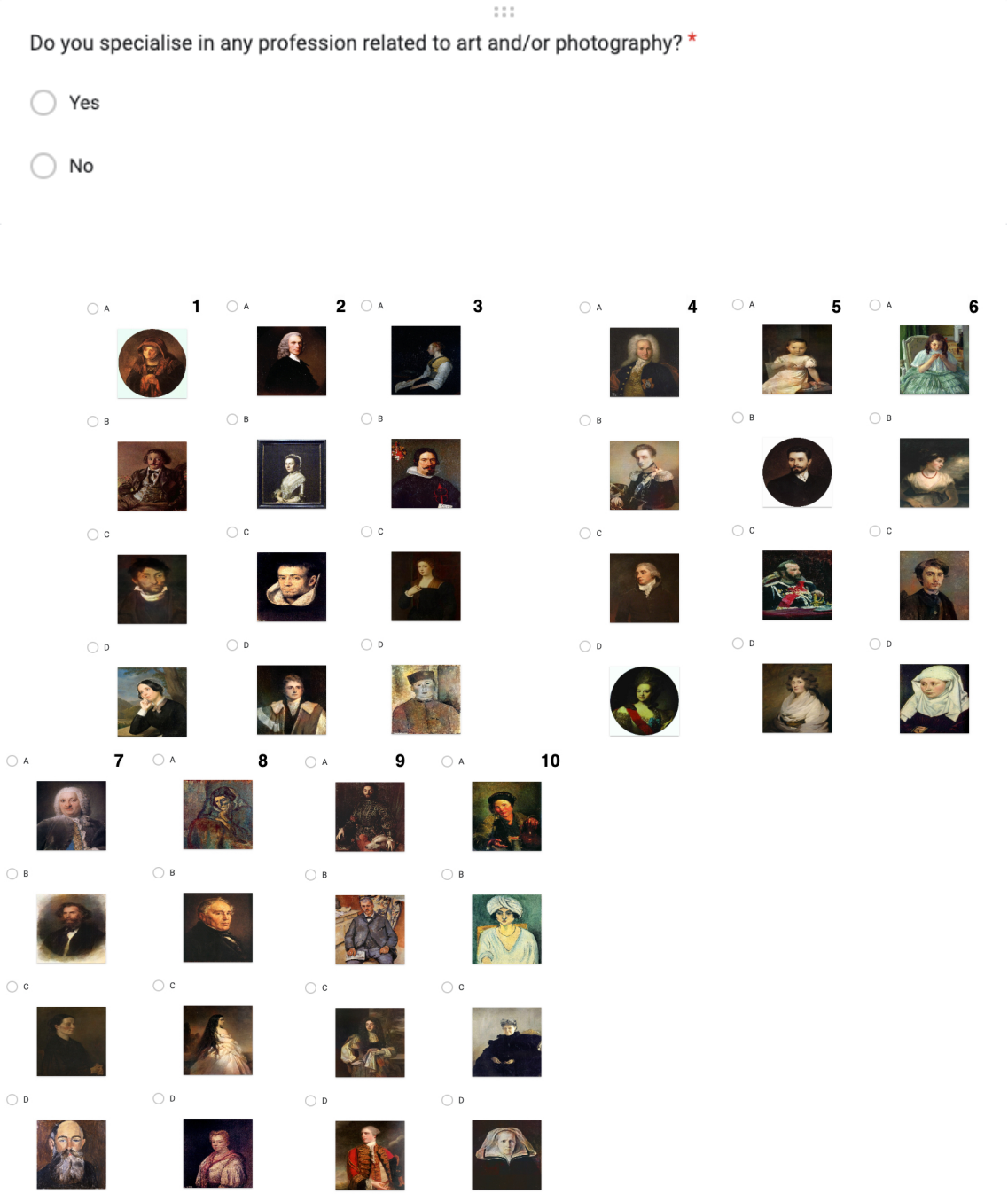

Figure A.1: Images used in colourisation turing test for the auto-encoder

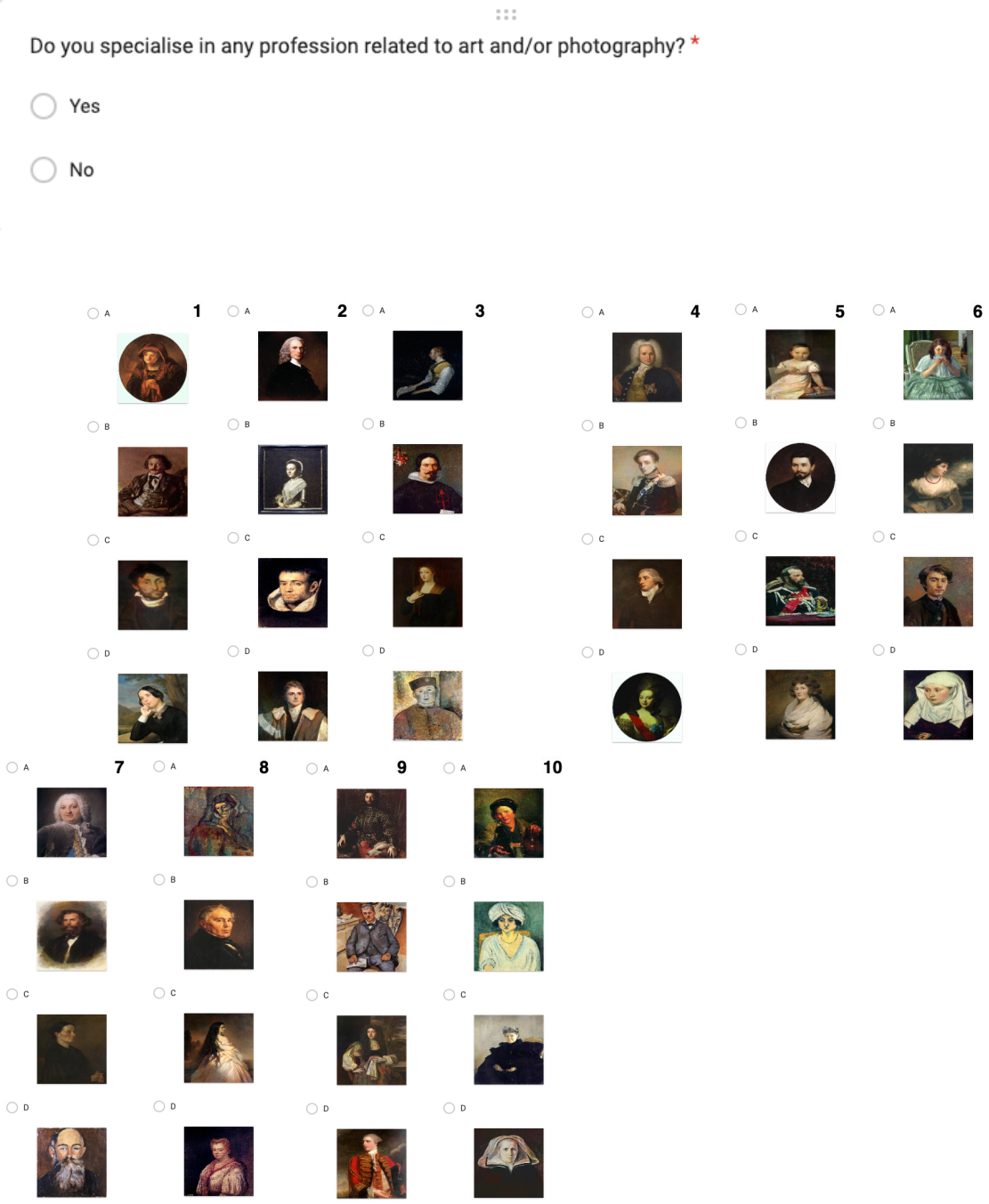

Figure A.2: Images used in colourisation turing test for the GAN