SAVITRIBAI PHULE PUNE UNIVERSITY

A PRELIMINARY PROJECT REPORT ON

Image colorization with OpenCV and Deep Learning

SUBMITTED TOWARDS THE
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING (Computer Engineering)

BY

Sahil Shahane	B191224314
Darshan Zunjurde	B191224355
Vedant Mule	B191224282
Shantanu Wankhare	BECOMPB50

Under The Guidance of

Mrs. Shailaja Lohar

DEPARTMENT OF COMPUTER ENGINEERING
Pimpri Chinchwad College of Engineering And Research
Ravet

A.Y.2023-2024

Pimpri Chinchwad College Of Engineering And Research
DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Project Entitled

Image colorization with OpenCV and Deep Learning

Submitted by

Sahil Shahane	B191224314
Darshan Zunjurde	B191224355
Vedant Mule	B191224282
Shantanu Wankhare	BECOMPB50

is a bonafide work carried out by Students under the supervision of Mrs. Shailaja Lohar and it is submitted towards the partial fulfillment of the requirement of Bachelor of Engineering (Computer Engineering) Project.

Mrs. Shailaja Lohar
Internal Guide
Dept. of Computer Engg.

Dr. Archana Chaugule
H.O.D
Dept. of Computer Engg.

Dr. Abhijit Jadhav
Project Co-ordinator

Prof.Dr. H.U.Tiwari
Principal

EXTERNAL SIGNATURE

PROJECT APPROVAL SHEET

A Project Title

Image colorization with OpenCV and Deep Learning

Successfully Completed By

Sahil Shahane Exam No: B191224314

Darshan Zunjurde Exam No: B191224355

Vedant Mule Exam No: B191224282

Shantanu Wankhare Exam No: B191224506

At

DEPARTMENT OF COMPUTER ENGINEERING
Pimpri Chinchwad College of Engineering And Research
Ravet

A.Y.2023-2024

Mrs. Shailaja Lohar	Dr. Archana Chaugule
Internal Guide	H.O.D
Dept. of Computer Engg.	Dept. of Computer Engg.

Abstract

This paper introduces an effective approach for image colorization using a combination of OpenCV and deep literacy ways. We punctuate the significance of image colorization in varied fields, from Medical Imaging Enhancement, historical photograph restoration to art and entertainment. Our system employs Convolutional Neural Networks(CNNs) to learn the complex connections between grayscale and color images, with a custom colorization head integrated into a pre-trained neural network.

We explore the indefectible integration of OpenCV, a protean computer vision library, for image preprocessing, post-processing, and visualization. This enhances the colorization process and offers an accessible channel for users. We address practical aspects like dataset selection, model training, and hyperparameter tuning, giving perceptivity to smart practices. Performance evaluation demonstrates the superiority of our approach compared to being styles.

This paper is a precious resource for experimenters and interpreters seeking to achieve high-quality image colorization. It presents an effective frame that combines the power of deep knowledge with OpenCV's versatility to deliver accurate and visually fascinating colorization results in various operations.

Acknowledgments

It gives us great pleasure in presenting the preliminary project report on ***'Image colorization with OpenCV and Deep Learning'****.*

I would like to take this opportunity to thank my internal guide ***Mrs. Shailaja Lohar*** *for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful.*

I am also grateful to ***Dr. Archana Chaugule****, Head of Computer Engineering Department, PCCOER for her indispensable support, suggestions.*

In the end our special thanks to ***Prof. Abhijit Jadhav*** *for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for Our Project.*

Sahil Shahane
Darshan Zunjurde
Vedant Mule
Shantanu Wankhare
(B.E. Computer Engg.)

INDEX

List of Figures

List of Tables

CHAPTER 1

SYNOPSIS

1.1 PROJECT TITLE

Image colorization with OpenCV and Deep Learning

1.2 PROJECT OPTION

Internal Project

1.3 INTERNAL GUIDE

Mrs. Shailaja Lohar

1.4 SPONSORSHIP AND EXTERNAL GUIDE

No sponsorship.

1.5 TECHNICAL KEYWORDS (AS PER ACM KEYWORDS)

Technical Key Words:

- Mathematics of Computing
- Colorization
- Vision for Graphics
- CNNs
- Self-supervised learning

1.6 PROBLEM STATEMENT

Developing a robust colorization system using advanced deep learning techniques like Generative Adversarial Networks (GANs) to accurately add realistic colors to grayscale images and providing a user-friendly interface for practical utilization.

1.7 ABSTRACT

Image colorization, the process of adding color to black and white or grayscale images, has been a long-standing challenge in computer vision and image processing. This paper presents a comprehensive approach to image colorization using a combination of the OpenCV library and deep learning techniques. We discuss the background and significance of image colorization, highlighting its applications in Medical Imaging Enhancement, historical photo restoration, art, and entertainment. The proposed method leverages Convolutional Neural Networks (CNNs) to learn the intricate relationships between grayscale images and their corresponding color counterparts. We explore the architecture of the deep learning model, which involves a pre-trained neural network as the backbone and a custom colorization head. This approach allows us to achieve impressive results in terms of colorizing various types of images.Utilizing Generative Adversarial Networks (GANs) and CNN to achieve realistic and accurate colorization of grayscale images.

Furthermore, we delve into the integration of OpenCV, a powerful computer vision library, with the deep learning model to enhance the colorization process. OpenCV is used for image preprocessing, post-processing, and visualization, providing a user-friendly and efficient pipeline for colorization tasks. We discuss practical considerations such as dataset selection, model training, and hyperparameter tuning, providing insights into the best practices for achieving high-quality colorization results. Additionally, we present a performance evaluation of our approach, comparing it to existing methods and highlighting its advantages.

Overall, this paper presents a valuable resource for researchers and practitioners interested in image colorization, offering a detailed explanation of the techniques and tools required to achieve accurate and visually appealing colorization results. Our approach not only provides state-of-the-art performance but also a user-friendly framework for real-world colorization applications.

1.8 GOALS AND OBJECTIVES

- Develop an Efficient Colorization System: Create a system that automatically adds realistic colors to grayscale images using advanced deep learning technologies.
- Train a Deep Learning Model: Select and train an effective deep learning model for colorization, integrating it with OpenCV for easy application.
- Create a User-Friendly Interface: Develop an intuitive interface for users to upload and colorize images, ensuring accessibility and practicality.
- Enable Real-Time Adjustments: Integrate real-time image parameter adjustments to enhance usability and flexibility in various workflows.

1.9 RELEVANT MATHEMATICS ASSOCIATED WITH THE PROJECT

Convolution: Convolutional Neural Networks (CNNs) employ convolution operations to extract features from images. Understanding the convolution operation is crucial for designing CNN architectures.

Activation Functions Sigmoid, ReLU (Rectified Linear Unit), and other activation functions are used in neural networks to introduce non-linearity.

Loss Functions: Mathematical formulations of loss functions, such as Mean Squared Error (MSE) or perceptual loss, are used to measure the difference between the predicted and actual color images.

Gradient Descent Optimization: Understanding how gradient descent algorithms update model parameters to minimize the loss function is critical for training deep learning models.

Backpropagation: The backpropagation algorithm involves computing gradients of the loss function with respect to the network's weights, which is essential for updating the model during training.

Eigenvalues and Eigenvectors Principal Component Analysis (PCA), which relies on eigenvalues and eigenvectors, can be used for dimensionality reduction and feature extraction in image data.

Fourier Transform: Fourier transforms can be applied for image preprocessing and feature extraction, allowing the identification of image patterns in the frequency domain.

ϕ = Failures and Success conditions.

- Success Conditions:
    1. Accurate colorization results.
    2. Seamless OpenCV integration.
    3. Efficient processing
- Failure Conditions:
    1. Inaccurate colorization.
    2. Complex user interfaces
    3. Hardware failure.
    4. Software failure.

1.10 NAMES OF CONFERENCES / JOURNALS WHERE PAPERS CAN BE PUBLISHED

- IJARCSMS(International Journal of Advance Research in Computer Science And Management Studies)
- IJTRA(International Journal of Technical Research and Applications)

- IEEE(Institute of Electrical and Electronics Engineers)
- PGCON(Post Graduate Conference of Computer Engineering)

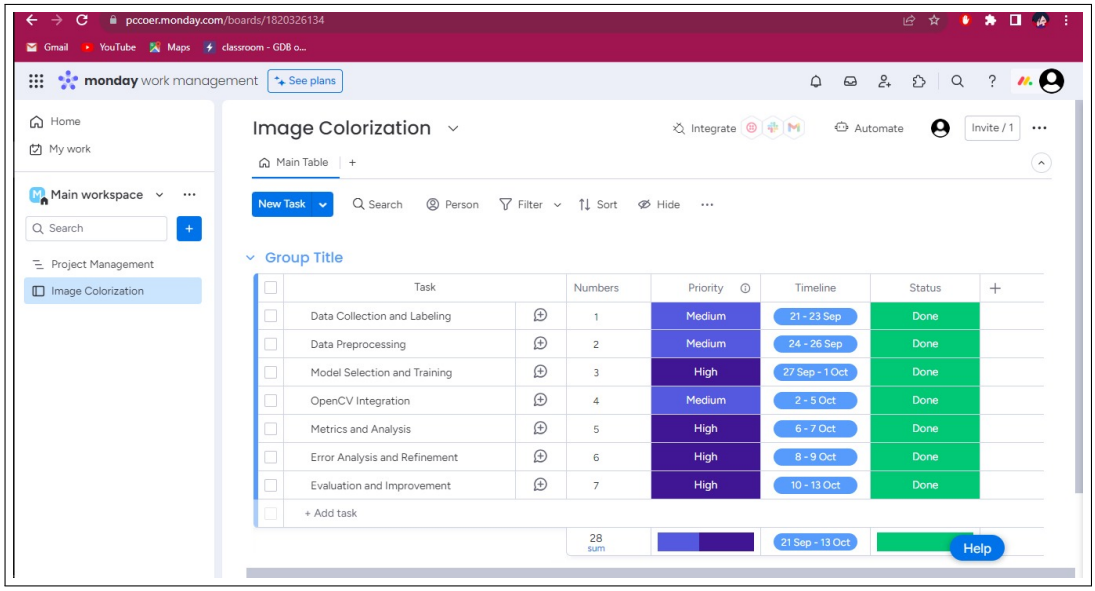
1.11 REVIEW OF CONFERENCE/JOURNAL PAPERS SUPPORTING PROJECT IDEA

- Levin, Anat Lischinski, Dani Weiss, Yair: This paper presents a system that allows users to colorize grayscale images by providing colorful scribbles on the target image. The color information from these scribbles is also propagated to the rest of the target image using least-square optimization. While this approach shows pledge, it has the limitation of being time-consuming and may lead to color bleeding vestiges.
- Huang, Yi-Chin Tung, Yi-Shin Chen, Wu, and Ja-Ling: The authors of this paper introduce a system that employs an adaptive edge discovery algorithm to reduce color bleeding artifacts around the area perimeter in image colorization. They also integrate this technique with colorization based on scribbles. However, this method may struggle to detect subtle intensity changes and works best with high-resolution images.
- Hertzmann, Aaron Jacobs, Charles Oliver, Brian Salesin: This paper introduces a technique involving a source image (A), a filtered source (A'), and an original target image (B) to generate a filtered target (B'). It assumes alignment between A and A ' and relies on pixel correspondence to change B to B '. While this system is effective for numerous scripts, it can be grueling to apply automatically to complex and larger-scale cultural styles, taking customized approaches.
- Yatziv, Liron Sapiro, Guillermo: The authors present a method for rapid colorization of pixels, using blending textures created based on the distances between user scribbles and neighboring pixels. These user scribbles represent combinations of shaded areas and their corresponding colors. However, a limitation of this method is that it may become relatively slow and memory

intensive when dealing with a large number of scribbles with different chrominances, such as in video frames.

- Welsh, Tomihisa Ashikhmin, Michael Mueller, Klaus: This paper describes a method that uses pixel intensity and neighborhood statistics to find similar pixels in an image and transfer their colors to target pixels. While this approach is effective in many cases, it struggles with tasks like classifying faces, distinguishing between skin and lips, and may encounter challenges when dealing with items like clothes and hair.

1.12 PLAN OF PROJECT EXECUTION



CHAPTER 2

TECHNICAL KEYWORDS

2.1 AREA OF PROJECT

The project, " Image Colorization with OpenCV and Deep Learning " is at the nexus of computer vision and artificial intelligence, focusing on the detailed art of infusing color into black and white or grayscale images. This undertaking holds substantial significance in colorful disciplines. First, it enables the enhancement and refinement of medical imaging, providing clearer insights into anatomical structures and pathological conditions, ultimately improving diagnostic accuracy and treatment efficacy in healthcare settings. Additionally, the project has broad applications in the realm of historical photograph art, enabling artists and designers to experiment with color palettes and reimagine classic works of art in a contemporary context. The project also contributes to the field of education by making historical materials more engaging and relatable through colorization in educational resources and documentaries. This interdisciplinary project combines the power of deep learning, a leading-edge technology, and OpenCV, a versatile computer vision library, to create an efficient and accessible framework for image colorization. It addresses critical aspects like dataset preparation, model training, and fine-tuning, making it a valuable resource for researchers and practitioners alike. In essence, this project transforms grayscale images, making them vibrant and relevant across various domains, with the potential to revolutionize how we perceive and interact with visual content.

2.2 TECHNICAL KEYWORDS

Technical Key Words:

- Mathematics of Computing:
  The mathematics of computing encompasses mathematical concepts and techniques applied in computer science.

- Colorization:
  Colorization is the process of adding color to a black-and-white, grayscale, or monochrome image or video, transforming it into a version with a realistic or artistically chosen color scheme. This technique is typically applied to historical photographs, film footage, or other visual content to make them more vivid

and lifelike. Colorization can be done manually by artists or automatically using computer algorithms, including deep learning-based methods, which learn to predict and apply relevant colors to different elements within the image.

- Vision For Graphics:

  It refers to the use of computer vision technologies in the field of computer graphics. It involves the development and integration of visual perception systems, often based on AI and machine learning, to enhance and improve various aspects of computer-generated graphics and visual content.

- CNNs:

  Convolutional Neural Networks are deep learning models specially designed for processing visual data. They excel at tasks such as image recognition, object detection, and image segmentation by automatically learning hierarchical patterns and features from the input images through convolutional, pooling, and fully connected layers. They have revolutionized computer vision and have applications in various other fields, including natural language processing and speech recognition.

- Self-supervised learning:

  Self-supervised learning trains models on unlabeled data by predicting parts of the input from other parts. This approach leverages the data's inherent structure, aiding tasks like representation and transfer learning without needing labeled datasets. This method is valuable for tasks such as representation learning and transfer learning, enabling models to extract meaningful features and improve performance across various domains.

- GANs (Generative Adversarial Networks):

  GANs are machine learning frameworks that generate new data samples resembling a given dataset. They consist of a generator, which creates fake data, and a discriminator, which evaluates it against real data. This adversarial process improves the generator's ability to produce realistic data, useful in image synthesis and data augmentation.

CHAPTER 3

INTRODUCTION

3.1 PROJECT IDEA

The project “Automatic Image Colorization for Medical Imaging Enhancement” is a significant endeavor aimed at improving the clarity and interpretability of medical images by automating the colorization process. Many medical scans, such as X-rays and MRIs, exist solely in grayscale, which can hinder detailed analysis and diagnosis. Manual colorization is time-consuming and subjective, requiring specialized expertise. This project leverages the power of OpenCV for image preprocessing and employs deep learning, specifically Convolutional Neural Networks (CNNs), to predict color information from grayscale input. The user-friendly interface allows healthcare professionals to easily upload medical images and apply automated colorization, making the enhancement process accessible and efficient. Moreover, quality control mechanisms ensure that the colorization meets accuracy standards, aiding in precise diagnosis and treatment planning. The potential applications are vast, including improved diagnostic accuracy, better educational tools for medical students, and enhanced patient communication. However, it comes with its set of challenges, such as the need for a comprehensive dataset for training, addressing ethical concerns regarding color choices, and balancing automation with user intervention. Ultimately, this project offers a promising solution to enhance medical imaging, contributing to better healthcare outcomes and educational experiences.

3.2 MOTIVATION OF THE PROJECT

The motivation behind embarking on this research journey is rooted in recognizing the immense transformative power of image colorization within the expansive domain of computer vision and its profound implications across various fields. Grayscale images, although inherently possessing substantial informational content, frequently fall short in conveying the richness, vibrancy, and nuanced detail that colorization can afford. This inherent limitation impedes their broader application and hinders the depth of interpretability they could otherwise offer.

Propelled by a desire to unlock the latent potential of grayscale imagery, this project endeavors to leverage the unparalleled capabilities of Generative Adversar-

ial Networks (GANs) and cutting-edge deep learning techniques. By amalgamating these advanced methodologies, the project aspires to pioneer a groundbreaking approach aimed at imbuing monochromatic images with a newfound vitality, thereby transcending the boundaries of traditional image processing paradigms. This innovative effort not only promises to enhance the aesthetic and practical value of historical photographs and film footage but also holds the potential to revolutionize various applications across fields such as medical imaging, remote sensing, and digital art restoration.

3.3 LITERATURE SURVEY

No.	AUTHOR	TITLE	METHODOLOGY	LIMITATIONS
1	Levin, Anat & Lischinski,& Dani & Weiss, Yair	Colorization using optimization	The user will provide colorful scribbles on the grayscale target image	requires substantial efforts from user to provide scribbles
2	Huang, Yi-Chin & Tung, Yi-Shin & Chen, & Wu, Ja-Ling	An adaptive edge detection based colorization algorithm and its applications	used adaptive edge detection algorithm to reduce the color bleeding artifact around the region boundaries	requires high res images to accurately detect images
3	Hertzmann, Aaron & Jacobs, Charles & Oliver, Brian & Salesin, David	Image Analogies	(A), a filtered source (A'), and an original target (B). Its objective is to generate a filtered target (B')	It struggles with complex, larger-scale artistic styles and requires tailored approaches
4	Yatziv, Liron & Sapiro, Guillermo	Fast Image and Video Colorization Using Chrominance Blending	It achieves fast colorization of pixels by creating blending texture	If there are a large number scribbles of different chrominances. the algorithm could be relatively slow and pricey in memory for video frames as images are stored in memory
5	Welsh, Tomihisa & Ashikhmin, Michael & Mueller, Klaus	Transferring Color to Greyscale Images	uses pixel intensity and neighborhood statistics to find a similar pixel in image & then transfer the color of matched pixel to the target pixel	This technique does not work very well with faces & struggles in classifying the between skin & lips. Sometimes clothes & hair

No.	AUTHOR	TITLE	METHODOLOGY	LIMITATIONS
6	Irony, Revital and Cohen-Or, Daniel and Lischinski, Dani	Colorization by Example	combines user-guided colorization by matching neighbor pixels while avoiding manual effort. It uses a texture-based classifier derived from a feature-based analysis of image	Example based methods share the limitation that the colorization quality relies heavily on example images provided by the user and finding a suitable reference image is a difficult task
7	Chia, Alex & Zhuo, Shaojie & Gupta, Raj & Tai, Yu-Wing & Cho, David & Tan, Ping & Lin, Stephen	Semantic Colorization with Internet Images	The system identifies candidate reference regions from the internet based on image features and user input, then transfers colors to the target image using a graph-based optimization	Scenes with many foreground objects may be time-consuming for users to label and segment
8	Liu, Xiaopei & Wan, Liang & Qu, Yingge & Wong, Tien-Tsin & Lin, Stephen & Leung, Chi-Sing & Heng, Pheng-Ann	Intrinsic Colorization	Generate color scribbles from the reference reflectance image and target illumination image. Transfer color to the target image using the color scribbles	Intrinsic image colorization assumes that the illumination in all images is white light. This is not always the case, and can lead to inaccurate results
9	Yan, Zhicheng, Hao Zhang, Baoyuan Wang, Sylvain Paris, and Yizhou	Automatic Photo Adjustment Using Deep Neural Networks	It uses deep neural networks to transform photos	The DNN architecture is a time-consuming trial-and-error process to optimize
10	Zheng, Songfeng & Yuille, Alan & Tu, Z.	Detecting Object Boundaries Using Low-, Mid-, and High-level Information	The approach uses probabilistic boosting trees (PBTs) to learn and combine low-, mid-level cues, and then uses a shape matching algorithm to engage high-level shape	It only works for single objects, the high-level model is not adequate to capture bigger variations, and the link between low-, mid-, and high-level information is not clear

CHAPTER 4

PROBLEM DEFINITION AND SCOPE

4.1 PROBLEM STATEMENT

Developing a robust colorization system using advanced deep learning techniques like Generative Adversarial Networks (GANs) to accurately add realistic colors to grayscale images and providing a user-friendly interface for practical utilization..

4.1.1 Goals and objectives

Goal and Objectives:

- Develop a Robust and Efficient Black and White Image Colorization System: Create a system that automatically adds realistic and vibrant colors to grayscale images using state-of-the-art deep learning technologies.
- Select and Train an Appropriate Deep Learning Model: Choose an effective deep learning architecture for colorization, train the model on a prepared dataset, and integrate the trained model with OpenCV for easy application.
- Create a User-Friendly Interface: Develop an intuitive interface for users to upload and colorize images, ensuring accessibility and practicality.
- Enable Real-Time Image Parameter Adjustment: Integrate functionality for real-time adjustment of image parameters to enhance the usability and flexibility of the colorization system in various workflows.

4.1.2 Statement of scope

The project is designed to develop a robust and user-friendly image colorization system. By integrating OpenCV and deep learning, it offers an extensive solution for transforming grayscale or black-and-white images into vibrant, colored versions. The primary objectives encompass image preprocessing to enhance input quality, training a Convolutional Neural Network (CNN) for accurate colorization, and seamless integration with the OpenCV library for streamlined image handling. A user-friendly interface will empower users to easily upload images and apply automated colorization, adjust image parameters in real time, and enabling creative expression. Quality control mechanisms will ensure that the results meet accuracy standards.

While it doesn't include data acquisition, ethical color choices, or complex interpretations, the project addresses a critical need for contributing to improved healthcare outcomes and educational experiences by enhancing of medical imaging.

4.2 SOFTWARE CONTEXT

In the context of software development, Image colorization using OpenCV typically involves utilizing libraries and techniques to add color information to grayscale images. OpenCV is the primary software library for image processing and computer vision tasks, including image colorization. It provides a wide range of functions and algorithms for image manipulation. You need grayscale images as input for colorization. These images can be loaded using OpenCV functions such as cv2.imread(). This application leverages modern web technologies and frameworks to provide features such as real-time alert systems, incident reporting, live location sharing, educational content, and community engagement.Image colorization often leverages deep learning models, such as Convolutional Neural Networks (CNNs), to predict color information.

4.3 MAJOR CONSTRAINTS

Constraints can be defined as limiting factor states of restriction or lack of spontaneity of software. Constraints are the limitations, and hurdles that stop the software team from fulfilling their responsibility. Constraints are anything that restricts or dictates the actions of the project team. Any constraints that will impact the manner in which the software is to be specified, designed, implemented, or tested are given here.

3 core major constraints:

- Time - This refers to the time required to produce a deliverable which in this case would be the end result of the project. Naturally, the amount of time required to produce the deliverable will be related to the amount of requirements that are part of the end result along with the amount of resources allocated to the project.

- Cost - This is the estimation of the amount of money that will be needed to complete the project. Cost itself includes various things, such as resources, labor rates for contractors, risk estimates, bills of materials, etc. All aspects of the project that have a financial component are made part of the overall cost structure.

- Scope - These are the functioning elements that when completed, make up the end deliverable for the project. The scope itself is generally identified upfront so as to give the project the best chance of success. A common success measure for the scope of a project is its innate quality upon delivery.

- Computational Resources -Deep learning for image colorization can be computationally intensive. Limited access to high-performance GPUs may result in longer processing times and could restrict the project's scalability.

- Colorization Accuracy - Achieving high colorization accuracy, especially for complex or ambiguous images is a significant constraint based on the current state of deep learning models.

4.4 METHODOLOGIES OF PROBLEM SOLVING AND EFFICIENCY ISSUES

Efficiency in the project is achieved through the utilization of deep learning methodologies, such as CNNs and transfer learning. Preprocessing improves image quality, while data augmentation enhances dataset diversity. Efficiency is addressed by leveraging GPUs, optimizing batch processing, applying model optimizations, parallel processing, caching mechanisms, and asynchronous handling in real-time applications. It also uses Generative Adversarial Networks to improve the accuracy of the model. Balancing accuracy with speed remains a key challenge in the project's development.

Algorithm and Technique:

- Loss Functions

- Data Augmentation
- Convolutional Neural Networks
- Generative Adversarial Networks
- OTP Generation Algorithm
- AES Decryption Algorithm

4.5 SCENARIO IN WHICH MULTI-CORE, EMBEDDED AND DISTRIBUTED COMPUTING USED

Multi-core computing is deployed for parallel processing in applications like gaming and data centers. Embedded computing is used in IoT devices, vehicles, and medical equipment. Distributed computing is essential for big data processing, cloud services, and blockchain technology, enabling scalability and real time data handling. These computing approaches cater to diverse needs across various domains, from performance and efficiency to real-time control and massive data analysis.

- Software as a service: It is a software delivery model in which applications are hosted by a service provider and made available to customers over a network.
- Platform as a service: It is a development platform provided as a service that supports the full software life cycle. It allows users to develop cloud-based services and applications.
- Infrastructure as a service: It is a cloud service model that provides basic data storage and computing capabilities as standardized services on the network.

4.6 OUTCOME

The project aims to deliver an automated image colorization tool with a user-friendly interface, offering high-quality results for medical imaging enhancement and also allow to adjust image parameters in real time. Seamless integration with OpenCV and quality control mechanisms will ensure efficient and effective colorization of grayscale and black and white images.

4.7 APPLICATIONS

The project's applications span a diverse range, leveraging technological advancements to inject vibrancy into grayscale and historical images. One notable application lies in the domain of medical imaging enhancement, where the project holds the potential to revolutionize diagnosis and treatment planning. By enhancing medical images with realistic and detailed colorization, healthcare professionals can gain clearer insights into anatomical structures and pathological conditions, leading to more accurate diagnoses and informed treatment decisions. This advancement not only improves patient care but also streamlines medical workflows, ultimately enhancing the efficiency and efficacy of healthcare systems. In cultural heritage preservation, the project restores old photographs, making history more relatable to contemporary audiences. For artists and designers, it enables creative expression by infusing life into grayscale artworks. In education, colorized historical visuals engage students, while in media and entertainment, the technology enhances visual storytelling. Additionally, it revitalizes personal projects, breathing new life into family photos and collections. The project contributes to cultural heritage preservation and the creation of visually captivating content across various fields.

4.8 HARDWARE RESOURCES REQUIRED

- Processor - i5-4590
- Graphic card- GTX1080
- RAM - 8Gb (min)
- Keyboard - Standard Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - LED Monitor

4.9 SOFTWARE RESOURCES REQUIRED

- Operating System - Windows 10

- Programming Language - Python
- Library and Frameworks - NumPy, Matplotlib, PIL, OpenCV, TF or Pytorch for training GANs
- Deep Learning Framework- CNN
- Tools - Google Colab for interactive development and testing.
- Database - ImageNet

CHAPTER 5

PROJECT PLAN

5.1 PROJECT ESTIMATES

Use the Waterfall model and streams derived from assignments 1,2, 3, 4 and 5(Annex A and B) for estimation.

1. Requirement Gathering: Requirement gathering and plan for the initial part of the project was as follows: Understanding the problem definition. Understanding the reliable factors for data sharing systems. Gathering information about required software. Gathering information about required Hardware Resources Preparing preliminary design of overall work flows of project deciding the modules required for overall execution.

2. Analysis : Analysis of system collects systems requirement. Detailed plan of project and estimation of budget is studied in this stage. It is useful for the future development of project. Document containing system requirements is end product of this phase. Management and user requirements, alternative plans are further described by this document.

3. Design: Design focuses on high level design like, software i.e. Net beans for J2ME software are needed and their interaction with each other, then interface design to visualize how the project going to look like.

4. Implementation: In this phase code is generated from the designs.

5. Testing: System is tested in this phase. Individual modules are tested then whole system is tested. The system is tested to check inter related working of modules i.e. integration testing and that the system does what the user requires.

6. Maintenance: Maintenance is required for system. Because of some unexpected input values into the system the, changes in the system occurred.

5.1.1 Reconciled Estimates

5.1.1.1 Cost Estimate

5.1.1.2 Time Estimates

5.1.2 Project Resources

Project resources [People, Hardware, Software, Tools, and other resources] based on Memory Sharing and Concurrency derived using appendices to be referred.

5.2 RISK MANAGEMENT W.R.T. NP HARD ANALYSIS

This section discusses risks and the approach to managing them.

5.2.1 Risk Identification

For risk identification, a review of the scope document, requirements specifications, and schedule is done. Answers to the questionnaire revealed some risks. Each risk is categorized as per the categories mentioned in [1]. Please refer to table 5.1 for all the risks. You can refer to the following risk identification questionnaire.

1. Have top software and customer managers formally committed to supporting the project?
2. Are end-users enthusiastically committed to the project and the product to be built?
3. Are requirements understood by the software engineering team and its customers?
4. Have customers been involved fully in the definition of requirements?
5. Do end-users have realistic expectations?
6. Does the software engineering team have the right mix of skills?
7. Are project requirements stable?
8. Is the number of people on the project team sufficient to do the job?

9. Do all customer constituencies agree on the importance of the project and on the requirements for the system/product to be built?

5.2.2 Risk Analysis

The risks for the Project can be analyzed within the constraints of time and quality

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	System Failure	Low	Low	High	High
2	Connection Failure	Low	Low	High	High

Table 5.1: Risk Table

Probability	Value	Description
High	Probability of occurrence is	$> 75\%$
Medium	Probability of occurrence is	$26 - 75\%$
Low	Probability of occurrence is	$< 25\%$

Table 5.2: Risk Probability definitions [1]

Impact	Value	Description
Very high	$> 10\%$	Schedule impact or Unacceptable quality
High	$5 - 10\%$	Schedule impact or Some parts of the project have low quality
Medium	$< 5\%$	Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

Table 5.3: Risk Impact definitions [1]

5.2.3 Overview of Risk Mitigation, Monitoring, Management

Following are the details for each risk.

Risk ID	1
Risk Description	Description 1
Category	Development Environment.
Source	Software requirement Specification document.
Probability	Low
Impact	High
Response	Mitigate
Strategy	Strategy
Risk Status	Occurred

Risk ID	2
Risk Description	Description 2
Category	Requirements
Source	Software Design Specification documentation review.
Probability	Low
Impact	High
Response	Mitigate
Strategy	Better testing will resolve this issue.
Risk Status	Identified

Risk ID	3
Risk Description	Description 3
Category	Technology
Source	This was identified during early development and testing.
Probability	Low
Impact	Very High
Response	Accept
Strategy	Example Running Service Registry behind proxy balancer
Risk Status	Identified

5.3 PROJECT SCHEDULE

5.3.1 Project task set

Major Tasks in the Project stages are:

- Task 1:Requirement Analysis (Base Paper Explanation).
- Task 2:Project Specification (Paper Work).
- Task 3:Technology Study and Design.
- Task 4:Coding and Implementation (Module Development).

5.3.2 Task network

Individual tasks and subtasks have interdependencies based on their sequence. A task network is a graphic representation of the flow of a project. Project tasks and their dependencies are noted.

5.4 TEAM ORGANIZATION

The manner in which staff is organized and the procedure for reporting are noted.

5.4.1 Team structure

The team structure for the project is identified. Roles are defined.

5.4.2 Management reporting and communication

Procedure for progress reporting and inter/intra team communication are identified as per the assessment sheet and lab timetable.

5.4.3 Timeline Chart

A project timeline chart is presented. This may include a timeline for the entire project. The above points should be covered in Project Planner as Annex C and you can mention them here and Please refer to Annex C for the planner.

Sr No.	Month	Description
1	June	Discussion with guide regarding domain.
		Searching for IEEE paper for domain.
2	July	Short listing of IEEE papers within domain.
		Selection of IEEE paper.
3	August	Deciding Project name.
		Submission of Synopsis.
4	September	Requirement analysis.
		Designing of models.
5	October	Report preparation.
		Stage-I report submission.

Table 5.4: Management plan [1]

Task Name	Start date	End date	Duration
Initiate the project	04/8/2023	24/8/2023	21
Communication	04/8/2023	10/8/2023	7
Literature survey	11/8/2023	17/8/2023	7
Define scope	18/8/2023	19/8/2023	2
Develop SRS	20/8/2023	24/8/2023	5
Plan the project	25/8/2023	5/10/2023	42
Design mathematical model	25/8/2023	31/8/2023	7
Feasibility Analysis	01/9/2023	07/9/2023	7
Develop work breakdown structure	08/9/2023	09/9/2023	2
Planning project schedule	10/9/2023	14/9/2023	5
Design UML and other diagrams	15/9/2023	21/9/2023	7
Design test plan	22/9/2023	28/9/2023	7
Design risk management plan	29/9/2023	5/10/2023	7
Execute the project	05/01/2024	29/03/2024	84
Build and test basic functional unit	05/01/2024	25/01/2024	21
Build and test database with login	26/01/2024	15/02/2024	21

Table 5.5: Time-line Chart [1]

CHAPTER 6

SOFTWARE REQUIREMENT SPECIFICATION

6.1 INTRODUCTION

6.1.1 Purpose and Scope of Document

1. Facilitating other Documentation: The SRS forms the basis for a load of other important documents such as the Software Design Specification.

2. Product Validation: It basically helps in validating with the client that the product that is delivered, meets the requirements.

6.1.2 Characteristics of a Software Requirement Specification

1. Accuracy: This is the first and the foremost requirement. The development team will get nowhere if the SRS which will be the basis of the process of software development, is not accurate.

2. Completeness : The software requirement specification should not be missing any of the requirements stated in the business requirements documentation that the user specified.

3. Prioritization of Requirements : Software Requirement Specification should not simply be a wish list. The requirements should follow the order of priority and preference.

6.1.3 Overview of responsibilities of Developer

The responsibilities from developer perspective are to develop the application as per the requirement with user friendly user interface. For this the developer is required to go through the concepts required to be implemented. The developer follows the software design life cycle, starting from requirement gathering, analysis, design, coding, and then implementation of the system. After the implementation is completed, the graphical data representation and memory utilization is tested manually for random files or data.

6.2 USAGE SCENARIO

To provide the analyzing data from different perspectives and summarizing it into useful information - information that can increase revenue, cut costs, or both. It can be useful where a huge amount of data is to be stored. Ex. Google, Social Networking Application Etc.

Sr No.	Actor	Description
1	User	The User has to firstly register itself by filling the registration form and be the active member of the system. If a User is already the member of the system then he or she can perform login process. Then Data Owner can be performing operations like file uploading, file downloading, sending request for encryption key, sending request for decryption key, sending request for OTP, etc.
2	Authority	The Authority has to firstly login himself. Then Authority can be performing operations like view encryption key request, view decryption key request, change password, etc.
3	Cloud Server	The Cloud Server has to firstly login himself. Then Cloud Server can be performing operations like view all users, view all files, view result, etc.

Figure 6.1: diagram

6.2.1 Use-cases

A use case for a project outlines how a system or application will be used by its users.

One compelling use case for image colorization with OpenCV and Deep Learning is the enhancement of medical imaging. Many medical scans, such as X-rays and MRIs, are captured in grayscale, which can sometimes limit the clarity and interpretation of crucial details. By leveraging OpenCV and deep learning models,

these grayscale medical images can be automatically colorized, enhancing their visual clarity and aiding healthcare professionals in diagnosis and treatment planning. This technology not only improves the accuracy of medical interpretations but also streamlines workflows in healthcare settings. Additionally, colorized medical images can enhance educational materials, providing students with more engaging and informative resources for learning about anatomy and pathology. Overall, image colorization in medical imaging enhancement bridges the gap between grayscale images and vivid visual representations, contributing to improved healthcare outcomes and educational experiences.

6.2.2 Use Case View

A use case diagram is a graphical representation of a user input with the system and presents the specifications of a use case. A use case diagram can show the different types of clients of a system and the various ways in which they interact with the system. Use case diagrams are used to find the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to get its functionality use cases are prepared and actors are identified. The purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interaction among the actors.

6.3 DATA MODEL AND DESCRIPTION

6.3.1 Data Description

The controller is responsible for responding to user input and does the interactions on the data model objects. The controller receives the input; it validates the input and then performs the business operation which modifies the state of the data model.

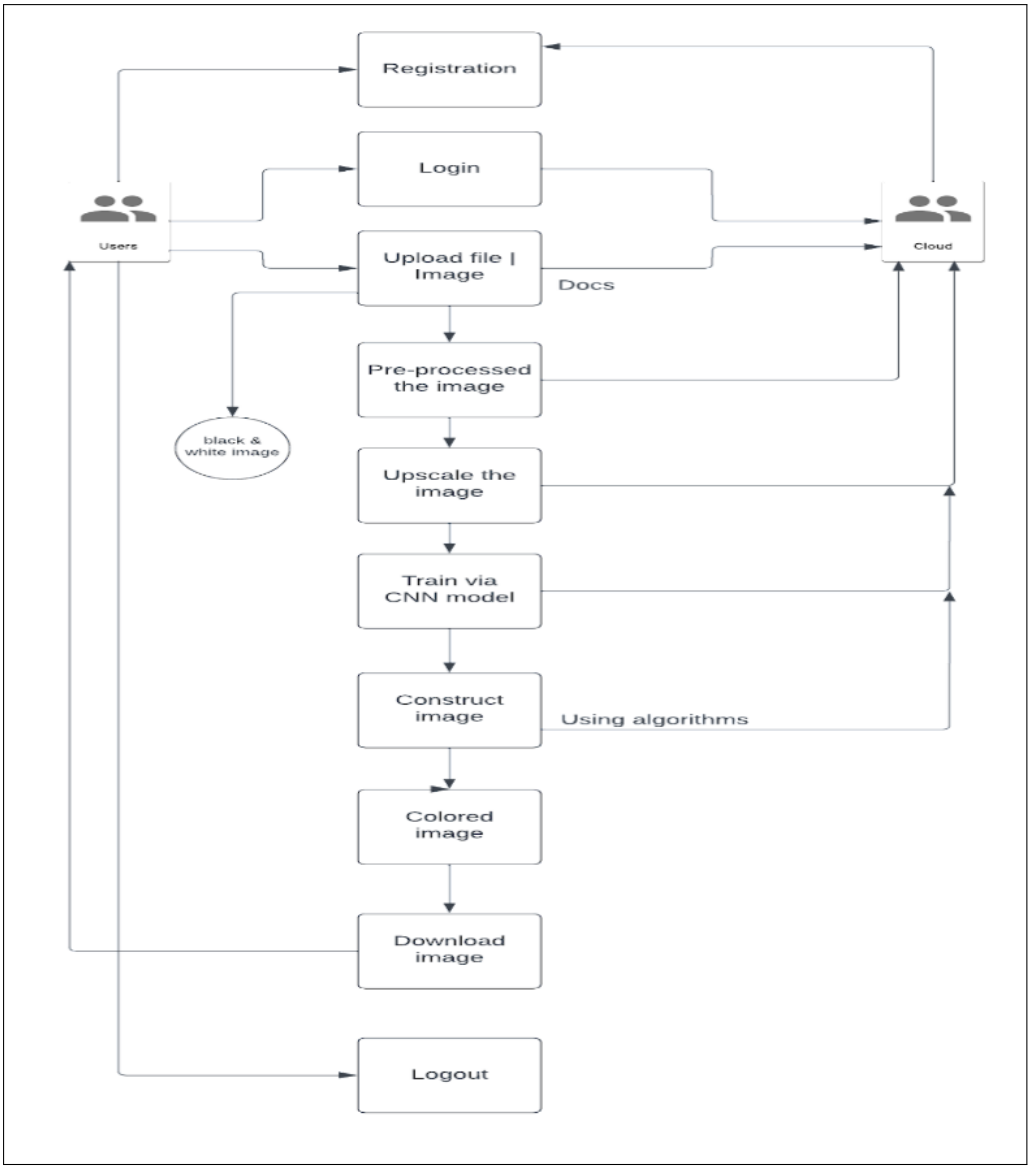


Figure 6.2: Use - Case diagram

6.3.2 Data objects and Relationships

The model is responsible for controlling the data of the application. It responds to the request from the view and it also responds to instructions given by the controller to update itself.

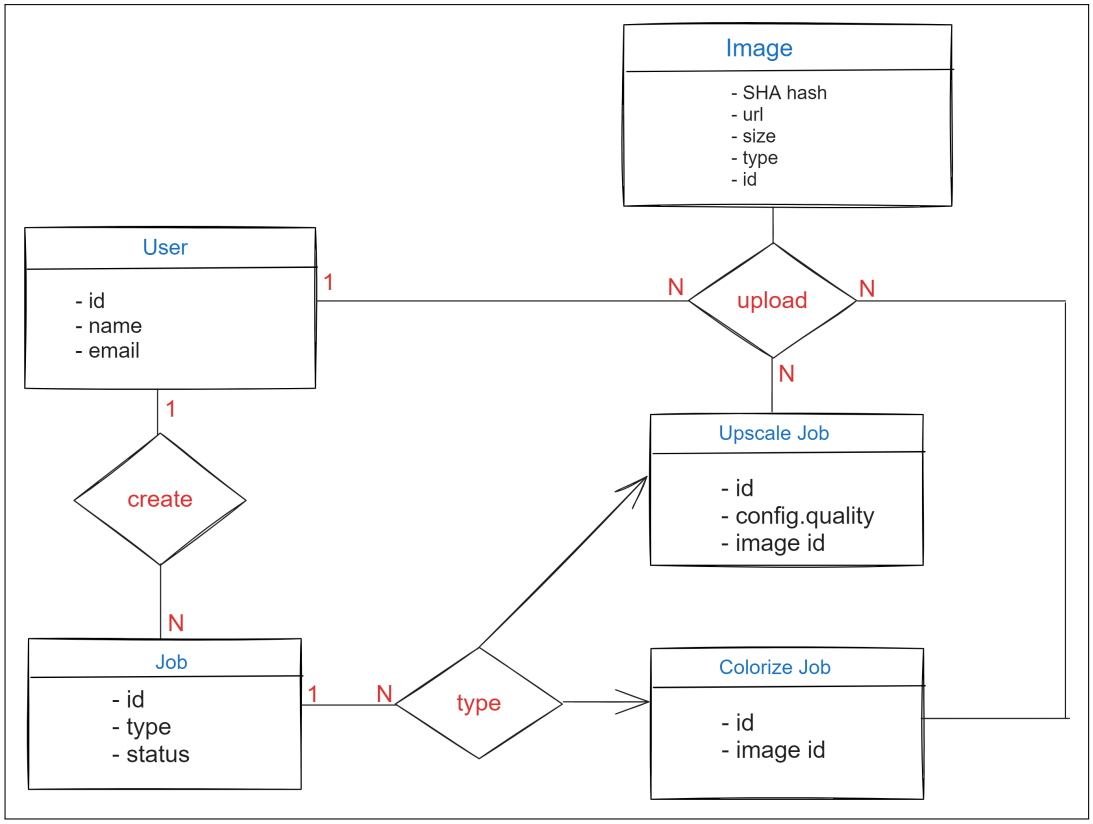


Figure 6.3: ER diagram

6.4 FUNCTIONAL MODEL AND DESCRIPTION

6.4.1 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can be elaborated. DFDs is also be used for the visualization of data processing (structured design).A DFD shows what kinds of information will be the input and output from the system, where the data will come and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will also operate in sequence or in parallel (which is shown on a flowchart).

6.4.1.1 Level 0 Data Flow Diagram

The DFD Level 0 identifies external entities and processes of the system. Level 0 explains the architecture that would be used for developing a software product. A context diagram is a toplevel (also known as ”Level 0”) data flow diagram. It

only contains a process node (”Process 0”) that generalizes the function of the entire system in relationship to external entities.

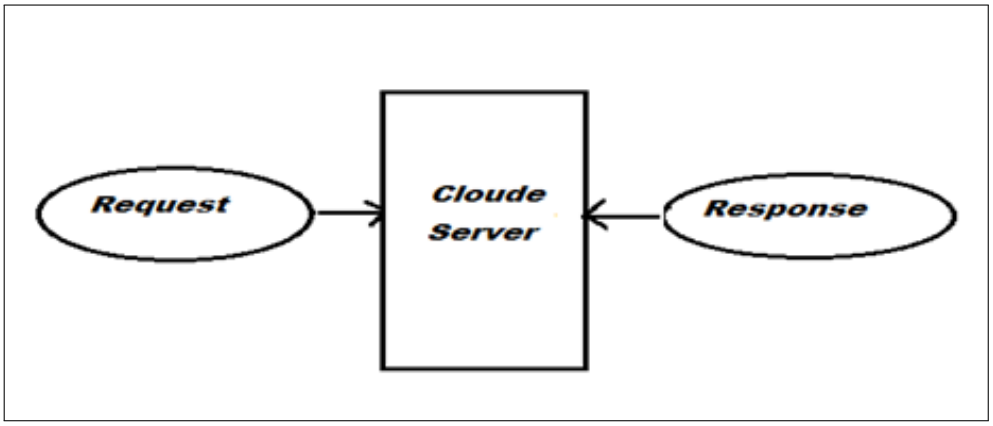


Figure 6.4: DFD0 diagram

6.4.1.2 Level 1 Data Flow Diagram

This DFD Level 1 shows the main processes in the work and the entities involved in it. We begin withdrawing a context diagram, a simple representation of the whole system. To elaborate further, we drill down to a level 1 diagram with further information about the major functions of the system. Level 1 is an extension of the level 0 diagram.

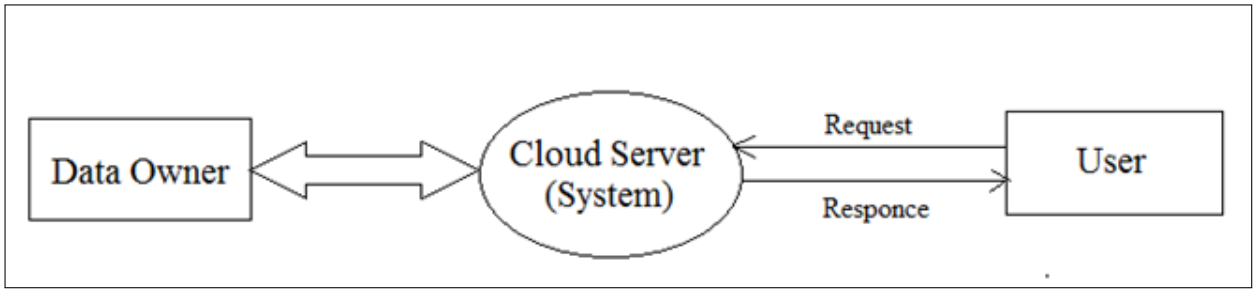


Figure 6.5: DFD1 diagram

6.4.2 Description of functions

A description of each software function is presented. A processing narrative for function n is presented.(Steps)/ Activity Diagrams. For Example Refer 6.7

6.4.3 Activity Diagram:

Activity diagrams are graphical representations of workflows activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are made to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control. Activity diagrams are made from a limited number of shapes, connected with arrows. The most important shape types:

- Rounded rectangles represent actions;
- Diamonds represent decisions;
- Bars represent the start (split) or end (join) of simultaneous activities;
- A black circle represents the start (initial state) of the flow;
- A black circle represents the end (final state).

Arrows run from the start towards the end and indicate the order in which activities happen. Hence they can be regarded as a form of flowchart. Typical flowchart techniques lack constructs for ex-pressing concurrency. However, the join and split symbols in activity diagrams only resolve simple cases; the meaning of the model is not clear when they are randomly combined with decisions or loops.

- Activity Diagram for User

6.4.4 Non Functional Requirements:

- Interface Requirements
- Performance Requirements
- Software quality attributes such as availability [related to Reliability], modifiability [includes portability, reusability, scalability] , performance, security, testability and usability[includes self adaptability and user adaptability]

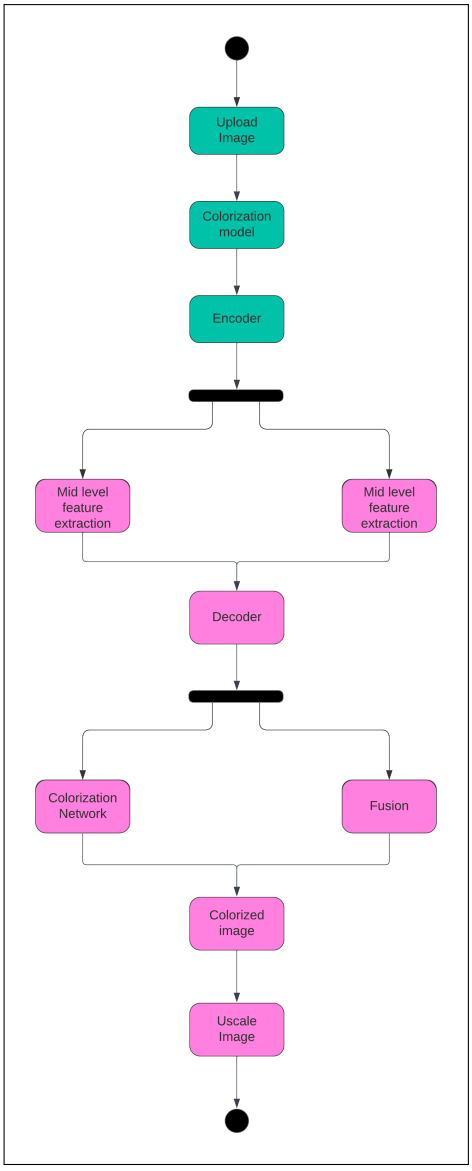


Figure 6.6: Activity diagram

6.4.5 State Diagram:

State Transition Diagram

Fig.6.8 example shows the state transition diagram of Cloud SDK. The states are represented in ovals and the state of the system gets changed when certain events occur. The transitions from one state to another are indicated by arrows. The Figure shows important states and events that occur while creating a new project.

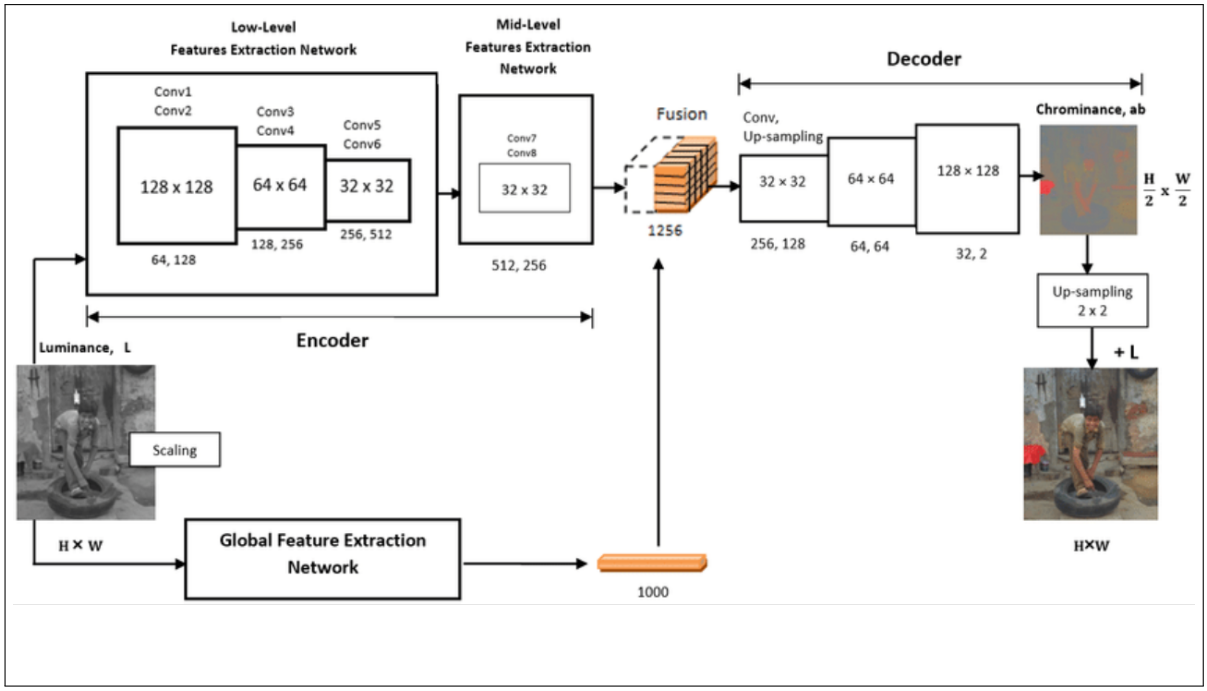


Figure 6.7: State Diagram

6.4.6 Design Constraints

Design constraints include hardware limitations, the availability and quality of training data, ethical considerations for historical content, accommodating users of varying technical proficiency, optimizing resource efficiency, model training time, real-time processing capabilities, and the need for a stable internet connection. Achieving high colorization accuracy, especially in complex images, is also a challenge. Data privacy and security concerns must be addressed in handling user-uploaded images. These constraints inform the project's development, ensuring practicality and ethical use of the technology.

6.4.7 Software Interface Description

The interface for the "Image Colorization with OpenCV and Deep Learning" project offers a user-friendly experience. Users can effortlessly upload grayscale images, initiate colorization, and preview the results. It may include options for quality control, manual adjustments, and educational content, ensuring ease of use and enhancing the overall colorization process.

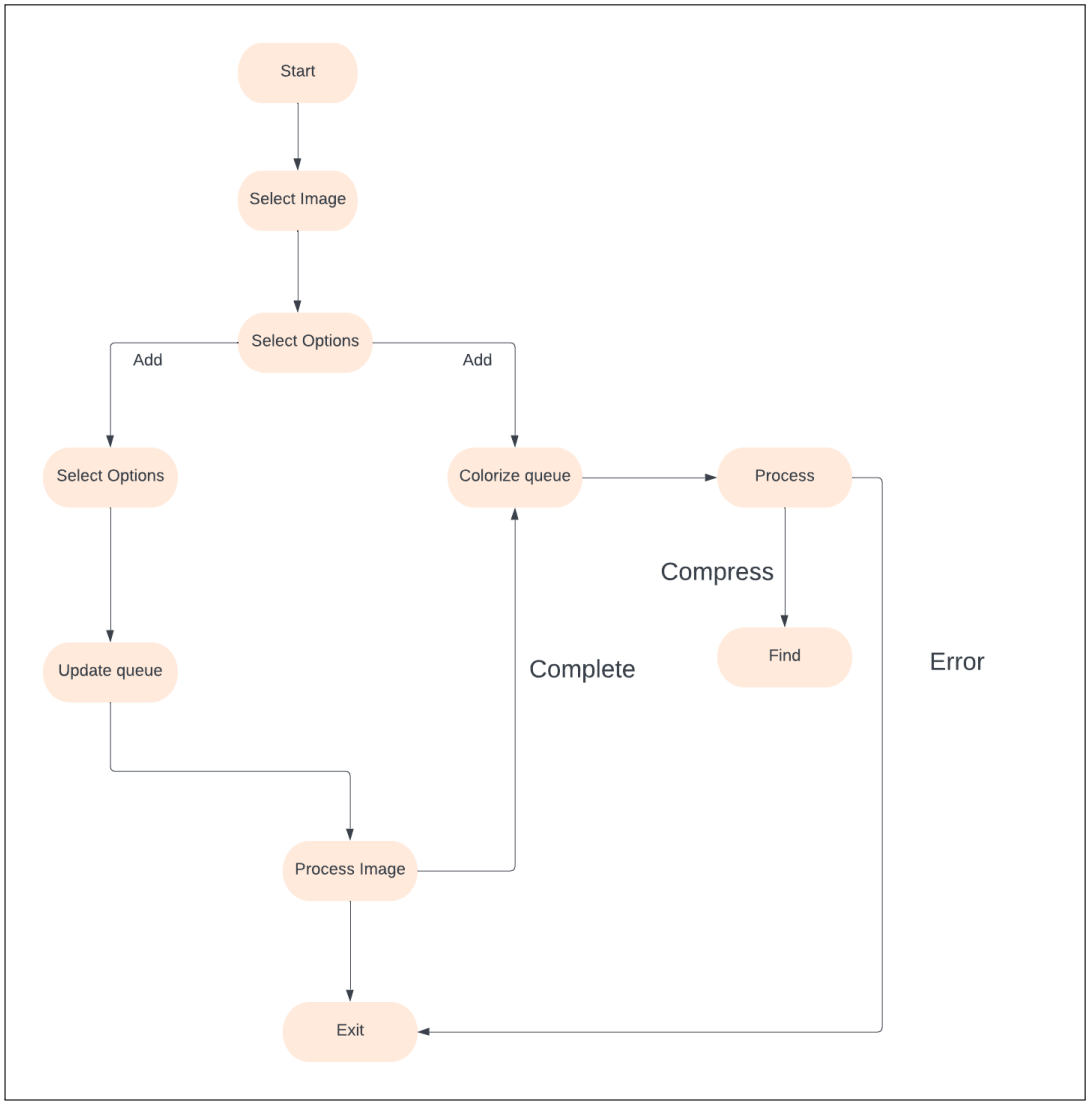


Figure 6.8: State transition diagram

CHAPTER 7

DETAILED DESIGN DOCUMENT USING APPENDIX A AND B

7.1 INTRODUCTION

This document specifies the design that is used to solve the problem of the Product.

7.2 ARCHITECTURAL DESIGN

A description of the program architecture is presented. Subsystem design or Block diagram, Package Diagram, and Deployment diagram with description are to be presented.

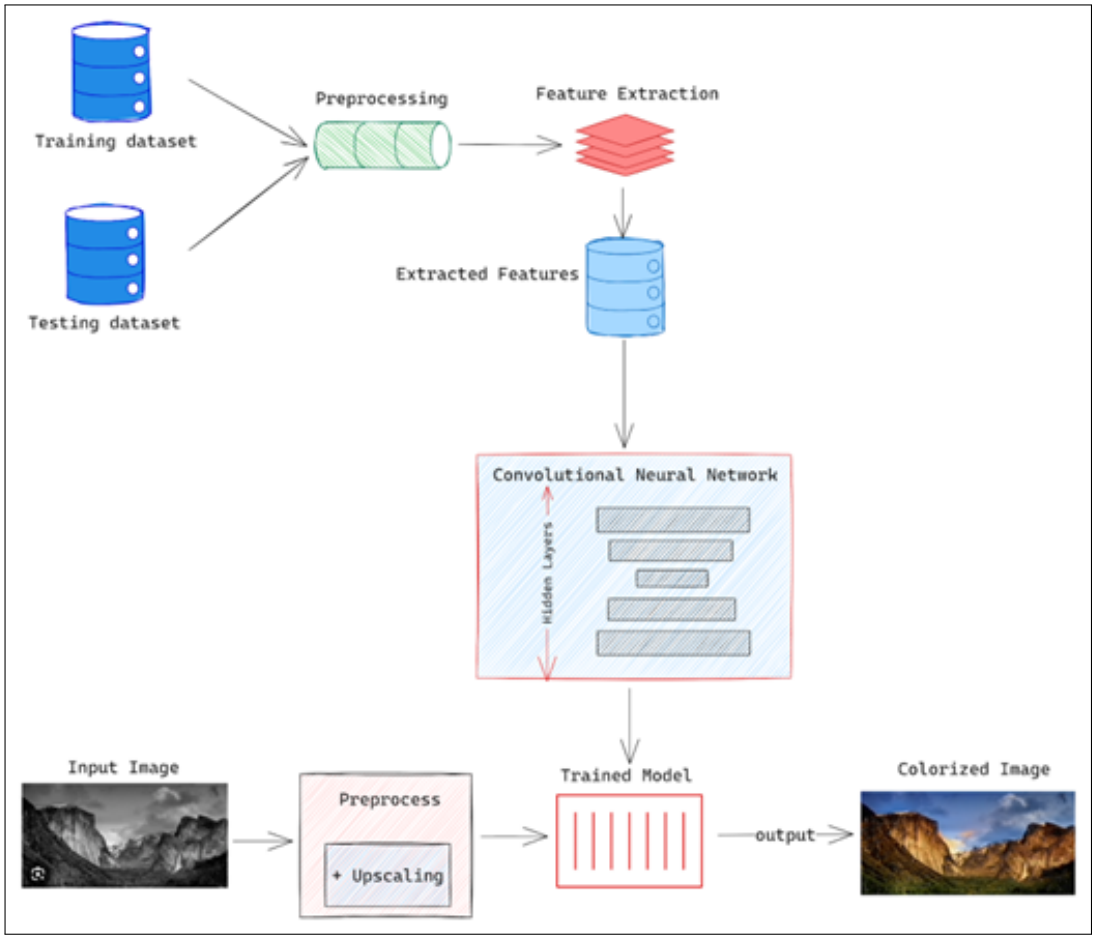


Figure 7.1: Architecture diagram

GANs Architecture: The generator network takes grayscale images as input and generates colorized images as output. The discriminator network receives either real color images or fake colorized images and aims to classify them correctly. Both

networks are designed using convolutional neural network (CNN) architectures to capture spatial dependencies in images effectively.

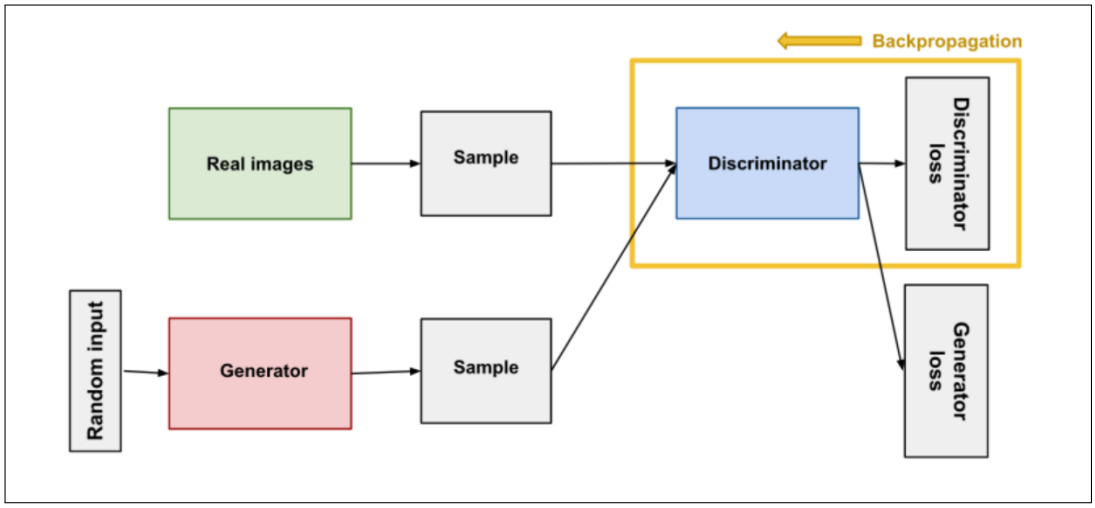


Figure 7.2: GANs Architecture diagram

7.3 DATA DESIGN (USING APPENDICES A AND B)

A description of all data structures including internal, global, temporary data structures, database design (tables), file formats.

7.3.1 Internal software data structure

Data structures that are passed among components of the software are described.

7.3.2 Global data structure

Data structures that are available to major portions of the architecture are described.

7.3.3 Temporary data structure

Files created for interim use are described.

7.3.4 Database description

Database(s) / Files created/used as part of the application is (are) described.

7.4 COMPONENT DESIGN

Class diagrams, Interaction Diagrams, Algorithms. Description of each component description required.

7.4.1 Class Diagram

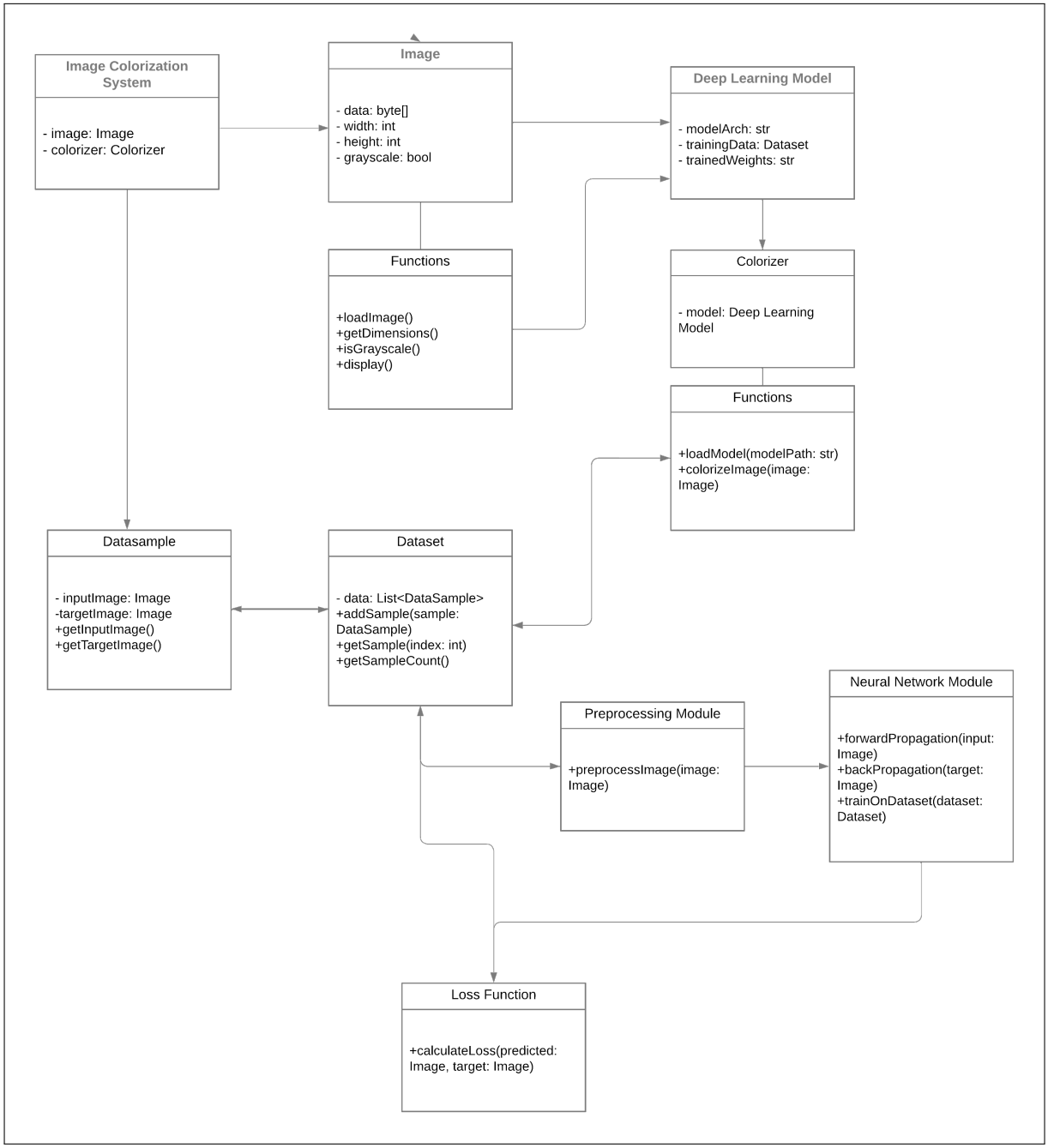


Figure 7.3: Class Diagram

CHAPTER 8

PROJECT IMPLEMENTATION

8.1 INTRODUCTION

Image colorization has been a significant advancement in the field of computer vision and image processing, enhancing the visual appeal and usability of grayscale images by automatically adding color. This complex task is often tackled using deep learning techniques, which have proven effective in handling the intricacies involved in predicting and applying appropriate colors. Leveraging Generative Adversarial Networks (GANs) and Convolutional Neural Networks (CNNs), researchers can achieve realistic and accurate colorization, transforming monochrome images into vibrant, lifelike versions. This technological progress not only improves the aesthetic quality of images but also broadens their applicability across various domains.

8.2 TOOLS AND TECHNOLOGIES USED

8.2.1 Python

Python is a programming language known for its simplicity and ease of use. Python's emphasis on readability, accessibility, open-source development, high-level abstraction, and interpreted execution make it a powerful and approachable language for a wide range of programming tasks.

8.2.2 NumPy

NumPy serves as a foundational library for numerical computing in Python. It offers a high-performance data structure: the multidimensional array. This structure, along with accompanying tools, facilitates efficient manipulation of these arrays. Beyond numerical data, NumPy can also function as a versatile container for multidimensional data of various types. Notably, NumPy's ability to handle user-defined data types enables seamless and rapid integration with diverse databases.

8.2.3 Matplotlib

Matplotlib is a powerful Python library for making data come alive visually. It's a go-to tool in data science and machine learning for creating all sorts of charts and graphs, from lines and bars to scatters and histograms. Some of the common

charts you can make with Matplotlib include line graphs, bar charts, histograms, scatter plots, and pie charts. It shows great versatility by creating different kinds of visualizations and allowing customization.

8.2.4 TensorFlow

TensorFlow constitutes an open-source software library designed for high-performance numerical computation. Within the domain of machine learning, it offers a versatile framework for the development and training of complex models. This enables researchers and practitioners to conduct data analysis, identify patterns within datasets, and generate predictions. Notably, TensorFlow's capability lies in its adept handling of large datasets and its ability to execute computations across diverse computing architectures, encompassing Central Processing Units (CPUs), Graphics Processing Units (GPUs), and Tensor Processing Units (TPUs). By virtue of these attributes, TensorFlow has emerged as a prominent tool for the construction and deployment of machine learning applications.

8.2.5 Keras

Keras serves as a high-level application programming interface (API) designed to streamline the development of deep learning models with the TensorFlow library. This user-friendly interface facilitates the rapid prototyping and experimentation with various neural network architectures. Keras offers a collection of pre-built building blocks, including layers and optimizers, that can be seamlessly integrated to construct complex deep learning models. Furthermore, it boasts a concise and readable syntax, fostering code maintainability and collaboration among researchers. In essence, Keras acts as a bridge between the underlying computational power of TensorFlow and the needs of deep learning practitioners, enabling them to focus on model design and experimentation.

8.3 METHODOLOGIES FOR PREDICTION

8.3.1 Fundamental Analysis

we employed a multi-stage training process. Initially, we trained a CNN model. To guide the learning process, we utilized mean squared error (MSE) as the loss function, aiming to minimize the difference between the colorized output and the ground truth color images. Throughout training, we meticulously monitored the model's performance on both the training data and a separate validation set, evaluating metrics such as MSE and perceptual similarity index (PSNR). The training procedure encompassed multiple epochs, typically lasting until convergence or employing early stopping mechanisms to prevent overfitting. Our objective was to generate vibrant and lifelike colorizations from grayscale input images, with accuracy metrics quantifying the fidelity of the colorized outputs.

8.3.2 Technical Analysis

In our image colorization project using CNNs and GANs, the technical approach involves leveraging advanced deep learning techniques to enhance the visual appeal and realism of grayscale images. The GAN framework facilitates an adversarial game between the generator and discriminator networks, where the generator learns to produce realistic colorized images from grayscale inputs, while the discriminator learns to distinguish between real and fake color images. Concurrently, the CNN model refines the colorization process by learning hierarchical patterns and features from the input images. Through extensive training on large-scale datasets containing paired grayscale and color images, our goal is to improve the accuracy and efficiency of image colorization, ultimately enhancing the usability and aesthetic quality of grayscale visual content.

8.3.3 Machine Learning

The image colorization project represents an innovative application of machine learning, harnessing advanced computational techniques to predict and apply colors to grayscale images. Supervised learning algorithms were employed to train models on

labeled datasets containing pairs of grayscale and color images. Particularly, convolutional neural networks (CNNs) were instrumental in automatically extracting and learning hierarchical features from input images, enabling accurate colorization predictions. The project focused on optimizing model architectures, training parameters, and loss functions to achieve the desired colorization accuracy. By leveraging machine learning, we aimed to revolutionize the process of image colorization, offering automated and efficient solutions for enhancing visual content across diverse applications and domains.

8.4 ALGORITHM

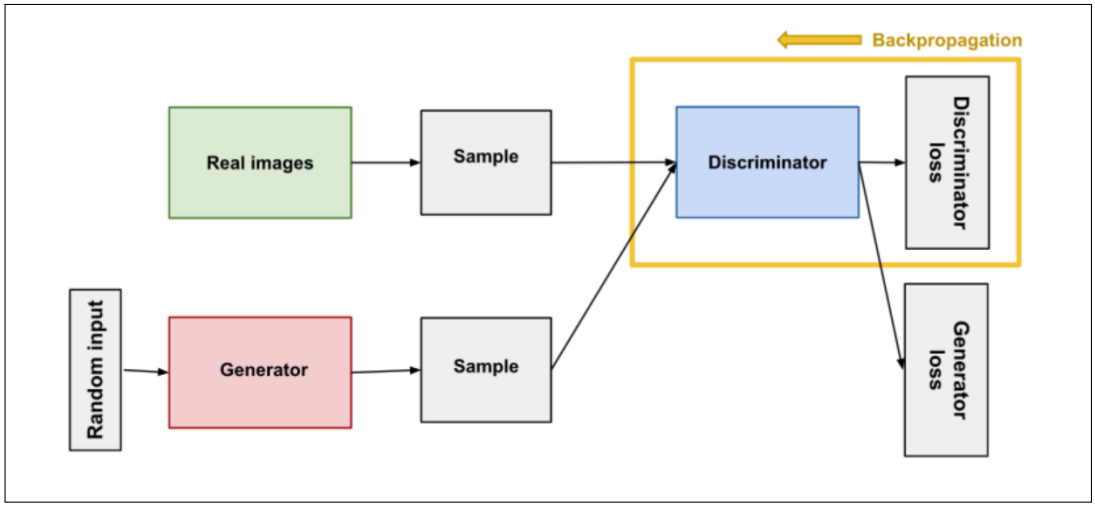


Figure 8.1: GANs Architecture diagram

We utilized a sophisticated algorithmic approach combining Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs). CNNs, specifically architectures like U-Net, were employed to perform the primary colorization task. These models extract hierarchical features from grayscale input images, refining the colorization process through multiple convolutional layers. Concurrently, GANs were incorporated to enhance the realism and quality of colorized outputs. Through adversarial training, the generator network learned to produce convincing colorizations, while the discriminator network became adept at distinguishing real from fake images. The training process involved iteratively updating both networks using techniques like batch normalization and advanced loss functions to optimize

colorization accuracy. By leveraging CNNs and GANs, our algorithmic approach aimed to transform grayscale images into vibrant and lifelike color versions, offering a novel solution for enhancing visual content.

8.4.1 Preprocessing Data

Data preprocessing is a critical step in preparing the dataset for our image colorization project using GANs and CNNs. To ensure consistency, all images are resized to a fixed dimension and normalized to a range of 0 to 1, which accelerates training and helps the model converge more quickly. Data augmentation techniques, such as random rotations, flips, and zooming, are applied to increase the diversity of the training set and enhance the model's robustness. For datasets containing only color images, grayscale conversion is performed to create the necessary input pairs. The dataset is then split into training, validation, and test sets to enable model training, hyperparameter tuning, and final performance evaluation. Each grayscale image is paired with its corresponding color image, facilitating supervised learning where the model learns to map grayscale inputs to their color counterparts. Batch processing is employed to handle large datasets efficiently, leveraging GPU acceleration for faster training times. This thorough preprocessing pipeline ensures that the input images are standardized, augmented, and properly paired, providing a solid foundation for effective model training and high-quality colorization results.

8.4.2 Model Architecture

The generator network takes grayscale images as input and generates colorized images as output.The discriminator network receives either real color images or fake colorized images and aims to classify them correctly.Both networks are designed using convolutional neural network (CNN) architectures to capture spatial dependencies in images effectively.

8.4.3 Training Process

The training process for image colorization using Generative Adversarial Networks (GANs) involves an adversarial game between two neural networks: the generator

and the discriminator. The generator's goal is to produce realistic colorized images from grayscale inputs, aiming to fool the discriminator. Meanwhile, the discriminator learns to distinguish between real color images and those generated by the generator. Through iterative training, where the generator improves its outputs to deceive the discriminator, and the discriminator enhances its ability to detect fake images, both networks progressively improve. This dynamic interaction results in the generation of highly realistic and accurate colorized images. Additionally, we used a loss function to guide the training process and monitored the accuracy for both training and validation sets. The model was trained for 50 epochs, with early stopping likely implemented based on validation performance to prevent overfitting.

CHAPTER 9

SOFTWARE TESTING

9.1 FORMAL TECHNICAL REVIEWS

Technical reviews were conducted four times in each semester. The project was presented by all team members to the internal guide as well as an external guide. In the first technical review conducted, we explained the objective, motivation, and scope of the project. The algorithms were studied and compared, and the possible output of the project was discussed for further improvement. During this formal technical review, the panel members guided us by giving suggestions regarding the system. The reviews were focused on the following aspects:

- Find accuracy for each model by providing test cases.
- Decide overall flow of the system.

In the next technical review, the system module of the proposed system was explained in detail. A more thorough explanation of the working and algorithm was provided along with the implementation. The following points were suggested:

- Improve the Model of Detection (Model Selection and Optimization).
- Create a more user-friendly GUI.

The final review concluded with the complete implementation and a detailed explanation of the project and its applications.

CHAPTER 10

RESULTS

10.1 TESTING RESULTS

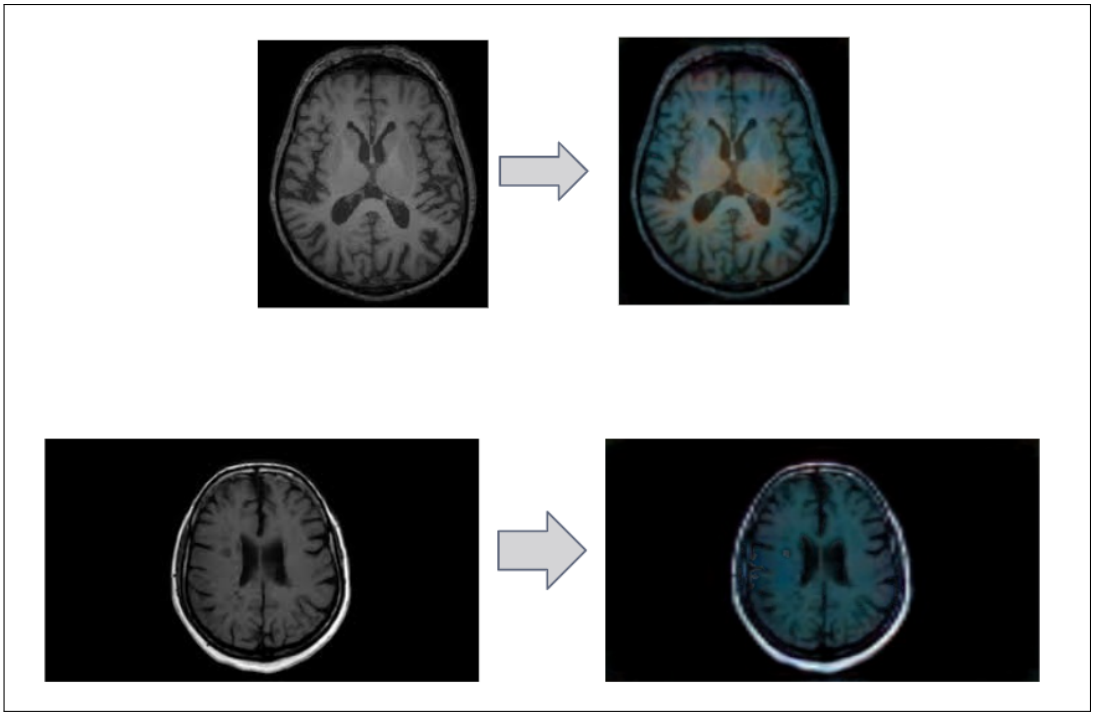


Figure 10.1: Test output

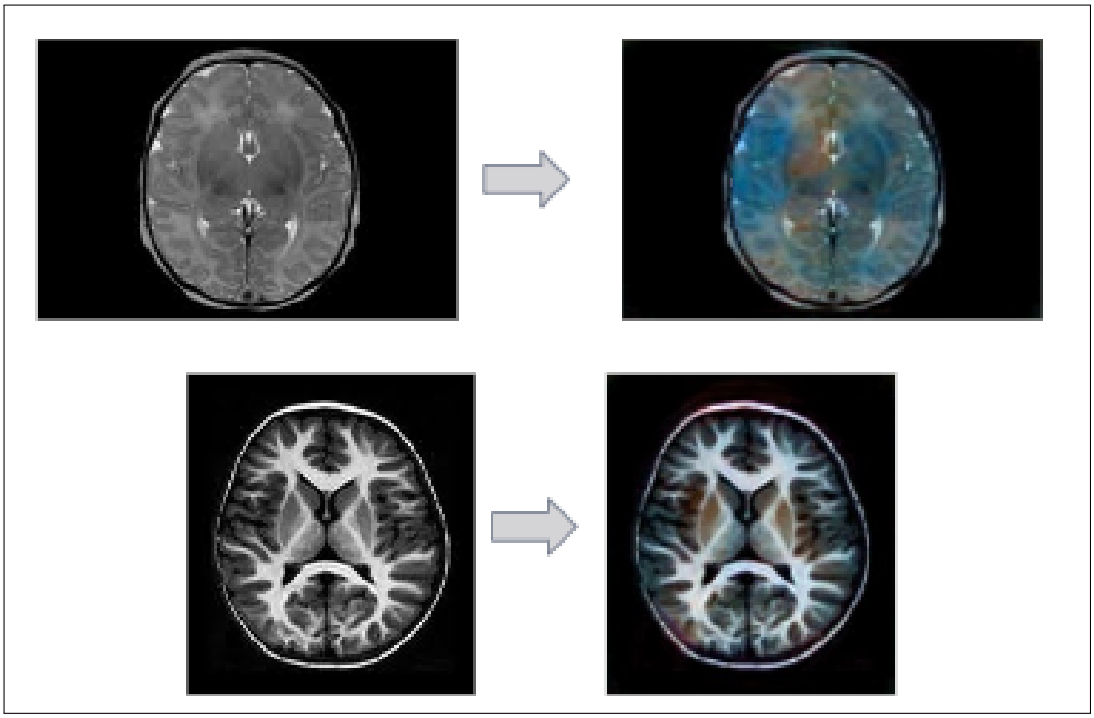


Figure 10.2: Test output

CHAPTER 11

DEPLOYMENT AND MAINTENANCE

11.1 INSTALLATION

To deploy and maintain the Image colorization application, follow these steps for installation and setup:

11.1.1 Processing

- **Python Installation**: Ensure Python is installed on your system. The recommended version is Python 3.6 or higher.
- **Library Installation**: Install the necessary libraries by running the following commands:

  pip install opencv-python

  pip install numpy

  pip install matplotlib

  pip install keras

- **Import the Model**:Download the trained model. This can typically be done using TensorFlow or Keras APIs.

  model_ dir = os.path.join(base_ dir, ”model”)

  PROTOTXT = os.path.join(model_ dir, ”colorization_ deploy_ v2.prototxt”)

  POINTS = os.path.join(model_ dir, ”pts_ in_ hull.npy”)

  MODEL = os.path.join(model_ dir, ”colorization_ release_ v2.caffemodel”)

- **Framework Installation**: Install Tensorflow, open-source deep learning framework developed by Google Brain. It is designed to facilitate the development, training, and deployment of machine learning models:

  pip install tensorflow

By following these installation steps, you can set up the necessary environment to deploy and run the Image colorization application effectively. Regular maintenance involves keeping the libraries and frameworks up to date, monitoring system performance, and addressing any issues that arise during operation.

CHAPTER 12

SUMMARY AND CONCLUSION

Summary:

"Image colorization with OpenCV and Deep Learning" is a cutting-edge project that leverages the power of computer vision and deep learning techniques to automatically add color to grayscale images. This innovative approach combines the versatility of the OpenCV library with the capabilities of neural networks, particularly convolutional neural networks (CNNs) and Generative Adversarial Networks (GANs). The project involves tasks like image preprocessing, feature extraction, model architecture design, and fine-tuning to achieve its goal. Central to its success is the creation of a specialized dataset and the application of transfer learning to boost model performance. Beyond addressing technical challenges, this project offers practical applications in digital media, Medical Imaging Enhancement, historical photograph restoration, and various creative domains, enhancing visual content in an automated and efficient manner.

Conclusion:

In conclusion, Our project aims to advance the field of image colorization by providing a fully automatic and user-friendly system. We anticipate that this system will significantly enhance the quality and appeal of grayscale images, making them more visually engaging and usable. Moreover, the application of our techniques holds promise for medical imaging enhancement, potentially aiding in the interpretation of medical scans and facilitating clearer diagnostics.

CHAPTER 13

REFERENCES

References:

1. Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 415-423

2. Dahl, R.: Automatic colorization. In: http://tinyclouds.org/colorize/. (2016)

3. Charpiat, G., Hofmann, M., Scholkopf, B.: Automatic image colorization via multimodal predictions. In: Computer Vision ECCV 2008. Springer (2008) 126-139

4. Ramanarayanan, G., Ferwerda, J., Walter, B., Bala, K.: Visual equivalence: towards a new standard for image delity. ACM Transactions on Graphics (TOG) 26(3) (2007) 76

5. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

6. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence 35(8)(2013) 1798-1828

7. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: Proceedings of the 28th international conference on machine learning (ICML-11). (2011) 689-696

8. Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 37-45

9. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia, ACM (2014) 675-678

10. Krahenbuhl, P., Doersch, C., Donahue, J., Darrell, T.: Data-dependent initializations of convolutional neural networks. International Conference on Learning Representations (2016)

CHAPTER 14

REFERENCES

[1] R. S. Pressman, *Software Engineering (3rd Ed.): A Practitioner's Approach*. New York, NY, USA: McGraw-Hill, Inc., 1992.

[2] P. Kulkarni, *Knowledge Innovation Strategy*. Pune: Bloomsbury Publication, 2015.

[3] P. Sinha, *Electronic Health Record*. IEEE Press Wiley.

[4] McKinsey, "Big data: The next frontier for innovation, competition, and productivity," tech. rep.

[5] "Digital competition:http://www.mckinsey.com/insights."

[6] "Government website to support entrepreneurs:http://msme.gov.in/mob/home.aspx."

ANNEXURE A

LABORATORY ASSIGNMENTS ON PROJECT ANALYSIS OF ALGORITHMIC DESIGN

- To develop the problem under consideration and justify feasibilty using concepts of knowledge canvas and IDEA Matrix.

  Refer [2] for IDEA Matrix and Knowledge canvas model. Case studies are given in this book. IDEA Matrix is represented in the following form. Knowledge canvas represents about identification of opportunity for product. Feasibility is represented w.r.t. business perspective.

I	D	E	A
Increase	Drive	Educate	Accelerate
Improve	Deliver	Evaluate	Associate
Ignore	Decrease	Eliminate	Avoid

Table A.1: IDEA Matrix

- Project problem statement feasibility assessment using NP-Hard, NP-Complete or satisfy ability issues using modern algebra and/or relevant mathematical models.

- input x,output y, y=f(x)

ANNEXURE B

LABORATORY ASSIGNMENTS ON PROJECT QUALITY AND RELIABILITY TESTING OF PROJECT DESIGN

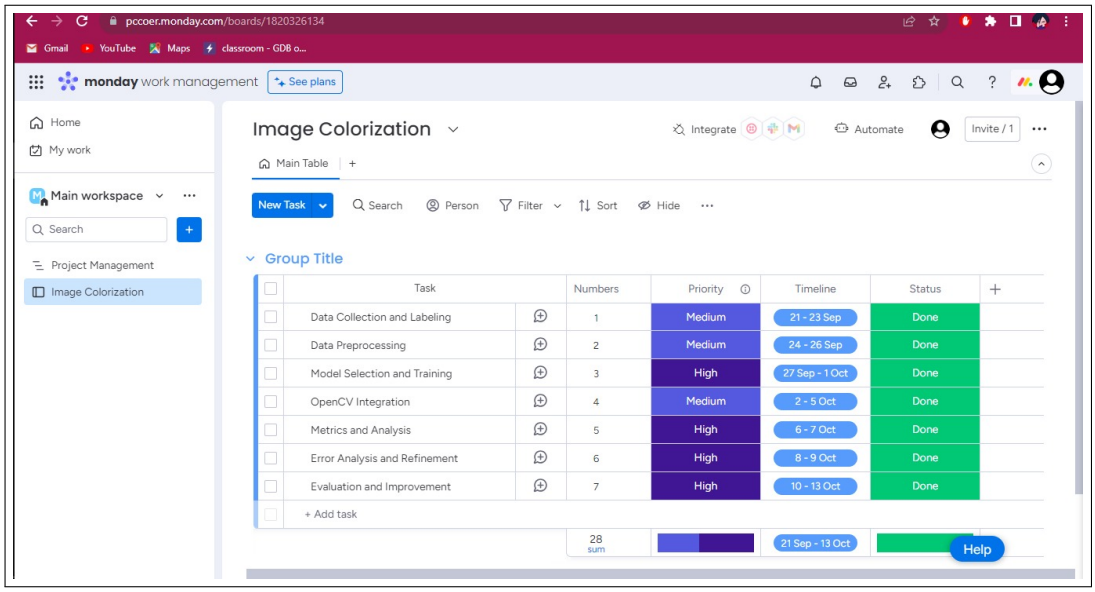
It should include assignments such as

- Use of divide and conquer strategies to exploit distributed/parallel/concurrent processing of the above to identify object, morphisms, overloading in functions (if any), and functional relations and any other dependencies (as per requirements). It can include Venn diagram, state diagram, function relations, i/o relations; use this to derive objects, morphism, overloading
- Use of above to draw functional dependency graphs and relevant Software modeling methods, techniques including UML diagrams or other necessities using appropriate tools.
- Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability. Write also test cases [Black box testing] for each identified functions. You can use Mathematica or equivalent open source tool for generating test data.
- Additional assignments by the guide. If project type as Entreprenaur, Refer [3],[4],[5], [6]

ANNEXURE C

PROJECT PLANNER

Using planner or alike project management tool.



ANNEXURE D

REVIEWERS COMMENTS OF PAPER SUBMITTED

(At-least one technical paper must be submitted in Term-I on the project design in the conferences/workshops in IITs, Central Universities or UoP Conferences or equivalent International Conferences Sponsored by IEEE/ACM)

1. Paper Title:
2. Name of the Conference/Journal where paper submitted :
3. Paper accepted/rejected :
4. Review comments by reviewer :
5. Corrective actions if any :

ANNEXURE E

PLAGIARISM REPORT

Plagiarism report

RE-2022-286839-plag-report

ORIGINALITY REPORT

14%	9%	2%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Savitribai Phule Pune University Student Paper	3%
2	Submitted to CTI Education Group Student Paper	1%
3	idoc.tips Internet Source	1%
4	Submitted to CSU Northridge Student Paper	1%
5	technodocbox.com Internet Source	1%
6	Zezhou Cheng, Qingxiong Yang, Bin Sheng. "Colorization Using Neural Network Ensemble", IEEE Transactions on Image Processing, 2017 Publication	1%
7	www.mdpi.com Internet Source	1%
8	github.com Internet Source	1%

9	www.essaysauce.com Internet Source	<1 %
10	shubhankar30.github.io Internet Source	<1 %
11	Submitted to Universiti Tenaga Nasional Student Paper	<1 %
12	pdfcoffee.com Internet Source	<1 %
13	cdn.techscience.cn Internet Source	<1 %
14	pdfslide.net Internet Source	<1 %
15	Submitted to SASTRA University Student Paper	<1 %
16	www.cc.gatech.edu Internet Source	<1 %
17	dokumen.pub Internet Source	<1 %
18	Submitted to iGroup Student Paper	<1 %
19	Submitted to TSU, University College Student Paper	<1 %
20	Submitted to International School of Management and Technology (ISMT), Nepal	<1 %

Student Paper

21	Submitted to University of Alabama at Birmingham Student Paper	<1 %
22	"Computer Vision – ECCV 2018 Workshops", Springer Science and Business Media LLC, 2019 Publication	<1 %
23	Submitted to Middlesex University Student Paper	<1 %
24	programtalk.com Internet Source	<1 %
25	Submitted to The Open Polytechnic of New Zealand Student Paper	<1 %
26	ierjournal.org Internet Source	<1 %
27	Submitted to University of Birmingham Student Paper	<1 %
28	www.geeksforgeeks.org Internet Source	<1 %
29	"Intelligent Systems Design and Applications", Springer Science and Business Media LLC, 2020 Publication	<1 %

30	Submitted to Colorado Technical University Online Student Paper	<1%
31	Submitted to Letterkenny Institute of Technology Student Paper	<1%
32	happydaycards.org Internet Source	<1%
33	www.biorxiv.org Internet Source	<1%
34	medium.com Internet Source	<1%
35	thebusinessresearchcompany.com Internet Source	<1%
36	www.slideshare.net Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		