# Abstract

At present the systems available right now only give you the feature of converting your black and white photo to a colored one but they do not give you that editing tools.

To work in the similar advancing scenario, we came up with a web application "COLORME – Add colors to life" that will help you to convert you old black and white to a colored one.

COLORME will help the user to get a colorized image instantly and fast without letting him to learn any complex software. It will also save his time and money and the processed image can be easily shared.

In this era where we're standing, we can say that Artificial Intelligence is a miracle invention of human brains. We will be able to depend on it almost entirely within a very short time that we can predict. In short we can say Artificial Intelligence is an intelligence which is been developed by a software or a machine. AI research is highly technical and specialized, and is deeply divided into sub fields that often fail to communicate with each other. AI research is also divided by several technical issues. A few sub fields focus on to solve the critical reasoning. The supreme area and problem where Artificial intelligence is working are reasoning, knowledge, planning, learning, natural language processing, perception and the ability to move and manipulate objects.

This Web Application will help you to relive your old memories that were taken in a black and white format converted into a colored one. The user will only need to upload the photo on the Website and with the help of different Machine Learning and Neural Networks technique that black and white photo will be converted into a colored one. After getting the processed image if the user wants to apply more editing such as changing contrast, lightness, hue, etc. the user will be provided with the same editing tools.

# Table of Content

**List of Figures** ..... **ix**

**List of Tables** ..... **x**

**1. Introduction**.....**1**

1.1 Rationale .....5

1.2 Goal .....5

1.3 Objective.....5

1.4 Overview of the Project.....5

1.5 Contribution of Project.....6

1.6 Report Organization..... 7

**2. Project Plan** .....**8**

2.1 Risk Management .....8

2.1.1 Project Risk.....8

2.2 Project Schedule.....8

2.2.1 Milestones & Deliverables..... 8

2.2.2 Schedule Chart ..... 10

2.3 Role & Responsibility of team members ..... 11

2.4 Process model adapted..... 12

**3. Requirement Engineering** ..... **13**

3.1 Requirements ..... 13

3.1.1 Functional Requirements ..... 13

3.1.2 Non Functional Requirements ..... 13

3.2 Use case Diagrams..... 14

3.3 Replacement of legacy system..... 15

3.4 Requirement flow..... 16

**4. Analysis and Conceptual Design and Technical Architecture**..... **17**

4.1 Technical Architecture..... 17

4.2 Sequence Diagrams..... 18

4.3 Data Flow Diagram ..... 18

4.4 Requirement Trace…….. 20

**5. Methodology**..... **22**

5.1 Proposed Algorithm .....22

5.2 Tools Required..... 23

**6. Implementation and testing**..... **27**

6.1 User Interface and Design .....27

6.2 Implementation Approaches..... 28

6.3 Testing Approaches..... 30

6.3.1 Unit Testing..... 30

6.3.2 Integration Testing..... 31

6.4 Modification and Improveements..... 31

**7. Results and Discussion**.....**32**

**Conclusion & Future Work**…….33

**Appendix A:** Project Synopsis……..34

**Appendix B:** Guide Interaction Report…….37

**Appendix C:** Project Snapshot……..38

# List of Figures

**Page No.**

Fig 1: Incremental Waterfall Diagram……..... 12

Fig 2: Use case Diagram ...……..... 15

Fig 3: Requirement flow diagram ……..…… .16

Fig 4: Technical Architecture ……... .17

Fig 5: Sequence Diagram …….. 18

Fig 6: Level 0 – Data Flow Diagram ……..…… 19

Fig 7: Level 1 – Data Flow Diagram ……..…… 19

Fig 8: Level 2 – Data Flow Diagram ……..……..19

Fig 9.1: Grid pixel in Black and White photo……..22

Fig 9.2: Representation of three color spectrum……..22

Fig9.3: Color encoding using three layers……...23

Fig 10: Project Snapshots ……...…38

# List of Tables

**Page No.**

Table 1: Risk Table……. 8

Table 2: Milestones and Deliverables……. 9

Table 3: Schedule Chart…….10

Table 4.1: Requirement trace (a) ..…….20

Table 4.2: Requirement trace (b) ..…….21

Table 5: White Box Testing……..31

Table 6: Guide Interaction Report……..38

# CHAPTER 1

# INTRODUCTION

# Chapter 1

# Introduction

At this era where we're standing, we can say that Artificial Intelligence is a miracle invention of human brains. We will be able to depend on it almost entirely within a very short time that we can predict. In short we can say Artificial Intelligence is an intelligence which is been developed by a software or a machine. AI research is highly technical and specialized, and is deeply divided into sub fields that often fail to communicate with each other. AI research is also divided by several technical issues. A few sub fields focus on to solve the critical reasoning. The supreme area and problem where Artificial intelligence is working are reasoning, knowledge, planning, learning, natural language processing, perception and the ability to move and manipulate objects.

To work in the similar advancing scenario, we came up with a web application "COLORME – Add colors to life" that will help you to convert you old black and white to a colored one.

This Web Application will help you to relive your old memories that were taken in a black and white format converted into a colored one. The user will only need to upload the photo on the Website and with the help of different Machine Learning and Neural Networks technique that black and white photo will be converted into a colored one. After getting the processed image if the user wants to apply more editing such as changing contrast, lightness, hue, etc. the user will be provided with the same editing tools. Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is François Chollet, a Google engineer. Chollet also is the author of the XCeption deep neural network model.

In 2017, Google's TensorFlow team decided to support Keras in TensorFlow's core library. Chollet explained that Keras was conceived to be an interface rather than a

standalone machine-learning framework. It offers a higher-level, more intuitive set of abstractions that make it easy to develop deep learning models regardless of the computational backend used. Microsoft added a CNTK backend to Keras as well, available as of CNTK v2.0.

Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel.

In addition to standard neural networks, Keras has support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization, and pooling.

Keras allows users to productize deep models on smartphones (iOS and Android), on the web, or on the Java Virtual Machine. It also allows use of distributed training of deep-learning models on clusters of Graphics Processing Units (GPU) and Tensor processing units (TPU).

TensorFlow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017. While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). TensorFlow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

TensorFlow computations are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors. During the Google I/O Conference in June 2016, Jeff Dean stated that 1,500 repositories on GitHub mentioned TensorFlow, of which only 5 were from Google.

A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes. Thus a neural network is

either a biological neural network, made up of real biological neurons, or an artificial neural network, for solving artificial intelligence (AI) problems. The connections of the biological neuron are modeled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed. This activity is referred as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be −1 and 1.

Unlike von Neumann model computations, artificial neural networks do not separate memory and processing and operate via the flow of signals through the net connections, somewhat akin to biological networks.

These artificial networks may be used for predictive modeling, adaptive control and applications where they can be trained via a dataset. Self-learning resulting from experience can occur within networks, which can derive conclusions from a complex and seemingly unrelated set of information.

The preliminary theoretical base for contemporary neural networks was independently proposed by Alexander Bain (1873) and William James (1890). In their work, both thoughts and body activity resulted from interactions among neurons within the brain.

For Bain, every activity led to the firing of a certain set of neurons. When activities were repeated, the connections between those neurons strengthened. According to his theory, this repetition was what led to the formation of memory. The general scientific community at the time was sceptical of Bain's theory because it required what appeared to be an inordinate number of neural connections within the brain. It is now apparent that the brain is exceedingly complex and that the same brain "wiring" can handle multiple problems and inputs.

James's theory was similar to Bain's, however, he suggested that memories and actions resulted from electrical currents flowing among the neurons in the brain. His model, by focusing on the flow of electrical currents, did not require individual neural connections for each memory or action.

C. S. Sherrington (1898) conducted experiments to test James's theory. He ran electrical currents down the spinal cords of rats. However, instead of demonstrating an increase in electrical current as projected by James, Sherrington found that the

electrical current strength decreased as the testing continued over time. Importantly, this work led to the discovery of the concept of habituation.

McCulloch and Pitts (1943) created a computational model for neural networks based on mathematics and algorithms. They called this model threshold logic. The model paved the way for neural network research to split into two distinct approaches. One approach focused on biological processes in the brain and the other focused on the application of neural networks to artificial intelligence.

In the late 1940s psychologist Donald Hebb created a hypothesis of learning based on the mechanism of neural plasticity that is now known as Hebbian learning. Hebbian learning is considered to be a 'typical' unsupervised learning rule and its later variants were early models for long term potentiation. These ideas started being applied to computational models in 1948 with Turing's B-type machines.

Farley and Clark (1954) first used computational machines, then called calculators, to simulate a Hebbian network at MIT. Other neural network computational machines were created by Rochester, Holland, Habit, and Duda (1956).

Rosenblatt (1958) created the perceptron, an algorithm for pattern recognition based on a two-layer learning computer network using simple addition and subtraction. With mathematical notation, Rosenblatt also described circuitry not in the basic perceptron, such as the exclusive-or circuit, a circuit whose mathematical computation could not be processed until after the backpropagation algorithm was created by Werbos (1975).

Neural network research stagnated after the publication of machine learning research by Minsky and Papert (1969). They discovered two key issues with the computational machines that processed neural networks. The first issue was that single-layer neural networks were incapable of processing the exclusive-or circuit. The second significant issue was that computers were not sophisticated enough to effectively handle the long run time required by large neural networks. Neural network research slowed until computers achieved greater processing power. Also key in later advances was the backpropagation algorithm which effectively solved the exclusive-or problem (Werbos 1975).

The parallel distributed processing of the mid-1980s became popular under the name connectionism. The text by Rumelhart and McClelland (1986) provided a full exposition on the use of connectionism in computers to simulate neural processes.

Neural networks, as used in artificial intelligence, have traditionally been viewed as simplified models of neural processing in the brain, even though the relation between this model and brain biological architecture is debated, as it is not clear to what degree artificial neural networks mirror brain function.

## 1.1 Rationale

Coloring a black and white photo with the help of software's such as Adobe Photoshop is a tiring and hardworking job. However, there are certain systems that help to colorize the photo without these tools but are costly or does not provide other features. So we came up with COLORME a simple and efficient web application that will provide features:

- Coloring Black and White photo
- Converting to various color modes

## 1.2 Goal

As compared to the traditional way of converting black and white photo into colored one using Photoshop and other photo software editing tools we can use COLORME in an efficient way of converting photo.

## 1.3 Objective

The objective of the works is to propose options for editing your black and white photo. To do this it requires to:

- To understand the topic of neural networks and study the previous used algorithms.
- Find out difference in all the existing systems.
- To apply different machine learning techniques and training of the neural network.
- To apply to make a new approach for defining the system and making it more faster and efficient
- To analyze the faults in the system and what improvements are to be done in the system

## 1.4 Overview

- The planning and design stage:

  The planning and design stage is concerned primarily with defining of classes, objects, database schema, etc. The feedback activities are compiled into a feedback plan for the unit/module that specifically tries to satisfy the stated learning outcomes.

- The development stage:

  The development stage is concerned with the creation of resources to support the user experience and with the creation of mobile feedback and assessment environments.

- The performance and evaluation stage:

  The performance is concerned with plotting of different types of graphs & evaluating performance and reliability of our application. We have used firebase analytics to analyze the efficiency of database schema.

## 1.5 Contribution of the project

With the help of this project the users will be able to save their time and energy. The user could get rid of some advance software's such which allows the user to colorize your black and white photo manually. It will definitely save time and money for the user.

### 1.5.1 Market Potential

Our project will have an adverse effect on the historians as well as normal people. Excessive use of the app will only make the application more efficient and could help better predict the colors in it.

### 1.5.2 Innovativeness

The existing systems available right now only provide the feature of adding colors but with the help of our web application the user could not only color his photo but after coloring could add extra effects in editing as per his choice.

### 1.5.3 Usefulness

COLORME will help the user to convert his black and white photo to the colored one. With the added feature of adding extra effects and editing the photo according to his wish.

## 1.6 Report Organization

The remaining section of the report is structured as follows.

**Chapter 2** provides information about different steps taken for project plan and project schedule.

**Chapter 3** provides details of requirements and analysis of the diagrams drawn for a better understanding of the project work and its progress.

**Chapter 4** deals with how the project has been implemented from design to layout to the functionality of the application. Chapter 4 also includes the analysis of conceptual design and technical architecture of the project.

**Chapter 5** describes the methodology that we applied for the development of the project and what tools we used.

**Chapter 6** provides details of Implementation & testing of application.

**Chapter 7** describes the future scope of project and conclusion.

# CHAPTER 2

# PROJECT PLAN

# CHAPTER 2

# Project Plan

## 2.1 Risk Management

### 2.1.1 Project Risk

Risk assessment is the determination of quantitative or qualitative estimate of risk related to a well-defined situation and a recognized threat (also called hazard). Once risks have been identified, they must then be assessed as to their potential severity of impact (generally a negative impact, such as damage or loss) and to the probability of occurrence. Numerous different risk formulae exist, but perhaps the most widely accepted formula for risk quantification is: "Rate (or probability) of occurrence multiplied by the impact of the event equals risk magnitude.

| Risk | Affects | Description |
|---|---|---|
| Team failure | Project | Experienced team member leaves the project before finished |
| Management Change | Project | There will be a change of organization management with different priorities |
| Hardware Unavailability | Project | Hardware that is essential for project to run on. |
| Requirements change | Project and product | There will be a large number of changes to the requirements than anticipation. |
| Specification delay | Project and product | Specifications of essential interface are not available on schedule. |
| Technology change | Business | The underlying technology on which the system is built is superseded by new technology. |
| Project competition | Business | A competitive product in market before the product is being completed. |

Table 1: Risk Table

## 2.2 Project Schedule

### 2.2.1 Milestone and Deliverables

A deliverable is a measurable and tangible outcome of the project. They are developed by project team members in alignment with the goals of the project. Milestones on the other hand are checkpoints throughout the life of the project.

They identify when one or multiple groups of activities have been completed thus implying that a notable point has been reached in the project.

The project team defines the milestones during project planning in order to identify the appropriate project activities to be accomplished; and to set planned targets. Milestones can indicate the completion of key project tasks, the commencement or conclusion of the plan. This means that milestones can be inserted at the start, finish and at notable check points throughout the project schedule. The milestone represents the completion of a major step in the project that requires the commitment of a certain amount of time, resources and effort; and the deliverable indicates the measurable item that was created during this commitment of time, resources and effort.

| Deliverable | Content |
|---|---|
| Deliverable 0 | Preliminary Project Plan |
| Deliverable 1 | Part 1 Interim Progress<br>* Project Plan<br>* Improved Understanding Document<br>* PowerPoint |
| Deliverable 2 | Part 1 Final Report<br>* Project Plan<br>* Improved Understanding Document<br>* Mock Prototype |
| Deliverable 3 | Part 2 Interim Progress<br>* Project Plan<br>* Improved Understanding Document<br>* Traceability Matrix |
| Deliverable 4 | Part 2 Final Report<br>* Project Plan<br>* Improved Understanding Document<br>* Traceability Matrix<br>* PowerPoint<br>* Final Prototype |

Table 2: Milestones and Deliverables

### 2.2.2 Schedule Chart

| S. No. | Activity | Date |
|---|---|---|
| 1 | Formulation of Project Team & Guide Allotment | 25/7/18 |
| 2 | Synopsis Submission to Project Guide | 27/7/18 |
| 3 | Revise Project Synopsis & Discussion | 6/8/18 |
| 4 | Project Plan Submission by Project Team | 16/8/18 |
| 5 | Domain Modelling<br>Domain Diagrams | 03/9/18 |
| 6 | Requirement Definitions<br>Identify use-cases<br>Organize use-cases in logical groups<br>domain objects<br>Use-case Descriptions<br>Review Requirements | 20/9/18 |
| 7 | Analysis & Conceptual Design & Technical Architecture Design | 11/10/18 |
| 8 | Sequence Diagrams<br>Class Diagrams<br>Database Design<br>Requirement Traceability | 25/10/18 |
| 9 | End Semester Presentation | As per RGPV Notification |
| 10 | 7th Semester Completion presentation | 05/02/19 |
| 11 | System Design | 26/02/19 |
| 12 | Presentation and submission of coding | 27/03/19 |
| 13 | Testing | 02/04/19 |
| 14 | Final Presentation and Demonstration | 09/04/2019 |
| 15 | Report Submission -In prescribe format only<br>a. Group Member + 2 (Hard Copies)<br>b. 02 Soft Copy - In DVD Only<br>c. Log Book Signed by Guide | On the day of Internal Viva-Voce |

Table 3: Schedule Chart

## 2.3 Role & Responsibility of Team Members

### Amandeep Singh:

**Designer**

The designer is responsible for understanding the business requirements and designing a solution that will meet the business needs. There are many potential solutions that will meet the client's needs. The designer determines the best approach.

**Coder**

Advising on economic feasibility of implementing designs / requirements in available programming languages.

**Tester**

The Tester ensures that the solution meets the business requirements and that it is free of errors and defects.

### Aniketh Vyas:

**Designer**

The designer is responsible for understanding the business requirements and designing a solution that will meet the business needs. There are many potential solutions that will meet the client's needs. The designer determines the best approach.

**GUI Designer**

Guiding look-and-feel decisions. Helping write and clarify user stories and personas. Elaborating user interaction guidelines and standards. Analyzing the system-in-use so as to continually refine usability and the user interface.

**Tester**

The Tester ensures that the solution meets the business requirements and that it is free of errors and defects.

### Arpit Khatri:

**Coder**

Advising on economic feasibility of implementing designs / requirements in available programming languages.

**Developer**

The Developer is responsible for the actual building of the solution.

## Ashray Saraf:

**Coder**

Advising on economic feasibility of implementing designs / requirements in available programming languages.

**Subject Matter Expert**

A Subject Matter Expert (SME) has superior (expert) knowledge of a discipline, technology, product, business process or entire business area.

**Architect**

Ensuring the product or service under development achieves its performance and other qualitative requirements.

## 2.4 Process Model adopted

We've used incremented waterfall model for our project. The waterfall model is a sequential approach, where each fundamental activity of a process represented as a separate phase, arranged in linear order. The phases of the waterfall model are: Requirements, Design, Implementation, Testing, and Maintenance.

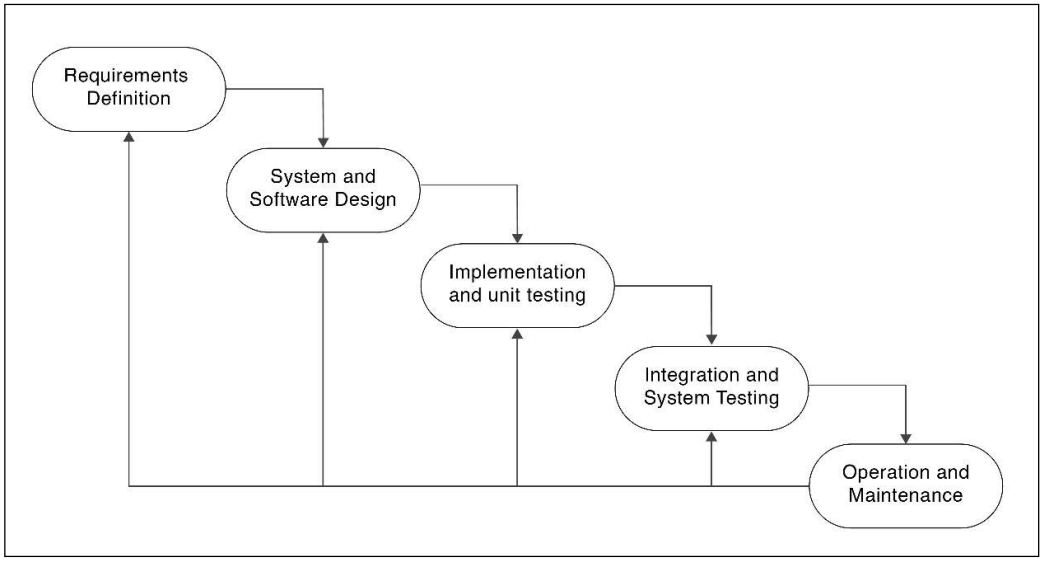


Fig 1: Incremented Waterfall Model

# CHAPTER 3

# REQUIREMENT ENGINEERING

# Chapter 3

# Requirement Engineering

## 3.1 Requirements

### 3.1.1 Functional Requirements

- Business rules
  - A business rule defines some aspects of business and always resolves to either true or false. Business rules are intended to assert business structure or to control or influence the behavior of the business. Business rule describe the operations, definitions and constraints that apply to an organization.
- Unique code generation
  - The code generated in this algorithm possesses a unique code that could generate faster and better results.
- Authentication

### 3.1.2 Non Functional Requirements

- Scalability
  - Scalability is the capability of a system to handle a growing amount of work or its potential to be enlarged to accommodate that growth.
- Maintainability
  - The ease with which a software system component can be modified to correct faults, improve performance or other adapt to a changed environment.
- Response/Processing time
  - Response time is the total time it takes from when a user makes a request until they receive a response. Response time can be affected by changes to the processing time of your system and by changes in latency, which occur due to changes in hardware or resource utilization
- Availability
  - The degree to which a system, subsystem, or equipment is in a specified operable and committable state at the start of a mission when a mission is called for at an unknown, i.e. a random, time.

- Efficiency
  - Efficiency is the degree to which a software can generate desired results more efficiently and faster.
- Usability
  - Usability is the degree to which a software can be used by a specified consumer to achieve quantified objective with effectiveness, efficiency and satisfaction in a quantified context of use.
- Reliability
  - The probability of failure-free software operation for a specified period of time in a specified environment.

## 3.2 Use case Diagrams

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circle or ellipses.

Due to their simplistic nature, use case diagrams can be a good communication tool for stakeholders. The drawings attempt to mimic the real world and provide a view for the stakeholder to understand how the system is going to be designed. Siau and Lee conducted research to determine if there was a valid situation for use case diagrams at all or if they were unnecessary. What was found was that the use case diagrams conveyed the intent of the system in a more simplified manner to stakeholders and that they were "interpreted more completely than class diagrams".

The purpose of the use case diagrams is simply to provide the high level view of the system and convey the requirements in laypeople's terms for the stakeholders. Additional diagrams and documentation can be used to provide a complete functional and technical view of the system.

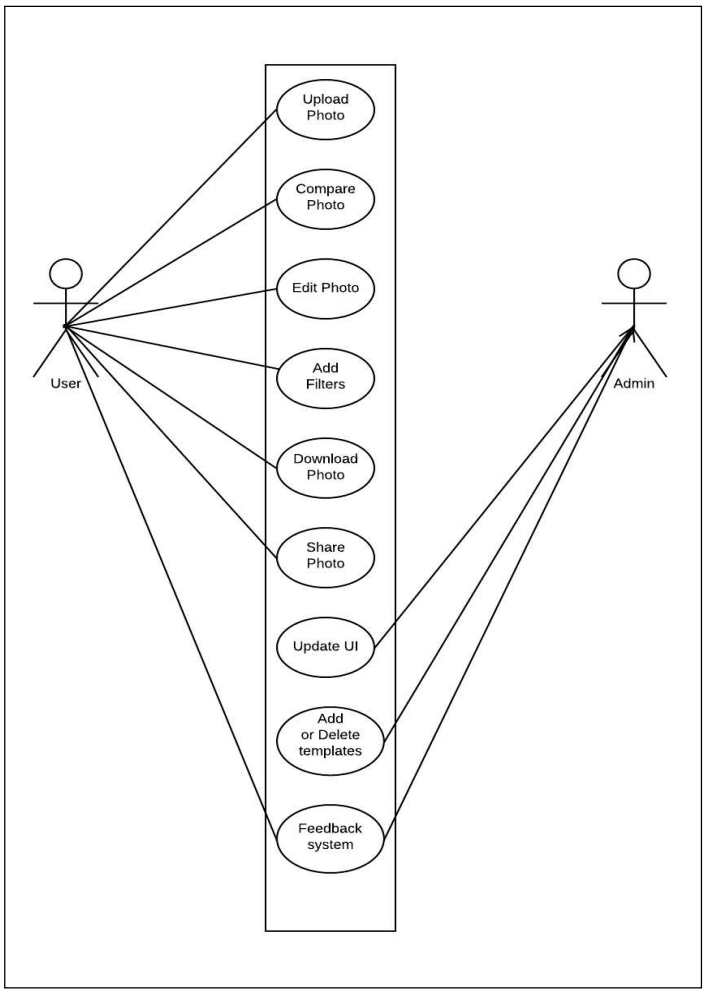


Fig 2: Use Case Diagram

## 3.3 Replacement of legacy system

If you know about artificial intelligence (AI), but are doing everything manually, you're not alone. Although Photoshop and other software tools have made colorization of Black and White photographs more efficient, many users might be performing time-consuming, error-prone manual processes without realizing it's costing you time and money.

But imagine that you are now using AI for the processes and the changes you are seeing are encouraging. There are less chances for error and you would be completed in seconds. Using the technique of AI, you not only complete your work faster but also efficiently.

In today's world, time has become more important than money, so many software machine developers are designing algorithms that are less complex and faster

processing. Algorithms play. Using Deep learning algorithms will not only make our work easier but also faster.

## 3.4 Activity Diagram

Activity diagrams are graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

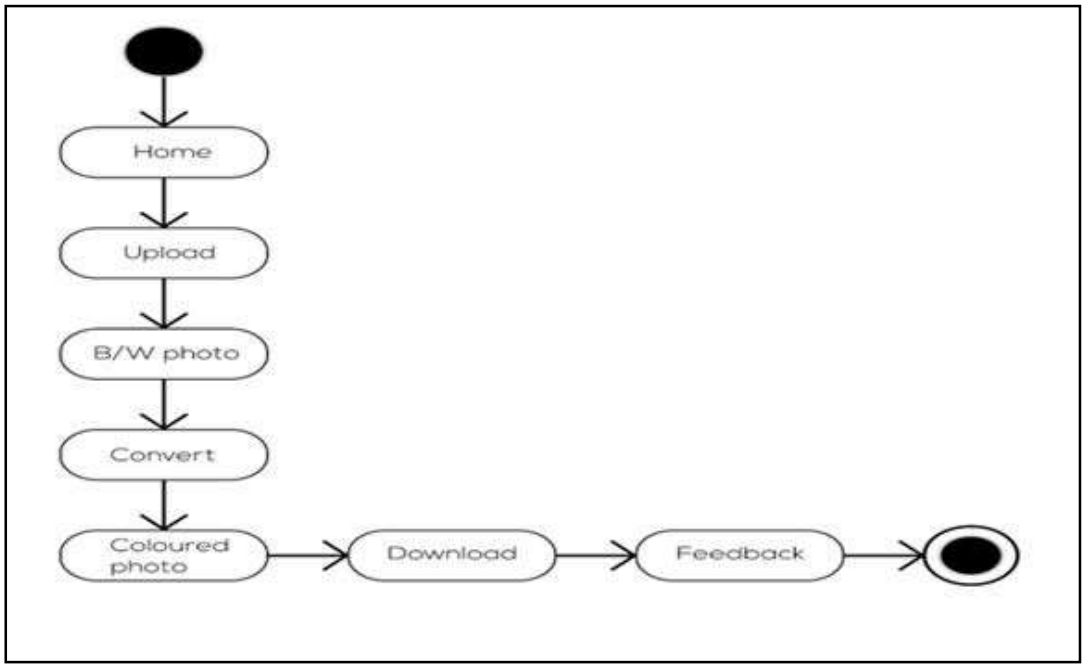


Fig 3: Requirement Flow

# CHAPTER 4

# ANALYSIS & CONCEPTUAL DESIGN & TECHNICAL ARCHITECTURE

# Chapter 4

# Analysis & Conceptual Design & Technical Architecture

## 4.1 Technical Architecture

A Technical architecture of a system is the conceptual model that defines the structure, behavior and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behavior of the system.

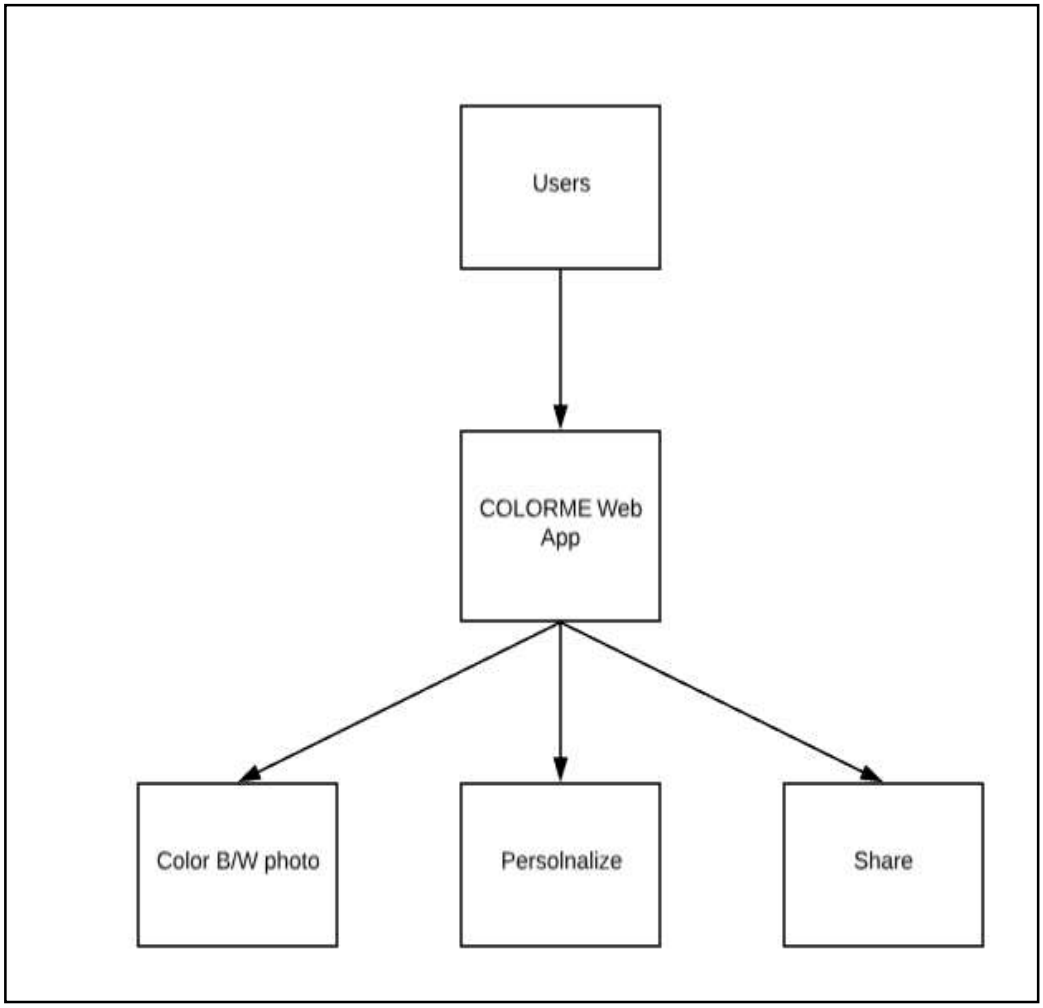


Fig 4: Technical Architecture

## 4.2 Sequence Diagram

Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines, different processes or objects that live simultaneously and as horizontal arrows, the message exchanged between them, in the order in which they occur.

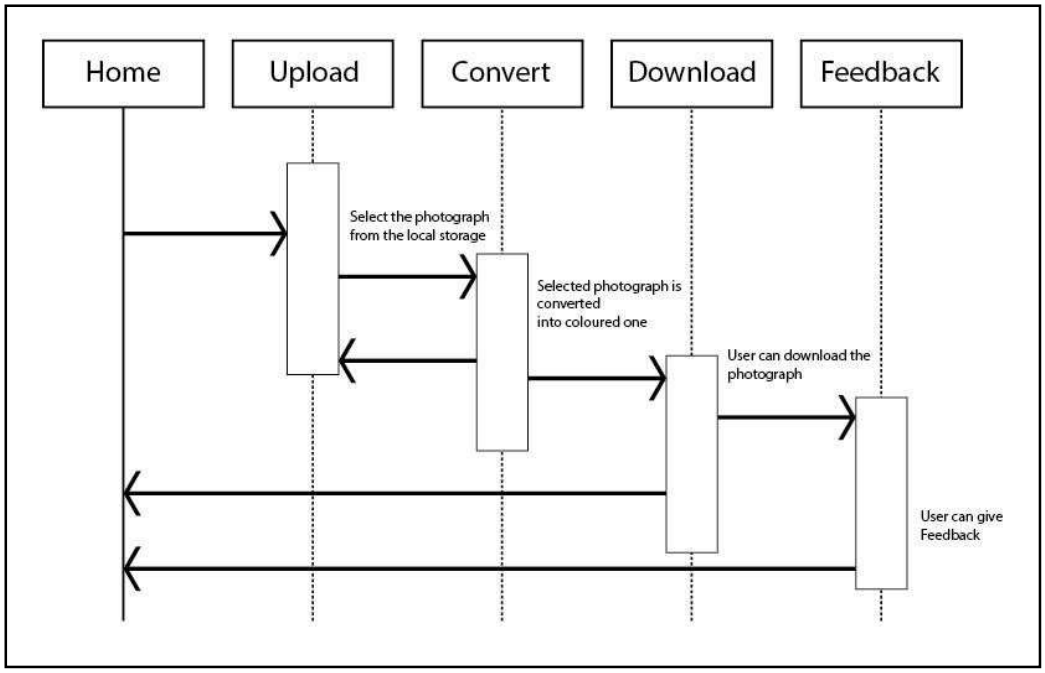


Fig 5: Sequence Diagram

## 4.3 Data Flow Diagram

A data flow diagram is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great details which can later be elaborated. DFDs can also be used for the visualization of data processing. A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system and where the data will be stored.

**Level 0**: A context diagram is a top level (also known as "Level 0") data flow diagram. It only contains one process node ("Process 0") that generalizes the function of the entire system in relationship to external entities.

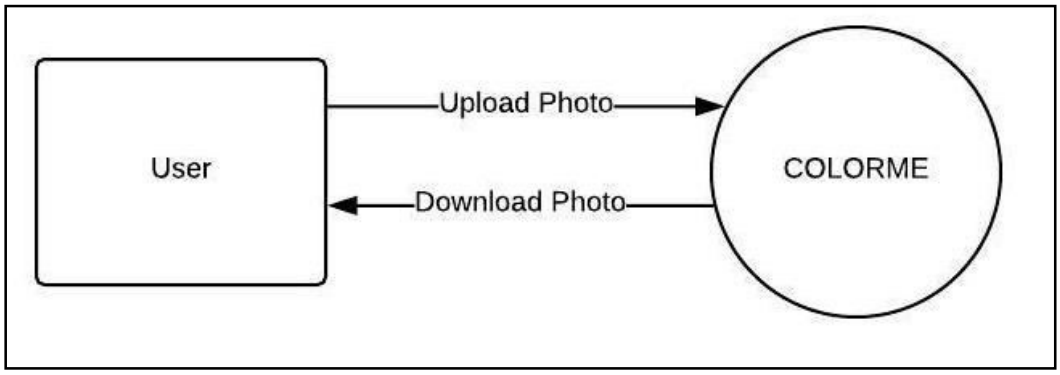


Fig 6: Level 0

**Level 1**: Interaction between 2 different business applications. This is primarily used to explain the process to business and tech leads, QA leads.

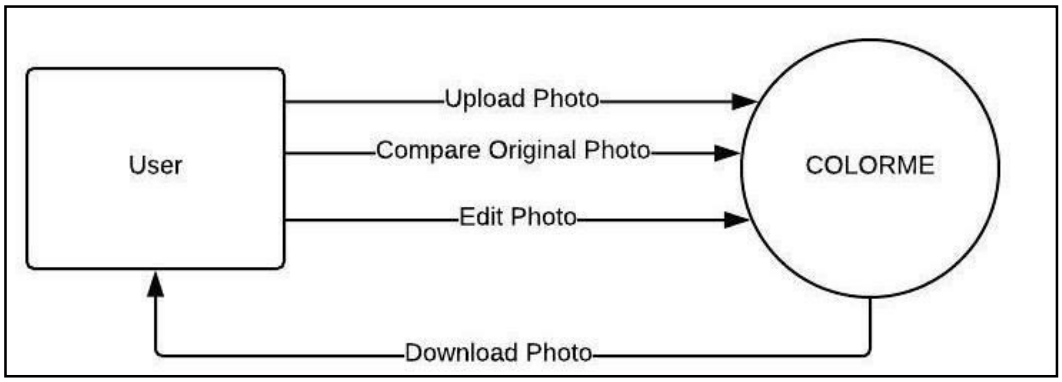


Fig 7: Level 1

**Level 2**: Interaction between 2 different modules within an application. This is used to explain the flow to developer teams.

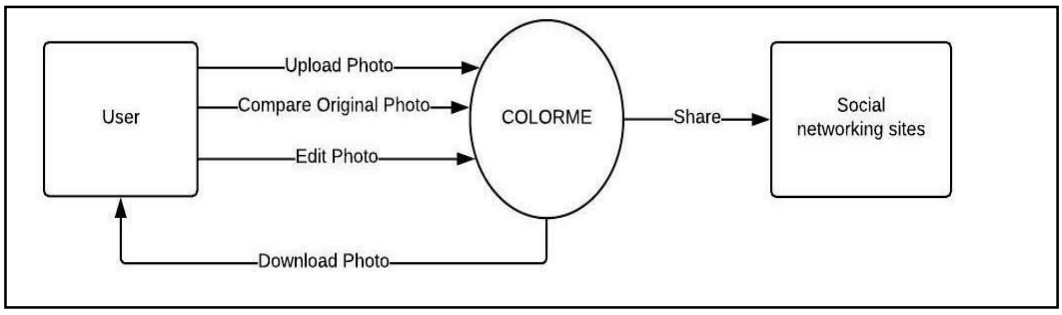


Fig 8: Level 2

## 4.4 Requirement Trace

### 4.4.1 Business Requirements

Business requirements are specifications which once delivered, provide value, it describes the characteristics of the proposed system from the viewpoint of system end user like a CONOPS and is also called stakeholder requirements specification (StRS). Products, systems, software, and processes are ways of how to deliver, satisfy, or meet business requirements. Consequently, business requirements are often discussed in the context of developing or procuring software or other systems.

| Business Requirement | Module Name | User | Business Requirement Description |
|---|---|---|---|
| **BSR1** | Home Page | User | User can successfully see the homepage. |
| **BSR2** | Uploading Photo | User | User can successfully upload a photo. |

Table 4.1: Requirement Trace (a)

### 4.4.2 Functional Requirements

In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behaviour between outputs and inputs. Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describe all the cases where the system uses the functional requirements, these are captured in use cases. Functional requirements are supported by non-functional requirements (also known as "quality requirements"), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). Generally, functional requirements are expressed in the form "system must do <requirement>," while non-functional requirements take the form "system shall be

<requirement>." The plan for implementing functional requirements is detailed in the system design, whereas non-functional requirements are detailed in the system architecture.

| Functional Requirements | Description |
|---|---|
| **FTR1** | Uploading photo is necessary to get a result |
| **FTR2** | The photo path cannot be invalid |
| **FTR3** | The photograph should not exceed the maximum size limit |
| **FTR4** | If photograph is uploaded correctly, then conversion is successful |
| **FTR5** | If photograph is not uploaded correctly, then conversion is unsuccessful |
| **FTR6** | User can see the homepage |

Table 4.2: Requirement Trace (b)

# CHAPTER 5

# METHODOLOGY

# Chapter 5

# Methodology

## 5.1 Proposed Algorithm

Black and white images can be represented in grids of pixels. Each pixel has a value that corresponds to its brightness. The values span from 0–255, from black to white.

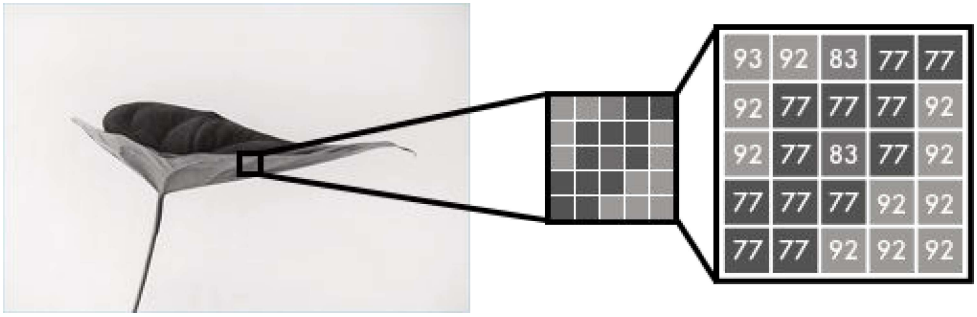


Fig 9.1: Grid pixel in Black and White photo

Color images consist of three layers: a red layer, a green layer, and a blue layer. This might be counter-intuitive to you. Imagine splitting a green leaf on a white background into the three channels. Intuitively, you might think that the plant is only present in the green layer.

But, as you see below, the leaf is present in all three channels. The layers not only determine color, but also brightness.

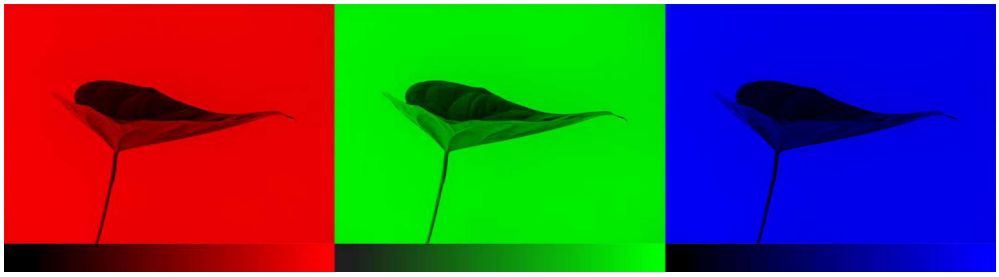

Fig 9.2: Representation of three color spectrum

To achieve the color white, for example, you need an equal distribution of all colors. By adding an equal amount of red and blue, it makes the green brighter. Thus, a color image encodes the color and the contrast using three layers:

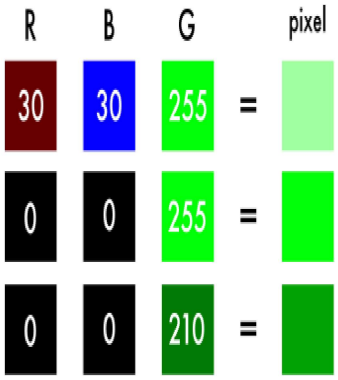


Fig 9.3: Color encoding using three layers

Just like black and white images, each layer in a color image has a value from 0–255. The value 0 means that it has no color in this layer. If the value is 0 for all color channels, then the image pixel is black.

As you may know, a neural network creates a relationship between an input value and output value. To be more precise with our colorization task, the network needs to find the traits that link grayscale images with colored ones.

In sum, we are searching for the features that link a grid of grayscale values to the three color grids.

## 5.2 Tools Required

### Python

Python is a multi-paradigm programming language mainly used for high level programming. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta-programming and meta-objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's design offers some support for functional programming in the Lisp tradition. It has filter(), map(), reduce() functions; list comprehensions, dictionaries, and sets; and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

While offering choice in coding methodology, the Python philosophy rejects exuberant syntax (such as that of Perl) in favor of a simpler, less-cluttered grammar.

**HTML**

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets.

Tags such as <img/> and <input/> tag directly introduce content into the page. Other tags such as <p>surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

**ANACONDA IDE**

Anaconda is a scientific Python distribution. It has no IDE of its own. The default IDE bundled with Anaconda is Spyder which is just another Python package that can be installed even without Anaconda.

Anaconda bundles a whole bunch of Python packages that are commonly used by people using Python for scientific computing and/or data science. It provides a single download and an install program/script that install all the packages in one go. Alternate is to install Python and individually install all the required packages using. Additionally, it provides its own package manager (conda) and package repository. But it allows installation of packages from PyPI using pip if the package is not in Anaconda repositories. It is especially good if you are installing on Microsoft Windows as it can easily install packages that would otherwise require you to install C/C++ compilers and libraries if you were using pip. It is certainly an added advantage that conda, in addition to being a package manager, is also a virtual environment manager allowing you to install independent development environments and switch from one to the other (similar to virtualenv).

There is a minimal Anaconda Python without all the packages, called Miniconda. After installing miniconda, you can use conda and install only those scientific packages that you wish and avoid a bloated installation.

**Jupyter Notebook**

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

A Jupyter Notebook can be converted to a number of open standard output formats (HTML, presentation slides, LaTeX, PDF, ReStructuredText, Markdown, Python) through "Download As" in the web interface, via the nbconvert library or "jupyter nbconvert" command line interface in a shell.

To simplify visualisation of Jupyter notebook documents on the web, the nbconvert library is provided as a service through NbViewer which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user.

# CHAPTER-6

# Implementation and Testing

# Chapter 6

# Implementation and Testing

In this stage physical system specifications are converted into a working and reliable solution. This is where the system is developed. It is followed by testing and then implementation.

**Implementation Phases:**

1. Coding:

Includes implementation of the design specified in the design document into executable programming language code. The output of the coding phase is the source code for the software that acts as input to the testing and maintenance phase.

2. Integration and Testing:

Includes detection of errors in the software. The testing process starts with a test plan that recognizes test-related activities, such as test case generation, testing criteria, and resource allocation for testing. The code is tested and mapped against the design document created in the design phase. The output of the testing phase is a test report containing errors that occurred while testing the application.

3. Installation:

In this stage the new system is installed and rolled out.

**Key Deliverables:**

- Fully Installed system
- Fully trained users
- User and Operational Documentation

## 6.1 User Interface and Design

User interface is the most important part of any software application. A GUI allows the user to interact with the software application more easily and can carry out the work more efficiently.

The user interface and design of our software application i.e. “COLORME” mainly includes a homepage or main page on which there is a panel where an option to upload the image will be available.

By clicking the Choose File button near the tag Upload Image a dialog box will be opened from where you can select the image that you want to colorize.

Once the image is selected and uploaded on the software application click on the submit button to give instruction to the software to run the code for colorization on this image.

Once the computation gets completed the colorized image will be displayed on the another panel available on the screen. Now a Download button will be enabled by clicking on which you will be able to download the colorized image.

In this whole process you just need to upload and submit the image (i.e. Black and white image) and you'll get the output i.e. colorized image. There is manual work needed to colorize the image the whole process is automated and done by the machine itself.

The user interface and design of our software application is done using HTML, CSS, JavaScript on the frontend and FLASK at the backend. We Flask is a Python web micro framework. It is classified as a micro framework because it does not require particular tools or libraries. Flask supports extensions that can add application features as if they were implemented in Flask itself.

## 6.2 Implementation Approaches

There are a variety of options that a project manager could consider when implementing a solution. There are advantages and disadvantages to each type, and the choice usually depends on the client organizational setup and the complexity of the solution to be implemented.

For example, an international client with multiple offices needs to upgrade a certain e-mail system in all offices by a certain "go-live" date. In such a scenario, a project manager is faced with huge logistical and technical challenges, and the implementation strategy is pivotal in deciding on the rollout.

These implementation choices available to a project manager are:

- Parallel implementation
- Phased implementation
- Crash implementation

**Parallel Implementation**

A parallel implementation or approach implies that a new solution is implemented parallel to the current operating system in use. Those who are using the system will not see major downtime once it is implemented. The trick here is to implement the system.

Once the new solution is tested and up and running, it is "switched" on and the older version is "switched" off. The advantages with a parallel implementation include (1) less disruption to the business and (2) no loss of business if the new system suddenly fails.

**Phase Approach**

Sometimes trying to implement a solution all at once is not feasible because many clients have essential operations that run during normal working hours and cannot afford the luxury of having their entire operation close down for a lengthy period in time.

Often, clients have front office staff that attend to these operations (such as call centers, help desks, etc.), and they work in 24-hour shifts. This is why many clients approve of a phased implementation approach, and the project team must ensure that the phased implementation is possible. This approach involves implementing the solution to a certain amount of users and then rolling them onto the new solution, while the rest of the users are rolled out in a similar fashion until the entire solution is rolled out within the client environment.

The phase approach works well because (1) there is minimal disruption to the client's operation, and (2) problems are resolved quickly. The phased approach could also be used if there is more than one department. The project manager could decide that implementing the solution in one department at a time could be more reliable than trying to roll out all departments at the same time.

**Crash Implementation**

Careful planning needs to take place when considering a crash (also known as full-blown) implementation. It takes an incredible amount of planning and re-planning to ensure no problems arise. In fact, with this type of implementation, the necessary

contingencies need to be prepared and reviewed well in advance of the actual implementation, in order to minimize any potential failure.

The necessary IT support staff also needs to be available on the chosen implementation period. A full-blown implementation should be scheduled to take place over a slow period, such as a holiday or weekend.

## 6.3 Testing Approaches

### 6.3.1 Unit Testing

Unit testing is a level of software testing where individual units/ components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

**Unit Testing – Advantages:**

- Reduces Defects in the newly developed features or reduces bugs when changing the existing functionality.
- Reduces Cost of Testing as defects are captured in very early phase.
- Improves design and allows better refactoring of code.
- Unit Tests, when integrated with build gives the quality of the build as well.

**White Box Testing:**

| Test Case ID | Dataset Size (no. of photos) | No. of Epochs | Result |
|---|---|---|---|
| **ID 01** | 418 | 5-10 | Reddish Hue observed |
| **ID 02** | 1000 | 50-100 | Brownish Tint observed |
| **ID 03** | 1500 | 2000 | Real face colours started to be seen |
| **ID 04** | 1500 | 2500 | Nearly optimal colours appeared |

Table 5: White Box Testing

### 6.3.2 Integration Testing

Integration testing is the process of testing the interface between two software units or module. It's focus on determining the correctness of the interface. The purpose of the integration testing is to expose faults in the interaction between integrated units. Once all the modules have been unit tested, integration testing is performed.

## 6.4 Modification and Improvements

While testing the neural network on our personal machine we observed that to train a neural network over 1500 photos and 2500 epochs it takes 18 hrs to train. The neural network runs on a localhost server that makes it static with the computer machine. With using advanced technologies, we can put our training model on a dedicated server while using cloud GPUs and CPU runtimes which will eventually make our training model to learn more faster and efficiently. It will also observed, due to small storage spaces in our laptops we could only mount maximum of 1500 photos in one run. When we are using the servers, we could also use the unlimited cloud storage to store up to a million photos that would help our neural network to learn more efficiently.

# CHAPTER-7

# Results and Discussion

# Chapter 7

# Results and Discussion

The neural network uses a combination of InceptionResNet v2 as one of the classifiers and an encoder and a decoder. Inception-ResNet-v2, a convolutional neural network (CNN) that achieves a new state of the art in terms of accuracy on the ILSVRC image classification benchmark. Inception-ResNet-v2 is a variation of our earlier Inception V3 model which borrows some ideas from Microsoft's ResNet papers Residual connections allow shortcuts in the model and have allowed researchers to successfully train even deeper neural networks, which have led to even better performance. This has also enabled significant simplification of the Inception blocks**.** Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is François Chollet, a Google engineer. Chollet also is the author of the XCeption deep neural network model.

At present due to lack of computational power the script runs on Jupyter Notebook that provides a localhost to run the script and save's the trained model until the machine is turned off. The folder through which the python script is called, consists of three folders namely- Train, Test & Result. The train folder consists of the photos through which our network learns and tries to learn how the color pigments are placed and at where they are placed. The Test folder consists of the Greyscale photos on which the neural network will try to place colors according to the previously trained model. The converted photos will be saved in the result folder from where it could be retrieved later and can be seen.

# Conclusion & Future Works

The project will be able to bridge the gap between the former black and white world with the modern colorful world, so that it would be easier for the later generations to visualize what were the colors that were in trend in those days, what type of lifestyle they preferred and what kind of fashion did they carried.

The current version of the neural network tries to colorize the greyscale images in 5-10 seconds. It colorizes the greyscale images according to the colored images on which it is trained. It's a very simple, fast and efficient method to get a colorized image from a greyscale image

At present there is no application through which it could run due to the limited computational power of the machine. But in nearby days we can make our script to run on a dedicated server. This dedicated server will also have a lot of memory space to be able to store 10k to 50k photos on which our model could be trained. We will also provide a dedicated GPU for our script to run more efficiently and faster. This will make our photos to be colorized near to perfection.

Once the setup of the server is completed, we could access the neural network with the help of a mobile application. This mobile application will have the feature of uploading the greyscale image as well as a unique feature of scanning any black and white image on the go. It will directly upload the image to the server and will convert it and forward it to the mobile application from where it could be saved in your phone.