A PROJECT REPORT ON

Astronomical Image colorization and super-resolution using GANs

SUBMITTED TOWARDS THE
PARTIAL FULFILMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING (Computer Engineering)

BY

Shreyas Kalvankar	Exam No:
Hrushikesh Pandit	Exam No:
Pranav Parwate	Exam No:
Atharva Patil	Exam No:

Under The Guidance of

Prof. Dr. S.M. Kamalapur

Department of Computer Engineering
K. K. Wagh Institute of Engineering Education & Research
Hirabai Haridas Vidyanagari, Amrutdham, Panchavati,
Nashik-422003
Savitribai Phule Pune University
A. Y. 2020-21 Sem I

K. K. Wagh Institute of Engineering Education and Research
Department of Computer Engineering

CERTIFICATE

This is to certify that the Project Titled

Astronomical Image colorization and super-resolution using GANs

Submitted by

Shreyas Kalvankar	Exam No:
Hrushikesh Pandit	Exam No:
Pranav Parwate	Exam No:
Atharva Patil	Exam No:

is a bonafide work carried out by Students under the supervision of Prof. Dr. S.M. Kamalapur and it is submitted towards the partial fulfilment of the requirement of Bachelor of Engineering (Computer Engineering) Project during academic year 2020-21.

Prof. Dr. S.M. Kamalapur   Prof. Dr. S. S. Sane
Internal Guide   Head
Department of Computer Engineering   Department of Computer Engineering

Abstract

Automated colorization of black and white images has been subject to much research within the computer vision and machine learning communities. Beyond simply being fascinating from an aesthetic and artificial intelligence perspective, such capability has broad practical applications. It is an area of research that possesses great potentials in applications: from black and white photo reconstruction, image augmentation, video restoration to image enhancement for improved interpretability.

Image downscaling is an innately lossy process. The principal objective of super resolution imaging is to reconstruct a low resolution image into a high resolution one based on a set of low-resolution images to rectify the limitations that existed while the procurement of the original low-resolution images. This is to insure better visualization and recognition for either scientific or non-scientific purposes. No matter how good an upscaling algorithm is, there will always be some amount of high frequency data lost from a downscale-upscale function performed on the image. Ultimately, even the best upscaling algorithms cannot effectively reconstruct data that does not exist. Traditional methods for image upsampling rely on low-information, smooth interpolation between known pixels. Such methods can be treated as a convolution with a kernel encoding no information about the original image. A solution to the problem is by using Generative Adversarial Networks (GANs) to hallucinate high-frequency data in a super-resolved image that does not exist in the smaller image. Although they increase the resolution of an image, they fail to produce the clarity desired in the super-resolution task. By using the above mentioned method, not a perfect reconstruction can be obtained albeit instead a rather plausible guess can be made at what the lost data might be, constrained to reality by a loss function penalizing deviations from the ground truth image. A huge number of raw images lie unprocessed and unseen in the Hubble Legacy Archives. These raw images are typically low-resolution, black and white and unfit to be shared with the world. It takes huge amounts of hours to process them. This processing is necessary because astronomers often struggle to distinguish objects from the raw images. Random and synthetic noise from the sensors in the telescope, changing optical characteristics in

the system and noise from other bodies in the universe all make the processing further necessary. Furthermore, colorization is needed to help highlight small features that ordinarily wouldn't be able to be picked out against noise of the image. The processing of the images is so time consuming that the images are rarely seen by human eyes. The problem is only likely to get worse. Not only is new data being continuously produced by Hubble Telescope, but new telescopes are soon to come online. A simplification of image processing by using artificial image colorization and super-resolution can be done in an automated fashion to make it easier for astronomers to visually identify and analyze objects in Hubble dataset.

Acknowledgments

please enter text here.

Shreyas Kalvankar
Hrushikesh Pandit
Pranav Parwate
Atharva Patil
(B.E. Computer Engg.)

INDEX

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1   PROJECT IDEA

- The idea of the project is to create a efficient mathematical model for image colorization and super resolution using Generative Adversarial Networks (GANs)

- Having two networks compete will stimulate greater performance by the virtue of minimization of loss functions that traditional Convolutional Neural Network cant do

1.2   MOTIVATION OF THE PROJECT

- Image colorization seems to be evolving as computers get better and better at predicting missing variables. Different methods and techniques have been applied for colorizing images that have shown promising results

- Introduction of convolutional neural networks has made this task even more precise and accurate. Convoluting grayscale images to RGB provides unprecedented results with reference to visual inspection

- With the introduction of Generative Adversarial Networks, this particular task can be developed and modified to increase the efficiency and precision for colorizing images instead

- Furthermore, image upscaling is another problem that has been under research in the computer vision community. It has been studied and many approached have been developed to accurately predict the missing pixel values while upscaling an image

- Application of GANs to this discipline has successfully improved the performance and computers are getting better and better at predicting accurate missing pixel values and upscaling images many folds the original size

- All this computation power can be used for astronomical research by processing large data archives

- A large number of images lie dormant in most of the space survey data archives which never go through any kind of processing and are low resolution and black & white. These images could be processed automatically by an algorithm that will colorize and super-resolve the images which can make it easier for astronomers to visually inspect the images

1.3 LITERATURE SURVEY

1.3.1 Image Colorization

1.3.1.1 Hint Based Colorization

[2] proposed using colorization hints from the user in a quadratic cost function which imposed that neighboring pixels in space-time with similar intensities should have similar colours. This was a simple but effective method but only had hints which were provided in form of imprecise colored scribbles on the grayscale input image. But with no additional information about the image, the method was able to efficiently generate high quality colorizations. [3] addressed the color bleeding issue faced in this approach and solved it using adaptive edge detection. [4] used luminescence based weighting for hints to boost efficiency. [5] extended the original cost function to enforce color continuity over similar textures along with intensities.

[6] had proposed another approach that reduced the burden on the user by only requiring a full color example of an image with similar composition. It matched the texture and luminescence between the example and the target grayscale image and received realistic results as long as the example image was sufficiently similar.

Regardless of the scribble based or example based approach, the algorithms still needed sufficient human assistance in form of hand drawn or colorized images.

1.3.1.2 Deep Colorization

Owing to recent advances, the Convolutional Neural Networks are a de facto standard for solving image classification problems and their popularity continues to rise with continual improvements. CNNs are peculiar in their ability to learn and differentiate colors, patterns and shapes within an image and their ability to associate

them with different classes.

[7] proposed a per pixel training for neural networks using DAISY [8], and semantic [9] features to predict the chrominance value for each pixel, that used bilateral filtering to smooth out accidental image artifacts. With a large enough dataset, this method proved to be superior to the example based techniques even with a simple Euclidean loss function against the ground truth values.

Finally, [10] successfully implemented a system to automatically colorize black & white images using several ImageNet-trained layers from VGG-16 [11] and integrating them with auto-encoders that contained residual connections. These residual connections merged the outputs produced by the encoding VGG16 layers and the decoding portion of the network in the later stages. [12] showed that deeper neural networks can be trained by reformulating layers to learn residual function with reference to layer inputs. Using this *Residual Connections*, [12] created the *ResNets* that went as deep as 152 layers and won the 2015 ImageNet Challenge.

1.3.1.3 Generative Adversarial Networks

[13] introduced the adversarial framework that provides an approach to training a neural network which uses the generative distribution of $p_g(x)$ over the input data x.

Since it's inception in 2015, many extended works of GAN have been proposed over years including DCGAN [14], Conditional-GAN [15], iGAN [16], Pix2Pix [17].

[14] applied the adversarial framework for training convolutional neural networks as generative models for images, demonstrating the viability of *deep convolutional generative adversarial networks*.

DCGAN is the standard architecture to generate images from random noise. Instead of generating images from random noise, Conditional-GAN [15] uses a condition to generate output image. For e.g. a grayscale image is the condition for colorization of image. Pix2Pix [17] is a Conditional-GAN with images as the conditions. Besides learning the mapping from input image to output image, it can also learn a separate loss function to train this mapping. Pix2Pix is considered to be the

state of the art architecture for image-image translation problems like colorization.

1.3.2 Image Upscaling

1.3.2.1 Frequency-domain-based SR image approach

[18] proposed the frequency domain SR method, where SR computation was considered for the noise free low resolution images. They transformed the low resolution images into Discrete Fourier transform (DFT) and further combined it as per the relationship between the aliased DFT coefficient of the observed low resolution image and that of unknown high resolution image. Then the output is transformed back into the spatial domain where a higher resolution is now achieved.

While Frequency-domain-based SR extrapolates high frequeny information from the low resolution images and is thus useful, however they fall short in real world applications.

1.3.2.2 The interpolation based SR image approach

The interpolation-based SR approach constructs a high resolution image by casting all the low resolution images to the reference image and then combining all the information available from every image available. The method consists of the following three stages (i) the registration stage for aligning the low-resolution input images, (ii) the interpolation stage for producing a higher-resolution image, and (iii) the deblurring stage for enhancing the reconstructed high-resolution image produced in the step ii).

However, as each low resolution image adds a few new details before finally deblurring them, this method cannot be used if only a single reference image is available.

1.3.2.3 Regularization-based SR image approach

Most known Bayesian-based SR approaches are maximum likelihood (ML) estimation approach and maximum a posterior (MAP) estimation approach.

While [19] proposed the first ML estimation based SR approach with the aim to find the ML estimation of high resolution image, some proposed a MAP estimation

approach. MAP SR tries to takes into consideration the prior image model to reflect the expectation of the unknown high resolution image.

1.3.2.4 Super Resolution - Generative Adversarial Networks (SR-GAN)

The Genrative Adversarial Network [13], has two neural networks, the Generator and the Discriminator. These networks compete with each other in a zero-sum game. [20] introduced SRGAN in 2017, which used a SRResNet to upscale images with an upscaling factor of 4x. SRGAN is currently the state of the art on public benchmark datasets.

CHAPTER 2

PROBLEM DEFINITION AND SCOPE

2.1 PROBLEM STATEMENT

The problem can be divided into two sub-problems:

- Create an efficient model to colorize grayscale images

- Take a colorized image and upscale it *n* times the original size

2.1.1 Goals and objectives

Goal and Objectives:

- Auto-Colorization:

  - The first model will be given input a grayscale, low resolution image of dimensions $(64 \times 64 \times 1)$

  - The model will perform a series of mathematical operations that will increase the channel width of the image from 1 (single channel grayscale image) to 3 (RGB)

  - The output of the model will be a colorized version of the input image with dimensions $(64 \times 64 \times 3)$

- Upscaling/super-resolution:

  - The input to the model will be a colorized image of shape $(64 \times 64 \times 3)$

  - The model will increase the dimensions of the image from (64×64) to $((64 \cdot n) \times (64 \cdot n))$ by performing a series of upscaling operations and predicting information that may be lost while downscaling

  - The output of the model will be an upscaled RGB image with dimensions $((64 \cdot n) \times (64 \cdot n) \times 3)$

- The models may be combined to form a single model that will take a low resolution, grayscale image as its input and produce a high resolution, colorized image as its output

2.1.2 Statement of scope

- The model will consist of neural networks implemented using deep learning frameworks that will accept images of input format *JPEG*

- The input will be grayscale images of size 64×64

2.1.3 Statement of scope

- The model will consist of neural networks implemented using deep learning frameworks that will accept images of input format *JPEG*

- The input will be grayscale images of size 64×64

- Input bounds:

  * Lower bound: $64 \times 64 \times 1$

  * Upper bound: no limit

- The output will be produced in two phases:

  * A colorized output of model 1 with shape $64 \times 64 \times 3$

  * A upscaled output of model 2 from the colorized output of model 1 with shape $(64 \cdot n) \times (64 \cdot n) \times 3$

- The model will:

  * take input black & white images

  * produce colorized images of the same size

  * produce upscaled images of size *n* times the input size (currently 64)

- The model will **not**:

  * take a colorized image as an input

  * take an image of size less than (64×64) in size

  * produce accurate upscaling or coloring albeit merely make a guess at what the lost values might be

2.2 MAJOR CONSTRAINTS

- The astronomical image data required for training purposes is mostly raw. There exists no structured dataset that is already cleaned. The unavailability of a dataset is a major constraint for the project

- Scraped data from the archives is noisy and requires heavy processing and cleaning in order to be usable by the model

- The images available for download are of low resolution, which sets an upper bound on the maximal upscale factor

- The image data is large and needs high computation power to process

- The data needs to be cleaned manually as there exist no methods to automatically do this particular task

- The model involves neural networks which heavily rely on computation power for its training. The hardware required for training is not readily available because of absence of a workstation supporting heavy computations

- The training part requires large amount of memory

- Absence of an NVIDIA workstation GPU will slow down the training further

2.3 METHODOLOGIES OF PROBLEM SOLVING AND EFFICIENCY ISSUES

- Data gathering and processing

  - Data Scraping

    * Owing to unavailability of a dataset, raw data can be acquired by the means of web scraping
    * Images from the snapshots of entire night sky can be obtained in such a way from the Hubble Legacy Archive

  - Data Cleaning

* The scraped data consists of snapshots of the entire night sky with 1 degree deviation of the telescope

* This results in large amount of noisy, overexposed, irregular data images

* This data needs to be cleaned manually before it can be used for any kind of study

- Image colorization

  - The problem of image colorization has been solved using multiple methodologies

  - [10] used convolutional neural networks with residual encoders using the VGG16 architecture

  - Though the system performs extremely well in realistically colorizing various images, it consisted of $L2$ loss which was a function of the Euclidean distance between the pixel's blurred color channel value in the target and predicted image

$$L2loss = \sum_{i=1}^{n} (y_{true} - y_{predicted})^2 \qquad (2.1)$$

  - This is a regression based approach and the pixel-wise L2 loss will impose an averaging effect over all possible candidates and will result in dimmer and patchy colorization

  - Generative Adversarial Networks introduced by [13] use a minimax loss which is different than the $L2$ loss as it will choose a color to fill an area rather than averaging. This is similar to a classification based approach

- Image Upscaling

  - One of the most popular approach to image upscaling was sparse-coding. This approach assumes that images can be sparsely represented by a dictionary of atoms in some transformed domain [21]. The dictionary is learned during the training process.

- The main drawback for this was that the optimization algorithm was computationally expensive

- Dong et. al explored super-resolution using convolutional neural network and calling it SRCNN [22]. They explained how CNN had many similarities to the sparse-coding-based super-resolution.

- Kim et. al improved upon SRCNN's results using their very own model inspired from the VGG-net architecture[23].

- After the introduction of GANs, Ledig et. al applied them to super-resolution (SRGAN) using a network inspired by the ResNets [20][12].

- SR-GAN works well with for single image super-resolution as it also uses an intelligent content loss function that uses pre-trained VGG-net layers. However, Ledig et. al noted that further information could be used if this network were to be adapted to a video, such as temporal information.

- A generative network, G, is meant to learn the underlying distribution of a data set, Y. For e.g. we can train a GAN over face images to generate images similar to those faces. With just a generative network however, we must visually assess the quality of network outputs and judge how we can adapt the network to produce more convincing results.

- With a discriminative network D, we can incorporate this tweaking directly into training. The discriminative network takes in both fabricated inputs generated by G and the real inputs from Y. It's sole purpose is to classify if the input has come from G or Y.

- The key idea is back propagation of the gradients from the results of D's classification to G so that G gets better at producing images and in turn fooling D.

- For the project, we split the data into three categories: X that serves as the data for the Y, which are its corresponding labels.

- G_1 takes in a low resolution $x \in X$ which is black & white and produces $\hat{y}$, a colorized version of x. The descriminator D, in turn takes in a colorized image and outputs the probability that the image comes from Y, instead of as outputs from G, $G(x)$. As such, if the discriminator is fooled by out generator, it should output a probability greater than 0.5 for the set of inputs coming from $G(x)$ and a probability less than 0.5 for images coming from Y.

- The same is the process for generator G_2 with the only difference being that the X is the set of colorized images but having low resolution and Y is the set of high resolution images that serve as the labels for underlying mapping of X. G_2 takes in the low resolution image $x \in X$ and produces $\hat{y}$ and the discriminator outputs a probability determining whether the image is super-resolved by G_2 or the ground truth images from Y.

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

Explain the scenario in which multi-core, embedded and distributed computing methodology can be applied.

2.5 OUTCOME

- Outcome of the project

2.6 APPLICATIONS

- Applications of Project

2.7 HARDWARE RESOURCES REQUIRED

2.8 SOFTWARE RESOURCES REQUIRED

Platform :

1. Operating System:

Sr. No.	Parameter	Minimum Requirement	Justification
1	CPU Speed	2 GHz	Remark Required
2	RAM	3 GB	Remark Required

Table 2.1: Hardware Requirements

2. IDE:

3. Programming Language

CHAPTER 3

PROJECT PLAN

3.1 PROJECT ESTIMATES

3.1.1 Reconciled Estimates

3.1.1.1 Cost Estimate

The model followed is the Constructive Cost Model (COCOMO) for estimating the efforts required in the completion of the porject. Like all estimation models, the COCOMO model requires sizing information. This information can be specified in the form of:

- Object Point

- Function Point(FP)

- Lines of Source Code(KLOC)

For our project, sizing information in the form of Lines of Source Code is used. The total lines of code,

KLOC = 750

Equations: The initial effort(Ei) in man-months is calculated using equations:

$$E = ax(KLOC)^b$$

where, a = 3.0, b = 1.12, for a semi-detached project E = Efforts in person-hours E = 4.5 PM

$$D = ax(E)^b$$

Where, a = 2.5,

b = 0.35, for a semi-detached project

D = Duration of Project in months

D = 4 Months

3.1.1.2   Time Estimates

$$C = D * Cp * hrs$$

Where, C = Cost of project

D = Duration in Hours

Cp = Cost incurred per person-hour

hrs = hours

Total of 4.5 person-months are required to complete the project successfully.

Duration of Project D = 6 Months

The approximate duration of the project is 4 months

3.1.2   Project Resources

- Google Collab

- IEEE Access Provided by Institute

3.2   RISK MANAGEMENT

This section discusses Project risks and the approach to managing them.

3.2.1   Risk Identification

1. Dataset needs to be processed in order to get clean data

2. Vanishing Gradients

3. Mode Collapse

4. Failure to Converge

3.2.2   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	Garbage images in dataset	Medium	Low	High	High
2	Vanishing Gradients	Low	Low	High	High

Table 3.1: 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 3.2: 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 3.3: Risk Impact definitions [1]

3.2.3 Overview of Risk Mitigation, Monitoring, Management

Following are the details for each risk

Risk ID	1
Risk Description	There are Garbage images in dataset which can lead to improper training of discriminator. It can lead to undesired outputs
Category	Pre processing.
Source	Software requirement Specification document.
Probability	Low
Impact	High
Response	Mitigate
Strategy	To manually remove garbage images from the dataset
Risk Status	Occurred

Risk ID	2
Risk Description	If discriminator is too good , then generator training can fail due to Vanishing gradients
Category	Requirements
Source	Software Design Specification documentation review.
Probability	Low
Impact	High
Response	Mitigate
Strategy	Wasserstein loos and Modifed minimax loss are desigend to prevent vanishing gradients.
Risk Status	Identified

3.3 PROJECT SCHEDULE

3.3.1 Project task set

Major Tasks in the Project stages are:

- Task 1: Colledting Dataset

- Task 2: Cleaning Dataset

- Task 3: Creating GAN model of image colorization and super-resolution

- Task 4: Training the GAN model

- Task 5: Fine tuning the GAN model

3.4   TEAM ORGANIZATION

- Team of 4 members

- 1 Project guide

- 1 Project coordinator

3.4.1   Team structure

Team of 4 members

3.4.2   Management reporting and communication

- Weekly meeting with guide about the work

- Weekly meeting among team for updates and work distribution

- Daily updates on work progress

CHAPTER 4

SOFTWARE REQUIREMENT SPECIFICATION

4.1 INTRODUCTION

4.1.1 Purpose and Scope of Document

The purpose of SRS and what it covers is to be stated

4.1.2 Overview of responsibilities of Developer

What all activities carried out by developer?

4.2 USAGE SCENARIO

This section provides various usage scenarios for the system to be developed.

4.2.1 User profiles

The profiles of all user categories are described here.(Actors and their Description)

4.2.2 Use-cases

All use-cases for the software are presented. Description of all main Use cases using use case template is to be provided.

Sr No.	Use Case	Description	Actors	Assumptions
1	Use Case 1	Description	Actors	Assumption

Table 4.1: Use Cases

4.2.3 Use Case View

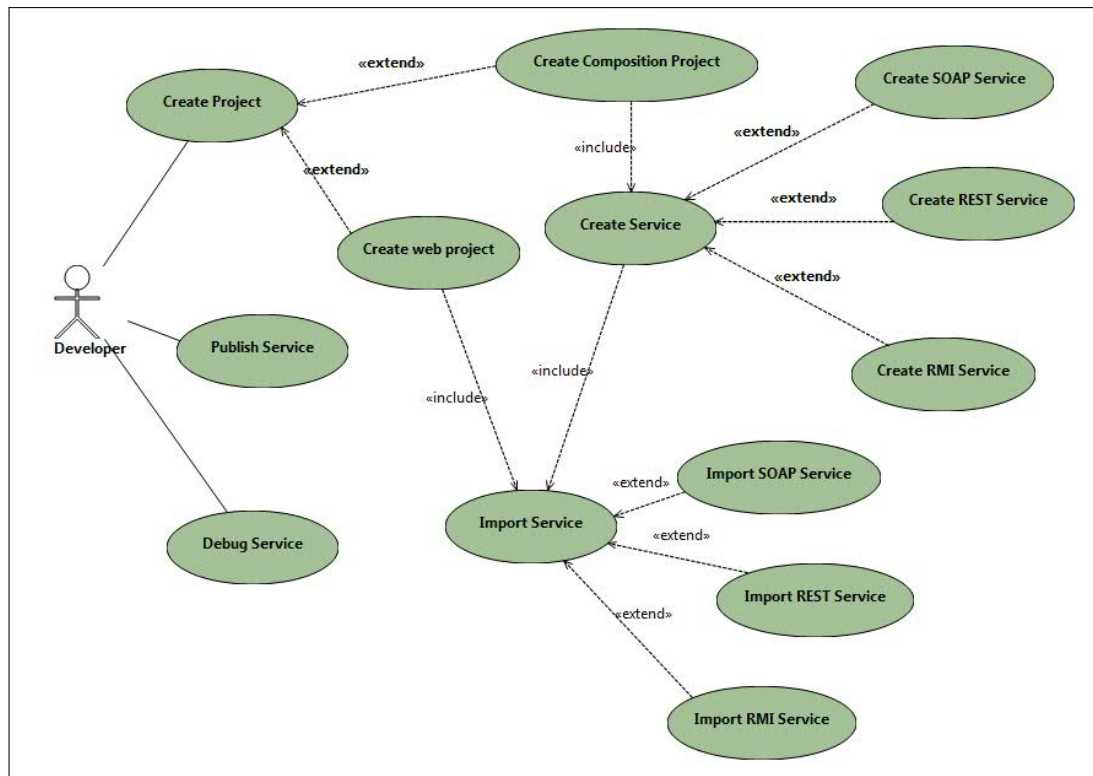Use Case Diagram. Example is given below

Figure 4.1: Use case diagram

4.3 DATA MODEL AND DESCRIPTION

4.3.1 Data Description

Data objects that will be managed/manipulated by the software are described in this section. The database entities or files or data structures required to be described. For data objects details can be given as below

4.3.2 Data objects and Relationships

Data objects and their major attributes and relationships among data objects are described using an ERD- like form.

4.4 FUNCTIONAL MODEL AND DESCRIPTION

A description of each major software function, along with data flow (structured analysis) or class hierarchy (Analysis Class diagram with class description for object oriented system) is presented.

4.4.1 Data Flow Diagram

4.4.1.1 Level 0 Data Flow Diagram

4.4.1.2 Level 1 Data Flow Diagram

4.4.2 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 4.2
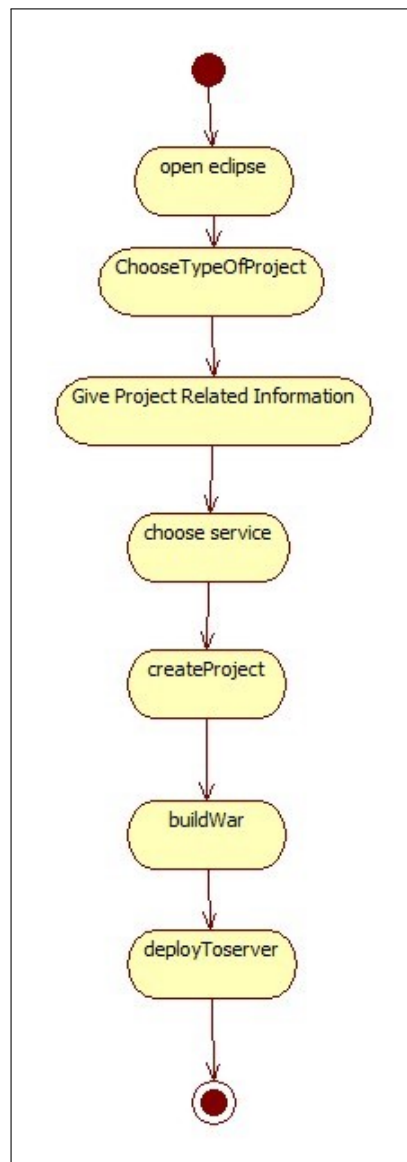
Figure 4.2: Activity diagram

4.4.3 Activity Diagram:

- The Activity diagram represents the steps taken.

4.4.4 Non Functional Requirements:

- Interface Requirements

- Performance Requirements

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

4.4.5 State Diagram:

State Transition Diagram

Fig.4.3 example shows the state transition diagram of Cloud SDK. The states are
represented in ovals and state of system gets changed when certain events occur.
The transitions from one state to the other are represented by arrows. The Figure
shows important states and events that occur while creating new project.

4.4.6 Design Constraints

Any design constraints that will impact the subsystem are noted.

4.4.7 Software Interface Description

The software interface(s)to the outside world is(are) described. The requirements for
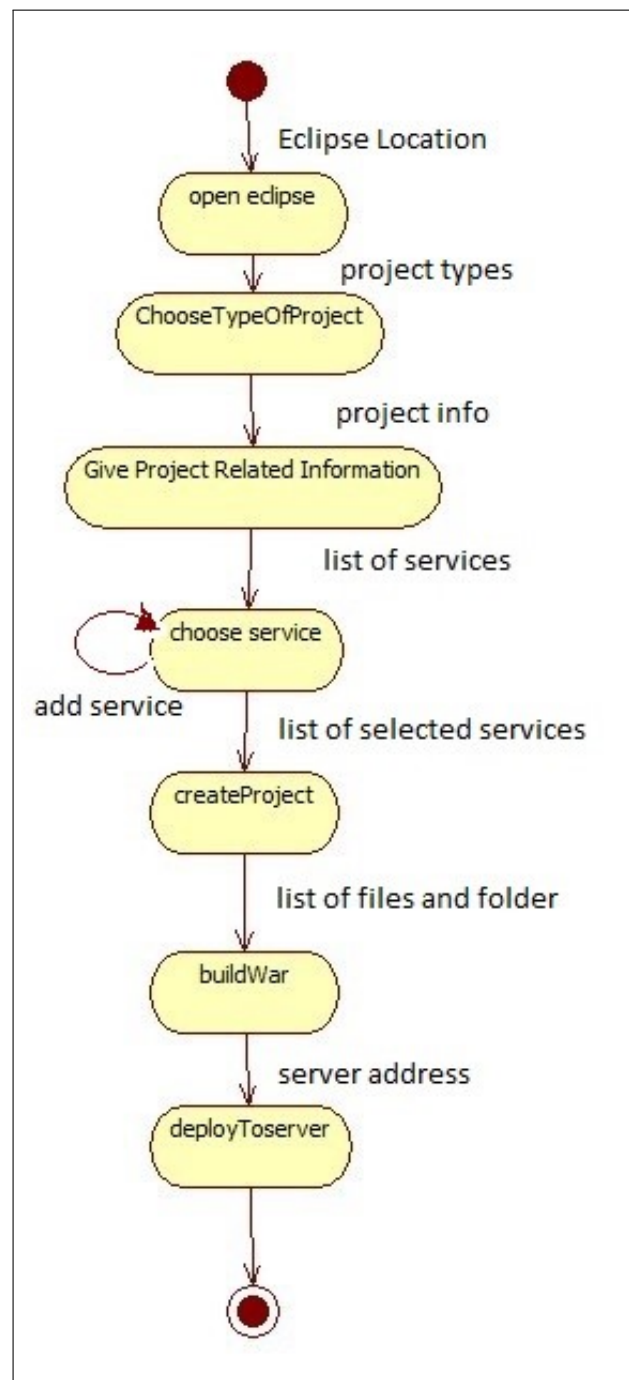interfaces to other devices/systems/networks/human are stated.

Figure 4.3: State transition diagram

CHAPTER 5

DETAILED DESIGN DOCUMENT

5.1 INTRODUCTION

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

5.2 ARCHITECTURAL DESIGN

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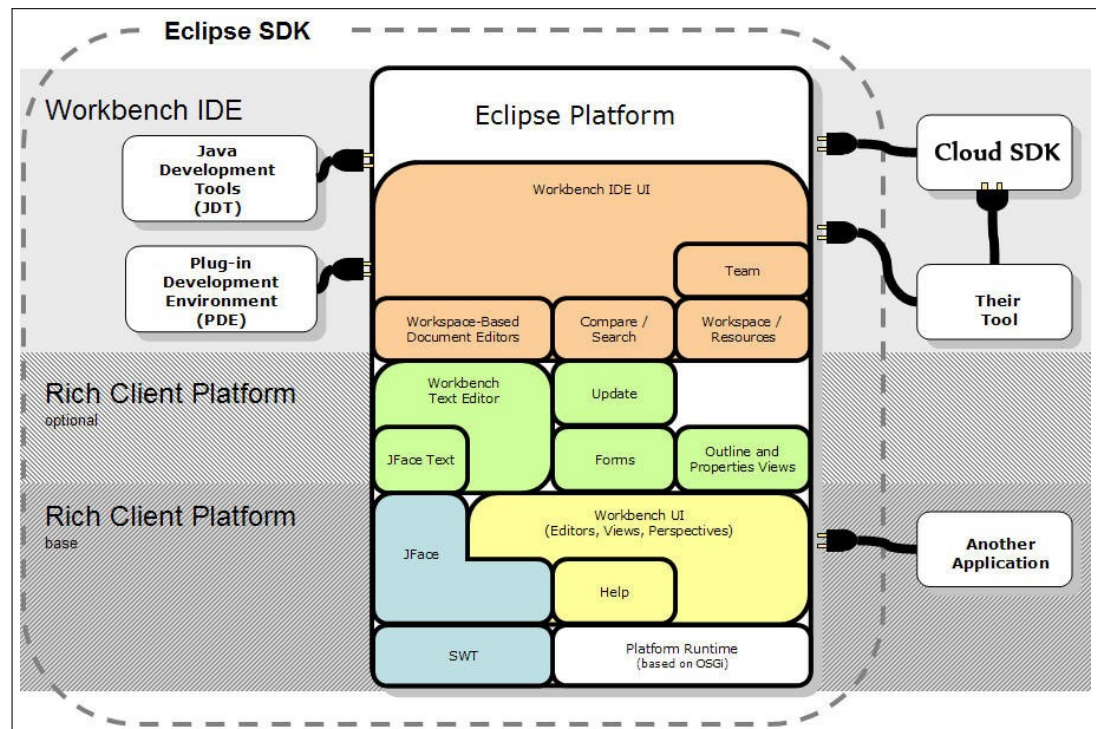

Figure 5.1: Architecture diagram

5.3 DATA DESIGN

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

5.3.1 Internal software data structure

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

5.3.2 Global data structure

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

5.3.3 Temporary data structure

Files created for interim use are described.

5.3.4 Database description

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

CHAPTER 6

DATASET AND EXPERIMENTAL SETUP

CHAPTER 7

SUMMARY AND CONCLUSION

Write one page summary and conclusion

32

REFERENCES

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

[2] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM SIGGRAPH 2004 Papers*, pp. 689–694, 2004.

[3] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu, "An adaptive edge detection based colorization algorithm and its applications," pp. 351–354, 01 2005.

[4] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1120–1129, 2006.

[5] Y. Qu, T.-T. Wong, and P.-A. Heng, "Manga colorization," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 1214–1220, 2006.

[6] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," *ACM Trans. Graph.*, vol. 21, pp. 277–280, 07 2002.

[7] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization," 2016.

[8] E. Tola, V. Lepetit, and P. Fua, "A fast local descriptor for dense matching," *Proc. CVPR*, 06 2008.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2015.

[10] R. Dahl, "Automatic colorization," 2016.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[14] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2016.

[15] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014.

[16] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," 2018.

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2018.

[18] R. TSAI, "Multiframe image restoration and registration," *Advance Computer Visual and Image Processing*, vol. 1, pp. 317–339, 1984.

[19] Tom and Katsaggelos, "Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images," in *IEEE International Conference on Image Processing* (Anon, ed.), vol. 2, pp. 539–542, IEEE, jan 1996. Proceedings of the 1995 IEEE International Conference on Image Processing. Part 3 (of 3) ; Conference date: 23-10-1995 Through 26-10-1995.

[20] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," 2017.

[21] Jianchao Yang, J. Wright, T. Huang, and Yi Ma, "Image super-resolution as sparse representation of raw image patches," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[22] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 184–199, Springer International Publishing, 2014.

[23] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1646–1654, 2016.

ANNEXURE A

MATHEMATICAL MODEL

ANNEXURE B

PLAGIARISM REPORT

ANNEXURE C

PAPER PUBLISHED (IF ANY)

ANNEXURE D

SPONSORSHIP DETAIL (IF ANY)