

# **“ARTIFY : THE ART GENERATOR”**

A

***Project Report***

*submitted*

*in partial fulfillment*

*for the award of the Degree of*

***Bachelor of Technology***

***in Department of Computer Science and Engineering***



**Project Mentor:**

Name: Neha Mathur

Designation : Assistant Prof.

**Submitted By :**

(Khushi Mittal, 17ESKCS080)

(Lipi Singh Parihar, 17ESKCS085)

(Ishan Goyal, 17ESKCS069)

**Department of Computer Science and Engineering  
Swami Keshvanand Institute of Technology, M & G, Jaipur  
Rajasthan Technical University, Kota  
Session 2019-2020**

---

**Swami Keshvanand Institute of Technology,  
Management & Gramothan, Jaipur**  
Department of Computer Science and Engineering

**CERTIFICATE**

This is to certify that Ms. KHUSHI MITTAL, Ms. LIPI SINGH PARIHAR, Mr.  
ISHAN GOYAL, student of B.Tech(Computer Science & Engineering) VII  
semester has submitted his/her Project Report entitled "ARTIFY: The Art  
Generator" under my guidance.

**Mentor**

Name - Mrs. Neha Mathur

Designation - Assistant Prof.

Signature.....

**Coordinator**

Name - Mrs. Anjana Sangwan

Designation - Assistant Prof.

Signature.....

# DECLARATION

We hereby declare that the report of the project entitled "Artify: The Art Generator" is a record of an original work done by us at Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur under the mentorship of Mrs. Neha Mathur, Assistant Professor (Dept. of Computer Science and Engineering) and coordination of Mrs. Anjana Sangwan, Assistant Professor (Dept. of Computer Science and Engineering). This project report has been submitted as the proof of original work for the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology (B.Tech) in the Department of Computer Science and Technology. It has not been submitted anywhere else, under any other program to the best of our knowledge and belief.

## Team Members

(Khushi Mittal, 17ESKCS080)

(Lipi Singh Parihar, 17ESKCS085)

(Ishan Goyal, 17ESKCS069)

## Signature



# Acknowledgement

A project of such a vast coverage cannot be realized without help from numerous sources and people in the organization. We take this opportunity to express our gratitude to all those who have been helping us in making this project successful.

We are highly indebted to our faculty mentor Mrs. Neha Mathur, Assistant Professor, Department of Computer Science and Engineering. She has been a guide, motivator source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We also thank our project coordinator Mrs. Anjana sangwan,, Assistant Professor, Department of Computer Science and Engineering, for her co-operation, encouragement, valuable suggestions and critical remarks that galvanized our efforts in the right direction.

We would also like to convey our sincere thanks to Prof. Dr. Mukesh Gupta, HOD, Department of Computer Science and Engineering, for facilitating, motivating and supporting us during each phase of development of the project. Also, we pay our sincere gratitude to all the Faculty Members of Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur and all our Colleagues for their co-operation and support.

Last but not least we would like to thank all those who have directly or indirectly helped and cooperated in accomplishing this project.

## **Team Members:**

(Khushi Mittal, 17ESKCS080)

(Lipi Singh Parihar, 17ESKCS085)

(Ishan Goyal, 17ESKCS069)

# Contents

<b>1</b>	<b>Project Chapter</b>	<b>2</b>
1.1	Problem Statement and Objective . . . . .	2
1.2	Literature Survey /Market Survey/Investigation and Analysis . . . . .	2
1.3	Introduction to Project . . . . .	3
1.4	Proposed Logic / Algorithm / Business Plan / Solution / Device . . . . .	3
1.5	Scope of the Project . . . . .	4
<b>2</b>	<b>Software Requirement Specification</b>	<b>6</b>
2.1	Overall Description . . . . .	6
2.1.1	Product Perspective . . . . .	6
2.1.1.1	System Interfaces . . . . .	8
2.1.1.2	User Interfaces . . . . .	8
2.1.1.3	Hardware Interfaces . . . . .	8
2.1.1.4	Software Interfaces . . . . .	8
2.1.1.5	Communications Interfaces . . . . .	9
2.1.1.6	Memory Constraints . . . . .	9
2.1.1.7	Operations . . . . .	9
2.1.1.8	Project Functions . . . . .	9

2.1.1.9	User Characteristics . . . . .	10
2.1.1.10	Constraints . . . . .	10
2.1.1.11	Assumption and Dependencies . . .	10
<b>3</b>	<b>SYSTEM DESIGN SPECIFICATION</b>	<b>12</b>
3.1	Module Decomposition Description . . . . .	12
3.1.1	Upload Image . . . . .	12
3.1.2	Select/Upload Style Image . . . . .	12
3.1.3	Generate Styled Image . . . . .	12
3.1.4	Download Image . . . . .	13
3.2	High Level Design Diagrams . . . . .	14
3.2.1	Use Case Diagram . . . . .	14
3.2.2	Activity Diagram . . . . .	16
3.2.3	Data-Flow Diagram . . . . .	17
3.2.4	Sequence Diagram . . . . .	18
<b>4</b>	<b>METHODOLOGY AND TEAM</b>	<b>20</b>
4.1	Introduction to Waterfall Framework . . . . .	20
4.2	Team Members, Roles & Responsibilities . . . . .	23
<b>5</b>	<b>System Testing</b>	<b>24</b>
5.1	Functionality Testing . . . . .	24
5.2	Performance Testing . . . . .	25
5.3	Usability Testing . . . . .	25
<b>6</b>	<b>TEST EXECUTION SUMMARY</b>	<b>26</b>
<b>7</b>	<b>PROJECT SCREENSHOTS</b>	<b>27</b>

<b>8</b>	<b>PROJECT SUMMARY AND CONCLUSIONS</b>	<b>29</b>
8.1	Conclusion . . . . .	29
<b>9</b>	<b>FUTURE SCOPE</b>	<b>30</b>
	<b>References</b>	<b>30</b>

## List of Figures

3.1	Use Case Diagram . . . . .	14
3.2	Activity Diagram . . . . .	16
3.3	Data Flow Diagram . . . . .	17
3.4	Sequence Diagram . . . . .	18
4.1	WaterFall model . . . . .	21



---

# List of Tables

6.1	Table of Results . . . . .	26
-----	----------------------------	----

---

# Chapter 1

## Project Chapter

### 1.1 Problem Statement and Objective

Neural style transfer is an optimization technique used to take two images—a content image and a style reference image (such as an artwork by a famous painter)—and blend them together so the output image looks like the content image, but “painted” in the style of the style reference image. The project uses this concept to generate images of high perceptual quality.[1]

The objective is to minimize the total loss such that the stylized image simultaneously matches the content representation of the content image and the style representation of the style image.

### 1.2 Literature Survey /Market Survey/Investigation and Analysis

The paper by Gatys, Leon A.; Ecker, Alexander S.; Bethge, Matthias (26 August 2015) named “A Neural Algorithm of Artistic Style” was first to introduce the concept of Neural Style Transfer. The project is

---

inspired by this concept from the paper.[2]

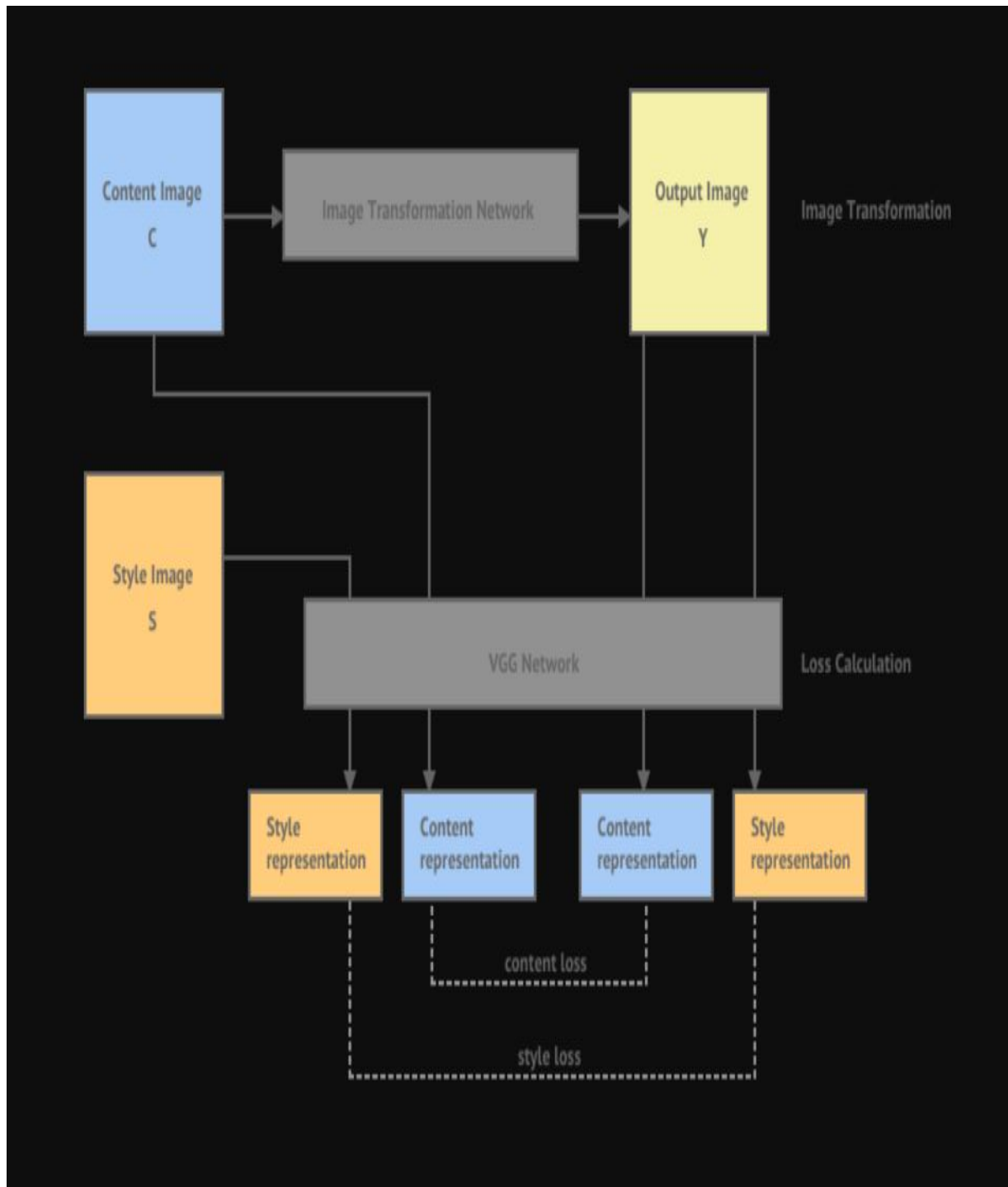
The generated image needs to be iterated a certain number of times to get better results. Generally higher iterations produce better results but these also have high computational cost. So we need to optimise the results to minimise the total loss and to also keep check on the computational cost.

### **1.3 Introduction to Project**

Artify is an art generator that combines the content of one image with the artistic style of another using Neural Style Transfer. It generates unique images which looks like the content image, but are “painted” in the style of the style reference image. This empowers people all around the world to experiment with their own creativity and generate artistic images.

### **1.4 Proposed Logic / Algorithm / Business Plan / Solution / Device**

The project uses deep learning algorithm like Neural Style Transfer to generate the desired image. The basic logic behind it is how to minimise the total loss which comprises the style loss and the content loss.



## 1.5 Scope of the Project

If we're being honest with ourselves, not everyone is born an artist. But with recent advances in technologies like style transfer, almost anyone can enjoy the pleasure that comes along with creating and shar-

---

ing an artistic masterpiece. This is where the transformative power of style transfer lies. Artists can easily lend their creative aesthetic to others, allowing new and innovative representations of artistic styles to live alongside original masterpiece.[3]

---

## **Chapter 2**

# **Software Requirement Specification**

### **2.1 Overall Description**

#### **2.1.1 Product Perspective**

Artify- The Art Generator is a Web Application that uses Deep Learning to generate Artistic images. This project is entirely based on Neural Style Transfer and Convolutional Neural Networks.

#### **Neural Style Transfer:**

This Deep Learning technique helps you generate an output image (denoted as G) based on the content image (denoted as C) but drawn or painted in the style of another image (called the style image, denoted as S).



## Convolutional Neural Networks and their learning:

When an image is passed onto the network, and the training is started, each layer would have some activation. Now, if you try to visualize these activations of each layer, you'll find that the initial layers would try to recognize the simple features of the image like edges or borders or shade of a particular color. Whereas the activations of the deeper layers would tend to learn more complex features of the images like shapes, patterns, objects, textures, etc. Approaching the end, the final layers would be able to identify foreground or background objects like cats, dogs, cars, etc.

This is the basic intuition we use in Neural Style Transfer i.e. to separate the style and content of images. We extract out all the features from the style image, separate it, and similarly, we also separate out the complex features from the content image, this helps us in the generation of the output image.[4]

---

#### **2.1.1.1 System Interfaces**

The project can be accessed through any system like Windows, Linux, etc which can fulfil the software requirements.

#### **2.1.1.2 User Interfaces**

The user interface includes the following-

- 1) User friendly Web Application of the system.
- 2) Uploading content image.
- 3) Selecting/Uploading style image.
- 4) Generating resultant image.
- 5) Download the generated image.

#### **2.1.1.3 Hardware Interfaces**

The application is designed in such a way that it is accessible on both Mobile and Desktop. A stable internet connection is required for flawless operation of the application.

#### **2.1.1.4 Software Interfaces**

Artify- The Art generator is a web application which can operate on any browser like Mozilla Firefox and Google Chrome. The software used in the project include Anaconda, Jupyter Notebooks and Sublime Text 3. The project is in python 3. The Deep Learning libraries used are TensorFlow and Keras.



---

#### **2.1.1.5 Communications Interfaces**

- Client (customer) on Internet will be using HTTP/HTTPS protocol.
- System user on Internet will be using HTTP/HTTPS protocol.
- A stable Internet connection is required to perform any task/operation.

#### **2.1.1.6 Memory Constraints**

There should be enough space to store the images and to load the pretrained model and to perform the desired transformations on the images.

#### **2.1.1.7 Operations**

Artify- The Art generator is a web application which can operate on any browser like Mozilla Firefox and Google Chrome. The application is designed in such a way that it is accessible on both Mobile and Desktop.

#### **2.1.1.8 Project Functions**

- 1) Upload image
- 2) Select/Upload style image
- 3) Generate styled image
- 4) Download image

---

#### **2.1.1.9 User Characteristics**

- 1) Artists – An artist can use this tool to bring his/her imagination to reality. He can also use the generated image as a reference image for his/her artworks.
- 2) Photo Editors – Editors can convert an image in an artistic form to reflect creativity.
- 3) Others – Anyone who wishes to add creativity to an image can use this application.

#### **2.1.1.10 Constraints**

- 1) Heavy processing power for running neural networks, thus it might take more time to generate an image.
- 2) Stable and fast internet connection, as a slow connection can cause a delay in the response.
- 3) Limited to HTTP/HTTPS.

#### **2.1.1.11 Assumption and Dependencies**

Assumptions-

- 1) The image being uploaded should be in the specified format i.e. png, jpg or jpeg.
- 2) The uploaded images should be clear and should convey the information properly for better results.

---

### Dependencies-

- 1) Stable and Fast Internet Connection.
- 2) Web Browsers like Mozilla Firefox, Google Chrome, Microsoft Edge etc.

---

## **Chapter 3**

# **SYSTEM DESIGN SPECIFICATION**

### **3.1 Module Decomposition Description**

#### **3.1.1 Upload Image**

The system takes an image as input. The image is selected by the user for which he wants to make the changes. The system will take the image, then it will be processed for further transformations.

#### **3.1.2 Select/Upload Style Image**

The user can upload the style image of style that he wants the content image to be painted in.

#### **3.1.3 Generate Styled Image**

The system will ask the user whether he wants to select the style image which is already present in the alternatives or he wants to use the style image of his own choice. If he wants to use his own style image, then he will need to upload the style image.

---

#### **3.1.4 Download Image**

After generation of the styled image, the system will give an option to download the generated image. The user can easily download the image.

## 3.2 High Level Design Diagrams

### 3.2.1 Use Case Diagram

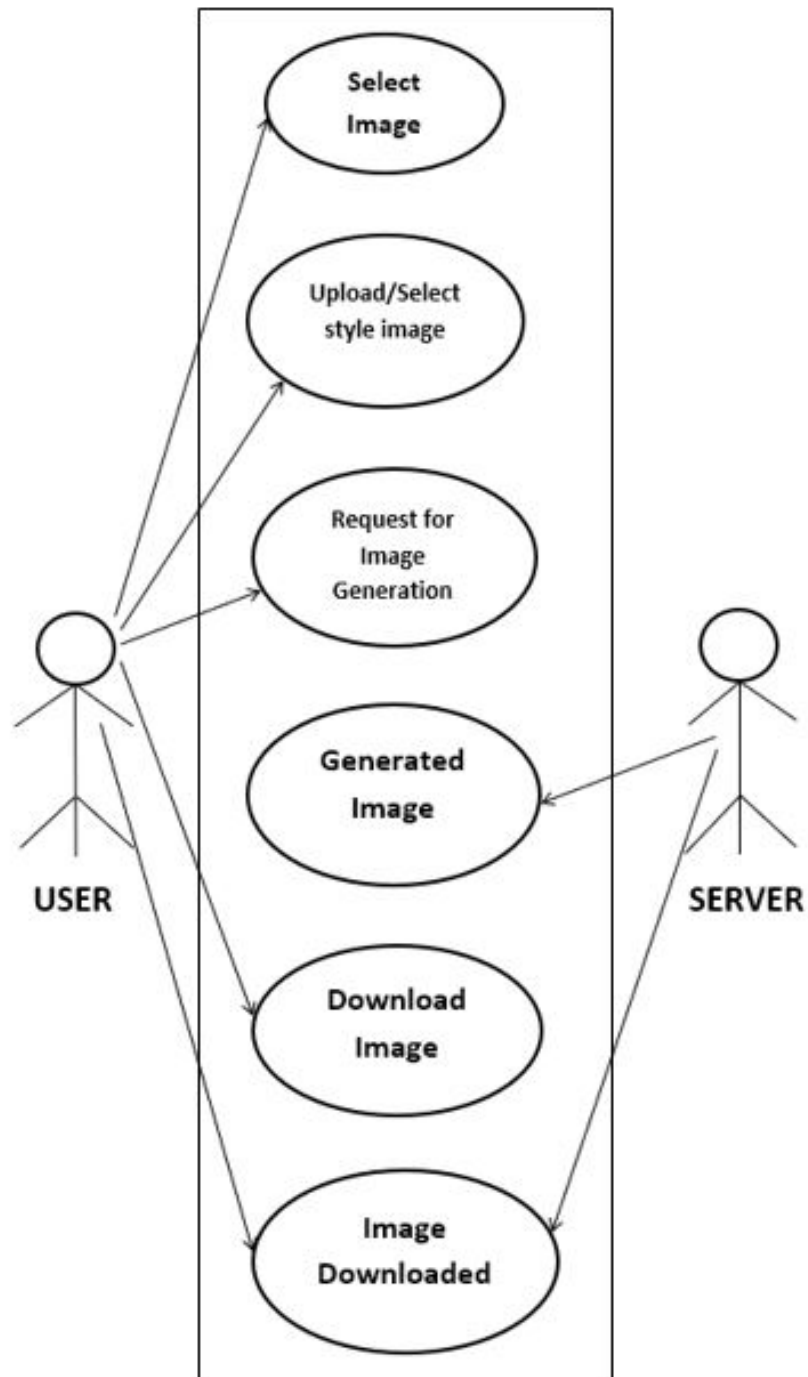


Figure 3.1: Use Case Diagram

---

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 the user is involved.

In this system, the user is responsible for uploading content image, selecting/uploading style image, requesting Image generation and downloading image. The server is responsible for providing the user with generated image.

### 3.2.2 Activity Diagram

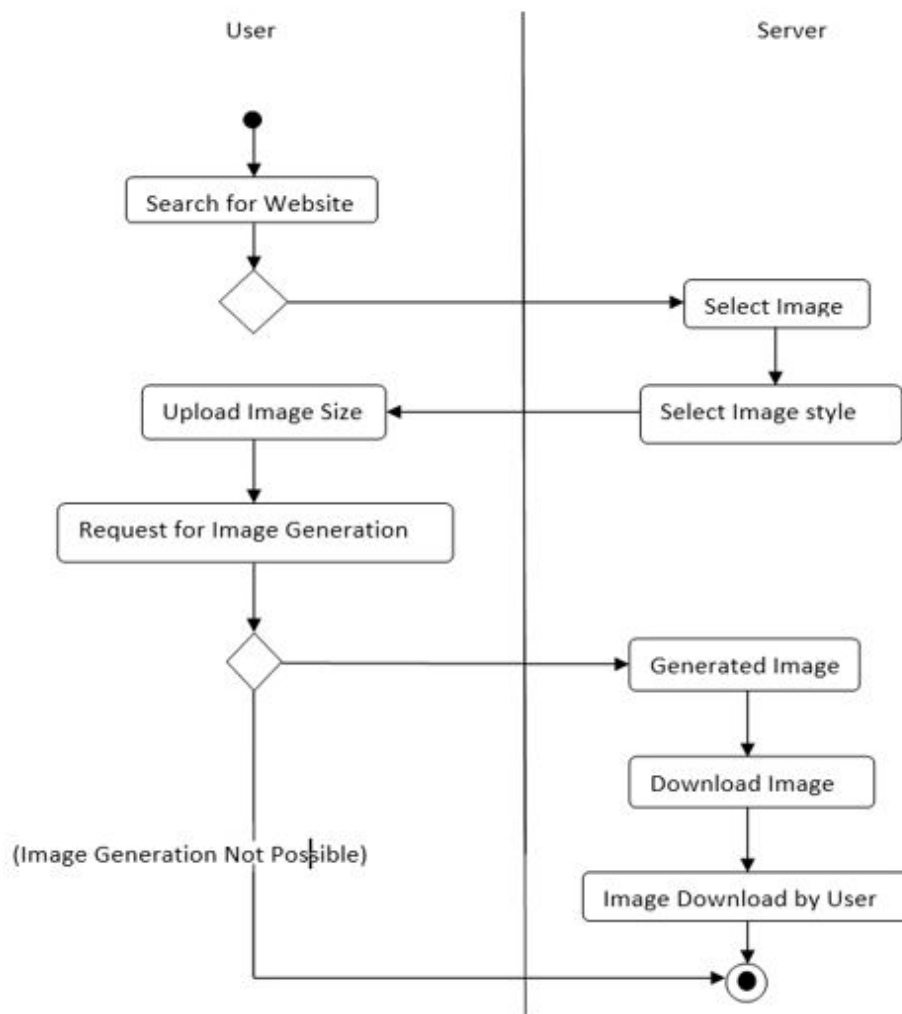


Figure 3.2: Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

In this Application, the user interacts with the application to perform all the operations and in return the server provides the resultant image to the user which can be downloaded.



### 3.2.3 Data-Flow Diagram

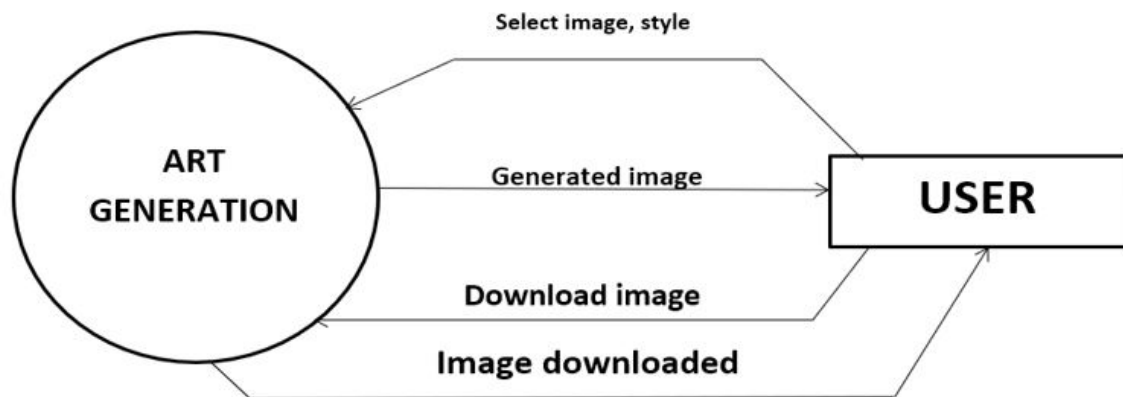


Figure 3.3: Data Flow Diagram

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.

In this Application, the user can interact with the system to generate an image. The user needs to input the content and style image and the system will return a generated image. The user can also download the generated image.

### 3.2.4 Sequence Diagram

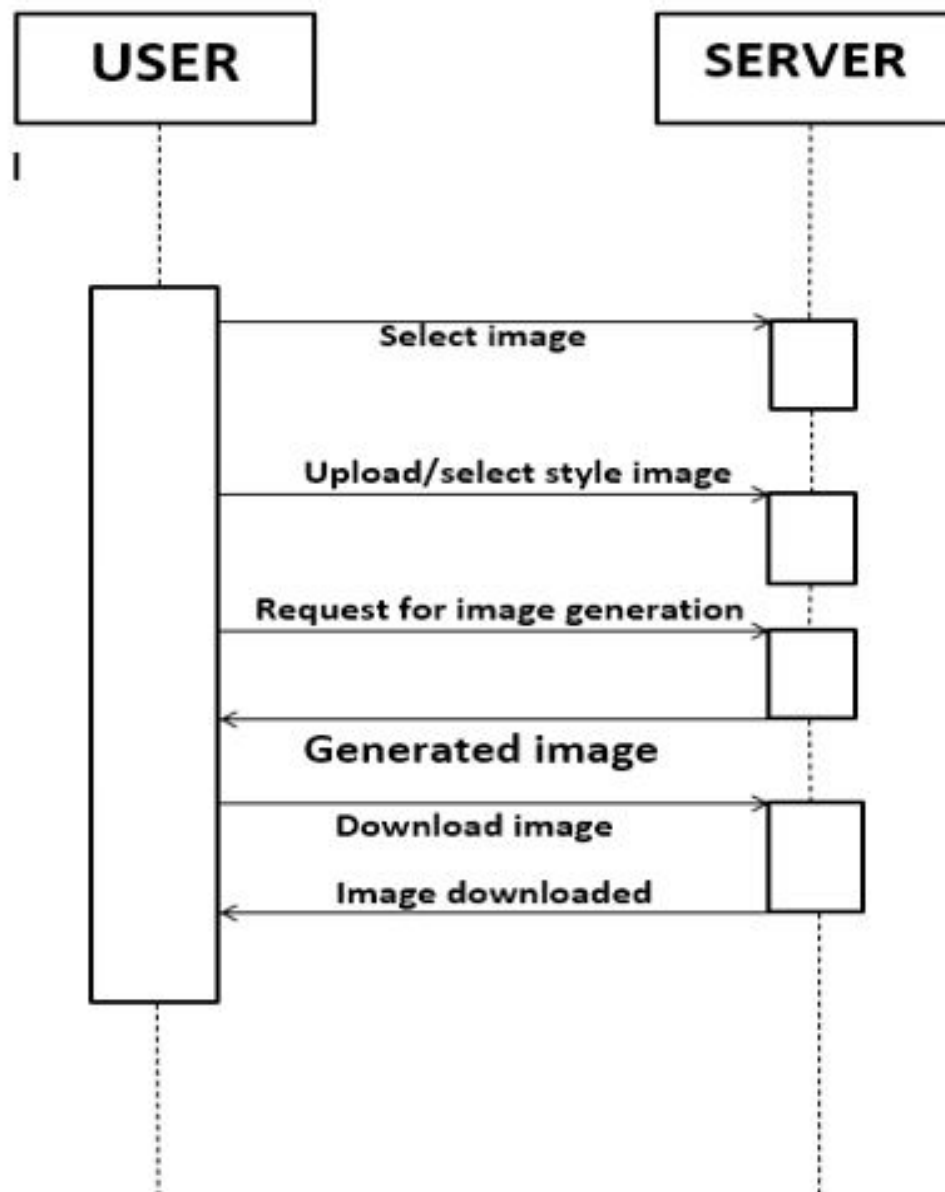


Figure 3.4: Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.

---

In this Application, the user first needs to select the image he wants to transform and then select/upload the style image. After this the user can request for generating the resultant image. The server will present the generated image before the user. The user can also download the image

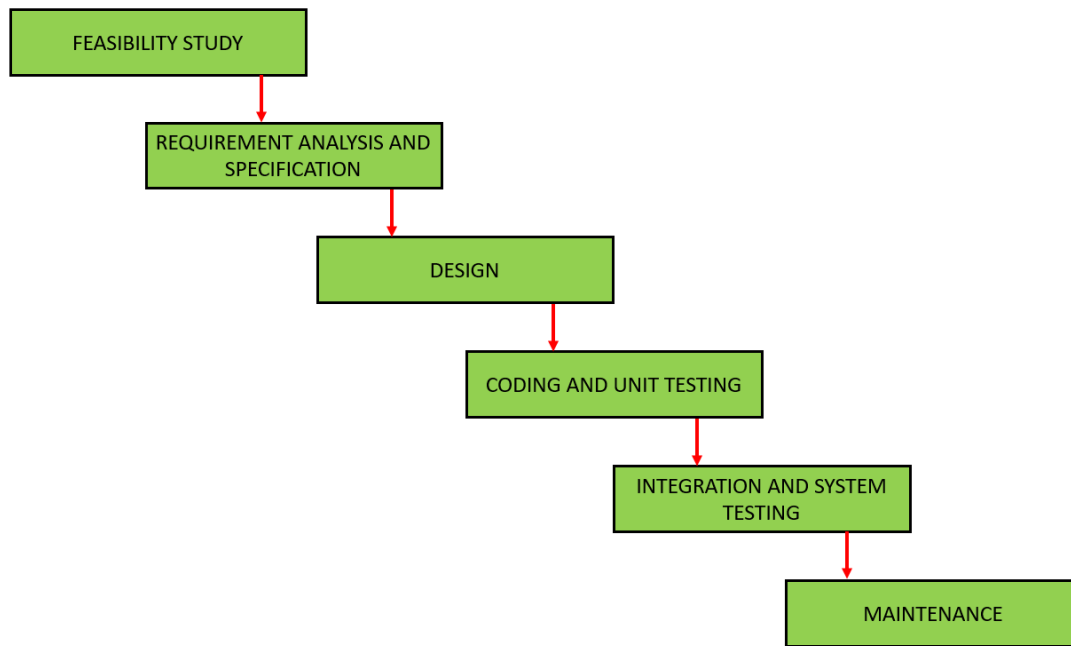
---

## Chapter 4

# METHODOLOGY AND TEAM

### 4.1 Introduction to Waterfall Framework

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as an input for the next phase sequentially. Following is a diagrammatic representation of different phases of waterfall model.



**Figure 4.1: WaterFall model**

The sequential phases in Waterfall model are-

1. **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
2. **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
3. **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
4. **Integration and Testing:** All the units developed in the imple-

---

mentation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

5. **Deployment of system:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

6. **Maintenance:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

### **Waterfall Model- Advantages and Disadvantages**

**Advantage** The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

---

**Disadvantage** The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

## **4.2 Team Members, Roles & Responsibilities**

Khushi Mittal - Deep Learning Application, Back End

Lipi Singh Parihar - Front End

Ishan Goyal - Front End

---

## Chapter 5

### System Testing

The designed system has been testing through following test parameters.

#### 5.1 Functionality Testing

In testing the functionality of the web sites the following features were tested:

##### 1. Links

- (a) Internal Links: All internal links of the website were checked by clicking each link individually and providing the appropriate input to reach the other links within.
- (b) External Links: Till now no external links are provided on our website.
- (c) Broken Links : Broken links are those links which so not divert the page to specific page or any page at all. By testing the links on our website, there was no link found on clicking which we did not find any page.



---

## 2. Forms

- (a) Error message for no input : Error messages have been displayed as and when we do not enter any one or both the images.
- (b) Optional and Mandatory fields : All the fields are mandatory fields. As we need both the style and content images for generating a styled image.

3. Database - There is no database needed for the project.

## 5.2 Performance Testing

The application is tested for various input images and the output images thus generated were quite satisfactory.

## 5.3 Usability Testing

In case of high resolution inputs the time taken to generate the output image also increases. While in case of low resolution pictures, the generation process is faster.

---

## Chapter 6

### TEST EXECUTION SUMMARY

Execution Test Summary Report is an overall view of Testing Process from start to end. Test Plan comes at the starting of project while Test Summary Report comes at the end of the testing process. This report is given to the client for his understanding purpose. The Test Summary Report contents are :

1. Content Image
2. Style Image
3. Final Total Loss
4. Number of Iterations
4. Status of Test Cases

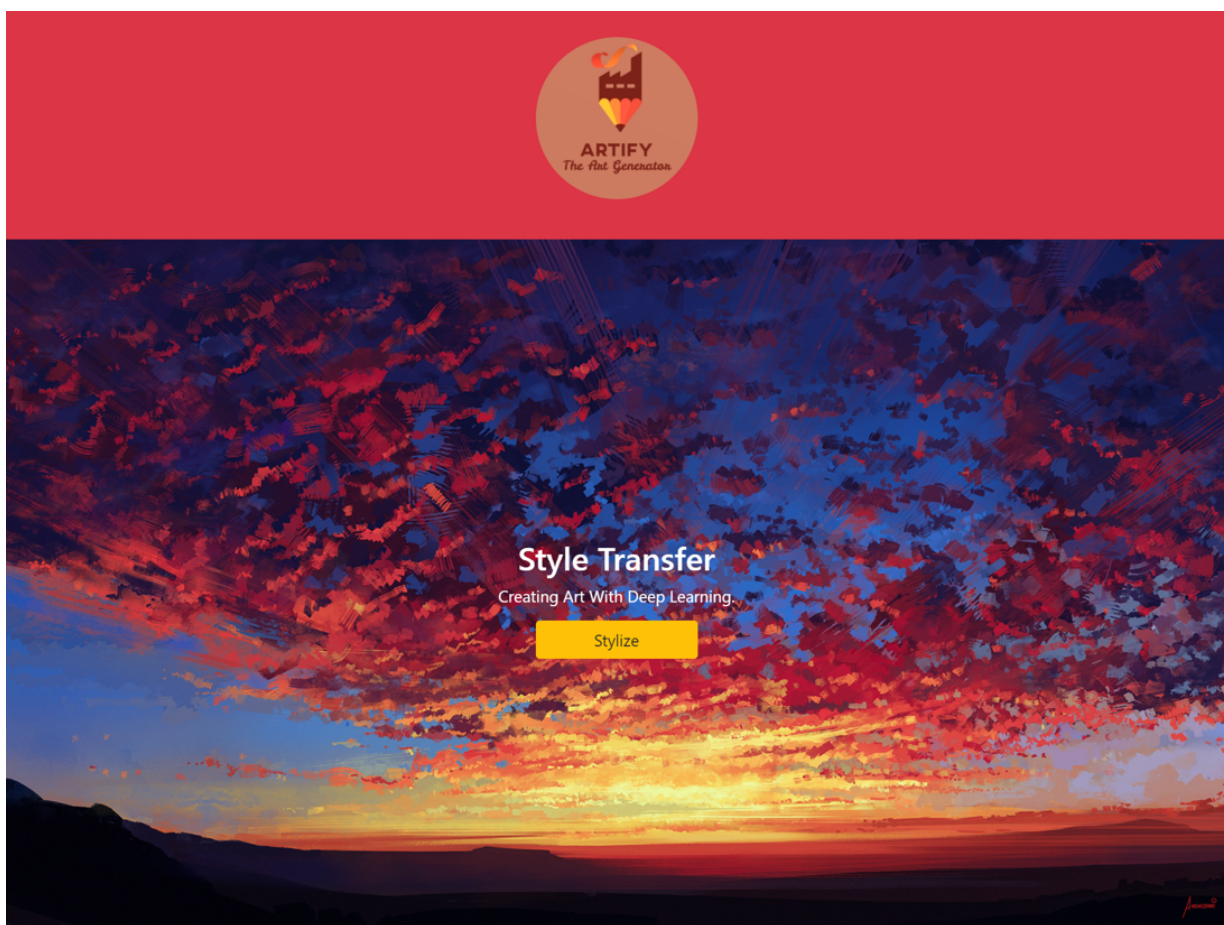
S.No	Content Image	Style Image	Test Case Status	Iterations	Final Total Loss
1	elephant.jpg	style1.jpg	Done	10	2819.6
2	elephant.jpg	style1.jpg	Done	20	1999.8
3	venice.jpg	style2.jpg	Done	10	4186.5
4	venice.jpg	style2.jpg	Done	30	2356.7
5	sydney.jpg	style3.jpg	Done	10	3367.6

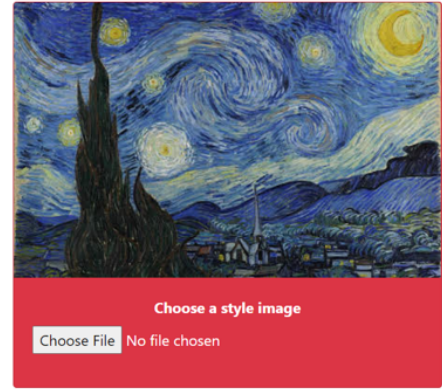
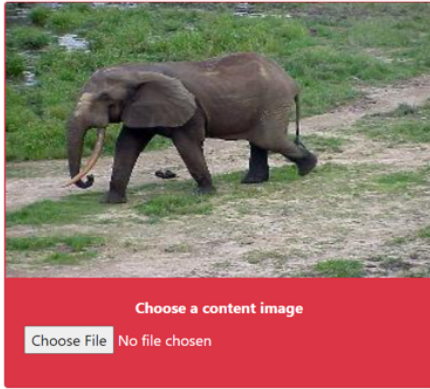
**Table 6.1:** Table of Results

---

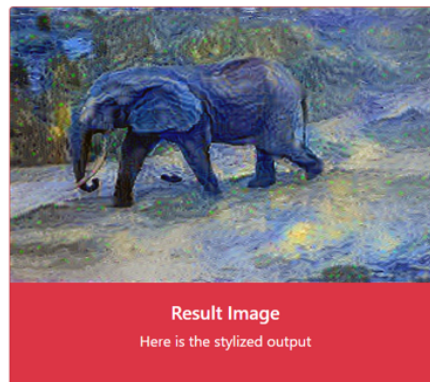
## Chapter 7

### PROJECT SCREENSHOTS





Stylize Image



---

## Chapter 8

# PROJECT SUMMARY AND CONCLUSIONS

### 8.1 Conclusion

We can conclude that :

- The project successfully generates the desired image.
- For faster results, use GPU to perform the transformation.
- If not satisfied with resultant image, then change the number of iterations.

---

## Chapter 9

### FUTURE SCOPE

If we're being honest with ourselves, not everyone is born an artist. But with recent advances in technologies like style transfer, almost anyone can enjoy the pleasure that comes along with creating and sharing an artistic masterpiece. This is where the transformative power of style transfer lies. Artists can easily lend their creative aesthetic to others, allowing new and innovative representations of artistic styles to live alongside original masterpieces.

Some of the areas where this project can be extended to are:

- Photo and video editors
- Artist-Community engagement
- Commercial art
- Gaming and Virtual Reality

---

## References

- [1] *www.medium.com; "Basic Example of Neural Style Transfer"*
- [2] *Gatys, Leon A.; Ecker, Alexander S.; Bethge, Matthias (26 August 2015). "A Neural Algorithm of Artistic Style".*
- [3] *www.heartbeat.fritz.ai; "Machine learning on mobile: What can you actually do with it?"*
- [4] *www.towardsdatascience.com; "Art With AI: Neural Style Transfer"*