TEST PLAN FOR

<<NEURAL STYLE TRANSFER>>

ChangeLog

Version	Change Date	By	Description
001	30.10.23	Palak Singh	Initial Draft

1 Introduction

NST, or Neural Style Transfer has revolutionized the field of image processing by allowing the amalgamation of artistic styles to photographs. First introduced by Gatys et al., NST relies on a slow and iterative optimization process. However, recent advances have introduced faster and more efficient approaches, such as Adaptive Instance Normalization (AdaIN) and Johnson's method. Gatys's method, which laid the foundation for NST, uses a convolutional neural network (CNN) to extract information about the content and style of an image. This is based on minimizing the difference between the feature representations of content images and stylized images. This approach, although revolutionary, is very time consuming. AdaIN introduced a revolutionary approach by reinterpreting version normalization to quickly combine content and style from arbitrary images. It eliminates the need for laborious optimization, allowing for real-time style transfer with great flexibility. Johnson's method takes a different route by using perceptual loss and a pre-trained network for high-level feature comparison. This allows for precise training using network capabilities and real-time image transformation, giving the best of both worlds. This paper provides a comprehensive look at NST techniques and their evolution, shedding light on the future of image processing and style transfer.

1.1 Scope

1.1.1 In Scope

Functional Requirement:-

Image Input: The system must accept input images in various formats (e.g., JPEG, PNG) for both content and style images.

Style Transfer Algorithm: The system shall implement a neural style transfer algorithm to combine the content of one image with the artistic style of another image.

User Interface: The system should provide a user-friendly interface for users to upload content and style images and initiate the style transfer process.

Style Customization: Users can adjust parameters or select predefined styles for customization.

Output Options: Users should have options to save, download, or view the generated images.

Non-Functional requirement:-

Performance:

Speed: The system should perform style transfer efficiently, with minimal processing time.

Scalability: It should be able to handle a scalable number of concurrent requests.

Quality:

Image Quality: The generated images must exhibit high-quality style transfer, retaining content and style faithfully.

Style Accuracy: The style transfer should closely match the chosen style image.

1.1.2 Out of Scope

Out of scope for the neural style transfer project are advanced artistic styles, highly specialized hardware, real-time video processing, and in-depth image editing features, among others.

1.2 Quality Objective

- Ensure the Application Under Test conforms to functional and non-functional requirements
- Ensure the AUT meets the quality specifications defined by the client
- Bugs/issues are identified and fixed before go live
- Ensure that the style transfer process is optimized for efficiency and speed, allowing for rapid transformation of images while maintaining high-quality results.

- Prioritize a user-friendly and intuitive interface to enhance user experience, making it easy for users to upload images and customize style transfer.
- Ensure that the system functions correctly on a variety of devices, browsers, and operating systems to maximize user accessibility.

1.3 Roles and Responsibilities

Detail description of the Roles and responsibilities of different team members like

- **QA Analyst**- Palak Singh
- **Test Manager**- Mr. Abhishek Goyal
- **Configuration Manager**- Prof. Akankskha
- **Developers**- Ragini Rani, Kalash Jain
- **Installation Team**- Ragini Rani, Kalash Jain

2 Test Methodology

2.1 Overview

An Agile methodology is the most suitable for my project. It allows for flexibility, ongoing testing, and adaptation, which are essential for projects that involve machine learning, image processing, and AI. Agile enables you to respond to changing requirements and refine the style transfer algorithm as you gain insights from testing and user feedback.

2.2 Test Levels

The testing to be performed is black box testing.
The testing is performed by the developers team along with QA and Configuration Manager.

Unit Testing:

Scope: Individual components and functions of the neural style transfer algorithm.

Objective: To verify that each component works as intended, including layers, loss functions, and optimization steps.

Testing Approach: Developers and machine learning engineers conduct unit tests to validate the correctness of the algorithm at a granular level.

Integration Testing:

Scope: The interactions and interfaces between various components, libraries, and frameworks used in the project.

Objective: To ensure that the integration of different components does not introduce errors or inconsistencies in the style transfer process.

Testing Approach: Developers and testers assess the data flow and interactions between components and detect any integration issues.

Functional Testing:

Scope: The complete neural style transfer system.

Objective: To validate that the system functions according to specified requirements and that it performs accurate style transfers.

Testing Approach: Testers execute functional tests by providing input images and verifying that the output images meet the desired content and style transfer criteria.

Performance Testing:

Scope: Assessing the system's speed and efficiency in handling style transfer tasks.

Objective: To measure how well the system performs in terms of processing time, memory utilization, and resource consumption.

Testing Approach: Performance tests evaluate the system's response time and resource usage under various loads and conditions.

Usability Testing:

Scope: The user interface and user experience.

Objective: To assess how user-friendly and intuitive the interface is for users uploading content and style images.

Testing Approach: Usability tests involve users interacting with the system to evaluate the ease of use, clarity, and navigation of the interface.

Security Testing:

Scope: The system's security mechanisms, especially for handling user data.

Objective: To identify and mitigate potential security vulnerabilities, including data breaches and unauthorized access.

Testing Approach: Security testing includes penetration testing, data encryption checks, and access control assessments.

Compatibility Testing:

Scope: The system's compatibility with various platforms and devices.

Objective: To ensure that the system functions correctly on different browsers, operating systems, and devices.

Testing Approach: Testers verify that the system is compatible with a range of devices and configurations.

Regression Testing:

Scope: The entire system after updates or changes.

Objective: To confirm that new changes or enhancements do not introduce defects or negatively impact existing functionality.

Testing Approach: Automated regression tests are executed to validate that previously tested features still work as expected.

2.3 Test Completeness

Here you define the criterias that will deem your testing complete.

For instance, a few criteria to check Test Completeness would be

- 100% test coverage
- All Manual & Automated Test cases executed
- All open bugs are fixed or will be fixed in next release
- All content and style transfer tests have been executed, ensuring that various input images have been processed successfully and meet the defined content and style transfer criteria.
- Automated regression tests have been executed, and previously tested features still work as expected after updates or changes.

3 Test Deliverables

Test cases:-

Test Case	Test Objective	Test Data	Expected Result		Actual Result	Pass/Fail
1	User Login	User Id and Password	Only Valid User login in the system		Unauthorized User can not login	Pass
1	Model Training	Training data with content and style images	Model converges, successfully learns style transfer		Model successfully learns style transfer	Pass
2	Inference	Test images (256x256 pixels) with varying content and styles	Generated images (256x256 pixels) exhibit desired style while retaining content		Generated images (256x256 pixels) exhibit desired style while retaining content	Pass
3	Content Preservation	Generated images	Content remains recognizable		Content recognizable	Pass

4	Style Transfer Quality	Generated images	Style closely matches style image		Style of the output closely matches style image	Pass
5	Validate content layer	Sample content images, pre-trained model weights	Content layer produces expected feature maps		Content layer produces expected feature maps	Pass
6	Validate style layer	Sample style images, pre-trained model weights	Style layer produces expected Gram matrices		Style layer produces expected Gram matrices	Pass
7	Verify style transfer accuracy	Input images (256x256 pixels) with known content and style	Output images (256x256 pixels) match the desired style and content		Output images (256x256 pixels) match the desired style and content	Pass
8	Verify data security	Testing with simulated security breaches	No unauthorized access or data breaches detected		Unauthorized User can not access	Pass
9	Assess the impact of updates	System before and after updates	Previously tested features still work as expected		Previously tested features work accurately	Pass
10	Image Size Verification	Test images (256x256 pixels) of various sizes	Generated images (256x256 pixels) maintain aspect ratio and content using a 3x3 kernel		Aspect ratio preserved, content recognizable using a 3x3 kernel	Pass

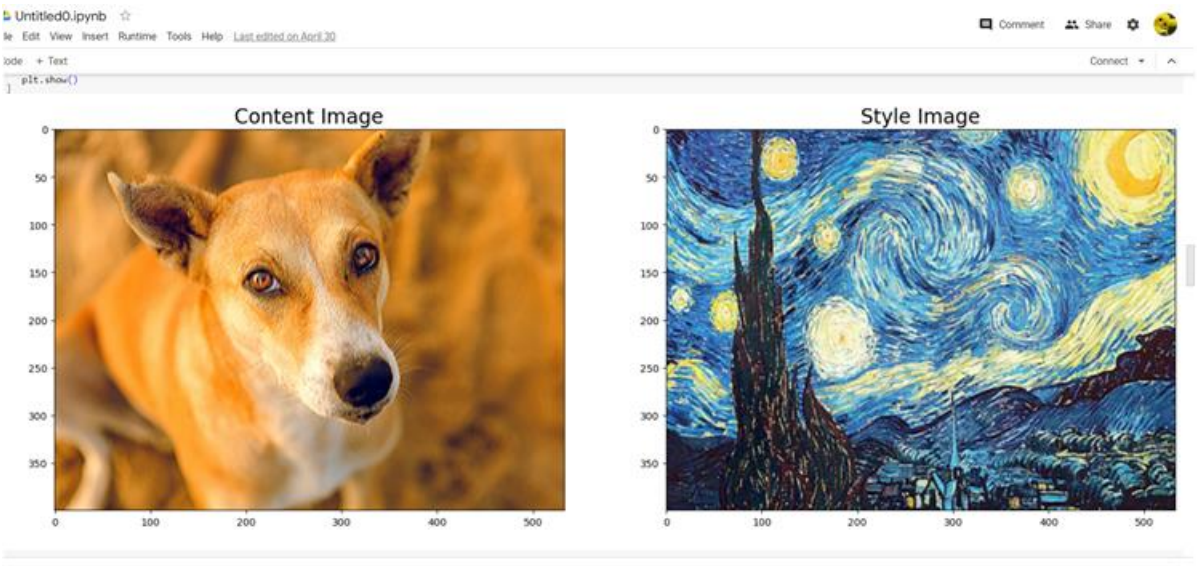
Decision Table for User Login

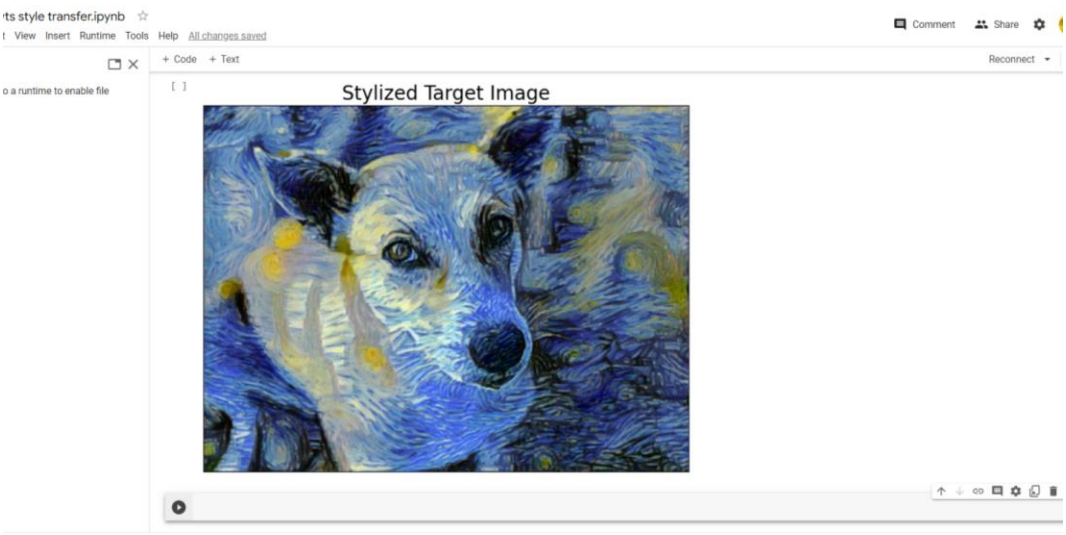
Conditions	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**
Username	False	True	False	True
Password	False	False	True	True
Output(e/h)	error	error	error	homepage

Decision Table for Image format

Conditions	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**
Image in png	False	True	False	True
Image in jpg	False	False	True	True
Output(e/a)	error	accepted	accepted	accepted

Test case Output Images





4 Resource & Environment Needs

4.1 Test Environment

It mentions the minimum **hardware** requirements that will be used to test the Application.

Following **software's** are required in addition to client-specific software.

- Windows 8 and above
- Office 2013 and above
- MS Exchange, etc.

5 Terms/Acronyms

Make a mention of any terms or acronyms used in the project

TERM/ACRONYM	DEFINITION
API	Application Program Interface
AUT	Application Under Test