
Software Requirements Specification

for

BLIND IMAGE RESTORATION AND DATA AU GMENTATION

Prepared by TEAM BRUTE FORCE

Avishi Tayal

Nandini Tyagi

Piyush Gupta

KIET GROUP OF INSTITUTIONS(AKTU)

COMPUTER SCIENCE

21/02/2023

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	2
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	3
2.5 Design and Implementation Constraints	3
2.6 User Documentation	4
2.7 Assumptions and Dependencies	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3
4. System Features	4
4.1 System Feature 1	4
4.2 System Feature 2 (and so on)	4
5. Other Nonfunctional Requirements	5
5.1 Performance Requirements	5
5.2 Safety Requirements	5
5.3 Security Requirements	5
5.4 Software Quality Attributes	5
5.5 Business Rules	6
6. Other Requirements	6
Appendix A: Glossary	6
Appendix B: Analysis Models	6
Appendix C: To Be Determined List	6

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This document aims to provide a comprehensive explanation of the well-known web application of blind image restoration and data augmentation. The interfaces and features of the web application are described in this document. It will inform users about the restrictions that must be met before utilizing the website so that it can be used effectively by potential industrial developers and end users.

Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

- a) Common users, such as hospitals and online retailers, who can use the web-application to examine every angle of any cell or product for improved diagnosis and analysis.
- b) Advanced/Professional Users, such as computer vision engineers or researchers, who can utilize the software for data augmentation and image restoration.
- c) Programmers who are interested in working on the project to improve it or fix bugs that are already there.

1.4 Product Scope

Blind Image Restoration and data augmentation web application is a tool that can be used to collect the data and restore the images. Nowadays, industries are obsessed with high-resolution datasets of human faces. These datasets can be used to train their models for a variety of applications, including virtual try-on, clothing translation, and many others. The data that is generated is accurate, and this dataset can be used in a variety of ways. For instance, if local brands are unable to afford superstars for advertising, Stylegan can be used to generate the synthesized image by mixing styles. Additionally, actors are unable to even challenge them for the deed. The same holds true for numerous other industrial applications.

1.2 References

1 To study GPEN Model

https://openaccess.thecvf.com/content/CVPR2021/html/Yang_GAN_Prior_Embedded_Network_for_Blind_Face_Restoration_in_the_CVPR_2021_paper.html

<https://github.com/yangxy/GPEN>

2. To study StyleGAN Model

<https://github.com/NVlabs/stylegan3>

<https://arxiv.org/abs/2208.07862>

3. Overall Description

3.1 Product Perspective

- **PRODUCT OVERVIEW-** Blind image restoration and data augmentation are both important techniques in the field of computer vision and image processing. The images generated by blind image restoration and data augmentation website will depend on the specific needs of the target audience, the available technology, and the market demand.
- **PRODUCT FUNCTIONS-** This section describes the functions of the application, including user registration, image restoration, image colorization, image inpainting and dataset generation.

3.2 Product Functions

- **USER REGISTRATION** – The Application should allow users to create accounts and register with the service.
- **IMAGE UPLOADING-** The Application should allow the user to upload a image from local system.
- **IMAGE RESTORATION–** The Application should allow to restore the distorted images uploaded by user.
- **GENERATE DATASET-** The Application should synthesize the different poses of the input image.

3.3 User Classes and Characteristics

- **Typical Users,** such as industrialist, who want to use Aaina for analyzing the image in a 360 degree and generating a clear image along with multiple views of the image.
- **Doctors** can use it to view the disease cell from all views.

- Programmers who are interested in working on the project by further developing it or fix existing bugs

3.4 Operating Environment

a) **Hardware requirements:** The StyleGAN and GPen models are computationally intensive, so we may need to specify the minimum hardware requirements for running our software, such as CPU, RAM, and GPU.

b) **Software dependencies:** Our software may have dependencies on other software libraries or frameworks. We may need to specify the required versions of these dependencies and any installation instructions.

c) **Operating system compatibility:** We may need to specify the operating systems that our software is compatible with.

d) **Installation instructions:** We may need to include step-by-step instructions for installing and configuring our software.

e) **Model requirements:** We may need to specify the requirements for running the StyleGAN and GPen models, such as the necessary input data, model files, and trained weights.

f) **Performance requirements:** If our software is designed for real-time image generation or other performance-critical tasks, we may need to specify the minimum performance requirements, such as minimum FPS or response time.

g) **Security requirements:** If our software is designed to process sensitive data or interact with other systems, you may need to specify the security requirements, such as encryption and authentication protocols

3.5 Design and Implementation Constraints

a) **Performance constraints:** The StyleGAN and GPen models are computationally intensive, so our software may have performance constraints. We may need to optimize our code to reduce computational overhead or to minimize memory usage. Additionally, the size of the generated images may need to be limited to ensure reasonable processing times.

b) **Memory constraints:** The size of the StyleGAN and GPen models can be quite large, which can lead to memory constraints. We may need to use techniques such as model pruning or data compression to reduce memory usage.

c) **Resource constraints:** Your software may have constraints on available resources, such as limited disk space or restricted access to GPU resources. You may need to design your software to work within these resource constraints.

d) **Compatibility constraints:** Our software may need to be compatible with other software or systems, which can introduce compatibility constraints. For example, we may need to ensure that the software works with a specific version of Python or TensorFlow.

e) **Data constraints:** The quality and quantity of the input data used to train the

StyleGAN and GPen models can impact the quality of the generated images. We may need to design your software to handle missing or incomplete data or to incorporate data augmentation techniques to improve the quality of the generated images.

f) **Regulatory constraints:** Our software may need to comply with regulatory constraints, such as data privacy laws or ethical guidelines. We may need to ensure that your software handles sensitive data appropriately and that the generated images do not infringe on copyright or other legal protections.

g) **User interface constraints:** If our software includes a user interface, you may need to design the interface to be user-friendly and accessible to a wide range of users. Additionally, the interface may need to be designed to work with a variety of devices and screen sizes.

3.6 User Documentation

- Overview of the Application
- Getting Started
- User interface
- Features
- Data Entry
- Data Output
- Troubleshooting
- Contact Information

4. Other Nonfunctional Requirements

4.1 Performance Requirements

- Response time: The application should respond to user requests within a specified time frame, such as 2 seconds, to ensure a fast and seamless user experience.
- Scalability: The application should be designed to handle a large volume of requests and users, without experiencing performance degradation or downtime. The system should be scalable to meet increasing demand as the user base grows.
- Availability: The application should be available 24/7, with a minimum uptime requirement of 99%, to ensure that users can access the service at all times.
- Reliability: The application should be reliable, with a low error rate and minimal downtime, to ensure that users can access the service when needed.

4.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

4.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

4.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

4.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

5. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>