

# **Software Project Management Report**

**Project Name:** Image processing visualization system

**Team Members:** Gongzikang\_202230310303

Ximingyuan\_202230310277

Xukunhai\_202230310302

Xuyuzhuo\_202230310272

**Completion Date:**2023.5

## Contents

1. Project Overview.....	3
2. Requirement Analysis.....	4
3. Project Plan and Completion Status....	14
4. Milestone Achievements.....	20
5. Risk Plan.....	23
6. Test Plan.....	25
7. Reflection and Summary.....	33
8. Iteration Plan.....	34

# 1. Project Overview

## 1.1 Project Name: Image Processing Visualization System

## 1.2 Project Background

With the rapid advancement of computer vision and image processing technologies, there is an increasing interest and demand for image processing and visualization. However, for beginners, learning and understanding these complex concepts and algorithms can be challenging. In order to meet the needs of beginners and help them better comprehend and learn image processing and visualization, we have decided to develop an image processing visualization system specifically designed for beginners.

The goal of this system is to provide an interactive learning environment that offers intuitive interfaces and real-time feedback to assist users in gaining a deep understanding of fundamental concepts and common algorithms in image processing. The system not only provides various visualizations for image processing tasks such as image filtering, edge detection, and image enhancement, but also includes a practical example of license plate detection and recognition based on deep learning. Through this example, users can experience firsthand the application, principles, and workflow of deep learning in solving real-world problems.

By using this system, beginners can enhance their understanding of image processing and visualization through practical hands-on exercises. They can explore different image processing algorithms and observe the real-time changes in the results. Furthermore, the system offers relevant documentation and tutorials to guide users step-by-step in mastering the basics of image processing and visualization.

In summary, the objective of this project is to provide a user-friendly, intuitive, and interactive learning platform for beginners to grasp the fundamental concepts and techniques of image processing and visualization. Through practical exercises and example demonstrations, users can quickly get started and gradually improve their skills. We hope that this system will inspire more people to develop an interest in the field of image processing and visualization, nurture talent, and drive further advancements in this domain.

## **2. Requirement Analysis**

### **2.1 Positioning and Objectives**

In today's highly advanced computer technology era, using computers to process large amounts of information and handle various tasks has become a common means to reduce manpower and improve work efficiency. With the rapid development of artificial neural networks, computers have become increasingly powerful in image processing, attracting more and more people's interest in computer image processing technology. However, for beginners, the initial encounter with rigid formulas and code can make it difficult to get started. Moreover, most neural network models used for image processing are difficult to intuitively demonstrate the impact of various hyperparameters on the final results. Therefore, to facilitate researchers in evaluating models and enhance the user experience for non-professionals in using models, we have developed an image processing visualization software. The image processing visualization software presents commonly used image enhancement techniques and simple image recognition neural networks to users in a visual and interactive manner. It aims to spark users' interest and allow them to understand the effects of various operations in a more intuitive way. Our software is primarily designed for experiential teaching of image processing techniques.

By providing a user-friendly interface and interactive features, the software allows users to explore and experiment with different image processing techniques. It provides visual feedback and real-time results, enabling users to directly observe the effects of various operations on images. Additionally, the software includes documentation and tutorials to guide users in acquiring basic knowledge and skills in image processing. Through hands-on experience and interactive demonstrations, users can quickly grasp the concepts and techniques of image processing. Our goal is to make image processing technology more accessible and comprehensible, particularly for beginners and non-professionals. We believe that by providing a visually engaging and interactive learning platform, we can inspire more people to take an interest in image processing and contribute to the further advancement of this field.

## 2.2 Object

The expected readers of this requirements analysis are:

Students who want to learn image processing techniques.

Teachers involved in image processing education.

Researchers engaged in image processing-related studies.

Trofessionals who need to perform basic image processing tasks.

## 2.2 Relevant Definitions

Definitions of Relevant Terms and Abbreviations used in the document:

1. Image Smoothing: Image smoothing refers to image processing methods used to highlight large areas, low-frequency components, main structures, or suppress image noise and interference high-frequency components. The goal is to make the image's brightness transition smoothly, reduce abrupt gradients, and improve image quality.

2. Image Sharpening: Image sharpening is the process of enhancing the contours, edges, and regions of gray level transitions in an image to make it appear clearer. It can be performed in either the spatial domain or the frequency domain.

Additional information (expanding the content):

Image smoothing techniques are commonly employed to reduce noise and make the image appear more visually pleasing. By suppressing high-frequency components, such as noise or fine details, the image is smoothed, resulting in a more uniform appearance and reduced abrupt transitions in brightness or color. Various methods, such as Gaussian smoothing or median filtering, can be used to achieve image smoothing.

On the other hand, image sharpening techniques aim to enhance the details and edges in an image, making them more prominent. This can improve the visual clarity and overall perception of the image. Spatial domain techniques, such as unsharp masking or Laplacian sharpening, enhance the local contrast by subtracting a smoothed version of the image from the original. Frequency domain techniques, such as using high-pass filters or applying the Fourier transform, can also be used to enhance image sharpness.

Understanding these terms is essential in the context of image processing, as they form the foundation for various algorithms and techniques employed in the field. By utilizing image smoothing and sharpening techniques appropriately, one can manipulate and enhance images to meet specific requirements, whether for visual aesthetics, noise reduction, or feature extraction purposes.

## **2.3 Overview of Requirements**

"Image Processing Visualization System" primarily targets the following requirements:

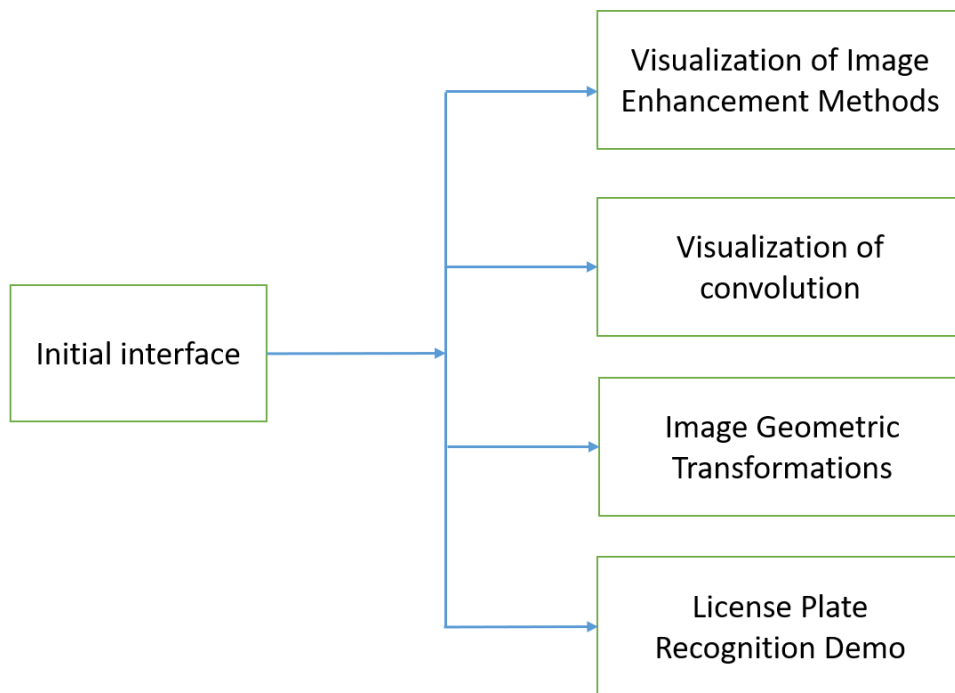
1. Visualization of Image Enhancement Methods: Utilizing OpenCV, the system enables users to apply grayscale transformation, histogram equalization, filtering, and other image enhancement techniques. Users can upload an image and observe the processed image in real-time on the interface. They can also adjust parameters to visually perceive the impact on image enhancement.

2. Convolutional Neural Network Visualization: As a fundamental technique in computer vision, the system visualizes the results achieved by adjusting parameters such as the size of convolutional kernels.

3. Demonstration of Simple Image Recognition: The system provides a neural network model for license plate recognition. Users can upload an image and experience the image recognition capability of the system.

These requirements ensure that users of the "Image Processing Visualization System" have access to visualized demonstrations of image enhancement methods, convolutional neural networks, and basic image recognition functionality. By offering these features, the system aims to facilitate users' understanding and exploration of image processing techniques in an interactive and user-friendly manner.

## **2.4 System Architecture**



## 2.5 Overview of Functional Requirements

Functionality	Detailed Description	Permissions
Visualization of Image Enhancement Methods	Translation: Users can utilize the graphical interface to apply various image enhancement methods to the loaded images in the system, including image smoothing, image sharpening, contrast adjustment, brightness adjustment, grayscale transformation, histogram equalization, and filtering. The processed images will be displayed in the corresponding window for visualization.	All

Geometric transformations of images	Users can perform geometric operations on images through the graphical interface, including image translation, image scaling, horizontal mirroring, vertical mirroring, image transpose, image cropping, and image rotation. The transformed images will be displayed in the respective window.	All
Convolutional Network Visualization	Images can be visualized by adjusting the size and stride of convolutional kernels in a convolutional neural network, allowing users to observe the visualization results of the network under different parameter settings.	All
Simple Image Recognition Demonstration	Provide a network model for license plate detection for users to experience. Users can upload images to see the output results of the model.	All
Graphical User Interface (GUI)	Provide a graphical user interface (GUI) that is easy to use for all operations, even without specialized knowledge.	



## **2.6 Functional Requirements Description**

### **2.6.1 Image Enhancement Method Visualization Module**

#### **Image Loading Functionality**

- 1) Load an image from the specified file path provided by the user.
- 2) Maintain the display of the image in a specific window.◦

#### **Image Smoothing Functionality:**

- 1) Use smoothing techniques to make the brightness of the image smoother, improving the image quality. Users can perform image smoothing operations by clicking on the icon.
- 2) Display the image after applying the image smoothing operation.
- 3) Users can save the processed image

#### **Image Sharpening Functionality:**

- 1) Use image sharpening techniques to enhance the contours of the image and make it sharper.  
Users can adjust the sharpening level by dragging the progress bar.
- 2) Display the image after applying the image sharpening operation.
- 3) Users can save the processed image.

#### **Contrast, Brightness, and Grayscale Adjustment Functionality:**

- 1) Users can adjust the contrast of the image by dragging the progress bar.
- 2) Display the image after making the adjustments.
- 3) Users can save the processed image.

#### **Brightness Adjustment:**

- 1) Users can adjust the brightness of the image.
- 2) Display the image after making the adjustments.
- 3) Users can save the processed image.

### **Grayscale Adjustment:**

- 1) Users can adjust the grayscale of the image.
- 2) Display the image after making the adjustments.
- 3) Users can save the processed image.

### **Histogram Equalization Functionality:**

- 1) Apply histogram equalization technique to achieve uniform distribution of grayscale levels in the image, thus increasing contrast and enhancing image details..
- 2) Users submit the image for processing and display the adjusted image.
- 3) Users can save the processed image.

### **Filtering Functionality:**

- 1) Apply median filtering algorithm to remove isolated noise points in the image and achieve image denoising.
- 2) Users submit the image for processing and display the adjusted image.
- 3) Users can save the processed image.

## **2.6.2 Image Geometric Transformation Module:**

### **Image Translation Functionality:**

- 1) Users can interact with a dialog box to set the translation height and width, adjusting the position of the image based on user input.
- 2) Display the adjusted image.
- 3) Users can save the processed image.

### **Image Scaling Functionality:**

- 1) Users can adjust the scaling ratio through a dialog box to resize the image accordingly.
- 2) Display the scaled image.
- 3) Users can save the processed image.

### **Horizontal Mirroring Functionality:**

- 1) Users can interact by clicking an icon to perform a horizontal mirroring operation on the image.
- 2) Display the adjusted image.
- 3) Users can save the processed image.

### **Vertical Mirroring Functionality:**

- 1) Users can interact by clicking an icon to perform a vertical mirroring operation on the image.
- 2) Display the adjusted image.
- 3) Users can save the processed image.

### **Image Transposition Functionality:**

- 1) Users can interact by clicking an icon to perform a transposition operation on the image.
- 2) Display the adjusted image.
- 3) Users can save the processed image.

### **Image Cropping Functionality:**

- 1) Users can specify the cropping ratio through a dialog box to crop the image accordingly.
- 2) Display the adjusted image
- 3) Users can save the processed image.

### **Image Rotation Functionality:**

- 1) Users can specify the rotation angle through a dialog box to rotate the image based on the user's input.
- 2) Display the adjusted image.
- 3) Users can save the processed image.

## **2.6.3 Convolutional Neural Network Visualization Module**

### **Parameter Adjustment Functionality:**

- 1) Users can adjust the number, size, and stride of convolutional kernels in the convolutional

neural network through a dialog box.

- 2) The parameters of the convolutional neural network are adjusted based on the user-provided values.

#### **Convolutional Neural Network Visualization Functionality:**

- 1) Visualize the output results of the convolutional neural network and display them in a specific window.
- 2) Users can save the results.

### **2.6.4 Simple Image Recognition Demonstration Module**

#### **Image Recognition Functionality:**

- 1) Implement image recognition using a convolutional neural network.
- 2) Display the results in a specific window and compare them with the original image.
- 3) Retrieve and display intermediate feature maps from the deep neural network.

## **2.7 User Requirements**

1. Beginner-Friendliness: The system should be designed with a user-friendly interface that is accessible to beginners, allowing them to easily use and understand the system's functionalities and operations. Beginners may not be familiar with concepts and algorithms in image processing and visualization, so the system should provide clear guidance to help them get started quickly.

2. Educational Aspect: The system should have an educational aspect that allows beginners to learn and understand the basic concepts and techniques of image processing and visualization through hands-on practice.

3. Interactivity: The system should be interactive, allowing users to interact with images in real-time and observe the results. Users should be able to preview and compare the effects of different image processing algorithms instantly, enabling them to have a visual understanding of the impact of different operations. They should also be able to adjust parameters in real-time to achieve the desired effects.

4. Operation guidance: The system should provide clear operation instructions to assist

beginners in using the system correctly and performing image processing operations. Users should be able to easily select and apply different image processing algorithms, and understand the purpose and impact of each option.

5. Example demonstrations: The system can provide example demonstrations based on real-world problems to help users better understand the applications of image processing and visualization. For example, through an example of license plate detection and recognition based on deep learning, users can learn about the application and working principles of deep learning in real-world problems, and experience the effects and performance of the algorithm through practical operations.

The core of user requirements is to provide a friendly, educational, and interactive learning platform for beginners, helping them understand and master the fundamental concepts and techniques of image processing and visualization. Through clear operation guidance, example demonstrations, and rich learning resources, users can deepen their understanding of image processing and visualization through practical operations and gradually improve their skills.

## **2.8 User Interface Requirements**

1. Clean and Intuitive: The user interface should be designed in a clean and intuitive manner, enabling beginners to easily understand and operate the system's functionalities. The interface layout should be clear, avoiding excessive complex elements and cluttered visual effects, allowing users to quickly find the desired functionalities and operational options.

2. Navigability: The user interface should have good navigability, allowing users to easily browse and access various functionalities of the system. The main functionalities and operational options should be presented in an intuitive manner, with easily understandable labels and buttons, enabling users to quickly locate and select the desired features.

3. Real-time Preview: The user interface should support real-time preview functionality, allowing users to immediately observe the effects of image processing operations. When users select different algorithms or adjust parameters, the interface should be able to update and display the image processing results in real time, helping users to intuitively understand the impact of different operations.

4. **Parameter Adjustment:** The user interface should provide convenient ways to adjust parameters, allowing users to flexibly modify the parameters of image processing algorithms according to their needs. Interactive elements such as sliders, input fields, or drop-down menus can be provided to enable users to intuitively adjust the parameters and observe the impact of parameter changes on the image processing results in real time.

5. **Accessibility:** The user interface should take into consideration accessibility requirements to ensure that all users can easily access and use the system. The interface should support keyboard navigation and screen readers, and provide sufficient contrast and legible fonts so that users can conveniently browse and interact with the interface content.

By meeting the aforementioned user interface requirements, the system can provide an intuitive, user-friendly, and well-navigated interface that helps beginners easily understand and utilize the various features and operations of the image processing system. The interface design should be clear and clutter-free, with a well-organized layout that allows users to quickly locate and access the desired functions and options. Additionally, real-time preview functionality should be integrated to provide immediate feedback on the effects of image processing operations, allowing users to visually comprehend the impact of different adjustments and parameter settings. With a focus on accessibility, the interface should be inclusive and considerate of users with different abilities, supporting alternative input methods and ensuring legibility for users with visual impairments. Overall, a well-designed user interface enhances the learning experience, empowers users to explore and experiment with image processing techniques, and promotes a seamless interaction between users and the system.

### **3. Project Plan and Completion Status**

The software development project was divided into four main phases: Project Selection and Preparation, Knowledge Exploration and Learning, Core Functionality Development, and Feature Integration and Testing. The details and progress of each phase are outlined below:

Project Selection and Preparation:

- 1) Conducted market research and identified the need for an image processing software.
- 2) Defined project objectives, scope, and requirements.

- 3) Established project team and allocated resources.
- 4) Completed project planning, including timeline and budget estimation.

#### Knowledge Exploration and Learning:

- 1) Conducted in-depth research on image processing techniques, algorithms, and tools.
- 2) Explored various software development methodologies and selected an appropriate approach.
- 3) Completed the selection of the development tools and technologies to be used.

#### Core Functionality Development:

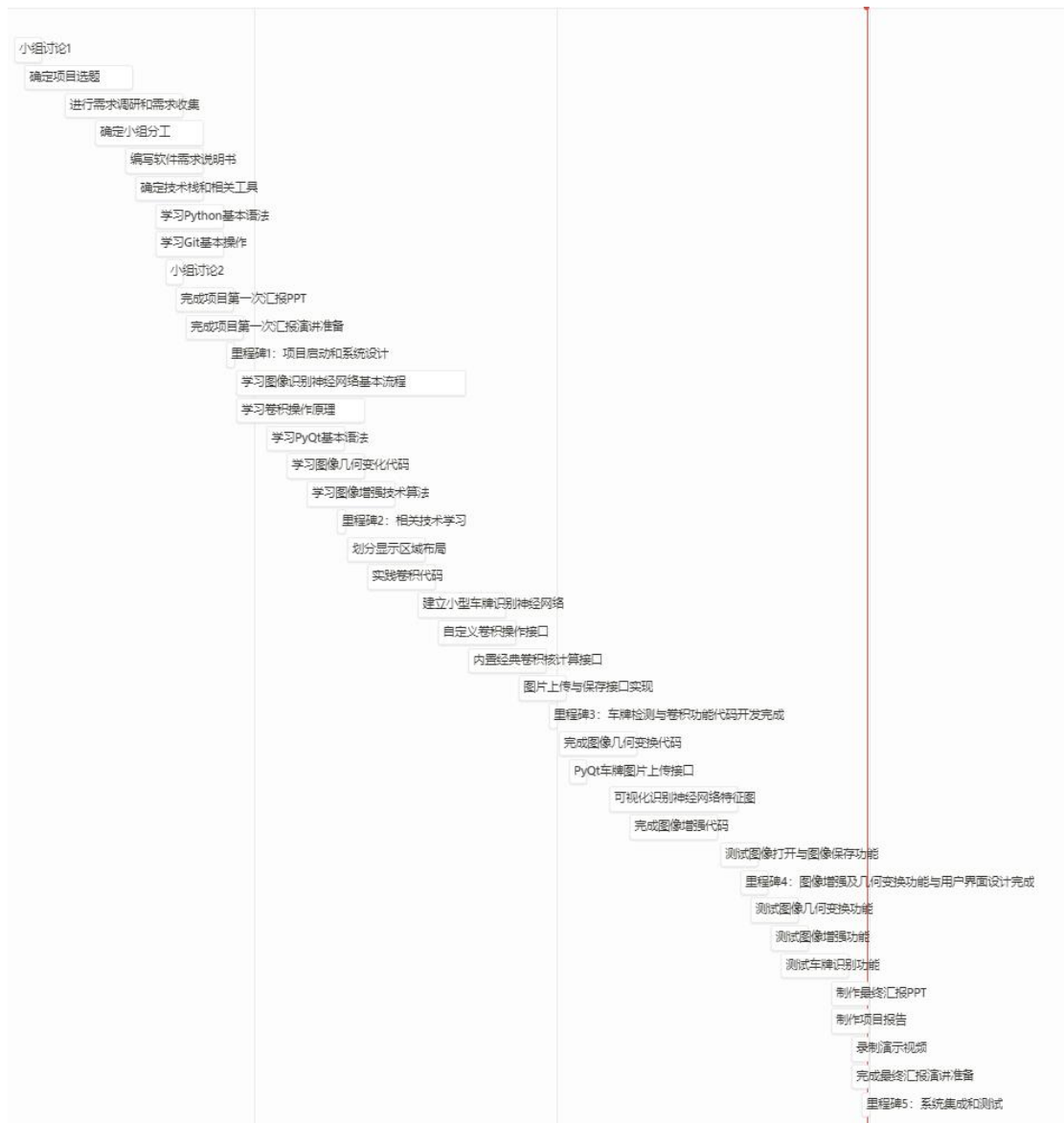
- 1) Developed the foundational features of the image processing software, such as image loading, basic image manipulation, and visualizations.
- 2) Implemented algorithms for common image processing operations, including image enhancement, filtering, and transformation.
- 3) Integrated user interface components with the core functionality to provide a seamless user experience.
- 4) Conducted iterative testing and debugging to ensure the reliability and performance of the core features.

#### Feature Integration and Testing:

- 1) Conducted extensive testing, including functional testing, performance testing, and user acceptance testing.
- 2) Identified and addressed any issues or bugs discovered during testing.

Overall, the project has been progressing according to the planned schedule, with each phase building upon the previous one. The core functionality has been successfully developed, and the team is currently integrating advanced features and conducting thorough testing. The project is on track for completion within the estimated timeline, and the team is confident in delivering a high-quality image processing software that meets the project objectives and user requirements.

Please refer to the Gantt chart below for a visual representation of the detailed task breakdown and the progress of each phase:



Milestones were set in each phase as significant markers for the completion of partial progress and the beginning of the next phase. Additionally, the completion dates of each milestone were established as a means to control the project's progress.

During the Project Design and Initiation phase, the project topic was determined, followed by feasibility exploration. Requirement research and collection were conducted, and a software requirements document was drafted. The team's roles and responsibilities were defined, along with the identification of necessary development tools. Learning the usage of relevant tools was also a part of this phase.

In the Knowledge Exploration and Learning phase, team members conducted research on the technical aspects required for their respective tasks. They learned relevant knowledge and



concepts and put them into practice to assess the feasibility of the development plan. Once the feasibility of the chosen technologies was confirmed, the project moved into the Core Functionality Development phase. During this phase, each team member focused on developing their assigned components, resulting in experiment code that could be run independently.

In the subsequent phase, interface design and interface-functionality integration were performed. Additional auxiliary features were developed as well. In the final phase, all the functionalities were integrated to ensure smooth operation through the user interface. Automated testing scripts were written to simulate various real-world scenarios and assess the system for any significant defects. If major defects were identified, a meeting would be held to discuss and establish a plan for their resolution.

Here are the tasks assigned to each team member:

GongZiKang	UI Design, Content Integration
XiMingYuan	Image Enhancement and Image Processing
XuKunHai	License Plate Detection and Recognition Example
XuYuZhuo	Testing, Convolution Visualization

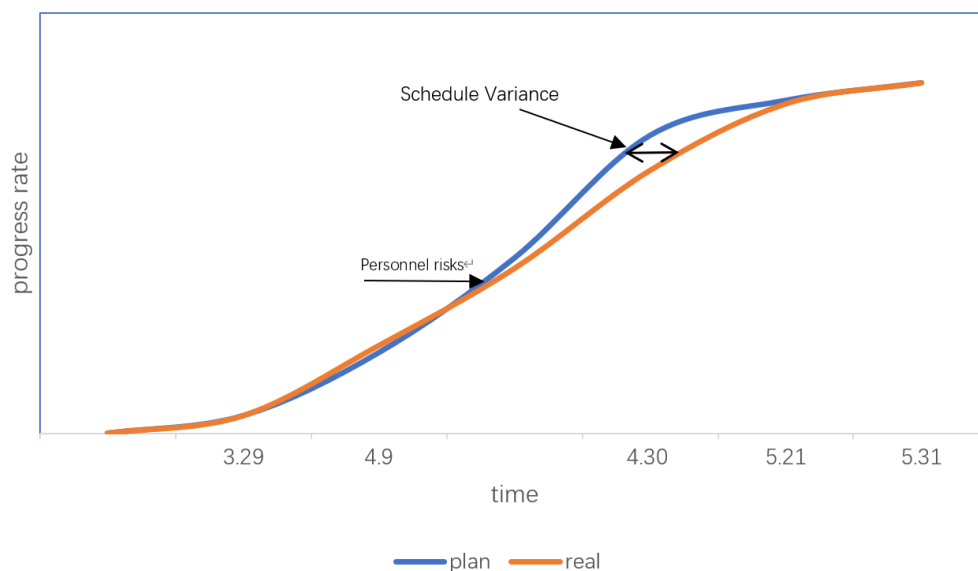
The completion status of the tasks is as follows:

Aa Task	📅 Date	☰ Member	🕒 Status
小组讨论1	March 8, 2023 → March 10, 2023	GZK XMY XKH XYZ	Completed
确定项目选题	March 9, 2023 → March 19, 2023	GZK XMY XKH XYZ	Completed
进行需求调研和需求收集	March 13, 2023 → March 24, 2023	GZK XMY XKH XYZ	Completed
确定小组分工	March 16, 2023 → March 26, 2023	GZK XMY XKH XYZ	Completed
编写软件需求说明书	March 19, 2023 → March 26, 2023	GZK	Completed
确定技术栈和相关工具	March 20, 2023 → March 26, 2023	GZK XMY XKH XYZ	Completed
学习Git基本操作	March 22, 2023 → March 28, 2023	GZK XMY XKH XYZ	Completed
学习Python基本语法	March 22, 2023 → March 28, 2023	GZK XMY XKH XYZ	Completed
小组讨论2	March 23, 2023 → March 24, 2023	GZK XMY XKH XYZ	Completed
完成项目第一次汇报PPT	March 24, 2023 → March 29, 2023	GZK XMY	Completed
完成项目第一次汇报演讲准备	March 25, 2023 → March 30, 2023	GZK XMY	Completed
里程碑1：项目启动和系统设计	March 29, 2023 → March 29, 2023	GZK XMY XKH XYZ	Completed
学习图像识别神经网络基本流程	March 30, 2023 → April 21, 2023	XKH	Completed
学习卷积操作原理	March 30, 2023 → April 11, 2023	XYZ	Completed
学习PyQt基本语法	April 2, 2023 → April 9, 2023	GZK XMY XKH XYZ	Completed
学习图像几何变化代码	April 4, 2023 → April 11, 2023	XMY	Completed
学习图像增强技术算法	April 6, 2023 → April 14, 2023	XMY	Completed
里程碑2：相关技术学习	April 9, 2023 → April 9, 2023	GZK XMY XKH XYZ	Completed
划分显示区域布局	April 10, 2023 → April 17, 2023	GZK	Completed
实践卷积代码	April 12, 2023 → April 18, 2023	XYZ	Completed
建立小型车牌识别神经网络	April 17, 2023 → April 25, 2023	XKH	Completed
自定义卷积操作接口	April 19, 2023 → April 26, 2023	XYZ	Completed
内置经典卷积核计算接口	April 22, 2023 → April 29, 2023	XYZ	Completed
图片上传与保存接口实现	April 27, 2023 → May 1, 2023	GZK	Completed
里程碑3：车牌检测与卷积功能代码开发	April 30, 2023 → April 30, 2023	GZK XMY XKH XYZ	Completed
完成图像几何变换代码	May 1, 2023 → May 8, 2023	XMY	Completed
PyQt车牌图片上传接口	May 2, 2023 → May 3, 2023	GZK	Completed
可视化识别神经网络特征图	May 6, 2023 → May 18, 2023	XKH	Completed
完成图像增强代码	May 8, 2023 → May 16, 2023	XMY	Completed
测试图像打开与图像保存功能	May 17, 2023 → May 20, 2023	XYZ	Completed

里程碑4：图像增强及几何变换功能与测试	May 19, 2023 → May 21, 2023	GZK XMY XKH XYZ	Completed
测试图像几何变换功能	May 20, 2023 → May 24, 2023	XYZ	Completed
测试图像增强功能	May 22, 2023 → May 25, 2023	XYZ	Completed
测试车牌识别功能	May 23, 2023 → May 29, 2023	XYZ	Completed
制作最终汇报PPT	May 28, 2023 → May 31, 2023	XKH XYZ	Completed
制作项目报告	May 28, 2023 → May 31, 2023	XMY	Completed
录制演示视频	May 30, 2023 → May 31, 2023	GZK	Completed
完成最终汇报演讲准备	May 30, 2023 → May 31, 2023	XKH XYZ	Completed
里程碑5：系统集成和测试	May 31, 2023	GZK XMY XKH XYZ	Completed

milestone	program completion date	actual completion date	whether in time
Project initiation and system design	2023.3.29	2023.3.29	√
Related technical learning	2023.4.9	2023.4.8	√
Core code development	2023.4.30	2023.5.11	×
UI interface and other functions	2023.5.21	2023.5.21	√
Integration and testing	2023.5.31	2023.5.31	√

From the above images and table, it can be seen that in this project management course design, all functionalities have been successfully completed. However, in the core functionality development phase, which is the third phase, the milestone was not completed within the specified time. The specific development progress is shown in the following graph:



From the above line graph, it can be observed that in the development plan, the completion rate of the software was initially low because this phase was in the first stage according to the development plan requirements. Topic selection and discussions were underway, and the development progress remained slow as all team members were in the phase of technical

verification and learning. In the middle stage of development, the progress significantly accelerated as each member worked on their assigned tasks without concerns for other functionalities. Towards the later stage of the project, the progress slowed down, which was due to the need for integrating and testing all the functionalities in the project. However, during the actual development process, due to personnel risks, the mid-term progress of the project was lower than the project plan. It required adjustments to the development plan to ensure the smooth progress of the project and timely completion of all development tasks. In response to the project risks, our team decided to form teams of two individuals responsible for similar tasks to mitigate the situation where a team member was absent, ensuring that no functionality was left unattended. We simplified the functionalities by retaining the core features and transferring the development tasks to other members within the team. The absent team members were assigned to subsequent testing tasks. In the later stages of the project, we successfully caught up with the project plan and completed all development tasks within the designated timeframe.

## **4. Milestone Achievements**

### **Milestone 1: Project Initiation and System Design**

- Milestone Content:

- Define project objectives and scope.
- Conduct requirement research and gathering.
- Complete system architecture design.
- Determine the selection of technology stack and relevant tools.
- Define development tasks and assign responsibilities.

- Achievement Summary:

- Planned Completion Date: March 29, 2023
- Actual Completion Date: March 29, 2023
- Summary of Achievement: The system design and requirement analysis tasks were completed according to the planned schedule within the specified date. The project's initial proposal for the course was also completed.

- Achievement Analysis:

- Next Steps: Begin the learning process for the technologies required for the project.
- Current Risks: There are certain technical risks that will be analyzed and addressed in the next phase of work.

## **Milestone 2: Knowledge Exploration and Learning**

- Milestone Content:

- Learn the syntax of OpenCV and PyQt.
- Learn the basic process of image recognition neural networks.
- Learn the principles and code of convolution operations.

- Achievement Summary:

- Planned Completion Date: April 9, 2023
- Actual Completion Date: April 8, 2023
- Summary of Achievement: The related technology learning tasks were completed within the specified time, and simple demo tests for each functionality were performed.

- Achievement Analysis:

- Impact on Project Progress/Quality: The successful completion of this phase demonstrates the feasibility of the chosen technologies and mitigates technical risks.
- Next Steps: Proceed with the development of core code for license plate detection and image enhancement.

## **Milestone 3: Completion of License Plate Detection and Convolution Functionality Code**

- Milestone Content:

- Implement the code for license plate detection functionality.
- Implement the code for convolution functionality.
- Perform unit testing and integration testing to ensure the correctness of the functionalities.

- Achievement Summary:

- Planned Completion Date: April 30, 2023
- Actual Completion Date: May 11, 2023
- Summary of Achievement: The core code for license plate detection was delivered within the specified time (April 27, 2023), while the core code for image convolution missed this milestone. This was due to a member's unexpected hand fracture, which hindered the development progress of this module.
- Achievement Analysis:
  - Impact on Project Progress/Quality: The accident involving team member Xu Yuzhuo directly affected the development progress of the image convolution functionality and subsequently the user interface development in the next phase.
  - Next Steps: Implement the image convolution functionality using the 2D convolution method in OpenCV (cv2.filter2D), and let Gong Zikang handle the research on classical convolution kernels. Xu Yuzhuo will be responsible for code testing after recovering from the injury.
  - Current Risks: Personnel risk, progress risk.

## **Milestone 4: Completion of Image Enhancement, Geometric Transformation, and User Interface Design**

- Milestone Content:
  - Implement the code for image enhancement and geometric transformation functionalities.
  - Design the style and layout of the user interface.
  - Implement the interaction and response functionalities of the user interface.
  - Test and optimize the user interface.
- Achievement Summary:
  - Planned Completion Date: May 21, 2023
  - Actual Completion Date: May 21, 2023
  - Summary of Achievement: All planned functionalities of the system were successfully

implemented within the specified time, and basic functionality testing was passed.

- Achievement Analysis:

- Next Steps: Perform system-level testing and finalize the project documentation for the final delivery.

## **Milestone 5: System Integration and Testing**

- Milestone Content:

- Integrate the various modules and ensure the coordination and consistency of functionalities.

- Perform system-level testing, including functionality testing and performance testing.

- Fix and adjust any identified issues and defects.

- Release the system.

- Achievement Summary:

- Planned Completion Date: May 31, 2023

- Actual Completion Date: May 31, 2023

- Summary of Achievement: System testing, project documentation organization, and project delivery were completed within the specified time.

- Achievement Analysis:

- Next Steps: Improve the system functionalities and project documentation based on user feedback and suggestions from the final defense.

## **5. Risk Plan**

### **3.1 Risk Identification**

Technical Risk: Risk related to technology selection

Personnel Risk: Unexpected absence of key team members, team collaboration issues

Requirements Risk: Frequent changes in requirements

Other Risks: Such as holidays, etc.

### 3.2 Risk Assessment

	Probability of Occurrence	Negative Impact	Priority
Technical Risk	20%	Replacing the technology used requires a reevaluation of the project's feasibility and the development of a new project plan. Extending the time for technology learning may directly result in missing milestones and affecting the project timeline.	C
Personnel Risk	40%	In the event of an accident involving key project members, it can have varying impacts on the project. In milder cases, it may affect the project timeline, while in more severe cases, it could potentially lead to project failure.	A
Requirements Risk	25%	Frequent changes in requirements can lead to technical risks and schedule risks.	B
Other Risks	5%	If holidays are not included in the development plan, it may result in a pause in progress and a delay in submission deadlines.	D

### 3.3 Risk Mitigation Strategies

Technical Risk: Conduct early technical validation and prototype development to ensure technical feasibility before proceeding with formal development.

Personnel Risk:



1. Conduct regular knowledge sharing sessions within the team to discuss technical implementation details and learn from each other.
2. Assign at least two individuals to collaborate on the development of critical modules to mitigate the risk of key personnel unavailability.

#### Requirement Risk:

1. Conduct feasibility assessment collectively as a team before accepting requirement changes to ensure their necessity.
2. Project manager should schedule and allocate resources appropriately for handling requirement changes, establish a new timeline for the changes, and conduct periodic evaluations.

#### Other Risks:

1. Allocate sufficient development time during the early stages of the project plan;
2. Reserve buffer time to accommodate unforeseen circumstances and ensure a certain level of flexibility.

### **3.4 Risk Review**

1. Conduct risk review every three weeks to assess the effectiveness of risk mitigation strategies.,
2. Summarize lessons learned and incorporate them into the improvement of the project management process.

## **6. Test Plan**

### **6.1 Testing objectives**

The main objectives of this software testing are to check for any critical errors (bugs) in the program and ensure that it performs various functions correctly to achieve the expected results. We expect the program to handle image files correctly and display processed files accurately in the designated area. Additionally, the program should handle user input correctly and provide appropriate warnings or prompts when invalid information is entered. For critical functionalities, it is important to ensure quick response times to provide a good user experience.

The key testing objective in this software testing is to test the license plate recognition feature of the program. This includes testing the accuracy of license plate recognition for different scenarios, GUI testing to ensure correct prompts when no image is selected, and testing the ability to complete license plate recognition tasks within 1.5 seconds. In addition to license plate recognition, the testing also covers all image processing functionalities in the program, such as convolution, image segmentation, image blurring, and more.

## 6.2 Testing Plan

The system will be tested using a unit testing strategy. This testing will be fully automated and will utilize the patch method to simulate user inputs. The testing will be conducted using QTest combined with unittest to write test cases, and XTestRunner will be used to generate test reports.

For this testing, a set of 10 different images will be selected as test resources. These images will include blurred images, images with smaller license plates, images with rotated license plates, images with license plates positioned off-center, images of performance cars' license plates, images with multiple license plates, and normal images, among others. These images will be used to assess the system's performance under various scenarios.

The testing approach will ensure that the system can handle different types of images and accurately recognize license plates. It will also verify the system's ability to respond to user inputs, generate appropriate warnings or prompts, and meet the expected performance requirements.

The use of automated testing and comprehensive test cases will help identify any issues or errors in the system and ensure its functionality is thoroughly tested. The generated test reports will provide valuable insights into the system's performance and assist in identifying any areas that require further improvement or optimization.

Testing work will be carried out by Xu Yuzhuo, who will test all system functionalities. For simple functionalities, a single test case will be written to check if the function can achieve its intended purpose correctly. For regular functionalities, at least three test cases will be written to test if the function can perform correctly under different conditions. For critical

functionalities (such as license plate recognition), at least seven test cases will be written. The testing period will be in the 15th and 16th weeks, and all testing and test report generation should be completed before Wednesday of the 16th week.

The specific test case writing plan is as follows:

**Open\_image:**

- 1) Test whether the function can correctly open a .jpg file and display the image in the corresponding location.
- 2) Test whether the function can correctly open a .png file and display the image in the corresponding location.
- 3) Test whether the function can correctly open a .jpeg file and display the image in the corresponding location.
- 4) Test whether the function can correctly open a .webp file and display the image in the corresponding location.
- 5) Test whether the function can correctly open a file under a Chinese path.
- 6) Test whether the function can open non-image files, and if not, whether it provides a prompt.
- 7) Test whether the function can open larger images.

**Save image:**

- 1) Test whether the function can save the image in .jpg format.
- 2) Test whether the function can save the image in .png format.
- 3) Test whether the function can save the image in .jpeg format.
- 4) Test whether the function can save the image in a Chinese path.
- 5) Test whether the function allows the user to save as a non-image file and whether it provides a prompt.

**Vertical flip:**

- 1) Verify if the image can be vertically flipped, with a size of 640\*480 pixels.
- 2) Verify if the image can be vertically flipped, with a size of 1600\*1200 pixels
- 3) Verify if the image can be vertically flipped, with a size of 557\*373 pixels.

**Horizontal flip:**

- 1) Verify if the image can be **horizontal** flipped, with a size of 640\*480 pixels.

- 2) Verify if the image can be **horizontal** flipped, with a size of 1600\*1200 pixels.
- 3) Verify if the image can be **horizontal** flipped, with a size of 557\*373 pixels.

**Rotate image:**

- 1) Verify if the function can rotate the image by a positive angle.
- 2) Verify if the function can rotate the image by a negative angle.
- 3) Verify if the function can rotate the image by a floating-point angle.

**Crop image:**

- 1) Verify if the function can correctly crop the image with a crop ratio of (100, 100, 200, 200).
- 2) Verify if the function can correctly crop the image with a crop ratio of (10, 20, 30, 40).
- 3) Verify if the function can correctly crop the image with a crop ratio of (10, 10, 20, 20) .

**Convert to gray:**

Verify if the function can correctly convert the image to grayscale.

**Apply gaussian blur:**

- 1) Check if the function can apply the Gaussian blur to the image correctly when the input convolution kernel is valid.
- 2) Check if the function displays a warning dialog when the input convolution kernel is a floating-point number or an integer.

**Sharpen:**

- 1) Check if the image is correctly sharpened with a size of 640\*480.
- 2) Check if the image is correctly sharpened with a size of 1600\*1200.
- 3) Check if the image is correctly sharpened with a size of 557\*373.

**Apply histogram equalization:**

- 1) Check if the histogram equalization can be applied correctly to the image with a size of 640\*480.
- 2) Check if the histogram equalization can be applied correctly to the image with a size of 1600\*1200.
- 3) Check if the histogram equalization can be applied correctly to the image with a size of 557\*373.

**Apply gamma transform:**

- 1) Test if the gamma transformation can be correctly applied when a floating-point number is entered.
- 2) Test if the gamma transformation can be correctly applied when an integer is entered.
- 3) Test if the gamma transformation can be correctly applied when a negative number is entered.

**Conv:**

- 1) Test if the convolution operation can be correctly executed when an integer is entered.
- 2) Test if the convolution operation can be correctly executed when a floating-point number is entered.
- 3) Test if the convolution operation can be correctly executed when a negative number is entered.
- 4) Test if a warning dialog is correctly displayed when an invalid convolution kernel is entered.

**Upload image:**

- 1) Test if the function can correctly load and save the image in the corresponding path in Window 2.

**Detect image:**

- 1) This function can correctly identify the license plate number of different types of pictures.
- 2) The middle picture and the final result picture can be correctly displayed.
- 3) The identification time is within 1.1s.
- 4) When the user does not select the picture, will the prompt dialog box pop up

## **6.3 Test execution**

Use the following images as test case resources:







During the test execution, it is necessary to wrap all test cases using the TestSuite method from unittest into a test suite, and execute the test suite using the runner.run method. Out of all the test cases, there were 10 tests that did not pass. The details are as follows:

XTestRunner		Test Report						
TEST GROUP/TEST CASE	DESCRIPTION	DURATION	COUNT	PASS	FAIL	ERROR	VIEW	SCREENSHOTS
test_Test_Showcv	测试系统是否能正确读取文件并运行展示		52	41	10	1	<a href="#">Detail</a>	
test_apply_gaussian_blur_01	检查在输入正确的卷积核时能否正确应用高斯模糊增加图片的模糊程度	0.23 s			fail			
test_apply_gaussian_blur_02	检查在卷积核大小为浮点数时会不会弹出警告对话框	0.2 s			fail			
test_apply_gaussian_blur_03	检查在卷积核大小为4时会不会弹出警告对话框	0.19 s			fail			
test_detect_img_09	检查是否能以正确识别拥有多个车牌图像的车牌号	1.4 s			fail			
test_open_image_04	测试open——image是否能正确打开图像文件并展示在窗口中，中文路径	0.26 s			fail			
test_open_image_06	测试系统能否打开其他文件，如果不能打开其他文件，是否有提示	0.26 s			fail			
test_rotate_image_03	旋转小数	0.27 s			fail			
test_save_image_04	测试能否将图像存储为中文路径下	0.26 s			fail			
test_save_image_05	测试能否将图像存储为指定图像文件的格式，如果可以，是否输出提醒	0.33 s			fail			
test_upload_image	测试在窗口2中能否正确加载文件，并且正确处理，保存在对应路径下	0.29 s			fail			
Cases Total:			52	41	10	1		
XTestRunner 1.7.0; 2023 © 壹壹科技 - <a href="http://test.info">http://test.info</a>								

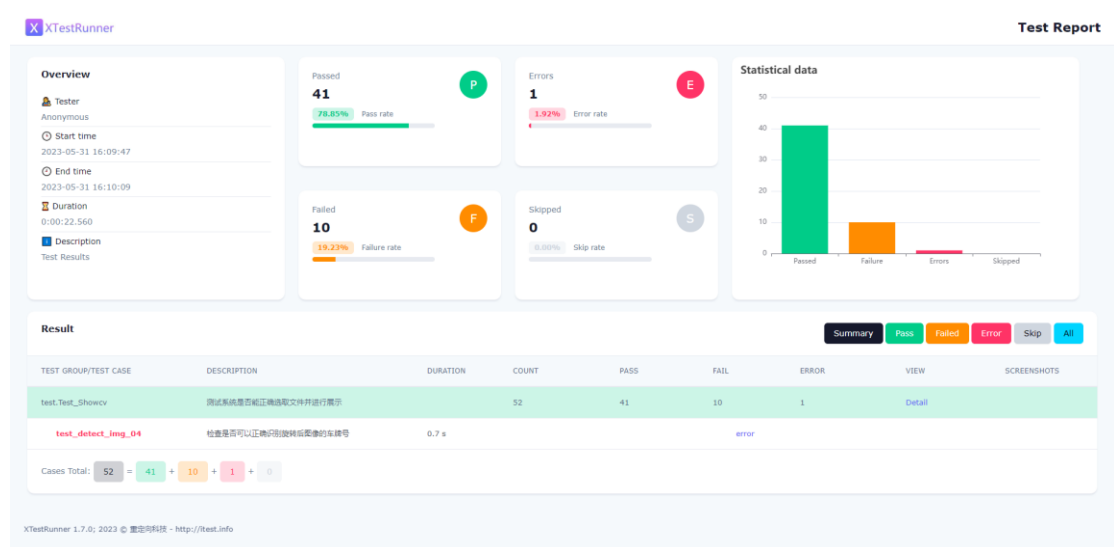
Among them, there were significant bugs in the testing of the Gaussian blur functionality. The function did not correctly execute the image blurring process and did not prompt the user for input. In all the functions related to opening and saving images, users are unable to choose paths with Chinese characters. This can cause the program to crash. The specific reason is that the `getOpenFileName` method in the `Dialog` is unable to read Chinese paths correctly. Users need to save the images to be opened in paths with English characters in order to load them correctly, but there is no such prompt in the system. In the image rotation functionality, users are unable to enter floating-point numbers. When the user inputs a decimal point ('.'), the window does not read it and only reads the numerical part. Additionally, for the most important functionality, license plate recognition, we used various images to assess the correctness of the recognition feature. In the case of multiple license plates present in an image, the system is unable to recognize them correctly, but it does output the resulting image and provides output content. However, for rotated images, the system is unable to recognize them correctly and cannot generate the final result image. This may be due to the elimination of the part containing the license plate during the image scaling process.

## 6.3 Test Report

Use XTestRunner to generate the unit test report for the entire system. The report includes statistics on the pass rate, failure rate, and error rate of all test cases. The report will be saved in the specified folder in the format of an HTML file. Please refer to the following image for the



detailed test report:



The Overview section records the details of the testing personnel, testing time, and relevant descriptions. The middle section contains the record of the number of passes, pass rate, number of errors, error rate, number of failures, and failure rate for this test. The table on the right displays a histogram showing the number of passes, failures, and errors for this test. The Results section stores all the test cases for this test and categorizes them as pass, failed, or error.

The pass rate for this test is 78.85%, with 41 test cases passing. The reasons for the other cases not passing or resulting in errors have been analyzed in the previous section in relatively specific terms. Overall, most of the functionalities tested in this round are relatively complete. However, there are still some major bugs in the system that need to be fixed, as well as some minor bugs that need improvement.

## 7. Reflection and Summary

This report aims to provide a comprehensive summary and evaluation of the project, reviewing its achievements, challenges, lessons learned, and experiences gained.

### 1. Project Achievements:

- Developed an image processing visualization software that presents commonly used image enhancement techniques and simple image recognition neural networks to users in a visual and interactive manner.

- Achievement 1: Successfully implemented the project's key functionalities and requirements.
- Achievement 2: Overall, most of the critical milestones were completed as planned, ensuring stable project progress.
- Achievement 3: The team demonstrated close collaboration, efficient communication, and fostered a positive work environment.
- Achievement 4: Each team member experienced personal growth and breakthroughs in technical skills and software project management.

## 2. Project Challenges:

- Challenge 1: Changes in requirements and scope expansion resulted in adjustments to the project plan.
- Challenge 2: Unforeseen injuries to team members posed risks to project timelines.
- Challenge 3: Insufficient communication and coordination led to delays and duplication of work.

## 3. Lessons Learned:

- Lesson 1: Recognize the importance of requirement management and change control, and establish more effective change management processes.
- Lesson 2: During the project planning phase, thoroughly establish mitigation measures such as collaborative learning during the technical learning phase, assigning multiple individuals to develop core functionalities simultaneously, etc.
- Lesson 3: Strengthen communication and collaboration by establishing regular meetings and communication channels to ensure effective communication and coordination among team members.

# 8. Iteration Plan

The current version of the system has successfully completed all the fundamental functionalities. However, due to technical adjustments made during the development process, some functional details did not meet the initial expectations. In the next iteration, we will focus on the following improvements:

1. Evaluate the severity of bugs identified during testing and prioritize them based on their

impact. We will address the bugs in the order of their significance, enhancing the system's robustness.

2. Implement an "Undo" functionality, allowing users to revert to the previous step. This requires adding a button and implementing a data structure within the program to store the previous image, facilitating easy restoration.
3. Introduce a feature that enables users to compare the modified image with the original one, providing a clearer visualization of the image processing effects.
4. Enhance the custom convolution operation by allowing users to adjust the kernel size and stride.
5. Provide explanatory text or images to help users understand the purpose of each functional button, such as explaining the parameters of linear transformations.

By incorporating these improvements, we aim to refine the system's functionality, enhance user experience, and provide more flexibility and control over image processing operations.