

Restoring the Past

Interim Report

DT 282

BSc in Computer Science (International)

**Chung Ho Lung**

**C20348256**

**Jack O Neil**

School of Computer Science

Technological University, Dublin

**08/11/2023**

Abstract

As time progresses, so do the changes around us—characters slowly age, and our memories inevitably fade away. Only the photos in our hands serve as a tangible preservation of those moments. However, photos, too, are not immune to the effects of time; they gradually become faded, worn, or damaged.

As technology advances, the task of restoring these aging photos has become more accessible. Photo restoration no longer needs to be a cumbersome and expensive process.

This project proposes a user-friendly program designed for easy photo repair. Whether users are on a mobile phone or computer, they can effortlessly restore old photos.

To achieve this functionality, the application is web-based and designed to be light on computing power. This ensures that even mobile phone users or those with older computers can implement the function seamlessly.

As time passes and technology evolves, photos may still experience fading, wear, or damage, but there is no longer a need for concern. Technological advancements empower us to repair and preserve photos more easily.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

Chung Ho Lung\_\_\_\_

Chung Ho Lung

08/11/2023

Acknowledgements

I, Michael Chung, would like to take this opportunity to express my gratitude to all those who contributed to the creation of this project.

I would particularly like to thank Jack O’Neil, my supervisor, for his invaluable assistance in all aspects of this project. From suggesting useful materials to providing moral support during the writing of necessary logs and reports, Jack has been instrumental. Having someone passionate and engaged in the development cycle of this project has been highly beneficial. Through countless meetings, whether in person or otherwise, Jack has provided valuable guidance and support throughout the lifespan of this project.

Finally, I would like to express my appreciation to those around me, including family and peers, who assisted in the application’s testing process during the development cycle. The honest feedback on the application's many iterations, design decisions, and features has been invaluable, and I firmly believe it has helped steer me in the right direction

Table of Contents

[1. Introduction 7](#_Toc119925914)

[1.1. Project Background 7](#_Toc119925915)

[1.2. Project Description 7](#_Toc119925916)

[1.3. Project Aims and Objectives 7](#_Toc119925917)

[1.4. Project Scope 7](#_Toc119925918)

[1.5. Thesis Roadmap 7](#_Toc119925919)

[2. Literature Review 8](#_Toc119925920)

[2.1. Introduction 8](#_Toc119925921)

[2.2. Alternative Existing Solutions to Your Problem 8](#_Toc119925922)

[2.3. Technologies you’ve researched 8](#_Toc119925923)

[2.4. Other Research you’ve done 8](#_Toc119925924)

[2.5. Existing Final Year Projects 8](#_Toc119925925)

[2.6. Conclusions 8](#_Toc119925926)

[3. System Design 9](#_Toc119925927)

[3.1. Introduction 9](#_Toc119925928)

[3.2. Software Methodology 9](#_Toc119925929)

[3.3. Overview of System 9](#_Toc119925930)

[3.X. Other Sections 9](#_Toc119925931)

[3.X. Conclusions 9](#_Toc119925932)

[4. Testing and Evaluation 10](#_Toc119925933)

[4.1. Introduction 10](#_Toc119925934)

[4.2. Plan for Testing 10](#_Toc119925935)

[4.3. Plan for Evaluation 10](#_Toc119925936)

[4.4. Conclusions 10](#_Toc119925937)

[5. Prototype Development 11](#_Toc119925938)

[5.1. Introduction 11](#_Toc119925939)

[5.2. Prototype Development 11](#_Toc119925940)

[5.3. Other Sections 11](#_Toc119925941)

[5.4. Conclusions 11](#_Toc119925942)

[6. Issues and Future Work 12](#_Toc119925943)

[6.1. Introduction 12](#_Toc119925944)

[6.2. Issues and Risks 12](#_Toc119925945)

[6.3. Plans and Future Work 12](#_Toc119925946)

[6.3.1. GANTT Chart 12](#_Toc119925947)

[Bibliography 13](#_Toc119925948)

# Introduction

This section contains a summary of the planned project completion, along with the documentation and development processes. The fundamentals of the application will be covered in this chapter, including the project background and proposed plans for development methods.

The project concept revolves around the belief in image restoration and image storage. This project aims to create a web-based application, allowing users to restore their damaged images and store the restored versions.

The application enables users to upload their images and restore them using different functions. Users can select specific functions to restore the damaged image, and these functions facilitate immediate restoration and display. Users can also activate advanced mode for better restoration results, continuing the process until they click the "Done" button. The restored image serves as a template, allowing users to restore other damaged areas in the image using various functions or export the restored image.

To ensure accessibility for everyone, this application will be web-based, allowing users on any device to restore their old images and save time or memory. Anyone seeking image repair will benefit from this application.

Among the language alternatives for image restoration, such as OpenCV, scikit-image, or JavaCV, Python with OpenCV is chosen for this project due to its large community and ease of setup. Despite Python's performance limitations, OpenCV, written in C/C++, efficiently addresses computationally intensive tasks in computer vision. This allows for dedicated time to research and understand the algorithms for image restoration.

As Python is the development language, choosing the best backend is essential. Two web frameworks, Django and Flask, are considered for implementation. Django, a high-level web framework following the Model-View-Controller (MVC) pattern, provides an ORM system for easy database work. Flask, a lightweight and flexible web framework following the WSGI standard, is known for simplicity and is suitable for smaller projects. For this application's focus on lightweight and flexible functionality, Flask is chosen to set up the website.

The combination of OpenCV with Python and Flask, supported by a strong community, makes it an ideal solution for this project. A weekly log of project work will be created, supported by regular commits to a GitHub page. These updates will showcase the project's lifecycle, from initial wireframes and intentions to the completed product.

This application presents ample opportunities for expansion, including the possibility of integrating machine learning algorithms. There is room for additional features to aid in restoring various types of damaged images, potentially extending to the restoration of digital images. Machine learning may offer users true auto restoration without the need to manually select functions for image restoration.

The user interface of the website only has a navigation bar, functional sidebar and image upload area. To keep this website clear and have minimal buttons there, ensuring users have a better user experience there. The website will be like a light version of this website : <https://www.photopea.com/>

In order to protect users' privacy, this website does not have any database to store users' image data. All these image uploads are only stored during the process and deleted after the user closes the website.

## Project Background

While valiant efforts have been made to painstakingly, manually restore old films to their former glory by enthusiasts, this process is slow and makes for little headway into the preservation of the vast archives of decaying films. Even a single minute of film contains over 1000 individual frames and each frame must be manually processed. Technology has since advanced to a point that much of this time-consuming but important work can be automated. The expertise for automating this process can be found in the area of Computer Vision and Image Processing. The extent to which this processing can be automated, depends on the intelligence of the algorithm used. However, even just semi-automation could speed up the process considerably and allow for the rescue of whole archives of otherwise lost gems. In this project, a post-processing algorithm will be designed to tackle the main effects of film damage, towards automating / semi-automating the restoration process

## Project Description

This project aims to develop an application dedicated to restoring damaged images that have undergone various types of deterioration. The focus will be on addressing five main types of damage: Fading, Scuffs, Tears and Rips, Stains and Spots, as well as Water and Mold damage. Different algorithms and functions will be employed to effectively restore images affected by these diverse forms of deterioration.

Images serve as tangible proof of people's past experiences and their lives in the present, capturing every moment of their existence. They act as a historical record, allowing individuals to reminisce about the joys and sorrows they have experienced. Unfortunately, even images, which serve as a timeless repository of 'time,' are inevitably susceptible to the wear and tear brought about by the passage of time. This project seeks to mitigate the impact of such degradation by employing advanced algorithms tailored to each specific type of damage, thus ensuring the preservation of these invaluable visual records for future generations.

## Project Aims and Objectives

1. **Understanding Types of Image Damage:**
   * Identify and categorize the main types of image damage: Fading, Scuffs, Tears and Rips, Stains and Spots, Water, and Mold damage.
2. **Algorithm Selection:**
   * Choose at least one algorithm for restoring a specific type of image damage. This will serve as a proof of concept for the project's feasibility.
3. **Language Selection:** 
   * There are several languages capable of image restoration, such as C/C++, Python with OpenCV, or Java with JavaCV.
4. **Wireframe and Frontend Setup:**
   * Create a wireframe for the application's frontend. Define the user interface (UI) components, layout, and functionality.
   * Set up the frontend using a framework like React, Angular, or Vue.js. Establish the necessary buttons and functions based on the categorized types of image damage.
5. **Backend Implementation:**
   * Set up the chosen backend.
   * Implement at least one main functionality to restore images. This could involve integrating the selected algorithm for a specific type of damage.
   * Output the restored image result.
6. **Additional Functionalities:**
   * Expand the backend to include functionalities for restoring other main types of damaged images identified in step 1.
   * Integrate corresponding algorithms for each damage type.
   * Test and ensure the application is capable of effectively restoring images with various forms of damage.

By following these steps, you'll systematically progress from understanding the problem to implementing a proof of concept and finally developing a comprehensive image restoration application. This approach ensures a structured and manageable development process for your project.

## Project Scope

**Project Scope**

1. **Objective:**
   * Develop an image restoration application capable of addressing various types of damage, including Fading, Scuffs, Tears and Rips, Stains and Spots, Water, and Mold damage.
2. **Algorithm Proof of Concept:**
   * Select and implement at least one algorithm for restoring a specific type of image damage to demonstrate the feasibility of the project.
3. **Language and Backend Selection:**
   * Choose a suitable programming language for the project, considering options like C/C++, Python with OpenCV, or Java with JavaCV.
   * Decide on the language backend for the application.
4. **Backend Implementation:**
   * Establish the chosen backend infrastructure.
   * Implement at least one main functionality for image restoration, incorporating the selected algorithm for a specific type of damage.
   * Ensure the application can output restored image results.
5. **Additional Functionalities:**
   * Expand the backend to accommodate functionalities for restoring other main types of damaged images identified earlier.
   * Integrate corresponding algorithms for each damage type.
   * Thoroughly test and validate the application's capability to effectively restore images with diverse forms of damage.
6. **Structured Development Process:**
   * Systematically progress through each step from understanding the problem to implementing a proof of concept and developing a comprehensive image restoration application.

**What the Project Isn't About:**

Understanding what the project isn't about is equally important to avoid scope creep and manage expectations. It helps clarify limitations and prevents the project from expanding beyond its intended boundaries. In the context of an image restoration project, some aspects that might fall outside the project scope could include:

1. **Advanced Image Analysis:**
   * The project may not delve into highly complex image analysis techniques beyond the scope of restoring damaged images.
2. **Hardware Integration:**
   * the primary goal is software-based image restoration, the project may not involve integrating specific hardware components for image processing.
3. **Extensive User Authentication and Authorization:**
   * The project not extensively cover user authentication and authorization mechanisms unless security is a primary concern.
4. **Integration with External Systems:**
   * The project scope not include extensive integration with external systems or databases unless it directly contributes to image restoration functionalities.

## Thesis Roadmap

# Literature Review

## 2.1. Introduction

This chapter I will talk about what kind of the programme language and backend able to success this functionality and why I pick the language and why not other.

## 2.2. Alternative Existing Solutions to Your Problem

C++ or Python 🡺 Opencv

OpenCV, or Open Source Computer Vision Library, is an open-source computer vision and machine learning software library. It provides a wide range of tools and functions for image and video analysis, as well as computer vision applications. OpenCV is written in C++ and has bindings for various programming languages, including Python.

The library includes a comprehensive set of functionalities, such as image processing, feature detection, object recognition, machine learning, and computer vision algorithms. OpenCV is widely used in academic research, industrial applications, and hobbyist projects for tasks like image and video analysis, facial recognition, gesture recognition, augmented reality, and robotics.

OpenCV has a large and active community of developers and researchers who contribute to its continuous improvement and expansion of features. It plays a crucial role in the development of computer vision applications due to its versatility and performance.

Java 🡺 JavaCV

avaCV is a Java wrapper for OpenCV (Open Source Computer Vision Library) and other computer vision libraries. It allows Java developers to access and use the functionalities provided by OpenCV and other related libraries seamlessly in Java applications.

JavaCV provides Java bindings for various popular computer vision and multimedia processing libraries, including OpenCV, FFmpeg, ARToolKitPlus, and more. This makes it easier for Java developers to incorporate computer vision capabilities into their applications without having to write native code.

During this project, I will use Opencv with Python to develop my application .

There have several reasons :

1. **OpenCV Python Binding :**
   * Opencv has well-maintained Python binding that allow developers to access the full functionality of the library in Python. This makes it straightforward to use OpenCV’s extensive set of computer vision algorithms without having to write low-level code.
2. Rich Ecosystem:
   * Python has a vast ecosystem of libraries and frameworks that complement OpenCV. NumPy, SciPy, and Matplotlib, among others, seamlessly integrate with OpenCV, providing powerful tools for numerical operations, scientific computing, and data visualization.
3. Community Support:
   * Python has a large and active community of developers. The popularity of OpenCV with Python means that there is a wealth of resources, tutorials, and community support available. This makes it easier for developers to find help and share knowledge.

Overall, the ease of use, extensive community support, and seamless integration between OpenCV and Python make this combination a popular choice for a wide range of computer vision applications.

## 2.3. Technologies you’ve researched

In this project I will develop a Web Based Application. As the reasons I required a backend to success all of these image processing functions. In chapter 2.2, I will use python to develop my project .

Flask and Django and the best backend that I can use . As they are Python backend

* Flask is a micro web framework written in Python. It is lightweight and designed to be simple and easy to use, making it a popular choice for building web applications and APIs.
* Django is a high-level web framework written in Python that encourages rapid development and clean, pragmatic design. It follows the Model-View-Controller (MVC) architectural pattern, but in the Django world, this is referred to as the Model-View-Template (MVT) pattern.

As my project is a Web Based Application. There will not have many page which mean I not need Django to success my project and Flask is the best option for me.

## 2.4. Other Research you’ve done

Image restoration is a domain within the field of computer vision and image processing that focuses on the enhancement or recovery of images that have undergone degradation or corruption. This degradation can occur due to various factors such as noise, blurring, compression artifacts, and other forms of distortion.

Key areas of research within image restoration include:

1. Denoising:
   * Denoising techniques aim to remove or reduce unwanted noise from images. Noise can be caused by various factors, including sensor limitations, low-light conditions, or transmission errors. Common methods include the use of filters, wavelet transforms, and deep learning approaches.
2. Deblurring:
   * Deblurring is concerned with removing blur from images. Image blur can result from motion, defocus, or other factors. Deblurring algorithms attempt to recover the sharpness and clarity of the original image by estimating and compensating for the blur.
3. Super-Resolution:
   * Super-resolution techniques are designed to enhance the resolution of an image beyond its original size. These methods aim to reconstruct high-resolution details that may be missing or lost in the original low-resolution image. Super-resolution can be achieved through single-image or multi-image approaches.
4. Inpainting:
   * Inpainting involves filling in missing or damaged regions of an image. This can be useful in scenarios where parts of an image are corrupted or need to be removed. Inpainting algorithms attempt to predict and generate plausible information to complete the missing regions.

## 2.5. Existing Final Year Projects

## 2.6. Conclusions

# 3. System Design

## 3.1. Introduction

## 3.2. Software Methodology

The software Methodology in this project will be Agile method.

There have many reasons that I choosing to use Agile methodologies to implement my application :

1. **Flexibility and Adaptability** 
   * Agile embraces change and allow me to adapt to evolving requirements, priorities, and market conditions. This flexibility is particularly valuable in industries where rapid innovation and responsiveness are crucial.
2. **Customer Centric Approach :**
   * Agile places a strong emphasis on delivering value to customers continuously. By involving customers throughout the development process and incorporating their feed back, Agile ensures that the final product aligns with customer needs and expectations.
3. **Incremental Progress :**
   * Agile divide projects into small, manageable increments or iterations, allowing for the continuous delivery of functional and valuable features. This incremental approach provides stakeholders with tangible benefits and helps manage expectation effectively.
4. **Reduced Risk of Project Failure :**
   * The iterative nature of Agile development allows me to identify and address issues early in the process. This proactive approach to risk management reduce the likelihood of major problems surfacing late in the project, contributing to overall project success
5. **Continuous Improvement :**
   * Agile promotes a culture of continuous improvement. After each iteration, I can reflect on my performance and processes, seeking ways to enhance efficiency, quality, and collaboration in subsequent iterations.
6. **Increased Transparency :**
   * Agile practices, such as regular demonstrations and open communication channels, contribute to increased transparency with me and supervisor. This transparency helps identify and address challenges more effectively.
7. **Focus on Customer Vale :**
   * Agile prioritizes delivering features that provide real value to the customer. This ensures that the development efforts are aligned with the most critical needs of the end users
8. **Adaptation to Complex Projects :**
   * Agile methodologies are well suited for complex and dynamic project where requirements are likely to change. The adaptive nature of Agile allow me to respond effectively to complexity and uncertainty

图示

描述已自动生成

## 3.3. Overview of System

My application will have both a frontend and a backend.

* **Frontend:**
  + Comprising simple HTML, CSS, and JavaScript templates.
  + The frontend will incorporate UI and UX components, along with buttons facilitating configuration and execution of functions in the backend.
* **Backend:**
  + Composed of OpenCV, Python, and Flask to establish the backend server.
  + This section will host all the functions and algorithms essential for image restoration.

## 3.X. Conclusions

# 4. Testing and Evaluation

## 4.1. Introduction

## 4.2. Plan for Testing

|  |
| --- |
| Testing Plan 1 : Restore a Faded Image |
| |  |  | | --- | --- | | Step 1.1 | Tester upload a faded Image to the Website | | Step 1.2 | Tester select the “Faded” function | | Step 1.3 | Tester able to see a restored Image show on screen | | Step 1.4 | Tester kick the save button to save the restored image | | Step 1.5 | Tester able to select and store the image in .jpg , .jpeg or .png | | Step 1.6 | Tester able to open the new image in their device | |

|  |
| --- |
| Testing Plan 1 : Restore a scuffs , Tears or Rips Image |
| |  |  | | --- | --- | | Step 1.1 | Tester upload a scuffs Image to the Website | | Step 1.2 | Tester select the “Scuffs” function | | Step 1.3 | Tester able to see a restored Image show on screen | | Step 1.4 | Tester kick the save button to save the restored image | | Step 1.5 | Tester able to select and store the image in .jpg , .jpeg or .png | | Step 1.6 | Tester able to open the new image in their device | |

## 4.3. Plan for Evaluation

Evaluation Form

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Question | 1 | 2 | 3 | 4 | 5 |
|  | Does this website able to restore your image? |  |  |  |  |  |
|  | How is the faded function work? |  |  |  |  |  |
|  | How is the scuffs function work? |  |  |  |  |  |
|  | How is the colour swift function work? |  |  |  |  |  |
|  | How is the spots function work? |  |  |  |  |  |
|  | How is the water damaged function work? |  |  |  |  |  |
|  | Does this website fast enough? |  |  |  |  |  |
|  | How is the UI of this website |  |  |  |  |  |
|  | What is your using experience for this website? |  |  |  |  |  |
|  | Will you use it again? |  |  |  |  |  |
|  | Will you like to let your friend to use it? |  |  |  |  |  |

## 4.4. Conclusions

# 5. Prototype Development

**As least 2 pages, but as many as you like (but lots of code samples).**

## 5.1. Introduction

## 5.2. Prototype Development

## 5.4. Conclusions

# 6. Issues and Future Work

## 6.1. Introduction

## 6.2. Issues and Risks

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk No** | **Risk Item** | **Risk Level** | **Risk Frequency** |
| **1** | **The Core (Restore) function not work** | **Very high** | **Very Low** |
| **2** | **Not able Connect Backend** | **Very high** | **Very Low** |
| **3** | **Processing too long time** | **Medium** | **Medium** |
| **4** | **Bad UI desgin** | **Low** | **High** |
| **5** | **Bad UX design** | **Mediu** | **High** |

|  |  |  |
| --- | --- | --- |
| Risk No | Explain | Solution |
| 1. | These functions are the idea for this application. Without these, the application not able to implement. | Make sure one of these functions able to work before start to develop the application |
| 2. | As this is a Web based application. All of these functions required backend to success. | Study and setup the server after the one of the core functions succussed |
| 3. | This may happened cause the core functions use too many resources. | Need to find or evaluate the functions code. |
| 4. | Too much button and colour contrast may cause User feel bad with the website | Using light colour for the background colour and reduce the unnecessary button display on screen at the same time |
| 5. | Too many button and unclearly . Which make user not able to find the functions or materials that they need | Reduce the number of button and using simple word to allow user can easily to use it |

## 6.3. Plans and Future Work

For now, I already have two core functions able to show this application is doable and setup a simple server with flask. The important part for next work will be allow this website able to run with at least one function in the website. This is the main goal of this application. As it able to show this application already have to minimum functions able to work.

Next, I will like to develop all of other core functions in editor and make sure they able to work well and transplant these functions in my website.

Finally, I will work with my Testing plan and make this website have better UI and UX.

### 6.3.1. GANTT Chart

# Bibliography