TED University

Department of Computer Engineering

Senior Design Project

Low-Level Design Report

Group Members:

- Umay ŞAMLI [52255193822]
- Rümeysa ÖZ [10216424224]
- İhsan Melih ŞİŞMAN [56725508390]
- Fatih Mehmet YİĞİTEL [38651154902]

Supervisor:

- Yücel ÇİMTAY

Jury Members:

- Gökçe Nur YILMAZ
- Saiful ISLAM

Low-Level Design Report
March 26, 2023

The Low-Level Design document graduation project course is offered under CMPE492.

Table of Contents

1. Introduction ... 3

1.1 Object Design Trade-Offs ... 3

1.1.1.1 Memory vs Performance ... 3

1.1.1.2 Usability vs Functionality ... 3

1.1.1.3 Object-Oriented Programming vs Development Time ... 3

1.1.1.4 Quality Assurance vs Time ... 3

1.1.1.5 Security vs Performance ... 3

1.1.1.6 Data Reliability vs Maintenance Time ... 3

1.2 Interface Documentation Guidelines ... 4

1.3. Engineering standards ... 4

1.4. Definitions, Acronyms, and Abbreviations ... 5

2. Packages ... 6

2.1 Client ... 7

2.2 Back-End ... 7

3. Class Interfaces ... 8

3.1 Client ... 8

3.2 Back-end ... 9

4. Glossary ... 10

5. References ... 10

1. Introduction

1.1 Object Design Trade-Offs

1.1.1.1 Memory vs Performance

The memory usage of the system is almost zero because the data is not kept in the system continuously. This ensures that memory usage has minimal impact on the performance of the system. System performance depends entirely on how well the artificial intelligence algorithm works. When a new image is uploaded to the application, the old image is deleted from the system and the new image is uploaded. This does not cause unnecessary memory usage in the system.

1.1.1.2 Usability vs Functionality

The system is very easy in terms of usability. There are a total of 5 buttons and 1 user input form on a single home screen. The user can easily do where and what to do by following the explanations of the buttons. The user can use the facilities offered to him in the system without any bugs or loss of functionality.

1.1.1.3 Object-Oriented Programming vs Development Time

While coding the functions in the application, they are divided into sub-parts and functions as in real life. Object-Oriented programming was used to reflect this more easily in the code. While coding the application, the time-consuming part is to apply artificial intelligence algorithms efficiently and to train artificial intelligence with data after application.

1.1.1.4 Quality Assurance vs Time

Since the system does not contain too many functions, quality control can be done easily and quickly in the main functions.

1.1.1.5 Security vs Performance

Since the system does not have any user login, it was not deemed necessary to take any security measures.

1.1.1.6 Data Reliability vs Maintenance Time

Since the system does not store any data in memory, maintenance time is almost non-existent.

1.2 Interface Documentation Guidelines

All class names are used in the singular throughout this report. They are all called "ClassName." The convention for attributes (variables) and methods is "variableName," "methodName()”. The order of the class name, characteristics, and methods is established in the class descriptions. Below is a thorough summary:

ClassName Description of Class Attributes

Attribute Name

Type of Attribute Methods

Method name

Parameters

Return Value

1.3. Engineering standards

The class interfaces, diagrams, scenarios and use cases, subsystem compositions, and representation of hardware-software components are all described using UML (Unified Modeling Language) design concepts, as in the prior reports. We picked UML because of its ease of use and popularity. The project's design adheres to IEEE engineering standards.

1.4. Definitions, Acronyms, and Abbreviations

- **Dehaze Net’s Capsule Networks:** Capsule Neural Networks are a form of artificial neural network that may be used to better describe hierarchical connections. The method is an attempt to recreate organic neuron architecture more accurately. It is used in this project to remove fog.
- **Client:** A desktop computer or workstation that can access information and programs from a server on a network.
- **Dehazing:** This is a somewhat distinct phenomenon in which extremely minute, dry particles are suspended in the air rather than water droplets. These particles are undetectable to the human eye, yet they contribute to the opalescent look of the air. Dehazing is the process of removing this haze from the video or camera.
- **Defogging:** The term fog refers to the resultant visibility of less than 1 km. However, in public predictions, this often refers to visibility less than 180 m. Defogging is the process of removing this fog from the video or camera.
- **Wavelet:** A wavelet is an oscillation with a wave-like amplitude that starts at zero, grows or decreases, and then returns to zero, one or more times.
- **Homomorphic Filtering:** Homomorphic filtering is an extended image and signal processing approach that involves a nonlinear mapping to a separate domain where linear filter approaches are used, followed by mapping to the original domain.
- **Atmospheric Light Scattering:** Scattering happens when particles or big gas molecules in the atmosphere contact electromagnetic radiation, causing them to be deflected from their intended course.
- **Retinex Approach:** The Retinex algorithm is used to extract light from reflectance in a picture.
- **Single-Scale Retinex:** Retinex on a Single Scale Using Digital Signal Processors. The Retinex image enhancement algorithm enhances a picture's brightness, contrast, and sharpness.
- **Multi-Scale Retinex:** The MSR (Multi-Scale Retinex) is a colored picture improvement system based on a human perception that consistently offers color constancy and dynamic range compression.
- **Dynamic Range Compression:** This method describes an image correction processing technique that involves changing the area image intensity values of an image using a dynamic range compression image transform.

- **Inverse Filtering:** Inverse filtering is the process of obtaining a system's input from its output. Once the degradation function is known, this is the easiest method for restoring the original image.
- **Degradation Pattern:** It refers to the degradation of visual quality caused by a variety of reasons. When there is picture degeneration, the image quality suffers significantly and becomes hazy. The practice of repairing or enhancing the quality of an image using image editing software is known as image restoration.
- **Wavelet Transforms:** Wavelet transforms are mathematical tools for analyzing data where features vary over different scales. Signals can have frequencies, properties, transitions that change over time, or slow-changing trends. For images, features include edges and textures.

2. Packages

In our project, we just have the client and back-end packages.

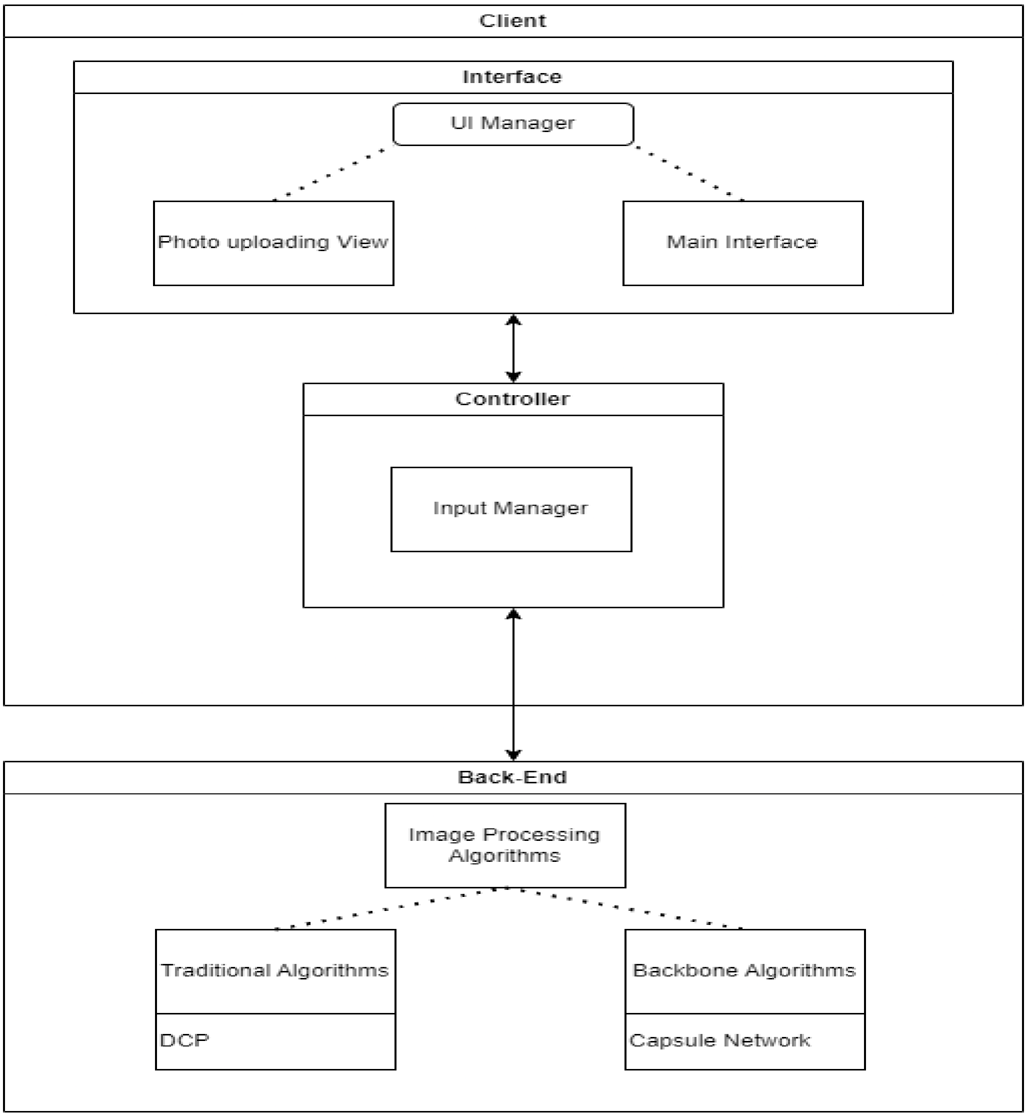

Figure 1: The System Design

2.1 Client

In the client part, we are having the photo from the user and convert the input data to a convertible image file (if it is possible). Otherwise, we gave an error about importing a document into the system.

- **Interface**

- **Photo Uploading View:** The class-specific class will give a client the ability to choose the specific jpeg/PNG file that he wants to upload.
- **Main Interface:** The specific class will be responsible for the main view of the program.

- **Controller**

The part where we organize the inputs that come from the client. We look at whether is uploaded file is acceptable. If it is not, it will give an error to the user. The controller will also be responsible for having the output from the back-end part.

- **Input Handler:** It looks at the type of file that is uploaded. If it is a proper input, then we send it to the back-end part to process the image. The input handler will be responsible to manage the outputs of the back-end part.

2.2 Back-End

At back, end part, we will dehaze the algorithm by the chosen algorithm. Firstly, we will choose the algorithm type that we will use. Then we will choose the specific algorithm from the pool of chosen algorithm types.

- **Traditional Algorithms:** the pool of Classic dehazing algorithms that we can find commonly in use.
- **Backbone Algorithms:** Pool of new Dehazing algorithms that we implement.

3. Class Interfaces

3.1 Client

Class Interface

The interface class is responsible for interface control and managing inputs from the user.

Attributes

- Private image foggy image
- Private image non-foggy image

Methods

- Def ___init____(self): Starts the program from the front-end part.
- Def dehazeWithDCP(image): Sends an image to the DCP algorithm.
- Def dehazeWithCapsuleNetwork(image): Sends an image to the capsule network algorithm.

Getters and Setters

Class Controller

This class is responsible to control the inputs given by the user.

Attributes

- Private image theDownloadedFile
- Private boolean isFileValid

Methods

- Def document_controller(image): Controls the that uploaded by the user. If the file is not valid then it returns false. If the input is valid, then it returns true.

Getters and Setters

3.2 Back-end

Class image_Processing_Algorithms
Class includes all the algorithms that we will use in the Project.
Attributes
- Private image foggyImage - Private image non-FoggyImage
Methods
- Def DCP(image): Returns the dehazed image. The image will be dehazed by using the DCP algorithm. - Def CapsuleNetwork(image): Returns the dehazed image. The image will be dehazed by using the Capsule Network algorithm.
Getters and Setters

4. Glossary

❖ **DehazeNet:** It is an end-to-end system to remove haze in the images.

❖ **CNN:** CNN is a network architecture for deep learning algorithms. It is especially used for tasks that involve the processing of pixel data and recognition of images.

❖ Capsule Network: The network is trained by minimizing the Euclidean distance between the image and the output of a CNN that reconstructs the input from the output of the terminal capsules.

5. References

- Link 1
- Link 2
- Link 3
- Link 4
- Link 5
- Link 6
- Link 7
- Link 8
- Link 9