Bilkent University

Department of Computer Engineering

CS 491 - Senior Design Project I

DeePaint

High Level Design Report

Project Group Members:

Yavuz Bakman - 21703309

Hande Sena Yılmaz - 21703465

Alperen Öziş - 21703804

Zübeyir Bodur - 21702382

Duygu Nur Yaldız - 21702333

Supervisor: Uğur Doğrusöz

Jury Members: Dr. Shervin Arashloo and Dr. Hamdi Dibeklioglu

Table of Contents

1. Introduction

During the last decade, social media applications have become widely used and part of our daily lives due to the easy accessibility of mobile devices and the increasing quality of their cameras. All around the world, people take photos constantly and share them on social media. For instance, 1074 photos are uploaded on Instagram every second [1]. While sharing photos, people have an urge to get rid of the imperfections in the photo such as a stranger in the background of a selfie or a car in a beautiful scene.

However, applying these changes requires knowledge and experience in photoshop. A regular person can not achieve to remove an object from a photo in a realistic way. Nevertheless, this became a need for regular people on a daily basis. Therefore, applications that make these adjustments automatically have gained more and more demand. Those applications with a user-friendly interface, fast modification speed, and low error rate in modification (i.e. more realistic results) are preferred by the users.

Considering the demand for easy and accurate photoshop applications, we came up with the DeePaint mobile application. In this report, we provide an overview of the architecture and design of the application, as well as the purpose of the system, design goals, consideration of various factors in engineering design, and teamwork details.

1.1. Purpose of the System

The DeePaint mobile application aims to make photoshopping more accessible and easier to use for all kinds of users, who want to edit their photograph, for use cases ranging from content creation to simple editing.

The application achieves this mostly through the following features:

- Image segmentation
- Image blending
- Image filling, when some parts of the image are removed

In fact, the application also has the following features which aid that purpose:

- Image cropping, figure moving and rotating
- Image resolution enhancing

Another purpose of the application is that those operations are done efficiently and error-prone. Since those features take time to execute on an average CPU, the application aims to solve this problem by using AWS Cloud for efficiency. Moreover, the application aims to be error-prone, so that edge cases for image segmentation, blending, and figure removal are also considered, for the best user experience. Last but not least, the application aims to be user-friendly so that varying users can also utilize photoshopping with ease.

1.2. Design Goals

1.2.1. Extensibility

The system should:

- be easy to maintain, in other words, open to updates when necessary.
- be available on most of the Android versions (4.1 and above) so that it can run almost all of the Android devices.

1.2.2. Reliability

The system should:

- undo any changes that are not yet processed in case of failure.
- generate a realistic, hence reliable, output. For this, we will use reliable and proven open-source libraries. For example, for smart removal, we will use LaMa impainting with Fourier Convolutions [2] repository, which is a reliable one for image inpainting.

1.2.3. Usability

The system should:

- be self-explanatory and user-friendly.
- be clear in terms of display and language when prompting.
- present a neat and well-organized user interface with themes that the user can select.
- not require prior knowledge from users.

1.2.4. Accessibility

The system should:

- be downloadable for free.
- be available in GooglePlay Store.

1.2.5. Efficiency

The system should:

- not lag when communicating with the server. In case of a lag, the process should be canceled if the lag is longer than 10 seconds.
- not delay much in the mobile version when receiving touch-input from the user.
- not wait longer than 5 seconds for the neural network to generate the output.

1.3. Definitions, Acronyms, and Abbreviations

API

Application Programming Interface. The connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software.

AWS

Amazon Web Services. Reliable, scalable, and inexpensive cloud computing services provided by Amazon. Remote servers can be rented through this service.

CPU

Central Processing Unit. The main processor or just processor is the electronic circuitry that executes instructions comprising a computer program.

FAQ

Frequently asked questions. Reliable, scalable, and inexpensive cloud computing services provided by Amazon. Remote servers can be rented through this service.

GUI

Graphical User Interface. An interface through which a user interacts with electronic devices such as computers, hand-held devices, and other appliances.

HTTP

Hypertext Transfer Protocol. HTTP is the protocol used to transfer data over the web. It is part of the Internet protocol suite and defines commands and services used for transmitting webpage data.

2. Current Software Architecture

The task of manipulating an image can of course be performed via traditional photoshop tools such as Adobe which requires knowledge and is especially hard to use for mobile devices. For automating this task, there are many hot papers coming out and frameworks that a programmer can use, but there are not many commercial applications that utilize these latest technologies. The ones who do still require the user to manually configure some inputs like specifying the boundary of an object to remove which is a pretty challenging task to do on a mobile device.

3. Proposed Software Architecture

3.1. Overview

DeePaint is a photoshop mobile application and allows users to edit their photographs with AI power. It brings together different machine learning models and tasks in one application. The goal of the project is to minimize human interaction in the process of photo editing/shop. Meaning that users will only prompt a high level of instructions to the application to edit/shop their photographs.

DeePaint provides various functions to its users. Detecting and doing and instance segmentation of the figures in an image, our app will allow users to remove, resize, rotate, etc. these figures. They will also be able to add new figures which are exported from other images. We will utilize the latest advanced AI papers to find a user-friendly solution to such a prevalent problem.

DeePaint application adopts client-server architecture since it requires running machine learning applications in strong machines. The details of the proposed software architecture are explained in the following sections.

3.2. Subsystem Decomposition

The DeePaint application has two main components: client and server. The Client is the part of the application responsible for displaying images, receiving the editing prompts from the user and showing the final results, etc. The Server is responsible for applying the requested changes to target images by running machine learning algorithms and returning the edited images to the client.

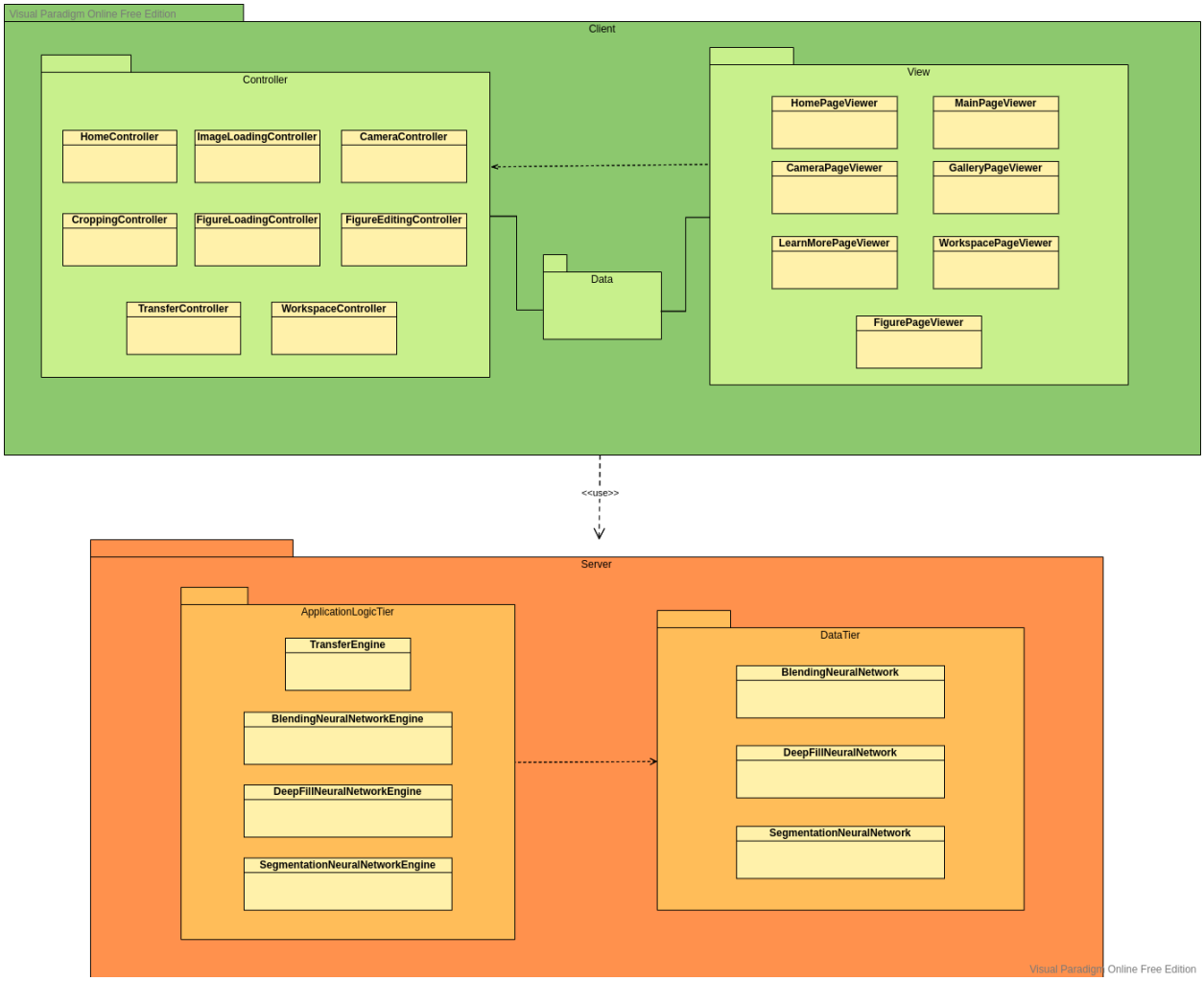

Figure 1: Subsystem Decomposition

3.3. Hardware/Software Mapping

DeePaint is a mobile application that runs on the Android operating system. Both the hardware and software are compatible with varying Android releases. However, since the

computational power of a smartphone is not strong enough to perform machine learning tasks, necessary parts of the application, namely the Server Tier, are deployed on AWS.

The communication between these two systems will be made through HTTP requests. The robustness of the AWS Cloud System will solve the computing problem mentioned above.

For example, the client-side, which is implemented in Java, will send an HTTP POST request to AWS each time the user uses a smart feature. Then, the information will be processed in the AWS, which is implemented in Python. Then, the server-side will send a response for this HTTP request, which will be basically the edited version of the image.

Below is the corresponding component diagram for this hardware/software mapping.

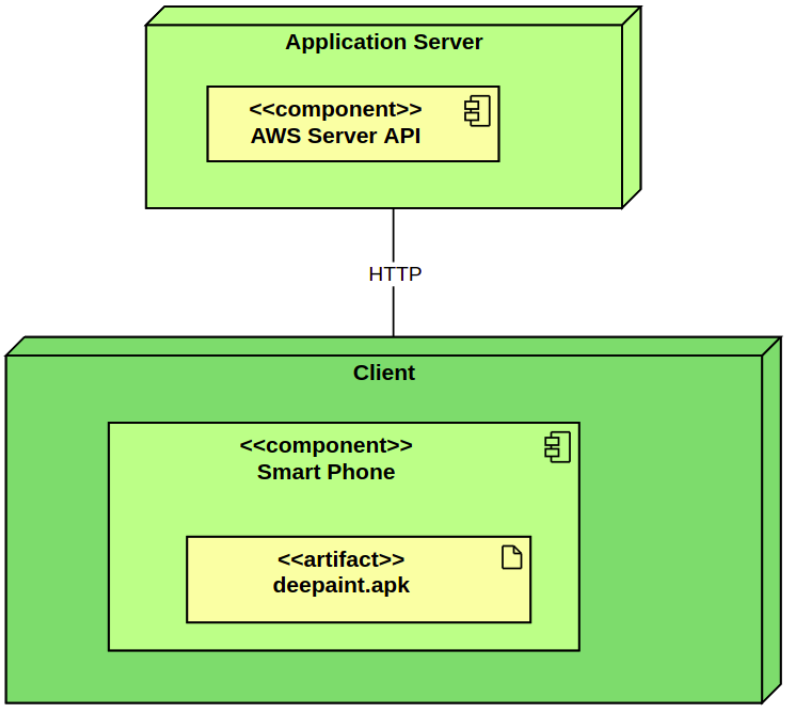

Figure 2: Hardware/Software Mapping

3.4. Persistent Data Management

On many levels, persistent data is a bottleneck in the system: the majority of the system's functionality revolves around producing or managing persistent data. As a result, data should be accessible quickly and consistently. The whole system will be sluggish if obtaining data is slow. Complete system failure is likely if data corruption is a possibility.

In our system, we want to make our server side and we care about the privacy of the users' data. Therefore, we do not store the users' image data in our server. We just send the necessary requests to the server which is differentially private and get the output which is also private.

We store the extracted figures, generated images on the local storage which is the users' phone. There is no need for a database because our system does not require complicated data structure. We only save the images and figures. There is no user id or password data which requires a database system.

3.5. Access Control and Security

Users can use DeePaint without creating an account. The images generated and figures saved by the application are stored on their local device, instead of a database.

However, in order for the users to use the application to its fullest, they need to permit the application to access the following;

- Storage
- Camera
- Gallery

3.6. Global Software Control

We'll go through how the whole system is managed on a global basis in this part. We go through how requests are made and how subsystems communicate with one another. Finally, several concurrency difficulties are addressed.

There are server and client side and totally 3 phases in our system which are request, processing and response phases.

In the request phase, the user selects the image from its local storage and open the edit page. In the edit page, the user decides the operations which are image blending, deep-fill and auto instance segmentation. After the decision, the user clicks the process button and sends the operation type, the image and how the operation is done.

In the processing phase, the server gets the necessary input from the client and utilizes the neural networks to process the input. There might be multiple requests from different users. In this case, the server has a priority algorithm to select which operation is done first.

In the response phase, the output is sent to the client from the server side. This processing phase and response phase should be fast because of the usability concerns. We want to bring concurrency to this phase, as well as the request phase, by finding an appropriate port configuration that minimizes the client-side delay.

3.7. Boundary Conditions

Boundary conditions for DeePaint can be categorized under three conditions which are initialization, termination and failure.

Initialization

Users of DeePaint need to download our app from the google play store or find an apk version. They need to give our app access to their gallery and/or their camera if they want to retrieve images from there. Since we do not require authentication and login, basic functionalities of the app can be used offline like taking a photo, creating new figures manually etc but internet connection will be needed for the image to be edited since we will be using cloud servers for that task. So when the app is first initialized, connection with the server is established if there is internet connection.

Termination

When the DeePaint gets terminated, any running process on the device and on the server will also be terminated. If there are any unsaved changes, they will be lost. Memory that is saved by DeePaint will be released after the termination.

Failure

There are several failure cases a user can experience while using DeePaint. If there is a failure while taking a photo via camera, the user will be redirected to the main page where he can either try with the camera again or choose another photo from his gallery.

If there is a failure while sending a request to the server due to no or weak internet connection, the user will be notified that there is a problem with the internet connection and will be redirected to the edit page without losing information about the photo he sent to the server.

If the request is sent to the server but there is a problem in the server and cant process the image at that moment, the user will be notified about the problem in the server and asked to try again later.

If there is no response from the server after passing a specified timeout duration, the user will be notified about the problem and asked to try again later. He can also send a report to the developers about the problem.

If there is not enough free space in the device's disk for the saving of an edited image or a figure, the user will be notified and asked to open up space in the device.

4. Subsystem Services

In this section, the subsystems of DeePaint are explained. The primary subsystems are client and server, which are also illustrated in detail.

4.1. Client

The Client subsystem has three main subsystems which are Controller, View, and Data. Details of these subsystems are explored in the following sections.

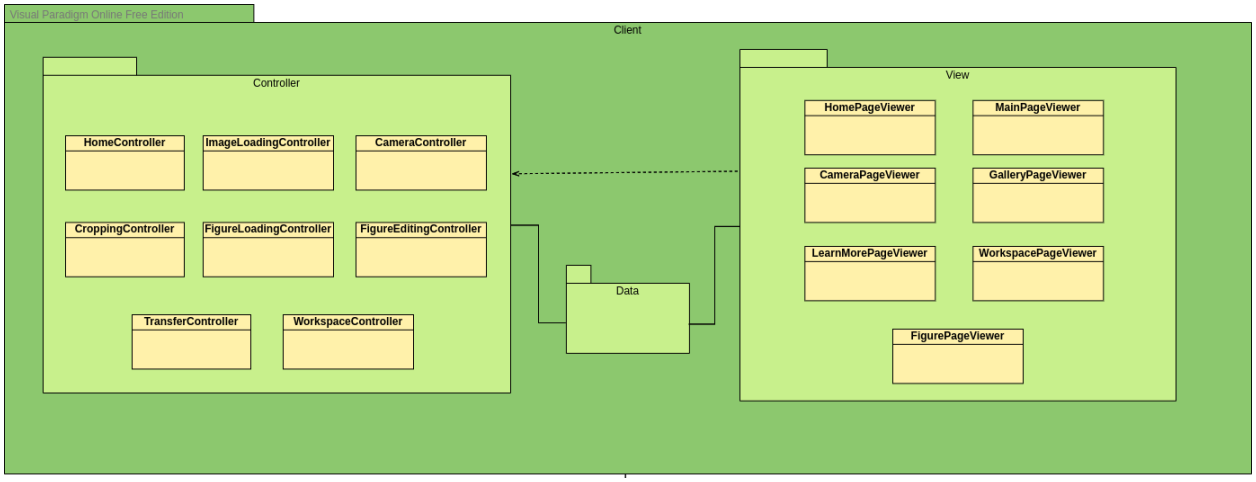

Figure 3: Client Subsystem

4.1.1. Controller Subsystem

The Controller is responsible for the communication between the client and the server.

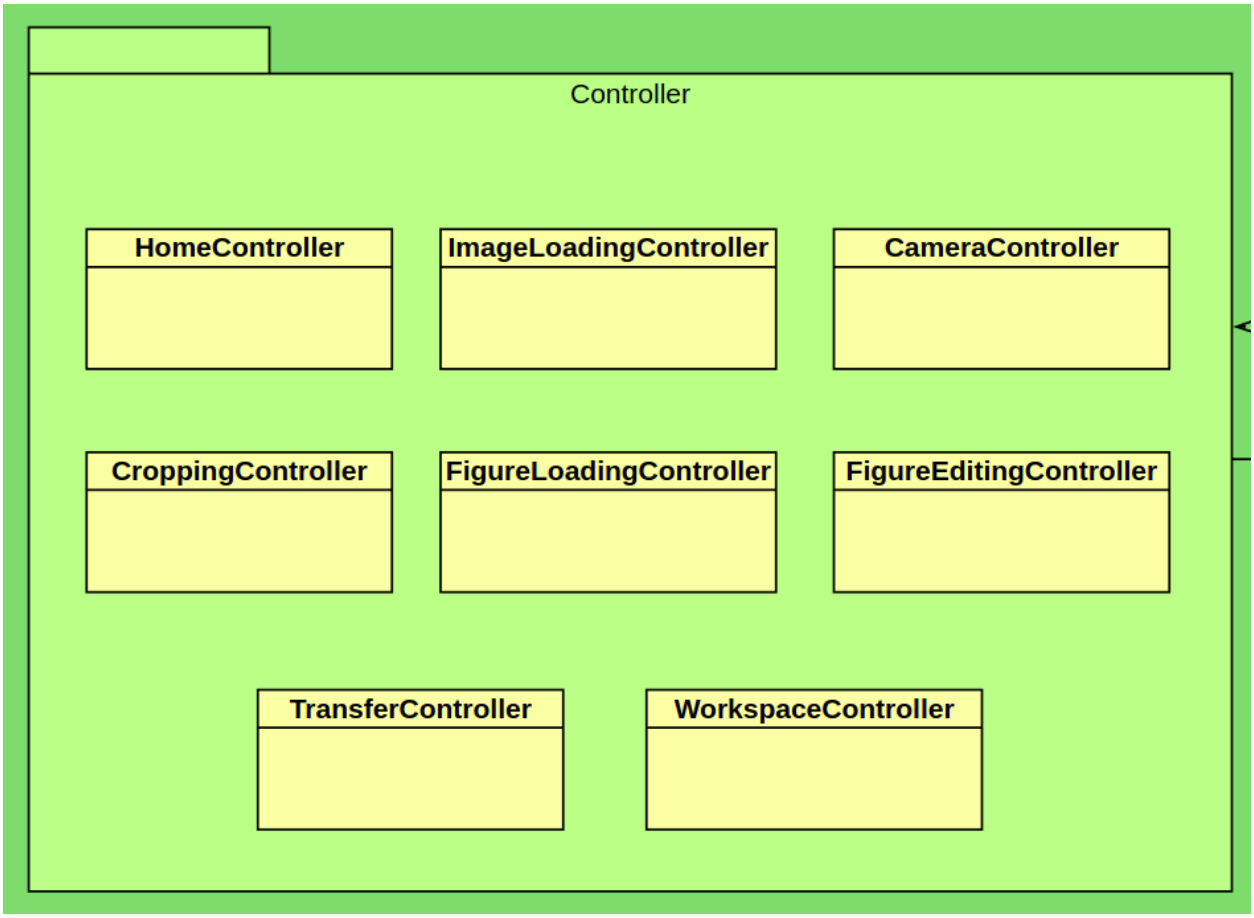

Figure 4: Controller Subsystem

HomeController

HomeController is responsible for the general flow of the application. It loads the initial requirements for the program to run the program. It also establishes the connection between the client and the server.

ImageLoadingController

ImageLoadingController is responsible for loading images from the local storage and saving edited images into the user's image gallery.

FigureLoadingController

FigureLoadingController is responsible for loading saved figures from local storage and saving extracted figures from an image to the local storage.

CameraController

CameraController is responsible for connecting to the camera and letting the user take a picture.

WorkspaceController

WorkspaceController is responsible for controlling the main functionalities of the program on the client-side such as cropping, figure editing, segmentation, etc.

FigureEditingController

FigureEditingController is responsible for the manipulation of extracted figures. This manipulation includes rotating, scaling, relocating.

CroppingController

CroppingController is responsible for extracting figures from the given coordinates. These coordinates can be given manually from the user or can be retrieved from the SegmentationNeuralNetworkEngine via TransferManager.

TransferManager

TransferManager is responsible for sending and receiving information to/from the server.

4.1.2. View Subsystem

The view subsystem is responsible for GUI operations and sending requests to the Controller subsystem.

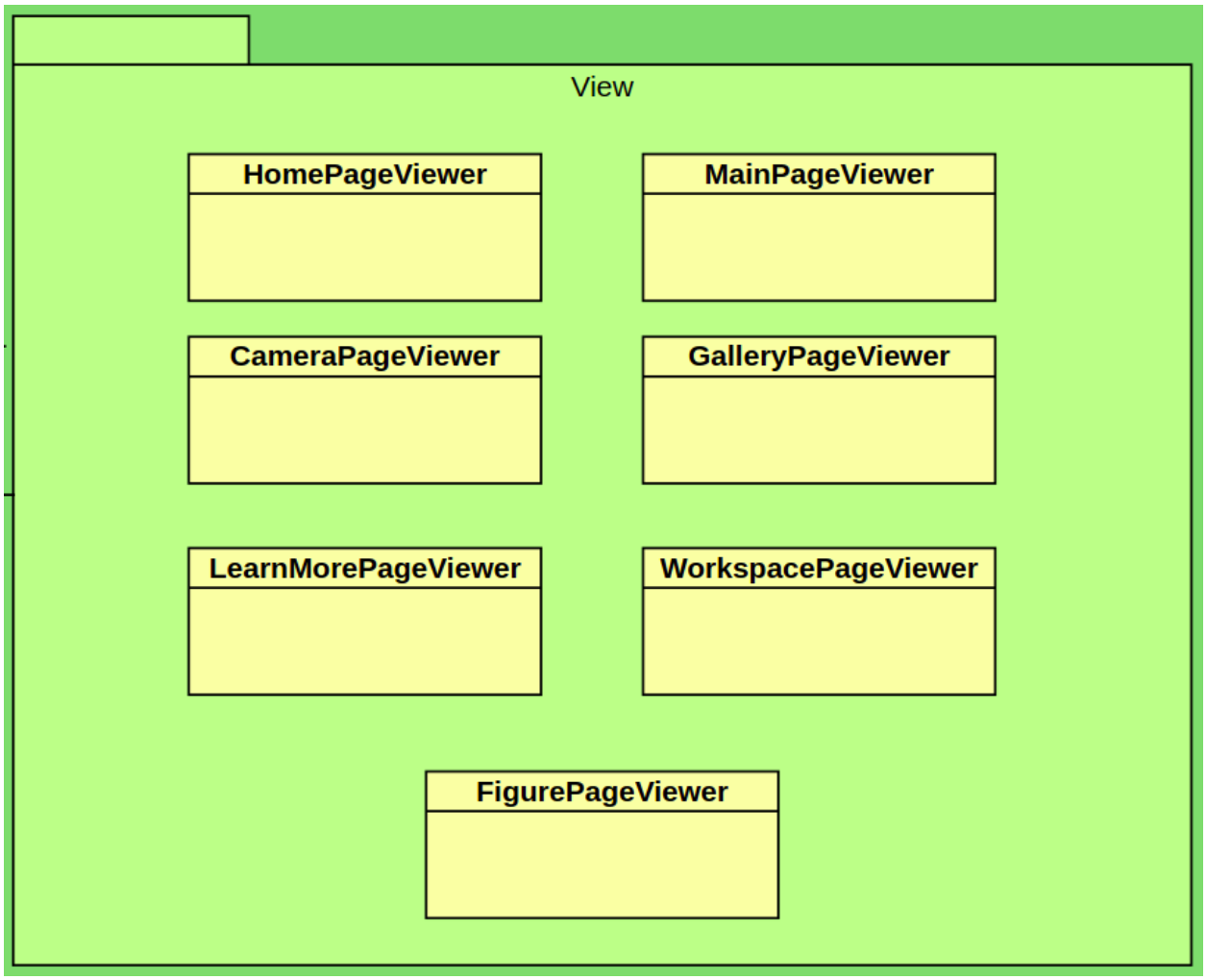

Figure 5: View Subsystem

The View subsystem consists of the following interfaces.

HomePageViewer

This view contains the GUI items for displaying the welcome page when the application is started.

MainPageViewer

This view is responsible for showing the initial actions a user can take, i.e. opening the camera, displaying saved figures, or exporting an image from the gallery.

CameraPageViewer

This view item displays the camera and allows the user to take photos, which they will be editing in the workspace view.

GalleryPageViewer

This view item displays the list of photos from the user's gallery and allows them to choose a photo to edit in the workspace view.

LearnMorePageViewer

This view item displays the buttons, "Contact Us", "FAQ" and "Team". The "Contact Us" button will redirect the user to the DeePaint website. The FAQ button will redirect the FAQ page of the DeePaint application. Finally, the Team button will display a modal that shows the names of developers who contributed to this project.

WorkspacePageViewer

The workspace page is the main page where the users will interact with the photo and edit them. There will be GUI items for, including but not limited to, displaying the saved figures, segmenting the image, manually editing the image, and saving the edited image.

FigurePageViewer

The figure page will be opened whenever the user presses the Saved Figures button on the Main Page. It will show the list of saved figures so far by the user.

4.1.3. Data Subsystem

This subsystem stands for managing local storage in which final edited pictures of the users will be stored.

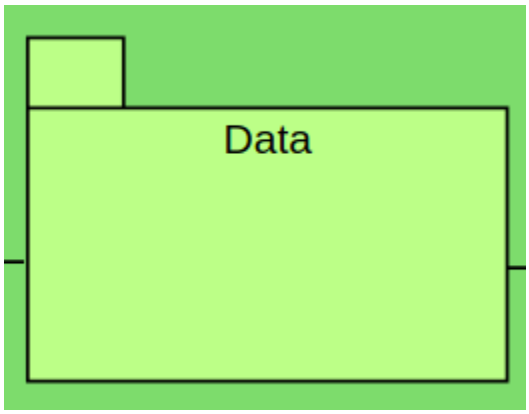

Figure 6: Data subsystem

4.2. Server

The server tier is the tier where the photo will be processed. It consists of two subsystems, the Data Tier and Application Logic Tier.

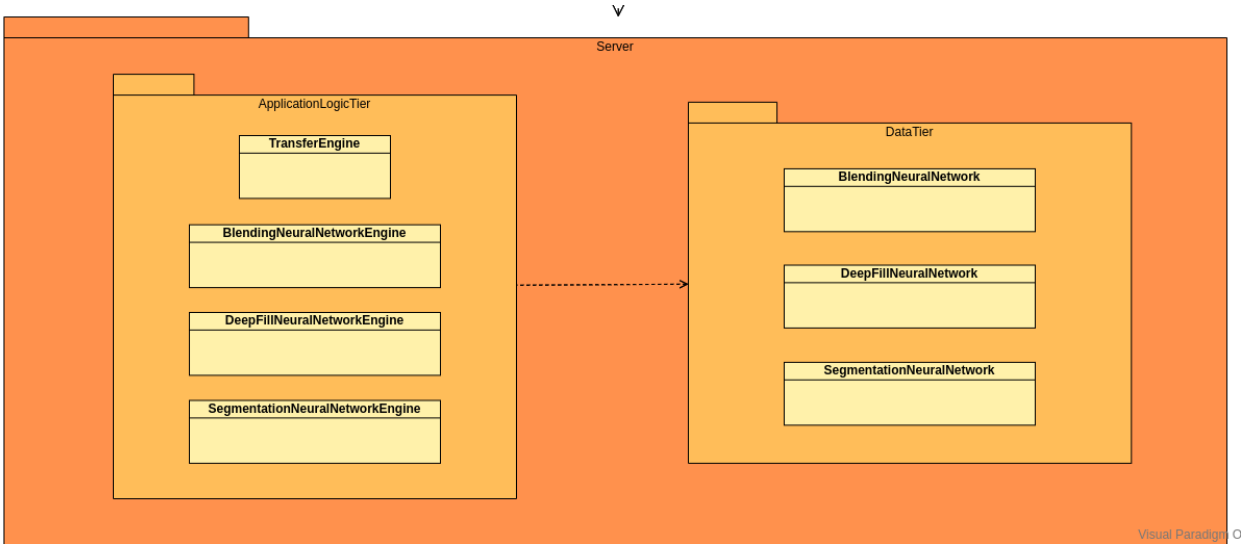

Figure 7: Server Subsystem

4.2.1. Application Logic Tier

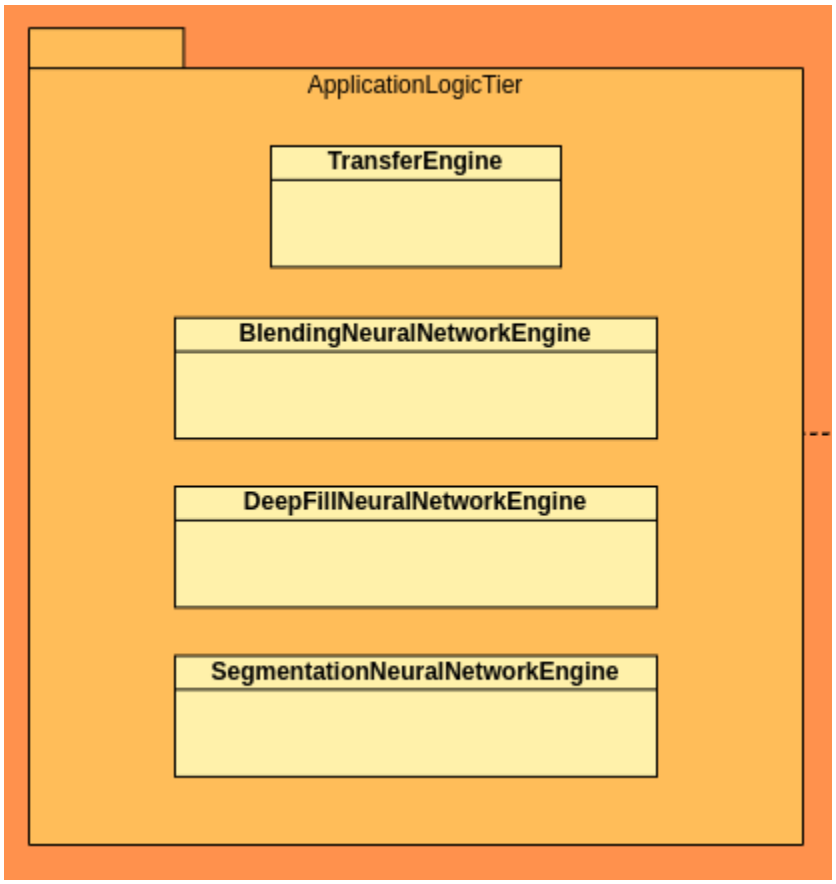

Figure 8: Application Logic Tier

The Application Logic Tier consists of the following interfaces.

TransferEngine

The Transfer Engine interface is responsible for sending and receiving images to be processed in the cloud, in AWS. After receiving the information from the cloud, it will send this information to the Controller Tier.

BlendingNeuralNetworkEngine

The Blending Neural Network Engine interface calls and uses the corresponding model, a.k.a. Blending Neural Network, with the parameters given to them by the TransferEngine.

DeepFillNeuralNetworkEngine

The Deep Fill Neural Network Engine interface calls and uses the corresponding model, a.k.a. Deep Fill Neural Network, with the parameters given to them by the TransferEngine.

SegmentationNeuralNetworkEngine

The Segmentation Neural Network Engine interface calls and uses the corresponding model, a.k.a. Segmentation Neural Network, with the parameters given to them by the TransferEngine.

4.2.2. Data Tier

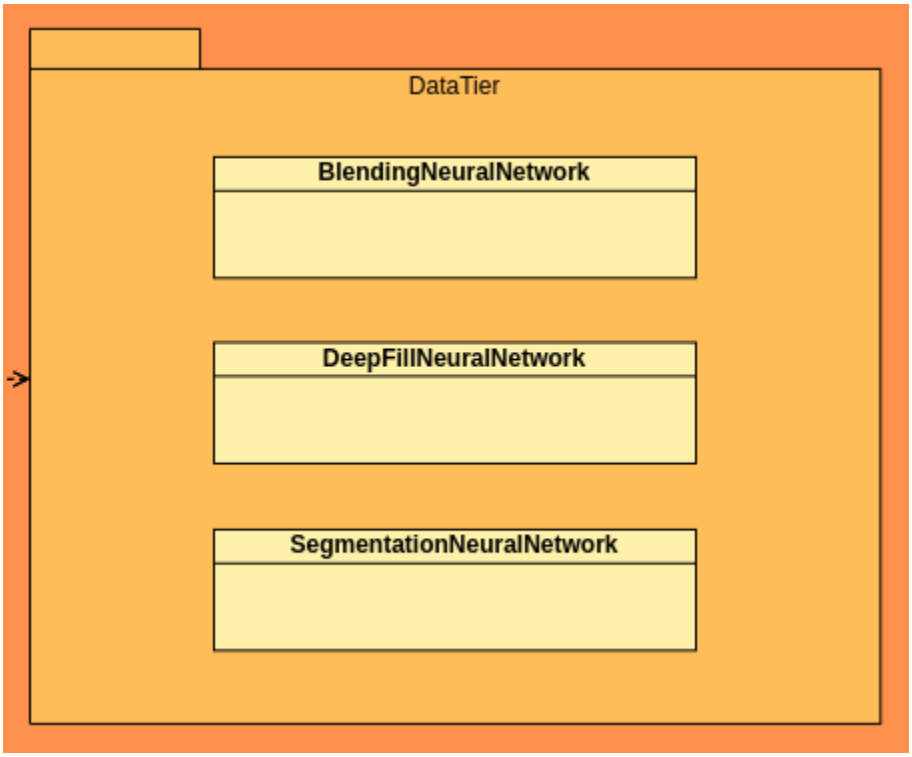

Figure 9: Data Tier

The Data Tier consists of the following interfaces.

BlendingNeuralNetwork

BlendingNeuralNetwork is the deep learning model for image blending tasks. That includes trained model architecture and parameters.

DeepFillNeuralNetwork

DeepFillNeuralNetwork is the deep learning model for deep-fill tasks. That includes trained model architecture and parameters.

SegmentationNeuralNetwork

DeepFillNeuralNetwork is the deep learning model for instance segmentation task. That includes trained model architecture and parameters.

5. Consideration of Various Factors in Engineering Design

Public Health

Our project is a mobile application so people are required to use their mobile phones and spend some time applying changes in the image. Using mobile phones excessively causes some physical problems such as neck and eye problems [3]. This is not special to our application but as the program increases the amount of time spent on mobile phones, we should consider this factor. Therefore, we should design a smooth user interface such that people can make photoshop easily and do not have to spend excessive time. However, the most important health issue we care about is psychological effects. According to the latest research, social media causes psychological problems such as anxiety and depression [4]. People compare themselves with other people and use photo-shop applications to make themselves more beautiful or handsome. They try to make their photos better to get more likes. Our program is designed for making photographs more desired. However, we do not want to connect our application to social media apps. For instance, after editing the photograph, there is no option such as ‘post it on Instagram. We want people to change their photographs because only they want. Therefore, we do not encourage users to post their edited photos on social media.

Public Safety

Our project has an interesting side effect for public safety. For instance, one can add someone into an image where crime is happening. Then, he can publish the image on the internet and spread fake news. Therefore, innocent people can be canceled by society. This is one example. As our application has the capability of generating realistic fake images, a malevolent person can abuse it. To prevent it, we want to add a mark to our generated images saying that it is generated by DeePaint. Therefore, people can easily understand whether the image is fake or not. Furthermore, as the program takes image input from the user and sends it to a server to process it, we have to care about the security of the data because it is the online environment. Therefore, we use the servers of AWS which is a well-known company that cares about the privacy and security of the users’ data.

Public Welfare

Our program is free to download. Therefore, there is no effect on public welfare.

Global Factors

Our project is designed for each person around the world. Therefore, the language of the app should be English. Also, in the future, we may put additional languages as options.

Cultural Factors

Our program has no correlation with cultural factors.

Social Factors

People generally use photo-shop applications for social media. However, we do not want to increase the amount of time spent on social media. Therefore, we do not connect our application with any social media app.

	Effect Level	Effect
Public Health	8	Physical and psychological problems
Public Safety	5	Data privacy and Fake News
Public Welfare	0	-
Global Factors	4	The program's language should be English
Cultural factors	0	-
Social Factors	3	Increase the amount of time spent on social media

Figure 10: Evaluation criteria for factors in engineering design

6. Teamwork Details

6.1. Contributing and functioning effectively on the team

When collaboration doesn't feel organic, it can seem incredibly tiresome. When practiced effectively, however, as a team we are aware that the importance of teamwork in software development, or any type of business for the matter, is paramount. Thus, we have divided the workload as effectively as possible to drive a functioning project plan.

As our application depends on neural networks and the working principle is stochastic, we first want to be sure that current neural network technology will satisfy our expectations. Therefore, we divide our team into 2 parts: people mostly dealing with neural networks and people mostly working on the application development side. However, this separation is not strict, and each member is welcome to participate in both parts.

We have divided the entire work of the project into small work packages. According to this, we have 12 packages: Analysis Report, High-Level Design Report, Implementation of Neural Networks for segmentation and Image Generation Tasks, Development of User Interface and Backend in Android, Connecting Android Application with Neural Networks for Object Removing Task Only, Demo in Desktop, Low-Level Design Report, Add Other Functionalities into Neural Networks, Add Other Functionalities into Back-end and Front-end, Connect server-application, Mobile Phone Demo, Final Report. So far, Analysis Report, High-Level Design Report, Implementation of Neural Networks, some other Functionalities of Back-end and Front-end, and Connecting server-application for Demo in Desktop have been completed.

For both Analysis and High-Level Design Report, all members are involved and contribute to different sections. For Demo in Desktop, Duygu, Yavuz, and Alperen contribute to the implementation of Neural Networks for segmentation and Image Generation. Zübeyir and Hande contribute to it in the application development side with the user interface, server connection, and more.

6.2. Helping create a collaborative and inclusive environment

Productivity, effective communication, exchange of ideas, and enhanced performance are the key qualities of a functioning project team. Moreover, this helps create unity within the team through inclusiveness and collaboration.

To achieve this, we have communicated on a weekly basis using synchronous environments. We have also communicated persistently in asynchronous environments.

The synchronous environments used are the following;

- Online meetings, through Zoom. All members have opened their cameras and were on time.
- Face-to-face meetings. Especially in the implementation phase, we have arranged face-to-face meetings to communicate and collaborate better and make sure an inclusive environment is achieved.
- Short phone calls when necessary.

The asynchronous environments used are the following;

- Messaging through WhatsApp.
- Communicating through GoogleDocs comments to share ideas while writing project documents.

6.3. Taking lead role and sharing leadership on the team

Since we have divided the entire workload into small work packages as mentioned above, each package is led by different members of the group.

Duygu Nur Yaldız: Taking the lead role in High-Level Design Report, Demo in Desktop, Low-Level Design Report, Connect server-application, and Final Report.

Zübeyir Bodur: Taking the lead role in Analysis Report, and Add Other Functionalities into Neural Networks.

Yavuz Bakman: Taking the lead role in the Implementation of Neural Networks for segmentation and Image Generation Tasks.

Hande Sena Yılmaz: Taking the lead role in the Development of User Interface and Backend in Android.

Alperen Öziş: Taking the lead role in Connecting Android Application with Neural Networks for Object Removing Task Only, Add Other Functionalities into Neural Networks, and Mobile Phone Demo.

Each member of the group integrates with the members involved in developing each work package which is led by mentioned group members and every member is welcome to share the work at any point to develop the best functioning products.

7. Glossary

Object: In our project, we refer to the word ‘object’ to imply the objects segmented in an image (e.g. human, apple, tree, carr, etc).

Editing Image: The image that the user is editing.

Neural Network: Algorithms that reflect the behavior of the human brain, allowing programs to recognize patterns [5].

8. References

[1] “Instagram by the Numbers: Stats, Demographics & Fun Facts,” *Omnicore Agency*, Jan 3, 2021 [Online]. Available: https://www.omnicoreagency.com/instagram-statistics/ [Accessed Oct 10, 2021].

[2] saic-mdal. “LaMa: Resolution-robust Large Mask Inpainting with Fourier Convolutions,” *GitHub*, Nov. 10, 2021 [Online]. Available: https://github.com/saic-mdal/lama [Accessed Dec. 21, 2021.

[3] “Signs and Symptoms of Cell Phone Addiction,” *PsychGuised*, [Online]. Available: https://www.psychguides.com/behavioral-disorders/cell-phone-addiction/signs-and-symptoms/. [Accessed Nov. 13, 2021].

[4] “Social Media and Mental Health,” *Help Guide*, [Online]. Available: https://www.helpguide.org/articles/mental-health/social-media-and-mental-health.htm. [Accessed Nov. 13, 2021].

[5] “Neural Networks,” *Intel*, [Online]. Available: https://www.ibm.com/cloud/learn/neural-networks . [Accessed Nov. 15, 2021].