Bilkent University

Department of Computer Engineering

CS 491/2 Senior Design Project

DeePaint

Analysis 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 project. In the rest of the report, analysis of the system will be given in detail. First we will describe the current system discussing their disadvantages or missing features. Then we will introduce our proposed system, giving the functional, non-functional and pseudo requirements of the project. Afterwards we will provide our system models. Possible use case scenarios will be given together with a use case diagram. After examining several scenarios, an object class diagram will be given which will construct the core of our project and allow us to actualize these scenarios. Then dynamic models will be given to better understand the flow of the program. The mock-up user interface will also be provided to give a hint on how our application will look like. Lastly an analysis of various elements will be conducted like risks and alternatives, project plan, our responsibilities etc.

2. Current System

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 apps 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 in a mobile device.

3. Proposed System

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 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.

Since our target audience is the general public and our aim is to create an easy to use app, DeePaint will be a free mobile application.

3.2. Functional Requirements

In this section, functional requirements of DeePaint are discussed in terms of both system functionality and user functionality.

3.2.1. System Functionality

- The system should ask the user's permission to reach the gallery and use camera functionality.
- The system should display the images in the gallery.
- The system should receive an image to be edited as input from the user and display it.
- The system should segment the objects in the selected image and display them.
- The system should get the information of the boundaries of the region that the user drew for an object.
- The system should display all the saved objects and allow the user add any of them to the editing image.

- The system should allow the user to delete a saved object.
- The system should get the position and size information of the objects that are located by the user on the image that is being edited.
- The system should receive the position and size information of the areas that are empty in the image (i.e object is removed).
- The system should display the edited version of the image as well as the original one.
- The system should fill the empty areas in the image in a natural way.
- The system should blend the objects to the image in a natural way.
- The system should be connected to a server.
- The system should send images to the server in order to process machine learning algorithms (e.g. segmentation, auto-fill, blending).
- The system should receive processed images from the server.
- The system should receive segmentation information from the server.
- The system should provide an option to save a processed image and should save an image to the local gallery if the user decides.
- The system should provide an option for the user to reset the changes in the editing image.

3.2.2. User Functionality

- The user can allow the DeePaint to reach the user gallery and the camera.
- The user can select a photo from the local gallery or can take a new photo and upload it to the system to edit.
- The user can view the pre-segmented objects in the image.
- The user can draw the boundaries of an object in the image.
- The user can view saved objects and delete any of them.
- The user can relocate/resize any objects that are obtained by either segmentation, drawing or loading from saved objects.
- The user can remove any object in the image that is selected by either segmentation or drawing.
- The user can save any object in the image that is obtained by either segmentation or drawing for future usage.
- The user can reset the changes in the image.
- The user can view the original version of the image as well as the edited version.

- The user can save the edited image to the local gallery.

3.3. Nonfunctional Requirements

Performance

While the user is choosing what to do with his photograph, (What to remove from the scene etc.) our program should be smooth with its reactions, namely in 100ms. Once the user has inputted his desire to our app, we should return the edited photo back to the user in at most 15 seconds.

Privacy

Since users will upload their personal photographs to our app, we must ensure every user can only access photographs uploaded by themselves. We will render this possible via basic authorization

Usability

Since one of the rationales of our project is to rescue our users from dealing with the difficulty of the traditional photoshop apps, our app should be easy to use. A new user should be able to adapt to it in 2 minutes.

Extendibility

Given the highly increasing number of innovations in AI in recent years, our design patterns should allow us to easily add a new feature to our application or improve the existing ones in terms of accuracy, usability, etc.

Accuracy

Although there is not a universal metric to measure accuracy of our tasks numerically, our app should be able to output realistic images, to the extent of tricking a FakeorNot robot.

Security

Since the actual editing of the photo will be done in the cloud servers and not in the local workspace of the user, we cannot allow bots to use our application and keep our servers busy. Therefore we will use basic authentication before allowing users to utilize our app.

3.4. Pseudo Requirements

- DeePaint mobile application will be written in Java programming language, and it will be on the Android platform.

- Since the mobile application is in the Android platform, Android Studio IDE will be used for development.
- For implementing deep learning features of the application, a server-side is required. Since those libraries, such as PyTorch, TensorFlow, Keras, Theano, are libraries that all can be used in Python, the server-side will be implemented in Python.
- Object-oriented programming will be used throughout the project.
- GitHub will be used for version control and collaboration, as well as opening issues if necessary.
- The mobile application will be free to use, and there will be no in-app purchases. Hence, all of the users will be able to use the features without making any payment.
- The application will require the user to have an internet connection since processing of the image will be carried out in the cloud.

3.5. System Models

In this section, we provide possible action scenarios of the DeePaint application, a use case model to show them visually, an object and class model, dynamic models and finally navigation paths and screen mock-ups of the DeePaint.

3.5.1. Scenarios

Scenario 1

Name: Select an Image to Edit from Gallery

Participating Actor: User

Entry Conditions:

- The application is open.

Exit Conditions:

- The user clicks on the “Select” or “Cancel” button.

Flow of Events:

1. The user selects the “Open Gallery” button on the main page.
2. DeepPaint reaches the gallery and displays it.
3. The user selects an image from the gallery or cancels the process using the “Select” or “Cancel” buttons.

Scenario 2

Name: Use Camera to Select an Image to Edit

Participating Actor:User

Entry Conditions:

- The application is open.

Exit Conditions:

- The user clicks on the “Shoot” or “Cancel” button.

Flow of Events:

1. The user selects the “Use Camera” button on the main page.
2. The user takes a photo or cancels the process by clicking on the “Shoot” or “Cancel” buttons.

Scenario 3

Name: Take a Photo Using the Camera

Participating Actor:User

Entry Conditions:

- The user is already on camera page (clicked on the “Use Camera” button)
- The user already took a photo.

Exit Conditions:

- The user clicks on the “Accept” or “Cancel” button.

Flow of Events:

1. The user either accepts the shot by clicking on the “Accept” button or discards that photo and goes to taking the photo screen again by clicking on the “Cancel” button.

Scenario 4

Name: Choose an Object on the Image from Segmented Objects

Participating Actor: User

Entry Conditions:

- The user has already selected an Image to edit.
- The user clicks on the “Segment” button.

Exit Conditions:

- The user clicks on the “OK” or “Cancel” button.

Flow of Events:

1. The user clicks on the “Segment” button.
2. DeePaint uses its algorithms to segment objects on the image and displays them.
3. The user selects desired objects.
4. The user clicks on the “OK” or “Cancel” button.

Scenario 5

Name: Use Pencil to Define Boundaries of an Object on the Image

Participating Actor: User

Entry Conditions:

- The user has already selected an Image to edit.
- The user clicks on the “Pencil” button.

Exit Conditions:

- The user clicks on the “OK” or “Cancel” button.

Flow of Events:

1. The user clicks on the “Pencil” button.
2. The user draws the boundaries of the desired region.
3. The user either accepts that region as an object by clicking on the “OK” button or cancels the process by using the “Cancel” button.

Scenario 6

Name: Load a Saved Object

Participating Actor: User

Entry Conditions:

- The user has already selected an Image to edit.

Exit Conditions:

- The user clicks on the “Select” or “Cancel” button.

Flow of Events:

1. The user clicks on the “Saved” button on the editing screen.
2. DeePaint displays the saved objects.
3. The user selects an object.
4. The user clicks on the “Select” or “Cancel” button.

Scenario 7

Name: Save an Object

Participating Actor: User

Entry Conditions:

- The user has already selected an object on the image.

Exit Conditions:

- The user clicks on the “Save” button.

Flow of Events:

1. The user clicks on an object.
2. The user clicks on the “Save” button to save the object for future uses or presses on the “Cancel” button to cancel.

Scenario 8

Name: Move Selected Object across the Image

Participating Actor: User

Entry Conditions:

- The user has already selected an object on the image.

Exit Conditions:

- The user stops doing gestures.

Flow of Events:

1. The user holds on to the object that is already put on the image or that is on the object list and drags it to a desired region of the image.

Scenario 9

Name: Resize Selected Object

Participating Actor: User

Entry Conditions:

- The user has already selected one or more objects on the image.
- Target object should already be put inside the image boundaries.

Exit Conditions:

- The user stops doing gestures.

Flow of Events:

1. The user gestures on an object using two fingers to resize the object.

Scenario 10

Name: Remove an Object

Participating Actor: User

Entry Conditions:

- The user has already selected an object on the image.

Exit Conditions:

- The user clicks on the “Remove” or “Cancel” button.

Flow of Events:

1. The user clicks on an object that is already on the selected objects.
2. The user clicks on the “Remove” or “Cancel” button.

Scenario 11

Name: Process the Image

Participating Actor: User

Entry Conditions:

- The user has already selected an Image to edit.

Exit Conditions:

- The user clicks on the “Save” or “Discard” button.

Flow of Events:

1. The user clicks on the “Process” button.
2. DeePaint fills the empty areas.
3. DeePaint puts the objects in desired places in a natural way.
4. DeePaint displays the final image.
5. The user either clicks on the “Save” or “Discard” button.

Scenario 12

Name: Cancel Image Editing

Participating Actor: User

Entry Conditions:

- The user has already selected an Image to edit.

Exit Conditions:

- The user clicks on the “Yes” or “No” button.

Flow of Events:

1. The user presses the “Cancel” button on the editing screen.
2. DeePaint asks the user if the user is sure.
3. The user clicks on the “Yes” or “No” button.

Scenario 13

Name: Reset Image

Participating Actor: User

Entry Conditions:

- The user has already selected an Image to edit.

Exit Conditions:

- The user clicks on the “Yes” or “No” button.

Flow of Events:

1. The user presses the “Reset” button on the editing screen.
2. DeePaint asks the user if the user is sure.
3. The user clicks on the “Yes” or “No” button.

Scenario 14

Name: Delete a Saved Object

Participating Actor: User

Entry Conditions:

- The application is open.

Exit Conditions:

- The user clicks on the “Remove” or “Cancel” button.

Flow of Events:

1. The user selects the “Saved Objects” button on the main page.
2. DeepPaint displays the saved objects.
3. The user selects one or more objects from the list and clicks on “Remove” or “Cancel button”.

3.5.2. Use Case Model

In this section, we provide use-case diagram of the DeePaint application (Figure 1).

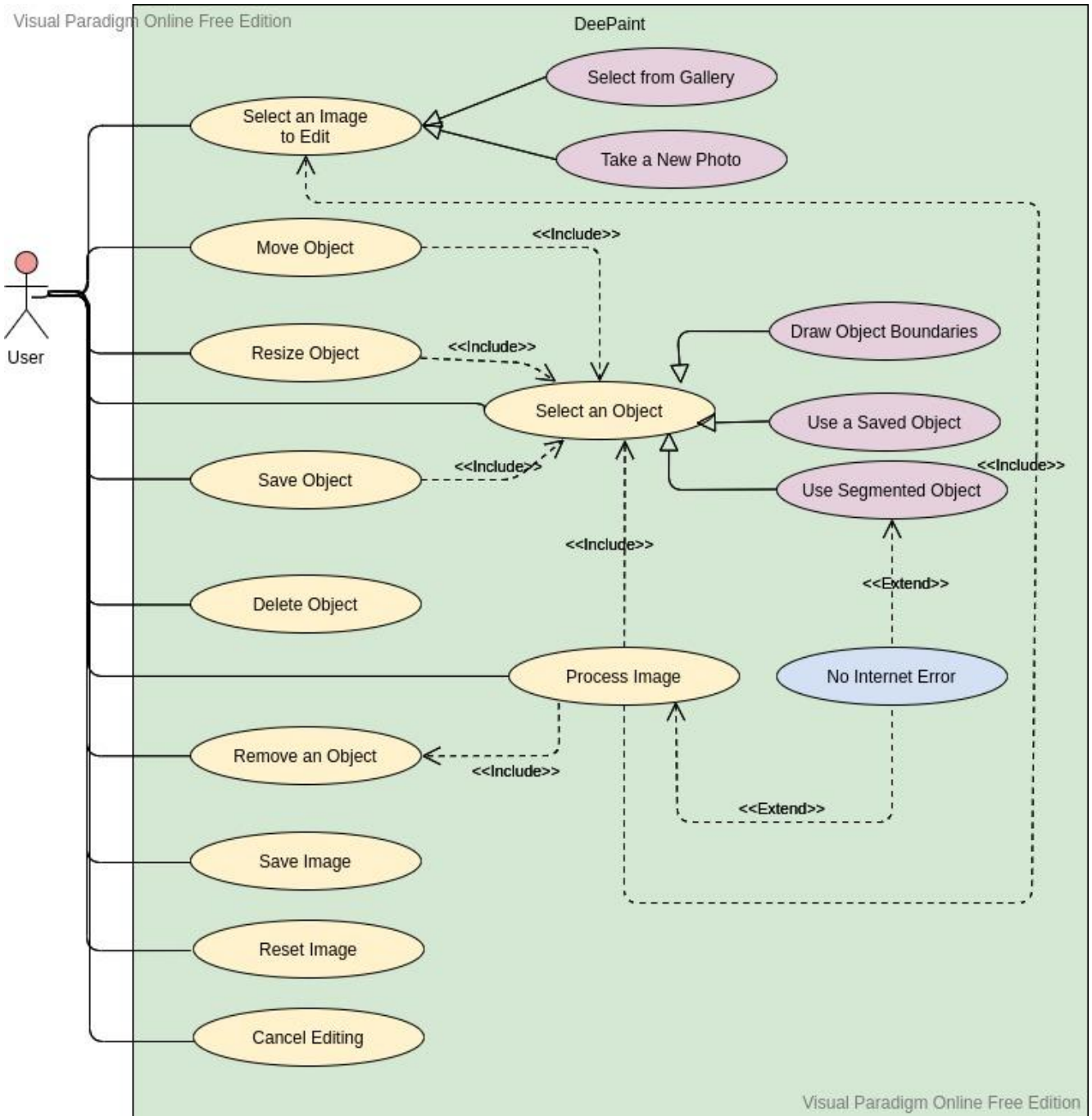

Figure 1: Use-Case Model of DeePaint

3.5.3. Object and Class Model

- **Image:** The data frame we will use to present a photo. It can present different photo extensions like .jpg or .png.
- **ImageLoaderandSaver:** The class that will handle interactions between device's storage and the application. It will allow us to import an image with different methods and save an edited image.
- **Figure:** Also an image but instead of having a rectangular boundary, a figure can have a custom boundary. If this figure is on an image, its center attribute will denote where in the image this figure is located on.
- **FigureLoaderandSaver:** The class that will allow us to save a figure we created from the image to a directory which we will be able to utilize later for different photographs.
- **EditTool:** This is the class that will handle the user interactions with the image and the figures. It will allow the user to elaborate on what he wants to change specifically. Adding or removal of a saved figure, creation of a new figure, resizing, rotating etc of an existing figure will all be handled by EditTool.
- **ImageProcessor:** This is an interface that will cover the actual work done by our application. Once the user has specified the changes he wants, functions provided by this interface will carry out the process.
- **FigureSegmentator:** A type of ImageProcessor who is responsible for detecting and doing the instance segmentation of the figures in an image. Given an image it will return the figures on it.
- **SmartRemover:** A type of ImageProcessor that is responsible for removal of the selected figures on the image and filling the spaces with reasonable pixels that will look realistic.
- **SmartIntegrator:** A type of ImageProcessor that is responsible for addition of a figure to our image. Again this class will utilize deep learning methods to make the integration of the figure look realistic.

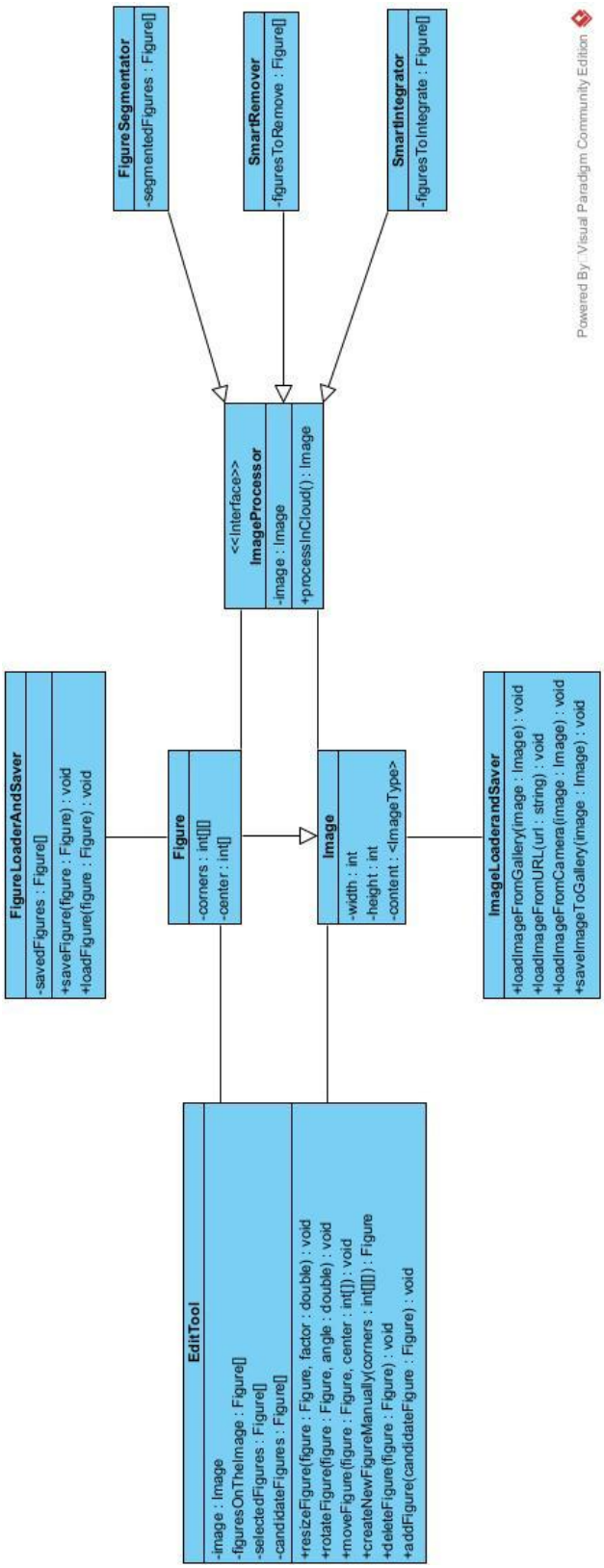

Figure 2: Object Class Diagram

3.5.4. Dynamic Models

In this section, we provide activity and sequence diagrams of the DeePaint application.

3.5.4.1. Activity Diagrams

- **Application Main flow Activity Diagram**

This diagram describes the main application flow of the DeePaint mobile app. Important details of the application is described in the next activity diagram.

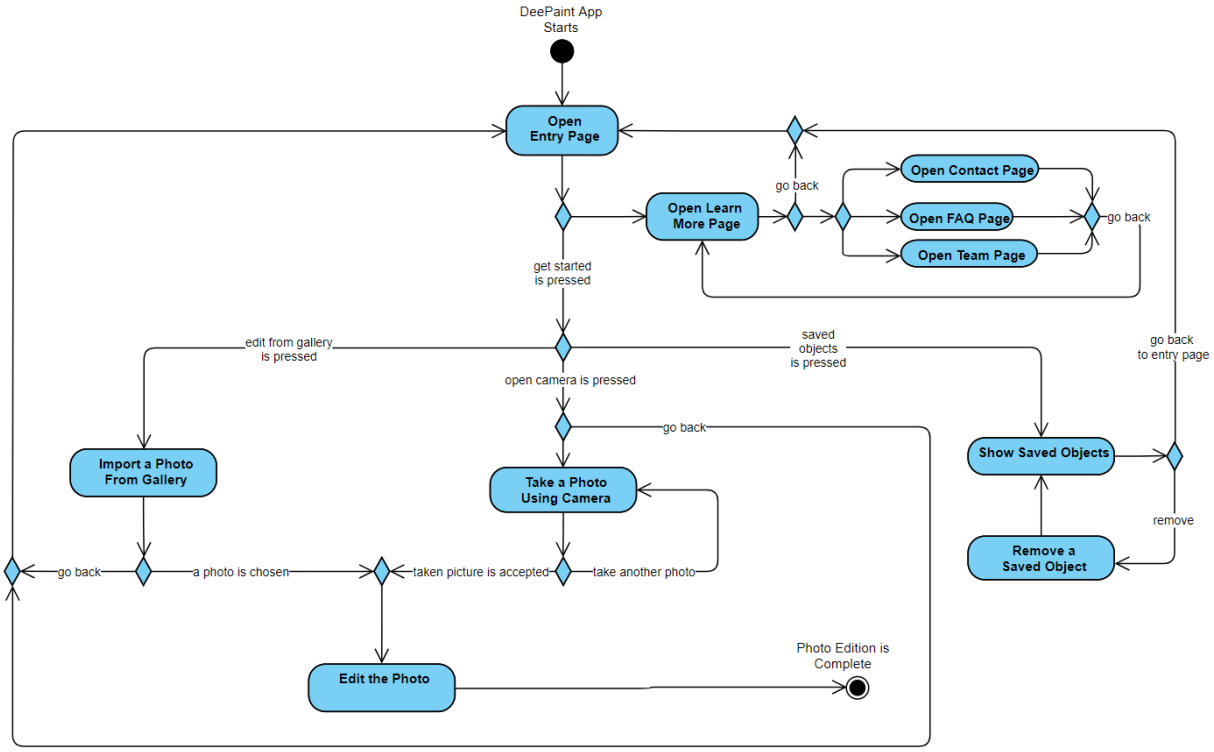

Figure 3: Main Flow activity diagram

- **Application Main flow Activity Diagram**

The following diagram shows the process by which the user can edit their photo in detail. It also shows how a user can save an object for future use in the application as well.

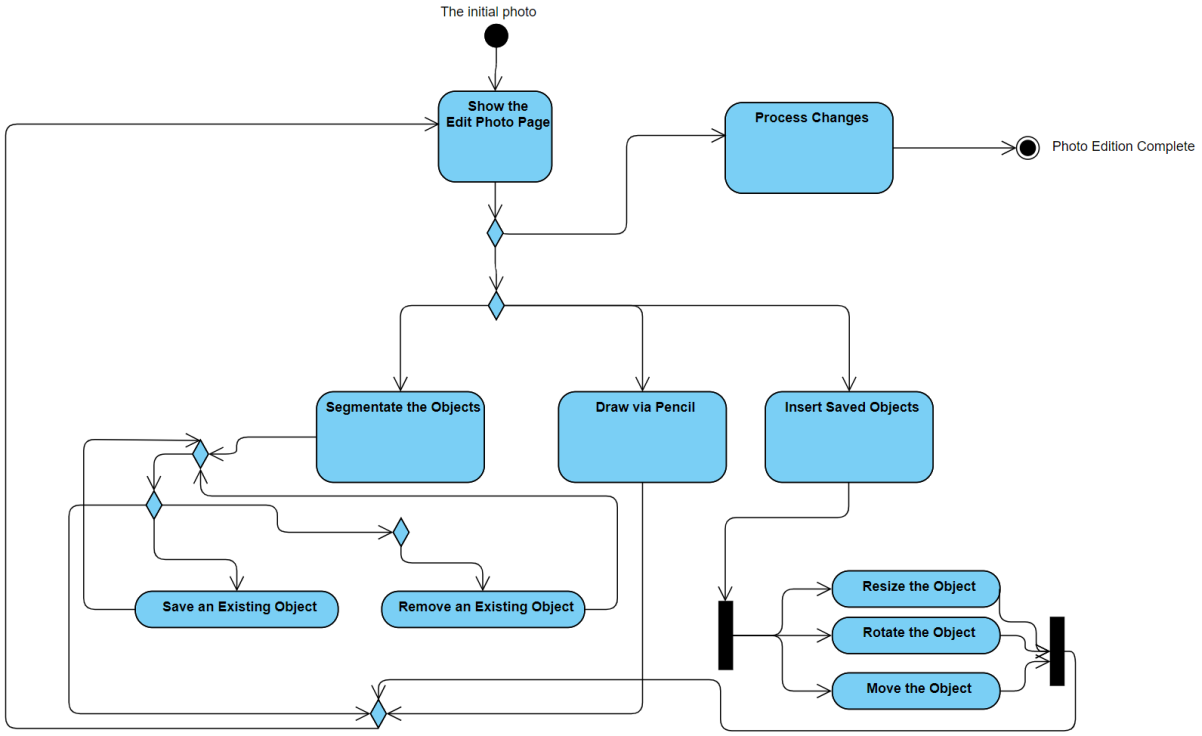

Figure 4: Editing a photo activity diagram

3.5.4.2. Sequence Diagrams

- **Removing an object within the image Sequence Diagram.**

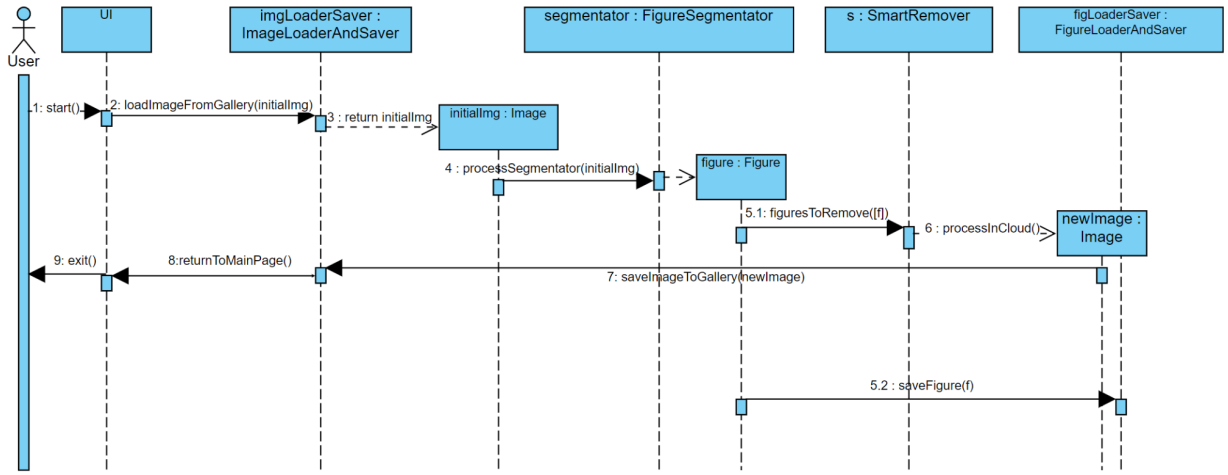

Figure 5: Removing a figure on an image

- **Replacing an object within the image Sequence Diagram.**

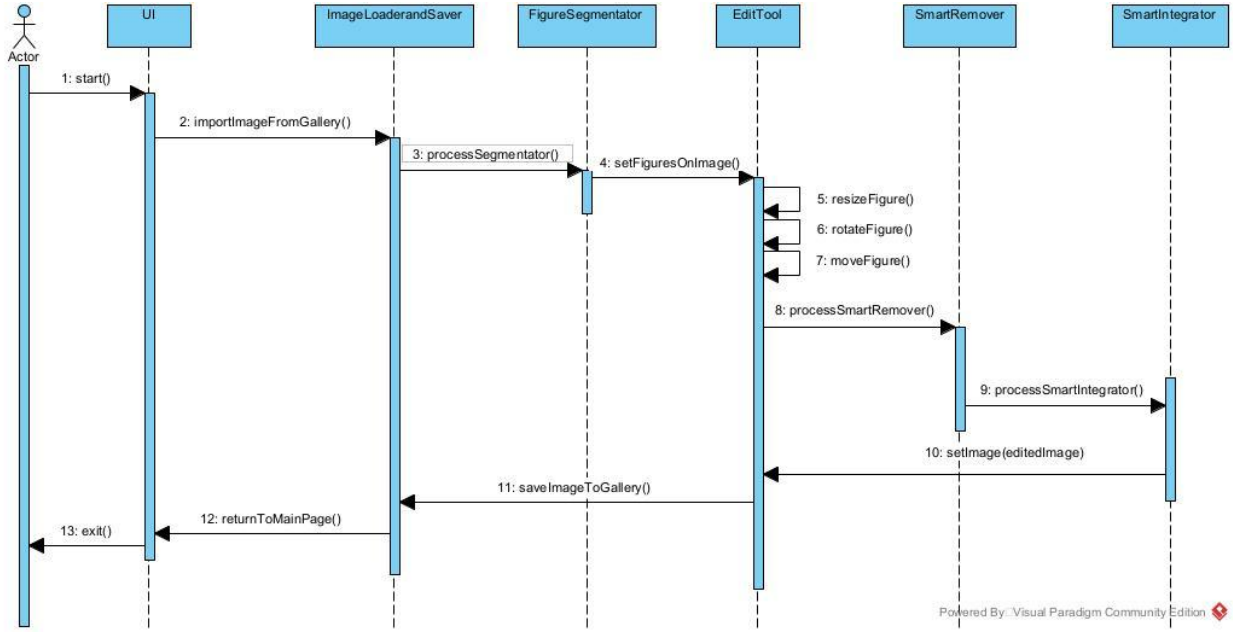

Figure 6: Replacing a figure on the image.

- **Adding an object from another image Sequence Diagram.**

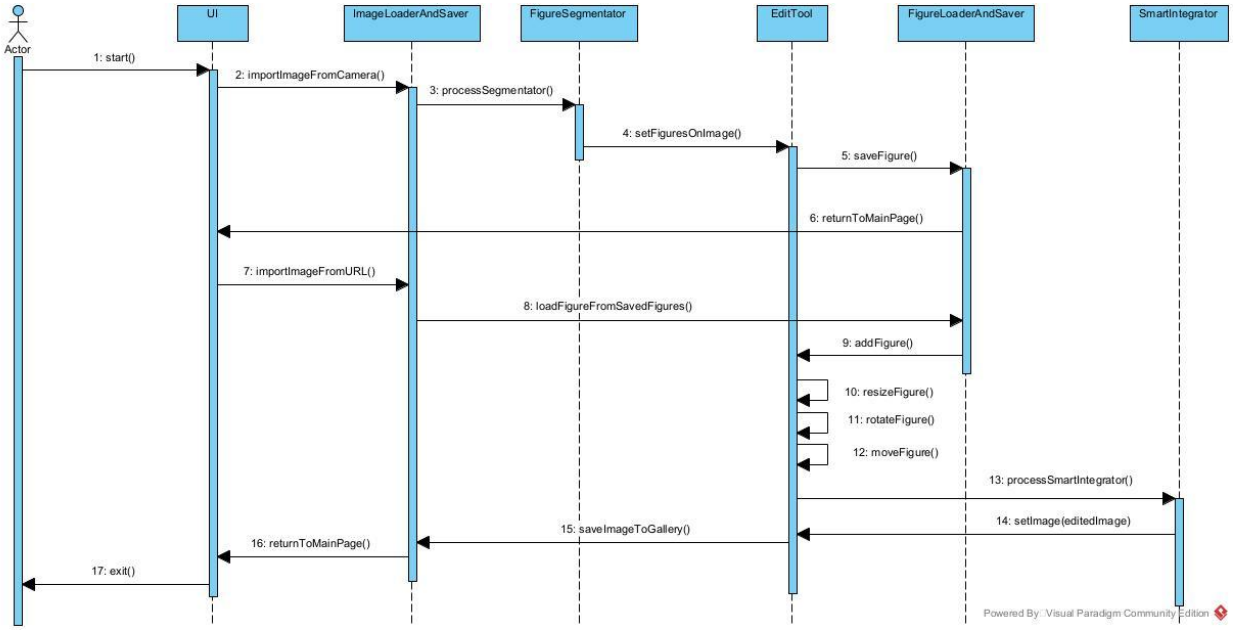

Figure 7: Adding a figure that is exported from another image.

- **Segmenting a figure manually and replacing it Sequence Diagram.**

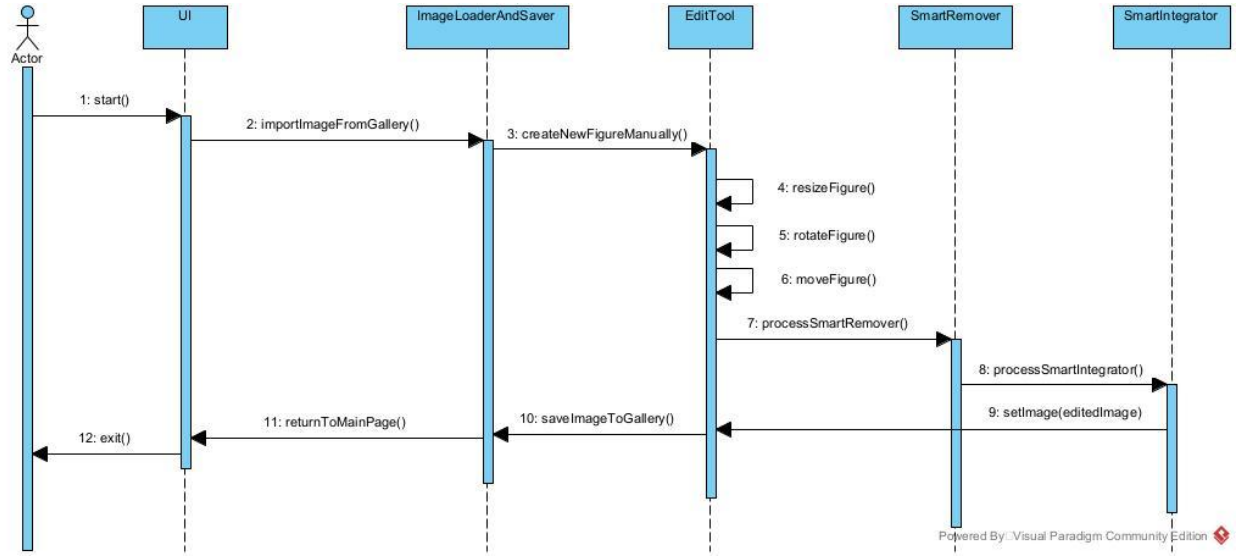

Figure 8: Segmenting a figure manually and replacing it.

- **Main Page Navigation Sequence Diagram**

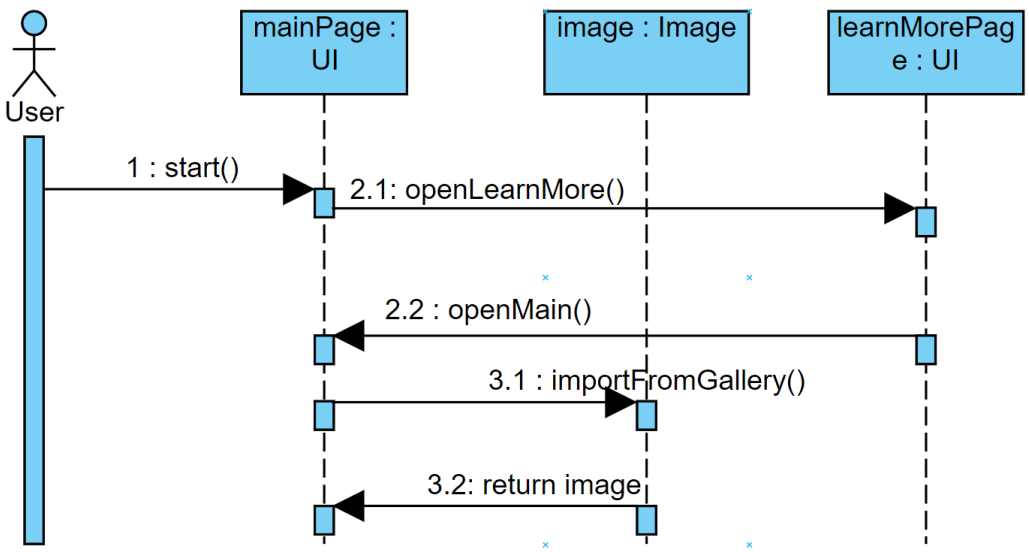

Figure 9: Main page navigation.

3.5.5. User Interface - Navigational Paths and Screen Mock-ups

In this part of the system models, the user interface of *DeePaint* is presented with navigational paths and screen mock-ups. Each mock-up is provided with an explanation to further point out the requirements and navigational paths of the application which are already mentioned above.

Entry Page

When the application is opened, the user is welcomed with *Entry Page* shown in Figure 10. The page consists of the logo of *DeePaint* and two options for the user to continue within the application. At this page, the user either can choose *Get Started* and actually start operating within the application or choose *Learn More* and navigate to *Learn More Page* where information and details of *DeePaint* reside.

Figure 10 : Entry Page

Get Started Page

In *Get Started Page* of the application, the user has three options to continue operating within the app. As Figure 11 shows, the user may tap *Edit from Gallery* button to choose an already existing photo from his/her gallery to start editing it, or tap *Open Camera* button to open the camera and take a new photo to use for editing. When the user choose the third option which is *Saved Objects*, it navigates to *Saved Objects Page* where the user can check previously saved objects and may select them accordingly for a possible removal.

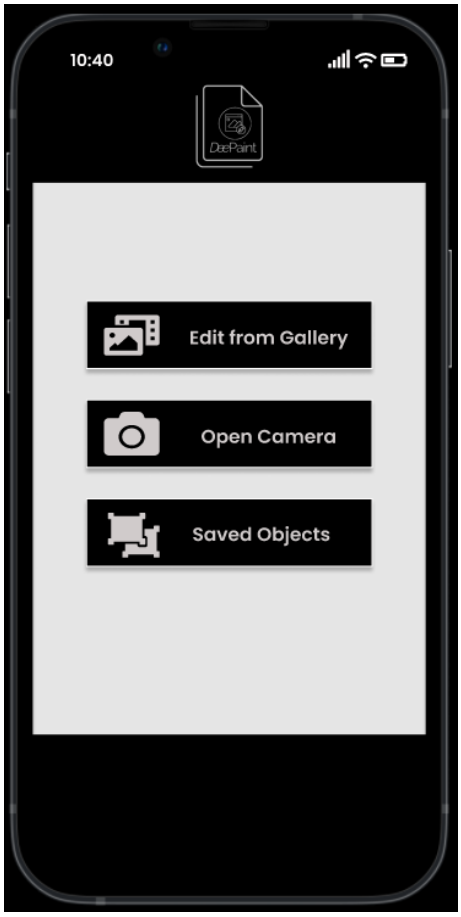

Figure 11 : Get Started Page

Choose from Gallery Page

When the user taps *Edit from Gallery* button in *Get Started Page,* it leads to *Choose from Gallery Page* shown in Figure 12. In this page, the user may select an existing photo from the gallery for editing and load it by clicking *Select* at the top-right corner to *Start Editing Photo Page* shown in Figure 15. In this page, the user may also choose *Open Camera* option by clicking the camera icon at the bottom-left corner or check *Saved Objects* by clicking its icon at the bottom-right corner. The user may cancel the process and go back to *Get Started Page* anytime by clicking the left arrow cancel icon at the top-left corner.

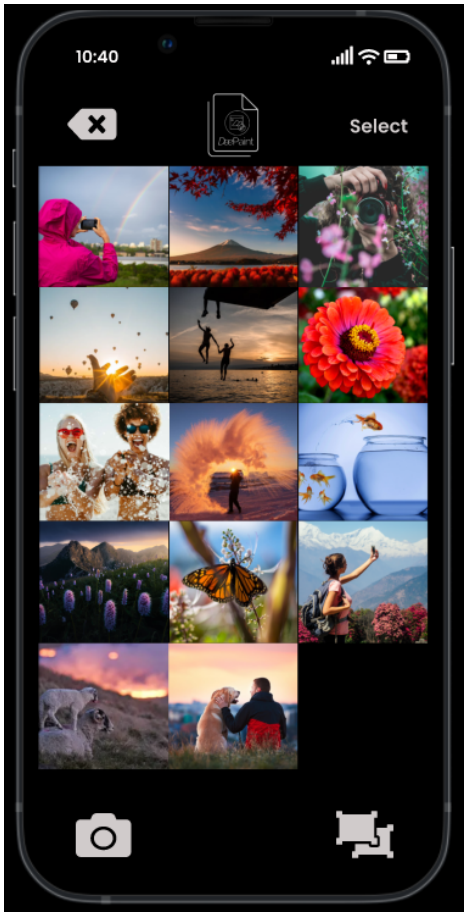

Figure 12 : Choose from Gallery Page

Open Camera Page

If the user clicks the *Open Camera* button in *Get Started Page* or camera icon button in *Choose from Gallery Page, Open Camera Page* shown in Figure 13 is loaded. In this page, the camera opens and the user is capable of taking a desired photo by clicking the circle at the bottom-center. The user may decide on front or back camera before taking the shot by clicking the flip icon at the top-right corner. In this page, the user may also choose *Edit from Gallery* option by clicking its icon at the bottom-left corner or check *Saved Objects* by clicking its icon at the bottom-right corner. As specified in the other pages, the user may also cancel the process and go back to *Get Started Page* anytime by clicking the left arrow cancel icon at the top-left corner.

Figure 13 : Open Camera Page

Load Taken Photo Page

After the user takes the photo by clicking the circle at the bottom-center of *Open Camera Page, Load Taken Photo Page* shown in Figure 14 is opened where the user decides whether to choose the taken photo for editing and continue or go back to camera to take another one. The user loads the taken photo to *Start Editing Photo Page* by tapping *Accept,* or cancels the taken photo and goes back to *Open Camera Page* by clicking the left arrow cancel icon at the top-left corner.

Figure 14 : Load Taken Photo Page

Start Editing Photo Page

When the user selects a preferred photo from the library or takes a new desired photo by opening the camera, it will be loaded to *Start Editing Photo Page* shown in Figure 15. In this page the user has three options for editing the photo that are *Saved, Pencil, and Segment*. By clicking *Saved icon at the bottom-left corner, the user is navigated to Saved Objects Page* shown in Figure 21 where previously saved objects are listed to be used for editing the photo. By clicking the *Pencil* icon located at the bottom-center, the user is navigated to *Pencil Page* shown in Figure 20 where the segmentation of an object is allowed and the user determines the borders of the object by using pencil. The last option is *Segment* in which its icon button is located at the bottom-right corner. When the user chooses this option, auto segmentation starts and it is navigated to *Segment Page* shown in Figure 16 where segmented objects from the

current photo are listed to be used for editing. Finally, the user may also cancel the process and go back anytime by clicking the left arrow cancel icon at the top-left corner.

Figure 15: Start Editing Photo Page

Segment Page

When the user choose *Segment* option, it is navigated to *Segment Page* shown in Figure 16 where automatically segmented objects from the current photo are listed for the user to be selected with *Select* button at the top-right corner. The user may scroll the list and *Choose* by clicking the button located at the bottom-left corner, from several selected objects to use and manipulate them on the currently editing photo. In this phase, the user may also save the selected objects to *Saved Objects* and store them for a later use by clicking *Save Object* button located at the bottom-right corner. Finally, the user may also cancel the process and go back anytime by clicking the left arrow cancel icon at the top-left corner.

Figure 16 : Segment Page

Continue Editing Photo Page

When the users choose one of three options (*Saved, Pencil, Segment*) in *Start Editing Photo Page* (Figure 15), and continue operating in the app by choosing the objects, they will be navigated to *Continue Editing Photo Page* shown in Figure 17. Here, they can manipulate (resize, move, change direction, etc.) chosen objects around the photo and put them to desired places by dragging and dropping. They may remove these objects by dragging them to *Remove* button located at the bottom-right corner. In this phase, the users may choose *Process* by clicking it at the top-right corner to let the changes operate in the photo and wait to see the addition and removal of the objects naturally. Before processing, they may also prefer to continue with choosing objects and manipulating them with *Saved* and *Pencil* options located at the bottom. In this page, since the automatically segmented objects are already decided and opened for

manipulating, the users don't have to resegment the photo everytime they want to choose an object. They may also choose these objects from Saved Objects if they previously saved them. Finally, the user may also cancel and go back anytime by clicking the left arrow cancel icon at the top-left corner.

Figure 17 : Continue Editing Photo Page

Process Changes Page

When the users click *Process* icon at the top-right corner of *Continue Editing Photo Page* (Figure 17) and the app finishes processing the changes made in the photo, they are navigated to *Process Changes Page* shown in Figure 18. In this page, the users may click *Reset* at the bottom-left corner and go back to *Start Editing Photo Page* (Figure 15) and restart editing from scratch. They may also redo or undo the changes

they made by clicking the arrows located at the bottom-center (left arrow for undo, right arrow for redo). The users may choose and compare the currently edited photo with its plain form before editing by holding and pulling the icon located at the bottom-left corner. If everything is as desired, they may choose to *Continue* with the edited photo and go to the next page, or they may want to cancel and go back by clicking the left arrow cancel icon at the top-left corner.

Figure 18 : Process Changes Page

Save Edited Photo Page

When the users click *Continue* button at the top-right corner of *Process Changes Page* (Figure 18), they are navigated to *Save Edited Photo Page* shown in Figure 19 with the final edited photo (in this example, the given photo is edited by placing the black pen at the bottom to the top of the white notebook next to white watch with keeping its

original place as well). In this page the users may *Save* the edited photo to their gallery by clicking its icon at the top-right corner, or they may start editing again if they want to by choosing one of three options (*Saved, Pencil, Segmented*). Finally, the user may also cancel and go back anytime by clicking the left arrow cancel icon at the top-left corner.

Figure 19 : Save Edited Photo Page

Pencil Page

When the users click the *Pencil* icon located at the bottom-center of the pages shown in Figure 15, 17, and 19, they are navigated to *Pencil Page* shown in Figure 20 where they are allowed to segment an object by specifying its borders with a pencil. After drawing around the object and segmenting it, they may choose and use it for manipulation in the editing page by clicking *Choose* button at the bottom-left corner. In this phase they may also save this object that they specified by pencil to Saved Objects and store it there for a possible later use if they want to. For this, they use *Save Object*

button located at the bottom-right corner. As specified in the other pages as well, the user may also cancel the process and go back anytime by clicking the left arrow cancel icon at the top-left corner.

Figure 20 : Pencil Page

Saved Objects Page from Edit Photo Page

If the users click *Saved* icon located at the bottom-left corners of the pages shown in Figure 15, 17, and 19, they are navigated to *Saved Objects Page from Edit Photo Page* shown in Figure 21 where they can see the list of previously Saved Objects. They may scroll this list and select desired ones by first clicking *Select* button at the top-right corner and then they can choose these selected objects to use and manipulate them in the editing pages by clicking *Choose* button located at the bottom-right corner.

As specified in the other pages , the user may also cancel the process and go back anytime by clicking the left arrow cancel icon at the top-left corner.

Figure 21: Saved Objects Page from Edit Photo Page

Saved Objects Page from Get Started Page

This *Saved Objects Page* is loaded only when the users click *Saved Objects* button in *Get Started Page* shown in Figure 11. This page also allows the users scroll the Saved Objects list to show the currently saved objects. In this page only, the users may first *Select* undesired objects and *Remove* them if they want to by clicking the buttons at the top and bottom left corners as shown in Figure 22. This *Saved Objects Page* is only used for removal of the selected objects since at this phase, there is no photo chosen for

editing yet. The user may also cancel the process and go back to *Get Started Page* (Figure 11) anytime by clicking the left arrow cancel icon at the top-left corner.

Figure 22: Saved Objects Page from Get Started Page

Learn More Page

The users are directed to this page if they tap *Learn More* button in *Entry Page* (Figure 10). As shown in Figure 23, *Learn More Page* has three buttons to direct the users to *Contact, Team,* and *FAQ.* They may go to the previous page (Entry Page, Figure 10) by clicking the left arrow cancel icon at the top-left corner.

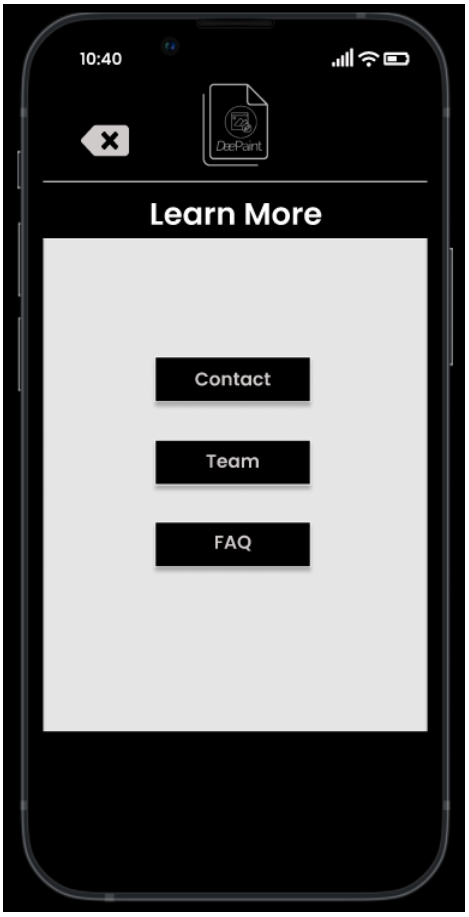

Figure 23: Learn More Page

4. Other Analysis Elements

In this section, we present how our project is related to real life elements.

4.1. 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 to apply changes in the image. Using mobile phones excessively cause some physical problems such as neck and eye problems [2]. This is not special to our application but as the program increases the amount of time spent for 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 [3]. 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 cancelled by society. This is one example. As our application has a 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 the 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

4.2. Risks and Alternatives

There are different risks during the development of DeePaint. We have considered various B plans regarding these problems.

1. Insufficient Accuracy and Capacity of Neural Networks

   We will use different neural network architectures for different tasks. We found the state of art models for each of the tasks. However, their performance still might not be good enough. For instance, our pre-trained instance segmentation network has a class capacity. If we observe that the number of classes is not enough for our application, we can re-train the model with more classes. If we see that the accuracy of the network is not enough, we can increase the capacity of the model and train it with more data.

2. Insufficient Speed Performance of Neural Networks

   As we want to create a smooth application, we want to make our program fast. However, neural networks we want to use have lots of parameters and might work slow even if in GPU. Therefore, we might adopt some machine learning acceleration techniques such as network pruning and quantization which make models faster.

3. One of Team Members Leaving

   It is possible that one person in the team can leave the project because of various reasons such as health issues, withdrawal of course etc. Therefore, we design our

teamwork such that every person in the team knows the other's work. Therefore, if someone leaves, there is always a person knowing the leaving person's job and can complete it.

4. Infeasible Server Costs

   As neural networks' size is big, we cannot store and process them in mobile phones. Therefore, we put in a server having powerful GPUs and process the images in the servers. However, these servers have a cost and we should pay it. If it becomes infeasible to be paid, we should use cheaper servers. However, note that cheaper servers mean less computational power and less speed which decreases the smoothness of the program. Therefore, we have to find a sweet spot at this point.

	Likelihood	Effect on the Project	B Plan
Insufficient Accuracy and Capacity of Neural Networks	Medium	Quality of the generated images decreases	Training the model with more data
Insufficient Speed Performance of Neural Networks	Medium	Speed of the program decreases	Quantization and Pruning of neural networks
One of Team Members Leaving	Low	The work power decreases	Re-plan of works
Infeasible Server Costs	Low	The application can be closed	Using more optimal servers

4.3. Project Plan

Planning a software project is a very crucial step. Therefore, our team has created a plan to meet requirements. 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. That does not mean the separation is strict. Every person will participate in both parts of the work.

We have divided the whole work into small work packages. According to this, we have N 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.

WP1: Analysis Report	Leader: Zübeyir Bodur	Members Involved: All Members
Objective: Provide a detailed analysis of the project to ensure proper teamwork and proper development plan.		
Start Date:1 November 2021 End Date: 15 November 2021		
Tasks: 1. Determining Functionalities 2. Diagrams 3.Research on Neural Networks		
Deliverables: Analysis Report		

WP2: High Level Design Report	Leader: Duygu Nur Yaldız	Members Involved: All Members
Objective: Creating a detailed design of the project to make implementation smoother.		
Start Date: 12 December 2021 End Date: 24 December 2021		
Tasks: 1.Detailed Implementation Plan		
Deliverables: High Level Design Report		

WP3: Implementation of Neural Networks for segmentation and Image Generation Tasks	Leader: Yavuz Faruk Bakman	Members Involved: Alperen Öziş, Duygu Yaldız
Objective: Implementing the state-of-art neural networks for segmentation, image blending and inpainting tasks		
Start Date: 25 December 2021 End Date: 3 January 2022		
Tasks: 1. Determining Which Neural Networks are used 2. Implementing them on Pytorch or Tensorflow		
Deliverables: None		

WP4: Development of User Interface and Backend in Android	Leader: Hande Sena Yılmaz	Members Involved: Zübeyir Bodur, Duygu Yaldız
Objective: Developing the graphical user interface of the mobile application. Ease of use is our first priority.		
Start Date: 25 December 2021 End Date: 3 January 2022		
Tasks: 1. Development of Front-End 2. Development of Back-End		
Deliverables: None		

WP5: Connecting Android Application with Neural Networks for Object Removing Task Only	Leader: Alperen Öziş	Member Involved: All Members
Objective: Connecting the Neural network development and application development for a simple task to show we are in the right direction		
Start Date: 4 January 2022 End Date: 8 January 2022		
Tasks: 1. Connection between Neural Network and App		
Deliverables: None		

WP6: Demo in Desktop	Leader: Duygu Yaldız	Members Involved: All Members
Objective: A simple demonstration of the application on the desktop. The application will work with few options		
Start Date: 8 January 2022 End Date: 15 January 2022		
Tasks: 1. Desktop Demo		
Deliverables: First Demo		

WP7: Low Level Design Report	Leader: Duygu Yaldız	Members Involved: All Members
Objective: Preparing a comprehensive low level design report for the implementation		
Start Date: 3rd week of the semester End Date: 5th week of the semester		
Tasks: 1. Changes 2. Extended Features		

Deliverables: Low Level Design Report

WP8: Add Other Functionalities into Neural Networks	Leader: Alperen Öziş	Members Involved: Yavuz Bakman, Duygu Yaldız
Objective: Adding instance segmentation and image blending features to the neural networks		
Start Date: After low level design Report End Date: 7th week of the semester		
Tasks: 1. Implementing new Neural Networks		
Deliverables: None		

WP9: Add Other Functionalities into Back-end and Front-end	Leader : Zübeyir Bodur	Members Involved: Yavuz Bakman, Hande Sena Demir
Objective: Adding instance segmentation and image blending features to the back-end and front-end of the application		
Start Date: After low level design Report End Date: 7th week of the semester		
Tasks: 1. Back-end for new features 2. Front-end for new features		
Deliverables: None		

WP10: Connect server-application	Leader : Duygu Yaldız	Members Involved: All Members
Objective: Putting neural networks into a server and making connection with the server and the mobile application		
Start Date: 7th week of the semester End Date: 8th week of the semester		
Tasks: 1. Deploying neural networks into the server 2. Connection between server and mobile App		
Deliverables: None		

WP11: Mobile Phone Demo	Leader : Alperen Öziş	Members Involved: All Members
Objective: Finishing the development by adding all facilities into the mobile phone application.		
Start Date: 8th week of the semester End Date: 11th week of the semester		
Tasks: 1. Development in Backend 2. Development in Frontend 3. Development in NNs		
Deliverables: Mobile Phone Application		

WP12: Final Report	Leader: Duygu Yaldız	Members Involved: All Members
Objective: Preparing a comprehensive final report describing all features of DeePaint.		
Start Date: 11th week of the semester End Date: The last week of the semester		
Tasks: 1. Final Architecture 2. Maintenance Plan		
Deliverables: Final Report		

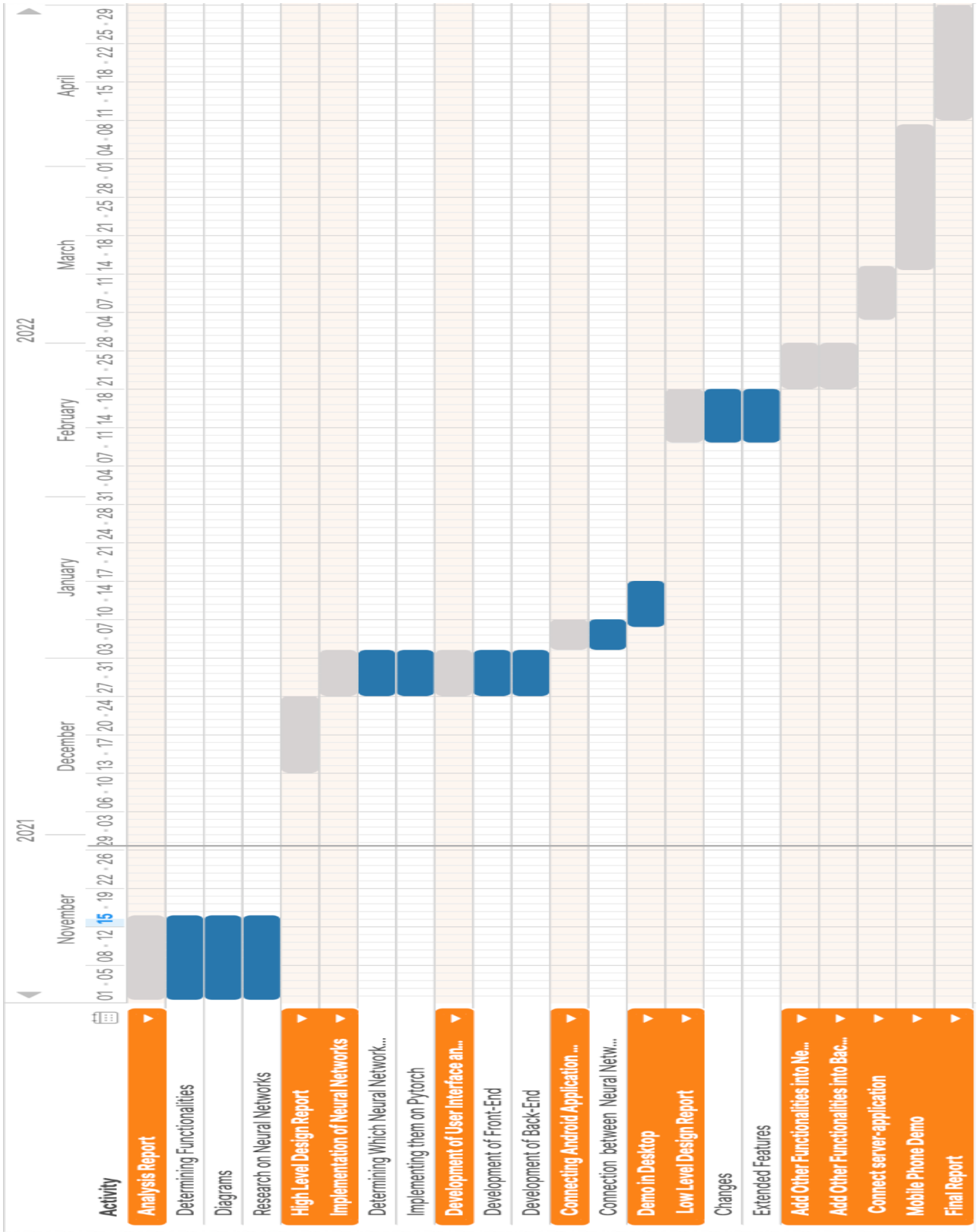

Figure 24: Gantt Chart of the Plan

4.4. Ensuring Proper Teamwork

We are aware of the reality that software projects require good communication and proper teamwork. To ensure that, we are meeting after each CS-491 seminar. Also, we are using a Whatsapp group for informal chat. For formal communication, we have a Slack channel. For the implementation part, we will divide big tasks into smaller parts and assign each of them to one member of the team. However, it does not mean each person only cares about his/her work. We will schedule review sections to control and give feedback about every person's work. By that way, each member will be aware of the general flow of the project. For each task, we set a start and end time. These times are flexible but should be strict in some sense because the deadlines are strict in the course. Furthermore, as we did group projects together before, most group members know other members' strengths and weaknesses. Therefore, group members can help with other member's tasks.

4.5. Ethics and Professional Responsibilities

In our application data privacy is very important. Therefore, we follow the data privacy code [4]. Another important factor in our program is the biases in the neural networks. Neural networks can carry implicit biases against some ethnicities, cultures etc. However, there are also well-studied methods to reveal biases and prevent biases. We will ensure that our neural networks used in photo-shop do not carry such biases. The libraries, APIs will be used according to their license agreements. Furthermore, we will follow the ACM code of ethics [5].

4.6. Planning for New Knowledge and Learning Strategies

Some of the group members do not have any experience in mobile application development. These members will learn android studio from Udemy Courses and the public tutorials on the internet. We do not have too much experience in generative and segmentation neural networks although we have experience in neural networks and machine learning in general. Therefore, we read the papers we found about the following topics:

1. Instance Segmentation:
    a. SipMask: Spatial Information Preservation for Fast Image and Video Instance Segmentation
2. Image Inpainting:
    a. Contextual Residual Aggregation for Ultra High-Resolution Image Inpainting

    b. SESAME: Semantic Editing of Scenes by Adding, Manipulating or Erasing objects
3. Image Blending:
    a. Deep Image Blending
    b. Region-aware Adaptive Instance Normalization for Image Harmonization

Besides the literature review, we do not know how to upload a neural network into a server and communicate with mobile applications. To learn this, we will utilize Udemy courses, tutorials. Also, our supervisor, Uğur Doğrusöz can provide a pathway to do that task.

5. Glossary

- Object : On 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.
- GPU : Graphics Processing Unit. A specific hardware that is designed for parallel processing. GPU’s are very efficient in operations such as matrix multiplication and convolution. This makes GPU’s important for us since our neural network models are taking advantage of those operations [6].
- Neural Network: Algorithms that reflect the behaviour of the human brain, allowing programs to recognize patterns [7].
- WP: Work Package. Smallest amount of work of the project that we need to follow.
- WBS: Work Package Breakdown Structure. The general structure that determines the projects in terms of deliverables and work packages.

6. 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] “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].

[3] “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].

[4] “Data Privacy,” *Code of Conduct*, [Online]. Available: https://code-of-conduct.roche.com/en/data-privacy.html. [Accessed Nov. 13, 2021].

[5] “ACM Code of Ethics and Professional Conduct,” *ACM*, [Online]. Available: https://www.acm.org/code-of-ethics . [Accessed Nov. 13, 2021].

[6] “What Is a GPU?,” *ACM*, [Online]. Available: https://www.intel.com.tr/content/www/tr/tr/products/docs/processors/what-is-a-gpu.html. [Accessed Nov. 15, 2021].

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