



Bilkent University

Department of Computer Engineering

---

# Senior Design Project

*Project name: Upfix*

## High Level Design Report

Meryem Efe, Hamza Pehlivan, Özge Yaşayan, Hazal Aksu, Rabia Nur Önal

**Supervisor:** Uğur Gündükbay

**Jury Members:** Can Alkan, Çiğdem Gündüz Demir

# Contents

<b>1. Introduction</b>	<b>4</b>
1.1. Purpose of the System	5
1.2. Design Goals	6
1.2.1. Security	6
1.2.2. Usability	6
1.2.3. Cost	6
1.2.4. Performance	6
1.2.5. Extendibility	6
1.2.6. Marketability	7
1.3. Definitions, Acronyms, and Abbreviations	7
1.4. Overview	7
<b>2. Current Software Architecture</b>	<b>8</b>
<b>3. Proposed Software Architecture</b>	<b>9</b>
3.1. Overview	9
3.2. Subsystem Decomposition	10
3.3. Hardware / Software Mapping	11
3.4. Persistent Data Management	12
3.5. Access Control and Security	12
3.6. Global Software Control	13
3.7. Boundary Conditions	13
3.7.1. Initialization	13
3.7.2. Termination	13
3.7.3. Failure	13
<b>4. Subsystem Services</b>	<b>14</b>
4.1. Web Application	14
4.1.1. User Interface	14
4.1.1.1. Web View	14
4.1.1.2. Home Page	14
4.1.1.3. Select Game/ Algorithm Page	14
4.1.1.4. Loading Page	15
4.1.1.5. Download Page	15
4.1.2. Options	15
	2

4.1.3. Download Manager	15
4.1.4. Display Manager	15
4.1.5. Video Request Handler	15
4.1.6. Server Interface	16
4.2. Desktop Application	16
4.2.1. User Interface	16
4.2.1.1. Upload Video Page	16
4.2.1.2. Select Game/ Algorithm Page	16
4.2.1.3. Loading Page	16
4.2.1.4. Download Page	16
4.2.2. Options	17
4.2.3. File Manager	17
4.2.4. Display Manager	17
4.2.5. Deep Learning Interface	17
4.3. Video Request	17
4.3.1. WebScrapers	17
4.4. Server	17
4.4.1. Request Handler	18
4.4.2. Deep Learning Interface	18
4.5. Deep Learning	18
4.5.1. Upscaler	18
4.5.2. Preprocess	18
4.5.3. Network	18
4.6. Video Pipeline	18
4.6.1. Post-Process	19
4.6.2. Video Renderer	19
4.6.3. Voice Adjuster	19
<b>5. Consideration of Various Factors in Engineering Design</b>	<b>19</b>
<b>6. Teamwork Details</b>	<b>20</b>
6.1. Contributing and Functioning Effectively on the Team	20
6.2. Helping creating a collaborative and inclusive environment	20
6.3. Taking Lead Role and Sharing Leadership on the Team	21
<b>7. References</b>	<b>22</b>

# High-Level Design Report

*Project Short-Name: Upfix*

## 1. Introduction

Gaming is undoubtedly a significant source of entertainment for many people. The game industry has been growing rapidly, notably with the ever-rising popularity of video games as well as online versions of classic games such as chess.

The number of people who play games is increasing thanks to such popular games and online platforms [1, 2]. Consequently, the number of people who consume video contents related to gaming is increasing simultaneously as well. As a result, a new sub-industry named Gaming Video Content (GVC) has emerged. According to data collected in 2017, the number of GVC viewers has reached 666 million globally [2]. In the GVC industry, Twitch is the leading platform, accounting for 54% of the gaming video content platform revenue in 2017 with Youtube following right behind [3]. Even though these platforms are widely popular, users can still face issues in watching game videos in high quality which is mostly due to limited internet connection. Since watching videos in high resolution consumes more data, users opt to watch videos in low resolution even though it may not be desired. Furthermore, the same problem occurs when people want to upload a game video on the internet. Therefore, all these issues compel us to pay attention to the need of improved video upscaling techniques.

Note that big game companies are trying to provide high quality game videos for their viewers. For example, Valve provides GOTV to stream Counter Strike tournaments and DotaTV for Dota2 tournaments [4, 5]. Their approach requires geographically distributed proxy servers, which small game companies may not prefer. Furthermore, in general, tournaments and important events are streamed on the network with this approach. This means viewers who want to watch gameplays other than predetermined contents may not be able to obtain high quality videos.

The purpose of our Senior Design Project is to design and implement an application which will provide a platform that provides higher quality game videos. Initially, we are planning to provide chess, go, Age of Empires [6] and Among Us [7] game videos. Using our application, games of relatively small companies, which do not prefer investing money in proxy servers, can be watched with high quality. Moreover, viewers who want to watch their favorite streamer - some chess or go player for example- with high quality can use our application.

In this report, we are going to narrate a detailed description of the application as well as constraints, professional and ethical issues about the project.

## **1.1. Purpose of the System**

Upfix is a tool for upscaling gameplay videos. The users can choose between the website and the desktop application, according to their needs.

The website is capable of upscaling a video provided via a URL from Twitch or YouTube, as well as a video that the user uploads to the system. The upscaling is done on the server, whereas the desktop application makes use of the resources of the user's computer, which is quite useful when the user does not have a reliable Internet connection. A video from the user's local file system can be upscaled on the desktop application.

Upfix uses artificial neural networks in order to upscale videos. A pre-trained neural network provides a way for users to enjoy watching gameplay without being affected by the speed of their Internet connection, which is the main advantage of using Upfix. Upfix aims to improve the video viewing experience for users by eliminating interruptions and increasing the quality of the videos.

Currently, there are video upscaling applications on the market, however, none of them are specialized for use in the gaming industry.

## **1.2. Design Goals**

### **1.2.1. Security**

- The system should not disclose the data of its users to third parties.
- The system does not require ID information, e-mail, or password.

### **1.2.2. Usability**

- Both the web and desktop applications will be equipped with a user-friendly interface, as well as a simple manual for new coming users.

### **1.2.3. Cost**

- Our application will not demand any charge from its users.

### **1.2.4. Performance**

- The system should start delivering high quality video fluently within two minutes, after processing of low resolution video starts.

### **1.2.5. Extendibility**

- The design of the system should allow easy integration of other video sharing platforms like Dailymotion.
- New deep learning algorithms for super resolution should be easy to integrate into the system.
- If the necessary technical conditions are met, the content under which the application runs can be expanded. For instance, the models could be trained not only for the game videos we chose but also the popular video games.

### **1.2.6. Marketability**

- The system can be used to decrease internet traffic, therefore, content receivers can choose to use our system to obtain much more quality content with less usage of data.

### 1.3. Definitions, Acronyms, and Abbreviations

Term	Definitions, acronyms, and abbreviations
OpenCV	“OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms [8].”
CNN	Convolution Neural Network [9]
GAN	Generative Adversarial Networks [10]
Transfer Learning	Transfer learning is a machine learning technique where a model developed for a task is reused as the starting point for a model on another related task [11, 12].
React.js	React.js is a JavaScript library used in web development to build interactive elements on websites [13].
Spring	Spring is a framework which provides a comprehensive programming and configuration model for modern Java-based enterprise applications [14].
JavaFX	“JavaFX is a set of graphics and media packages that enables developers to design, create, test, debug, and deploy rich client applications that operate consistently across diverse platforms [15].”
GVC	Gaming Video Content
SR	Super Resolution

### 1.4. Overview

Upfix is a website and a desktop application for upscaling videos. It does not require making an account/logging in. The users can simply download the desktop app or provide a link to the website to start using the functionality of Upfix.

Both the website and the desktop application will provide a media player for the user to watch the upscaled videos.

The user can upload a video from their local files to the desktop application and wait for the neural network to upscale it. After the video is done being upscaled, the user can play the video and save it to their computer. The desktop application uses the resources of the user's computer to upscale the videos.

Similarly, the user can provide a link to a video such as a YouTube or Twitch link, and wait for the neural network to upscale it. After the upscaling process, the user can play the video and download it to their computer. The web application uses the remote server's resources to upscale the videos.

## **2. Current Software Architecture**

As it is discussed in the introduction, some big companies are trying to provide high quality game videos for their viewers. However, in fact, upscaling is not just a concern for the GVC industry. Upscaling, also known as super resolution, technologies are trying to be improved to get a high quality version for old films, old photographs, or for scientific purposes. OpenCV is one of the open source libraries which provides deep learning based pre-trained models for super resolution [16]. There are currently 4 different super resolution models provided by OpenCV. These are EDS, ESPCN, FSRCNN, LapSRN [17]. All these models differ in accuracy, size, and speed. The best performing, biggest, slowest model is EDSR. ESPCN and DSRCNN are smaller and faster models as well as they can do real-time video upscaling. Lastly, LapSRN is a medium sized model and can upscale by a factor of 8 while the other models can upscale by a scale of 4 at most.

In fact, as the Upfix development team, we were not dealing with the pre-trained models at the beginning of the development process. However, we realized that even though we chose a specific content to train our models, we still have technical insufficiencies. Therefore, we decided that while we try to improve the models we trained from scratch, at the same time, we are trying to integrate pre-trained models into our application. The super resolution models in OpenCV are trained on general pictures. That is, they are



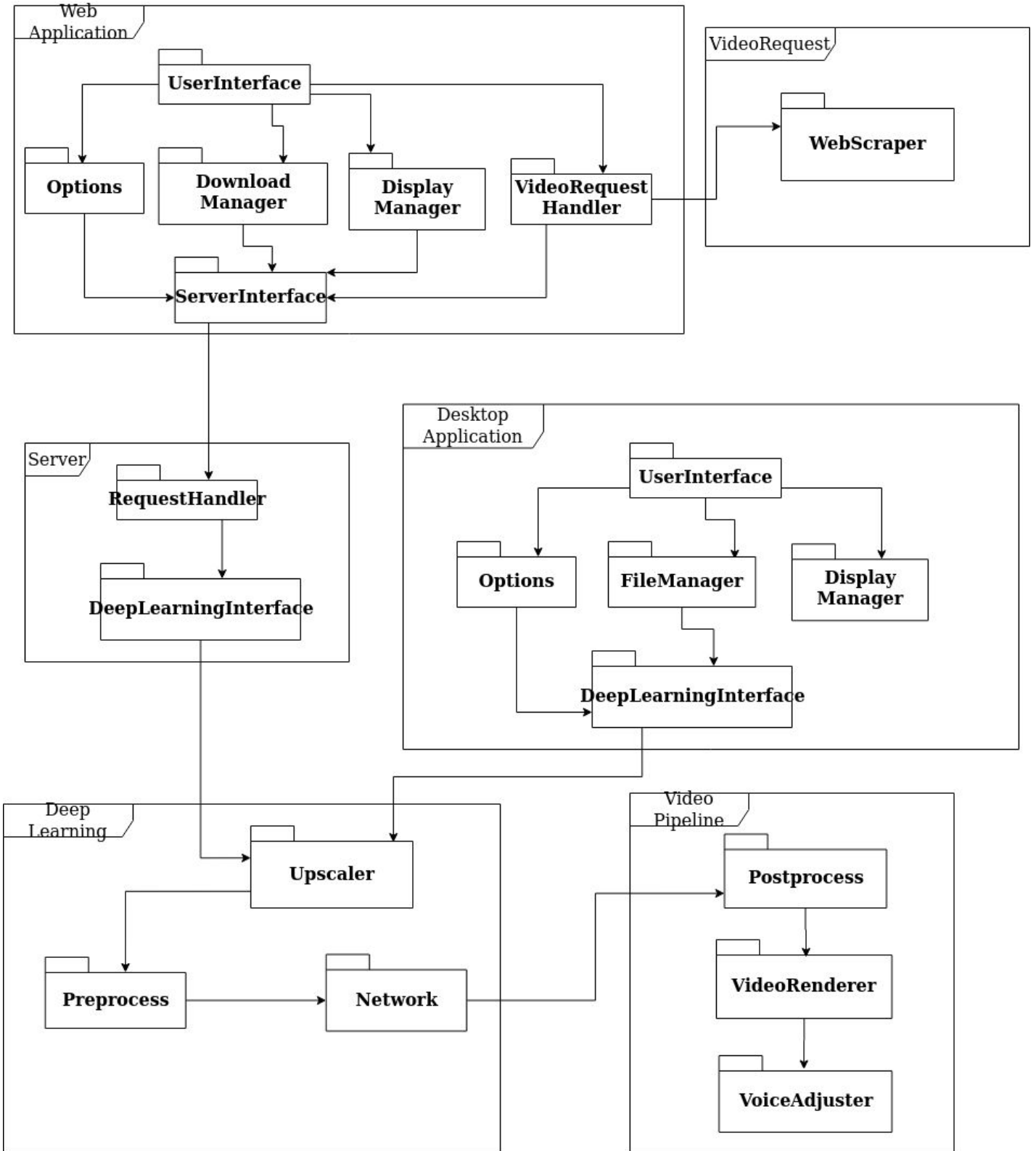
not trained on a specific content such as vegetables, human faces, phones etc. Therefore, we planned to apply transfer learning on these pre-trained models to train them in our scope in order to improve the results.

## **3. Proposed Software Architecture**

### **3.1. Overview**

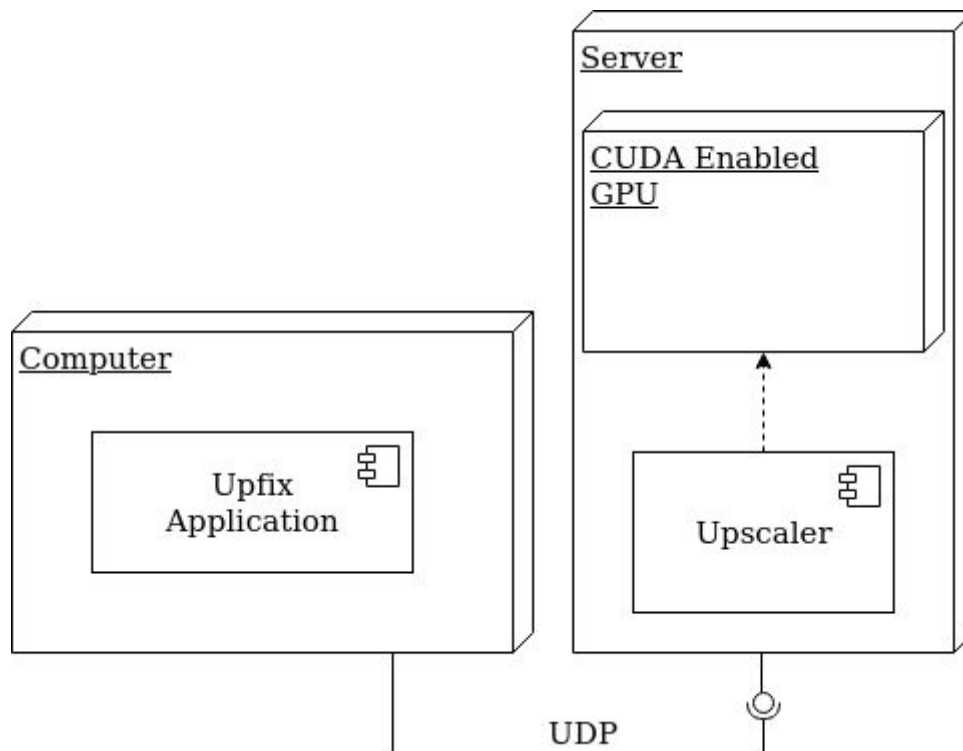
In this part, we are going to explain subsystem decomposition which describes the subsystem structure of our system in detail with diagrams. After, we will provide hardware / software mapping of the system. Then, we will mention persistent data management, access control and security, global service control, and boundary conditions.

### 3.2. Subsystem Decomposition

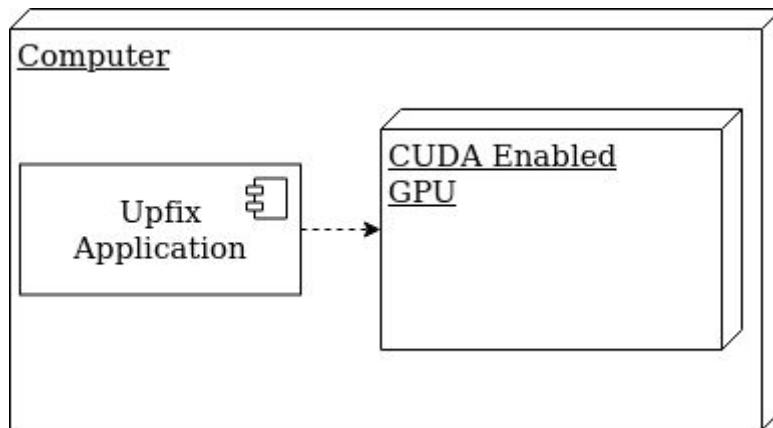


### 3.3. Hardware / Software Mapping

In our web application, the communication between server and the client is established via UDP connection. We chose UDP because there is no need for fully reliable connection and there is need for fast communication. The server utilizes a dedicated GPU to accelerate the inference and video rendering. We summarized all these components in the following picture:



The user must have a CUDA enabled GPU to use our desktop application. Like in the web application, the GPU will be used to speed up inference and video rendering time.



### 3.4. Persistent Data Management

For the desktop and web versions of Upfix, we need to persist different data. In the web application, neural network models need to be persistent. Since the models are trained by our team and not changed but used by users, we will use a filesystem to persist neural network models in the server side.

In the desktop application, similarly, models need to be persistent. We will use a filesystem to persist models for the desktop application as well. However, differently from the web app, we will not use a server. Instead, we will store the models in the user's localhost. Additionally, in the desktop application, we will provide user settings options. In the settings, users can choose the auto save location, theme of the application, volume of videos and can see previously upscaled videos' list. These persistent data will be serializable and stored in the filesystem.

### 3.5. Access Control and Security

Everyone can use Upfix without signing up to the system. Therefore, for accessing, there is no requirement except that Upfix Web App requires stable internet connection.

For security, Upfix doesn't track user's information since it has no account or profile functions. It also doesn't share the content of the videos which are uploaded by users with the third party applications. Also, the uploaded videos

are not stored in the database for any purposes. Therefore, it assures to take care of the data privacy of users.

### **3.6. Global Software Control**

The software control will use centralized event driven software control. The user will choose what service will be run and s/he will initiate every action. Because of the event driven nature, the software always waits for the user input. After it receives the input, it will provide the requested service to the user.

### **3.7. Boundary Conditions**

#### **3.7.1. Initialization**

Upfix will provide users with both web and desktop applications. None of these versions requires any sign in / sign up processes. Hence, when the application is started, users will directly meet the homepage where they can learn how to use this app. When a new user launches the application, they will see a pop-up window for a quick tutorial. They can navigate the application through this tutorial or they can opt by clicking "Don't show it again." not to see this tutorial. After they close this pop-up window, they will see the main page where they can upload to a low resolution video.

#### **3.7.2. Termination**

Users can close the application at any time they desire without logging out since Upfix doesn't require authentication. However, the application will ask them "Are you sure?" in a pop up window, in case users unintentionally request shutdown. If users insist on closing the application, the system approves this request and terminates without saving any process.

#### **3.7.3. Failure**

As it is said before, in case the application is closed unintentionally, it asks users to approve their shutdown request. If another error causes the

application to crash, the system will not save the current status of the uploading or converting process. Since the desktop application doesn't require the Internet connection, its failures will not result in wasted data consumption. However, the failures in the web application will cause users to reupload their videos and to consume more data. Therefore, in the beginning, users are informed that they need a stable internet connection to start using Upfix Web App.

## **4. Subsystem Services**

### **4.1. Web Application**

It handles the user interface and user related services of the web application. According to the user input, it communicates with the Server and Video Request subsystems.

#### **4.1.1. User Interface**

##### **4.1.1.1. Web View**

The Web View subsystem is responsible for the visual interactions between the user and the web application. The breakdown of the Web View's components is as follows:

##### **4.1.1.2. Home Page**

The Home Page is the first and main page seen when the user enters the webpage of the web application. This is the main page that controls the interactions between the user and the web application. From here the user can enter a URL of the gaming video which they would like to upscale.

##### **4.1.1.3. Select Game/ Algorithm Page**

This is the page that is shown to the user once a URL is entered the "Next" button on the Home Page is clicked. From here the user can select the type of

the game in the video of the URL they entered from a dropdown menu and select the upscaling algorithm they want the web application to use.

#### **4.1.1.4. Loading Page**

Upon clicking “Next” on the Select Game/Algorithm Page, the user will be met with a recess page which contains a loading screen that shows the progress of the video being upscaled in the server.

#### **4.1.1.5. Download Page**

Once the video is upscaled, this page will show up. The user can watch the upscaled video from this page and/or download it to their local device. They can also redirect to the Home Page.

#### **4.1.2. Options**

This specifies the type of the video to be upscaled in order to apply the correct deep learning method.

#### **4.1.3. Download Manager**

The Download Manager makes the users able to download the upscaled video from the server to their computers.

#### **4.1.4. Display Manager**

The Display Manager is responsible for displaying the upscaled video to the user.

#### **4.1.5. Video Request Handler**

This is responsible for taking the URL of the video to be upscaled from the user and sending it to the WebScraper.

#### **4.1.6. Server Interface**

This makes the connection between the web application and the server where the upscaling takes place.

### **4.2. Desktop Application**

It handles the user interface and user related services of the desktop application. According to the user input, it communicates with the Deep Learning subsystem.

#### **4.2.1. User Interface**

##### **4.2.1.1. Upload Video Page**

This is the first page that the user will see when they launch the desktop application. Here, the user can upload a video to the application from their local memory.

##### **4.2.1.2. Select Game/ Algorithm Page**

After uploading their preferred video, the user will then, similar to the web application, be shown a page that asks them to choose the game type and upscaling algorithm.

##### **4.2.1.3. Loading Page**

Just like the web application, the user will be shown a loading screen while the video is being upscaled, this time via their own device's CPU.

##### **4.2.1.4. Download Page**

This is the final screen that the user will see after the video upscaling is completed. Same as the web version, the user can choose to watch the upscaled video, save it on their device or redirect to the upload page.



#### **4.2.2. Options**

This specifies the type of the video to be upscaled in order to apply the correct deep learning method.

#### **4.2.3. File Manager**

The File Manager makes the user able to select the video from her computer's local storage. It is also responsible for saving the upscaled version of the video.

#### **4.2.4. Display Manager**

The Display Manager is responsible for displaying the upscaled video to the user.

#### **4.2.5. Deep Learning Interface**

This enables the communication between Desktop Application and Deep Learning subsystems.

### **4.3. Video Request**

It is responsible for finding a video on the internet via web crawling.

#### **4.3.1. WebScrapers**

This takes the URL of the video from the web application Video Request Handler and returns the content of the video found on the Internet.

### **4.4. Server**

It handles the requests coming from clients and provides them GPU as a service.

#### **4.4.1. Request Handler**

This maps the upscaling requests from the web application and reaches the Deep Learning system through an interface.

#### **4.4.2. Deep Learning Interface**

This enables the communication between Server and Deep Learning subsystems.

### **4.5. Deep Learning**

It prepares the coming video and inferences them using the neural network.

#### **4.5.1. Upscaler**

Upscaler behaves as a controller for Deep Learning subsystem. It takes the input and forwards them to proper classes.

#### **4.5.2. Preprocess**

It performs essential preprocessing operations. Firstly, it extracts frames of the video and normalizes these extracted frames. Then, it converts them into a proper input structure in order to send them to the network part.

#### **4.5.3. Network**

It takes preprocessed frames and applies the desired neural network model on the input frames. In this way, it upscales the frames.

### **4.6. Video Pipeline**

It applies post processing to images which are coming from the Deep Learning Subsystem. Post processed images are rendered to create high resolution video. After rendering, it merges the high resolution video with the audio of the original low resolution video.

#### **4.6.1. Post-Process**

It applies post processing to images that are provided by Deep Learning Subsystem. This package will include image enhancing algorithms like image sharpening.

#### **4.6.2. Video Renderer**

Upscaled images will be rendered and converted to video in this package.

#### **4.6.3. Voice Adjuster**

After creating a high resolution video, we need to merge it with the audio of the original low resolution video. This package will include necessary software for this operation.

## **5. Consideration of Various Factors in Engineering Design**

After thorough deliberation, several conclusions on various factors that could affect and/or change the way the application was designed and will be implemented has been reached.

It has been determined that Upfix does not pose any threat to public health, welfare, or safety. This is because the application, by its very nature and design, does not affect public health or welfare. It is not a threat to public safety because it does not hold any information about its users, nor does it share and/or sell data to any third parties.

As a software application that is free of charge, environmental and economic factors did not play a big role in the designing of Upfix.

Global, cultural, and social factors have little effect in Upfix's design, because it is an application that interacts with its users on a one-on-one basis rather than being a platform where users of the application can interact. Thus, the

only thing taken into consideration while designing the application was the language displayed on its UI, and it was made sure that the language is inclusive and respectful towards all races and genders.

## **6. Teamwork Details**

This section is going to define the mechanics of the teamwork among our group members for the senior project.

### **6.1. Contributing and Functioning Effectively on the Team**

Our group consists of different teams that have different responsibilities. Teams communicate among the members of the same team and we all try to communicate with everyone in the group to summarize our team's progress weekly. Having subgroups, or teams, as opposed to one big group, helps us keep track of the tasks more effectively and contributing is easier than having just one group because the tasks can be assigned more easily and we can spend more time focusing on our tasks rather than constantly reporting to a big group of people. It is faster to communicate with 2-3 members. We can also spend more time on defining the tasks more clearly and discussing the specifications among our teams, and since there are less members, it is easier to reach a consensus.

### **6.2. Helping creating a collaborative and inclusive environment**

Before starting the project, every group member talked about their past experience with relevant technologies and mentioned what part of the project they would prefer to work on. Having this discussion allowed us to focus on our strengths as individuals and divide the work among ourselves in a way that would emphasize our skills. Some members took machine learning classes, and they had more experience with generating models, therefore they offered to work on the machine learning model. The rest of the members were more experienced in web and desktop application development, so they chose to focus on those aspects of the project. There were also some

members who worked with web crawlers before, and they agreed to contribute to that part of the project when it was time to implement that. Allowing members to choose what they wanted to work with, increased the collaboration and made sure that everyone was included in the project. Everyone had a chance to put their past experiences and existing skills into practice with the project and build upon those skills.

### **6.3. Taking Lead Role and Sharing Leadership on the Team**

Our group does not have a clear leader, however, some members are more involved in organizing meetings and arranging the report templates as well as reminding others about the deadlines. Overall, subgroups divide the work among themselves however they want, and then report to each other. Everyone contributes to the reports as well. That being said, there are members who take more initiative in order to regulate the teamwork among our group.

## 7. References

- [1] “Hours of Gaming Unboxing Videos on YouTube Watched on Mobile,” *Think with Google*. [Online]. Available: <http://www.thinkwithgoogle.com/marketing-strategies/video/gaming-video-watch-time-statistics-on-youtube/>. [Accessed: 27-Dec-2020].
- [2] A. Guttman, “Topic: Gaming video content market,” *Statista*. [Online]. Available: <http://www.statista.com/topics/3147/gaming-video-content-market>. [Accessed: 27-Dec-2020].
- [3] “2020 Video Game Industry Statistics, Trends & Data,” *WePC*, 09-Nov-2020. [Online]. Available: <http://www.wepc.com/news/video-game-statistics>. [Accessed: 27-Dec-2020].
- [4] “Counter-Strike: Global Offensive Broadcast,” *Valve Developer Community*. [Online]. Available: [https://developer.valvesoftware.com/wiki/Counter-Strike:\\_Global\\_Offensive\\_Broadcast\\_ast/](https://developer.valvesoftware.com/wiki/Counter-Strike:_Global_Offensive_Broadcast_ast/). [Accessed: 27-Dec-2020].
- [5] “Dota Watch,” *Dota*. [Online]. Available: <http://www.dota2.com/watch/>. [Accessed: 27-Dec-2020].
- [6] “Age of Empires Franchise - Official Website,” *Age of Empires*, 27-Aug-2020. [Online]. Available: <https://www.ageofempires.com/>. [Accessed: 27-Dec-2020].
- [7] “Among Us on Steam,” *Steam*. [Online]. Available: [https://store.steampowered.com/app/945360/Among\\_Us/](https://store.steampowered.com/app/945360/Among_Us/). [Accessed: 27-Dec-2020].
- [8] *OpenCV*, 13-Oct-2020. [Online]. Available: <https://opencv.org/>. [Accessed: 27-Dec-2020].
- [9] “Convolutional Neural Networks,” *Coursera*. [Online]. Available: <https://www.coursera.org/learn/convolutional-neural-networks>. [Accessed: 27-Dec-2020].

- [10]** “Generative Adversarial Networks (GANs),” Coursera. [Online]. Available:  
<https://www.coursera.org/specializations/generative-adversarial-networks-gans>. [Accessed: 21-Nov-2020].
- [11]** J. Brownlee, “A Gentle Introduction to Transfer Learning for Deep Learning,” *Machine Learning Mastery*, 16-Sep-2019. [Online]. Available:  
<https://machinelearningmastery.com/transfer-learning-for-deep-learning/>. [Accessed: 27-Dec-2020].
- [12]** F. 7 and S. Martin, “What Is Transfer Learning?: NVIDIA Blog,” *The Official NVIDIA Blog*, 14-Feb-2019. [Online]. Available:  
<https://blogs.nvidia.com/blog/2019/02/07/what-is-transfer-learning/>. [Accessed: 27-Dec-2020].
- [13]** “React – A JavaScript library for building user interfaces,” *React*. [Online]. Available: <https://reactjs.org/>. [Accessed: 27-Dec-2020].
- [14]** “Spring makes Java simple.,” *Spring*. [Online]. Available:  
<https://spring.io/>. [Accessed: 27-Dec-2020].
- [15]** *JavaFX*. [Online]. Available: <https://openjfx.io/>. [Accessed: 27-Dec-2020].
- [16]** X. Weber, “Deep Learning based Super Resolution with OpenCV,” *Medium*, 18-Jul-2020. [Online]. Available:  
<https://towardsdatascience.com/deep-learning-based-super-resolution-with-opencv-4fd736678066>. [Accessed: 27-Dec-2020].
- [17]** Opencv, “opencv/opencv\_contrib,” *GitHub*. [Online]. Available:  
[https://github.com/opencv/opencv\\_contrib/tree/master/modules/dnn\\_superres](https://github.com/opencv/opencv_contrib/tree/master/modules/dnn_superres). [Accessed: 27-Dec-2020].