Bilkent University

Department of Computer Engineering

Senior Design Project

Project name: Upfix

Project Specification Report

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

Supervisor: Uğur Güdükbay

Jury Members: Can Alkan, Çiğdem Gündüz Demir

Contents

Project Specification 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. This popularity could be attributed to improvements in video game graphics [1]. The improved graphics technology gives the opportunity of designing more challenging and exciting video games to game producers. Today, Counter-Strike, Dota2, League of Legends, Fifa and NBA games could be given as some of the best examples of such games [2]. These games are so popular that every year, eSports tournaments related to them are organized all around the world [3, 4, 5].

The number of people who play video games is increasing thanks to such popular games [6, 7]. 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 [7]. In the GVC industry, Twitch is the leading platform, accounting for 54% of the gaming video content platform revenue in 2017 [8]. In second place, YouTube is following Twitch. Even though these platforms are widely popular, users can still face issues in watching game videos in high quality. This 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 [9, 10]. 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 in which users can watch game videos in higher quality. Initially, we are planning to provide chess, go, Age of Empires [11] and Among Us [12] 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. Description

Upfix is an application that provides the functionality of improving the video watching experience of users by converting the videos to higher quality ones. The advantage of this application is that the users do not have to stream/watch videos at a high resolution and spend data while doing so, but instead they get to minimize their data usage and still watch the video at a higher resolution. The users also will not be affected by the fluctuations in the quality of their WiFi signals and will not experience any interruption while they are immersed in the video.

The users have the option to choose from a web based application and a desktop application. If their internet connection allows them, they can use the web application to upscale their videos on the server. The users are able to provide a YouTube or Twitch link to the video that they want to be upscaled. Then the application will generate a higher quality version of the video. This provides a way for users to enhance their engagement with their favorite gaming videos.

Users can also download the desktop application in order to upscale their videos by using the local resources of their computers. If their internet connection is not reliable, they may prefer this alternative.

The application uses deep learning technology in order to upscale the video quality. Neural networks are trained with various datasets that each further improve the ability of the neural network, which is to improve the resolution of videos, specifically the ones that are related to gaming. Convolutional Neural Networks [13], Recurrent Neural Networks [14] and Generative Adversarial Neural Networks [15] could be used as architecture. After upscaling, we can apply image processing techniques such as sharpening or histogram equalization to improve the look further.

1.2. Constraints

1.2.1. Implementation Constraints

- The version control will be done via Github [16].
- The deep learning models and image processing parts will be written in Python 3.
- Tensorflow 2.0 [17] and Pytorch 1.6.0 [18] libraries will be used to create artificial neural networks.
- Google Colab [19] will be used to train the models efficiently.
- To perform image processing tasks, OpenCV 4 [20] library will be used.
- Frontend of the website will be written using JavaScript and React framework [21].
- Backend of the website will be written in Java using Spring framework [22].
- Desktop version of the application will be written in Java.

1.2.2. Economic Constraints

- For the project web page, github.io will be used and it is for free.
- All the libraries which will be used are open source; therefore, they are for free.

- Google Colab which will be used to train models is for free.
- Frameworks which will be used in developing the project website are for free.
- For deployment of the project, domain and host costs will be considered.

1.2.3. Technological Constraints

- Internet connection will be needed for some of the services.
- An Nvidia graphics card for CUDA enabled processing will be needed when a user prefers inferring the data with GPU.

1.2.4. Language Constraints

- The language of the system will be English.

1.2.5. Environmental Constraints

- There will be no environmental constraints because our project will be designed as a software product which will not cause any physical pollution.

1.2.6. Sustainability Constraints

- Deep learning technology has been rapidly improved. Therefore, the models which we will train can be improved with new developments.
- We will try different methods to train models. According to users' feedback, we will try to develop better solutions.

1.2.7. Data Constraints

- Some of the videos will be taken from YouTube and Twitch.
- High quality images that are going to be used in training and image processing will be collected by us.

1.2.8. Usage Constraints

- The usage of the application will be similar to current video sharing applications. Therefore, it will not be difficult to understand.
- There will be a user manual to teach users how to use the application.
- There will be proper success, error, and warning messages to respond to users' requests.

1.2.9. Ethical Constraints

- We will not share user’s data with third party applications when the user prefers using the web application.

1.3. Professional and Ethical Issues

The neural network will not be used for any other purpose than the one declared. The manipulation of the data will be in compliance with the KVKK and the GDPR regulations.

User data of the application will not be shared with or sold to any third party.

This project will be implemented using open source libraries and tools to refrain from possible issues with regard to software copyright.

2. Requirements

2.1. Functional Requirements

Desktop Application:

- The user can upload a video to be upscaled.
- The program can upscale a video using different algorithms with GPU or CPU offline.
- The user can watch the upscaled video.

Web Application:

- The user can upload a video to be upscaled.
- The user can give a link to a video to be upscaled.
- The program can find a video from the given link.
- The program can upscale a video using different algorithms on a server online.
- The user can watch the upscaled video.
- The user can download the upscaled video.

2.2. Non-functional Requirements

2.2.1. Security

- The system should not disclose the data of its users to third parties.

2.2.2. 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.

2.2.3. Cost

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

2.2.4. Performance

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

2.2.5. 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.

2.2.6. 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.

3. References

[1] "Video Game Market Size, Share: Industry Report, 2020-2027." *Video Game Market Size, Share | Industry Report, 2020-2027*, May 2020, www.grandviewresearch.com/industry-analysis/video-game-market.

[2] "Steam: Game and Player Statistics." *Welcome to Steam*, store.steampowered.com/stats/Steam-Game-and-Player-Statistics?l=english.

[3] Hore, Jamie. "The Biggest Esports Games." *The Loadout*, 7 Sept. 2020, www.theloadout.com/biggest-esports-games.

[4] Esguerra, Tyler. "League of Legends, CS:GO, and Dota 2 Were the Most-Watched Esports of 2019." *Dot Esports*, 7 Feb. 2020, dotesports.com/league-of-legends/news/league-of-legends-csgo-and-dota-2-were-the-most-watched-esports-of-2019.

[5] Kobek, Patrick. "League Of Legends, CS:GO, And Dota 2 Were The Most Watched Esports Last Year." *TheGamer*, 9 Feb. 2020, www.thegamer.com/league-of-legends-csgo-and-dota-2-most-watched-esports-2019/.

[6] "Hours of Gaming Unboxing Videos on YouTube Watched on Mobile." *Think with Google*, Google, www.thinkwithgoogle.com/marketing-strategies/video/gaming-video-watch-time-statistics-on-youtube/.

[7] Guttmann, A. "Topic: Gaming Video Content Market." *Statista*, www.statista.com/topics/3147/gaming-video-content-market/.

[8] "2018 Video Game Industry Statistics, Trends & Data - The Ultimate List." *WePC*, 14 Aug. 2020, www.wepc.com/news/video-game-statistics/.

[9] Counter-Strike: Global Offensive Broadcast. (n.d.). Retrieved October 06,2020,from https://developer.valvesoftware.com/wiki/Counter-Strike:_Global_Offensive_Broadc ast

[10] Dota2. (n.d.). Retrieved October 06, 2020, from http://www.dota2.com/watch/

[11] Age of Empires Franchise - Official Web Site. (2020, August 27). Retrieved October 06, 2020, from https://www.ageofempires.com/

[12] Among Us on Steam. (n.d.). Retrieved October 06, 2020, from https://store.steampowered.com/app/945360/Among_Us/

[13] A. Kappeler, S. Yoo, Q. Dai and A. K. Katsaggelos, "Video Super-Resolution With Convolutional Neural Networks," in IEEE Transactions on Computational Imaging, vol. 2, no. 2, pp. 109-122, June 2016, doi: 10.1109/TCI.2016.2532323.

[14] Huang, Y., Wang, W., & Wang, L. (1970, January 01). Bidirectional Recurrent Convolutional Networks for Multi-Frame Super-Resolution. Retrieved September 26, 2020, from http://papers.nips.cc/paper/5778-bidirectional-recurrent-convolutional-networks-for-multi-frame-super-resolution

[15] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Shi, W. (2017, May 25). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. Retrieved September 26, 2020, from https://arxiv.org/abs/1609.04802

[16] Build software better, together. (n.d.). Retrieved September 26, 2020, from https://github.com/

[17] TensorFlow. (n.d.). Retrieved September 25, 2020, from https://www.tensorflow.org/

[18] PyTorch. (n.d.). Retrieved September 25, 2020, from https://pytorch.org/

[19] Google Colab. (n.d.). Retrieved September 25, 2020, from https://colab.research.google.com/

[20] OpenCV. (2020, September 21). Retrieved September 25, 2020, from https://opencv.org/

[21] React – A JavaScript library for building user interfaces. (n.d.). Retrieved September 26, 2020, from https://reactjs.org/

[22] Spring makes Java simple. (n.d.). Retrieved September 26, 2020, from https://spring.io/