

DEPARTMENT OF COMPUTER SCIENCE

ITCS 6166 COMPUTER COMMUNICATION AND NETWORKS

GROUP 13

Mahija Shivani Dadisetty- 801308384

Lakshmi Srujitha Gali- 801307126

Sai Vignesh Ramisetty-801282784

Poornima Pulakandam-801263499

IMAGE DRIVEN TALKING HEAD

**PROBLEM STATEMENT**

The goal of this project is to create a web interface that can generate a talking head in response to a driving image or video. The talking head should be able to move in sync with the driving image or video and produce a realistic output.

**INTRODUCTION**

Computer communication and networking have revolutionized the way people communicate with each other, and one of the most interesting areas of research is the development of talking heads that can mimic human expressions and movements. In this project, we aimed to deploy an image-driven talking head using WebRTC, a real-time communication protocol that enables peer-to-peer communication between web browsers.

**MILESTONES**

* Understanding the machine learning model that can generate a talking head in response to a driving image or video.
* Understanding a real-time video processing pipeline that can generate a talking head in sync with the driving image or video.
* Setting up the environment for the model to run without any errors.
* Designing and implementing a web interface that allows users to upload images or videos and integrate with the trained model to give the required output.

**REQUIREMENTS**

There are many requirements for this project to run which are listed below:

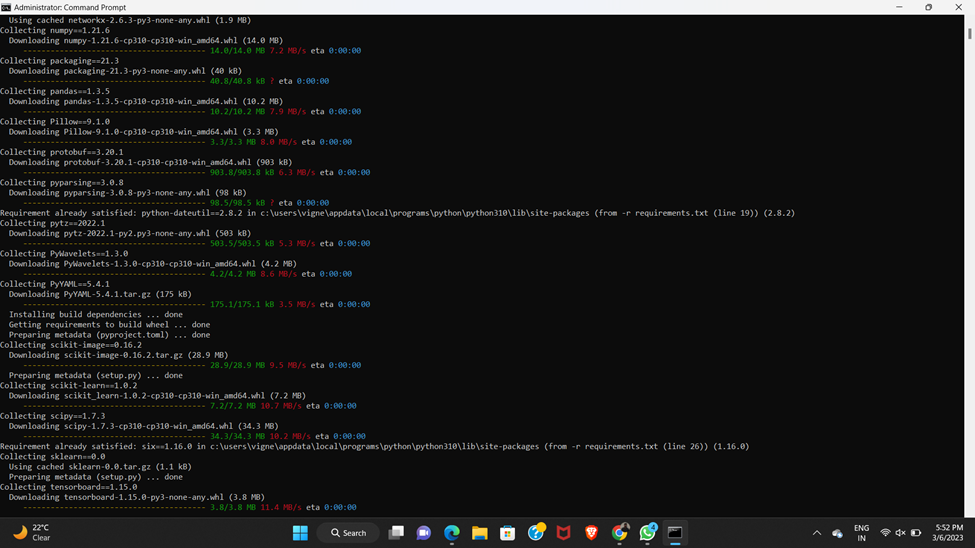
1. Python version 3.9 along with pip
2. All the libraries listed in requirements.txt file
3. Other extra libraries ( Ktinker, flask, etc...)
4. Pytorch latest version (for torch vision)
5. Nvidia GPU (mandatory for CUDA libraries)
6. Linux is optional

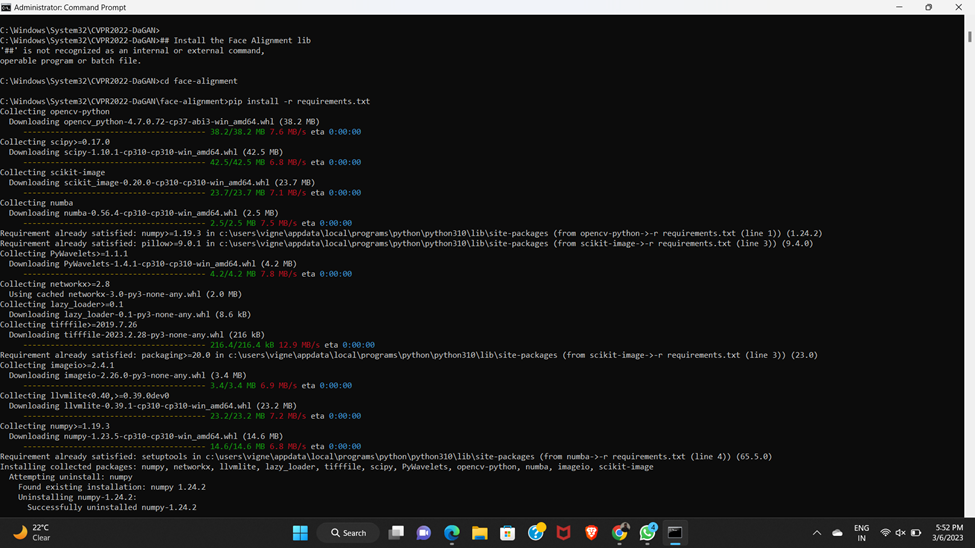
**IMPLEMENTATION**

The project was implemented using Python, various libraries such as PyTorch, OpenCV, Flask, and HTML, and the WebRTC communication protocol. The web interface was developed using Flask, which allows users to upload images or videos for training the model. In our systems, the machine learning model was implemented using PyTorch, and the output of the model was processed in real-time using OpenCV. The final output was transmitted in real-time using WebRTC to a web browser.

The implementation of this project is the main goal. So first we tried developing a graphical user interface to get the video input in the form of a series of images and train the model using that data and get the output. But that process is complicated, and our systems are also not compatible for working on the project. And the output takes time to generate to use realtime data to get the output video. So, we switched to developing a local web interface where we can give a video and image as input and get a generated video as output. So, this is much more compatible and easy to use and understand.

The first step is to run the trained model on our PC and see the working of the project. For this the first step is to download python, preferably version 3.9. Then the second step is installing all the dependencies and libraries that are listed in the requirements.txt file using command prompt.





These libraries are not sufficient to run the project. We mainly need NVidia and CUDA libraries to run the project. So, we then started to run the project and find out which libraries to install. There are so many libraries to install and we also used different versions of python and libraries to make the project work. Finally, after trying to run the model 8-9 times and rectifying the errors at every step, we are successful in making the project work. We used Linux platform which is recommended. We used a pre-trained checkpoint to run the project. Because even if we train the model again, the output and accuracy will be similar. So, the model is at a position where we can give inputs in a system function and get the output in results.mp4 file. The working model till this part is shown in the below YouTube link.

Link for command prompt system working: [command prompt model working](https://youtu.be/psLh8MZWj-I)

So, the next step of the project is to develop a web interface which takes the .png and .mp4 inputs and gives a processed .mp4 file as output. We achieved this using html and python coding. We then started developing a simple web interface which contains two input tabs and a submit button. These input tabs take .mp4 and .png inputs. So, we connected the html file to the vig.py file which we created to run the system function in the background. We used different extra libraries like Tkinter, flask, etc to make this work and embed the web interface with the trained model. So, we can upload a source image and driving video and the result will be stored as results.mp4 which will be opened directly using the vlc media player in our laptop. We can develop an attractive interface if we have more time, but this is the best we can do in the given time. The user interface and its working is shown in the YouTube link below. This is also the final demonstration.

Link for working web interface of the model: [CCN Working web interface](https://youtu.be/942Altmj4fI)

This is the complete web interface of the project. This concludes the deployment of the project in the local server.

**CHALLENGES**

* The project was challenging due to our lack of experience with deployment and the many obstacles we encountered.
* We found every step of the project to be a task in itself.
* Out of all the tasks, iteration was one of the most significant.
* Finding the right version of Python and importing the necessary libraries was a very difficult aspect of iteration.
* We had to switch between three different PCs because some did not support the required Python versions.
* The main challenge is finding out that the model only runs with the help of nvidia GPU and CUDA libraries. Finding a PC to run the project is also a task we have to overcome.
* One PC became overloaded with different Python versions, which further complicated the process.
* Developing a web-based interface was another significant challenge that required a lot of time and effort.
* The main challenge faced during the implementation of this project was the integration of the WebRTC protocol into the project. WebRTC is a complex protocol that requires specialized knowledge and expertise to implement.

**CONCLUSION**

In conclusion, the image-driven talking head deployment project using WebRTC has successfully achieved its goal of creating a web interface that can generate a talking head in response to a driving image or video and transmit the output in real-time to a web browser. The project demonstrates the potential of using computer communication and networking to enhance human interaction and create more realistic virtual environments. However, there is still room for improvement, and future work could focus on improving the accuracy and performance of the model and incorporating more advanced features, such as lip syncing and facial expressions.