**Yash Sanjeev Kolhe**

**Vishwas Venkatram**

**PROJECT TITLE:**

Implementation of Face NeRF: Audio-driven talking face generation based on dynamic NeRF (Audio Driven Neural Radiance Fields)

**PROJECT SUMMARY:**

We are presenting an audio-driven talking head method that maps the audio features to dynamic NeRFs for face image rendering without any loss.

We decompose the neural radiance fields of human portrait scenes into two branches to model the head and torso deformation respectively, which helps to generate more natural talking head results

Trained a talking face model based on our data and tested the model with the new audio. We also tested the model with an entirely new dataset.

**PROJECT DESCRIPTION:**

Our work is composed of the modelling of the deformed human heads(Head) and upper bodies(Torso) by neural scene representation. We extracted the audio features and mapped the audio feature to the neural radiance field. We then rendered the visual faces.

We had generated the speaking style of the person chosen as target. We have tried to achieve this by separately estimating the heads and upper bodies with auto parsing method. Transformed the volumetric features into the canonical space. In the transformation of the volumetric features into the novel canonical space, the head and upper bodies are rendered separately by their models giving us the best results.

We implemented the execution of a conditional radiance field of talking head of a person using a conditional implicit function with an additional audio code as input. In order to extract the semantic audio meaningfully we employed the DeepSpeech model to predict the 29 dimensional feature code for each 20ms audio clip and have used only the head pose information for network.

The purpose of rendering the head is that it is dynamically moving frame to frame. The code also uses the implementation of Principal Component Analysis for extracting the most important features to train on.

The rendering process is accumulating the sampled density and the RGB value for the pixels in head part. Then the torso part will find and fill the missing gap by predicting the pixels as it is helpful to model the head and upper body movements.

For the dataset we had taken short video with audio track of 3-5 minutes for training . We had observed that video above 25 fps does not get processed. There are three main steps to preprocess the training dataset:

1. We PASRES to label the different semantic regions for each frame.
2. We have applied the multi-frame optical flow estimation method to get dense correspondences across video frames.
3. We have detected the landmarks for the images from github.
4. We had parsed the face and extracted background image
5. We have saved the training images and estimated the head position made in the step3.
6. We saved the params and had it configured.

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, application, Word

Description automatically generatedGraphical user interface, text, application

Description automatically generated

For the implementation we have used Google Colab, in which we had created the conda environment to run the commands for pytorch framework which we are using. We train each model for 10k iteration. In each iteration we had randomly created the batch of 2074 rays passing through image pixels. We trained the network with RTX 20270 and train each model for 10k iterations for 5 minute video. It takes close to 10 hours to train two NeRFs.

**PROBLEMS FACED:**

Initially started the work on Windows OS using the Jupyter notebook. There were troubles in executing the environment.yml file that installs the various packages into the conda environment.

We then installed a virtual box for Ubuntu on Windows and were able to successfully proceed till the training stage. When there was a requirement of GPU in training phase, we faced an error as Ubuntu on a virtual box had no access to the GPU .

We then turned to Google Colab where we used MiniConda to install a conda environment and installed all the packages into it. We ran into multiple versioning issues with Colab in which case we had to individually install various packages. We then managed to execute the code and modeled the two individual neural radiance individual one for the head part and one for the torso part. We had divided the part into head and torso for face parsing and have trained the head part by parsing map.

**Results:**

1.With given the dataset we had got the result as

A person wearing a suit and tie

Description automatically generated with medium confidence A person wearing a suit and tie

Description automatically generated with medium confidence A person wearing a suit and tie

Description automatically generated with medium confidence A person wearing a suit and tie

Description automatically generated with medium confidence A person wearing a suit and tie

Description automatically generated with medium confidence A person wearing a suit and tie

Description automatically generated

We have rendered a total of 299 images from the video to obtain the result. Here size of file was 25.4 mb and the pixel obtained is having low resolution as we have restricted the training for only 10k iterations due to the limited capacity of resources and time.

2.Using the new dataset we had got the following we got the results:

A person with blonde hair

Description automatically generated with low confidence A picture containing text, indoor, shelf, person

Description automatically generated A person with blonde hair

Description automatically generated with low confidenceA picture containing text, indoor, shelf, book

Description automatically generated A person with blonde hair

Description automatically generated with low confidence A picture containing text, indoor, person, shelf

Description automatically generated

Here we have rendered 76 images from the video to obtain the result.

This method can generate talking head frames with freely adjusted viewing directions and various background images. Each row from left to right, original frames from a video, reconstructed results with audio and pose from the original video.

Code for implementation is placed in the following GitHub repository.

<https://github.com/VishLivecoder/Implementation-of-AD-NeRF>

<https://github.com/curiosity148/Computer-Gaming-Project>

Code for the original research work can be found in the following GitHub repository.

<https://github.com/YudongGuo/AD-NeRF>

**Citations:**

@inproceedings{guo2021adnerf,

title={AD-NeRF: Audio Driven Neural Radiance Fields for Talking Head Synthesis},

author={Yudong Guo and Keyu Chen and Sen Liang and Yongjin Liu and Hujun Bao and Juyong Zhang},

booktitle={IEEE/CVF International Conference on Computer Vision (ICCV)},

year={2021}

}