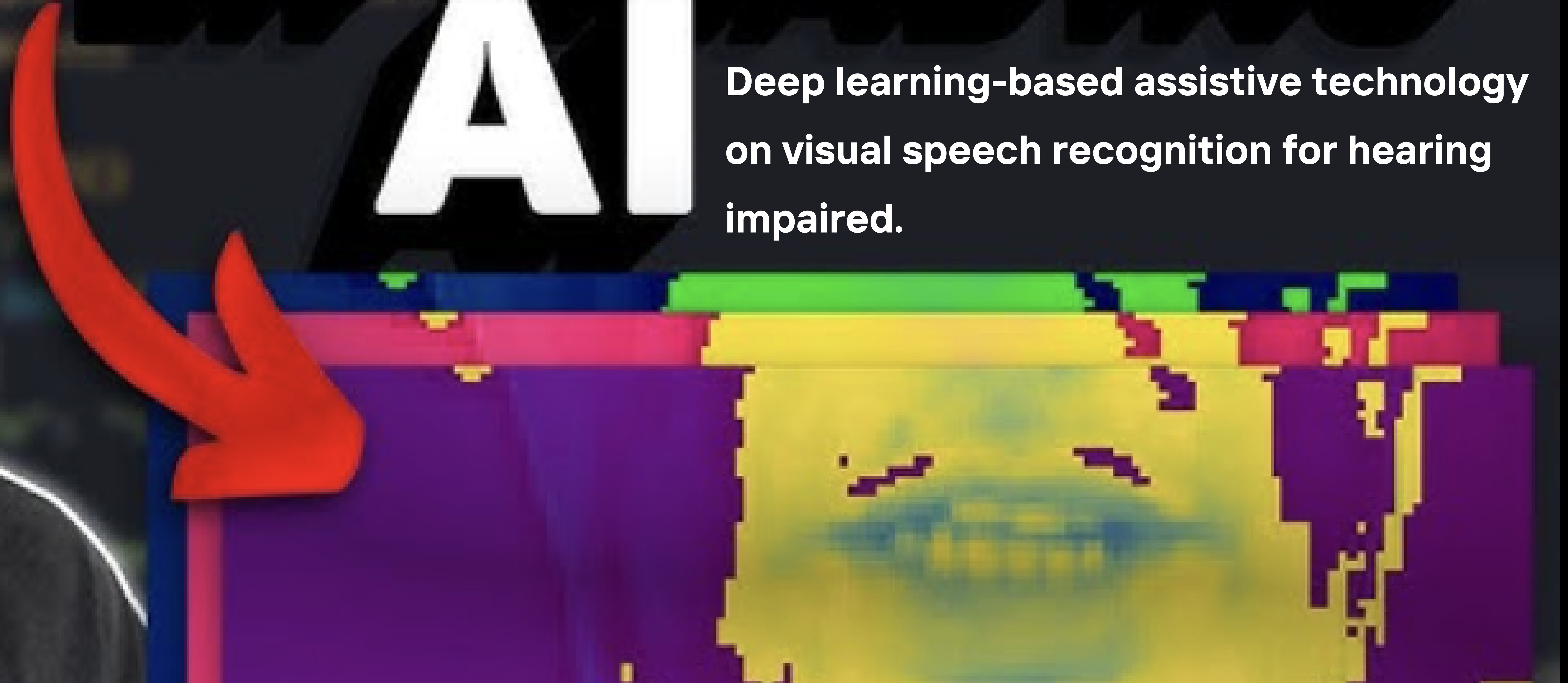
LIPREADING AI

Deep learning-based assistive technology on visual speech recognition for hearing impaired.



Introduction

Speech is a continuous sound signal the fundamental mode of human interaction whereas hearing impaired people recognize words spoken by reading a lip. Some of the widely used practices by hearing-impaired students for learning is sign language.

Challenges are listed from the perspective of the facilitator and student.

Teacher's perspective

- Lack of trained sign language teachers
- Lack of awareness of new technologies.

Students perspective

- Lack of user-friendly devices.
- Difficulty in interpretation.
- High-Cost Assistive Devices.
- Lack of e-learning content with subtitles.

Reference: https://www.sciencedirect.com/science/article/pii/S2666307422000031

Project Overview

Goal:

The project aims to develop a lip reading machine learning model capable of deciphering lip movements and converting them into text.

By leveraging deep learning techniques, the goal is to create a system that can assist individuals with hearing impairments in understanding spoken language through visual cues.

Objective:

- Develop a data preprocessing pipeline to extract relevant features from video data.
- Design and implement a deep learning model architecture to decode lip movements accurately.
- Train the model on 1 speaker dataset to achieve high accuracy in lip reading.
- Evaluate the model's performance and implement a streamlit application to present the model.

Libraries and Resources

Libraries:

- List of libraries required for the project (OpenCV, TensorFlow, cv2, matplotlib, imageio, gdown numpy, streamlit)

Dataset:

- The GRID Audiovisual Sentence Corpus is a widely used dataset in the field of lip reading and audiovisual speech recognition.
- It consists of audio and video recordings of 34 speakers, each uttering over 1000 sentences.
- We are considering only 1 speaker's videos and alignments to train our model.

Additional Resources:

- Streamlit: Potential tool for developing user-friendly interfaces for deploying and testing the lip reading model.

Reference: https://spandh.dcs.shef.ac.uk/gridcorpus/

Data Processing

Video and Alignment Preprocessing:

- **Load_Video Fn -** Get the video convert it into frames - loop through each frame and capture the lip section using a static slicer.
- **Vocab_v -** Every single character - “abcdefghijklmnopqrstuvwxyz'?!123456789 ”
- **Load_alignments Fn -** Load up our alignments with vocab - split each line and remove the sil 'silence' from the file - char_to_num and num_to_char conversion.
- **Load_data Fn -** This function will simultaneously load our data and the alignment and return frames and alignments.
- **Save as GIF -** All the frames from the video that are going as input tensors to the model are combined and saved as a GIF - To visually understand what is being input into the model.

Reference: https://spandh.dcs.shef.ac.uk/gridcorpus/

Building the Data Pipeline

Overview of Creating the Data Pipeline:

UsingTensorFlow's Dataset API to construct an efficient data pipeline for processing video and text data.

- **Data Extraction -** Grab the data from the s1 folder with an extension of .mpg format
- **Shuffle -** Randomizes the order of elements in the dataset, reducing bias and improving model generalization.
- **Padding -** Created a padded batch - each alignment with 40 and each frame with 75 by making it of a standard length.
- **Preload** the data when the ML model is learning for passing the next set.
- **Split** for training and testing.

Reference: https://github.com/nicknochnack/LipNet

Model Architecture

Description of Model Architecture:

The lip reading model architecture comprises 3D convolutional layers, LSTM layers, and Dense layers.

- 3D Convolutional Layers: Extract spatial and temporal features from the video data.
- LSTM Layers: Capture temporal dependencies and sequence information.
- Dense Layers: Predict characters based on the extracted features.

Introduction of CTC Loss Function:

- The Connectionist Temporal Classification (CTC) loss function is utilized to handle sequence-to-sequence mapping tasks where the input and output sequence lengths may vary.
- It is particularly suitable for tasks like lip reading, where the alignment between input (visual) and output (textual) sequences may not be one-to-one.

Model Layers:

Conv3D(128..), Conv3D(256..), Conv3D (75..), TimeDistributed(Flatten), Bidirectional(LSTM(128..), Bidirectional(LSTM(128..), Dense.

Reference: https://github.com/nicknochnack/LipNet

Training and Evaluation

Training Process:

- Model Compilation: The model is compiled with the Adam optimizer and the custom CTC loss function defined earlier.
- Callbacks: Two callbacks are employed during training: ModelCheckpoint to save model weights and LearningRateScheduler to adjust the learning rate after epoch 30.
- Training Epochs: The model is trained for 50 epochs using the specified optimizer and loss function.

Evaluation: After training, the model's performance is evaluated using the testing dataset.

Potential Improvements: Based on the evaluation results, the model can be trained on the other sets of data available on the GRID dataset, and potential improvements to the model's architecture or training process can be explored for enhanced performance in lip reading tasks.

DEMO!

References

Links that were used for reference to implement this project.

Video Reference: https://www.youtube.com/watch?v=uKyojQjbx4c&t=3607s

Original Paper: https://arxiv.org/abs/1611.01599

Another Paper: https://www.sciencedirect.com/science/article/pii/S2666307422000031

Associated Code for Paper: https://github.com/rizkiarm/LipNet

Dataset: https://spandh.dcs.shef.ac.uk/gridcorpus/

Code Reference: https://github.com/nicknochnack/LipNet

ARS Tutorial: https://keras.io/examples/audio/ctc_asr/#model

AI Tool: https://chat.openai.com/

THANK YOU!