**LIP READING FROM VIDEO FOR TEXT EXTRACTION AND IMPROVED COMMUNICATION**

**Team Members:**

Nikhil Rao Tannier(11615781)

Sai Kiran Reddy Kancharla(11605339)

Bhargav Ram Pushadapu(11647795)

Hemanth Kumar Kakani(11655430)

**Instructor**

Sayed Khushal Shah

CSCE 5310 Section 002 – Methods in Empirical Analysis

Department of Computer Science

Sayed.shah@unt.edu

University of North Texas

**INTRODUCTION:**

Decoding textual content from a speaker's lip movement is called lipreading. The venture was traditionally divided into two degrees: designing or gaining knowledge of visual cues, and prediction. However, greater latest deep lip reading strategies are quit-to-cease trainable and do word categorization in place of sentence-degree series prediction. By using LipNet, an quit-to-cease sentence-level lipreading model this venture seeks to overcome this constraint. LipNet is an architecture advanced ordinarily the usage of CNN and LSTM, for specifically spotting text from videos. The model makes use of spatiotemporal convolutions, a recurrent community, and the connectionist temporal classification loss to convert a variable-length series of video frames to textual content.

**IDEA DESCRIPTION:**

The intention of the task is to create the advanced lipreading version LipNet, which can decode spoken phrases through observing mouth movements. To improve communique in a whole lot of instances, LipNet might be built to conduct sentence-level sequence prediction.

**GOALS AND OBJECTIVES:**

The primary goals and objectives of this project are as follows:

1. Develop LipNet: Build a sturdy give up-to-cease lipreading version that accurately converts video frames to text while capturing each visual and temporal traits.

2. Achieve High Accuracy: Develop LipNet to outperform existing models and human lip readers in sentence-level lipreading.

3. Assess Performance: Carry out thorough assessments to gauge LipNet's performance in word- and sentence-level lipreading, contrasting it with human performance.

4. We are planning to develop the LipNet Model, an application using User Interface to extend the paper.

**MOTIVATION:**

This research is driven by the need to improve voice recognition technology and improve communication for people with hearing problems. Correct sentence-level lipreading can help people communicate more effectively and has uses in a variety of industries, including security and accessibility. This can also help in further understanding, language translation for the videos and eventually generate the same video with lip sync of different languages, which can save a lot of time and effort in many scenarios, say movies made in different languages.

**SIGNIFICANCE:**

This project is significant because of the possible applications of sophisticated lipreading technology. It has the potential to empower people with hearing impairments, improve security through lip biometrics, and improve human-computer interface with powerful speech recognition systems. It also aids in the evolution of deep learning and computer vision.

**LITERATURE SURVEY:**

The existing research on lipreading, voice recognition, and deep learning techniques is primarily focused on generating better text. Lipsync technology research includes investigations in both computer graphics and audio processing. Researchers have investigated approaches for precisely coordinating lip movements with speech to improve the realism of virtual characters and animations. Machine learning developments have enabled automatic lipsync generation in this industry. Among the notable works is the application of deep learning algorithms and audio-visual datasets to increase lip sync accuracy. This is a complex study merging both computer vision and NLP.

**OBJECTIVES:**

The following are the primary goals of the project:

- Data collection and preprocessing for lipreading training.

- To create a LipNet, a spatiotemporal convolutional and recurrent model,

- Using the connectionist temporal classification loss to train the model.

- LipNet's performance will be evaluated and compared to human lip readers and existing models.

**FEATURES:**

This project's features include:

- the design and development of an end-to-end lipreading model.

- Combining spatiotemporal convolutions and recurrent networks.

- A thorough examination utilizing the GRID corpus dataset.

- A comparison of performance with human lip readers and word-level models.

**RELATED WORK (BACKGROUND):**

The majority of lipreading research has traditionally avoided deep learning, relying on extensive preprocessing, temporal processing or handcrafted vision pipelines.

Using hand-segmented phones and hidden Markov models (HMMs) on a small dataset, Goldschen et al. (1997) pioneered visual-only sentence-level lipreading. Afterwards, using a combination of hand-engineered features and HMMs to improve performance in noisy environments, Neti et al. (2000) accomplished the first audiovisual speech recognition at the sentence level on the IBM ViaVoice dataset. To illustrate the efficacy of their approach, Potamianos et al. (2003) reported word error rates that were both speaker-independent and speaker-adapted using the same dataset and the DIGIT corpus.

Using an HMM/GMM system with LDA-transformed mouth region characteristics, the study by Gergen et al. (2016) achieves speaker-dependent accuracy on the GRID corpus. Although it works well, there are still issues with motion feature extraction and generalization amongst speakers. Using deep learning, LipNet tackles these problems.

Attempts to anticipate complete phrase sequences in the field of deep learning-based lipreading have mostly concentrated on identifying words or phonemes. Multiple approaches include visual features for conventional speech-style processing, multimodal representations, and combinations of these. Sentence-level sequence prediction and varied sequence lengths are not well supported by spatial and spatiotemporal convolutional neural networks, which are based on VGG.

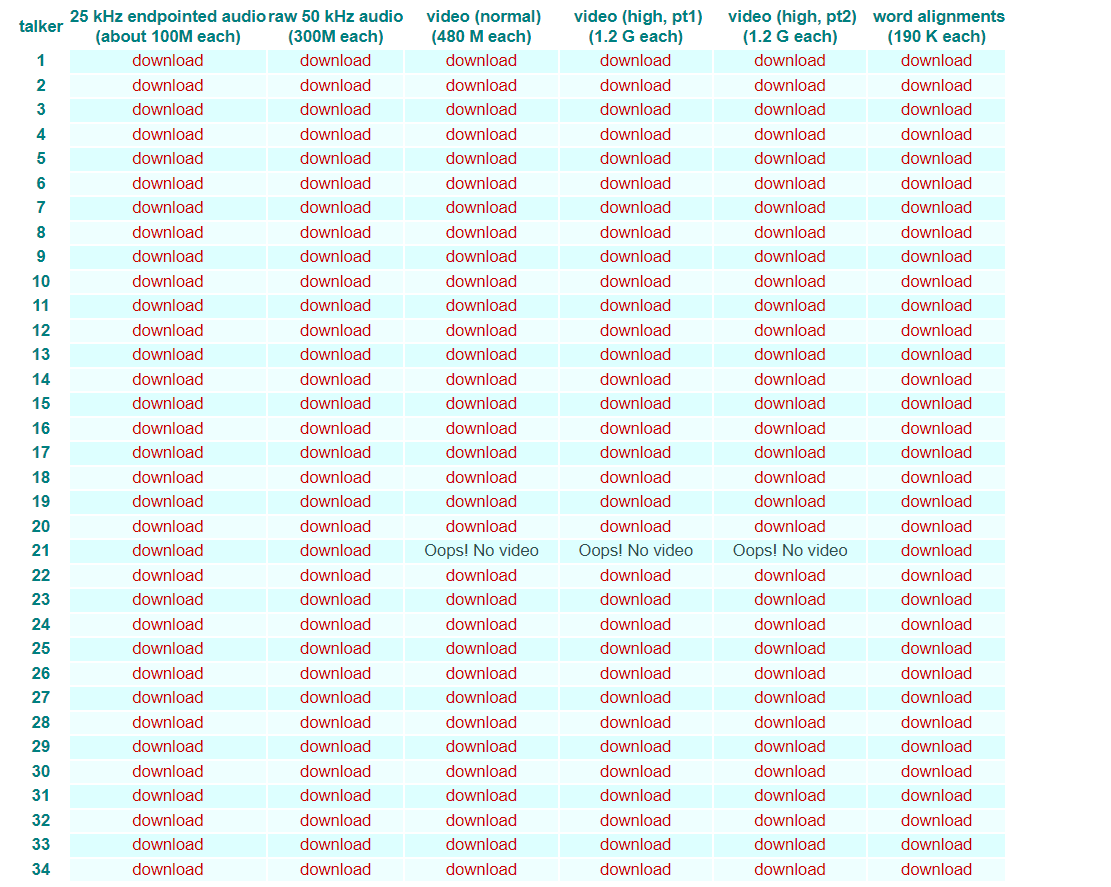
Wand and colleagues (2016) present LSTM recurrent neural networks for lipreading, but they do not address sentence-level sequence prediction or speaker independence. Garg et al. (2016) classified words and phrases using pre-trained VGG and the MIRACL-VC1 dataset, with a top model that achieved reasonable accuracy. The report highlights the ways in which lipreading has advanced simultaneously with deep learning, emphasizing the effect of recent advances in automated speech recognition in particular.

The first end-to-end model for visual speech recognition to provide sentence-level sequence prediction is LipNet. In other words, we present the first work that doesn't require alignments and is trained end-to-end using CTC. Given a sequence of photos as input, it generates a distribution over sequences of tokens.

**DATASET:**

We took a video data file from the below dataset for the LipNet Analysis

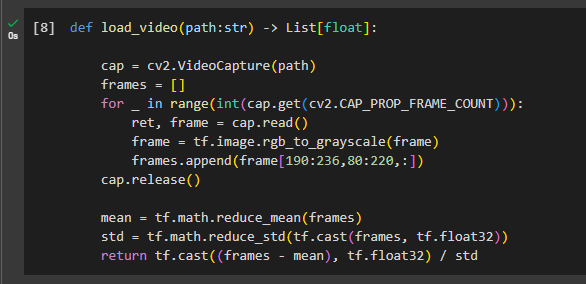
Later, the video is loaded from the url(grid dataset)

******

Next Step is pre-process the annotations, Annotations refers to sentences that the person in the video talk about.

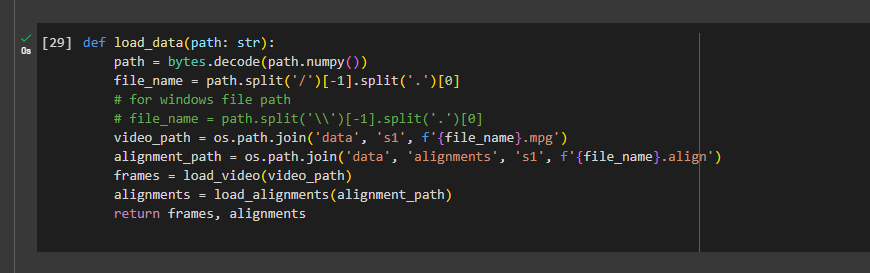
**Detail Design of Features:**

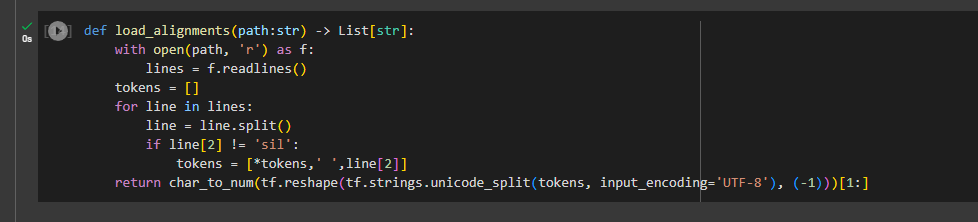
We have load video function to load video into array, it will loop through all the frames in the video and store them in an array to perform operations on them to extract the frame [190:236,80:220,:] that contains the mouth of the person in the video.



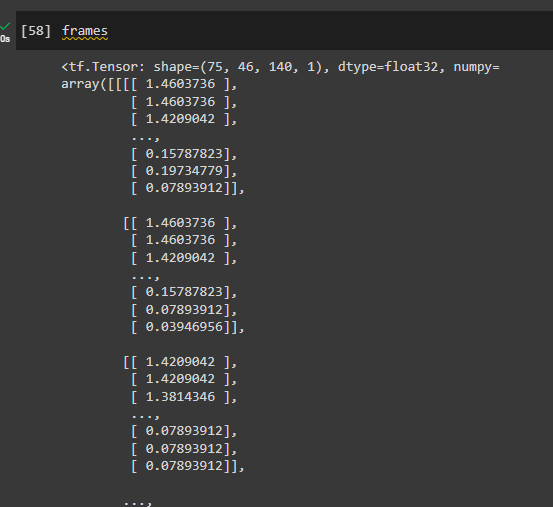
Next we have this StringLookup function in tensor flow keras library to convert string to integer and vice versa by passing in invert flag to true.

We have two functions here Load alignments and load data to preprocess the video and alignments.

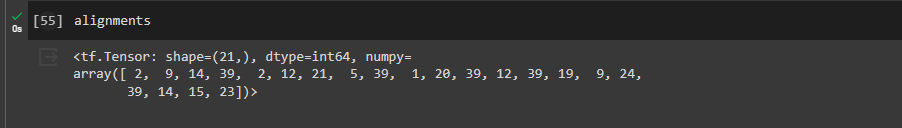




Here is the output of preprocessed frames.



Coming to alignments we have converted our alignment to encoded sequence which will be passed to machine learning model.

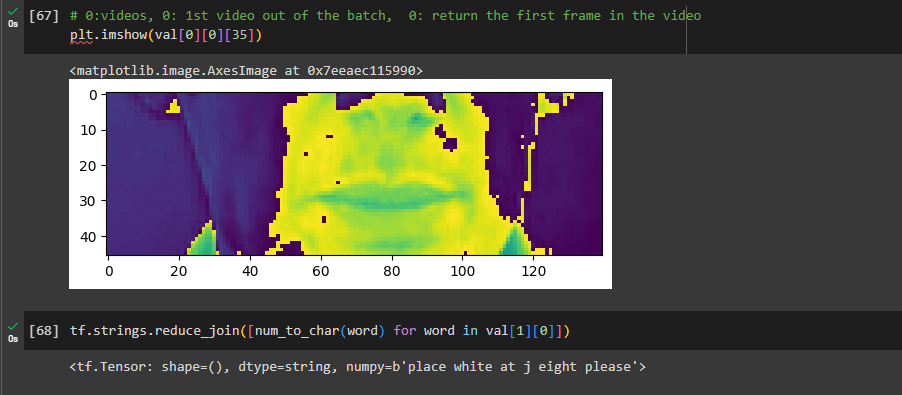


So clearly, we are extracting the features from the video and passing it to the Deep Neural Network.

**Analysis:**

Next step is defining data pipeline which is very crucial for Deep Learning Model. First Step is to create tensorflow dataset. We can see preprocessed images and annotations, which will be passed to the model using the above defined functions. We are splitting the data set into training and test dataset.

Training has 450 videos, and the test set has 50 videos.



**Deep Learning Model**:

We have designed a Sequential model in TensorFlow Keras for a 3D convolutional and bidirectional LSTM network. It consists of three 3D convolutional layers with activation and max-pooling, followed by a TimeDistributed layer and two Bidirectional LSTM layers with dropout. The final layer is a Dense layer with softmax activation, producing output for character prediction. The model is designed for a specific input shape of (75, 46, 140, 1) representing video frames, and it outputs probabilities for characters in the vocabulary. Adjustments may be necessary based on the specific requirements of your task.

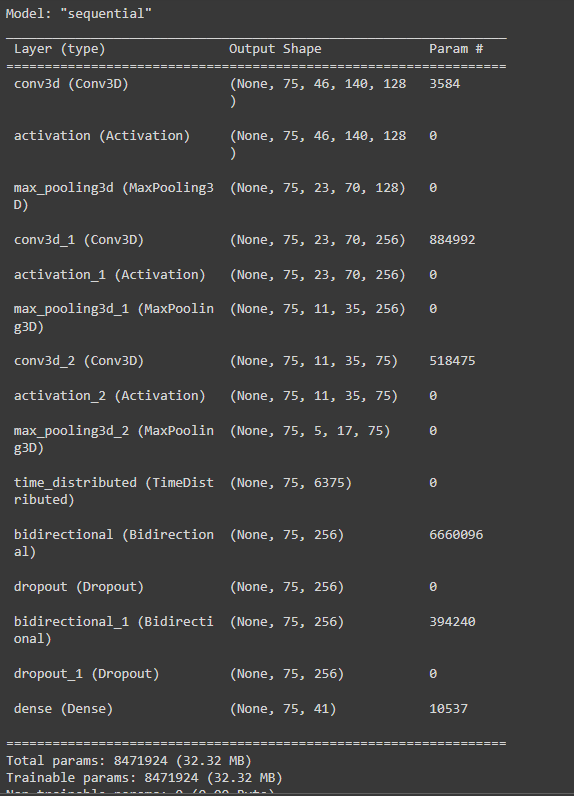
It defines a sequential model in TensorFlow Keras for a 3D convolutional and bidirectional LSTM network. It consists of three 3D convolutional layers with activation and max-pooling, followed by a TimeDistributed layer and two bidirectional LSTM layers with dropout. The final layer is a dense layer with softmax activation, producing output for character prediction. The model is designed for a specific input shape of (75, 46, 140, 1) representing video frames, and it outputs probabilities for characters in the vocabulary. Adjustments may be necessary based on the specific requirements of your task**.**

In the reference paper they have implemented Bi-directional Bi-GRU In our project we have implemented Bi-directional LSTM followed by ctc loss.

**Implementation:**

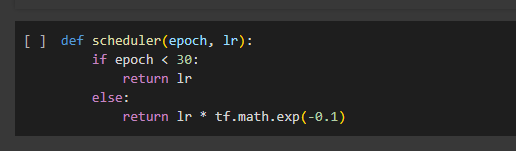


Here is the summary of the Model:



**Learning Rate:**

The scheduler function is a learning rate scheduler used in training a deep learning model. Here, for the first 30 epochs, it maintains the learning rate (lr) unchanged, and after that, it exponentially decays the learning rate by multiplying it with the exponential of -0.1.



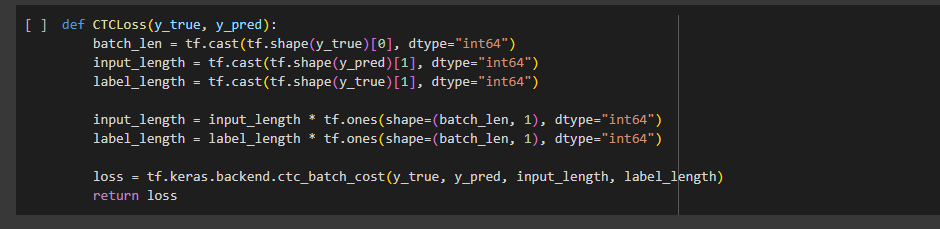
**Loss Function:**

Here, the below CTCLoss function computes the Connectionist Temporal Classification (CTC) loss between the ground truth labels (y\_true) and the predicted probabilities (y\_pred).

It involves casting and reshaping to ensure compatibility of the model

Then we used tf.keras.backend.ctc\_batch\_cost to calculate the CTC loss.

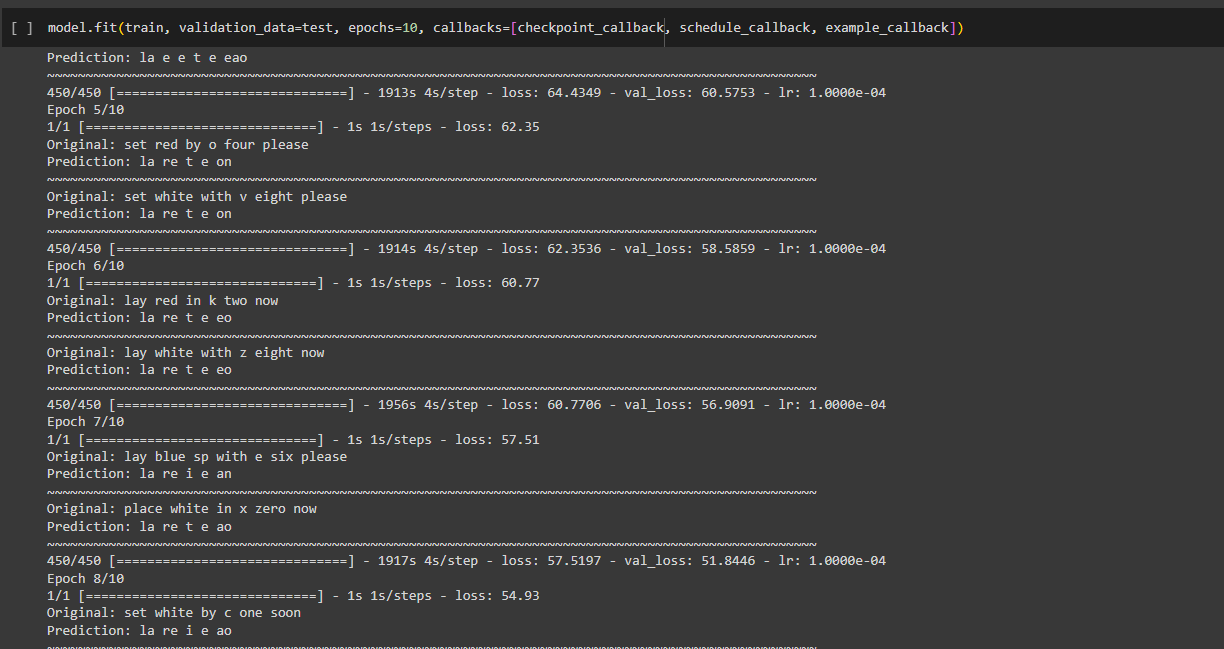
This loss function is commonly used in sequence-to-sequence model prediction for speech and handwriting recognition



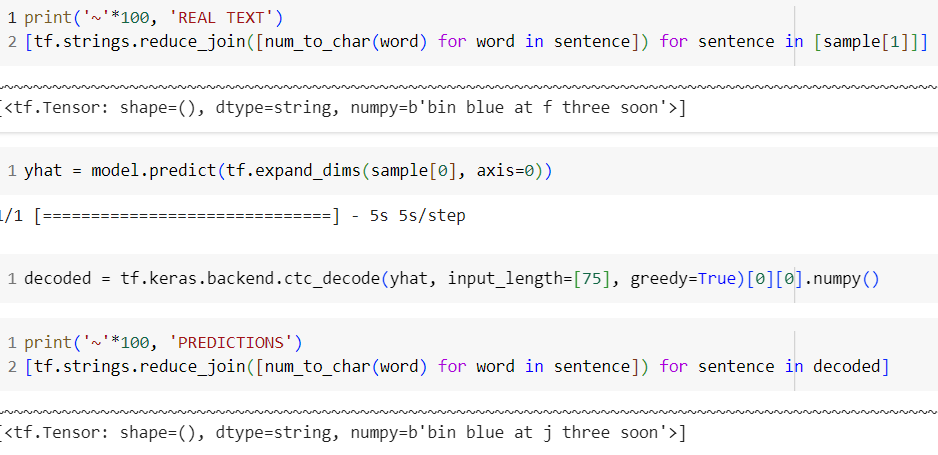
**Model Training:**

We trained the model using training dataset and validated it against a test dataset.

It runs for 10 epochs and utilizes callbacks, including ModelCheckpoint, LearningRateScheduler, and a custom callback for generating random examples during the training.



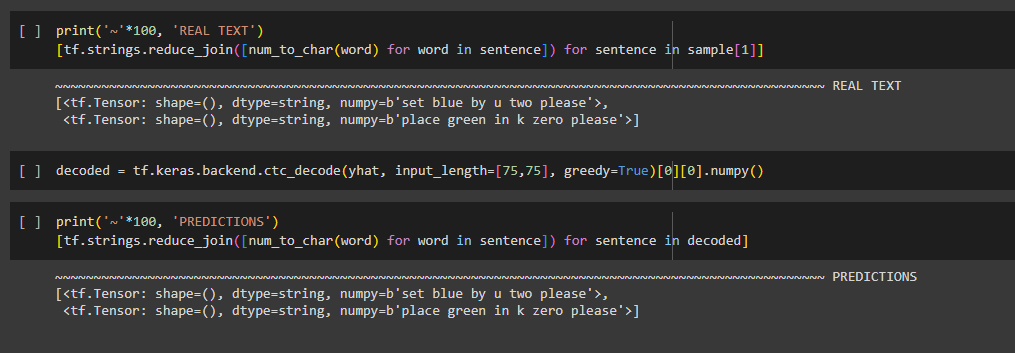
**Preliminary Results:**

****

The data is tested using a random video file from the input data. As we can below:

* the real text is getting an output of "bin blue at f three soon" &
* the predicted sentence is "bin blue at j three soon".

As we are training with just 10 epochs there is a mistake in predicted output but most of the sentence in the lipreading is similar. thus the Lipnet model here is predicting the sentences just by lipreading almost similar to original text.

****

Here, the data is trained with 100 epochs, as we can see the Real Text output is ‘set blue by u two please ’; ‘place green in k zero please’ and the predicted output is also same as the real text.

This model is predicted the exact expected output when we trained the model with high input data and training with more no of epochs for a training model. If we can train the model with different faces and different speaking conditions it will train well and expect the outputs with more accuracy.

**Project Management:**

**Implementation status report**

| **Wrok Complted** | **Name** | **Task** | **Contibution** |
| --- | --- | --- | --- |
| Sai Kiran Reddy Kancharla | Building Data Loading Functions, Make a Prediction | 100 |
| Hemanth Kumar Kakani | Create Data Pipeline, Test on video | 100 |
| Bhargav Ram Pushadapu | Design the Deep Neural Network, Make a Prediction | 100 |
| Nithil Rao Tannier | Setup Training options and Train, Test on video | 100 |

**Implementation to be done:**

In preparation for the upcoming second draft, significant progress has been made on the paperwork, laying a solid foundation for the subsequent phases of development. The focus now will be shifted towards enhancing the user interface (Ul) of the model, a critical component that plays a pivotal role in user interaction and experience. The incorporation of a well-designed Ul aims to not only elevate the overall aesthetic appeal but also to optimize functionality and usability. This next phase marks a crucial step in refining the project, ensuring that it aligns seamlessly with user expectations and industry standards. The commitment to delivering a polished and user-friendly interface underscores the project team's dedication to excellence and the pursuit of a successful end product.

**GITHUB:**

**References:**

[1. Assael, Y. M., Shillingford, B., Whiteson, S., & de Freitas, N. (2016). LipNet: End-to-End Sentence-level Lipreading. arXiv preprint arXiv:1611.01599. Retrieved from https://arxiv.org/abs/1611.01599]

[2. Wand, M., Gourina, O., and Zisserman, A. (2016). Lipreading with long short-term memory. In European Conference on Computer Vision, 282-298.]

[3. Chung, J. S., and Zisserman, A. (2016). Lip reading in the wild. In Asian Conference on Computer Vision, 87-103.]

[4. Almajai, S. Cox, R. Harvey, and Y. Lan. Improved speaker independent lip reading using speaker adaptive training and deep neural networks. In IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 2722-2726, 2016.]

[5. G. E. Dahl, D. yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing, 20(1): 30-42, 2012.]

[6. Dataset (<https://spandh.dcs.shef.ac.uk/gridcorpus/>)]

[7. Keras for Tensorflow (<https://keras.io/examples/audio/ctc_asr/#model>)]