Lip Reading for Word Classification

Piyush Onkar - 160050012
Apoorva Agarwal - 203050018

Problem Statement

- Our problem statement is based on prediction of word based on input video that does not have any audio based on movement of the lips.
- We implemented the code for this problem statement from scratch.
- Our main aim in this project was to increase the accuracy mentioned in the paper.
- For this project, we used pre-processing techniques such as face alignment, lip extraction.
- We also developed various models and compared the results between them

Original Paper

Link to the paper :
http://cs231n.stanford.edu/reports/2017/pdfs/227.pdf

Link to the implemented code of the paper :
https://github.com/adwin5/lipreading-by-convolutional-neural-network-keras

Link to the dataset :
https://www.kaggle.com/apoorvwatsky/miraclvc1

Extension to original Paper

- In extension to the original paper, we added lip extraction from the face as an extra pre-processing step.
- Paper only used Vgg16+BiLSTM. But we also developed multiple models and compared the results between them.
- Models are combinations for pretrained vgg16, resnet18, resnet50, resnet152 features and models BiLSTM and BiGRU.

Discussion of the technique

Dataset Preprocessing Steps :

1. Crop and Align Face
   The dataset had images of speakers with background, so we cropped and aligned faces using methods available in dlib.

2. Crop Lips :
   We used the shape_predictor_68_face_landmarks.dat file available with the dataset for detecting face landmarks in the image.There are 68 facial landmarks detected for the face. The landmarks from number 48 to 68 correspond to the lip region. After detecting landmarks for the lip region, the lip region is cropped.

Discussion of the technique

Dataset Preprocessing Steps :

3. Annotations
After the image processing we created the annotation json which had a list of image paths of each word and each for utterances.

4. Split Train and Test Dataset

Issue faced and how we resolved it
Issue: This process created a train file of approximate size of 5.5 GB. It was difficult to load this large into memory while training because of hardware restrictions.
Solution: As the file had a list of image labels and image numpy we split the list into lists of maximum length 1024. Then stored the smaller lists into multiple files. This created each file of size approximately 380 MB which was much faster and easier to load.

Discussion of the technique

Feature Generation:

We have used pretrained vgg16, resnet18, resnet50, resnet152 models to generate different features. The feature dimensions are 12800, 512, 2048, 2048 for vgg16, resnet18, resnet50, resnet152 features respectively.

To keep the file size small we have created different feature files for different feature types.

Issue faced and how we resolved it.
Issue: We planned to train vgg16 as it was done in the reference paper. But because of the large dataset and vgg16 being a deep model without gpu 1 epoch took 1hr to train.
Solution: Reference paper also used pretrained vgg16 so we generated features using pretrained vgg16 and saved them. Latter used classification models to get word labels.

Discussion of the technique

Developing Classification Model :

We had developed two classification models : Bi-LSTM and Bi-GRU and then added fully connected layer after that and then predicted the label for the sequence input to the model.

Evaluation Metrics :

The evaluation metrics consists of checking accuracy.
(Total number of correctly predicted labels for sequence / Total number of sequences)

Load Factor of the members

Work done by Apoorva :

- Created script to generate annotation. Wrote script for face alignment.
- Created script for splitting data into train and test data. Wrote scripts for breaking training and testing data into multiple files so that it can be used for training on machine having resource constraints.

Work done by Piyush :

- Wrote script for cropping lip region in the face. Created scripts for generating feature vector after passing through various pre-trained models.
- Developed scripts for generating Bi-LSTM and Bi-GRU models.

Self-Assessment

In initial submission proposal we mentioned to understand available code and implement Word Boundary but due various conflicts we created code from scratch.

Also we found word boundary paper difficult to implement and the number of resources were less.

We did preprocessing of dataset, generated cropped and aligned faces, extracted lips and created annotation files.