Prediction of real time speech from lip and jaw movements

Aditya Das
University of Texas at Arlington
aditya.das@mavs.uta.edu

Naman Jain Vimal Kumar
University of Texas at Arlington
namanjain.vimalkumar@mavs.uta.edu

Abstract

Here we are presenting our project on the prediction of real time speech from the movement of lips and jaws of a subject. This report shows the process how we are detecting faces and segmenting the facial features, before passing it to a combination of a CNN and a LSTM deep network model. The entire premise of the project is to provide real time closed captioning, if a camera is pointed towards a subject.

1. Introduction

Lip reading has innately been a difficult task for computers to deciphers. Though it could become the next big thing in the field of Human-Machine interaction, it's demanding nature and the fact that the subjects tend to have wildly varying speech, lip shapes, skin colors, speaking speeds, intensities, accents and jaw movements, make it difficult to develop generative lip-reading systems. Thus, most of the systems are limited to only certain words and, accents, speakers and commands.

1.2 Problem Statement

There has always been the absence of a normalized lip-reading system that can successfully work without overfitting to only certain types of speakers, accents or words in real time.

1.3 Related Work

The current lip-reading systems can be divided into three categories. The first category [14] takes in both visual as well as audio data. The frames of the visual data are matched to audio and the corresponding words are show as the result. Though a robust method to achieve speech detection, it is not a pure Computer Vision model. The AV models need a lot of metadata for training, like the start and stop times of each word etc. And gathering audio data during detection can also lead to issues because of noise and other sources.

The second category of lip-reading models have a non-rigid model that tracks the movements of the facial features. This models usually use an AAM [12][13] (Active Appearance Model) for the tracking the deformable face landmarks. This method is very robust in determining the exact angles and can easily be extrapolated to create a 3D model of the lips. But facial deformation tracking is a highly complex and resource demanding process. It is also susceptible to lighting conditions and variations to the texture of the subject.

The new third category which has emerged in recent times, tries to predict speech using both 2D images and depth map [10]. These systems have a rectification process to handle the motion of the speaker's face for a robust mouth region extraction. They also have an effective method for classification using a combination of appearance and motion descriptors. Thus, our project ventures into the new category, trying out new methods with a RGBD camera to model a depth map of the lip and jaw movements. We are also trying to predict phenomes instead of words or commands to remove the restriction of having a lip-reading system limited to certain words.

2. Problem Solution

2.1 Mouth Region Extraction

As illustrated in Figure 1, the proposed LR system takes a multi-modal video clip (i.e., 2D images and depth maps) representing a speech portion to recognize (e.g., a word, a command, a phrase, etc.). The system is divided into three phases:
1. Mouth Region Extraction
2. Description Computing
3. Classification

The process and the implementations of the phases and the future scope has been succinctly documented in this project report. A major part of the dataset collection was done on our own, because of the massive amount that we required for each of the image frames that we were capturing during the mouth region extraction phase.
This report will show the various algorithms implemented as well as the future scope of our project.

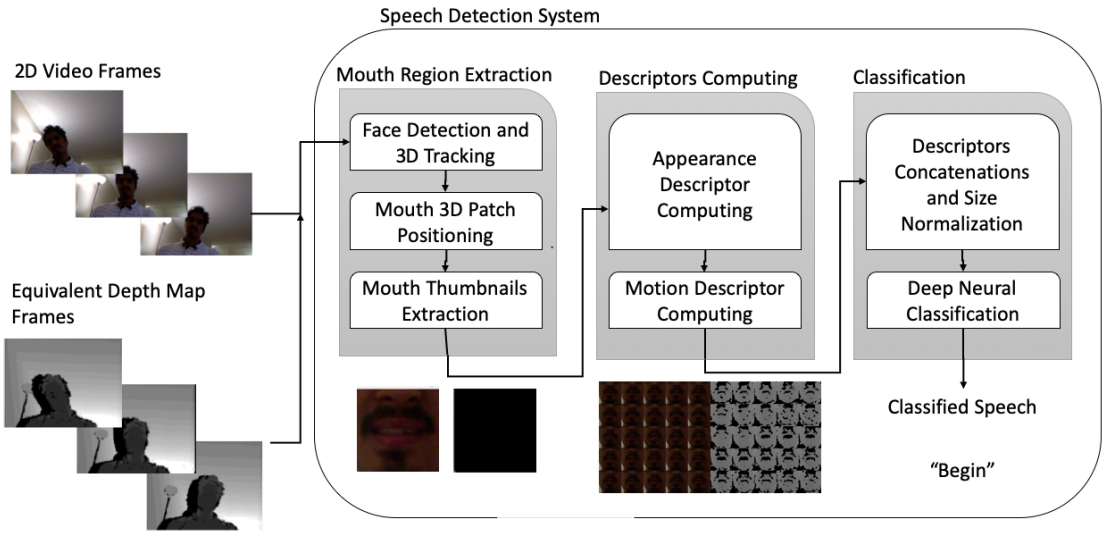


Figure 1

2.2 Face detection and 3D tracking

We needed a robust system to detect faces as close to real time as possible as missing a single frame out lead out to missing out on the word being spoken. After going through a fair number of face detection algorithms like, Viola Jones (which drops the frame rate to unacceptable levels) to Kanade Lucas, we finally implemented [1] using on OpenCV. This not only gave us a robust facial detector on the Kinect, but also the frame drop was acceptable enough as to not create any frame differences in the multi-modal video input. This SVM trained model gave us a very high accuracy and very few false positives.

2.3 Facial landmarks required in the face detection

The facial landmark process is two-step where:

1. We localize the face in the frame we receive
2. Detect the key facial structures on the face ROI.

2.3.1 Localization of the face in the frame:

The [1] History of Oriented Gradients and Object Detection was a 6-step grueling process which could be summarized:

Step 1: A P number of positive samples (images with faces) were taken to detect and extract the HOG descriptors from these samples. This was required to get a truth value for the model later to be used in distinguishing.

Step 2: A N number of negative samples (images without faces in them) were taken to detect and extract the HOG descriptors from these samples. This was required to get the false value required for the model later to be used in distinguishing.

Step 3: A Linear Support Vector machine was trained on the positive and negative samples that were stored.

Step 4: A hard-negative mining was applied. For each image and possible scale of the image, we applied a sliding window function across our images. At each window the HOG descriptors were computer and the classifier was applied. When our classifier (incorrectly) classified a given window as an object we recorded the feature vector associated with the false-positive patch along with the probability of the classification. This is a major step in the process of hard-negative mining. This is illustrated in fig. 2.

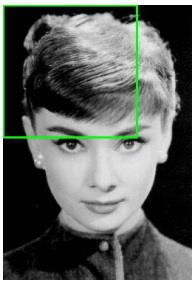


Figure 2

Step 5: The hard-negative samples were sorted according to their confidence levels and the classifier was again retrained.

Step 6: This trained classifier needed to be normalized because of the different scales of the image that we have trained the classifier on. This gave us multiple detection windows near the face. This is a fairly common problem and we were able to remedy this using the mean shift algorithm suggested by Triggs et al.[1][4]

2.3.2 Detection of Facial Structures

There were a multitude of facial landmarks that we could use for our Lip-reading system. We implemented the One Millisecond Face alignment with an ensemble of regression trees [6] with the help of the dlib library [9].

The manually labelled facial landmarks as illustrated in Figure 3, helped us to determine the required facial landmarks and their positions.

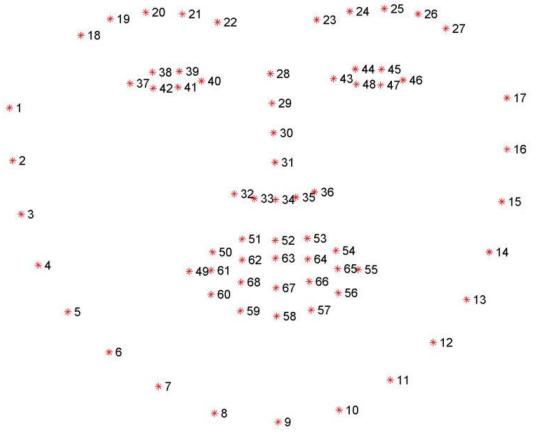


Figure 3

These annotations are part of the 68-point iBUG 300-W [6][7] dataset which the dlib facial landmark predictor was trained on. Though we could have used a 194-point model that was trained on the HELEN [5] dataset, it would have been moot to track all of them, as we were only interested in less than 50% of the points.

Using the 68-point iBUG 300W method, we only required points 49 till 68 for tracking the lips and points 1 to 17 for tracking the jaws.

An example of our implementation is illustrated in figure 4. Figure 4 shows the subject moving around, and the coordinates of the points being tracked.
The **one millisecond face alignment** [6] works on the basis that even if a single landmark is missing the entire detection window is removed. This proved to be an issue for lateral tracking of faces and facial landmarks.

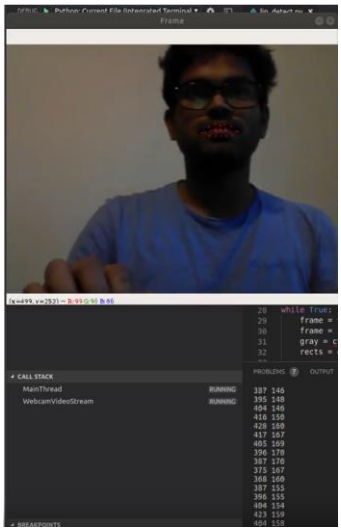


Figure 4

We soon realized that having a detection window around the detected lips would prove to be troublesome for normalizing the frames for the learning Neural model later.

Hence, we decided to take the midpoint of both the lips and the jaws and have hard margin window set to take in the extracted frame. It was as simple as selected the midpoint and having the frame set to a single sized margin from the point. Figure 5 illustrates the results of the extraction of the mouth patch in 3D space. The depth map with the values were stored per image as it's metadata.

Figure 5

Figure 5 is just one of the frames taken from the input video. Our program makes a thumbnail sized 5x5 frame matrix of such images to be fed into the descriptors computing part of the model. The thumbnail matrix with its depth maps is illustrated in figure 6.

The face matrix observed in Figure 6a shows us the sequence of the frames as the subject speaks a single word over the duration of 2 seconds. This frame is initially supposed to be of just 1 image of size 112 x 112.

K=25

Figure 6a

But the Neural model needs the temporal information per datapoint to be converted into datapoints per spatial.
But we faced one unique problem here. Each subject has its own speaking speed. We ended up with a lot of blank frames as illustrated in figure 6(b)

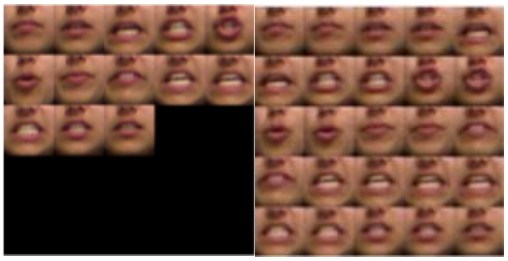


Figure 6b Figure 6c

To tackle this, we started stretching the sequence to fit over a length of 25, by normalizing the speaking speed. And we expect that this will help our validation accuracy, because all the speakers will have frames whose lips and jaws movements are moving at the same time sequence. The sequence is stretched by filling in the missing images with the nearest image, rather than filling it with 0's as shown in figure 6c. We used this equation [10] to stretch the images as shown in Figure 6c.

$$Stretch\_seq[i] = Orig\_seq\left[round\left(\frac{i * orig\_len}{25}\right)\right]$$

3. Computing the Descriptors

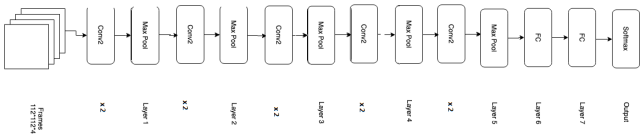
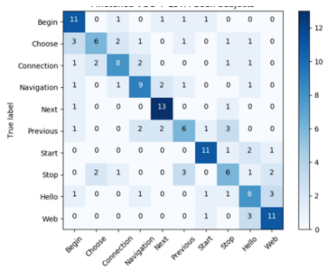


Figure 7

Figure 7 illustrators our neural network architecture that is used for computing the descriptors.

Our input during the creation of our dataset was limited to 1 second of video input, where the subject had to speak a pre decided word. This set of words illustrated in figure 8 are from the MIRACL- VC1 dataset [10]. They cover a variety of phonetics to be distinguishable from each other. Though the number of words is quite limited and cannot possibly predict all the words in the English language, it does give us a stepping stone in the right direction.

The entire premise of computing the descriptors lies on the soft max classifier at the end of our neural network. The soft max classifier here creates a confusion matrix of the words spoken [10] shown in figure 8 and determines the word from the input video.



Words	Phrases
Begin	Stop navigation.
Choose	Excuse me.
Connection	I am sorry.
Navigation	Thank you.
Next	Good bye.
Previous	I love this game.
Start	Nice to meet you.
Stop	You are welcome.
Hello	How are you?
Web	Have a good time.

Figure 8 Figure 9

3.1 Appearance Descriptor Computing

Our Neural Model is a deep network model comprising of 7 layers, which includes Convolution Layer, Max Pooling, Time Distributed Layer, LSTM Layer and SoftMax. The Convolution Layer helps to generate the edges, objects and features from the image set. And Max Pooling helps to down sample the image.

Using the Time Distributed Layer helps us to keep track of 25 frames, which we have used as a constant. The LSTM takes 25 inputs from the time distributed layer and passes it

to a SoftMax layer, which helps to generate the required output. As, you can see, the training dataset is of the array

[batch_size*tim_steps, height, width, channels]

We are using an ImageNet frame; thus, height and width are of size 112 respectively. The number of channels counts to four which are the RGB-D values. Timesteps is a constant in our case, which is equal to 25.

The RGB channels are normalized by dividing with the maximum intensity that is 255.0. The Depth channel is normalized by considering the lip points extracted from dlib to be 0 and the area around it subtracted by its original value (the lip depth values). This is achieved by subtracting, the complete Depth Map with the average depth to the lips derived.

The implementation formula used is given below:

$$lB = \mu(Dval(n)) ; n= [49\ \ 68]$$

$$DM = (DM - lB)$$

Where lB is equal the lip base. Dval is the depth values derived from the dlib library from points 49 till 68 and DM is the depth map.

Thus, using this, we can normalize the depths as depths are taken with respect to lips and will not change much generally.

Now every dataset comprises of 25 timesteps and (112, 112, 4) image size as input. That is a size of (25, 112, 112, 4). The Time Distributed Layer performs 2D Convolution on these with time_steps = 25, which helps us time distributed convolution outputs for the same.

Time Distributed Layer is applied along with all the layers of 2D-Convolutions, 2D Max Pooling Layers, and also with the last Flatten Layer. These outputs are parsed to the LSTM layer to generate a single output. Based on this output, SoftMax generates the desired output for the same.

The output is a sparse categorical output with the loss given by cross entropy.

From the image given, the convolution layers have an activation function of ReLU. The Layer 6 Dense Layer follows an activation of ReLU and the last Dense Layer follows an activation of SoftMax giving us a confusion matrix like figure 8.

4. Conclusion

In this project we tried various methods to make sure that the facial detection sub routine does not drop the frame rate down by a huge margin. Trained a History of Oriented Gradients and Object Detection and Linear SVM object detector to detect faces, instead of Viola Jones or KLT to minimize frame drops. After facial detection, we made use of a facial landmark detector that was trained on a deep learning neural net to detect the landmarks in our region of interest. This was later used to segment the face into its different sections. The lips were extracted into a 2D grid window, whose features were then classified by a CNN pre-trained on faces.

The lips extracted were then concatenated and processed to make sure we have 25 frames in the matrix before passing it on to the LSTM and finally the SoftMax layer to give the confusion matrix.

5. Future Work

Our project has a lot of work left. We are currently training our model on a set of just 10 words. These words, though varying in speech, do not constitute all 44 phenomes of the English Language. And it is not feasible to make all the subjects used in the training process to speak a large dictionary of words.

And we are facing issues if the texture or the skin color of the subject varies a lot. We are trying to normalize this by taking into consideration that no matter the skin color or texture, all the subjects have lips which usually tend to be pink and will have heavier weights if the frame is passed through a filter that gives weights to pixels if present in the red channel or spectrum.

And one of the biggest issues we faced was the different size of the lips. The varying size of the lips threw our neural network model way off its accuracy during training, because of the difference in the distance of the lips from the side of the frames. This gave a lot of false positives in the confusion matrix during the training phase.

These are few of the issues that we are currently facing in the project and that we intend to address in the near future. We are planning on creating our own dataset of subjects speaking all the 44 phenomes and having large amounts of metadata attached in each frame of the video such as the size of the lips, their 3D spatial information and their normalized color features while the camera takes in the data.

References

[1] Navneet Dalal and Bill Triggs in *Histograms of Oriented Gradients for Human Detection.* 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)

[2] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester and Deva Ramanan in *Object Detection with*

Discriminatively Trained Part Based Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, No. 9, September 2010

[3] Tomasz Malisiewicz, Abhinav Gupta, Alexei A. Efros in *Ensemble of Exemplar-SVMs for Object Detection and Beyond*. International Conference of Computer Vision 2011

[4] Mean Shift, https://en.wikipedia.org/wiki/Mean_shift

[5] Helen Dataset www.ifp.illinois.edu/~vuongle2/helen/

[6] Vahid Kazemi, Josephine Sullivan in *One millisecond face alignment with an ensemble of regression trees*. 2014 IEEE Conference on Computer Vision and Pattern Recognition

[7] Christos Sagonasa,, Epameinondas Antonakosa, Georgios Tzimiropoulosb, Stefanos Zafeirioua, Maja Pantica in 300 Faces In-The-Wild Challenge: database and result Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops

[8] Christos Sagonasa, Georgios Tzimiropoulos, Stefanos Zafeiriou1 and Maja Pantic in A semi-automatic methodology for facial landmark annotation. 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops

[9] Dlib dlib.net/

[10] Ahmed Rekik and Achraf Ben-Hamadou Walid Mahdi1 A New Visual Speech Recognition Approach For RGB-D Cameras.in *Image Analysis and Recognition*: 11th International Conference, ICIAR 2014, Vilamoura, Portugal, October 22-24, 2014, Proceedings, Part II

[11] A. Rekik, A. Ben-Hamadou, and W. Mahdi, "Human machine interaction via visual speech spotting," in Advanced Concepts for Intelligent Vision Systems. Springer, 2015, pp. 566–574.

[12] Shin,J.,Lee,J.,Kim,D.in*Real-time lips reading system for isolated Korean word recognition*. Pattern Recognition 44(3), 559–571 (2011)

[13] Pei, Y., Kim, T.K., Zha, H.in *Unsupervised random forest manifold alignment for lipreading*. In: ICCV. pp. 129–136 (2013)

[14] Futoshi Asano, Kiyoshi Yamamoto, Isao Hara, Jun Ogata, Takashi Yoshimura, Yoichi Motomura, Naoyuki Ichimura, Hideki Asoh in Detection and Separation of Speech Event Using Audio and Video Information Fusion. EURASIP journal on advances in signal processing 2004(11) · September 2004and Its Application to Robust Speech Interface .