Computer Vision Lip Reading

Grace Tilton, gtilton@stanford.edu

Abstract

This project compares and contrasts several leading research papers in the realm of Lip Reading and combined Audio Visual Recognition using either small convolutional neural networks or state-of-the-art deep learning models. This dialogue includes my own thoughts on potential reasoning of their results. It then goes onto describe and discuss my implementation of a basic lip reading model which pulls some unique ideas and features from each of the reviewed research efforts. Finally, this paper concludes with a list of further development recommendations given more time and resources.

Introduction

Lip reading is not a new science. In the def community lip reading has existed as long as language has existed. However, it takes many years to learn and can be difficult particularly for people who lose their hearing later in life. Lip reading also has deep roots in the intelligence community. The ability to capture long range video of a conversation where a 'bug' could not otherwise be planted is known as a thing of Bond movies. Other times full audio-video recordings exist where audio is corrupted or noisy so that speech is unintelligible.

Some interesting challenges face this implementation of Computer Vision (CV) and Natural Language Processing (NLP) including multiple orientations of a face to a camera, dropout where someone turns away from the camera, low vide resolution, physical speech impediments or adaptations (ventriloquism) which create non-normal physical patterns, and overlapping vowel patterns (try silently mouthing "elephant shoes" to your friend and see what they think, this is known as a homopheme).

This type of problem is a very good artificial intelligence/machine learning challenge because it stretches the capabilities of a machine to characterize what is a very instinctual and innate human ability. In addition, there is a good amount of data that can be used to train a machine to do this.

This project focuses on a foundational and de-scoped version of the CV lip reading problem where the data used will be simple face-forward video frames of people saying single English words. In this way, we eliminate a lot of the complexities described above. Our raw input data consists of JPEG images of theses video frames and the output of the model will be a classification. One interesting challenge I sought to address in this project was problems with training the immense size of the data and see how well a neural network could do with classification with instead a few coordinate points representing the general shape of the mouth.

Related Work

Quite a bit of work has been done on this subject and this paper/project has morphed just as nearly into a research review as it is an application project so do forgive me if this section is a bit long. (I guarantee it is probably the most interesting part of this document.

Lip Reading in the Wild (LRW)

The first paper I came across in this subject was Lip Reading in the Wild [1] (Chung & Zisserman, 2016). Chung and Zisserman made great strides in developing a massive dataset with over a million word instances and over a thousand unique speakers by automating data collection from public TV broadcasts. During collection and preprocessing, the video is cropped to only show the lower half of the speakers face (bridge of the nose to bottom of the chin and cheek bone to cheek bone). With is consistent "image box" that still includes other parts of the face, Chung and Zisserman were able to train four convolutional neural network (CNN) that were more robust to image jitter/shift between video frames. The four CNNs developed vary in architecture by digesting greyscale vs color images and which layer introduces the temporal element into the digestion. No CNN developed was more than five layers deep. Surprisingly, greyscale ingestion proved to be approximately 14% more accurate on average than the 3x larger RGB data. (I believe this is due to the greyscale image actually providing better definition for edge detection that is so commonly extracted in early convolutional layers) In addition, the “multiple towers” architecture (Figure 1) which first processed each frame as an individual image through a convolutional and pooling layer and then combined the results along the temporal axis in a later layer was most successful with an accuracy around 90% (for top-1 and top-10 accuracies). Somewhat expectedly, confusion matrices showed the highest confusion with plurals (“worker” vs. “workers”) and rhyming words (“troops” vs “groups”).

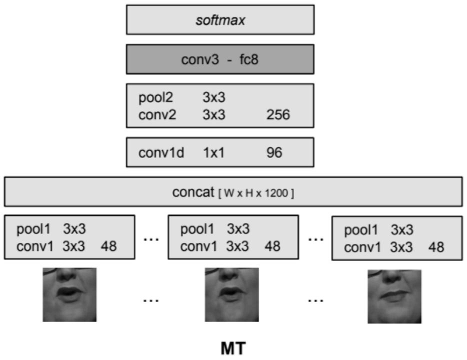

Figure 1: Chung and Zisserman's Multiple Towers Architecture

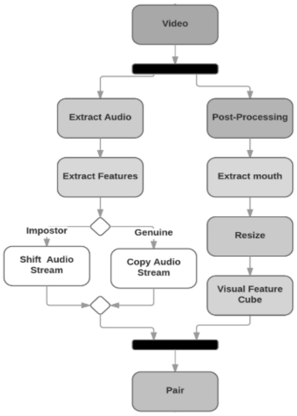

Figure 2: Dataset Processing Pipeline for Torfi et. All

3D Convolutional Neural Networks for Cross Audio-Visual Matching Recognition

The second paper I reviewed on this subject was 3D Convolutional Neural Networks for Cross Audio-Visual Matching Recognition [2] (Torfi et. All). I also leveraged parts of this project’s visual pipeline in compiling my own simplified application. This publicly available code can be found at https://github.com/astorfi/lip-reading-deeplearning. The novelty in Torfi et All’s approach was to create a common feature space of audio and visual modalities and using two parallel 3D CNNs to develop their model. Along one path, temporal sets of the actual audio waveforms were represented as an image while along the other, the video frames were passed through as a temporal set of images. The model then couples the results of later layers. This research also leveraged the LRW dataset making the results nicely comparable. One interesting thing to note is that with this methodology, researches found greater accuracy processing 3D (RGB) images as opposed to greyscale reaching approximately 94% accuracy in contrast to the findings above.

Deep Audio-Visual Speech Recognition

The most state-of-the-art paper I read in my research was Deep Audio-Visual Speech Recognition [3] (Afouras et. All). This paper compared two transformer-based models and addresses the problem as a character-based output which allows the models to recognize whole phrases and sentences. The first neural network considered in this paper was Connectionist Temporal Classification (CTC) which outputs a likelihood of each phenom (character sound) based on an input sequence. CTC does this for each frame and then looks across the frame for the phenom string that creates optimal alignment. The second model considered is a sequence to sequence (seq2seq) model which bases the output likelihood for each frame also on the output of the previous frame, eliminating the post-processing application of a language model. In general Afouras et. All found that the seq2seq model performed better (with lower word error rates) than CTC but that CTC generalized better to longer sequences. This approach is not directly comparable to the two papers above because it is applied over full phrases and sentences and success is thus measured by word error rate. However, industry appears to acknowledge this approach (lip reading as an open world problem) as far more superior and advanced in achieving the high level goals of this research area.

A New Visual Speech Recognition Approach For RGB-D Cameras

The fourth paper I wish to mention is the oldest, having been published in 2014. A New Visual Speech Recognition Approach For RGB-D Cameras [4] (Rekik et. All) is worth mentioning for two reasons: First, I utilized the same MIRACL-VC1 dataset described in this paper. And second, Rekik et. All added a fourth 2D layer of data to the 3-layer (color) 2D image in each frame by capturing a depth map with an MS Kinect sensor and zipping the depth layer to the other three.

This team's approach was to use SVMs to solve their classification problem and the results achieved between 90 and 95% accuracy for speaker dependent classification. However, this method showed poor 45-55% accuracy when attempted to generalize to speaker-independent classification. It is worth noting that the classification output space consisted of a total of only 20 words and phrases compared to an output space of over 500 words in the LRW dataset. This means that the high classification accuracy may be in part due to the dissimilarity between the chosen words and phrases and the small output space.

Data Set and Features

As noted in the previous section, I used the MIRACL-VC1 dataset [5] (Ben-Hamadou) for the application portion of my project. In reality, I only used a portion of the dataset by choosing to exclude a) the depth map information and b) all the phrase-based collected data. This reduced my output space to only 10 words: begin, choose, connection, navigation, next, previous, start, stop, hello, and web. Each data sample is a time series of 640 by 480 pixel images (video frames) of a word being spoken. Each sample includes one of 15 (ten female, five male) people speaking one of 10 words. There are 10 samples for each word spoken by each person totaling 1500 data samples. I choose to us an 80%/20% training and test split for this data.

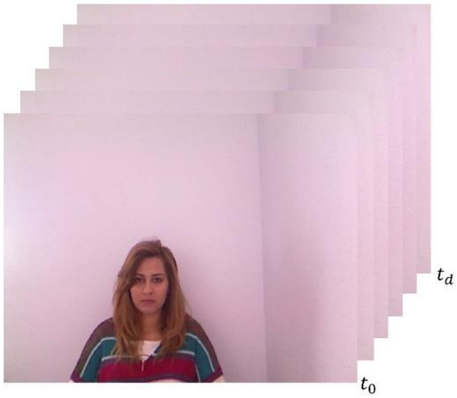

Figure 3: Example Data Sample from the MIRACL-VC1 Dataset

As opposed to the standard method found in all the papers above, I wanted to see if a CNN with a more minimal size for each data sample. IN order to do this, I leveraged work described in a blog titled Detect eyes, nose, lips, and jaw with dlib, OpenCV, and Python [6] (Rosebrock) to pre-process my data so each sample input was really a set of 21 (x,y) coordinates at each time step representing facial landmarks for the lips. Because the coordinates for landmarks appeared in slightly different locations for each person, I normalized the landmarks by setting the first point of the first frame to (0,0). In retrospect, I feel this may not have been necessary. I used the Dlib Python Library [7] to do this feature extraction. Dlib's pre-trained landmark predictor is based on Histogram of Oriented Gradients and Linear SVM Methods.

This method does introduce some error into my model because the image resolution of the video frames was not particularly high to begin with and this further reduces the amount of detailed data passed into the model. My objective here was to eliminate or reduce the discrepancy found in [4] (Rekik et. All) between speaker-dependent and speaker-independent classification

Methods

The following algorithms/methods were used in either the preprocessing or model for my application

Linear SVM – A supervised learning model that leverages relationships between features to reduce the input data size and characterize interdependencies/similarities.

Histogram of Oriented Gradients – A feature descriptor technique which counts occurrences of gradient orientation in areas of an image for the purpose of object detection.

Convolution – A cross-correlation layer in a neural network used to describe a relationship between adjacent points.

Pooling – A feature reduction layer in a neural network which taks an area of features and outputs a characteristic of that area, most commonly the mean or maximum)

Softmax – A normalizing function of weights representing likelihoods of possible classifications.

Experiments, Results, Discussion

I trained my model with 30 epochs, a batch size of 100, and decaying learning rate starting at 0.01. With this methodology I was able achieve a speaker-dependent accuracy of 76.4% on the training data and 73.8% on the test data. I believe this is because a lot of fidelity is lost in the conversion from image to facial landmark coordinates. This did however close the gap between speaker-dependent and speaker-independent accuracy, bringing the speaker-independent accuracy up to about 60%.

In general, I saw a lot of confusion/error between words of similar syllable length. I believe this is because I padded extra video frames of shorted recordings by repeating the last frame.

Conclusion, Future Work

In general, my initial solution was very limited. However, I do think it shows promise. I definitely did not get to spend as much time on this project as I would have liked and there are a myriad of other things I'd like to do in the future to continue.

First of all, I'd like to consider different data such as the LRW set where I can try to add the parallel sound-wave based model and merge results so that the model with higher confidence on any one word would be used.

I also do think there is a lot to be said for the phenom-based approach discussed in [4]. I think it would lead to higher accuracy and generalize well to new languages. I think the biggest limitation with such an effort is the compute for processing a DNN. Another concept I liked in [4] and think is promising is the post-processing used in the CTC model used for optimal alignment.

I think the most difficult development element was actually the data preprocessing which took some time to both set up and get my environment working correctly to implement.

Code Link

https://drive.google.com/open?id=1orBE04NJ3AH1oxBp6hbFuFDpaWs2Bljt

Contributions

I chose to do my final project as an individual. However, I leveraged guidance and implementation strategies from several individuals/resources in the reference below.

References

[1] Chung J.S., Zisserman A. (2017) Lip Reading in the Wild. In: Lai SH., Lepetit V., Nishino K., Sato Y. (eds) Computer Vision – ACCV 2016. ACCV 2016. Lecture Notes in Computer Science, vol 10112. Springer, Cham

[2] Torfi, A., Iranmanesh, S. M., Nasrabadi, N., & Dawson, J. (2017). 3d convolutional neural networks for cross audio-visual matching recognition. IEEE Access, 5, 22081-22091.

[3] Afouras, T., Chung, J. S., Senior, A., Vinyals, O., & Zisserman, A. (2018). Deep audio-visual speech recognition. IEEE transactions on pattern analysis and machine intelligence.

[4] Rekik, A., Ben-Hamadou, A., & Mahdi, W. (2014, October). A new visual speech recognition approach for RGB-D cameras. In International Conference Image Analysis and Recognition (pp. 21-28). Springer, Cham.

[5] Ben-Hamadou, A. (2019). MIRACL-VC1 - Achraf Ben-Hamadou. [online] Sites.google.com. Available at: https://sites.google.com/site/achrafbenhamadou/-datasets/miracl-vc1 [Accessed 5 Dec. 2019].

[6] Rosebrock, A. (2019). Facial landmarks with dlib, OpenCV, and Python - PyImageSearch. [online] PyImageSearch. Available at: https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/ [Accessed 5 Dec. 2019]

[7] PyPI. (2019). *dlib*. [online] Available at: https://pypi.org/project/dlib/ [Accessed 1 Dec. 2019].