Visual Speech Recognition with KNN and Hybrid KMeans-SVM EECS 442 Porject Final Report

Tian Zhang*
Computer Engineering, Senior year

Tongshuang Wu†
Computer Engineering, Junior year

Yi-Sheng Hsieh‡
EE:s
grad school 2nd year

Abstract

Visual Speech Recognition is a novel application in computer vision, combining the usage of image processing, artificial intelligence, object detection, pattern recognition, statistical modelling, etc. Using both visual information as an additional source for audio extraction, it has great impacts in occasions where the audio information cannot be effectively conveyed. We therefore propose to build a visual speech recognition (VSR) system, which uses visual information (e.g. lip motion) as the stand alone source to recognize/classify words being spoken. Working with an exisiting well-constructed dataset, we attempt to extract lip description features, and train it with hybrid KMeans-SVM to achieve a reasonable visual speech recognition on several words. This report summerizes our temprerary achievements and remaining milestones.

1 Introduction

Visual Speech Recognition, as introduced detailedly in [4] is a novel application in computer vision, combining the usage of image processing, artificial intelligence, object detection, pattern recognition, statistical modelling, etc. Using both visual information as an additional source for audio extraction, it has great impacts in occasions where the audio information cannot be effectively conveyed.

Several studies have touched this topic. However, most of the crrent audio-visual speech recognition technology focuses on using visual clues as aids to audio-visual automatic speech recognition (AV-ASR) system, which still rely heavily on audio, and the advantages of visual detection hasnt been fully taken. We therefore propose to build a visual speech recognition (VSR) system, which uses visual information (e.g. lip motion) as the stand alone source to recognize/classify words being spoken, and audio information as an optional auxiliary method for accuracy improvement. We will start from simple digits recognition. When its accuracy is confirmed, we will continue to try some temporarily discussed advanced features. We believe it can potentially be further developed for more social-related usage, such as aids for deaf and hard-hearing users in terms of lip-reading.

Professional works have been done in this field. Though in the consideration of performance, most works involve both facial feature classificaion and audio recognition. As our objective is to maximize the impact of facial features, our work will mainly focus on facial feature usage, while audio resources may be considered as a suppliment when we have polished the former process.

While feature definition may vary a lot, the most popular classification Machine Learning algorithm used in VSR are K-Nearest-Neighbors(KNN) and Hidden Markov Model (HMM). In our case, we used KNN to build the baseline, and implement KMeans to first build the viseme with the notion of state borrowed from HMM, or feature states, and then construct SVM for classification.

To the best of our knowledge, our work has the following contributions:

- We tried to extract features as detailed as possible to capture the lip motion precisely, which, as proved by the experiments, would play an important role in raising the performance from purely using lip width and length.
- We are the first to try to combine the concepts of Kmeans, HMM and SVM in data processing and classification, which helps to open up the probability of constructing customized viseme and using them in different kinds of classification methods.

The remaining report are constructed as the following: Section 2 goes through similar works that has been done in the related fields, Section 3 introduces our proposed method to solve the problem, Section 4 summerizes our experimental results, and Section **??** concludes the report with a discussion of potential future works.

2 Related Work

Visual Speech Recognition has become popular recently due to its wide application in human-computer interaction (HCI), audio-video speech recognition (AVSR) and sign language recognition. Although growing number of applications have emerged, most of the current work focus on building a visual recognition system to improve the performance of an audio-based speech recognition system. A complete overview of AVSR system can be found in this work [19].

Visual feature is the key of visual recognition system. There are three type of methods used to extract the visual information of the lip. First is the geometric-based feature, which captures the shape or contour [12] of lip as the visual representation [14]. However, it is often challenging to detect the exact location of lip, especially under illumination conditions. Another common technique is extracting the appearance-based feature, which captures parametric Eigen-space of the object of interest. Examples of appearance-based methods are principle component analysis [2], Linear discriminate analysis or two-dimensional Discrete Wavelet Transform [17]. Other work [3] combined the above two methods to attain better representation.

Several research investigating classification models are also appearing recently. One research [7] uses Support Vector Machine to generate a posterior probability estimate to classify spoken words. HMM [13] is another popular model used in visual speech recognition due to its similarity to the audio counterpart. The variation of HMM, which are the Coupled HMM [1] and Factorial HMM [6] are also used to take advantage of complementary sources of speech information. K Nearest Neighbor [8] is used in many of the research as baseline.

3 Technical Part

3.1 Overview

We propose to process the visual language detection mainly with the lip motion recognition, which is a branch of partial face detection. Some related concepts and works are listed in sequential

*tzha@umich.edu
†twuac@umich.edu
‡yshsieh@umich.edu

orders below. We also consider audio frequency spectrum classification as an auxiliary method for performance improvements.

The high level overview can be seen in Figure 1. In particular, Region of Interest (ROI) detection, feature extraction, viseme construction and classification algorithms are discussed in detail here.

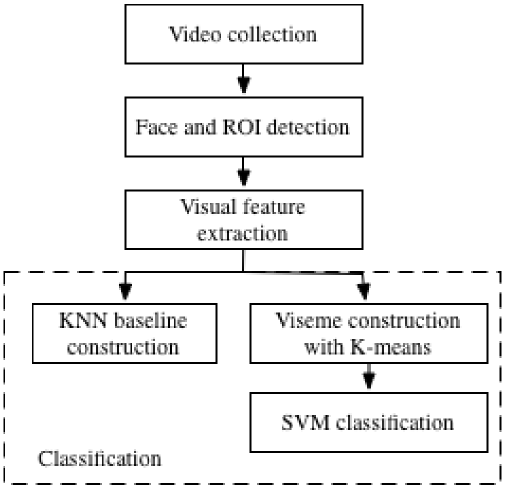
Video collection
Face and ROI detection
Visual feature extraction
KNN baseline construction
Viseme construction with K-means
SVM classification
Classification

Figure 1: Overview for VSR system workflow

3.2 ROI detection

Our ROI is the lip, which is one of the most significant sub-face features, and is part of face detection. It is can be regarded as a specific case of object-class detection, focusing on the detection of frontal human faces. It is analogous to image detection in which the image of a person is matched bit by bit. For accuracy and simplicity, we used the Viola-Jones algorithm [10], an efficient algorithm for detecting people's faces, noses, eyes, mouth, or upper body, and focus specifically on lips. In order to detect the mouth, the model is composed of weak classifiers, based on a decision stump, which use Haar features to encode mouth details.

3.3 Feature Extraction

We extract feature based on method used in [9]. We extract 8 features in total to capture characteristics of the the mouth area. These features are:

- The width of the mouth: we use the width of the ROI image directly to represnet the width of the outer contour of mouth.

- The height of the mouth: we use the height of the ROI image directly to represent the height of the outer contour of mouth.

- The Mutual Information in the discrete wavelet transform (DWT) domain between the current ROI and the previous ROI: It is calculated as

$$M(X;Y) = \sum_{x}\sum_{y} p(x,y) log(\frac{p(x,y)}{p(x)p()}y) \qquad (1)$$

  where X, Y represent the current and the previous ROI image frame and $p(x)$ and $p(y)$ represent the marginal probability mass function of X and Y respectively. Due to noise and lighting condition, we first transform the image from spatial domain to frequency domain using technique called Discrete Wavelet Transform (DWT). DWT captures both frequency and location information and in this application we use the Haar wavelet. We calculate Mutual Information of the four subbands after DWT individually and take average value of them as the final value. Mutual Information measure how two ROI images resmeble to each other. In other word, it captures the temporal correlation between two image frame with a single scalar value.

- The Image Quality Value in the discrete wavelet transform (DWT) domain between the current ROI and the previous ROI: The Image Quality Value measures how two consecutive images differs from each other. It capture the change of mouth spatial structure with a single scalar. The quality value range between -1 and 1 and the maximum difference between two frames occurs when the quality value equals to 0 and the minimum difference occur when the quality value is 1 or -1. We take similar approach to that of Mutual Information. That is, we take DWT first, calcualte Quality Value of each subbands seperately, and average the value as the final value.

- The ratio of the vertical to the horizontal edges in the discrete wavelet transform (DWT) domain: For each non-LL-subband, the larger difference between the coefficient and the mean, the more likely it is a feature/edges. As a result, we first do DWT and count the number of coefficient in non-LL-subband that is one standard deviation greater than the mean and one standard devation less than the mean. In this application, we take HL as the vertical edges and LH as the horizontal ones.

- The ratio of the vertical to the horizontal features in the discrete wavelet transform (DWT) domain: We use Sobel edge detector to calculate the edge ratio (ER) value. The formula:

$$ER = \frac{\sum_{x=1}^{W}\sum_{y=1}^{H}\sum_{i=1}^{1}\sum_{j=1}^{1}|ROI(x+i,y+j)(S_v(i+1,j+1))|}{\sum_{x=1}^{W}\sum_{y=1}^{H}\sum_{i=1}^{1}\sum_{j=1}^{1}|ROI(x+i,y+j)(S_h(i+1,j+1))|} \qquad (2)$$

  where W is the width of the ROI image and H is the hieght of the ROI image. Sv indicates the Sobel vertical filter and Sh indicates the Sobel horizontal filter. When the mouth stretch horizontally, ER decreases and vice versa.

- The amount of red color which represent the appearance of tongue in ROI: It is based on the assumption that the color of the lips is red. The formula for calculating it is:

$$RC = \frac{\sum_{x=1}^{W}\sum_{y=1}^{H} Red(ROI(x,y))}{WH} \qquad (3)$$

  where W is the width of the ROI image and H is the hieght of the ROI image.

- The amount of of visible teeth in the ROI: First we convert the pixels values of ROI to 1976 CIELAB colour space $(L*,a*,b*)$ and 1976 CIELUV colour space $(L*,u*,v*)$ and follow this fomula:

$$t = \begin{cases} 1 & a^* \leq (\mu_a - \sigma_a) \\ 1 & u^* \leq (\mu_u - \sigma_u) \\ 0 & otherwise \end{cases} \qquad (4)$$

where W is the width of the ROI image and H is the hieght of the ROI image.

After extracting features of each frame, we get a sequence of features for each word utterance, resulting in a $8 \times T$ matrix where T is the number of frame for each word utterance.

3.4 Classification

3.4.1 KNN Baseline construction

In this multiclass classification problem, we apply k-nearest neighbor (KNN) as our baseline, taking the advantages that it is simple but powerful implementation. For a data point (one $8 \times T$ matrix in this case), KNN find the nearest k closest neighbor of that data point based on a self-defined distance matrics in high dimensional space. The data point is thus classified to a class according to the majority vote of the k nearest neigbor. This approach is simple to implement but has high computational complexity in the testing phase.

To apply KNN, we need to first com up with a self-defined distance matrics. Due to the various length of each data point (i.e. T differs for evey data point), conventional similarity measurements such as cosine similarity or RBF kernel cannot apply. We therefore use Dynamic Time Warping (DTW) the solve this problem, which is an algorithm to measure similarity when two time series have variable length. Using this similarity as the distance, we can then compare a new test word with all other words and determine which one it is most likely to be. We will use cross-validation to determine an appropriate k

3.4.2 SVM algorithm implementation

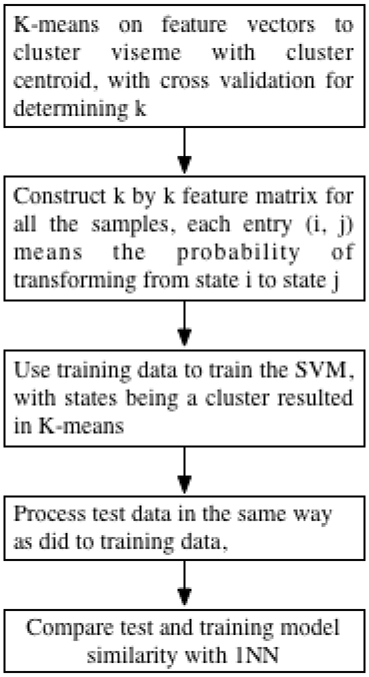
K-means on feature vectors to cluster viseme with cluster centroid, with cross validation for determining k
Construct k by k feature matrix for all the samples, each entry (i, j) means the probability of transforming from state i to state j
Use training data to train the SVM, with states being a cluster resulted in K-means
Process test data in the same way as did to training data,
Compare test and training model similarity with 1NN

Figure 2: Overview for Kmeans-SVM workflow

To use the feature matrices in classification, some pre-processing is needed, such that we could get rid of the size mismatch resulted from the length differences of the word being spoken, some unintentional lip shape outliers, while still preserve the overall trend. To do this, we would like to borrow the notion of state from Hidden Markov model. A Hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. A commonly used option in this case is to define viseme as states. A viseme is any of several speech sounds which looks the same [5]. In our review, neither did any paper discussed the approach to construct the viseme, nor ded we find any accessible implemented database. Further, since the extraction of features will surely affect our definition of the similarity degree, we decided to construct our own viseme in a intuiitive way: take the extracted feature matrices for this word, break them into per-frame-feature vectors, and use K-means algorithm, an unsupervised custer algorithm, to group them based on similarity. The resulted k clusters will represent the k viseme appeared, each one with a cluster centroid representing the core features of the cluster. In this way, our k states are also confirmed.

,

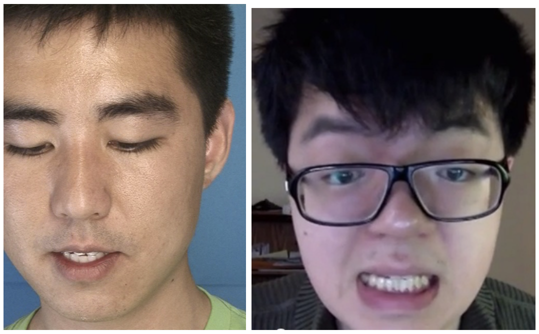

Figure 3: Screenshot of the original video of one speaker, both for self-made video (left) and videos in the database (right)

We are then able to build a state transformation probablity matrix for each training and testing data. Since the feature matrix is constructed using per frame features in a time-series base percedure, by comparing frame feature with viseme representing feature vectors and decide its current stage using 1NN, a sepecial case of KNN where the test data is just regarded as being similar to its nearest neighbor, and then compute probability between each two state to represent how much likely it is for it to translate for one to the other one. The resulting a $k \times k$ matrix will be our model representation, which will be the features for SVM classification.

The new test word will be processed in the way, and by comparing its probability matrices with existing models, we will be able to classify it properly.

The workflow can be seen in Figure 6.

4 Experiment Result

Following the workflow introduced in section 3, we introduce the detailes of each step.

4.1 Obtain Database

In this study, we are focusing on two-word-classification. For internal test, we first shooted some self-made video on our own. Two of us spoke two words, namely *dark* and *beach*, each of which for ten times. However, due to the difficulty to maintain a controlled and constant environment, we decided to search for a well-constructed database available online instead of collecting data by ourselves. We decided to use CMU-AMP Database [11]. This database is collected by CMU for a similar VSR research. It contains 3 speakers and 150 isolated words, each repeated 10 times. High quality recording in a sound-proof studio with a blue-screen background. We have contacted the original principal for this research work, and got permission to use their original video. A video screen shot can be found in Figure 3.

For consistency, we extracted the word *dark* and *beach*. This makes our database of limited size, but still reasonable to do a neat job. We devided our data into training and testing set for evaluation purpose.

We would also recommend M2VTS [15] in the cases the fund is sufficient. It is a useful tool for many authoritative experiments concerning tasks of facial feature and speech detection, detection. It consists big number of speakers, vocabularies, and well constructed

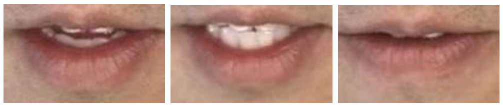

Figure 4: ROI examples extracted from the original video

features. However, it is not freely accessible, for which we did not use it.

4.2 ROI detection

We used the pre-implemented Viola-Jones algorithm in Matlab for accuarcy and simplicity[1].In detail, the cascade object detector in computer vision toolbox will detect face, mouth automatically with one function. Some sample ROI are shown in Figure 4.

The cascade object detector contains built-in classifier. The cascade classifier consists of stages, where each stage is an ensemble of weak learners. The weak learners are simple classifiers called *decision stumps*. Each stage is trained using a technique called boosting. *Boosting* provides the ability to train a highly accurate classifier by taking a weighted average of the decisions made by the weak learners. Each stage of the classifier labels the region defined by the current location of the sliding window as either positive or negative. Positive indicates an object was found and negative indicates no object. If the label is negative, the classification of this region is complete, and the detector slides the window to the next location. If the label is positive, the classifier passes the region to the next stage. The detector reports an object found at the current window location when the final stage classifies the region as positive.

To achieve fast detection, the stages are designed to reject negative samples as fast as possible, based on the assumption is that the vast majority of windows do not contain the object of interest. Conversely, true positives are rare, and worth taking the time to verify. A *true positive* occurs when a positive sample is correctly classified. A *false positive* occurs when a negative sample is mistakenly classified as positive. A *false negative* occurs when a positive sample is mistakenly classified as negative. To work well, each stage in the cascade must have a low false negative rate. If a stage incorrectly labels an object as negative, the classification stops, and there is no way to correct the mistake. However, each stage may have a high false positive rate. Even if it incorrectly labels a non-object as positive, the mistake can be corrected by subsequent stages.

There are some potential improvements to be considered. The *vision.CascadeObjectDetector* System object comes with several pre-trained classifiers for detecting frontal faces, profile faces, noses, upper body, and eyes. However, as we mentioned previously, these classifiers may not be sufficient for a particular application. In Matlab, Computer Vision System Toolbox provides the trainCascadeObjectDetector function to train a custom classifier.

Cascade classifier training requires a set of positive samples and a set of negative images. We may provide a set of positive images with regions of interest specified to be used as positive samples. We may also use the trainingImageLabeler app to label objects of interest with bounding boxes. The app outputs an array of structs to use for positive samples. We also should provide a set of negative images from which the function generates negative samples automatically. Set the number of stages, feature type, and other function parameters to achieve acceptable detector accuracy.

However, from observation, we found that there are cases where the upper lip will be missed in the detection, as shown in Figure 5. We surmise it is because the Matlab built-in model is designed for lips shapes in silence mode, which would fail for exaggerated motion.

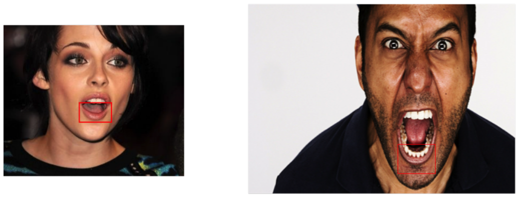

Figure 5: ROI examples extracted from the original video

Table 1: Classification result validation.

	CMU features	Features used in this study
KNN	N.A.	75%
Kmeans-SVM	80%	100%

4.3 Feature extraction

Implemented feature extraction algorithm introduced in 3.3, we can get the matrix. For example, for the left most ROI in Figure 4, the feature vector is:

$$\begin{bmatrix} 85.0 & 141.0 & 3.041 & 0.999 & 1.319 & 1.727 & 0.883 & 0.256 \end{bmatrix}$$

4.4 Classification

We ran both KNN and SVM on the extracted feature matrices. For validation purpose, we compared the results of our feature and classification methods with the ones for CMU project [11], in which only the lip width and length are taken into consideration. The result can be seen in Table 1. Here, the k used in K-means is empirically set to 5. However, we have tried $k = 3$ to $k = 8$, and the result are not affected greatly. One explanation could be, since as k grows larger, k-means tends to divide groups in to sub-groups, the state transition does not vary greatly.

Comparing KNN and Kmeans-SVM using our feature, Kmeans-SVM out performs KNN, which is reasonable given the distribution of the data as shown in Figure **??**, since KNN could make mistakes on those data near the boundary. Comparing CMU feature and our feature, ours outperform theirs, meaning that the 8-feature vector captures the lip shape more percisely.

From the table, we could safely conclude that doing classification using Kmeans-SVM with detailed feature could return a satisfying result.

5 Conclusion

In this project, we successfully implemented a VSR system, useing Viola-Jones algorithm for lip detection, and classified using Kmeans-SVM classification with well-defined features. Tested by processing two-word classification, the system worked considerably well, reaching a perfect separation, and significantly outperformed some related works. We proved that more detailed feature design could improve the results, and the mixture of different classification could work well.

There are several potential improvements. First, as discussed in Section 4.2, we could train customized model to better detect ROI. Also, we could try to process multi-word classification to see its performance.

[1] algorithm detail: www.mathworks.com/help/vision/ref/vision.cascadeobjectdetector-class.html

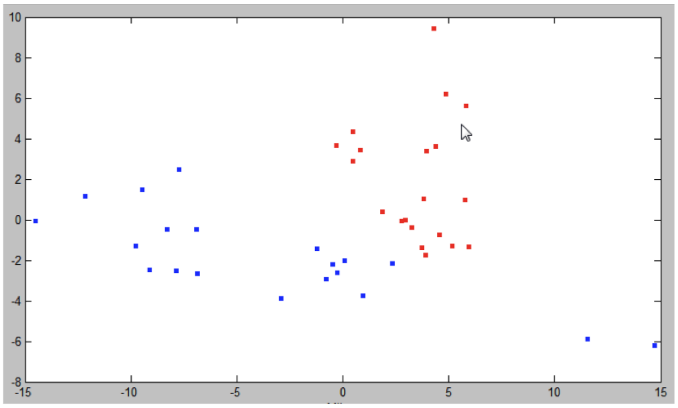

Figure 6: The distribution of DTW results for KNN, with the colors encoding the words being spoken

References

[1] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 994–999. IEEE, 1997.

[2] C. Bregler and S. M. Omohundro. Nonlinear manifold learning for visual speech recognition. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 494–499. IEEE, 1995.

[3] S. Dupont and J. Luettin. Audio-visual speech modeling for continuous speech recognition. *Multimedia, IEEE Transactions on*, 2(3):141–151, 2000.

[4] G. Erten. Audio visual speech processing, 2001.

[5] C. G. Fisher. Confusions among visually perceived consonants. *Journal of Speech, Language, and Hearing Research*, 11(4):796–804, 1968.

[6] Z. Ghahramani and M. I. Jordan. Factorial hidden markov models. *Machine learning*, 29(2-3):245–273, 1997.

[7] M. Gordan, C. Kotropoulos, and I. Pitas. A support vector machine-based dynamic network for visual speech recognition applications. *EURASIP Journal on Applied Signal Processing*, 2002(1):1248–1259, 2002.

[8] A. B. Hassanat and S. Jassim. Visual words for lip-reading. In *SPIE Defense, Security, and Sensing*, pages 77080B–77080B. International Society for Optics and Photonics, 2010.

[9] A. B. A. Hassanat. Visual speech recognition. *CoRR*, abs/1409.1411, 2014.

[10] O. H. Jensen. *Implementing the Viola-Jones face detection algorithm*. PhD thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2008.

[11] K. Kumar, T. Chen, and R. M. Stern. Profile view lip reading. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–429. IEEE, 2007.

[12] S. Lucey, S. Sridharan, and V. Chandran. An improvement of automatic speech reading using an intensity to contour stochastic transformation. *Barlow [Bar00]*, pages 98–103.

[13] S. Lucey, S. Sridharan, and V. Chandran. An investigation of hmm classifier combination strategies for improved audio-visual speech recognition. In *INTERSPEECH*, pages 1185–1188, 2001.

[14] J. Luettin, N. A. Thacker, and S. W. Beet. Speechreading using shape and intensity information. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 1, pages 58–61. IEEE, 1996.

[15] K. Messer, J. Matas, J. Kittler, and K. Jonsson. Xm2vtsdb: The extended m2vts database. In *In Second International Conference on Audio and Video-based Biometric Person Authentication*, pages 72–77, 1999.

[16] A. V. Nefian, L. Liang, X. Pi, L. Xiaoxiang, C. Mao, and K. Murphy. A coupled hmm for audio-visual speech recognition. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 2, pages II–2013. IEEE, 2002.

[17] C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, J. Sison, A. Mashari, and J. Zhou. Audio-visual speech recognition. In *Final Workshop 2000 Report*, volume 764, 2000.

[18] T.-L. Pao, W.-Y. Liao, and Y.-T. Chen. Audio-visual speech recognition with weighted knn-based classification in mandarin database. In *Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference on*, volume 1, pages 39–42. IEEE, 2007.

[19] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. W. Senior. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326, 2003.

[20] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.