Detection of Specific Mispronunciations using Audiovisual Features

Sébastien Picard[1,2], G. Ananthakrishnan[2], Preben Wik[2], Olov Engwall[2], Sherif Abdou[3]

[1]Electronics, Telecommunications and Computer Sciences, University of Lyon, France
[2]Centre for Speech Technology, KTH, (Royal Institute of Technology), Stockholm, Sweden
[3]Faculty of Computers & Information, Cairo University, Egypt

sebastien.picard@cpe.fr, {agopal, preben, engwall}@kth.se, s.abdou@fci-cu.edu.eg

Abstract

This paper introduces a general approach for binary classification of audiovisual data. The intended application is mispronunciation detection for specific phonemic errors, using very sparse training data. The system uses a Support Vector Machine (SVM) classifier with features obtained from a Time Varying Discrete Cosine Transform (TV-DCT) on the audio log-spectrum as well as on the image sequences. The concatenated feature vectors from both the modalities were reduced to a very small subset using a combination of feature selection methods. We achieved 95-100% correct classification for each pair-wise classifier on a database of Swedish vowels with an average of 58 instances per vowel for training. The performance was largely unaffected when tested on data from a speaker who was not included in the training.

Index Terms: Time Varying-DCT, Genetic Algorithms, MRMR, CAPT

1. Introduction

The problem of classifying audiovisual data, often known as automatic audiovisual speech recognition, has been of interest to researchers because of the ability of video-based features to enhance speech recognition in the presence of noise, making it more robust [1, 2]. Among the different visual features that could be used, some common features are high level statistical models representing the lips, such as active shape modeling [3], snake [4] and active appearance modeling [5]. Some low level features such as Region of Interest (ROI) transformations have also been used [6]. While high level features have in general shown a high accuracy, they also have a high dependence on either initialization or the initial annotation of the active shapes besides requiring large amounts of training data. For the audio features, the most common representation has been Mel Frequency Cepstral Coefficients (MFCCs). The audiovisual data being temporal in nature, Hidden Markov Models (HMM) [1] are the most popular classification algorithms, which are known for their capability to classify time-varying data.

Integrating these audio and visual features has been another problem of interest [7]. Early fusion or feature fusion concatenates the audio and video feature vectors and then applies machine learning to classify the phonemes. Late or decision fusion, on the other hand, employs two different classifiers for audio and visual features and then applies a variety of schemes to integrate the decisions. While early fusion schemes can employ simple classification techniques, the most important problems that such schemes face is the difference in frame rates. While audio features are usually extracted at 100 frames per second, visual features are available only at around 15 to 25 frames per second. There is also a large difference in the number of features required to parameterize each audio and visual frame. Additionally, estimating the relative importance between the two modalities is not trivial. Late fusion methods take care of these problems by allowing two channels of decisions to take place and applying probabilistic rules to determine the relative weights for the two modalities, e.g., Multi-stream HMMs [1]. The disadvantage of such methods is the complexity of the recognition system and the exponential increase in the number of parameters that need to be estimated, which is a drawback for sparse data.

This paper tries to apply the paradigm of automatic audiovisual phoneme classification to the problem of mispronunciation detection in a Computer Assisted Pronunciation Training (CAPT) system. Mispronunciation detection is a slightly different problem from phoneme/speech recognition, although they are related. A mispronunciation of a phoneme needs be signaled even if it is not classified as a different phoneme in the language. In case of second language learners, common mispronunciations depend on both the target language (L2) and the learner's native language (L1), i.e., the same type of errors are not made by speakers of different L1s. Thus, the training material for the mispronunciation detection needs to be L1-L2 specific. This often makes the available training data for the detection algorithm sparse. However, linguistic theory and second language teaching experience can provide insights to potential problems for speakers of a particular L1. For example, native speakers of Spanish find it difficult to distinguish between the Swedish vowels 'o' /u:/ and 'å' /o:/, since this distinction is lacking in Spanish. Using such expert knowledge it is possible to break down the mispronunciation detection problem into a series of binary classifiers to test the relevant mispronunciation hypotheses for the phoneme at hand. In this manner, the required training data is reduced to a set including the correct pronunciation of the phoneme and the common mispronunciation types that the binary classifier should test against. Furthermore, the binary classification allows identification of the type of error that the learner has made and a CAPT system can use this to provide more relevant feedback to help L2 learners correct their mispronunciations.

In order to solve this problem, we propose a framework which is capable of handling the typical CAPT environment, in which students often have access to simple equipment like a desktop microphone and a web-camera. We use Time Varying Discrete Time Coefficients (TV-DCT), described in Section 2, to parameterize both the audio and the visual features. The visual features are low-level features extracted from the ROI and do not require any manual annotation, initialization or training. The problem of audiovisual integration using feature selection algorithms is described in Section 3. The audiovisual classification is performed with Support Vector Machines (SVM) [8], since they are the preferred algorithm (compared to probabilistic models like HMM) when the data is sparse or noisy. Sections 4, 5 and 6 describe the data, experiments and the conclusions of the paper, respectively.

2. Feature Extraction

MFCCs, which approximate time-varying data by assuming it to be short-time stationary, represent the frequency spectrum and its perception by humans rather accurately, but require a suitable time-varying machine learning algorithm in order to model the temporal variability. Most suggested visual features have the same problem because the features are based on individual images. In addition, the frame rates of the two modalities are often very different, which poses a problem when they should be combined for audiovisual speech recognition tasks.

In order to solve these two problems, we propose a TV-DCT which performs a 2 dimensional (2D) DCT on the log-spectrum of the audio and a 3D-DCT on the sequence of images. These features incorporate dynamic information in both audio and visual (video) features, which avoids the need of a classifier handling temporal variability.

If $A(f,t):\{1 \le f \le F, 1 \le t \le T\}$ is the time-varying log spectrum of the audio signal, with F frequency sub-bands and T time-samples, then the audio features $\tilde{A}(p,q)$ are the 2D-DCT performed along the dimensions f (frequency) and t (time) [9] as illustrated in Figure 1. The dimensions $p:\{1 \le p \le P\}$ and $q:\{1 \le q \le Q\}$ are called 'quefrency' and 'meti'. 'Quefrency' are the spectral features with a time dimension and 'meti' parameterize the time dynamics of the audio signal and have a frequency dimension.

Similarly, the visual features $\tilde{V}(i,j,k)$ can be extracted from a video sequence of 2D images $V(x,y,\tau):\{1 \le x \le X, 1 \le y \le Y, 1 \le \tau \le \Gamma\}$ using 3D-DCT as described in equation 1.

$$\tilde{V}(i,j,k) = a(k)\sum_{x=1}^{X}\sum_{y=1}^{Y}\sum_{\tau=1}^{\Gamma}\left[V(x,y,\tau)\cos(t_{ix})\cos(t_{jy})\cos(t_{k\tau})\right] \quad (1)$$

where

$$t_{ix} = \frac{\pi(2x-1)(i-1)}{2X},\ t_{jy} = \frac{\pi(2y-1)(j-1)}{2Y},$$
$$t_{k\tau} = \frac{\pi(2\tau-1)(k-1)}{2\Gamma} \text{ and } a_3(k) = \begin{cases} 1 & k=1 \\ \frac{1}{\Gamma} & k>1 \end{cases} \quad (2)$$

The dimensions $i:\{1 \le i \le I\}$ and $j:\{1 \le j \le J\}$ are the image spectral features and $k:\{1 \le k \le K\}$ are the 'meti' features which parameterize the dynamics of the video segment.

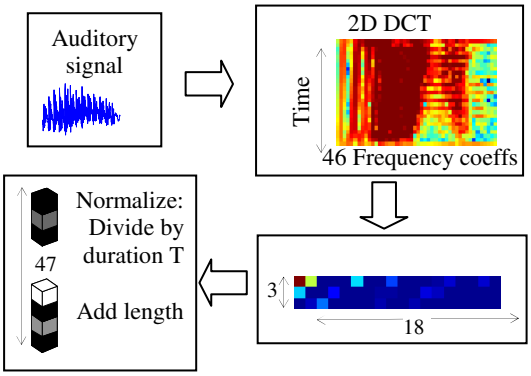


Figure 1: *Illustration of the extraction of 2D TV-DCT features from the audio signal.*

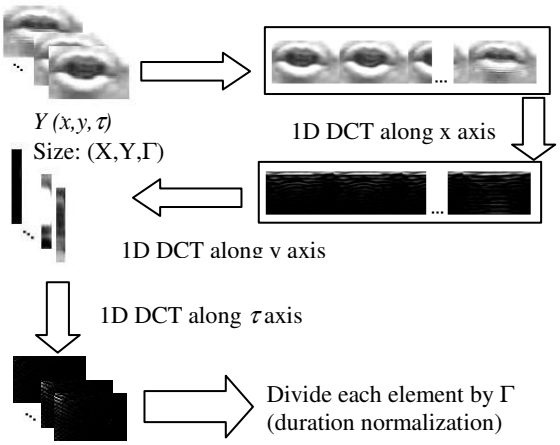


Figure 2: *Illustration of the extraction of 3D TV-DCT features from the sequence of images.*

Note that T and Γ represent the same time duration even if they may represent a different number of frames. The $\tilde{A}$ and $\tilde{V}$ features are time normalized by dividing them by T and Γ respectively to make them length-invariant. The process of extracting visual features is illustrated in Figure 2.

If the frequency sub-bands are in the perceptual scale like the 'mel' or 'bark' scale as used in this paper, then the vector of audio features with the first 'meti' coefficients ($\tilde{A}(1 \le p \le P, 1)$) are the same as the MFCCs of the central frame.

Thus these audiovisual features not only perform an information compression (since $P<<F$, $Q<<T$, $I<<M$, $J<<N$, $K<<\Gamma$), but also represent time-varying segments with different sampling rates as length invariant. This property is useful in the modeling, since the duration of the same phoneme may differ between different realizations in the training data. In some cases, the duration itself is useful and can therefore also be used as an additional feature. In our implementation we chose P=18, Q=3, I=13, J=19, K=4 by optimizing the Peak Signal to Noise Ratio (PSNR) between the original signal and the signals reconstructed from the chosen number of DCT coefficients. We leave out the first DCT component from both the visual and audio features to normalize for different intensities in different recordings.

Thus, we had 47 audio features, and 987 video features for each color (RGB) giving a total of 2961 video features, plus the duration of the sequence.

Another interesting benefit of such a feature representation is the conversion of time-varying data with different lengths to a time-varying representation with a fixed length feature vector. Classification can thus be performed by several powerful classifiers which are normally used to classify temporally stationary data, such as SVMs, Neural Networks etc. In our experiments, we used the MATLAB™ implementation of SVM, with a Radial Basis Function (RBF) kernel. The time duration could, in principal, encompass an entire utterance, but segments which span the duration of a phoneme or syllable are preferred. In the current set-up we only consider words pre-segmented into phonemes using a forced alignment of the phonemes [10] as described in Section 4.

3. Feature Selection

In order to ensure fast convergence of the SVM classifier, the number of features needed to be reduced. As a baseline, we applied Principal Component Analysis (PCA) to reduce the 47 audio features to 37, plus the duration feature, and the 2961 video features to 387, by keeping only 95% of the variance. Concatenating the two features vectors give a total of 424+1 features for the audiovisual case. However, the relative importance between the audio features and the video features for the classification task at hand is often unclear and it is far from certain that the features found with separate PCA on the two modalities are optimal for the audiovisual case. In addition, it may be of interest to further reduce the number of features. We try to solve this problem by using one filter based algorithm, namely Minimum Redundancy Maximum Relevance (MRMR) [11], and one wrapper based, Genetic Algorithms (GA) [12].

3.1. Minimum Redundancy Maximum Relevance

MRMR, being a filter based algorithm, the feature selection process is not connected to classification accuracy. The features are sorted according to the inherent relationships between the features and their discriminative abilities. It relies on estimating feature redundancy (selecting features that are dissimilar to each other) and relevance (maximize the contribution of the features towards classification), through a greedy search. The implementations of the two steps are denoted by Equations 2 and 3 respectively. Processing time varies linearly with the number of features to be retained. The mutual information for two discrete variables *(x, y)*, given their joint distribution probability $p_{xy}(x,y)$ and marginal densities $p_x(x)$ and $p_y(y)$, is defined as:

$$I(x,y) = \sum_{i,j} p_{xy}(x_i, y_i) \log\left(\frac{p_{xy}(x_i, y_i)}{p_x(x_i).p_y(y_i)} \right) \quad (1)$$

with the criteria

$$\min W_I \ , \quad W_I = \frac{1}{|S|^2} \sum_{xi,xj \in S} I(x_i, x_j) \quad (2)$$

$$\max V_I \ , \quad V_I = \frac{1}{|S|} \sum_{xi \in S} I(h, x_i) \quad (3)$$

S denotes the feature space, x_i the i^{th} feature and h the target classes. Both conditions (2)-(3) can be incorporated into a single criterion: max $(V_I - W_I)$. We used the MRMR algorithm implemented by Peng et al. [11]. Even though it does not tie the selection procedure to improvements in the classification, a hypothesis about the error rate is required for calculating the mutual information in Equation 1. This is provided by the SVM using the whole feature set.

3.2. Genetic Algorithms

While MRMR returns a compact set of N best features, they may not be the best combination of features for the task. In order to ensure fast convergence times for the SVMs, the poor set of features needs to be discarded. For this we use a wrapper based feature selection algorithm, which ties a particular combination of selected features to the performance of the classifier. We employ an implementation of GA [12] with a 4-fold cross validation scheme in order to select both the feature indices and to optimize the parameters of the SVM. Without good initial estimates, GAs are known for their long convergence times. With an initial number of 3000 features, the search space has roughly 2^{3000} combinations. Thus by forcing the initial feature population of the GA to correspond to the first few features selected by MRMR, we can constrain the search space, provide a good initial guess and thereby help the GA converge faster. Figure 3 illustrates how the GA helps in selecting the optimal number of features/parameters.

The entire process of creating pair-wise binary classifiers hence, consists of first extracting features from the auditory and video data sets from the two classes, then separating these into K different folds, applying MRMR to each fold, then applying the genetic algorithm to find the optimal features and the classifier parameters over all the folds. The optimum features and parameters are then used to train the SVM binary classifiers over each fold.

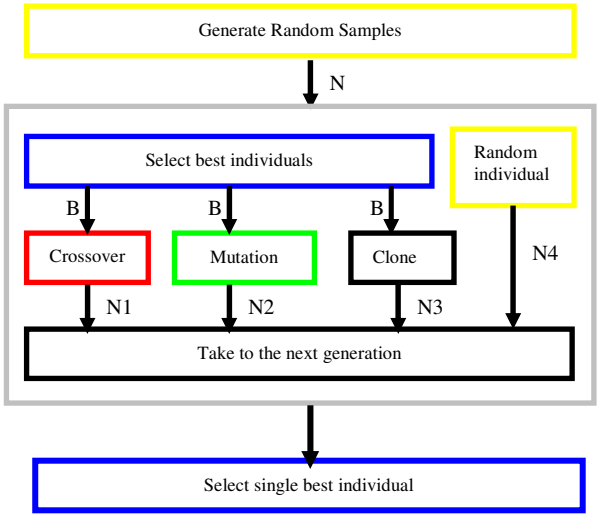


Figure 3: *Illustration of the Genetic Algorithm for feature selection. N1, N2, N3, N4 and B are the parameters (in this case set to 35, 5, 5, 5 and 5, respectively). An additional parameter 'k' is randomly chosen for each new Crossover and Mutation. The computations are repeated until the error rate falls below, or the number of iterations reaches, their respective thresholds.*

4. Data

An audiovisual corpus consisting of 118 Swedish one- or two-syllabic words was collected. The words constituted minimal pairs contrasting only between similar sounding vowels. The corpus was chosen so as to comprise all 18 vowels, long and short included, present in the Swedish language. Audio and video sequences of two native and two non-native speakers of Swedish pronouncing the 118 words were recorded. Figure 3a shows the setup for one of the speakers. The audio and video were aligned for every word in order to avoid drift due to different sampling rates, which were 16 kHz for the audio and 25 Hz for the video. The audio was segmented into the pronounced phonemes using a state-of-the-art HMM based aligner [10] and the corresponding video sequences were segmented using the time stamps provided by the aligner on the acoustic signal. In this paper we illustrate the algorithm by focusing on 6 specific cases of vowel mispronunciations that commonly occur among Swedish L2 learners:

/iː, ɪ/ *vs.* /ʏː, ʏ/ (e.g., rita [draw] *vs.* ryta [roar])

/ʏː, ʏ/ *vs.* /ʉː/ (e.g., sylar [awls] *vs.* sular [soles])

/eː, ɛ/ *vs.* /ø, œ/ (e.g., rest [remainder] *vs.* röst [voice])

/oː, ɔ/ *vs.* /uː, ʊ/ (e.g., båta [of avail] *vs.* bota [cure])

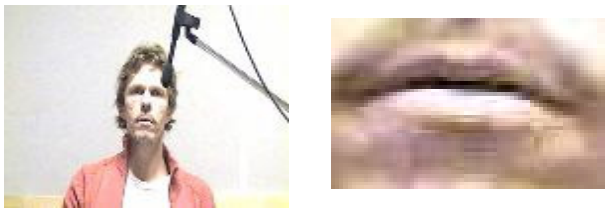
/ø, œ/ *vs.* /oː, ɔ/ (e.g., nös [sneezed] *vs.* nås [is reached])

/ʉː/ *vs.* /uː, ʊ/ (e.g., mus [mouse] *vs.* mos [mash])

Since not all vowels pronounced by the non-native speakers were correctly produced, two native speakers of Swedish judged the pronunciations and labeled them as correct or incorrect and only pronunciations labeled as correct by both the judges were accepted. In the current experiments we attempt discriminating between correctly pronounced instances of the contrasting vowels, but it should be noted that the classifier could as well be trained to discriminate between a correct pronunciation of a phoneme, and an incorrect one that does not constitute another valid phoneme in the target language.

For the video images, we used an ROI extraction algorithm [13] that extracts a rectangular area encompassing the lips, as shown in Figure 4b and Figure 5. The algorithm uses a particle filter to track the upper part of the face. The speaker's limited head movements then allows for a 2D template based deduction of the lips from the obtained position of the upper part of the face for each subject.

We performed two types of 4-fold cross-validation on the data that we had. The first was by splitting all the data for a particular class into 4 parts without considering who the speaker of the data was. We call this the 'Mixed Speaker' (MS) case. The second case, which we call the 'Speaker Excluded' (SE) case, is used for testing the performance on unknown speakers. This was an important evaluation aspect, because of the larger variability among the speakers as illustrated in Figure 5. We used a stratified 4-fold cross-validation where it was always ensured that data recorded for a particular speaker on whom the classification system was tested was not included in the training data. Within each fold of the cross-validation for the MS case, for every binary classifier, we had an average of 58 samples per class for training and around 19 samples for testing. The number of samples available for the SE case differed between speakers, since not all pronunciations for a particular speaker were labeled correct.



(a) (b)

Figure 4: *Left: The recording environment. Right: The lip region after automatic ROI extraction.*

Figure 5: *Inter-subject differences when producing the same vowel.*

5. Results and Discussion

Table 1 summarizes the error rates for the MS case in the baseline test, where the number of features was reduced using PCA, as described in Section 3. The audio only condition has a low error rate in most of the discrimination pairs, except for /oː, ɔ/ *vs.* /uː, ʊ/ and /iː, ɪ/ *vs.* /yː, ʏ/. This is expected, since these pairs are quite similar in the auditory domain. Since they are more distinct in the visual domain, because of differences in lip rounding (differences in protrusion between /oː, ɔ/ and /uː, ʊ/ and in rounding between /iː, ɪ/ and /yː, ʏ/), audiovisual discrimination could potentially improve the results. However, adding visual features selected by PCA, in fact, brings down the performance further for these pairs.

In addition, the accuracy rates also drop by a large amount for /yː, ʏ/ *vs.* /ʉː/ and [ʉː] *vs.* [uː, ʊ], where visual information can be misleading, due to the similarity in lip rounding. This highlights the basic problem in combining audiovisual features. Visual features being greater in number and not always informative may bring down the performance of the classifier. This asserts the need for good feature selection algorithms.

Table 2 shows the scenario where classification is done using the best 30 features selected by the MRMR algorithm. For most classification tasks, the MRMR selection improves the results in the acoustic only case as compared to using PCA, but the improvement is larger for the audiovisual case. The audiovisual improvement in the /iː, ɪ/ *vs.* /yː, ʏ/ case is quite striking. Evidently, MRMR succeeds in identifying the visual features which encode the lip rounding contrast between these vowels. However, there are some classification tasks where the audiovisual features degrade the performance as compared to audio only. It was found that by making a sub-selection of the best features as selected from the MRMR, a further reduction in error rates could be achieved. This was the motivation to try GAs, which can not only optimize the parameters 'sigma' for the RBF kernel but also reduce the feature set to a more optimum number.

Table 1: *Average Error rate (%) over 4-fold cross-validation when features are selected using PCA. The number within parenthesis indicates the error rate for always selecting the most likely class. Class 1 indicates the error rate when testing the candidates that belong to the first group and Class 2 is the error rate when testing on the second group.*

Classifier Class 1 / Class 2	Audio Only		Audiovisual	
	Class 1	Class 2	Class 1	Class 2
[iː, ɪ] *vs.* [yː, ʏ]	15 (51)	15 (51)	21 (51)	12 (51)
[yː, ʏ] *vs.* [ʉː]	8.2 (51)	2 (51)	31 (51)	19 (51)
[eː, ɛ] *vs.* [ø, œ]	0 (51)	6 (51)	4.6 (51)	11 (51)
[oː, ɔ] *vs.* [uː, ʊ]	16 (50)	3.1 (50)	30 (50)	25 (50)
[ø, œ] *vs.* [oː, ɔ]	3.0 (51)	1.5 (51)	4.6 (51)	0 (51)
[ʉː] *vs.* [uː, ʊ]	2.2 (50)	0 (50)	14 (50)	7 (50)

Table 2: *Average Error rate (%) over 4-fold cross-validation for binary classification with 30 best features selected using MRMR. Class 1 indicates the error rate when testing the candidates that belong to the first group and Class 2 is the error rate when testing on the second group.*

Classifier Class 1 / Class 2	Audio Only		Audiovisual	
	Class 1	Class 2	Class 1	Class 2
[iː, ɪ] *vs.* [yː, ʏ]	15	8.1	0	1.5
[yː, ʏ] *vs.* [ʉː]	6.1	4.1	8.3	10
[eː, ɛ] *vs.* [ø, œ]	0	4.5	3.1	0
[oː, ɔ] *vs.* [uː, ʊ]	7.8	4.7	9.5	11
[ø, œ] *vs.* [oː, ɔ]	1.5	1.5	0	1.5
[ʉː] *vs.* [uː, ʊ]	0	0	4.7	0

Table 3 shows how optimizing both the feature set and the SVM parameters using GAs contributes to keeping the performance of the audiovisual features better than or equal to

the audio only features. The total number of features required when the visual features are available is also reduced in many cases. The two step feature selection approach in addition reduced the usually long convergence times of GAs to a few minutes per classifier.

Table 3 also indicates the performance in the speaker independent (SE) scenario, giving a more realistic idea about the performance of this framework when tested with data from an unknown speaker, which would be the case in an audiovisual mispronunciation detector in CAPT. The initial results indicate that the framework is rather robust towards different speakers, except for the discrimination between [oː, ɔ] and [uː, ʊ], in spite of the variability between the speakers. However, these results may be optimistic, since all the subjects were male and within a small age range.

Table 3: *Average Error rate (%) over 4-fold cross-validation for binary classification after selecting 50 features using MRMR and then applying the GA to select an optimized feature/parameter set for the MS and SE cases. The cross-validation is stratified for the SE case. The number within parenthesis indicates the number of best features selected. For the audiovisual case, the number of both audio and visual features is mentioned as (audio features|video features).*

Classifier Class 1 / Class 2	Audio Only		Audiovisual	
	MS	SE	MS	SE
[iː, ɪ] *vs.* [yː, ʏ]	8.8 (29)	12	0 (0|4)	0
[yː, ʏ] *vs.* [ʉː]	1.0 (27)	0	1.0 (27|0)	1.0
[eː, ɛ] *vs.* [ø, œ]	0 (23)	0	0 (3|1)	0
[oː, ɔ] *vs.* [uː, ʊ]	3.1 (28)	3.1	2.4 (5|7)	8.7
[ø, œ] *vs.* [oː, ɔ]	0 (20)	0	0 (11|13)	0
[ʉː] *vs.* [uː, ʊ]	0 (31)	0	0 (5|6)	0

6. Conclusions and Future Work

This paper describes a novel framework to integrate time-varying audiovisual features in order to detect specific mispronunciations with sparse data. The detection framework is a series of binary classifiers which distinguish between commonly occurring phonemic ambiguities in a language, in this case Swedish. Using six pairs of similar vowels which commonly confound Swedish L2 learners, we demonstrated a classifier which could reach up to 95-100% classification accuracy for the tested vowel pairs. The experiments are indeed very limited in terms of both number of speakers and the test corpus. We do hence not argue that the experiments are a proper evaluation of the framework for mispronunciation detection, but rather that they illustrate the properties of the proposed framework, including the ability to define classifiers based on a very sparse training data set.

The framework can be extended to other types of phonemes such as transients because of the TV-DCT, which captures the dynamics of the audiovisual sequence. The feature selection algorithms presented here helped integrate the audiovisual features so as to utilize the advantage of each modality depending on the task. It is clear that audio features contribute to most of the information in this classification task, but visual features are more informative and useful for certain pairs of phonemes. Finally, this paper establishes a means to apply powerful classifiers like the SVM to time-varying data, which could lead to interesting future work.

Future work would also include testing this framework with other types of phonemes. While we know that this framework performs reasonably well even in sparse data, it would be interesting to see if it compares favorably with other state-of-the-art visual speech recognizers when larger quantities of data are available. Another interesting area of research could be to see whether the framework relies more on visual features in the presence of noise in the audio. A realistic scenario with L2 learners using a desktop microphone and web-camera would be the ideal test environment for this framework.

7. Acknowledgement

We would like to thank the Swedish Research Council projects 80449001, Computer-Animated Language Teachers (CALATEA) and 348-2005-6161 Audiovisual Detection of Errors in Pronunciation Training (ADEPT) and the ERASMUS student exchange program for their fincancial support.

8. References

[1] Dupont, S. and J. Luettin, J., "Audiovisual speech modeling for continuous speech recognition", IEEE Trans. Multimedia, 2(3): 141-151, 2000.

[2] Potamianos, G. and Graf, H., "Discriminative training of HMM stream exponents for audiovisual speech recognition", Proc. ICASSP, vol. 6, pp. 3733-3736, 1998.

[3] Cootes, T., Taylor, C., Cooper, D. and Graham, J., "Active shape models - their training and application", Proc. Comp. Vision and Image Understanding, 61(1):38–59, 1995.

[4] Chiou, G. and Hwang, J-N., "Lipreading from color video", IEEE Trans. Image Processing", 6(8):1192–1195, 1997.

[5] Matthews, I., Cootes, T., Cox, S., Harvey, R. and Bangham, J.A., "Lipreading using shape, shading and scale", Proc. AVS.P, pp. 73-78, 1998.

[6] Duchnowski, P., Meier, U. and Waibel, A., "See me, hear me: Integrating automatic speech recognition and lipreading", Proc. ICSLP, Yokohama, Japan, pp. 547–550, 1994.

[7] Potamianos, G., Neti, C., Gravièr, G., Garg, A. and Senior, A.W., "Recent Advances in the Automatic Recognition of Audiovisual Speech", Proc. IEEE, 91(9): 1306-1326, 2003.

[8] Cortes, C., and Vapnik, V., "Support-Vector Networks", Machine Learning, 20(3):273-297, 1995.

[9] Neiberg, D., Laukka, P., and Ananthakrishnan, G. "Classification of Affective Speech using Normalized Time-Frequency Cepstra". In Press, Proc. Prosody, 2010.

[10] K. Sjölander, M. Heldner, "Word level precision of the NALIGN automatic segmentation algorithm", Proc. Fonetik, 2004.

[11] Peng, H., Long, F. and Ding, C., "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy", IEEE Trans. Pattern Analysis and Machine Intelligence, 27(8):1226-1238, 2005.

[12] Goldberg, D., "Genetic Algorithms in Search Optimization and Machine Learning", Addison Wesley, 2005.

[13] Kjellström, H. and Engwall, O., "Audiovisual-to-articulatory inversion", Speech Communication 51(3):195–209, 2009.