Using Artificial Neural Networks in Lip-reading

Jacek C. Wojdeł   Leon J. M. Rothkrantz

Knowledge Based Systems Group,
Delft University of Technology,
Zuidplantsoen 4, 2628BZ Delft, The Netherlands
Phone: +31 15 278 8543
Fax: +31 15 278 7141

J.C.Wojdel@cs.tudelft.nl   L.J.M.Rothkrantz@cs.tudelft.nl

Keywords: Lip-reading (Speech-reading), Image Processing, Multimodal Speech Recognition, Artificial Neural Networks

Abstract

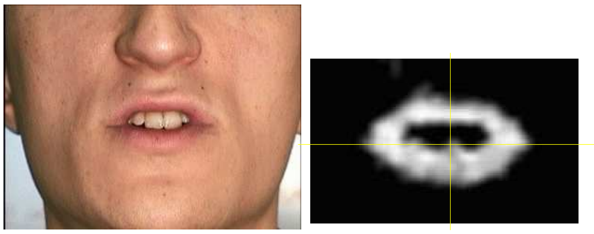
In this paper we present a model for video sequence processing that is aimed at extraction of the data describing the mouth shape in the image. We show how artificial neural networks (ANNs) can be used in both the feature extraction and classification part of the recognition system. Appropriate ANN architectures are proposed depending on the task being performed.

1 Introduction

When perceiving speech, human beings use not only the auditory signal; they use also other modalities such as lip movements, hand and body gestures etc. The importance of those modalities grows immensely in noisy environment or in case of hearing disabilities. It was shown that even well hearing subjects do lip-reading as a part of the speech recognition process. The influence of lip-reading is present even in perfect auditory conditions and it grows steadily with the degradation of the auditory signal [1].

Automated lip-reading can be used in combination with audio-based speech recognition to improve the recognition rate of a system [2]. In a little different approach, the visual information can be used to enhance the quality of the speech signal when transmitted on a distance [3].

Because of the huge amount of data contained in the visual signal it is usually not feasible to apply the signal processing techniques on it directly. Commonly, specific features or parameters are derived from each video frame and further processed. It is usually done by fitting a geometrical model on the image and using model parameters as the input for the recognition system [4]. In the following sections we will describe the multiple ways in which Artificial Neural Networks (ANNs) can be used to perform feature extraction and recognition that does not rely on a geometrical lip model.



(a) original   (b) hue-filtered

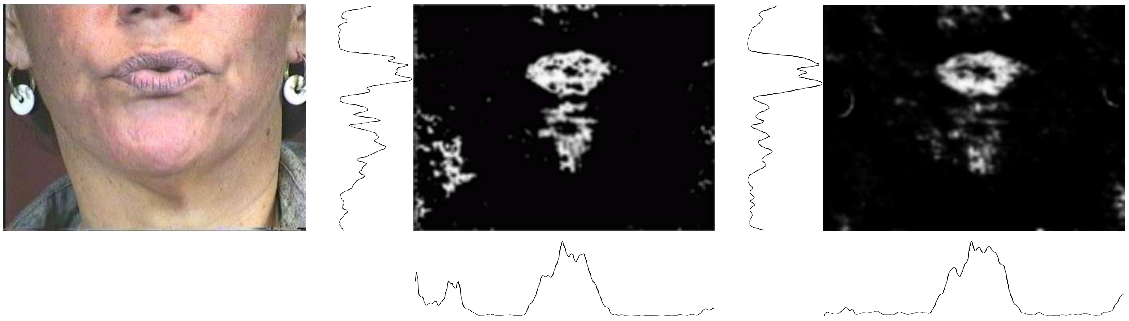
Figure 1: Parabolic shape hue-based lip-selective filtering example, (a) original image, (b) hue-filtered image with the mean value of resulting distribution depicted ($h_0 = 0$, $w = 0.25$).

2 ANNs in image processing

As the first step in video processing, we have to somehow locate the mouth in the image. Fortunately for us, when using the video containing only the face, the lips have a distinct coloring that allows finding them without a need for complicated object recognition techniques. In order to utilize this fact, we need to apply some sort of *lip-selective* filter to the image.

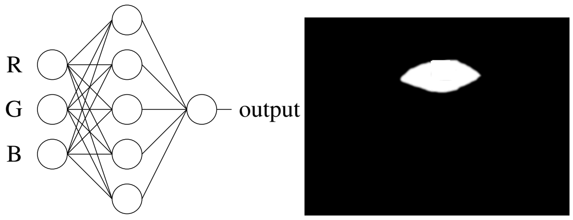
Although ANNs are widely used as a mean of signal processing, they are rarely utilized in lip tracking. The most common *lip-selective* filter that is used in this task is based on separation of specific *hue* values around the red color.

The hue-based filtering was first proposed in [4] and incorporates a parabolic shaped filter that cuts off



(a) original (b) hue-filtered (c) ANN-filtered

Figure 2: Using different lip-selective filters, (a) original image, (b) hue-filtered image with optimal filter parameters, (c) ANN filtered image. Both of the filtered images are presented together with their signatures.



(a) (b)

Figure 3: (a) Simple feed-forward neural network used in *lip-selective* filtering. (b) Target values to which the network was trained on the first frame of the video-sequence as depicted in Fig. 2.

the colors that fall outside a given interval:

$$F_{hue}(h) = \begin{cases} 1 - \frac{(h-h_0)^2}{w^2}, & |h - h_0| \leq w \\ 0, & otherwise \end{cases} \quad (1)$$

The filter is defined by the center of the interval h_0 and its half-width w. Filtering the image starts with calculating the *hue* color component of a given pixel and then passing it through the F_{hue} function. In order to select lips in the image, the value of h_0 must be around 0 (red color) and w about 0.2 (see Fig. 1). Both of the values have to be adjusted for a specific video sequence in order to retain the quality of the filtering.

This kind of filter is extremely simple and computationally efficient. It has however its shortcomings. There are two main problems with this type of the filter; its robustness highly depends on the color representation of the image and on the complexion of the subject. The color representation that can be achieved in a recorded image depends on the illumination conditions, the quality of the camera and the type of image encoding being used. On the other hand, even at proper conditions some people don't have enough *hue* variation between the skin and lips in order to provide good filtering results (compare Fig. 1b with Fig. 2b).

The effectiveness of the *hue* based filter is based on the assumption that the *hue* component of the observed color is illumination independent. This however is not entirely true. In case of typical consumer CCD cameras, the color reproduction quality degrades significantly when the sequence is recorded in sub-optimal illumination conditions. Not only changes in the light source influence the observed color, but also self-occlusion shadows on the face. Generally speaking, the lower intensity of the recorded pixel, the higher the noise factor in the *hue* component. Therefore the *hue* component itself is not sufficient for proper lip selection (see Fig. 2b). Instead of putting additional constraints on the filtered color (such as a threshold on its *value* or *saturation*), we decided on using a black-box approach. We trained a very simple feed-forward neural network and used it as a filter. The network that is used has only 5 neurons in a hidden layer and one output neuron (see Fig. 3). It is fed with the *RGB* values from a pixel in the image. The small size of the network allows for on-line training, so that we can use different networks for different video sequences as long as the first frame in the video sequence is prepared with the target filtering values. This can be simplified even further by using just a box placed tightly around the mouth as the set of target values for ANN. Although the box would contain some amount of the pixels with a skin color, they will be heavily outnumbered by the pixels with a lip color. The smoothing properties of a small ANN will assure that it will train to the proper state.

Even with optimal filtering of the mouth, it is possible that in some cases the filtered image contains unwanted artifacts. A typical example can be seen in Fig. 2, where the shadow under the chin cannot be distinguished from the lips. In order to reduce the impact of such occurrences, we must extract a region of interest (ROI) from the whole image. We do it by using vertical and horizontal signatures of the image (see Fig. 2(b-c)). Each signature is a vector containing sums of the pixel-values for a specific row or column. The boundaries of ROI can be found by thresholding

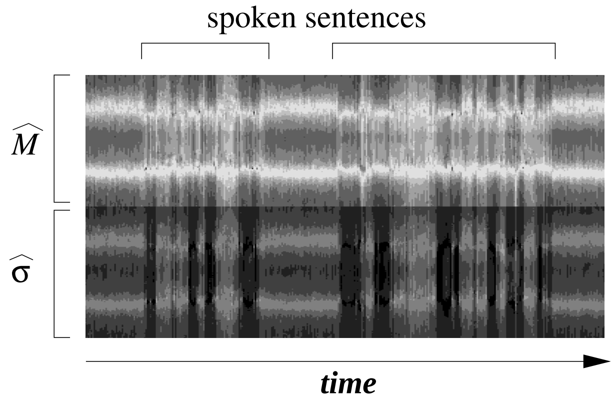


Figure 4: Pairs of $\widehat{M}(\alpha)$ and $\widehat{\sigma}(\alpha)$ vectors extracted from a video-sequence. The periods of silence around two spoken sequences can be seen clearly.

of the signatures in combination with looking for raising and falling edges in each of the vectors.

At this stage the superiority of the ANN based approach to the *hue* based filtering can be clearly seen. The ANN be retrained in the calibration process to compensate for the changes of the illumination conditions or person specific separability of the lip-color giving in the end much clearer signatures of the resulting image. We can see that by simply thresholding the signatures at 30% of the maximal value (see Fig. 2c). Even if the artifacts are big enough to appear on the image signatures, their influence can be removed by assuming temporal consistency of ROI boundaries between consecutive frames. Only the part of the filtered image that lays inside ROI is further processed.

3 Feature vectors extraction

From this point on, the filtered image is being treated as a bivariative distribution $I(X,Y)$ (after normalization). The mean of this distribution: $[EX, EY]$ approximates accurately and in a stable way the center of the mouth. Using this value, we transform the image into polar coordinates:

$$J(a,r) = I(EX + r\cos(a), EY + r\sin(a)) \quad (2)$$

and we concentrate on the conditional distributions $J(a=\alpha, r)$. We define therefore the following functions of mean and variance values for any angle α:

$$\begin{aligned} M(\alpha) &= \frac{\int_r J(\alpha,r)\cdot r\cdot dr}{\int_r J(\alpha,r)dr} \\ \sigma(\alpha) &= \frac{\int_r J(\alpha,r)\cdot(r-M(\alpha))^2 dr}{\int_r J(\alpha,r)dr} \end{aligned} \quad (3)$$

As the image is discrete rather that continuous, all of the values are obtained from summation rather than integration, so we only operate on estimations of those values, namely $\widehat{M}(\alpha)$ and $\widehat{\sigma}(\alpha)$. The vectors resulting from sampling of those functions for one of the video-sequences can be seen in Fig. 4.

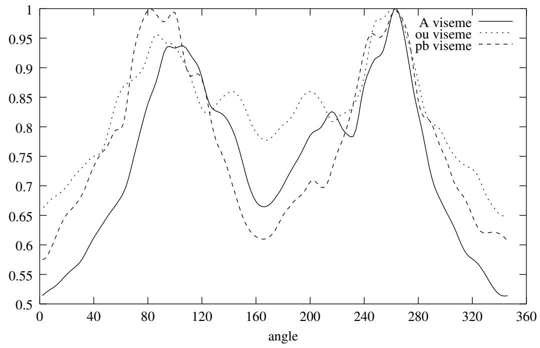


Figure 5: Average $\widehat{M}(\alpha)$ values for specific visemes in one of the video sequences.

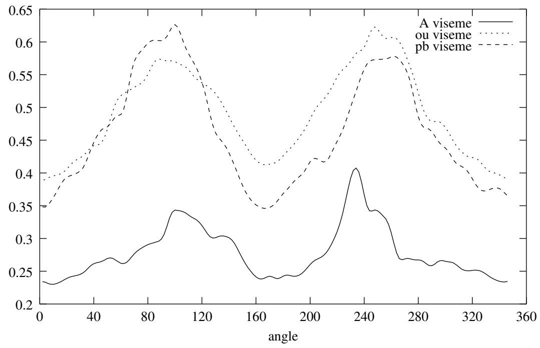


Figure 6: Average $\widehat{\sigma}(\alpha)$ values for specific visemes in one of the video sequences.

The value of $\widehat{M}(\alpha)$ for a specific angle relates directly to the distance of the lip from the center of the mouth in a given direction. Therefore a vector constructed from those values for a number of angles describes the shape of the mouth on a given image. Such shape profiles were gathered from a video sequence of spoken Dutch and grouped by the visemes. The Fig. 5 shows three profiles for visemes `[A]`, `[o u O y Y 2: 9:]` and `[p b m]`. In order to obtain scale independency the vectors were scaled so that all their values fit in the $\langle 0, 1\rangle$ interval. It can be seen that the `[ou]` viseme can be distinguished from the others by its relatively high $\widehat{M}$ value in the middle. There is however no feasible way of distinguishing between `[A]` and `[p b m]` visemes. The lip shapes that correspond to those visemes are completely different (mouth closed versus mouth opened), but scaling of feature vectors removed any of those differences.

As the values of $\widehat{M}$ alone are not sufficient for useful data extraction from the video-sequence, we use also additional information about variance of conditional distributions ($\widehat{\sigma}$). Those values are also scaled in order to obtain size independency, but the scaling factor is directly determined by the values of $\widehat{M}$ rather than $\widehat{\sigma}$. Therefore the resulting feature vectors provide additional information about the lips. As an example, the `[A]` viseme clearly shows much lower $\widehat{\sigma}$ values than the other visemes (see Fig. 6). Obviously, for a wide stretched mouth, the lips appear thinner than for a mouth closed, when related to the overall size of the mouth.

The last stage of feature vectors extraction is subsampling of the $\widehat{M}$ and $\widehat{\sigma}$ functions into vectors that can be further processed. Obviously, the choice of the sampling rate (vector dimensions) is a compromise between accuracy and processing efficiency. The longer the vectors, the more information on the original distribution they contain but also the longer it takes to extract and process them. After some experiments with the data we chose to use the 18-dimensional vectors for both $\widehat{M}$ and $\widehat{\sigma}$ (see [5]).

4 Validation of extracted data

In order to validate the correctness of the extracted data, we performed two different experiments related to lip-reading. The first one aimed at finding the boundaries of the spoken sentences in the video sequence, the other was to discriminate between vowels and consonants in it. In both cases we used a corpus of video recordings consisting of 30 Dutch sentences spoken by three different people. The subjects were asked to speak most of the sentences normally, some in a slower rate and to whisper the rest. In this way we obtained a varied data set that was then manually labeled with transcriptions of the spoken utterances.

It is important to notice that most of the information in the incoming feature vectors is in fact dynamically encoded. There is for example no significant difference between the lips being closed during the silence between sentences and the lips just before pronouncing the `[p b m]` viseme. There is however a huge difference in the dynamic properties of the feature vectors in a short time around the observation. The silence for example is characterized by only small fluctuations of the feature vector (independent on whether the mouth is actually closed or open at that time). In order to visualize this we used a Sammon mapping [6] onto a 2D plane using either single observation frames or their three-frame neighborhood (see Fig. 7). As it can be seen, using even a small three-frame time window provides the data that allows for a nice cluster of the `[p b m]` viseme to be formed. That observation leads us to the conclusion that the classification tool used further in the recognition system must deal with the temporal dependencies in the incoming data.

There are several ANN architectures designed for such task, of which the most prominent are Time Delayed Neural Networks [7] (TDNN) and Partial Recurrent Neural Networks [8] (PRNN). Both architectures were tested on the data extracted from the collected recordings. The PRNNs proved to be superior to TDNNs (see [5]). Specifically, the Elman Hierarchical NN (and its single-layer equivalent; Jordan NN) performed best of all tested architectures. Therefore the following sections describe the results obtained with this network type.

Table 1: Speech onset/offset detection results for networks trained on different subsets of the recordings.

	training set		
	normal	slow	whisper
avg. output error	0.157	0.068	0.123
avg. detection delay	0.37s	0.17s	0.34s
misclassified	10.56s	2.08s	5.13s

Table 2: Vowel/consonant discrimination results. Symbols are: '0' - inter-sentence silence, 'c' - consonant, 'v' - vowel, '-' - not labeled.

	Network response			
Target	0	c	v	-
0	**70.5%**	14.3%	3.8%	11.4%
c	2.6%	**72.2%**	19.5%	5.2%
v	0.0%	26.5%	**67.3%**	6.1%
-	1.7%	45.2%	12.2%	**40.9%**

4.1 Speech boundaries

As it can be seen in Fig. 4, the periods between sentences can be spotted in the data set even just by eye-inspection of the data stream. Therefore developing a classifier that would localize them is not a big problem. After experimenting with several ANN architectures [5], we settled with a Jordan recurrent NN. Table 1 summarizes the results of testing the networks that were trained on different subsets of our data. All networks were tested on about 3m40s of recordings containing all speech patterns (normal / slow / whisper). The table contains the average output error of the network as compared to target labeling of the data, the average delay of onset/offset detection (switching between two output states) and the total time of periods in which the network produced inappropriate output.

4.2 Vowel/consonant discrimination

In a second experiment, we investigated a problem of discrimination between two classes of the phonemes being spoken: vowels and consonants. We performed several experiments with different architectures of ANNs that were trained to put the incoming data into one of three different classes: vowel,

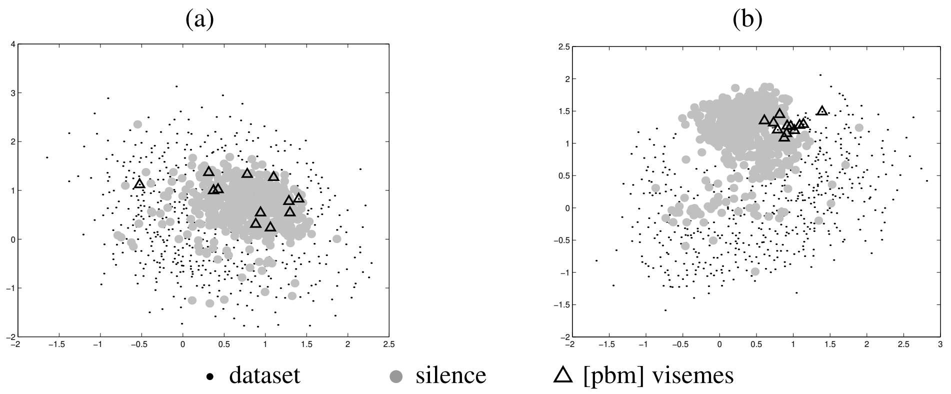


Figure 7: 2D visualization of data using Sammon mapping for (a) single feature vectors and (b) three concatenated vectors from consecutive video-frames.

consonant and silence between utterances.

The Elman Hierarchical Neural Network containing 3 hidden layers with respectively 20, 40 and 20 neurons in each of them proved to be the most efficient in vowel/consonant discrimination. The overall results of recognition are summarized in Table 2. Together with the three classes that can be recognized by the network (*0*, *v*, *c*) this table contains also a fourth column labeled (–), which corresponds to the output patterns that should not be classified as any of the three classes. Such patterns occur in our dataset for example in slow speech between some phonemes that with a space between them that cannot be considered as a silence yet. In the experiments we assumed that the network response was unspecified if none of the network's outputs had a higher activation than a threshold value of 0.1.

4.3 Comparison to HMMs

There is a possiblity to use a different recognition scheme than the ANNs. In the speech processing field the Hidden Markov Models (HMMs) are a well established mean of extracting time dependent information from the incoming speech signal. The HMMs proved to be much better in this field than ANNs and so seem to be a valid contender also in the lip-reading tasks. In order to fairly compare ANNs and HMMs we have to take into consideration their complexity, not only the obtained recognition rates. The number of free parameters that have to be estimated during training is summarised in the Tab. 3. We did experiments with HMMs containing 2 or 3 emitting states per model and with a different number of models representing recognized classes. In the simplest case we used only three models; one for the silence, vowel and consonant class. In further experiments we used multiple models for the different vowel and consonant viseme groups. We added also a short pause model derived from the silence model. The bold values in Table 3 represent those models that were actually tested in our experiments.

Despite the number of free parameters comparable or exceeding the number of parameters available in the NN approach, in none of the conducted experiments did HMMs achieve a recognition rate substantially higher than a random choice between each of the three classes. The resulting recognition rate varied between 35% and 45% with no class being recognized in any better way than the others. Such bad results suggest that the data is in some way not suited for HMM based recognition.

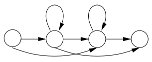
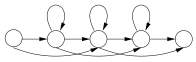
One of the possible interpretations of such a poor performance can be derived from our experiences with the more complex HMMs. In this case there are three emiting states in each model. In the initial experiment setup at least two of them were to be observed (there were no transitions $0 \rightarrow 2$ or $2 \rightarrow 4$). In our dataset there are however consonants that apear only in a single video frame and therefore cannot be modeled in this way. We were forced therefore to add the missing transitions. This suggests that there might be a fundamental problem with the sampling frequency of the lip-reading data that cannot be handled directly by HMMs. The problem can be handled by artificially subsampling the incoming data, adding transitions to the models or using models for larger speech structures such as words or syllables. This all will have to be investigated in the future.

5 Conclusions and future work

The proposed method of data extraction from the video sequence proves to be feasible in lip-reading related applications. The method's main advantages are it's simplicity and computational effectiveness. Those advantages are crucial when it comes to a real-time applications. The whole video processing is done as a single flow of rather low-level operations without need for per-frame computational heavy tasks (such as model optimization or template matching).

There are still some problems that have to be

Table 3: Number of free parameters in different recognition models.

models for:			Elman NN
silence, vovel, consonant	876	**1317**	4872
silence, multiple vowels, consonant	2920	**4390**	–
slience, multiple vowels and consonants	**4672**	**7024**	–

solved before applying this approach to the lip-reading system. For example the accuracy of *lip-selective* filtering depends on the parameters such as a set of ANN weights which cannot be set automatically yet. If we treat them however as a part of a calibration process that has to be done only once on the beginning of the session, we have a choice of using more sophisticated (and possibly computationally expensive) methods of setting these parameters.

There are some doubts whether the recurrent neural networks are actually suitable for the lip-reading task. In the very similar application of continuous speech recognition, the ANNs lost the battle to the Hidden Markov Models that give a more flexible way of handling time dependencies and contextual information. It has to be investigated however, if the same holds for the lip-reading with its inherently lower sampling rate and higher ambiguity. However at the current stage of our research we can say that the low sampling rate that is inherent to the video signal causes some serious problems with the HMM based recognition. The tools and techniques used in speech processing cannot be therefore simply applied on the lip-reading task without some deeper reconsideration.

In the near future, a bigger corpus of video recordings must be gathered in order to proceed with the development of the system for continuous lip-reading. Having a significantly larger set of recordings (and prompts) will allow also investigation in how much the results depend on the contextual information extracted by an Elman network.

References

[1] A. Adjoudani, T. Guiard-Marigny, B. Le Goff, L. Reveret, and C. Benoit. A multimedia platform for audio-visual speech processing. In Kokkinakis et al. [9].

[2] S. Nakamura, R. Nagai, and K. Shikano. Improved bimodal speech recognition using tied–mixture HMMs and 5000 word audio–visual synchronous database. In Kokkinakis et al. [9].

[3] L. Girin, G. Feng, and J.L. Schwartz. Noisy speech enhancement by fusion of auditory and visual information: a study of vowel transitions. In Kokkinakis et al. [9].

[4] Tarcisio Coianiz, Lorenzo Torresani, and Bruno Caprile. 2D deformable models for visual speech analysis. In Stork and Hennecke [10].

[5] Jacek C. Wojdeł and Leon J. M. Rothkrantz. Visually based speech onset/offset detection. In *Proceedings of Euromedia2000*, Antwerp, Belgium, May 2000.

[6] B. D. Ripley. *Pattern Recognition and Neural Networks*, chapter Multidimensional scaling, pages 305–311. Cambridge University Press, 1996.

[7] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme recognition using time delay neural network. IEEE *Transactions on Accoustics, Speech and Signal Processing*, 37:382–339, 1989.

[8] J. L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 190.

[9] G. Kokkinakis, N. Fakotakis, and E. Dermatas, editors. *Proceedings of ESCA, Eurospeech97*, Rhodes, Greece, 1997. ESCA.

[10] David G. Stork and Marcus E. Hennecke, editors. *Speechreading by Humans and Machines*, volume 150 of *NATO ASI Series, Series F: Computer and Systems Sciences*, Berlin, 1995. Springer Verlag.