Lip-Reading with Limited-Data Network

Adriana Fernandez-Lopez
Engineering School (DTIC)
Universitat Pompeu Fabra
Barcelona, Spain
adriana.fernandez@upf.edu

Federico M. Sukno
Engineering School (DTIC)
Universitat Pompeu Fabra
Barcelona, Spain
federico.sukno@upf.edu

***Abstract*—The development of Automatic Lip-Reading (ALR) systems is currently dominated by Deep Learning (DL) approaches. However, DL systems generally face two main issues related to the amount of data and the complexity of the model. To find a balance between the amount of available training data and the number of parameters of the model, in this work we introduce an end-to-end ALR system that combines CNNs and LSTMs and can be trained without large-scale databases. To this end, we propose to split the training by modules, by automatically generating weak labels per frames, termed visual units. These weak visual units are representative enough to guide the CNN to extract meaningful features that when combined with the context provided by the temporal module, are sufficiently informative to train an ALR system in a very short time and with no need for manual labeling. The system is evaluated in the well-known OuluVS2 database to perform sentence-level classification. We obtain an accuracy of 91.38% which is comparable to state-of-the-art results but, differently from most previous approaches, we do not require the use of external training data.**

***Index Terms*—Lip-reading, Visual Speech, Deep Learning**

I. Introduction

In the last decades, there has been a growing interest in Automatic Lip-Reading (ALR) systems. Similarly to other computer vision applications, methods based on Deep Neural Networks (DNNs) have permitted to substantially push forward the achievable performance.

The successful construction of end-to-end DNNs generally faces two main issues: a) the need for big amounts of training data; b) the need of complex models with millions of parameters that take a lot of time to train. In the context of ALR, data have been so far an important limitation, given that most audio-visual databases suitable for ALR are not sufficiently large or do not cover enough vocabulary to train end-to-end architectures that generalize well. Moreover, acquisition of new databases is challenging, especially due to the need for appropriate labeling (e.g. text or phonemes aligned with the video stream), which is time-consuming and error-prone.

A widespread alternative to avoid having to train DNNs from scratch, is to use pre-trained models. Unfortunately, it is difficult to find available models specifically trained for lip-reading. Thus, some authors have explored the use of pre-trained models designed for other computer vision applications, e.g. AlexNet, VGG, GoogLeNet or ResNet [1]–[4]. Those pre-trained models have shown to behave well in several classification scenarios, but the fact that they were not trained specifically for ALR leads to sub-optimal performance. For example, Chung et al. [2] reported experiments with two networks, both evaluated in the same dataset. The first one was trained from scratch for lip-reading and reported state-of-the-art performance; the second one combined the VGG-M network (pre-trained on the ImageNet database) with a Long-Short Term Memory (LSTM) network and reported poor accuracy. Thus, building ALR systems based on DNNs remains a time consuming and effort-intensive task.

In this work, we introduce an architecture that combines Convolutional Neural Networks (CNNs) and LSTMs, which can be trained without the need for large-scale databases. We propose to use weakly-supervised learning to label the data in an automatic manner in terms of visual units. These weak visual units are representative enough to guide the CNN to extract meaningful features that, together with context information, are sufficiently informative to train an ALR system in a very short time and with no need for manual labeling. The system is evaluated in the well-known OuluVS2 database [5] to perform sentence-level classification, and it obtains an accuracy of 91.38% which is comparable to state-of-the-art results but, differently from most previous approaches, it does not require the use of external training data.

II. Related work

As highlighted in the recent review in [6], the most promising DNN architectures for ALR stand out as the combination of CNNs and Recurrent Neural Networks (RNNs) (i.e. LSTMs or Gated Recurrent Units (GRUs)), which have achieved the highest Classification Rates (CR) so far [2]–[4], [7]–[9]. These CNN-RNN architectures, however, have proven to be especially data-hungry to train properly. In this work, we are interested in systems that perform sentence-level classification without the need for large-scale databases. Specifically, we will focus on those systems that have been evaluated in OuluVS2 because it is a widely used small-scale database.

The design of end-to-end architectures for small-scale databases is challenging because there are not enough samples to successfully train DNNs that produce state-of-the-art performance. Thus, DNN architectures evaluated in OuluVS2 have mainly followed two strategies: a) to use pre-trained models to avoid having to train DNNs from scratch; b) to deal with low resource data designing alternative architectures or data augmentation (DA) techniques. Among systems that use pre-trained models we find the work of Saitoh et al. [1]

978-9-0827-9703-9/19/$31.00

who propose to re-train 3 well-known CNNs (NIN, AlexNet and GoogLeNet) to perform phrase classification, obtaining the highest CR with GoogLeNet (85.60%) for frontal-view experiments. In contrast, Chung and Zisserman [2] proposed two models based on CNNs and LSTMs. The differences between the models fall on the CNN. The first model uses a pre-trained CNN, known as VGG-M, which was pre-trained on the ImageNet dataset [10] while the second model, named SyncNet, was pre-trained using the LRW dataset [11]. For the VGG-M based model they obtained a CR of 31.90%, while for the SyncNet based model they reported the state-of-the-art CR of 94.10%. In another work from them [11], a CNN system was pre-trained on the LRW dataset to perform phrase classification by processing video segments of 1-second at a time and reported 93.20% of CR.

Other researchers proposed the construction of ALR systems without using external data. Among them we firstly find the work from Lee et al. [7], who proposed to directly train from scratch an end-to-end network based on CNNs, LSTMs and Fully Connected (FC) layers, without using additional external data nor special training techniques to deal with the shortage of data. In this way, they reported 81.10% CR, which is 13% below the state-of-the-art performance. Consequently, researchers decided to deal with low resource data by augmenting the corpus or by exploring alternative architectures that deviate from the main CNN-RNN trend. For example, Fung and Mak [12] proposed an end-to-end model that follows the CNN-RNN baseline, but where a huge DA was crucial to circumvent the issue of insufficient training data. Their model fuses 3D-CNNs and BiLSTM together with maxout activation units and reported 87.60% CR in frontal-view experiments. On the other hand, Petridis et al. [13]–[15] proposed three alternative architectures based on an encoding network combined with directional and bidirectional LSTMs which were trainable without adding external data, reporting 84.50% CR, 91.80% CR and 91.80% CR, respectively for frontal-view experiments. In contrast to the above methods, in this work we propose to follow the main trend of most successful ALR systems, based on the CNN-RNN architectures. We show that, by appropriately adding weak intermediate labels to split the training process, we are able to get near state-of-the-art performance without the need for external data.

III. END-TO-END LIMITED DATA NETWORK

We introduce our Limited Data Network (LDNet), which consists of a visual module (CNN) followed by a temporal module (LSTM) which outputs the spoken phrase. The visual module receives color images of the mouth as input and extracts visual features that encode the mouth appearance. Then, for each frame, the output of the CNN is the input to a temporal module based on LSTMs that incorporates temporal context and accumulates the contribution of each frame to return the estimated phrase at the end of the sequence.

Specifically, the visual module architecture (Fig. 1) is based on VGG-M [16] because it was shown to perform well in classification tasks and contains fewer parameters than other

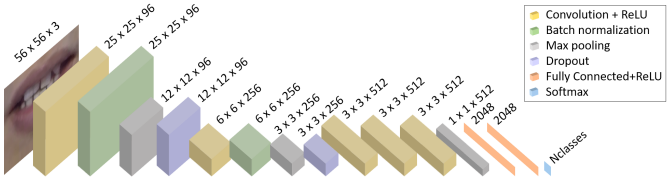

Fig. 1. The visual module architecture of LDNet. It inputs RGB images and outputs the most probable visual unit.

VGG models (e.g. VGG-16 or VGG-19 [10], [17]), leading to faster training [11]. The model contains 5 convolutional layers combined with batch normalization, max pooling and dropout, followed by two FC layers. The temporal module consists of a cascade of two LSTM layers with 256 hidden units which perform phrase classification at the end of the sequence, only after the whole stream has been processed.

A. Training with limited data

If we attempt to train an end-to-end system with a small scale database such as OuluVS2 we soon realize that we are short of data. In order to properly train an end-to-end system, we must find a balance between the amount of available training data and the number of parameters of the model. For example, the well known AlexNet for object classification contains 62 millions of parameters, which were trained from 14 million images ($\sim$22% of the numbers of parameters). In contrast, the number of parameters of our network is $\sim$15 millions, while the amount of available training samples is just 1,200 sequences. Thus, the ratio between parameters and training data exceeds 100 : 1, which makes it very difficult and time-consuming to train the network at once. Consequently, we propose to split the training by modules: the visual module ($\sim$12 million parameters) and the temporal module ($\sim$3 millions). It is quite intuitive to split the training in this way because each module has a specific goal which can be reached independently. The goal of the visual module is to parametrize the visual information observable at a given time instant or window. On the other hand, the temporal module aims to map the visual features into speech units while incorporating temporal constraints to ensure that the decoded message is coherent.

Therefore, we use intermediate labels to subdivide each training sentence into smaller units that can be used to train separately the visual module. This is useful because the number of training samples is increased, making the model easier to train and generalize. Moreover, this allows to control that the CNN learns to extract meaningful features sufficiently informative to encode the mouth appearance in a way that is helpful for the temporal module to predict the correct phrase.

To do so, we ideally need a dataset that provides very accurate speech labels, i.e. phonemes or visemes. Unfortunately, most of the lip-reading datasets, including OuluVS2, provide only the text that corresponds to each phrase but do not provide phoneme or viseme labels per frame. Furthermore, while there exist semi-automatic programs such as Praat [18] or Montreal

Forced Aligner [19] to align the text with the audio stream, they often require considerable manual intervention to refine the boundaries of each phoneme, resulting in a challenging and time-consuming process that does not scale well.

As a solution, based on the observation that the CNN only needs to distinguish among visually separable classes, we propose to rely on weak labels that, while imperfect, can be generated automatically in such a way that they are still informative about the mouth appearance. Hence, we hypothesize that, if the CNN is able to differentiate among these weak visual labels, the features generated at the last step of the visual module (those at the FC layers) will properly encode the mouth appearance and will be helpful for the temporal module to decode visual speech. Once the visual module is trained, its classification layer is removed and the output from the FC layers is fed to the temporal module, which can be trained for phrase classification in a straight-forward manner.

B. Visual units

We propose to automatically generate weak frame labels to train the visual module to classify visual units. We define a visual unit as a collection of visually similar images constrained by phonetics, which we obtain by minimizing an energy function. We define a function f: $\mathbb{Z} \Rightarrow \mathbb{Z}$ that maps a frame m into a visual unit $v \in [1, V]$, where V is the number of visual units. We define the energy function to minimize as:

$$\begin{aligned}\arg\min_{f} \sum_{s,v} \sum_{\substack{m \\ f(m)=v}} \sum_{\substack{n \neq m \\ f(n)=v}} \| I(m) - I(n) \|^2 + \\ +\lambda_1 \sum_{s,m} \sum_{\substack{n \neq m \\ |n-m| \leq W}} \overline{\delta}(f(m) - f(n))\, \mathcal{B}(m, n; I_m, I_n) + \\ +\lambda_2 \sum_{t} \sum_{m,n \in t} \overline{\delta}(f(m) - f(n)) + \lambda_3 \sum_{s,u} | T_u - P_u |\end{aligned} \tag{1}$$

The first term from eq. (1) derives directly from our definition of visual units, which shall be groups of frames with similar appearance. However, it is evident that inter-subject differences should not affect the resulting grouping. Hence, we penalize that two frames m and n of the same subject s get assigned to the same visual unit v if there are large intensity differences between them. The second term controls temporal coherence. It is assumed that neighboring frames (with a maximum distance W) should correspond to the same visual unit v, unless they have a large appearance difference. Thus, we enforce that frames m and n that are temporally close and have similar appearance are assigned the same visual unit. We do so by penalizing the function $\overline{\delta}(f(m) - f(n)) = 1 - \delta(f(m) - f(n))$ (where $\delta()$ is the Kronecker delta) weighted by a temporal bilateral filter $\mathcal{B}$ that depends on both the appearance difference and the temporal proximity, defined as:

$$\mathcal{B}(m, n; I_m, I_n) = \phi_1(\| I_n - I_m \|)\phi_2(\| n - m \|) \tag{2}$$

where ϕ_1 and ϕ_2 are Gaussian kernels. These temporal constraints induce temporal segments t of frames labeled with the same visual unit until an appearance transition (Fig.2-*Left*).

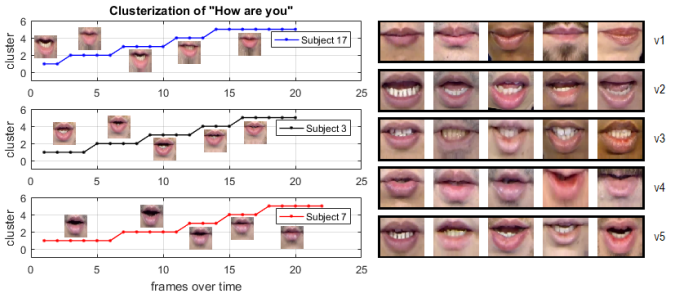

Fig. 2. Left: automatic visual units labeling of the phrase "How are you" for subjects 17, 3 and 7; for every segment we show the mean image over the segment. Right: examples of frames within 5 of the resulting visual units.

TABLE I
NUMBER OF EXPECTED TIME-SEGMENTS PER PHRASE

Phrase	*P*	Phrase	*P*
Excuse me	5	See you	3
Goodbye	4	I am sorry	5
Hello	3	Thank you	3
How are you	5	Have a good time	5
Nice to meet you	5	You are welcome	5

The third term controls speech consistency, which is adapted to the structure of OuluVS2, where all speakers are uttering the same phrases. Consider two recordings of the same phrase; we would expect that the visual units labeling their temporal segments are in correspondences. Thus, we assume that frames m and n within a time-segment t in any utterance of the same phrase belong to the same visual unit v. Finally, the last term is a regularization term that limits the number of segments T per utterance considering phonetics. We set the expected number of time-segments per sentence P using the phoneme-to-viseme mapping proposed by Jeffers and Barley [20], as it has been one of the most widely used mappings for English [21]–[24]. In Table I we show the number of time-segments per sentence.

Once the final set of visual units has been established, the visual module can be trained to predict them. Afterwards, the classification layer of the visual module is removed, all their weights are frozen and the LSTMs are connected after them to perform phrase classification.

IV. EXPERIMENTAL SETTINGS

A. Database

The OuluVS2 database [5] contains multi-view video recordings from 52 speakers uttering continuous digit sequences, short phrases and TIMIT sentences. We used the frontal-views of the second session where subjects were asked to read 10 daily-use English phrases. We tested our system in a speaker-independent setting. Following the testing procedure proposed by the database creators, we used 12 subjects for testing (s6, s8, s9, s15, s26, s30, s34, s43, s44, s49, s51 and s52) and 40 subjects for training and validation. Thus, we had 360 videos for testing (12 subj $\times$ 10 phrases $\times$ 3 repetitions), 1020 for training (40 subj $\times$ 10 phrases $\times$ 3 repetitions $\times$

0.85) and 180 for validation. We provide our results in terms of phrase-level classification (the standard for this dataset).

B. Data pre-processing

We used the cropped and aligned mouth regions provided by OuluVS2. Taking into account that our visual module is based on VGG-M, we normalized all sequences to 1/4 of the original VGG-M size (224x224) per axis, yielding a fixed size of 56x56 pixels. The motivation to do so is twofold; 1) to simplify the network by reducing the number of parameters; 2) because VGGs have been used with whole-face images, hence it seems reasonable to reduce the region considering the mouth size with respect to the whole face size.

C. Weak labeling in terms of visual units

The minimization of eq. (1) was done following an iterative process. Specifically, we start by splitting each phrase in P time-segments (Table I), which should theoretically be visually separable. These initial units will be speaker independent, i.e when a phrase is split into P segments, we expect to find units highly correlated with the spoken sentence but not with the speaker identity (i.e. similarly to visemes). Then, we start by clustering the frames of each phrase separately for each subject and then merge those units that are shared among different phrases from the same subject. Specifically, two visual units of different phrases will be fused into a single one, if and only if the distance between their centroids is smaller than the average minimum distance between visual units within the same phrase, and the above condition is met by at least 50% of the subjects in the training set.

The first condition sets a relative threshold to determine when two different segments should actually be considered instances from the same class. The second condition, ensures that such similarity is sufficiently consistent across subjects to be considered generic and not subject-specific. At the end of this merging process, we end up with a separate set of visual units for each subject. However, all sets have the same number of visual units and they are in correspondence across subjects. Thus, we can simply fuse those sets into a common one, that contains the contributions from all subjects and can be used to train our visual module in a subject-independent manner.

D. Architecture details

Both modules are trained using stochastic gradient descent (SGD) with a momentum of 0.2, mini-batches of size 32 for the visual module and 1 for the temporal module, and learning rate 0.01. The classifiers are a softmax that uses cross-entropy loss to classify among 13 visual units (visual module) or among 10 phrases (temporal module). In both cases, DA was necessary to deal with the short-scale training set. DA consisted of horizontal flips, rotation of a maximum of 10 degrees, width/height shifts and zooming up to 5% of the image resolution. To deal with over-fitting, we applied the following regularization methods to the visual module: batch normalization, dropout and L2-regularization. Batch normalization and dropout of 0.5 were performed between several convolutional layers (as indicated in Fig. 1). In contrast, L2-regularization with 0.1 weight was applied to fully connected layers to penalize highly positive or negative weights. Experiments were performed in a computer with an Intel Core i7-7700 processor (3.6 GHz), 16 GB RAM, and a single NVIDIA GeForce GTX 1060 graphic processing unit with 6 GB on-board graphics memory. The proposed model was implemented using the Keras framework with a Tensorflow backend. The total training time for our system was around 3.5 hours ($\sim$ 2 hours for the visual module), without requiring any pre-trained model nor additional training data.

V. Results

A. LDNet Results

As explained in Section III, instead of training the system fully end-to-end, we split the training by modules:

1) Visual module: We trained the visual module to classify among visually distinguishable units, which were determined by minimizing eq. (1) and resulted in 13 visual units. Fig.2-*Right* shows a few examples from 5 resulting visual units.

The CR obtained by the CNN module was 47.67%. While at first glance these results may seem modest, we will see that the features learned in this way by the CNNs are useful enough for the temporal module to produce high phrase recognition rates. Moreover, keeping in mind that our visual units are based on a similar definition to the one commonly used for visemes, our results are not far from those reported for phoneme and viseme classification in ALR [25]–[28].

2) Temporal module: The temporal module shows the performance of the whole system because it outputs the spoken phrase. Following the procedure from [12], [15] we obtained an average CR of 91.38% ($\pm$ 0.61% standard deviation) averaged over 10 runs of temporal module training.

B. Comparison to other ALR systems

In this section we compare the DNN architectures evaluated in the OuluVS2 database (Table II). Among systems using external training data, we firstly find the three systems proposed by Saitoh et al. [1]. Those systems used pre-trained models that were trained in external databases not related to lip-reading and were fine-tuned for OuluVS2. The GoogLeNet model achieved the maximum performance of 85.60% CR. Similarly, Chung and Zisserman [2], [9] proposed two systems specifically trained for lip-reading but pre-trained on much larger databases (LRW and LRS), and later fine-tuned for OuluVS2, achieving a maximum of 94.10% CR in [2].

There are several systems that do not use external data to train their model [7], [12]–[15]. Among them, the most direct comparison to our system are those based on similar architectures, combining CNNs and LSTMs [7], [12]. The main difference between those systems and ours is the training process. In the case of Lee et al. [7], they directly trained their ALR system end-to-end from scratch, achieving a rather low performance of 81.10% CR. More recently, Fung and Muk [12] proposed a training strategy based on a big DA and on adding maxout activation units for ensuring a better

TABLE II
COMPARISON WITH PREVIOUS WORK ON THE OULUVS2 DATABASE.

With pre-trained models		Without pre-trained models	
Architecture	**CR (%)**	**Architecture**	**CR (%)**
CFI+NIN [1]	81.10	CNN+LSTM [7]	81.10
CFI+AlexNet [1]	82.80	Encoder+LSTM [13]	84.50
CFI+GoogLeNet [1]	85.60	Encoder+BiLSTM [14]	91.80
VGG-M+LSTM [2]	31.90	Encoder+BiLSTM [15]	91.80
SyncNet+LSTM [2]	94.10	CNN+BiLSTM [12]	87.60
CNN+LSTM+Att. [9]	91.10	LDNet (Ours)	91.38

training. They achieved a higher accuracy (87.60% CR) with a system that combines 3D-CNNs with BiLSTMs. In contrast, in LDNet we follow a CNN-LSTM baseline, but propose an alternative training process. Specifically, we train the visual module separately to classify weakly labeled visual units, which are directly related with the spoken phrases. This has proven to be beneficial because it allows to increase the training samples while ensuring that the learned features are directly related to speech and not to other aspects such as speaker appearance. In this way, when the temporal module is added after the visual module, our system is able to achieve an average CR of 91.38%, which is quite competitive even with respect to systems using pre-trained models.

A different direction has been explored by Petridis et al. [13]–[15], who presented 3 systems based on an encoding network combined with BiLSTMs. Even though these systems do not follow the current trend in ALR, they reported 91.80% CR, which are state-of-the-art comparable results. However, analyzing these ALR systems we find that they were not trained end-to-end from scratch; instead, they pre-trained the encoding layer in a greedy layer-wise manner using Restricted Boltzmann Machines. They initialized their systems with the pre-trained encoder and trained the BiLSTMs while fine-tuning the encoder parameters. Compared to these 3 systems, LDNet provides a very similar accuracy, with low training time and maintaining a main-stream end-to-end ALR architecture, which is likely to benefit from the latest advances in the field, currently based on CNN-RNN architectures [6].

VI. CONCLUSIONS

In this work, we investigate the design of an end-to-end ALR system that is simple to train without the need of large-scale databases. The design of end-to-end architectures with limited training data is challenging because there are not enough samples to successfully train DNNs that produce top performance. Thus, we propose to weakly label the data in an automatic manner in terms of visual units, which are representative enough to disambiguate among different phrases when context information is also provided. Specifically, we introduce an ALR system that performs phrase-level classification combining a visual module based on CNNs and a temporal module based on LSTMs. We show that, thanks to the weak intermediate labels, it is feasible to obtain state-of-the-art performance by splitting the training by modules. We evaluated our system in the well-known OuluVS2 and reported a CR of 91.38% which is comparable to state-of-the-art results. Differently from previous approaches, our system does not require the use of any pre-trained model or external training data. LDNet training was completed in approximately 3.5 hours in a desktop computer with standard GPU hardware.

ACKNOWLEDGMENTS

This work is partly supported by the Spanish Ministry of Economy and Competitiveness under project grant TIN2017-90124-P, the Ramon y Cajal programme, and the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

REFERENCES

[1] T. Saitoh *et al.*, "Concatenated frame image based CNN for visual speech recognition," in *Proc. ACCV*, 2016, pp. 277–289.
[2] J. S. Chung and A. Zisserman, "Out of time: automated lip sync in the wild," in *Proc. ACCV*, 2016, pp. 251–263.
[3] T. Stafylakis and G. Tzimiropoulos, "Combining residual networks with LSTMs for lipreading," in *Pro. of Interspeech*, 2017, pp. 3652–3656.
[4] S. Petridis *et al.*, "End-to-end audiovisual speech recognition," in *Proc. ICASSP*, 2018.
[5] I. Anina *et al.*, "OuluVS2: A multi-view audiovisual database for non-rigid mouth motion analysis," in *Proc. FG*, vol. 1, 2015, pp. 1–5.
[6] A. Fernandez-Lopez and F. Sukno, "Survey on automatic lip-reading in the era of deep learning," *Image Vision Comput*, vol. 78, pp. 53–72, 2018.
[7] D. Lee *et al.*, "Multi-view automatic lip-reading using neural network," in *Proc. ACCV*, 2016, pp. 290–302.
[8] Y. M. Assael *et al.*, "Lipnet: Sentence-level lipreading," in *Proc. GTC*, 2017.
[9] J. S. Chung and A. Zisserman, "Lip reading in profile," in *Proc. BMVC*, 2017.
[10] A. Krizhevsky *et al.*, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.
[11] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Proc. ACCV*, 2016, pp. 87–103.
[12] H. L. Fung and B. Mak, "End-to-end low-resource lip-reading with maxout CNN and LSTM," in *Proc. ICASSP*, 2018.
[13] S. Petridis *et al.*, "End-to-end visual speech recognition with LSTMs," in *Proc. ICASSP*, 2017, pp. 2592–2596.
[14] ——, "End-to-end audiovisual fusion with LSTMs," in *Proc. AVSP*, 2017.
[15] ——, "End-to-end multi-view lipreading," in *Proc. BMVC*, 2017.
[16] K. Chatfield *et al.*, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. BMVC*, 2014.
[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.
[18] P. Boersma *et al.*, "Praat, a system for doing phonetics by computer," *Glot international*, vol. 5, no. 9/10, pp. 341–345, 2002.
[19] M. McAuliffe *et al.*, "Montreal forced aligner: trainable text-speech alignment using kaldi," in *Proc. of interspeech*, 2017, pp. 498–502.
[20] J. Jeffers and M. Barley, *Speechreading (lipreading)*. Thomas, 1971.
[21] L. Cappelletta and N. Harte, "Viseme definitions comparison for visual-only speech recognition," in *Proc. EUSIPCO*, 2011, pp. 2109–2113.
[22] N. Harte and E. Gillen, "TCD-TIMIT: An audio-visual corpus of continuous speech," *IEEE Trans. Multimed.*, vol. 17, no. 5, pp. 603–615, 2015.
[23] H. L. Bear and R. Harvey, "Decoding visemes: improving machine lip-reading," in *Proc. ICASSP*, 2016, pp. 2009–2013.
[24] H. L. Bear *et al.*, "Which phoneme-to-viseme maps best improve visual-only computer lip-reading?" in *Proc. ISVC*, 2014, pp. 230–239.
[25] A. Fernandez-Lopez and F. M. Sukno, "Automatic viseme vocabulary construction to enhance continuous lip-reading," *Proc. VISAPP*, vol. 5, pp. 52–63, 2017.
[26] K. Thangthai *et al.*, "Comparing phonemes and visemes with DNN-based lipreading," in *Proc. BMVC*, 2017.
[27] H. L. Bear and R. Harvey, "Phoneme-to-viseme mappings: the good, the bad, and the ugly," *Speech Communication*, vol. 95, pp. 40–67, 2017.
[28] Y. Lan *et al.*, "Insights into machine lip reading," in *Proc. ICASSP*, 2012, pp. 4825–4828.