*INTERSPEECH 2019*
September 15–19, 2019, Graz, Austria

# Speaker Adaptation for Lip-reading Using Visual Identity Vectors

*Pujitha Appan Kandala*, Abhinav Thanda*, Dilip Kumar Margam,*
*Rohith Chandrashekar Aralikatti, Tanay Sharma, Sharad Roy, Shankar M Venkatesan*

Samsung R&D Institute India, Bangalore

{pujitha.k, abhinav.t89, dilip.margam}@samsung.com,
{r.aralikatti, tanay.sharma, sharad.roy, s.venkatesan}@samsung.com

## Abstract

Visual speech recognition or lip-reading suffers from high word error rate (WER) as lip-reading is based solely on articulators that are visible to the camera. Recent works mitigated this problem using complex architectures of deep neural networks. I-vector based speaker adaptation is a well known technique in ASR systems used to reduce WER on unseen speakers. In this work, we explore speaker adaptation of lip-reading models using latent identity vectors (visual i-vectors) obtained by factor analysis on visual features. In order to estimate the visual i-vectors, we employ two ways to collect sufficient statistics: first using GMM based universal background model (UBM) and second using RNN-HMM based UBM. The speaker-specific visual i-vector is given as an additional input to the hidden layers of the lip-reading model during train and test phases. On GRID corpus, use of visual i-vectors results in 15% and 10% relative improvements over current state of the art lip-reading architectures on unseen speakers using RNN-HMM and GMM based methods respectively. Furthermore, we explore the variation of WER with dimension of visual i-vectors, and with the amount of unseen speaker data required for visual i-vector estimation. We also report the results on Korean visual corpus that we created.

**Index Terms**: lip-reading, visual i-vectors, factor analysis, speaker adaptation

## 1. Introduction

Pure visual speech recognition or lip-reading involves the recognition of text from a speaker's lip movements. In general, both human lip-reading as well as machine lip-reading suffer from high error rates. This is expected because lip-reading is based solely on the visible articulators, such as the lips and to some extent the tongue and teeth. Using deep neural networks (DNN), recent works on machine lip-reading [1–6] have demonstrated considerable improvements in word error rate (WER) compared to human counterparts, especially on limited vocabulary datasets like GRID corpus. In this work, we further improve the lip-reading performance on unseen test speakers using speaker adaptation.

In automatic speech recognition (ASR) systems, speaker adaptation serves an important purpose in reducing WER of unseen test speakers. This is done in two primary ways: model adaptation and feature adaptation. In model adaptation, a generic speaker-independent (SI) model is first trained on a larger speech corpus. Then using a limited amount of unseen speaker data, the SI model is transformed to perform better on a particular test speaker. In contrast, feature adaptation involves a feature transformation to make the test speakers perform better on the SI model [7].

Analogous to ASR, the problem of high WER in lip-reading can be mitigated to some extent by means of speaker adaptation. Given a sufficient amount of visual data of an unseen speaker, we can adapt the network to the patterns of lip movements specific to that unseen test speaker. The primary aim of this work is to demonstrate the usefulness of speaker adaptation in lip-reading.

Our work is an extension of [8] to the context of lip-reading. The method described in [8] involves addition of i-vector as an input to the ASR network, along with regular speech feature vectors of every speech frame. An i-vector embodies the characteristics of that speaker. Due to the presence of additional speaker specific input, the network learns both speaker-specific and phoneme-specific variations at the same time [8]. Accordingly, in lip-reading one can expect that similar visual identity vectors obtained from sequences of images of speaker's lips will help in adapting lip-reading model to that particular speaker. While learning to read lips, the model also learns the variations in lip shapes induced by that particular speaker's manner of speaking.

I-vectors are speaker specific latent variables or factors obtained by factor analysis (FA). FA methods are used in state-of-the-art speaker verification systems [9–11]. FA also finds applications in face recognition [12, 13]. FA can be understood as a generative model that maps from speaker's latent identity space to the observed feature space [12].

Similar to [2] our network comprises of spatio-temporal convolutional network followed by recurrent layers. The network is trained end-to-end using Connectionist Temporal Classification (CTC) cost function [14]. However, unlike [2], we train our model using word-CTC instead of character-CTC. Recent work [15] showed that on GRID corpus word-CTC performs better than character-CTC. In addition, we add the speaker's visual i-vector to the recurrent layer during training and test time.

To collect the sufficient statistics (SS) required to estimate speaker-specific visual i-vectors we use two methods. The first method is based on the standard GMM-UBM [16] (UBM$_{GMM}$) trained on visual features obtained from speaker's lip images. In the second method we follow [17]. Specifially we train an RNN-HMM model on visual features. The SS are collected from the posteriors of the output of RNN-HMM model. We use two posterior probabilities: one of the tied-triphone states (UBM$_{TS}$) and other of the monophones (UBM$_{PH}$). Our results indicate that visual i-vectors estimated from UBM$_{PH}$ perform slightly better than those estimated from UBM$_{GMM}$ or UBM$_{TS}$.

There are several reasons for the choice of i-vector method for lip-reading. Adaptation of lip-reading model by retraining the network over new speaker data can result in over-fitting to the particular speaker and consequently, loss of generalization

* Both the authors contributed equally to this work.

http://dx.doi.org/10.21437/Interspeech.2019-3237

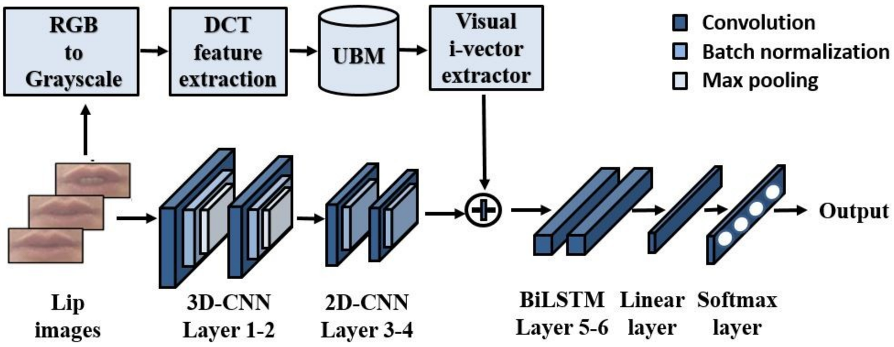

Figure 1: *Proposed method of speaker adaptation using visual i-vectors. The UBM can be GMM-UBM trained on DCT features or an ancillary UBM obtained from the sufficient statistics collected using RNN-HMM model trained on DCT features.*

of the network. In addition, unlike other speaker adaptation methods which involve an additional step of feature transformation at run time, in i-vector all of the speaker specific information is incorporated into a single vector which can be extracted off line.

The paper is organized as follows: section 2 provides the related work. In section 3, we describe the visual i-vector extraction procedure. In section 3, we also describe the lip-reading architecture and training with visual i-vectors. In section 4, we describe the experiments and results. Finally, we describe our future work in section 5.

## 2. Related work

### 2.1. Lip-reading

[1, 2, 5] are purely lip-reading models based on deep neural networks. [1] employs word level classification and achieves a word level accuracy of 79.6%. [2, 5] use spatiotemporal convolutions, a recurrent network, and the CTC loss, trained entirely end-to-end. On GRID audio-visual corpus [2] achieves a WER of 4.8% on seen speakers and 11.4% on unseen speakers. [5] uses cascaded attention CTC decoder and achieves WER of 2.9% on seen speakers.

Recent work on optimal architectures for lip-reading [15] has shown that when trained with CTC loss function, word level CTC performs better in comparison with character level CTC used in [2]. On unseen speakers on GRID corpus, [15] claims 2.8% absolute improvement in WER compared to [2]. Therefore, we consider [15] as our baseline and we apply visual i-vector adaptation technique over the architecture.

[3, 4, 6] focus on audio-visual ASR using deep models. They consider three different types of features as input to the ASR model: only audio features, only visual features and fusion of the two. In general, the performance of an ASR model degrades with the increase in noise level when audio features are employed. The results in [3, 4, 6] show that the above problem can be mitigated by the presence of visual modality as it gave a significant improvement in the system performance even at various noise levels. [4] shows that the above result holds true even when model is not trained with noise.

### 2.2. Speaker adaptation

Among GMM-HMM ASR systems, well known adaptation methods can be categorized as model adaptation methods and feature-space adaptation methods. Model adaptation involves transformation of the GMM-HMM model parameters methods usually using maximum a posteriori (MAP) [18] or maximum likelihood linear regression (MLLR) [19] estimation. The most commonly used feature space adaptation methods are constrained MLLR [20] and vocal tract length normalization (VTLN) [21]. Whereas MAP and MLLR are not ideal methods for deep models [8, 22] in general, constrained MLLR even though requires very little amount of adaptation data, is not suitable for lip-reading models where inputs are 2D images.

In the context of connectionist ASR models, [23] explores speaker adaptation from the perspective of feature adaptation. [24–28] discuss various techniques of model adaptation. These methods include the application of affine transforms at the input layer [23, 25, 26], at the output layer [26] or hidden layer [29]. The additional transformation parameters are trained on the adaptation data. Another set of methods include modifying the original network parameters. However, this may result in loss of generalization of the network. This problem is addressed in [24]. The authors re-train the entire network but with a KL Divergence regularization term for proper generalization of the adapted model. In [27] the author discusses adaptation by retraining portions of the network.

A third set of methods relevant to this work involves the addition of a speaker specific code as input to one or more layers of network [8, 30] in addition to the acoustic features. In [30] the user code is learnt simultaneously with the network parameters. During adaptation, only the speaker's code is updated using back-propagation algorithm. [8] instead uses the speaker's i-vector as input.

## 3. Method

### 3.1. Lip-reading model

We follow the architecture given in [15] which consists of two 3D CNN layers followed by two 2D CNN layers. The CNN layers are followed by two layers of BiLSTM as shown in Figure 1. The network is trained with 100x50x3 RGB image sequences of lips. The network is trained using word-level CTC as described in [15]. In this work, the speaker-specific visual i-vector is provided as additional input to the BiLSTM hidden layer as shown in Figure 1.

### 3.2. UBM-based visual i-vector estimation

**Training a UBM model:** Given a set of training data containing images of lips of S speakers, we extract 2D-DCT features denoted by $F(s) = (f_1(s), f_2(s), ..., f_{R(s)}(s))$ ,$f \in \mathbb{R}^N$, N being the feature dimension. R(s) is the number of feature vectors corresponding to a specific speaker, s and F(s) represents the sequence of all frames of all utterances spoken by a particular speaker. The UBM is a gaussian mixture model (GMM) with C mixture components which is trained on the data of all speakers in the training set. The UBM parameters, weight, mean and variance are denoted by $\phi$, $\mu$ and $\Sigma$. The density function of a specific feature vector $f_t$, given the UBM model parameters can be represented as,

$$p(f_t/\mu, \Sigma) = \sum_{c=1}^{C} \phi_c \mathcal{N}(f_t; \mu_c, \Sigma_c) \tag{1}$$

where $\mathcal{N}$ represents the normal distribution, $\phi_c$, $\mu_c$ and $\Sigma_c$ represents the weight, mean and variance of a particular gaussian component respectively.

**Defining visual i-vector model:** In the visual i-vector model, the speaker-specific mean sub-vector $\mu_c(s)$ can be modelled as,

$$\mu_c(s) = \mu_c + T_c w(s) \tag{2}$$

here, $w(s)$ is the speaker-adapted vector (visual i-vector) of dimension $N \times 1$ and $T_c$ is the submatrix with dimension $D \times N$ of factor loading matrix T.

The visual i-vector is the maximum a posteriori (MAP) estimation of w(s).

$$\hat{w}(s) = argmax(p(w(s)/F(s))) \tag{3}$$

The Baum-Welch statistics for a given sequence of feature vectors for a speaker s, are obtained from the posterior probabilities of the UBM. In the estimation step, the posterior expectation of the i-vector is obtained using sufficient statistics,

$$E[w(s)] = l^{-1}(s)T^T\Sigma^{-1}\theta_f(s) \tag{4}$$

Where, $l(s) = I + T^T\theta(s)\Sigma^{-1}T$,
$\theta_f(s) = \sum_{t=1}^{R(s)} \sum_{c=1}^{C} p(c/f_t(s), \mu_0, \Sigma_0)(f_t(s) - \mu_c(s))$ and, $\mu_0, \Sigma_0$ are intial model parameters.

The estimation of T and $\Sigma$ are derived using the posterior estimations and are updated in the maximization step using equations similar to (34) and (35) as in [16]. For detailed explanation refer [16].

### 3.3. RNN-HMM-based visual i-vector estimation

In [17] the authors show that a GMM-UBM model (UBM$_{GMM}$) can be substituted by an ancillary UBM estimated using posteriors from a DNN-HMM model. This method provides better SS as the DNN-HMM models are “phonetically aware” [31]. In the context of our paper, phonetic awareness corresponds to different visual articulators i.e. lip shapes. Thus, each output class in DNN can be considered as a represention for a particular shape of lip.

In our work, we use RNN-HMM model instead of DNN. We compare two ancillary UBMs each initialized with different types of posterior probabilities. The first type (UBM$_{TS}$) corresponds to posteriors of triphone states obtained from soft-max layer of RNN-HMM. The second type (UBM$_{PH}$) corresponds to phoneme posteriors obtained by mapping the state-posteriors from RNN-HMM to monophones.

Thus, the number of components in UBM$_{TS}$ is equal to the number of output classes of the RNN-HMM model. And in UBM$_{PH}$ the number of components in the ancillary UBM is equal to the number of monophones.

Each component in both the ancillary UBMs can be considered to be modelling a particular lip shape.The generative process in factor analysis using the ancillary UBM can be understood as follows: a point in the latent identity space corresponding to a particular speaker, is transformed into a UBM super-vector of mean vectors, through a linear transformation followed by the addition of some noise.

Further, the visual i-vector extraction process is similar to the UBM$_{GMM}$ model. In [31], once the ancillary UBM is initialized, the UBM parameters are not updated using EM algorithm. However, unlike [31] we update the parameters for 10 iterations.

## 4. Experiments and results

**Corpus:** Our experiments are performed on GRID audio-visual corpus [32]. GRID is an openly available corpus containing audio-visual data from 34 speakers with 1000 utterances per speaker. In the corpus available on the web, visual data for speaker number 21 is absent. For accurate comparison with results presented in [2, 15] we choose the same set of speakers for training and testing. For testing, we use the data of two male speakers (speaker-(1,2)) and two female speakers (speaker-(20,22)) and the remaining 29 speakers for training.

To further establish the validity of our approach, we also evaluate our method on a second corpus which is a Korean language audio-visual corpus locally created by us. The corpus consists of ten Korean language sentences each spoken ten times by 66 speakers. Out of the 66 speakers we use data of 58 speakers for training and use the remaining for testing.

**Visual i-vector estimation:** For each image in the data, we use YOLO [33] CNN architecture to detect the ROI, i.e. speaker's lip region. The ROI is cropped and resized to 100x50x3 RGB image and converted to grayscale. 10x10 2D-DCT features are extracted from above and vectorized to 100 dimensional feature vector. Following [15] we perform feature duplication by making 4 copies of each feature vector. It is important to note that mean and variance normalization is applied at speaker level to preserve speaker specific information. For visual i-vector estimation we use Kaldi speech recognition toolkit [34].

**GMM-UBM based method:** We train a diagonal UBM model with 1024 Gaussian mixture components. In contrast to standard literature, we observed that mean and length normalization of visual i-vectors deteriorated the performance.

**RNN-HMM based method:** The RNN-HMM model has 2 BiLSTM hidden layers which consist of 400 (first layer) and 640 (second layer) hidden units respectively. The cell dimension was set to 320. The output soft-max layer dimension was 1664 corresponding to the HMM pdfs obtained from UBM$_{TS}$ model. The 1664 pdfs correspond to 3726 HMM transitions which is used as the number of components in UBM$_{TS}$. UBM$_{PH}$ consists of 36 components corresponding to 36 monophones.

Table 1: *% WER comparison over different models used for visual i-vector extraction on GRID and Korean corpus. The visual i-vector dimension is 10 in all scenarios.*

| **Model** | **UBM** | **GRID (%WER)** | **Korean (%WER)** |
|---|---|---|---|
| LipNet | NA | 11.4 | - |
| Baseline | NA | 8.6 | 4.60 |
| Baseline + i-vector | $\text{UBM}_{GMM}$ | 7.7 | 4.40 |
| Baseline + i-vector | $\text{UBM}_{TS}$ | 7.6 | 4.34 |
| Baseline + i-vector | $\text{UBM}_{PH}$ | 7.3 | 4.20 |

Table 2: *%WER comparison for various dimensions of visual i-vectors extracted using $\text{UBM}_{GMM}$ model on GRID corpus.*

| **Model** | **Visual i-vector (dimension)** | **%WER** |
|---|---|---|
| Baseline | NA | 8.6 |
| Baseline + i-vector | 10 | 7.7 |
| Baseline + i-vector | 20 | 8.0 |
| Baseline + i-vector | 40 | 8.6 |
| Baseline + i-vector | 100 | 9.2 |

Table 3: *%WER comparison for various amount of unseen speaker data used for extracting visual i-vectors using $\text{UBM}_{GMM}$ model on GRID corpus.*

| **Model** | **Adaptation Data (mins)** | **%WER** |
|---|---|---|
| Baseline | NA | 8.6 |
| Baseline + i-vector | 5 | 8.3 |
| Baseline + i-vector | 10 | 7.8 |
| Baseline + i-vector | 25 | 7.7 |
| Baseline + i-vector | 50 | 7.6 |

**Lip-reading model (baseline):** The 100x50x3 RGB image sequence is input to the network. The architecture of the base-line [15] is as follows. The network consists of two 3D-CNN layers with (kernal (k)/ stride (s)/ pad (p)) parameters as (3x5x5/ 1,2,2/ 1,2,2) for first layer and (4x5x5/ 1,1,1/ 1,2,2) for second layer. Each convolution layer is followed by batch-normalization and a max-pooling layer with (k/s)-(1x2x2/ 1,2,2). The two 2D-CNN layers in Figure 1 have (k/s/p)-(5x5/ 2,2/ 2,2) for first layer and (3x3/ 2,2/ 2,2) for second layer. We perform batch-normalization after each 2D-CNN layer. BiLSTM layers each have 200 hidden units. The output soft-max layer is of size 53, including 51 words corresponding to the vocabulary of GRID corpus and additional symbols for space and blank labels.

As described in previous section, we input the visual i-vector at the recurrent layer. Following [15] we apply curriculum learning and train with word-CTC loss for 70 epochs with learning rate of 0.0001 and batches of size 32. Similar to [15], randomly drop or repeat the frames with probability 0.05. We use Tensorflow system [35] for training the above model.

### 4.1. Results:

Recent state-of-the-art lip-reading models are discussed in [2, 5, 6, 15]. Our main focus in this work is the reduction of WER on unseen test speakers because speaker adaptation is relevant to unseen test speaker case. Since the authors in LCANet [5] and in [6] do not consider the performance under unseen speaker scenario, we provide a comparison with [2, 15]. As mentioned in section 2 we consider [15] as our baseline because it performs better than LipNet [2] under unseen speaker case. However, in principle the visual i-vector method is independent of the architecture of lip-reading model and provides improvement on any model. As shown in Table 1, the result on GRID over unseen test speaker set, indicates that through the addition of UBM-based visual i-vector, we get a 10.4% relative improvement in WER over the baseline. Furthermore, addition of visual i-vectors derived from $\text{UBM}_{TS}$/$\text{UBM}_{PH}$ model gave an improvement of 11.6%/15% over baseline. We also obtained a performance improvement of 5.6%/8.7% using visual i-vectors derived from $\text{UBM}_{TS}$/$\text{UBM}_{PH}$ relative to the baseline system for Korean corpus.

**Dimensionality of visual i-vector:** The choice of appropriate dimension of visual i-vectors is empirically determined. As given in Table 1, we observe that as the dimensionality of visual i-vector increases, the performance improvement drops with optimal value being 10.

**Adaptation Data:** The amount of unseen speaker data required for extracting the optimal visual i-vector is as shown in Table 3. For GRID corpus around 25 minutes of new speaker data is required to get optimal performance improvements.

## 5. Future Work

In the current work, we demonstrated the significance of visual i-vectors for speaker adaptation in end-to-end lip-reading models, to improve WER on unseen speakers. To the best of our knowledge this is the first work to address the given problem. In our future work we intend to explore the visual i-vector method further by separating lip shape and texture. In addition, there is the challenge of adapting to varying facial features of the same person. For example, variations due to facial hair and make up. In this work, we assumed that the speaker pose is constant (frontal pose) over all utterances. In the future work, we will consider variation of visual i-vectors with different poses. A natural extension of our work would be to use factor analysis method for adaptation of audio-visual speech recognition models. In our future works, we intend to explore these avenues.

## 6. References

[1] M. Wand, J. Koutník, and J. Schmidhuber, "Lipreading with long short-term memory," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6115–6119.

[2] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, "Lipnet: End-to-end sentence-level lipreading," *arXiv preprint arXiv:1611.01599*, 2016.

[3] A. Thanda and S. M. Venkatesan, "Multi-task learning of deep neural networks for audio visual automatic speech recognition," *arXiv preprint arXiv:1701.02477*, 2017.

[4] A. Thanda and S. Venkatesan, "Audio visual speech recognition using deep recurrent neural networks," in *IAPR Workshop on Multimodal Pattern Recognition of Social Signals in Human-Computer Interaction*. Springer, 2016, pp. 98–109.

[5] K. Xu, D. Li, N. Cassimatis, and X. Wang, "Lcanet: End-to-end lipreading with cascaded attention-ctc," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG)*. IEEE, 2018, pp. 548–555.

[6] J. S. Chung, A. W. Senior, O. Vinyals, and A. Zisserman, "Lip reading sentences in the wild." in *CVPR*, 2017, pp. 3444–3453.

[7] Y. Miao, H. Zhang, and F. Metze, "Speaker adaptive training of deep neural network acoustic models using i-vectors," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 11, pp. 1938–1949, 2015.

[8] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors." in *ASRU*, 2013, pp. 55–59.

[9] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE transactions on speech and audio processing*, vol. 13, no. 3, pp. 345–354, 2005.

[10] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," *CRIM, Montreal,(Report) CRIM-06/08-13*, vol. 14, pp. 28–29, 2005.

[11] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[12] S. J. Prince and J. H. Elder, "Tied factor analysis for face recognition across large pose changes." in *BMVC*. Citeseer, 2006, pp. 889–898.

[13] D. Gong, Z. Li, D. Lin, J. Liu, and X. Tang, "Hidden factor analysis for age invariant face recognition," in *Proceedings of the ieee international conference on computer vision*, 2013, pp. 2872–2879.

[14] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[15] M. Dilip Kumar, A. Rohith, S. Tanay, T. Abhinav, A. K. Pujitha, R. Sharad, and V. Shankar M, "Lipreading with 3d-2d-cnn blstm-hmm and word-ctc models," *arXiv preprint arXiv:1906:12170*, 2019.

[16] Y. Zhang, "Useful derivations for i-vector based approach to data clustering in speech recognition," 2011.

[17] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition," in *Proc. Odyssey*, 2014, pp. 293–298.

[18] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE transactions on speech and audio processing*, vol. 2, no. 2, pp. 291–298, 1994.

[19] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer speech & language*, vol. 9, no. 2, pp. 171–185, 1995.

[20] J. Ganitkevitch, "Speaker adaptation using maximum likelihood linear regression," in *Rheinish-Westflesche Technische Hochschule Aachen, the course of Automatic Speech Recognition, www-i6. informatik. rwthaachen. de/web/Teaching/Seminars/SS05/ASR/Juri Ganitkevitch Ausarbeitung.pdf*. Citeseer, 2005.

[21] P. Zhan and A. Waibel, "Vocal tract length normalization for lvcsr," *Tech. Rep. CMU-LTI-97-150*, 1997.

[22] D. Wang and T. F. Zheng, "Transfer learning for speech and language processing," *arXiv preprint arXiv:1511.06066*, 2015.

[23] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *IEEE Workshop onAutomatic Speech Recognition and Understanding (ASRU)*, 2011, pp. 24–29.

[24] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 7893–7897.

[25] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker-adaptation for hybrid hmm-ann continuous speech recognition system," in *Fourth European Conference on Speech Communication and Technology*, 1995.

[26] B. Li and K. C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid nn/hmm systems," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[27] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*. IEEE, 2013, pp. 7947–7951.

[28] R. Doddipatla, M. Hasan, and T. Hain, "Speaker dependent bottleneck layer training for speaker adaptation in automatic speech recognition," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[29] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. De Mori, "Linear hidden transformations for adaptation of hybrid ann/hmm models," *Speech Communication*, vol. 49, no. 10-11, pp. 827–835, 2007.

[30] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid nn/hmm model for speech recognition based on discriminative learning of speaker code," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*. IEEE, 2013, pp. 7942–7946.

[31] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *AIEEE Workshop on utomatic Speech Recognition and Understanding (ASRU), 2015*. IEEE, 2015, pp. 92–97.

[32] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, 2006.

[33] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.

[34] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[35] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.