HYBRID CONVOLUTIONAL NEURAL NETWORKS FOR ARTICULATORY AND ACOUSTIC INFORMATION BASED SPEECH RECOGNITION

Vikramjit Mitra[1], Ganesh Sivaraman[2], Hosung Nam[34], Carol Espy-Wilson[2], Elliot Saltzman[35], Mark Tiede[3]

[1]Speech Technology and Research Laboratory, SRI International, Menlo Park, CA
[2]Department of Electrical and Computer Engineering, University of Maryland, College Park, MD
[3]Haskins Laboratories, New Haven, CT
[4]Department of English Language and Literature, Korea University, Seoul, South Korea
[5]Department of Physical Therapy and Athletic Training, Boston University, Boston, MA

ABSTRACT

Studies have shown that articulatory information helps model speech variability and, consequently, improves speech recognition performance. But learning speaker-invariant articulatory models is challenging, as speaker-specific signatures in both the articulatory and acoustic space increase complexity of speech-to-articulatory mapping, which is already an ill-posed problem due to its inherent nonlinearity and non-unique nature. This work explores using deep neural networks (DNNs) and convolutional neural networks (CNNs) for mapping speech data into its corresponding articulatory space. Our speech-inversion results indicate that the CNN models perform better than their DNN counterparts. In addition, we use these inverse-models to generate articulatory information from speech for two separate speech recognition tasks: the WSJ1 and Aurora-4 continuous speech recognition tasks. This work proposes a hybrid convolutional neural network (HCNN), where two parallel layers are used to jointly model the acoustic and articulatory spaces, and the decisions from the parallel layers are fused at the output context-dependent (CD) state level. The acoustic model performs time-frequency convolution on filterbank-energy-level features, whereas the articulatory model performs time convolution on the articulatory features. The performance of the proposed architecture is compared to that of the CNN- and DNN-based systems using gammatone filterbank energies as acoustic features, and the results indicate that the HCNN-based model demonstrates lower word error rates compared to the CNN/DNN baseline systems.

Index Terms— *automatic speech recognition, articulatory trajectories, vocal tract variables, hybrid convolutional neural networks, time-frequency convolution, convolutional neural networks.*

1. INTRODUCTION

Spontaneous speech typically includes significant variability that is often difficult to model by automatic speech recognition (ASR) systems. Coarticulation and lenition are two sources of such variability (Daniloff and Hammarberg, 1973), and speech-articulation modeling can help to account for such variability (Stevens, 1960). Several studies in the literature (Kirchhoff, 1999; Frankel and King, 2001; Deng and Sun, 1994; Mitra *et al.*, 2013a; Badino *et al.*, 2016; and several others) have demonstrated that speech-production knowledge (in the form of speech articulatory representations) can improve ASR system performance by systematically accounting for variability such as coarticulation. A comprehensive exploration of speech-production features and their role in speech recognition performance is provided in King *et al.* (2007). Further studies (Richardson *et al.*, 2003; Mitra *et al.*, 2010a, 2011a) have demonstrated that articulatory representations provide some degree of noise robustness for ASR systems.

Deep learning techniques (Mohamed *et. al.*, 2012) involving neural networks with several hidden layers have become integral to current ASR systems. Deep learning has been used for feature representation (Hoshen *et al.*, 2015), acoustic modeling (Seide *et al.*, 2011), and language modeling (Arisoy *et al.*, 2012). Given the versatility of the deep neural network (DNN) systems, it was observed in Mitra *et al.* (2014a) that speaker-normalization techniques such as vocal tract length normalization (VTLN) (Zhan and Waibel, 1997) are unnecessary for improving ASR accuracy, as the DNN architecture's rich, multiple projections through multiple hidden layers enable it to learn a speaker-invariant data representation. Convolutional neural networks (CNNs) (Sainath *et al.*, 2013; Abdel-Hamid *et al.*, 2012) motivated by the receptive field theory of the visual cortex are often found to outperform fully connected DNN architectures (Mitra *et al.*, 2014a-b). CNNs are known to be noise robust (Mitra *et al.*, 2014a), especially in those cases where noise/distortion is localized in the spectrum. Speaker-normalization techniques are also found to have less impact on speech recognition accuracy for CNNs as compared to for DNNs (Mitra *et al.*, 2014a). With CNNs, the localized convolution filters across frequency tend to normalize the spectral variations in speech arising from vocal tract length differences, enabling the CNNs to learn speaker-invariant data representations.

Studies have explored using DNNs (Mitra *et al.*, 2010b-c, Uria *et al.*, 2011; Canevari *et al.*, 2013) for learning the nonlinear inverse transform of acoustic waveforms to articulatory trajectories (a.k.a. speech-inversion or acoustic-to-articulatory inversion of speech). Results have demonstrated that using articulatory representations in addition to acoustic features improves phone recognition (Badino *et al.,* 2016; Canevari *et al.*, 2013; Mitra *et al.*, 2011a; Deng and Sun, 1994) and speech recognition performance (Mitra *et al.*, 2011b, 2013a, 2014c). Learning acoustic-to-articulatory transforms is quite challenging, as such mapping is nonlinear and non-unique (Mitra 2010c; Richmond, 2001). Speaker variation adds complexity to the problem and makes speech-inversion even harder (Sivaraman *et al.*, 2015; Ghosh and

Narayanan, 2011). Ghosh and Narayanan (2011) demonstrated that speaker-independent speech-inversion systems can be trained and can offer similar performance to that of speaker-dependent speech-inversion systems.

This work has two principal aims:

(1) Explore deep learning approaches to learn "speaker-independent" speech-inversion mappings using synthetically generated parallel articulatory speech data;

(2) Create a suitable DNN architecture, where retrieved articulatory variables can be used for spontaneous speech recognition purposes.

Note that, this work to the best of our knowledge, for the first time explores the use of Vocal tract constriction variables (TVs) and filterbank features as input to DNN/CNN acoustic models. Hence we explore DNN/CNN configuration to best utilize the TV+filterbank features.

Specifically, we explore CNNs with the hope of achieving more robust speech-inversion performance compared to the traditional DNNs. We investigate using different parameters (such as neural network size and data contextualization (splicing) windows over the input acoustic-feature space) and show that impressive gains can be achieved through careful selection of parameters. We use the trained DNN/CNN models to predict the articulatory trajectories of the training and testing datasets of the *Wall Street Journal* (WSJ1) corpus, the noisy WSJ0 corpus Aurora-4, and the 265-hour Switchboard-1 corpus, and we use the estimated trajectories to perform a continuous speech recognition task.

Further, we present a parallel CNN architecture, where time-frequency convolution is performed on traditional gammatone-filterbank-energy-based acoustic features, and time-convolution is performed on the articulatory trajectories. Each of the parallel convolution layers is followed by a fully connected DNN, whose outputs are combined at the context-dependent (CD) state level, producing senone posteriors. The proposed hybrid CNN (HCNN) architecture learns an acoustic space and an articulatory space, and uses both to predict the CD states.

The word recognition results from both the WSJ1 and Aurora-4 speech recognition tasks indicate that using articulatory information in addition to filterbank features provides sufficient complementary information to reduce the word error rates (WER) in clean, noisy, and channel-degraded conditions.

The novelties of this work are as follows:

(a) Use of large multi-speaker synthetic articulatory data to train robust speech-inversion models. Earlier work has used single-speaker model-based synthetic articulatory data (Mitra et al., 2014c). In this work, we simulate a diverse set of speakers, by varying vocal tract length, pitch, articulatory weights, etc.

(b) Investigate the effect of noise on speech-inversion performance, by comparing CNN and DNN models to analyze their robustness in noisy acoustic conditions.

(c) Explore joint-modeling of articulatory and acoustic spaces in speech recognition tasks. We show that the proposed HCNN better leverages articulatory information compared to simply combining articulatory and acoustic features and training one DNN or CNN acoustic model. In addition, we also explore fusing convolutional-layer feature maps generated from articulatory and acoustic features, and we found that such an approach yields promising results.

Overall, this study demonstrates that given articulatory trajectories and acoustic features are quite different in terms of the information they contain, it is useful to learn intermediate feature spaces from DNN hidden layers or CNN feature maps to fuse the information, rather than fusing the features on the input side and feeding the combined features to a single DNN or CNN acoustic model

2. VOCAL TRACT CONSTRICTION VARIABLES

Articulatory Phonology (Browman & Goldstein, 1989, 1992) is a phonological theory that views speech as a constellation of articulatory gestures that can overlap in time. Gestures are defined as discrete action units whose activation results in constriction formation or release by five distinct constrictors (lips, tongue tip, tongue body, velum, and glottis) along the vocal tract. The kinematic state of each constrictor is defined by its corresponding constriction degree and location coordinates, which are called vocal tract constriction variables (the time-function output of these variables are typically identified as tract-variable trajectories, in short TVs). Please refer to Table 1 and Figure 1 for more details regarding TVs. Table 2 presents the dynamic range and the measuring units for each TVs.

Each gesture is associated with a given constrictor and is specified by an activation onset and offset time, and by a set of dynamic parameters (target, stiffness, and damping); when a given gesture is activated, its parameters are inserted into the associated constrictor's TV equations of motion. These equations are defined as critically damped second-order systems (Saltzman & Munhall, 1989), as shown in (1):

$$M\ddot{z} + B\dot{z} + K(z - z_0) = 0 \qquad (1)$$

where M, B, and K are the mass, damping, and stiffness parameters of each TV (represented by z), and z0 is the TV's target position. Every parameter except M is a time-varying function of the corresponding parameters of the currently active set of gestures. Due to the assumption of constant mass and critical damping, the damping coefficients are constrained to be simple functions of the ongoing stiffness values. The articulatory gestures and their set of tract variable trajectories or time functions (TVs) for an arbitrary utterance can be generated by using the Haskins Laboratories Task Dynamics Application (TaDA, (Nam *et al.,* 2004)). Figure 2 shows the gestural activations for the utterance "miss you," and its corresponding TV trajectories.

Table 1. Constrictors and their vocal tract variables.

Constrictors	Vocal Tract (VT) Variables
Lip	Lip aperture (LA)
	Lip protrusion (LP)
Tongue Tip	Tongue tip constriction degree (TTCD)
	Tongue tip constriction location (TTCL)
Tongue Body	Tongue body constriction degree (TBCD)
	Tongue body constriction location (TBCL)
Velum	Velum (VEL)
Glottis	Glottis (GLO)

Note that gestural onsets and offsets are not always aligned to acoustic landmarks (e.g., the beginning of the frication for /s/ is delayed with respect to the onset of the tongue tip constriction gesture (TTCD) for /s/, due to the time needed for the tongue tip to attain a position close enough to the palate to generate turbulence).

It should also be noted that the TVs GLO and VEL, representing glottal and velic opening/closing, are only obtained in a synthetic speech setup, where the parameters are generated artificially using

TaDA. If deriving TVs from real articulatory measurements, then directly measuring those two TVs may not be possible, as positional data from those articulators may not be available in practice.

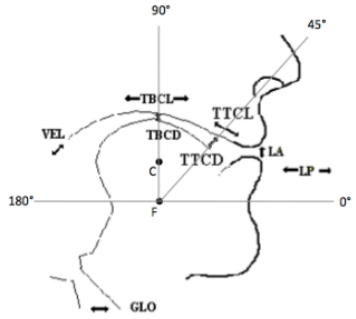

Figure 1. Vocal tract variables at five distinct constriction organs.

Table 2. Units of measurement and dynamic range of each TV.

TVs	*Unit*	*Dynamic range*	
		Max	*Min*
GLO	-	0.74	0.00
VEL	-	0.20	-0.20
LP	mm	12.00	8.08
LA	mm	27.00	-4.00
TTCD	mm	31.07	-4.00
TBCD	mm	12.50	-2.00
TTCL	degree	80.00	0.00
TBCL	degree	180.00	87.00

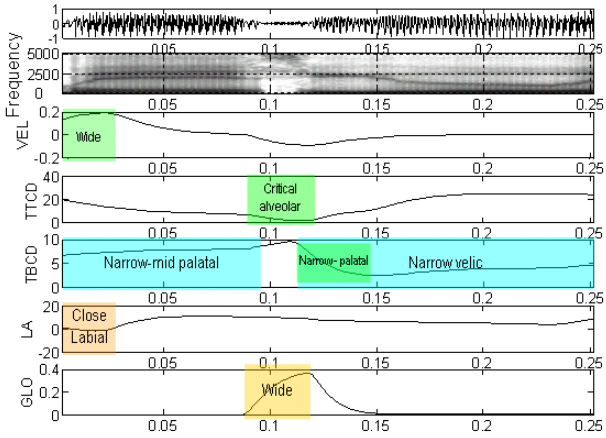

Figure 2. Gestural activations for the utterance "miss you." Active gesture regions are marked by rectangular solid blocks. Smooth curves in the background represent the corresponding TVs.

3. DATASET FOR TRAINING THE SPEECH-INVERSION SYSTEM

To train a model for estimating vocal tract constriction variable trajectories (a.k.a. TVs) from speech, we require a speech database containing ground-truth TVs. However, prior to this work, no speech datasets existed that contain recorded ground-truth TVs and their corresponding speech waveforms. Thus, we used the Haskins Laboratories' Task Dynamic model (TADA) (Nam *et al.,* 2004)) along with HLsyn (Hanson and Stevens, 2002) to generate a synthetic, English isolated word speech corpus along with TVs. TADA along with HLsyn is an articulatory-model-based text-to-speech (TTS) synthesizer that given text as input generates vocal tract constriction variables and corresponding synthetic speech.

In this work, we used the CMU dictionary [22] and selected 111,929 words, whose Arpabet pronunciations we then fed to TADA. In turn, TADA generated their corresponding TVs (refer to Table 1) and synthetic speech. Each word from the CMU dictionary was separately fed to TADA four or five times. For each iteration, TADA randomly selected (a) between a male and a female speaker, whose mean pitch was randomly picked from a uniform distribution; (b) a different speaking rate (fast, normal, or slow); and (c) a different set of articulatory weights to introduce speaker-specific traits. This process enabled simulating a diverse set of speakers. Altogether 534,322 audio samples were generated (approximately 450 hours of speech), out of which 88% of the data was used as the training set, 2% was used as the cross-validation set, and the remaining 10% was used as the test set. We name this as the Synthetic Multi-Speaker clean (SMS-clean) dataset. Note that TADA generated speech signals at a sampling rate of 8 kHz and TVs at a sampling rate of 200 Hz. In addition to the multi-speaker dataset, we have also generated a single-speaker version of the same 112K words from the CMU dictionary, and we call this the Synthetic Single-Speaker clean (SSS-clean) dataset. Please note that SMS-clean and SSS-clean sets are completely disjoint with respect to speaker characteristics. Note that the set of words used in the training, cross-validation and testing data splits were completely disjoint, that is, there were no overlapping words used in any of those data splits. Further, the set of words used in the training-testing-cross-validation splits of SSS-clean were same as those used in the SMS-clean data splits. The training, testing and cross-validation sets for SMS-clean and SSS-clean were created by a non-overlapping split of 88%, 10% and 2% of the respective datasets.

To assess the performance of the speech-inversion system under noisy conditions and to train speech-inversion models with noisy acoustic signals, we added noise to each of the synthetic acoustic waveforms. Fourteen different noise types (such as babble, factory noise, traffic noise, highway noise, crowd noise, etc.) were added with a signal-to-noise ratio (SNR) between 10 to 80 dB. We combined this noise-added data with the SMS-clean data, and the resulting combined dataset is named the Synthetic Multi-Speaker noisy (SMS-noisy) dataset. In addition, we selected a held-out set of ~50K test files and noise types different than that used to create the SMS-noisy set. This unseen noise types consisted of animal noises such as cricket-chirping, dog barking etc., and were added with SNR between 10 to 60 dB, we name this as SMS-unseen-noisy test set. This test set was created to assess the generalization capability of each of the speech inversion models explored in this work.

4. SPEECH INVERSION - TV ESTIMATION

The task of estimating articulatory trajectories (in this case, the TVs) from the speech signal is commonly known as speech-to-articulatory inversion or simply speech-inversion. During speech-inversion, the acoustic features extracted from the speech signal are used to predict the articulatory trajectories, where an inverse mapping is learned by using a parallel corpus containing acoustic and articulatory pairs. The task of speech-inversion is well known to be an ill-posed inverse transform problem, where the challenge

arises from the non-linearity and non-unique nature of the inverse transform (Richmond, 2001; Mitra, 2010c). However, tract variables being a relative measure (e.g., LA is a measure of the distance between the upper and lower lip, instead of an absolute flesh point location defined in Cartesian coordinates as in pellet data), are found to suffer less from non-linearity and non-uniqueness compared to traditional flesh-point measures such as pellet trajectories (McGowan, 1994; Mitra *et al.*, 2010b). Richmond (2001) demonstrated that the challenge of the inverse transform can be reduced by adding context to the input acoustic features.

Based on our previous observations (Mitra *et al.*, 2014c), we explored using speech subband amplitude modulation features such as normalized modulation coefficients (NMCs) (Mitra *et al.*, 2012). NMCs are noise-robust acoustic features obtained from tracking the amplitude modulations (AM) of gammatone-filtered subband speech signals in the time domain. The AM estimates were obtained by using the discrete energy separation algorithm based on the nonlinear Teager's energy operator. The modulation information after root-power compression was used to create a cepstral feature, where the first thirteen discrete cosine transform (DCT) coefficients were retained. These cepstral NMCs are usually known as the NMC cepstral or (NMCC). In addition, we also explored using the above features without the DCT transform, which resulted in a 40-dimensional feature vector, and we denote them as NMCs. The features were Z-normalized before being used to train the DNN/CNN models. Further, the input features were contextualized by splicing multiple frames. In this work, we separately explored the optimal splicing window for the DNN and CNN models.

We explored DNNs and CNNs for training speech-inversion models. Contextualized (spliced) acoustic features in the form of NMCs and NMCCs were used as input, and the TV trajectories were used as the targets. Initially, we kept the splicing fixed at 21 frames (10 frames on either side of the current frame), and we optimized the network size by using the development set. The network's hidden layers had sigmoid activation functions, and the output layer had linear activation. The networks were trained with stochastic gradient descent, where early stopping was used based on the cross-validation error. We optimized the number of hidden layers, the number of neurons, and the splicing window for the DNN- and CNN-based speech-inversion models using the SMS-clean dataset. We found that a four-hidden-layer DNN and a three-hidden-layer CNN containing 2048 neurons in each hidden layer was optimal for the speech-inversion task. The input feature-splicing window was also optimized, where we observed that a splicing size of 75 frames (~375 milliseconds of speech information) for the DNNs and a splicing size of 71 frames (~355 milliseconds of speech information) for the CNNs were good choices. Based on our earlier experiments, we used a convolution layer in the CNN model with 200 filters and a band size of eight, where max-pooling was performed over three samples.

Speech-inversion systems are typically found to be speaker sensitive, with the training of speaker-invariant models challenging. Speaker-dependent inverse models are usually more accurate than speaker-independent models, due to the acoustic variation introduced by different speakers. To compare the performance of speaker-independent models with respect to the speaker-dependent models, we used the SMS-clean and SSS-clean datasets to train and test two DNN speech-inversion systems, and the results are shown in Table 3.

The shape and dynamics of the estimated articulatory trajectories were compared with the actual ones using the Pearson product-moment correlation (PPMC) coefficient. The r_{PPMC} gives a measure of amplitude and dynamic similarity between the groundtruth and the estimated TVs, and are defined as follows-

$$r_{PPMC} = \frac{N\sum_{i=1}^{N} e_i t_i - [\sum_{i=1}^{N} e_i][\sum_{i=1}^{N} t_i]}{\sqrt{N\sum_{i=1}^{N} e_i^2 - (\sum_{i=1}^{N} e_i)^2}\sqrt{N\sum_{i=1}^{N} t_i^2 - (\sum_{i=1}^{N} t_i)^2}} \quad (2)$$

where *e* represents the estimated TV vector and *t* represents the actual TV vector having *N* data points.

Table 3 shows that the performance for the speaker-independent and for the speaker-dependent models are quite similar, where the latter outperforms the former marginally for five out of the eighth TVs. This indicates that the DNN model, given the diverse speaker dataset (SMS-clean), learned a speaker-invariant speech-inversion mapping that, enabled it to perform almost as well as the speaker-dependent model. We also compared the performance of the speaker-dependent model (trained by SSS-clean data) and multi-speaker model (trained by SMS-clean data), by using the multi-speaker held-out SMS-unseen-noisy test and the results are shown in table 4. It can be seen from table 4, that the model trained with multi-speaker data has higher generalization capability than the model trained single speaker data. For almost all TV trajectories in table 4, the SMS-clean trained model outperformed the SSS-clean trained model.

In automatic speech recognition, CNNs have become highly relevant due to their implicit data-driven filtering capability. Vocal tract shape varies with speaker, and the vocal tract is responsible for filtering the glottal source (which is itself variable). Naturally, the differences in vocal tract shape lead to differences in the speech signal's fine spectral structure. Variations in the speech signal due to vocal tract differences adversely impacts ASR performance, and typically vocal tract length normalization (VTLN) techniques are employed (Zhan and Waibel, 1997) to compensate for that. For DNN/CNN models using filterbank energy features, it has been observed that VTLN no longer seems to significantly improve speech recognition accuracy (Mitra *et al.*, 2014d), as the DNN/CNN models rich projections through multiple hidden layers allow them to learn a speaker-invariant representation of the data.

Table 5 presents the r_{PPMC} values from the test set obtained from the DNN and CNN systems trained and tested with SMS-clean data, and we name these models DNN_{CLEAN} and CNN_{CLEAN}. Table 5 shows that the DNN and CNN systems exhibit very similar performance with respect to each other for almost all the TVs for the SMS-clean test set. This similarity in performance can be attributed to the diversity of the training data, which contained numerous speaker configurations, consequently making the DNN system more robust to speaker variation. To investigate the robustness of speech-inversion models to noise, we trained DNN and CNN models using the SMS-noisy train set (we call these models DNN_{NOISY} and CNN_{NOISY}), using the same network configuration as learned from the experiments with the SMS-clean data. The last two rows of table 5 presents the r_{PPMC}values from the SMS-clean test set and shows that the CNN_{NOISY} model almost always outperforms the DNN_{NOISY} model.

Next, we evaluated the DNN_{CLEAN} and CNN_{CLEAN} speech-inversion models by using the noisy test set of the SMS-noisy data, and the results are shown in Table 6, where similar to Table 5, both the DNN and CNN models are found to perform quite similar to each other. We also evaluated the performance of DNN_{NOISY} and

CNN_{NOISY} models on the SMS-noisy test set and the results are shown in last two rows of Table 6. Table 6 shows that the CNN_{NOISY} model outperforms the DNN_{NOISY} model for all the TVs.

Finally, we compared the performance of the DNN_{CLEAN}, CNN_{CLEAN}, DNN_{NOISY} and CNN_{NOISY} models using the SMS-unseen-noisy test, to assess how the performance of these models generalize to unseen noisy types. Table 7 shows the noise trained models (DNN_{NOISY} and CNN_{NOISY}) performed much better than the models trained with clean data only (DNN_{CLEAN} and CNN_{CLEAN}). Table 7 also shows that the CNN_{NOISY} model performed significantly better than the DNN_{NOISY} model. Table 7 shows that the multi-condition trained speech inversion models can generalize well to other noise types.

Results from Tables 5, 6 and 7 suggest that the CNN model learned a more robust and invariant transform from speech to TVs compared to the DNN models, and is thus more noise robust than the DNN model. Comparing the performance of the DNN and CNN models trained with clean and noisy data, irrespective of which dataset was used to train and test the CNN models, their performance remained almost the same, indicating that a CNN model may be a better choice when dealing with varying acoustic conditions.

Table 3. r_{PPMC} for each TV obtained from a speaker-dependent (SSS-clean) and a speaker-independent (SMS-clean) data based DNN speech-inversion system, using held-out test sets from each of those two datasets

	GLO	VEL	LA	LP	TTCD	TTCL	TBCD	TBCL
SSS-clean	0.98	0.96	0.93	0.96	0.95	0.94	0.95	0.97
SMS-clean	0.97	0.95	0.91	0.97	0.95	0.94	0.94	0.96

Table 4. r_{PPMC} for each TV obtained from a speaker-dependent (SSS-clean) and a speaker-independent (SMS-clean) data based DNN speech-inversion system, using SMS-unseen-noisy test set.

	GLO	VEL	LA	LP	TTCD	TTCL	TBCD	TBCL
SSS-clean	0.63	0.62	0.61	0.68	0.62	0.64	0.59	0.78
SMS-clean	0.86	0.80	0.75	0.91	0.80	0.83	0.79	0.89

Table 5. r_{PPMC} for each TV obtained from the best DNN_{CLEAN}, CNN_{CLEAN}, DNN_{NOISY} and CNN_{NOISY} systems when evaluated with the SMS-clean dataset

	GLO	VEL	LA	LP	TTCD	TTCL	TBCD	TBCL
DNN_{CLEAN}	0.97	0.95	0.91	0.97	0.95	0.94	0.94	0.96
CNN_{CLEAN}	0.97	0.96	0.91	0.97	0.95	0.94	0.94	0.96
DNN_{NOISY}	0.96	0.94	0.90	0.96	0.94	0.93	0.92	0.95
CNN_{NOISY}	0.97	0.96	0.92	0.97	0.96	0.94	0.94	0.97

Table 6. r_{PPMC} for each TV obtained from the best DNN_{CLEAN}, CNN_{CLEAN}, DNN_{NOISY} and CNN_{NOISY} systems when evaluated with the SMS-noisy dataset

	GLO	VEL	LA	LP	TTCD	TTCL	TBCD	TBCL
DNN_{CLEAN}	0.85	0.80	0.77	0.91	0.83	0.84	0.80	0.89
CNN_{CLEAN}	0.85	0.83	0.75	0.92	0.81	0.84	0.80	0.89
DNN_{NOISY}	0.93	0.90	0.87	0.95	0.92	0.91	0.89	0.94
CNN_{NOISY}	0.96	0.95	0.91	0.97	0.94	0.93	0.92	0.96

Table 7. r_{PPMC} for each TV obtained from the best DNN_{CLEAN}, CNN_{CLEAN}, DNN_{NOISY} and CNN_{NOISY} systems when evaluated with the SMS-unseen-noisy test set

	GLO	VEL	LA	LP	TTCD	TTCL	TBCD	TBCL
DNN_{CLEAN}	0.86	0.80	0.75	0.91	0.80	0.83	0.79	0.89
CNN_{CLEAN}	0.87	0.84	0.76	0.92	0.80	0.84	0.80	0.89
DNN_{NOISY}	0.93	0.90	0.86	0.95	0.91	0.90	0.88	0.93
CNN_{NOISY}	0.95	0.94	0.90	0.97	0.94	0.93	0.92	0.95

5. DATASET FOR SPEECH RECOGNITION EXPERIMENTS

The DARPA WSJ1 CSR dataset was used in the experiments presented in this paper. For training, a set of 35,990 speech utterances (77.8 hours) from the WSJ1 collection, having 284 speakers was used. For testing, the WSJ-eval94 dataset composed of 424 waveforms (0.8 hours) from 20 speakers was used. Note that for all the experiments reported here, speaker-level vocal tract length normalization (VTLN) was not performed. We denote this dataset as WSJ1 in our experiments described in this paper.

For the speech recognition task under noisy and channel-degraded conditions, we used the Aurora-4 (noisy *Wall Street Journal* [WSJ0]) dataset (Hirsch, 2001). Aurora-4 contains six additive noise versions with channel-matched and mismatched conditions. It was created from the standard 5K WSJ0 database and has 7180 training utterances of approximately 15-hours duration and 330 test utterances. In all experiments, we used 16 kHz sampled data for training and testing our speech recognition systems. Note that TADA, along with HLsyn, generates synthetic speech data sampled at 8 kHz; hence, our speech-inversion system can use a bandwidth of 0 to 4 kHz (corresponding to 8 kHz sampled data) to extract the TVs for speech recognition experiments. In Aurora-4, two training conditions were specified: (1) clean training, which is the full SI-84 WSJ training set without added noise; and (2) multi-condition training, with approximately half of the training data recorded by using one microphone, and the other half recorded by using a different microphone, with different types of added noise at different signal-to-noise ratios (SNRs). The Aurora-4 test data includes 14 test sets from two different channel conditions and six different added noises in addition to the clean condition. The SNR was randomly selected between 0 and 15 dB for different utterances. The six noise types used were: car, babble, restaurant, street, airport, and train station. The evaluation set consisted of 5K words under two different channel conditions. The original audio data for test conditions 1–7 was recorded with a Sennheiser microphone, while test conditions 8–14 were recorded by using a second microphone randomly selected from a set of 18 different microphones (more details in Hirsch, 2001).

In addition to the above two datasets, we also investigated the performance of the tract variable trajectories in the Switchboard (SWB-300) ASR task. For the SWB-300 task, the training data consisted of 262 hours of Switchboard data, which contained telephone-conversation speech between two strangers on a pre-assigned topic. The Hub5 2000 evaluation set was used to evaluate model performance, where 2.1 hours (21.4K words, 40 speakers) of Switchboard data and 1.6 hours (21.6K words, 40 speakers) of CallHome audio. The SWB-300 acoustic models were decoded with a 4-gram language model.

6. ASR SYSTEMS

We trained different acoustic models for the WSJ1 and Aurora-4 speech recognition tasks, where we explored traditional DNNs, CNNs, and time-frequency convolutional nets (TFCNNs) (Mitra and Franco, 2015). The acoustic models were trained with gammatone filterbank energies (GFBs). For SWB-300 ASR task, we trained a six-hidden-layer DNN acoustic model with 2048 neurons in each layer and Damped Oscillator Coefficients (DOCs) (Mitra et al., 2013b) as the acoustic feature. The DOC features model the auditory hair cells using a bank of forced damped oscillators, where gammatone filtered bandlimited subband speech signals are used as the forcing function. The oscillation energy from the damped oscillators are used as the DOC features after power-law compression. From our experiments with SWB-300, we observed that the DOC features after feature space maximum likelihood linear regression (fMLLR)-based speaker adaptation, using a sequence trained DNN model, provides a strong baseline system.

The gammatone filters are a linear approximation of the auditory filterbank of the human ear. In GFB processing, speech is analyzed by using a bank of 40 gammatone filters equally spaced on the equivalent rectangular bandwidth (ERB) scale. For this work, the power of the bandlimited time signals within an analysis window of ~26 milliseconds was computed at a frame rate of 10 milliseconds. Subband powers were then root compressed by using the 15^{th} root, and the resulting 40-dimensional feature vector was used as the GFB.

It was shown (Mitra *et al.*, 2014d) that CNNs give lower WERs compared to DNNs when using filterbank features for the Aurora-4 ASR task, and GFBs offered performance gain over mel-filterbank energies (MFBs). Hence, in this study, we used the GFB-CNN model as our baseline system; however, for the sake of clarity, we show the performance of the GFB-DNN systems as well.

To generate the alignments necessary for training the CNN system, a Gaussian mixture model (GMM)–hidden Markov model (HMM) model was used to produce the senones' labels. Altogether, the GMM-HMM system produced 3,162 context-dependent (CD) states for Aurora-4 and 1,659 CD states for WSJ1. The input features to the acoustic models were formed by using a context window of 15 frames (7 frames on either side of the current frame).

The acoustic models were trained by using cross-entropy on the alignments from the GMM-HMM system. For the CNN, 200 convolutional filters of size 8 were used in the convolutional layer, and the pooling size was set to 3 without overlap. The subsequent fully connected network had four hidden layers, with 1024 nodes per hidden layer, and the output layer included as many nodes as the number of CD states for the given dataset. The networks were trained by using an initial four iterations with a constant learning rate of 0.008, followed by learning-rate halving based on cross-validation error decrease. Training stopped when either no further significant reduction in cross-validation error was noted or when cross-validation error started to increase. Backpropagation was performed using stochastic gradient descent with a mini-batch of 256 training examples. For the DNN systems, we used five layers with 1024 neurons in each layer, with similar learning criteria as the CNNs.

The TFCNN architecture was based upon Mitra & Franco (2015), where two parallel convolutional layers were used at the input, one performing convolution across time, and the other across the frequency scale of the input filterbank features. That work showed that the TFCNNs gave better performance compared to their CNN counterparts. Here, we used 75 filters to perform time convolution, and 200 filters to perform frequency convolution. Note that the convolutional-layer configurations for the TFCNN model were investigated in our earlier work, Mitra & Franco (2015), and the optimal configuration learned from that work is used in the experiments reported in this paper. For time and frequency convolution, eight bands were used. A max-pooling over three samples was used for frequency convolution, while a max-pooling over five samples was used for time convolution. The feature maps after both the convolution operations were

concatenated and then fed to a fully connected neural net, which had 1024 nodes and four hidden layers.

In this work, we present a modified deep neural network architecture to jointly model the acoustic and the articulatory space. The diagram of the network is shown in Figure 3, illustrating two parallel neural networks trained simultaneously. These two parallel neural networks modeled two things: (1) learning the acoustic space from the GFB features and (2) learning the articulatory space from the TV trajectories. The acoustic space was learned by using a time-frequency convolution layer, where two separate convolution filters operate on the input GFB features. These two convolution layers had the same parameter specification as that used in the TFCNNs. The articulatory space was learned by using a time-convolution layer that contained 75 filters, followed by a max-pooling over five samples. Note that the cross-TV convolution operation may not produce any meaningful information, whereas time convolution on the TVs can help in extracting TV modulation-level information, which was the motivation behind selecting a time-convolution layer for learning the articulatory space. The fully connected DNN layers were different in size; we observed that 800 neurons was nearly optimal for learning the acoustic space, and that 256 neurons was nearly optimal for learning the articulatory space. Note that the parallel networks were jointly trained.

We also investigated fusing information at the feature-map level, where we jointly learned convolutional layers operating on acoustic features (frequency-convolution) and TV trajectories (temporal convolution). Unlike our previous work (Mitra et al., 2016) where feature maps generated from two frequency convolution layers each operating on a different acoustic feature were fused, we performed frequency convolution on the spectral acoustic features and time convolution on the TV trajectories, and we fused the feature maps from these two layers to feed a single, fully connected DNN, see figure 4. We name this configuration the fused CNN (fCNN), where 200 filters were used for acoustic feature frequency convolution and 75 filters were used for TV-trajectory time convolution.

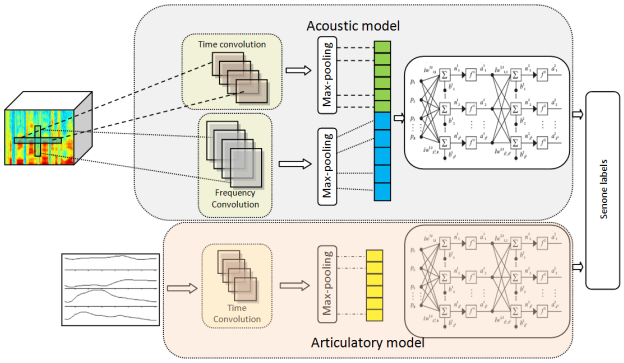

Figure 3. Schematics of the hybrid convolutional neural network (HCNN). The top layer represents the acoustic model, whose input is filterbank features, and the bottom layer represents the articulatory model, whose input is TV trajectories.

For clarity's sake, we also tried combining acoustic features (such as GFB features) with the TVs and then training one single network by using the combined feature. Such a system can have several drawbacks: a CNN or TFCNN may not be a technically sound architecture, as the spatial convolution filter operating across GFB features will capture meaningful information, but the same filter operating across TVs or TV-GFB boundaries may not be meaningful. The only meaningful architecture in such a case is a DNN; however, based on prior studies, we know that DNNs are slightly inferior to CNNs in terms of performance. Hence, to get the best of both DNNs and CNNs, we designed the proposed hybrid convolutional neural network, which performs relevant convolutional operations based on individual feature types.

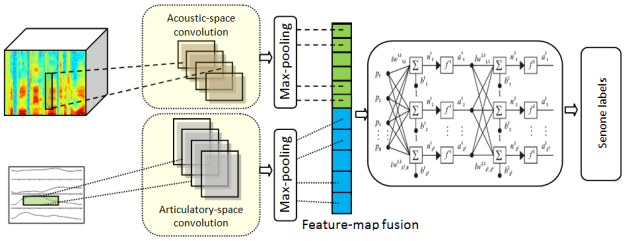

Figure 4. Schematics of a fused-feature-map convolutional neural network (fCNN). The top convolutional layer (across frequency scale) operates on the acoustic features, which are the filterbank energy features, and the bottom convolutional layer (across time scale) operates on articulatory features, which are the TV trajectories.

7. SPEECH RECOGNITION EXPERIMENTS AND RESULTS

The eight TVs given in Tables 1 and 2 are insufficient by themselves for use as ASR features (Mitra *et al.*, 2014c); hence, they are typically combined with standard acoustic features. Our initial ASR experiments were on Aurora-4, where the baseline system is the same as that reported in Mitra and Franco (2015). As baseline acoustic features, we tried both mel-filterbank (MFB) and gammatone filterbank (GFB) features. Given that Aurora-4 has 14 different evaluation conditions depending upon the noise conditions and microphone types, we used the standard partition of the evaluation set to report our results, which are outlined in table 8.

Table 8. Aurora-4 evaluation partitions.

	Sennheiser microphone	Randomly selected microphone
Clean speech	A	B
Noisy Speech	C	D

Table 9 shows the results from the DNN, CNN, and TFCNN systems, represented in the form of word error rates (WERs). Note that in all the Aurora-4 experiments reported in this paper, we used the standard trigram language model distributed with the WSJ0 dataset.

Table 9 shows that the CNN models perform better than the DNN models, and that TFCNN performs the best for both features, where the TFCNN-MFB system offers significant performance gain over the CNN-MFB system. However, with GFB features, the performance difference between the CNN and TFCNN systems is insignificant. Note that based on our prior observations (Mitra *et al.*, 2014a), we used a five-hidden-layer DNN and a four-hidden-layer + one-convolutional-layer CNN. The TFCNN had four

hidden layers + one frequency convolutional layer + one time convolutional layer.

Next, we extracted the estimated TV trajectories for the training, testing, and cross-validation sets for the Aurora-4 multi-conditioned train-test evaluation by using the speech-inversion systems presented in Section 4. The estimated eight TV trajectories were used in conjunction with the baseline acoustic features. Table 9 shows that the GFB features provide a better baseline than the MFB ones; hence, we used the GFB features as the baseline acoustic feature in conjunction with the estimated TV trajectories. The TV trajectories estimated from using the DNN_{CLEAN} and DNN_{NOISY} speech-inversion models are termed as the TV-DNN_{CLEAN} and TV-DNN_{NOISY}, respectively. Those obtained from the CNN_{CLEAN} and CNN_{NOISY} models are termed the TV-CNN_{CLEAN} and TV-CNN_{NOISY}, respectively. As initial experiments, we did a simple feature fusion of the GFB feature and the estimated TVs (TV-DNN_{CLEAN} and TV-CNN_{CLEAN}), and trained and tested the DNN, CNN, and TFCNN systems. Table 10 shows the results.

Table 9. WER on multi-conditioned training task of Aurora-4 (16 kHz) from the different acoustic models using MFB and GFB baseline features.

Features	Models	A	B	C	D	avg.
MFB	DNN	4.3	9.6	8.8	18.4	12.9
	CNN	3.5	6.2	6.6	15.7	10.1
	TFCNN	3.6	5.8	6.6	14.6	9.5
GFB	DNN	3.3	6.9	7.7	17.8	11.4
	CNN	3.1	6.1	5.2	14.5	9.4
	TFCNN	3.1	5.7	6.1	14.6	9.4

Table 10. WER on multi-conditioned training task of Aurora-4 (16 kHz) from the different acoustic models using GFB + estimated TVs.

Features	Models	A	B	C	D	avg.
GFB	TFCNN	3.1	5.7	6.1	14.6	9.4
GFB+ TV-DNN_{CLEAN}	DNN	3.8	7.4	8.8	19.5	12.5
	CNN	3.3	6.2	6.1	15.1	9.8
	TFCNN	3.2	5.9	6.3	14.8	9.5
GFB+ TV-CNN_{CLEAN}	DNN	3.7	7.7	8.9	19.2	12.4
	CNN	3.6	6.2	6.2	14.7	9.7
	TFCNN	3.2	6.0	6.0	15.0	9.6

Table 10 shows that simple combination of the acoustic features with the articulatory features did not work well. For all modeling types, the combination of the GFB features with the estimated TV trajectories resulted in increased WER, suggesting that simple feature-level fusion may not be a useful approach.

In our earlier experiments (Mitra et al., 2014c), we used a GMM-HMM acoustic model trained on a simple concatenation of acoustic features (mel-frequency cepstral coefficients, a.k.a MFCCs) and TV trajectories, which were dimensionality reduced by a principal component analysts (PCA) transform. In such a setup, the dimensionality reduction using PCA after acoustic feature and articulatory feature concatenation was a key component that helped to improve the performance beyond the acoustic-feature-only baseline. This work investigates DNN/CNN acoustic models and similar to before, we observed that a simple concatenation of the two feature spaces (acoustic and articulatory) may not be useful, as each of them may be capturing different linguistic attributes of speech, which can often be complimentary to each other.

Given the difference in characteristics of the acoustic feature (GFB) and the estimated TVs, using separate convolutional filtering on each of them is intuitive. GFB are spectral-level features that have spatial correlation across the feature dimensions; hence, frequency convolution is meaningful for such features. The TVs, on the other hand, are purely time trajectories of vocal tract constrictions; for them, temporal modulation extraction is more meaningful than cross-TV correlation extraction. Hence, we explored the hybrid convolutional net (HCNN), as shown in Figure 3. Note that the HCNN performs time-frequency convolution on the spectral feature (such as the GFB or MFBs) and only time convolution on the TVs. We further noticed that a much smaller number of neurons were sufficient for the hidden layers modeling the TV trajectories than those used for modeling the GFBs. We used 800 neurons in the four hidden layers processing the GFB features, and 256 neurons in the four hidden layers processing the TV trajectories; hence, the total number of neurons in the hidden layers between the HCNN and the baseline systems are comparable. Table 11 shows the WERs obtained from the HCNNs trained with the GFB+TV-DNN_{CLEAN}, GFB+TV-CNN_{CLEAN}, GFB+TV-DNN_{NOISY}, and GFB+TV-CNN_{NOISY} features. Note that the features used a splicing of 15 frames, meaning a combination of 7 frames from either side of the current frame, inclusive. Note that from prior experiments we have observed that TFCNNs almost always perform better than the CNNs (Mitra & Franco, 2015), more specifically for reverberated conditions. As reverberation is a temporal distortion, the time convolution in TFCNNs help to reduce the effect of reverberation. Also we observed that for MFBs, TFCNNs significantly improve the performance (see table 9) making them competitive with respect to noise robust features. Because of the versatility of TFCNNs performance over CNNs, we have used them in our HCNN architecture, rather than using the traditional CNNs to process the acoustic observations.

Table 11 shows that the HCNN systems overall perform better than the baseline system, demonstrating more than 5% overall relative reduction in WER compared to the baseline. Note that the TVs estimated from the CNN system performed a little better than those from the DNN system, giving lower error rates in clean and channel mismatched cases. This finding is also evident from the r_{PPMC} scores shown in Tables 6 and 7, where the CNN-based speech-inversion model trained on SMS-noisy data was found to perform better than the corresponding DNN model. Also, note that the HCNN's major contribution was in conditions C and D, which represented channel mismatch clean (C) and noisy (D) scenarios. The results in Table 11 indicate that the additional articulatory information helped to improve the ASR performance in both matched and mismatched conditions. Table 11 also show the results from the fCNN systems, where the results were similar as HCNN systems. These results and the results in Table 10 indicate that the benefits of the HCNN and fCNN systems derive both from the systems' individual convolutional layers tied to the acoustic features and articulatory features, and from using time convolution only for the articulatory features, which results in performance improvement over the GFB baseline, which was not observed in Table 10, where the features were concatenated together and were fed to the same convolutional layer.

To get a more detailed understanding regarding how the articulatory features helped in each noise and channel conditions in

Aurora-4, we compare the individual WERs for each testing condition, which is shown in Table 12.

Table 12 shows that for the clean conditions, using the articulatory features always improved the performance. We observed that for clean data conditions the percentage of correct recognitions increased, while both substitution and deletions decreased; indicating that the additional TV information helped to improve the discriminative power of the speech recognition model. Overall, the articulatory features reduced the WER in all conditions, except babble and street in the Sennheiser microphone condition. For the second microphone condition (i.e., using a microphone selected randomly from a set of 18 microphones), the articulatory features always improved performance. For train-station noise, using articulatory features always reduced the WER. Significance test on the results from the clean matched channel data indicated that the GFB+TV-CNN_{NOISY}-HCNN systems is significantly better ($p < 0.001$) than the GFB-TFCNN system.

In addition to the Aurora-4 speech recognition task, we also applied the HCNN architecture to the clean WSJ1 evaluation task. Similar to the findings in Table 9, we observed that the CNN models perform much better than the DNN acoustic models for the baseline GFB features. The DNN systems had five hidden layers with 1024 neurons in each layer, whereas the CNN systems had four hidden layers of 1024 neurons in each layer and one convolutional layer with 200 filters. Table 13 shows the WERs from the MFB, GFB, GFB+TV-DNN_{CLEAN} and GFB+TV-CNN_{CLEAN} systems for WSJ1 speech recognition evaluation task.

Table 11. WER on multi-conditioned training task of Aurora-4 (16 kHz) from the baseline system using GFB feature and the HCNN using GFB + estimated TV features.

	Model	A	B	C	D	avg.
GFB	TFCNN	3.1	5.7	6.1	14.6	9.4
GFB+TV-DNN_{CLEAN}	HCNN	3.3	5.7	5.5	14.2	9.2
GFB+TV-CNN_{CLEAN}	HCNN	3.0	5.7	5.5	14.2	9.1
GFB+TV-DNN_{NOISY}	HCNN	2.8	5.6	5.4	14.3	9.0
GFB+TV-CNN_{NOISY}	HCNN	2.6	5.6	5.5	14.0	**8.9**
GFB+TV-DNN_{NOISY}	fCNN	2.8	5.5	5.8	13.8	**8.9**
GFB+TV-CNN_{NOISY}	fCNN	2.8	5.5	5.4	14.1	9.0

Table 12. WER on multi-conditioned training task of Aurora-4 (16 kHz) from the baseline system using GFB feature and the HCNN using GFB + estimated TV features.

	Clean	Car	Babble	Restau-rant	Street	Airport	Train-station	Clean	Car	Babble	Restau-rant	Street	airport	Train-station	avg.
	Sennheiser Microphone							2^{nd} microphone selected randomly from a set of 18							
GFB-TFCNN	3.1	3.7	5.6	7.3	5.8	5.5	6.1	6.1	7.1	15.5	18.1	15.0	14.7	17.4	9.4
GFB+TV-CNN_{NOISY} HCNN	2.6	3.5	5.7	6.9	6.1	5.4	5.8	5.5	6.8	15.4	16.6	14.7	14.6	16.0	**8.9**
GFB+TV-DNN_{NOISY} fCNN	2.8	3.4	5.6	7.0	6.1	5.6	5.4	5.8	6.7	14.5	17.1	14.7	14.3	15.4	**8.9**

Table 13. WER from WSJ1 ASR experiments using baseline features (MFB and GFB) and GFB + estimated TV feature with different modeling techniques.

Features	Models	WER
MFB	DNN	6.7
GFB	DNN	6.4
GFB	CNN	5.7
GFB	TFCNN	5.7
GFB+TV-DNN_{CLEAN}	HCNN	**5.4**
GFB+TV-CNN_{CLEAN}	HCNN	5.6
GFB+TV-DNN_{CLEAN}	fCNN	5.6
GFB+TV-CNN_{CLEAN}	fCNN	5.7

Table 13 shows that the GFB features are a better baseline feature than the MFBs, and that the CNN models offer lower error rates than DNNs. Interestingly, using the TFCNN model with GFB feature shows no improvement over its CNN counterparts. When estimated TVs were used along with the GFBs in the HCNN models, a reduction in error rate was observed. The TVs from the DNN-based speech-inversion system gave the best performance in the WSJ1 ASR task, with an observed relative reduction of 5.2% in WER compared to the WER for the best CNN-GFB baseline system.

For the SWB-300 baseline model, we trained a six-hidden layer DNN having 2048 neurons, using fMLLR transformed damped oscillator cepstral coefficient (DOCC) (Mitra et al., 2013b) features as input. The estimated TVs from the DNN speech-inversion model was appended with the DOCC features, and they were fMLLR transformed to train a six-hidden-layered DNN with 2048 neurons. Table 14 shows the results from the sequence-trained DNN models.

Table 14 WER from SWB-300 ASR experiments using SWB part of the Hub5 eval data.

Features	WER
DOCC	12.8
DOCC+TV-DNN$_{CLEAN}$	12.1

8. CONCLUSION

In this work, we presented DNN- and CNN-based speech-inversion systems for estimating articulatory trajectories from the speech signal. We demonstrated that with suitable network parameter selection, we could estimate with high accuracy articulatory trajectories in the form of vocal tract constriction variables, where the correlation coefficient between the estimated and ground-truth trajectories correlated was greater than 0.9. We also investigated noise robustness of speech-inversion systems, where we observed that speech-inversion performance degrades with presence of noise in the acoustic signal. We observed that training the speech-inversion model with noisy data improves its noise robustness. We also observed that sufficient diversity of speaker data enables training speech-inversion models that perform as well as speaker-specific inversion models.

We proposed a hybrid convolutional neural network (HCNN), in which two parallel layers were used to jointly model the acoustic and articulatory spaces. The parameters of these two networks were learned jointly with one objective function, with these two networks sharing the same output layer. Speech recognition results on the Aurora-4 and WSJ1 recognition tasks showed that the proposed architecture using articulatory features demonstrated reduction in word error rates for each of the clean, noisy, and channel-mismatched conditions. For the Aurora-4 and WSJ1 ASR tasks, the best WERs from the HCNN system were found to be 8.9% and 5.4%, respectively, which, to the best of our knowledge, are state-of-the-art results for these datasets. We also observed significant improvement in performance for the SWB-1 speech recognition task when articulatory features were used with the DOCC features, compared to using the DOCC features alone.

In the future, we will investigate using HCNN for ASR tasks involving languages other than English. The impact of the time resolution of the tract variable trajectories for articulatory space modeling was not investigated in great detail in this study. Given that the HCNN performs time convolution on the articulatory features, we must investigate whether finer articulatory resolution could uncover more detail about articulatory trajectory temporal modulation. We also must explore if hidden variables in the form of bottleneck features can be used for ASR. Using bottleneck features derived from traditional acoustic features has recently shown significant performance gains, and using articulatory-trajectory-based bottleneck features may potentially introduce complementary information and hence possibly add to those performance gains.

9. ACKNOWLEDGMENTS

This research was supported by NSF Grant # IIS-0964556, IIS-1162046 and IIS-1161962.

10. REFERENCES

Abdel-Hamid, O., Mohamed, A., Jiang, H., and Penn, G. (2012) "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," *Proc. of ICASSP*, pp. 4277–4280,.

Arisoy, E., Sainath, T.N., Kingsbury, B., and Ramabhadran, B. (2012) "Deep neural network language models," *Proc. of NAACL-HLT Workshop*.

Badino, L., Canevari, C., Fadiga, L., and Metta, G. (2016) "Integrating articulatory data in deep neural network-based acoustic modeling," *Computer Speech & Lang*., Vol. 36, pp. 173–195.

Browman, C., and Goldstein, L. (1989) "Articulatory gestures as phonological units," *Phonology*, 6, pp. 201–251.

Browman, C., and Goldstein, L. (1992) "Articulatory phonology: An overview," *Phonetica*, 49, pp.155–180.

Canevari, C., Badino, L., Fadiga, L., and Metta, G. (2013) "Relevance-weighted reconstruction of articulatory features in deep neural network-based acoustic-to-articulatory mapping," *Proc. of Interspeech*.

Daniloff, R., and Hammarberg, R. (1973) "On defining coarticulation," *J. of Phonetics*, Vol.1, pp. 239–248.

Deng, L., and Sun, D. (1994) "A statistical approach to automatic speech recognition using atomic units constructed from overlapping articulatory features," *J. of Acoust. Soc. Am.*, 95(5), pp. 2702–2719.

Frankel, J., and King, S. (2001) "ASR—Articulatory speech recognition," *Proc. of Eurospeech*, pp. 599–602, Denmark.

Ghosh, P.K., and Narayanan, S.S. (2011) "A subject-independent acoustic-to-articulatory inversion," *Proc. of ICASSP*, pp. 4624–4627.

Hanson, H. M., and Stevens, K.N. (2002) "A quasiarticulatory approach to controlling acoustic source parameters in a Klatt-type formant synthesizer using HLsyn," *J. of Acoust. Soc. Am.*, 112(3), pp. 1158–1182.

Hirsch, G. (2001) "Experimental framework for the performance evaluation of speech recognition front-ends on a large vocabulary task," *ETSI STQ-Aurora DSR Working Group*.

Hoshen, Y., Weiss, R.J., and Wilson, K.W. (2015) "Speech acoustic modeling from raw multichannel waveforms," *Proc. of ICASSP*, pp. 4624–4628.

King, S., Frankel, J., Livescu, K., McDermott, E., Richmond, K., and Wester, M. (2007) "Speech production knowledge in automatic speech recognition," *J. Acoust. Soc. of Am.*, 121(2), pp. 723–742.

Kirchhoff, K. (1999) *Robust Speech Recognition Using Articulatory Information*, PhD Thesis, University of Bielefeld.

McGowan, R.S. (1994), "Recovering articulatory movement from formant frequency trajectories using task dynamics and a genetic algorithm: preliminary model tests," *Speech Comm.,* 14(1), pp. 19–48.

Mitra, V., Nam, H., Espy-Wilson, C., Saltzman, E. and Goldstein, L. (2010a) "Articulatory information for noise robust speech recognition," *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 19, Iss. 7, pp. 1913–1924, 2010.

Mitra, V., Nam, H., Espy-Wilson, C., Saltzman, E., and Goldstein, L. (2010b) "Retrieving tract variables from acoustics: A

comparison of different machine learning strategies," *IEEE Journal of Selected Topics on Signal Processing*, Sp. Iss. on Statistical Learning Methods for Speech and Language Processing, Vol. 4, Iss. 6, pp. 1027–1045.

Mitra, V. (2010c) *Articulatory Information for Robust Speech Recognition*, PhD Thesis, Department of Electrical and Computer Engineering, University of Maryland, College Park, December 2010, College Park, MD.

Mitra, V., Nam, H. and Espy-Wilson, C. (2011) "Robust speech recognition using articulatory gestures in a dynamic bayesian network framework," *Proc. of Automatic Speech Recognition & Understanding Workshop*, ASRU, pp. 131–136, Hawaii.

Mitra, V., Nam, H., Espy-Wilson, C., Saltzman, E. , Goldstein, L. (2011b) "Gesture-based dynamic Bayesian network for noise robust speech recognition," *Proc. of ICASSP*, pp. 5172–5175, 2011.

Mitra, V., Franco, H., Graciarena M. and Mandal, A. (2012) "Normalized amplitude modulation features for large vocabulary noise-robust speech recognition," *Proc. of ICASSP*, pp. 4117–4120.

Mitra, V., Wang, W., Stolcke, A., Nam, H., Richey, C., Yuan, J. and Liberman, M. (2013a) "Articulatory features for large vocabulary speech recognition," *Proc. IEEE ICASSP*, Vancouver.

Mitra, V. Franco, H. and Graciarena, M. (2013b) "Damped oscillator cepstral coefficients for robust speech recognition," in Interspeech, 2013, pp. 886–890.

Mitra, V., Wang, W., Franco, H., Lei, Y., Bartels, C. and Graciarena, M. (2014a) "Evaluating robust features on deep neural networks for speech recognition in noisy and channel mismatched conditions," *Proc. of Interspeech*.

Mitra, V., Wang, W., and Franco, H. (2014b) "Deep convolutional nets and robust features for reverberation-robust speech recognition,"*Proc. of SLT*, pp. 548–553.

Mitra, V., Sivaraman, G., Nam, H., Espy-Wilson, C., and Saltzman, E. (2014c) "Articulatory features from deep neural networks and their role in speech recognition," *Proc. of ICASSP*, pp. 3041–3045, Florence.

Mitra, V., Wang, W., Franco, H., Lei, Y., Bartels, C., and Graciarena, M. (2014d) "Evaluating robust features on deep neural networks for speech recognition in noisy and channel mismatched conditions," *Proc. of Interspeech*, pp. 895–899.

Mitra, V., and Franco, H. (2015) "Time-frequency convolution networks for robust speech recognition," *Proc. of ASRU*.

Mitra, V., VanHout, J., Wang, W., Bartels, C., Franco, H., Vergyri, D., Alwan, A., Janin, A., Hansen, J., Stern, R., Sangwan A., and Morgan, N. (2016) "Fusion Strategies for Robust Speech Recognition and Keyword Spotting for Channel- and Noise-Degraded Speech," in Proc. of Interspeech 2016.

Mohamed, A., Dahl, G.E., and Hinton, G. (2012) "Acoustic modeling using deep belief networks," *IEEE Trans. on ASLP*, vol. 20, no. 1, pp. 14 –22.

Nam, H., Goldstein, L., Saltzman, E., and Byrd, D. (2004), "TADA: An enhanced, portable task dynamics model in Matlab," *J. Acoust. Soc. of Am.*, 115(5), pp. 2430.

Richardson, M., Bilmes, J., and Diorio, C. (2003) "Hidden-articulator Markov models for speech recognition," *Speech Comm.*, 41(2-3), pp. 511–529.

Richmond, K. (2001) "Estimating articulatory parameters from the acoustic speech signal," *PhD Thesis*, Univ. of Edinburgh, UK.

Sainath, T., Mohamed, A., Kingsbury, B., and Ramabhadran, B. (2013) "Deep convolutional neural network for LVCSR," *Proc. of ICASSP*.

Saltzman, E., and Munhall, K. (1989), "A dynamical approach to gestural patterning in speech production," *Ecological Psychology*, 1(4), pp. 332–382

Seide, F., Li, G., and Yu, D. (2011) "Conversational speech transcription using context-dependent deep neural networks," *Proc. of Interspeech*.

Sivaraman, G., Mitra, V., Tiede, M., Saltzman, E., Goldstein, L., and Espy-Wilson, C. (2015) "Analysis of coarticulated speech using estimated articulatory trajectories," *Proc. of Interspeech*.

Stevens, K.N. (1960) "Toward a model for speech recognition," *J. of Acoust. Soc. Am.*, Vol.32, pp. 47–55.

Uria, B., Renals, S., and Richmond, K. (2011) "A deep neural network for acoustic-articulatory speech inversion," *NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning*.

Zhan, P., and Waibel, A. (1997) "Vocal tract length normalization for LVCSR," in *Tech. Rep. CMU-LTI-97-150*. Carnegie Mellon University.