Formant estimation and tracking: A deep learning approach

Yehoshua Dissen, Jacob Goldberger, and Joseph Keshet

Citation: The Journal of the Acoustical Society of America **145**, 642 (2019); doi: 10.1121/1.5088048
View online: https://doi.org/10.1121/1.5088048
View Table of Contents: https://asa.scitation.org/toc/jas/145/2
Published by the Acoustical Society of America

Formant estimation and tracking: A deep learning approach

Yehoshua Dissen,[1] Jacob Goldberger,[2] and Joseph Keshet[1,a)]
[1]*Department of Computer Science, Bar-Ilan University, Ramat Gan, 52900, Israel*
[2]*Faculty of Engineering, Bar-Ilan University, Ramat Gan, 52900, Israel*

(Received 23 May 2018; revised 5 December 2018; accepted 5 January 2019; published online 4 February 2019)

Formant frequency estimation and tracking are among the most fundamental problems in speech processing. In the estimation task, the input is a stationary speech segment such as the middle part of a vowel, and the goal is to estimate the formant frequencies, whereas in the task of tracking the input is a series of speech frames, and the goal is to track the trajectory of the formant frequencies throughout the signal. The use of supervised machine learning techniques trained on an annotated corpus of read-speech for these tasks is proposed. Two deep network architectures were evaluated for estimation: feed-forward multilayer-perceptrons and convolutional neural-networks and, correspondingly, two architectures for tracking: recurrent and convolutional recurrent networks. The inputs to the former are composed of linear predictive coding–based cepstral coefficients with a range of model orders and pitch-synchronous cepstral coefficients, where the inputs to the latter are raw spectrograms. The performance of the methods compares favorably with alternative methods for formant estimation and tracking. A network architecture is further proposed, which allows model adaptation to different formant frequency ranges that were not seen at training time. The adapted networks were evaluated on three datasets, and their performance was further improved.
 https://doi.org/10.1121/1.5088048

[CYE] Pages: 642–653

I. INTRODUCTION

Accurate formant measurements have been widely used in the study of vowels. They are used, for example, in characterizing the vowel space of men, women, and children (Peterson and Barney, 1952) and comparing dialects [Clopper and Bradlow, 2009; Clopper and Pierrehumbert, 2008; Clopper and Tamati, 2014 (CT)], languages (Bradlow, 1995; Smiljanić and Bradlow, 2005), and speaking styles (Bradlow, 2002; Pierrehumbert *et al.*, 2004). They are often used in studies of hearing-impaired speech production and perception (Ferguson and Kewley-Port, 2002; Monsen, 1978), and they are used as a tool in the study of pronunciation and the effects of phonological neighborhood density and word frequency (Munson and Solomon, 2004).

Formant measurements have been also used in the coding, synthesis, and enhancement of speech, as they can express important aspects of the signal using a very limited set of parameters (O'Shaughnessy, 2007). It has received considerable attention in speech recognition research as formant frequencies are known to be important in determining the phonetic content as well as articulatory information about the speech signal. They can either be used as additional acoustic features or utilized as hidden dynamic variables as part of the speech recognition model (Deng and Ma, 2000).

Originally, formants were defined as the peaks in the output spectrum envelope radiated from the mouth (Titze *et al.*, 2015), and similarly in the current Acoustical Society of America (ASA) standard of acoustic terminology (ANSI, 1994). Most commonly, the spectral envelope is estimated using a time-invariant all-pole linear system, and the formants are estimated by finding the peaks of the spectral envelope (McCandless, 1974; O'Shaughnessy, 2007). While this method is very simple and efficient, it lacks the accuracy required by some systems.

Most algorithms for tracking are based on traditional peak picking from linear predictive coding (LPC) spectral analysis or cross-channel correlation methods coupled with continuity constraints (Deng and Geisler, 1987; McCandless, 1974; O'Shaughnessy, 2007). The LPC-based methods assume that speech can be modeled by the source-filter model, and the vocal tract can be modeled by an all-pole digital filter.

More elaborate methods used dynamic programming and hidden Markov models (HMMs) to force continuity (Kopec, 1986; Lee *et al.*, 2005; Toledano *et al.*, 2006). Other algorithms for formant tracking are based on Kalman filtering (Deng *et al.*, 2004, 2007) and their extensions (Mehta *et al.*, 2012). Some authors used an autocorrelation sequence for representing speech in a noisy speech recognition system (Cadzow, 1982; Hernando *et al.*, 1997), while others used LPCs of the zero phase version of the signal and the peaks of its group delay function (Anand *et al.*, 2006; Murthy and Yegnanarayana, 2011; Ribas Gonzalez *et al.*, 2014).

In 2006, a publicly available corpus of manually annotated formant frequencies of read speech was released (Deng *et al.*, 2006). The corpus is a subset of the TIMIT corpus (Garofolo *et al.*, 1993), and includes around 30 min of transcription of the first four formants at the 10 ms frame resolution. The release of this database enables researchers to develop and evaluate new algorithms for formant estimation.

a)Electronic mail: jkeshet@cs.biu.ac.il

In this paper, we present deep learning methods for estimating and tracking formant frequencies using deep networks trained on the aforementioned annotated corpus. In the task of formant *estimation* the input is a stationary speech segment (such as the middle of a vowel), and the goal is to estimate the first three formants. In the task of formant *tracking* the input is a sequence of speech frames, and the goal is to predict the sequence of the first three formants corresponding to the input sequence. Estimation can be used in the study of vowels as it is more accurate and assumes the input of a single vowel, while tracking can be used for speech coding and synthesis and as features in speech recognition models.

In both tasks the signal is represented as either a set of two spectral envelope representations or the raw spectrograms. The first set is composed of LPC cepstral coefficients extracted from a *range* of LPC model orders and, additionally, of cepstral coefficients derived from quasi-pitch-synchronous spectrum. The second representation is based on merely the raw spectrogram. Then we use multilayer-perceptron (MLP) and convolutional neural networks (CNNs) for the task of estimation, and recurrent neural networks (RNNs) and convolutional recurrent neural networks (CRNNs) for the task of tracking.

This paper builds on our earlier work (Dissen and Keshet, 2016), where we used MLP for the task of formant estimation and RNN for formant tracking. The current paper extends our previous work in several directions: (i) Borrowing ideas from state-of-the-art recent speech recognition systems, we propose the use of the CNN architecture with raw spectrograms as the acoustic features for the estimation task, and the use of CRNNs for the tracking task. (ii) We propose a novel technique to adapt the trained MLPs to frequency ranges that were not seen at the training phase, and add experiments on new datasets. (iii) We add more in depth results and analysis for our early findings.

The paper is organized as follows. In Sec. II we describe the formal problem setting. Section III details the different datasets used for evaluation. Section IV describes the different sets of acoustic features used. Section V deals with the problem of formant estimation. Section VI proses a domain adaptation technique for improving estimation performance on new domains. Section VII is focused on the task of formant tracking and its evaluation. We conclude the paper in Sec. VIII.

II. PROBLEM SETTING

We start with a formal problem definition. Formant estimation and tracking is a time-series multiple-regression problem. The input is a sequence of n-dimensional real values, $\bar{\boldsymbol{x}} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$, where $\boldsymbol{x}_t \in \mathcal{X} \subseteq \mathbb{R}^n$ for $t = 1, \ldots, T$, and T is the number of frames in the input. Note that T is known but not fixed, and can be changed from one example to another. In the case of estimation the output is a k-dimensional real vector $\boldsymbol{y} \in \mathcal{Y} \subset \mathbb{R}^k$, where k is the number of formants we wish to estimate (in our case it was three). In the case of tracking the output is a sequence of k-dimensional real values, $\bar{\boldsymbol{y}} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T)$, where $\bar{\boldsymbol{y}} \in \mathcal{Y}^*$.

Our goal is to find a mapping that at least approximately models the dependency between the sequences $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$. Since this is not always possible we will focus on trying to find a mapping between sub-sequences of τ frames, between $(\boldsymbol{x}_{t-\tau}, \ldots, \boldsymbol{x}_t)$ and $(\boldsymbol{y}_{t-\tau}, \ldots, \boldsymbol{y}_t)$ or even just between a single frame $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$. Here we consider the parametric case $\boldsymbol{y}_t = f_\theta(\boldsymbol{x}_t)$, where f_θ is the prediction function parametrized by a set of parameters θ, and $\boldsymbol{y}_t$ in $\mathcal{Y}$ is the prediction. Thus, we wish to estimate the set of parameters, θ, such that $f_\theta(\boldsymbol{x}_t) \approx \boldsymbol{y}_t$ for all t.

The goodness of the fit is quantitatively measured in terms of a loss function. For each input $\boldsymbol{x}_t$ the difference between the predicted value $\hat{\boldsymbol{y}}_t$ and the target value $\boldsymbol{y}_t$, or more simply the "performance," are assessed using a local loss function. Here we use the basic absolute error loss

$$\gamma_t(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t) = |\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t|_1, \tag{1}$$

where $|\cdot|_1$ is the ℓ_1 norm.

Denote by $\bar{\boldsymbol{y}}' = f(\bar{\boldsymbol{x}}) = (\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_T)$ the predicted subsequence of formant frequencies, that is, each $\hat{\boldsymbol{y}}_t$ is a vector of three formants predictions for frame t. The overall performance measure is the local loss function averaged over multiple samples

$$\gamma(\bar{\boldsymbol{y}}, \bar{\boldsymbol{y}}') = \frac{1}{T}\sum_{t=1}^{T} \gamma_t(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t). \tag{2}$$

There are two regression tasks relating to the prediction of formant frequencies. The first task is called *formant estimation* where the input is a single feature vector $\boldsymbol{x}_t$ and the output $\boldsymbol{y}_t$ is the vector of three ($k = 3$) formant frequencies. In this task there are two options: either $\boldsymbol{x}_t$ is a single frame of speech or $\boldsymbol{x}_t$ stands for a series of frames, belonging to the same speech segment. In both cases only one set of formant frequencies need to be predicted. The second task is called *formant tracking*, which essentially is predicting the formants at each time frame t consecutively (can be many time frames for a single vowel) or more formally either $\boldsymbol{y}_t$ given $(\boldsymbol{y}_{t-1}, \boldsymbol{y}_{t-2}, \boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2})$ or $\boldsymbol{y}_t, \boldsymbol{y}_{t-1}, \boldsymbol{y}_{t-2}$ given $\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}$.

III. DATASETS

In order to get reliable results we used three different datasets to evaluate the performance of our system. In this section, we give a short description of each dataset. For consistency with other packages we re-sampled all waveforms to be 16 bit samples with a sampling rate of 16 kHz.

A. VTR

The first being the vocal tract resonance (VTR) corpus introduced by (Deng *et al.*, 2006), this corpus is composed of 538 utterances selected as a representative subset of the well-known and widely used TIMIT corpus (Garofolo *et al.*, 1993). These were split into 346 utterances for the training set and 192 utterances for the test set. These utterances were manually annotated for the first three formants and their bandwidths for every 10 ms frame. The fourth formant was

annotated by the automatic tracking algorithm described in Deng *et al.* (2004), and it is not used here for evaluation.

B. CT

The second dataset (CT) contains segments of acoustic signal from 20 female native English speakers aged 18–22 yr with no history of speech or language deficits. The participants were evenly split between two American English dialects (Northern and Midland). As part of a larger study (Clopper *et al.*, 2002), participants read aloud a list of 991 consonant-to-vowel-to-consonant (CVC) words. This study focused on 39 target words (777 tokens), which did or did not have a lexical contrast between either /ɛ/ vs /ae/ (e.g., dead-dad vs deaf-*daff) or /ɑ/ vs /ɔ/ (e.g., cot-caught vs dock-*dawk). Words with a lexical contrast are referred to as *competitor* items and those without are referred to as *no competitor* items.

C. HGCW

The third dataset consists of data from a laboratory study conducted by Hillenbrand *et al.* (1995; HGCW). It contains segments of acoustic signal from 45 men, 48 women, and 46 10–12-yr-old children (27 boys and 19 girls). 87% of the participants were raised in Michigan, primarily in the southeastern and southwestern parts of the state. The audio recordings contain 12 different vowels (/i, ɪ, ɛ, ae, ɑ, ɔ, ʊ, u, ʌ, ɝ, e, o/) from the words: heed, hid, head, had, hod, hawed, hood, who'd, hud, heard, hayed, hoed.

IV. ACOUSTIC SIGNAL REPRESENTATION

We now turn to describe the acoustic features used as input to the formant estimation and tracking models. As noted earlier, we study two sets of acoustic features. The first set is used in the standard feed-forward and recurrent networks, and it is composed of LPC cepstral coefficients extracted from a range of LPC model orders and, additionally, of cepstral coefficients derived from the quasi-pitch-synchronous spectrum. The second set is used in the convolutional feed-forward and recurrent networks, and it is based on merely the raw spectrogram.

A key assumption is that in the task of estimation, the whole speech segment is considered to be stationary, which mainly holds for monophthongs (pure vowels). In the task of tracking, the speech signal is considered stationary in frames of tens of milliseconds. In the former case the features are extracted from the whole segment, while in the latter case the input signal is divided into frames, and the acoustic features are extracted from each frame. The spacing between frames is 10 ms, and frames are overlapping with analysis windows of 30 ms. As with all processing of this type, we apply a pre-emphasis filter, $H(z) = 1 - 0.97\,z^{-1}$, to the input speech signal and a Hamming window to each frame.

At this phase, two sets of spectral features are extracted. The goal of each of the sets is to parametrize the envelope of the short-time Fourier transform (STFT). The first set is based on LPC analysis, while the second is based on the pitch-synchronous spectra. We now describe in detail and motivate each set of features.

A. LPC-based features

LPC model determines the coefficients of a forward linear predictor by minimizing the prediction error in the least squares sense. Consider a frame of speech of length N denoted by $\bar{s} = (s_1, \ldots, s_N)$, where s_n the nth sample. The LPC model assumes that the speech signal can be approximated as a linear combination of the past p samples

$$\hat{s}_n = \sum_{k=1}^{p} a_k s_{n-k}, \tag{3}$$

where $\boldsymbol{a} = (a_1, \ldots, a_p)$ is a vector of p coefficients. The values of the coefficients $\boldsymbol{a}$ are estimated so as to minimize the mean square error between the signal $\bar{s}$ and the predicted signal $\hat{s} = (\hat{s}_1, \ldots, \hat{s}_N)$,

$$\boldsymbol{a} = \arg\min_{\boldsymbol{a}} \frac{1}{N} \sum_{n=1}^{N} (s_n - \hat{s}_n)^2. \tag{4}$$

Plugging Eq. (3) into Eq. (4), this optimization problem can be solved by a linear equation system.

The spectrum of the LPC model can be interpreted as the envelope of the speech spectrum. The model order p determines how smooth the spectral envelope will be. Low values of p represent the coarse properties of the spectrum, and as p increases, more of the detailed properties are preserved. Beyond some value of p, the details of the spectrum do not reflect only the spectral resonances of the sound, but also the pitch and some noise. Figure 1 illustrates this concept by showing the spectrum of the all-pole filter with values of p ranging from 8 to 17. A disadvantage of this method is that if p is not well chosen (i.e., to match the number of resonance present in the speech), then the resulted LPC spectrum is not as accurate as desired (Birch *et al.*, 1988).

Our first set of acoustic features are based on the LPC model. Instead of using a single value of the number of LPC coefficients, we used a range of values between 8 and 17. This way the classifier can combine or filter out information from different model resolutions. More specifically, in our setting after applying pre-emphasize and windowing, the LPC coefficients for each value of p were extracted using the autocorrelation method, where the Levinson-Durbin recursion was used for the autocorrelation matrix inversion and the FFT for the autocorrelation computation.

The final processing stage is to convert the LPC spectra to cepstral coefficients. This is done efficiently by the method proposed by Atal (1974). Denoted by $\boldsymbol{c} = (c_1, \ldots, c_n)$ is the vector of the cepstral coefficients where $n > p$

$$c_m = \begin{cases} a_m + \sum_{k=1}^{m-1} \left(1 - \frac{k}{m}\right) a_k c_{m-k}, & 1 \le m \le p \\ \sum_{k=1}^{p} \left(1 - \frac{k}{m}\right) a_k c_{m-k}, & p < m \le n. \end{cases} \tag{5}$$

We tried different values for n and found that $n = 30$ gave reasonable results.

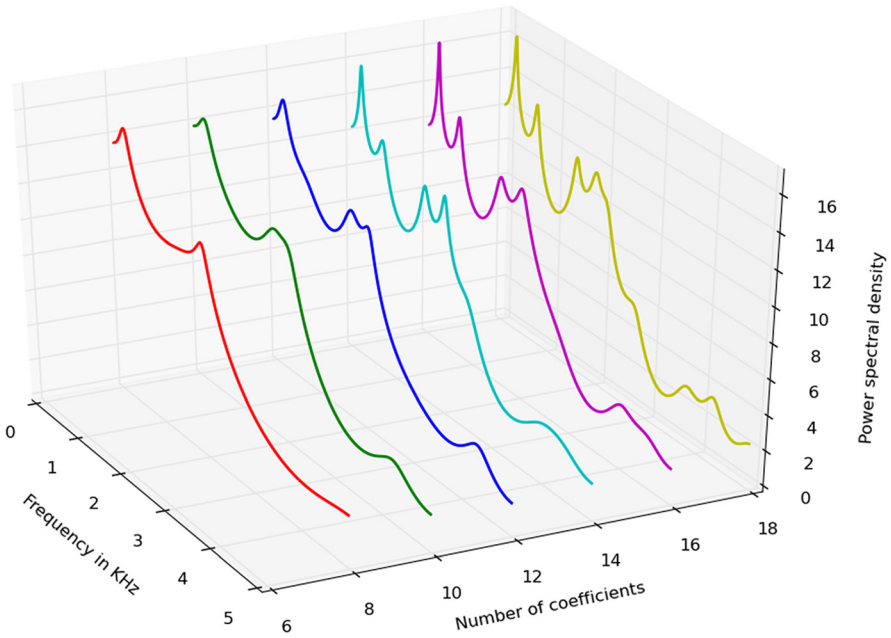

FIG. 1. (Color online) Long-term LPC spectrum of the whole vowel /u/ produced for 262 ms for values of p 8, 10, 12, 14, 16, and 17.

B. Pitch-synchronous spectrum-based features

The spectrum of a periodic speech signal is known to exhibit an impulse train structure located at multiples of the fundamental frequency period. A major concern when using the spectrum directly for locating the formants is that the resonance peaks might fall between two fundamental frequency periods, and then they are not "visible." The LPC model estimates the spectrum envelope to overcome this problem. Another concern in formant measurements is the bias by the particular fundamental frequency used to excite the formants (Shadle *et al.*, 2016).

In order to overcome these problems in our estimation we added another feature set, namely the *pitch synchronous spectrum* (Medan and Yair, 1989). According to this method the discrete Fourier transform (DFT) is taken over frames the size of the instantaneous pitch estimation.[1]

One of the main problems of this method is the need of a very accurate pitch estimator. Another issue is how to implement the method in the case of formant estimation when the input is a speech segment that represents a single vowel, which typically spans a few pitch periods, and the pitch in not fixed along the segment. We found out that using a pitch period that is close enough to its exact value is suitable in our application. This can be observed in Fig. 2, where the quasi-pitch-synchronous fast Fourier transform (FFT) for different values of pitch periods are depicted. It can be seen that except for extreme cases, the peaks of the spectra are well-smoothed and clearly defined.

In our implementation we extract quasi-pitch synchronous spectrum similar to (Medan and Yair, 1989). For the task of formant estimation we use the median pitch computed in frames of 10 ms along the input segment, and use the average spectra.

At the final stage, the resulting quasi-pitch-synchronous spectrum is converted to cepstral coefficients by applying log compression and then discrete cosine transform (DCT).

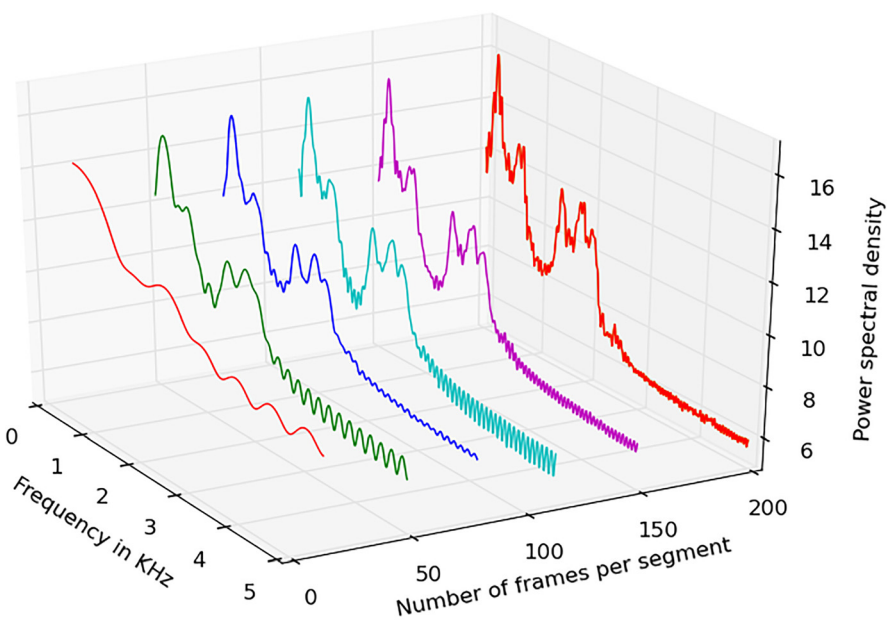

FIG. 2. (Color online) Quasi-pitch-synchronous averaged spectra of the vowel /u/ produced for 262 ms with different values of pitch. The true value of the pitch was 123.4 frames.

We use the first 50 DCT coefficients as our second set of features.

Overall, the first set of acoustic features is a vector is composed of cepstral LPC features for 10 values of p (order sizes) and the 50 coefficients of the quasi-pitch-synchronous spectrum

$$\boldsymbol{x}_t = (\mathbf{c}_t^8, \mathbf{c}_t^9, \ldots, \mathbf{c}_t^{17}, \mathbf{z}_t),$$

where $\mathbf{c}_t^p$ is the cepstral LPC coefficient vector of order p of the tth frame [whose elements are defined in Eq. (5)], and $\mathbf{z}_t$ are the coefficients of quasi-pitch-synchronous spectrum of the tth frame.

C. Spectrogram as features

While the first set of acoustic features is based on methods that highlight the peaks in the spectrum, the second set is merely the raw spectrum. It has been shown that CNNs applied to the spectral input features prior to any other processing improve automatic speech recognition performance (Abdel-Hamid *et al.*, 2012; Amodei *et al.*, 2016; Chan *et al.*, 2016; Hannun *et al.*, 2014; Sainath *et al.*, 2013). Following these models, we compute spectrograms of 100 Hz linearly spaced log filter banks and an energy term. The filter banks are computed over windows of 20 ms strided by 5 ms. In the task of estimation, we normalized the spectrogram by removing the DC component, simply put, we subtracted the mean of the whole spectrogram. We then used only a part of the spectrogram: frequencies until 5.5 kHz (55 filters) and time-span of 250 ms (50 frames), where if the vowel was shorter than 50 frames we zero padded it. The acoustic features are actually *images* of sizes 55×50.

In the task of tracking we normalize for each frequency band separately. We used the same frequency limit (55 filters), whereas the frames are fed one-by-one to the CRNN as overlapping images.

V. FORMANT ESTIMATION: FEED-FORWARD NETWORKS

Recall that in the task of estimation, the input is a speech segment representing a single vowel and the goal is to extract the first three formants. The networks are multiple regression networks, which predict all formants together and use the common loss function mean absolute error $\gamma(\boldsymbol{y}, \boldsymbol{y}) = |\boldsymbol{y} - \hat{\boldsymbol{y}}|_1$. We use neural nets over multiple regression models given that they have been shown to outperform multiple regression models (Brey *et al.*, 1996; Lathuilière *et al.*, 2018).

A. Network architectures

For the task of estimation we use two types of feed-forward networks based on the feature set used. An MLP was used for the spectral envelop-based data and a CNN for the raw spectrogram data.

1. MLP

We start with the best-known and most widely used form, the MLP, used for the LPC-based data. MLP is a mathematical function with a set of parameters. During the training phase, the parameters of the function are estimated so as to maximize the mean absolute error. The function is composed of a series of *layers*, where each layer gets as input the previous layer output, converts it linearly, and applies a nonlinear transformation called *activation*. The input to the first layer is the acoustic features, and the output of the last layer is the three predicted formant values. The nonlinearity is not applied to the last layer. All the layers but the first and the last are considered as *hidden* layers.

The input of the network is a vector of 350 features (30 DCT features for each of the 10 LPC model sizes plus 50 features of the quasi-pitch-synchronous spectrum), and the output is a vector of the three annotated formants. The network has three hidden layers with 1024, 512, and 256 neurons, and all of them are fully connected. The activations for said layers are sigmoid functions. The network was trained using Adagrad (Duchi *et al.*, 2011) to minimize the mean absolute error or the absolute difference between the predicted and true formant frequencies with weights randomly initialized. The training of the networks weights was done as regression rather than classification. The network predicts all three formants simultaneously to exploit inter-formant constraints.

2. CNN

CNN is another type of feed-forward neural network. The CNN is a variation of MLP and it is composed of convolutional layers, pooling layers, and fully connected layers. The convolutional layer applies a convolution operation to its input, passing the result to the next layer, where the filter of the convolution is estimated during the training. The pooling layer combines several outputs of convolutional layer into a single node, e.g., max pooling layer of size four outputs the maximum value for every four inputs. Convolutional layers alternate with max pooling layers, mimicking the nature of complex and simple cells in the mammalian visual cortex (Hubel and Wiesel, 1968). A CNN consists of one or more pairs of convolution and max pooling layers and finally ends with a fully connected neural network.

The input to the CNN are 55×50 spectrogram "images." The network has four two-dimensional (2D) convolutional layers with the rectified linear unit (ReLU) activation and 2D max-pooling after which there are two fully connected feed-forward layers with ReLU and linear activation functions, respectively, the output of the latter being the three formant frequencies. All convolution windows were 3×3 and all max-pooling windows were 2×2. As before the network was trained using Adagrad to minimize the mean absolute error.

B. Experimental results

The estimation algorithm was applied only to vowels (monophthongs and diphthongs). We used the whole vowel segments of the VTR, CT, and HGCW datasets. The test sets

TABLE I. Using different types of LPC-based features for the estimation of formant frequencies of whole vowels of the VTR corpus. Boldface indicates the best result in that category.

Feature set	F_1	F_2	F_3
LPC, $p=12$	59	123	179
LPC, $p=\{8-17\}$	60	86	110
Quasi-pitch-sync	51	115	164
LPC, $p=\{8-17\}$ + quasi-pitch-sync	**45**	**82**	**106**

of each dataset were chosen as follows. The VTR dataset has a predetermined test set. The CT and HGCW sets were split into train and test sets using two-thirds for train and one-third for test. This was done by removing one-third of the speakers from each dataset so the speakers of the training set and the test set were different. The formant annotation was at the 10 ms resolution. In the estimation task, we used the average formants along the vowel segments. Other types of annotations are reported subsequently. Results are given in mean absolute differences in Hz, unless stated otherwise. We use mean absolute difference as our main method of evaluation as we believe this most accurately describes the type of error we are trying to minimize for actual usage.

Table I shows the influence of the different spectral envelope and LPC-based feature sets trained with MLP. It can be seen that using different LPC model orders improves the performance on F_2 and F_3, and the performance on F_1 improves with the quasi-pitch-synchronous feature set. Results are reported on the VTR test set.

From now on, we denote by *DeepFormants* the results achieved by the MLP network with the set of features reported in the last row of Table I (LPC, $p=\{8–17\}$ + quasi-pitch-sync). This is the system that was described in our previous work (Dissen and Keshet, 2016).

Next, we compared the performance of CNNs with raw spectrogram to the performance achieved by *DeepFormants*. As a baseline we compared both our results to those of Praat, a popular tool in phonetic research (Boersma and Weenink, 2002). Formants were extracted from Praat using Burg's method with a maximum formant value of 5.5 kHz, a window length of 30 ms, and a pre-emphasis from 50 Hz. The results of our system and those of Praat's on the test set are shown in Table II. As seen in Table II, we have achieved better results across the board over Praat when comparing our respective estimations to the manually annotated reference with the CNN outperforming the LPC-based system.

TABLE II. Estimation of formant frequencies of whole vowels using *DeepFormants* (Dissen and Keshet, 2016), CNN, and Praat on the VTR corpus. Results are in respect to absolute error. Boldface indicates the best result in that category.

	Method	F_1	F_2	F_3
Mean	CNN	**45**	**65**	**94**
	DeepFormants	**45**	82	106
	Praat	75	115	151
Standard deviation	CNN	**41**	**66**	**108**
	DeepFormants	**41**	74	120
	Praat	120	138	205
Median	CNN	35	**48**	**63**
	DeepFormants	**34**	63	73
	Praat	48	78	88
Largest single error	CNN	**444**	760	**1349**
	DeepFormants	476	**752**	1516
	Praat	1611	1711	1633

TABLE III. Results measured by RMSE and AVG. Boldface indicates the best result in that category.

	Method	F_1	F_2	F_3
RMSE	CNN	62	**95**	**146**
	DeepFormants	**61**	111	160
	Praat	142	180	255
AVG	CNN	1	−16	−7
	DeepFormants	4	−2	−14
	Praat	62	26	45

Next, we look at two additional evaluations of our results following the analysis presented by Schiel and Zitzelsberger (2018). The first evaluation of our results is the loss in root-mean-square error (RMSE) and the second is AVG, which is the average difference between reference and predictions calculated to indicate if there is an imbalance in the direction of the errors. Results using RMSE and AVG measurements are seen in Table III. Here again, as measured by RMSE, the neural network-based estimators outperform Praat for all formants with the CNN giving better results for F_2 and F_3, directly correlating with the results measured using mean absolute error. Looking at averaged AVG measures in Table III, one can see that the AVG errors for *DeepFormants* and CNNs are almost perfectly balanced with a small tendency to underestimate F_2 and F_3, while Praat tended to overestimate all three formant frequencies.

Recall that in the estimation task the input is a whole vowel span over hundreds of milliseconds, but the formant frequencies are annotated every 10 ms. We now turn to analyze how the selection of the target formant frequencies influence predictions. We compared three types of methods to select the target frequencies that would be used in training: (i) average frequencies along the vowel segment, (ii) the frequency at the center of the vowel, and (iii) the average over 30 ms of the stable part of the vowel. The results are given in Table IV. We can see that the results do not dramatically change with different annotation methods. In the source code the user has the option to select which method to use at inference time.

TABLE IV. The mean results of the estimator trained on different methods to select the target frequencies.

Annotation	Method	F_1	F_2	F_3
Whole vowel mean	CNN	45	65	94
	DeepFormants	45	82	106
Vowel center	CNN	52	77	105
	DeepFormants	51	86	111
Stable part of the vowel	CNN	49	74	102
	DeepFormants	50	85	109

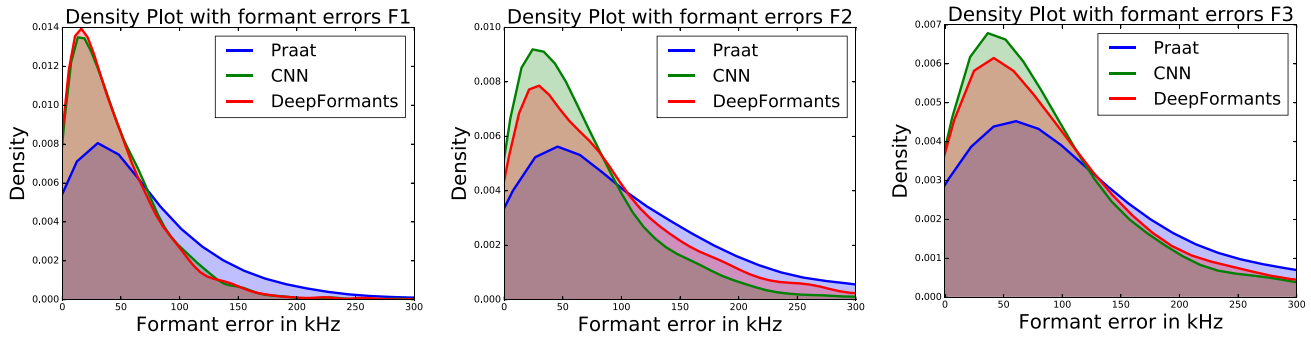

FIG. 3. (Color online) Density plots of each formant error across the estimators on the VTR corpus: Praat, *DeepFormants* (Dissen and Keshet, 2016), and CNN. The *y* axis is the density and the *x* axis is the error in kHz.

To better visualize how the errors of each method are spread, we presented the error distributions for each formant in Fig. 3. The graphs were cut off at the 300 Hz error mark. Figure 3 shows that the majority of errors *DeepFormants* and CNNs made are heavily concentrated around the median error, whereas Praat's errors are more spread out. Analysis of the predictions with the largest inaccuracies show that they broadly fall into three categories: (i) There are annotation errors and the system indeed did classify them accurately; (ii) the vowel segment was very short (<35 ms); and (iii) ambiguous spectrograms, where both the manual annotation and the predicted value can be considered as correct.

Finally, following the analysis presented in Schiel and Zitzelsberger (2018), we evaluate the performance for each gender as shown in Table V. The neural network estimators perform better than Praat in every category. As seen here across all estimators the F_1 performance was better for females although our estimators stay within 10 Hz, while Praat has a 35 Hz performance disparity. The second formant F_2 errors were mostly the same across genders with a slight preference for males primarily seen in the 9 Hz mean disparity by *DeepFormants*. Similarly the third formant F_3 performance seems to be indifferent to gender, save for Praat, which again has a 35 Hz disparity, this time, toward females. Overall it seems that the neural-network-based estimators do not have a significant bias toward either gender.

We also take a brief look into the differences between adults and children. Shown in Table VI are the results of the CNNs trained on VTR and tested on the HGCW dataset, with and without children in training, split into children and adults. Table VI demonstrates that adding children into the training significantly improves the results on this group, although the performance is still much worse for children's F_3 as opposed to adults. Praat was used to estimate format frequencies of the HGCW dataset so it produced no errors.

TABLE V. Mean absolute error and standard deviation results split by gender.

	Method	F_1	F_2	F_3
Male	CNN	49 ± 44	67 ± 68	97 ± 108
	DeepFormants	48 ± 43	79 ± 71	106 ± 115
	Praat	87 ± 142	115 ± 155	163 ± 228
Female	CNN	40 ± 34	69 ± 62	97 ± 108
	DeepFormants	38 ± 37	88 ± 79	107 ± 129
	Praat	52 ± 48	116 ± 95	128 ± 149

VI. FORMANT ESTIMATION: DOMAIN ADAPTATION NETWORKS

The networks trained on the VTR training set yield good results on the VTR test set. However, when the same model is applied to other datasets, especially with format frequency ranges that were not seen at training time, we found a significant performance degradation. This is probably due to lack of large amounts of data that are needed to train the deep network models. The goal of this study is to train a single network that can successfully help the trained models to handle different frequency ranges and adapt other datasets. Naturally, we do not assume that at train or test time the domain identity of a given input is revealed to the network. Rather we want the network to learn to adapt its predictions based on the original features. We assume that a network was already trained on a corpus of labeled examples (the VTR corpus). We call this network the *pre-trained network*. We also assume the number of training examples of the adapted corpus is relatively small.

A. Architecture

The domain adaptation network includes two new elements to the pre-trained network. First we use the output layer of the pre-trained network as input to another linear layer that produces a new formant estimation. We also add a selection neuron *s* whose input is the acoustic features, i.e., the same input of the original pre-trained network. This selection neuron controls the amount of network adaptation that should be applied to each given input.

TABLE VI. Mean absolute error and standard deviation of CNNs trained on VTR with and without children and tested on adults and children from HGCW.

	Training Method	F_1	F_2	F_3
Adults	VTR	44 ± 44	110 ± 105	161 ± 365
	VTR with children	28 ± 26	53 ± 52	121 ± 361
Children	VTR	71 ± 87	240 ± 278	397 ± 549
	VTR with children	34 ± 40	89 ± 139	253 ± 574

The control element s is a nonlinear function of the input feature vector $\boldsymbol{x}$ and is implemented by a linear operation followed by a sigmoid activation function. Namely,

$$s(\boldsymbol{x}) = \sigma(\boldsymbol{w}_s \cdot \boldsymbol{x} + b_s), \tag{6}$$

such that $\boldsymbol{x}$ is the input vector, and $\boldsymbol{w}_s$ and b_s are the network parameters that are learned at adaptation phase.

Denote the formant output of the pre-trained network by $\hat{y}_1, \ldots, \hat{y}_3$, and the new formant estimation by $\tilde{y}_1, \ldots, \tilde{y}_3$. The new formant estimation is done as follows:

$$\tilde{y}_i = \sum_{j=1}^{3} w_{ij}\hat{y}_j + b_i + v_i s_i(\boldsymbol{x}) \tag{7}$$

where w_{ij} and b_i are the parameters of the additional linear layer and v_i is a multiplicative term that defines the contribution of the dataset control element $s_i(\boldsymbol{x})$ to the estimation of the ith formant. A scheme of the domain adaptation network is depicted in Fig. 4.

One of the advantages of training the network in two steps is that there is only a small number of adaptation parameters so large amounts of data from each dataset are not necessary. Hence, domains with limited labeled data, as in our case, can still be used to learn a good formant estimator.

B. Experimental results

We trained the adaptation network on the three datasets from scratch starting by using a random initialization of the network parameters. We also trained the network in two steps. First, we trained the pre-trained network on the VTR dataset. Next, we froze the parameters of the pre-trained network and only estimated the adaptation parameters in Eqs. (7) and (6) using the VTR, CT, and HCGW datasets. This two step training procedure ensures that the pre-trained network is responsible for the core formant estimation, and the adaptation part of the network is responsible solely for adapting the formant estimation to the conditions of the specific input vector.

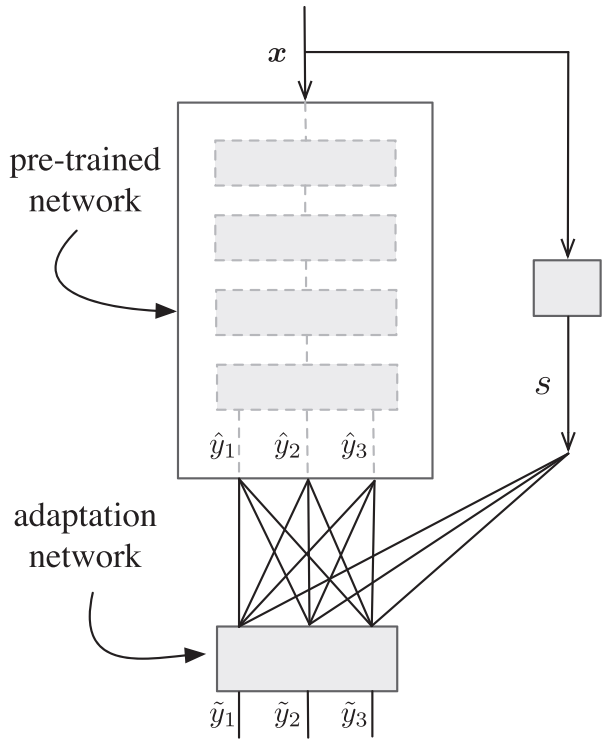

FIG. 4. A scheme of the domain adaptation formant estimation network.

Here, we present the results of the domain adaptation network results on the three databases described in Sec. III. Recall that the CT corpus includes mostly young women and the HCGW corpus includes children—namely, both datasets gave formant frequency ranges very different than that of VTR. We compared the two types of training methods to the WaveSurfer program (Sjölander and Beskow, 2000), a popular tool in phonetic research. The reason we are not using Praat as a baseline is due to the fact that some of the databases used Praat to give initial estimations to the formant frequencies, and then fixed the mistakes so the annotations are biased toward Praat. The results for the pre-trained model of MLP are shown in Table VII, where the loss is the mean absolute difference in Hz. Note that the CT dataset was only annotated for the first and second formants, hence, the third formant was not evaluated.

As seen in Table VII, we achieved better results across the board over WaveSurfer when comparing our respective estimations to the manually annotated reference. The domain adaptation network shows improvement over *DeepFormants* in both the CT and HGCW datasets with no significant drop off in accuracy on the VTR dataset. These results show the advantage of the proposed network architecture over standard networks based on fully connected layers.

When comparing these results to separate networks trained and tested on each of the databases, i.e., training a model with data from the CT dataset and then testing on the CT test set and another model trained and tested on the HGCW dataset in the same manner and so on for the VTR dataset, we obtained comparable results. Hence, there is no need for multiple models for each domain, this single network can separate between the speaker and speech domains and adjust its estimations accordingly.

Next, we demonstrate the need for the two step training procedure proposed in this study. Table VIII shows the formant estimation results of two training strategies of the

TABLE VII. Estimation of formant frequencies using deep learning with and without domain adaptation and compared to Wavesurfer. Boldface indicates the best result in that category.

Dataset	Method	F_1	F_2	F_3
VTR	WaveSurfer	70	96	154
	DeepFormants	**45**	**82**	106
	Domain adaptation	50	86	**104**
HGCW	WaveSurfer	68	190	182
	DeepFormants	71	160	131
	Domain Adaptation	**36**	**100**	116
CT	Wavesurfer	128	181	—
	DeepFormants	228	168	—
	Domain adaptation	**103**	**157**	—

TABLE VIII. Results of formant estimation for two possible training procedures of the proposed domain adaptation network. Boldface indicates the best result in that category.

Dataset	training method	F_1	F_2	F_3
VTR	Joint training	96	279	283
	Two step training	**50**	**86**	**104**
HGCW	Joint training	71	119	129
	Two step training	**36**	**100**	**116**
CT	Joint training	**70**	166	—
	Two step training	103	**157**	—

TABLE IX. Results of CNNs formant estimation with training on the VTR dataset alone and with data from all three datasets. Boldface indicates the best result in that category.

Dataset	training method	F_1	F_2	F_3
VTR	Joint training	**45**	68	**91**
	VTR training	**45**	**65**	94
HGCW	Joint training	**31**	**68**	**176**
	VTR training	55	163	259
CT	Joint training	**81**	**100**	—
	VTR training	165	159	—

domain adaptation network. In Table VIII we compared the results of the two step training to the results of a network with the same topology but trained jointly on all three datasets. As can be seen from Table VIII, other than for the first formant in the CT dataset the accuracy is greatly diminished across all other data. Moreover, adding the selection layer does not improve results over training a network identical to the pre-trained network but using all three datasets during training.

When using a CNN for estimation there is also some degree of overfitting, but in this case there is no need for two step training as training the model with a combination of all datasets gives it the ability to estimate accurately over all databases without corrupting the results for any other data domain. We can see in Table IX that the results improve dramatically for the CT and HGCW datasets without changing the VTR results as a result of combined training over training with the VTR dataset alone.

To better visualize what the selection neuron has learned, we show the histograms of the domain parameters activation values s for each dataset. The histogram axes are the number of examples in each of the ten buckets of activations between 0 and 1 (which is the output range of a sigmoid function). As seen in the right-most histograms in Fig. 5, the s values for the VTR database are almost exclusively concentrated in the same area, showing that the network automatically found that no adaptation is needed for data from the VTR dataset. This coincides with the fact that the original network was trained on the VTR dataset. In contrast, predictions from the HGCW dataset needed to be corrected occasionally, and predictions from the CT dataset consistently needed to be corrected, as seen by the variance in s values.

VII. FORMANT TRACKING

In this section we describe the network architectures that are used for formant tracking, where the input is a series of speech frames and the goal is to extract the corresponding series of values of the first three formants.

A. Recurrent architectures

RNN is a type of neural network that is a powerful sequence learner. In particular, the long short-term memory (LSTM) architecture has shown to provide excellent modeling of sequential data such as language, music, facial expressions, and speech (Graves *et al.*, 2013). Because in the tracking task our data are sequential, we will use RNNs that take into account temporal structure as opposed to standard feed-forward neural networks, which only look at each individual frame as an independent unit.

The idea behind RNNs is to make use of sequential information. In a traditional neural network we assume that all inputs (and outputs) are independent of each other. RNNs are called recurrent because they perform the same task for every element of a sequence with the output being dependent on the previous computations. Another way to think about RNNs is that they have a *memory* that captures information about what has been calculated so far. In theory, RNNs can make use of information in arbitrarily long sequences, but in practice they are limited to looking back only a few steps.

LSTMs do not have a fundamentally different architecture from RNNs, but they use a different function to compute the hidden state. The memory in LSTMs is called a *cell*. Internally these cells decide what to keep in (and what to erase from) memory by combining the previous state, the current memory, and the input. In turns out that these types

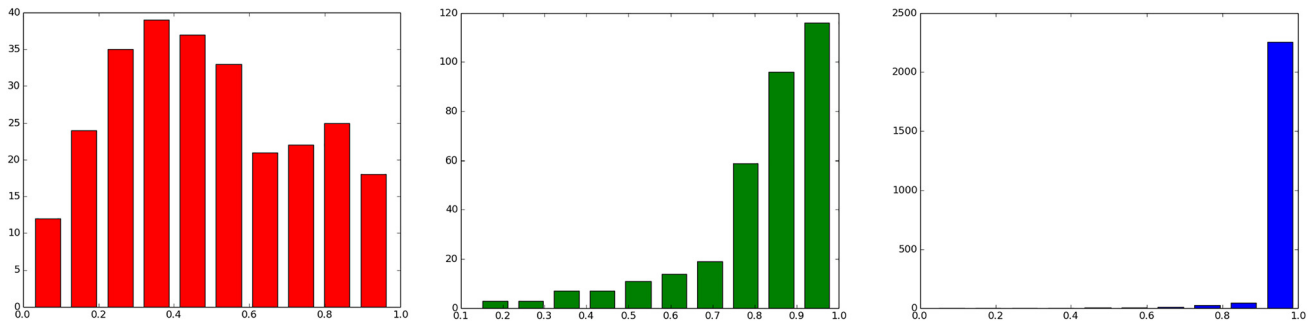

FIG. 5. (Color online) Histograms of domain bias activations for the datasets CT (left), HGCW (middle), and VTR (right). The x axis is the histogram buckets between 0 and 1 and the y axis is the number of vowels/samples in each bucket.

TABLE X. Tracking errors of on broad phone classes measured by mean absolute difference in Hz. Boldface indicates the best result in that category.

	Inter-labler			Wave Surfer			Praat			MSR (Deng *et al.*, 2004)			Deep Formants			CNN		
	F_1	F_2	F_3	F_1	F_2	F_3	F_1	F_2	F_3	F_1	F_2	F_3	F_1	F_2	F_3	F_1	F_2	F_3
Vowels	55	69	84	70	94	154	130 ± 210	230 ± 264	267 ± 321	64	105	125	54 ± 53	81 ± 88	112 ± 136	**53** ± 52	**72** ± 74	**108** ± 128 s
Semivowels	68	80	103	89	126	222	136 ± 163	295 ± 352	334 ± 375	83	122	**154**	**67** ± 63	114 ± 128	168 ± 184	68 ± 62	**111** ± 143	160 ± 187
Nasal	75	112	106	96	229	239	219 ± 258	409 ± 390	381 ± 405	67	**120**	**112**	**66** ± 67	175 ± 186	151 ± 140	69 ± 66	191 ± 208	158 ± 152
Fricatives	91	113	125	209	263	439	564 ± 367	593 ± 421	700 ± 441	**129**	**108**	**131**	131 ± 114	135 ± 134	159 ± 164	139 ± 118	142 ± 143	167 ± 156
Affricates	89	118	135	292	407	390	730 ± 473	515 ± 356	583 ± 377	**141**	**129**	**149**	164 ± 140	162 ± 102	189 ± 170	174 ± 146	173 ± 144	195 ± 164
Stops	91	110	116	168	210	286	258 ± 276	270 ± 290	351 ± 346	130	**113**	**119**	131 ± 103	135 ± 116	168 ± 157	**123** ± 102	135 ± 149	170 ± 168

of units are very efficient at capturing long-term dependencies. Here we define the input as $(\boldsymbol{x}_{t-\tau}, \ldots, \boldsymbol{x}_t, \boldsymbol{y}_{t-\tau}, \ldots, \boldsymbol{y}_{t-1})$, the predicted label is defined as $\hat{\boldsymbol{y}}_t$, and the previous target labels will be injected into the hidden layers.

1. RNN

For the LPC-based tracker, we use a RNN consisting of an input layer with 350 features as in the estimation task. In addition to these features extracted from the current segment of speech on account of the fact that this is a RNN, the predictions and features of the previous speech segment (i.e., temporal context) are taken into account when predicting the current segments formants. Next are two LSTM layers with 512 and 256 neurons, a time distributed fully connected layer with 256 neurons, and an output layer consisting of the 3 formant frequencies. As in the estimation network, the activations were all sigmoid, the optimizer was Adagrad, and the function to minimize was the mean average error.

2. CRNN

For the CRNN, we use 55×50 spectrograms as input as in the estimation CNN. The first three layers of the network are 2D convolutional LSTMs as described in Xingjian *et al.* (2015) Next the network has two time distributed 2D convolutional and max-pooling layers. Again, here, we use 3×3 window sizes for convolutions and ReLU as the activation function. Finally, we have two fully connected feed-forward layers with ReLU and linear activation functions, the output of the latter being the three formant frequencies. As in the estimation network, the activations were all sigmoid, the optimizer was Adagrad, and the function to minimize was the mean average error.

B. Experimental results

We now present the results for our tracking models. We evaluated the model on whole spoken utterances of VTR. We compared our results to Praat, WaveSurfer, and to the results obtained in Deng *et al.* (2006) from the MSR tracking algorithm. Table X shows the accuracy in mean absolute difference in Hz for each broad phonetic class. The inter-labeler variation is also presented in Table X for reference taken from Deng *et al.* (2006).

Both the LPC-based RNN and the spectrogram-based RCNN outperform Praat and WaveSurfer in every category and, compared to MSR, our models show higher precision with vowels and semivowels while MSR reports higher precision with nasals, fricatives, affricates, and stops. It is worth mentioning, though, that the phone class where formants are most indicative of speech phenomena is vowels. The higher precision reported by MSR in consonant phone classes is most likely due to the fact that the database obtained its initial trajectory labels from MSR and was then manually corrected (Deng *et al.*, 2006), so in phonemes without clear formants (i.e., consonants) there is a natural bias toward the trajectories labeled by MSR.

Also noted is the fact that the CNN performed better than LPC-based RNN only in vowels. This is most likely due to the fact that the LPC spectra forces the existence of clear peaks even in consonants, whereas the spectrogram of a consonant will not have clearly defined peaks.

In addition, the observed mean differences between our automated measurements and the manually annotated measurements are comparable in size to the generally acknowledged uncertainty in formant frequency estimation demonstrated in our dataset by the degree of inconsistency between different labelers in Table X and to the perceptual difference limens found in Mermelstein (1978) such that it is doubtful that higher accuracy can be achieved with automated tools seeing as manual annotation cannot.

We also examined the errors of the algorithms when limiting the error-counting regions to only the consonant-to-vowel (CV) and vowel-to-consonant (VC) transitions. The

TABLE XI. Same as for Table X except for the focus on temporal regions of CV transitions and VC transitions. Boldface indicates the best result in that category.

	Wave Surfer			Praat			MSR (Deng *et al.*, 2004)			Deep Formants			CNN		
	F_1	F_2	F_3	F_1	F_2	F_3	F_1	F_2	F_3	F_1	F_2	F_3	F_1	F_2	F_3
CV transitions	156	192	273	169 ± 200	225 ± 251	261 ± 286	**106**	**101**	**119**	110 ± 102	142 ± 135	165 ± 167	108 ± 101	142 ± 158	166 ± 171
VC transitions	59	88	157	344 ± 370	355 ± 358	495 ± 461	**48**	92	120	53 ± 54	**80** ± 97	111 ± 137	54 ± 54	81 ± 93	**110** ± 126

TABLE XII. Formant tracking performance of KARMA and deep learning in terms of the RMSE per formant. RMSE is only computed over speech-labeled frames. Boldface indicates the best result in that category.

Method	F_1	F_2	F_3	Overall
KARMA (Mehta *et al.*, 2012)	**114**	226	320	220
DeepFormants	118	**169**	**204**	**163**
RCNN	127	180	213	173
Praat	185	254	303	247
WaveSurfer	170	276	383	276

transition regions are fixed to be six frames, with three frames to the left and three frames to the right of CV or VC boundaries defined in the TIMIT database. The detailed results are listed in Table XI.

Results from other works on the VTR dataset include Mehta *et al.* (2012), and compared to his results seen in Table XII in both models, our precision is on par for the first formant but greatly improved for the second and third formants. Error is measured in RMSE.

VIII. CONCLUSIONS

Accurate models for formant estimation and tracking were presented with the former surpassing existing automated systems' accuracy and the latter within the margins of human inconsistencies. We proposed a formant estimation deep-learning architecture that achieves state-of-the-art results across several speech and speaker domains, which are very different in nature. We also proposed a training scheme that validates the claim that each component of the network is indeed responsible for the task it was designed to do, either formant estimation or domain adaptation.

Deep learning has proved to be a viable option for automated formant estimation tasks, and if more annotated data are introduced, we project higher accuracy models can be trained as analysis of the phonemes with the least accuracy on average seems to show that they were the ones that were represented the least in the databases.

In this paper we have demonstrated automated formant tracking and estimation tools that are ready to be added to the methods that socio-linguists use to analyze acoustic data. The tools are publicly available.[2]

ACKNOWLEDGMENTS

This research was supported by the MAGNET program of the Israeli Innovation Authority. We would like to thank Cynthia Clopper for allowing us to use their dataset.

[1]In the context of estimation of the fundamental frequency, we will use the term *pitch estimation*.

[2]https://github.com/MLSpeech/DeepFormants (Last viewed 1/25/2019).

Abdel-Hamid, O., Mohamed, A.-r., Jiang, H., and Penn, G. (**2012**). "Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4277–4280.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., Engel, J., Fang, W., Fan, L., Fougner, C., Gao, L., Gong, C., Hannun, A., Han, T., Johannes, L., Jiang, B., Ju, C., Jun, B., LeGresley, P., Lin, L., Liu, J., Liu, Y., Li, W., Li, X., Ma, D., Narang, S., Ng, A., Ozair, S., Peng, Y., Prenger, R., Qian, S., Quan, Z., Raiman, J., Rao, V., Satheesh, S., Seetapun, D., Sengupta, S., Srinet, K., Sriram, A., Tang, H., Tang, L., Wang, C., Wang, J., Wang, K., Wang, Y., Wang, Z., Wang, Z., Wu, S., Wei, L., Xiao, B., Xie, W., Xie, Y., Yogatama, D., Yuan, B., Zhan, J., and Zhu, Z. (**2016**). "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *International Conference on Machine Learning*, pp. 173–182.

Anand, M. J., Guruprasad, S., and Yegnanarayana, B. (**2006**). "Extracting formants from short segments of speech using group delay functions," in *Proceeding of Interspeech*.

ANSI (**1994**). S1.1-1994, *American National Standard Acoustical Terminology* (Acoustical Society of America, Melville, NY).

Atal, B. S. (**1974**). "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," J. Acoust. Soc. Am. **55**(6), 1304–1312.

Birch, G. E., Lawrence, P., Lind, J. C., and Hare, R. D. (**1988**). "Application of prewhitening to AR spectral estimation of EEG," IEEE Trans. Biomed. Eng. **35**(8), 640–645.

Boersma, P., and Weenink, D. (**2002**). "Praat, a system for doing phonetics by computer," Glot Int. **5**(9/10), 341–345.

Bradlow, A. R. (**1995**). "A comparative acoustic study of English and Spanish vowels," J. Acoust. Soc. Am. **97**(3), 1916–1924.

Bradlow, A. (**2002**). "Confluent talker- and listener-related forces in clear speech production," in *Laboratory Phonology 7*, edited by C. Gussenhoven and N. Warner (Mouton de Gruyter, Berlin), pp. 241–273.

Brey, T., Jarre-Teichmann, A., and Borlich, O. (**1996**). "Artificial neural network versus multiple linear regression: Predicting p/b ratios from empirical data," Mar. Ecol. Prog. Ser. **140**, 251–256.

Cadzow, J. A. (**1982**). "Spectral estimation: An overdetermined rational model equation approach," Proc. IEEE **70**(9), 907–939.

Chan, W., Jaitly, N., Le, Q., and Vinyals, O. (**2016**). "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964.

Clopper, C. G., and Bradlow, A. R. (**2009**). "Free classification of American English dialects by native and non-native listeners," J. Phon. **37**(4), 436–451.

Clopper, C. G., Carter, A. K., Dillon, C. M., Hernandez, L. R., Pisoni, D. B., Clarke, C. M., Harnsberger, J. D., and Herman, R. (**2002**). "The Indiana speech project: An overview of the development of a multi-talker multi-dialect speech corpus," Bloomington, Speech Research Laboratory, Indiana University, Research on Speech Perception Progress Report No. 25, pp. 367–380.

Clopper, C. G., and Pierrehumbert, J. B. (**2008**). "Effects of semantic predictability and regional dialect on vowel space reduction," J. Acoust. Soc. Am. **124**(3), 1682–1688.

Clopper, C. G., and Tamati, T. N. (**2014**). "Effects of local lexical competition and regional dialect on vowel production," J. Acoust. Soc. Am. **136**(1), 1–4.

Deng, L., Cui, X., Pruvenok, R., Chen, Y., Momen, S., and Alwan, A. (**2006**). "A database of vocal tract resonance trajectories for research in speech processing," in *Proceedings of 2006 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. I–I.

Deng, L., and Geisler, C. D. (**1987**). "A composite auditory model for processing speech sounds," J. Acoust. Soc. Am. **82**(6), 2001–2012.

Deng, L., Lee, L. J., Attias, H., and Acero, A. (**2004**). "A structured speech model with continuous hidden dynamics and prediction-residual training for tracking vocal tract resonances," in *Proceedings of 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing. (ICASSP'04)*, Vol. 1, pp. I–557.

Deng, L., Lee, L. J., Attias, H., and Acero, A. (**2007**). "Adaptive Kalman filtering and smoothing for tracking vocal tract resonances using a continuous-valued hidden dynamic model," IEEE Trans. Audio, Speech, Lang. Process. **15**(1), 13–23.

Deng, L., and Ma, J. (**2000**). "Spontaneous speech recognition using a statistical coarticulatory model for the vocal-tract-resonance dynamics," J. Acoust. Soc. Am. **108**(6), 3036–3048.

Dissen, Y., and Keshet, J. (**2016**). "Formant estimation and tracking using deep learning," in *INTERSPEECH*, pp. 958–962.

Duchi, J., Hazan, E., and Singer, Y. (**2011**). "Adaptive subgradient methods for online learning and stochastic optimization," J. Mach. Learn. Res. **12**, 2121–2159.

Ferguson, S. H., and Kewley-Port, D. (**2002**). "Vowel intelligibility in clear and conversational speech for normal-hearing and hearing-impaired listeners," J. Acoust. Soc. Am. **112**(1), 259–271.
Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., and Pallett, D. S. (**1993**). "DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1," NASA STI/Recon Technical Report No. 93.
Graves, A., Mohamed, A.-R., and Hinton, G. (**2013**). "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6645–6649.
Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., and Ng, A. Y. (**2014**). "Deep speech: Scaling up end-to-end speech recognition," arXiv:1412.5567.
Hernando, J., Nadeu, C., and Mariño, J. (**1997**). "Speech recognition in a noisy car environment based on LP of the one-sided autocorrelation sequence and robust similarity measuring techniques," Speech Commun. **21**(1), 17–31.
Hillenbrand, J., Getty, L. A., Clark, M. J., and Wheeler, K. (**1995**). "Acoustic characteristics of American English vowels," J. Acoust. Soc. Am. **97**(5), 3099–3111.
Hubel, D. H., and Wiesel, T. N. (**1968**). "Receptive fields and functional architecture of monkey striate cortex," J. Physiol. **195**(1), 215–243.
Kopec, G. E. (**1986**). "Formant tracking using hidden Markov models and vector quantization," IEEE Trans. Acoust., Speech Signal Process. **34**(4), 709–729.
Lathuilière, S., Mesejo, P., Alameda-Pineda, X., and Horaud, R. (**2018**). "A comprehensive analysis of deep regression," arXiv:1803.08450.
Lee, M., Van Santen, J., Möbius, B., and Olive, J. (**2005**). "Formant tracking using context-dependent phonemic information," IEEE Trans. Speech Audio Process. **13**(5), 741–750.
McCandless, S. S. (**1974**). "An algorithm for automatic formant extraction using linear prediction spectra," IEEE Trans. Acoust., Speech Signal Process. **22**(2), 135–141.
Medan, Y., and Yair, E. (**1989**). "Pitch synchronous spectral analysis scheme for voiced speech," IEEE Trans. Acoust., Speech Signal Process. **37**(9), 1321–1328.
Mehta, D. D., Rudoy, D., and Wolfe, P. J. (**2012**). "Kalman-based autoregressive moving average modeling and inference for formant and antiformant tracking," J. Acoust. Soc. Am. **132**(3), 1732–1746.
Mermelstein, P. (**1978**). "Difference limens for formant frequencies of steady-state and consonant-bound vowels," J. Acoust. Soc. Am. **63**(2), 572–580.
Monsen, R. B. (**1978**). "Toward measuring how well hearing-impaired children speak," J. Speech, Lang., Hear. Res. **21**(2), 197–219.
Munson, B., and Solomon, N. P. (**2004**). "The effect of phonological neighborhood density on vowel articulation," J. Speech, Lang., Hear. Res. **47**(5), 1048–1058.
Murthy, H. A., and Yegnanarayana, B. (**2011**). "Group delay functions and its applications in speech technology," Sadhana **36**(5), 745–782.
O'Shaughnessy, D. (**2007**). "Formant estimation and tracking," in *Springer Handbook of Speech Processing*, edited by J. Benesty, M. M. Sondhi, and Y. Huang (Springer, Berlin, Heidelberg).
Peterson, G. E., and Barney, H. L. (**1952**). "Control methods used in a study of the vowels," J. Acoust. Soc. Am. **24**(2), 175–184.
Pierrehumbert, J. B., Bent, T., Munson, B., Bradlow, A. R., and Bailey, J. M. (**2004**). "The influence of sexual orientation on vowel production (L)," J. Acoust. Soc. Am. **116**(4), 1905–1908.
Ribas Gonzalez, D., Lleida Solano, E., de Lara, C., and Jose, R. (**2014**). "Zero phase speech representation for robust formant tracking," in *2014 Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)*, pp. 1462–1466.
Sainath, T. N., Mohamed, A.-R., Kingsbury, B., and Ramabhadran, B. (**2013**). "Deep convolutional neural networks for LVCSR," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8614–8618.
Schiel, F., and Zitzelsberger, T. (**2018**). "Evaluation of automatic formant trackers," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC* 2018), European Language Resources Association (ELRA), Miyazaki, Japan.
Shadle, C. H., Nam, H., and Whalen, D. (**2016**). "Comparing measurement errors for formants in synthetic and natural vowels," J. Acoust. Soc. Am. **139**(2), 713–727.
Sjölander, K., and Beskow, J. (**2000**). "Wavesurfer—An open source speech tool," in *Interspeech*, pp. 464–467.
Smiljanić, R., and Bradlow, A. R. (**2005**). "Production and perception of clear speech in Croatian and English," J. Acoust. Soc. Am. **118**(3), 1677–1688.
Titze, I. R., Baken, R. J., Bozeman, K. W., Granqvist, S., Henrich, N., Herbst, C. T., Howard, D. M., Hunter, E. J., Kaelin, D., Kent, R. D., Kreiman, J., Kob, M., Löfqvist, A., McCoy, S., Miller, D. G., Noé, H., Scherer, R. C., Smith, J. R., Story, B. H., Švec, J. G., Ternström, S., and Wolfe, J. (**2015**). "Toward a consensus on symbolic notation of harmonics, resonances, and formants in vocalization," J. Acoust. Soc. Am. **137**(5), 3005–3007.
Toledano, D. T., Villardebó, J. G., and Gómez, L. H. (**2006**). "Initialization, training, and context-dependency in hmm-based formant tracking," IEEE Trans. Audio, Speech, Lang. Process. **14**(2), 511–523.
Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-C. (**2015**). "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems – Volume 1*, pp. 802–810.